

Microsoft® Operating System/2

Setup Guide

Information in this document is subject to change without notice and does not represent a commitment on the part of Microsoft Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. The purchaser may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use without the written permission of Microsoft Corporation.

Copyright Microsoft Corporation, 1987. All rights reserved. Simultaneously published in the U.S. and Canada.

Microsoft®, the Microsoft logo, MS®, MS-DOS®, and XENIX® are registered trademarks of Microsoft Corporation.

IBM® is a registered trademark of International Business Machines Corporation.

Intel® is a registered trademark of Intel Corporation.

Lotus® is a registered trademark of Lotus Development Corporation.

Contents

Welcome v

- System Requirements vi
- Notational Conventions vii
- How to Use This Guide vii

1 Installing MS OS/2 1

- Starting the Install Program 2
- Partitioning the Hard Disk 3
- Formatting the Hard Disk 4
- Creating Four Special Files 5

2 Customizing Your System with CONFIG.SYS 7

- What is a Configuration File? 8
- Configuration Commands 8
- A Sample CONFIG.SYS File 11
- Break 12
- Buffers 13
- Codepage 14
- Country 16
- Device 18
- Devinfo 20
- Diskcache 22
- Fcbs 23
- Iopl 24
- Libpath 25
- Maxwait 26
- Memman 27
- Pauseonerror 29
- Priority 30
- Protectonly 31
- Protshell 32
- Rem 33
- Rmsize 34
- Run 36
- Shell 37
- Swappath 39
- Threads 40
- Timeslice 41

3 Using Startup Files 43

- How Startup Files Work 45
- Creating Startup Files 46
 - Modifying AUTOEXEC.BAT with Edlin 46

Creating a new STARTUP.COMD with the Copy command 48
Using INITENV.COMD 49
How CONFIG.SYS Differs from These Files 50

- 4 Creating Partitions with Fdisk 51**
Checking for a Configured and Formatted Disk 52
Starting Fdisk 52
Creating an MS OS/2 Partition 53
 Creating More than One MS OS/2 Partition 55
 Creating an Extended MS OS/2 Partition 56
 Creating Logical Drives on Extended Partitions 57
Changing the Active Partition 58
Deleting an MS OS/2 Partition 59
 Deleting a Primary MS OS/2 Partition 60
 Deleting a Logical Drive 60
Displaying Partition Data 61
Selecting the Next Fixed-Disk Drive 62

- 5 Installing Device Drivers 63**
ANSI.SYS 64
COM.SYS 65
EGA.SYS 66
EXTDSKDD.SYS 67
MOUSEA_{xx}.SYS 69
POINTDD.SYS 71
VDISK.SYS 72

- Appendix A ANSI Escape Sequences 75**
Cursor Functions 76
Erase Functions 77
Screen Graphics Functions 77

- Appendix B Using Code Pages 79**
National Language Support Codes 80
National Language Support Commands 81
Setting the System Code Page 82
 Switching Between Code Pages 84
 Preparing Device Code Pages 84

Index 87

Welcome

Welcome to Microsoft® Operating System/2 (MS® OS/2). In this manual, you will learn how to install MS OS/2 and set up your system. If someone has already installed MS OS/2 for you, you may not need to use this manual now. But as you become more proficient with MS OS/2, or as your needs change, you may want to refer to this manual so that you can customize the way MS OS/2 works for you.

If you have used MS-DOS®, you'll find that many MS OS/2 commands work just like the MS-DOS commands you already know. If you've never used MS-DOS before, read the *Microsoft Operating System/2 Beginning User's Guide*.

System Requirements

Before you begin

To use MS OS/2, you should have the following:

- A 16-bit personal computer that runs MS OS/2 and has at least 512 kilobytes (K) of memory (one megabyte or more is recommended)
- At least two disk drives (one hard-disk drive and one high density floppy-disk drive)
- The MS OS/2 program disk
- A supplemental disk
- The *Microsoft Operating System/2 Setup Guide* (this manual), which explains how to set up MS OS/2 on a personal computer
- The *Microsoft Operating System/2 Beginning User's Guide*, which introduces you to the basics of using MS OS/2
- The *Microsoft Operating System/2 User's Reference*, which presents advanced features of MS OS/2 and includes a detailed description of MS OS/2 commands

In addition, three manuals discuss topics of interest to programmers:

- The *Microsoft Operating System/2 Programmer's Guide*, which introduces programmers to MS OS/2
- The *Microsoft Operating System/2 Programmer's Reference*, which describes all MS OS/2 system calls
- The *Microsoft Operating System/2 Device Drivers*, which describes how to use device drivers with MS OS/2 and explains device-driver commands

Notational Conventions

Throughout this manual, the following conventions are used to distinguish elements of text:

Text	Use
bold	Distinguishes commands, options, switches, and literal portions of syntax that must appear exactly as shown
<i>italic</i>	Distinguishes variables and placeholders that represent the kind of text to be entered by the user
monospace	Distinguishes sample command lines, program code and examples, and sample sessions
CAPITALS	Distinguish filenames, directory names, programming languages, and acronyms
SMALL CAPS [brackets]	Distinguish keys and key combinations Distinguish items that are optional. To include optional information, type only the information within the brackets, not the brackets themselves.
...(ellipsis)	Distinguishes options that can be repeated in a command line as often as desired.

A plus sign (+) used between two keynames indicates that those keys must be pressed at the same time. For example, "Press CONTROL+C" means that you should press and hold down the CONTROL key while you press and release the C key. Then release the CONTROL key.

How to Use This Guide

The following topics are covered in this manual:

Turn to	To find out
Chapter 1	How to install MS OS/2 on your computer
Chapter 2	How to use a configuration file and CONFIG.SYS commands

Elements of text

About this manual

Chapter 3	How to create and use the startup files AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM
Chapter 4	How to create a primary or extended MS OS/2 partition, review partition data, and delete an MS OS/2 partition
Chapter 5	How to use MS OS/2 installable device drivers
Appendix A	How to use ANSI escape sequences
Appendix B	How to use code pages

1 Installing MS OS/2

Microsoft Operating System/2 comes with its own system-installation program. This program, called Install, configures your personal computer and installs MS OS/2 on your computer's hard disk.

In this chapter, you will learn the following:

- How to install MS OS/2 with the system-installation program
- What partitions are, and how MS OS/2 uses them
- Which MS OS/2 directories are created by the installation program
- How MS OS/2 uses CONFIG.SYS, AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM files

Starting the Install Program

To start the menu-driven Install program, place the MS OS/2 program disk in a high density floppy-disk drive and turn on your computer.

Install allows you to select one of four options:

- **Option 1** Use this option if you have a new personal computer with no operating system installed. This option formats your hard disk using Fdisk, and copies the MS OS/2 system files onto your hard disk. For more information about Fdisk, see Chapter 4, "Creating Partitions with Fdisk."
- **Option 2** Use this option if you have a PC with MS-DOS version 3.x already installed and you want to remove the old operating system and install the MS OS/2 system files. This option reformats your hard disk before installing the system files.
- **Option 3** Use this option if you have a PC with MS-DOS version 3.x already installed and you want to retain the old operating system and install the MS OS/2 system files in addition to MS-DOS 3.x.
- **Option 4** Use this option if you have a PC with an earlier version of MS OS/2 already installed on the hard disk. This option updates the system files.

Warning If you choose option 1 or option 2, your hard disk will be formatted. Formatting a disk destroys any existing files. Therefore, if you have any files on your computer that you want to save, you must back them up before you run Install with option 1 or option 2. See the *Microsoft Operating System/2 Beginning User's Guide* for information on backing up your files.

Install starts automatically and leads you through the following steps:

- Partitioning and formatting your hard disk (option 1 only)
- Creating directories
- Copying the MS OS/2 program and supplemental disks to your hard disk
- Creating customized CONFIG.SYS, AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM files for your system

Partitioning the Hard Disk

If you have a new computer or a computer on which you want the current operating system to be replaced by MS OS/2, the Install program formats your hard disk. To do this, it first creates a primary MS OS/2 partition on your hard disk. Hard disks have up to four separate sections, called partitions. Partitions separate your hard disk into individual areas, each of which may contain a different operating system.

The following figure shows what your hard disk looks like:

[disk -- label:partition, sector, track, side]

Install sets aside space on your hard disk for use by MS OS/2. Install can create a primary MS OS/2 partition of up to 32 megabytes. If you want to allocate more space for MS OS/2, you can create an extended MS OS/2 partition with the Fdisk program.

MS OS/2 uses the letter of the physical hard-disk drive for your primary partition and assigns a new drive letter (for example, drive D) to the extended partition. MS OS/2 treats the new “logical” drive (the extended partition) the same as it treats physical drives. If your extended partition is large, MS OS/2 may require you to divide the partition into two or more logical drives. MS OS/2 assigns each new logical drive its own letter.

To use a logical drive, you must format it just as you would a physical drive. To access files kept on the logical drive, you must specify the logical-drive letter. For more information about creating an extended MS OS/2 partition, see Chapter 4, “Creating Partitions with Fdisk.”

What are partitions?

Formatting the Hard Disk

Once the hard disk is partitioned, the Install program formats the hard disk for MS OS/2. Formatting organizes the disk so that MS OS/2 knows where to look on the disk for certain key pieces of information, such as the File Allocation Table and the root directory. For more information about formatting disks, see the *Microsoft Operating System/2 Beginning User's Guide*.

MS OS/2 directories

Before it copies the MS OS/2 disks to your hard disk, Install creates these directories:

- \OS2
- \OS2\DEV
- \OS2\LIB
- \OS2\BIN
- \OS2\RBIN
- \OS2\PBIN

Install then prompts you to insert the program and supplemental disks one at a time into the disk drive. It copies the files from the program disk into your root directory. The device-driver files from the supplemental disk are copied into the \OS2\DEV directory. The \OS2\LIB directory contains dynalink libraries. Various MS OS/2 utilities are written to the \OS2\BIN, \OS2\RBIN, and \OS2\PBIN directories. The RBIN directory contains real-mode-only utilities; the PBIN directory holds protected-mode-only utilities.

Note Protected mode is an operating environment within MS OS/2 that lets you run more than one program at a time. When you run several programs at once, MS OS/2 “protects” one program from being affected by another. Real mode does not have the protection feature; therefore, only one application can run at a time in real mode. In real mode, you can run applications and other programs designed for MS-DOS version 3.x or earlier.

Creating Four Special Files

Next, Install creates four files: CONFIG.SYS, AUTOEXEC.BAT, INITENV.COMD, and STARTUP.COMD.

The first file, CONFIG.SYS, contains special configuration commands that tell MS OS/2 how your computer is set up. Install lets you choose whether to customize your CONFIG.SYS file or accept the default settings. For more information about setting up your CONFIG.SYS file, see Chapter 2, "Customizing your System with CONFIG.SYS."

The first time you start a protected-mode session, MS OS/2 searches for the STARTUP.COMD file in the root directory of the disk from which you started MS OS/2. When MS OS/2 reads this file, it executes a list of commands contained in the file. These commands may, for example, start one or more protected-mode applications you want to run on a regular basis.

The INITENV.COMD file includes several commands that MS OS/2 reads and executes in each new protected-mode session. This file is used to set up operating-environment variables like the search paths for commands, programs, and data files.

The first time you switch to the real-mode session, MS OS/2 searches for the AUTOEXEC.BAT file. This file contains commands similar to those in STARTUP.COMD and INITENV.COMD, but its commands are executed for the real-mode session only.

In the AUTOEXEC.BAT file, Install writes the following **path** command to reflect the directory structure that Install creates:

```
path c:\os2\bin;c:\os2\rbin
```

The following **path** command is written in the INITENV.COMD file:

```
path c:\os2\bin;c:\os2\pbin
```

Note The CONFIG.SYS, AUTOEXEC.BAT, INITENV.COMD, and STARTUP.COMD files are written to the root directory of your hard disk. Do not relocate them to another directory; MS OS/2 expects to find them in the root directory. For more information about these four files, see Chapter 2, "Customizing Your System with CONFIG.SYS," and Chapter 3, "Using Startup Files."

How these files work

1

2

3

2 Customizing Your System with CONFIG.SYS

Microsoft Operating System/2 uses a series of default settings that tell it how to operate on your computer. These settings may be changed to customize how MS OS/2 works with your computer. For example, if you work in France, you can change the default keyboard setting so that the character set for the United States is replaced with the character set for France.

When you use the Install program to put MS OS/2 on your system, the CONFIG.SYS file is automatically created. This file includes all of the changes to default settings that you specify when MS OS/2 is installed. You may find later that you want to change the configuration-command settings found in this file.

In this chapter, you will learn the following:

- What the configuration file, CONFIG.SYS, is and how MS OS/2 uses it
- What each of the CONFIG.SYS commands does, and how to use them to customize your system configuration

What is a Configuration File?

The configuration file, CONFIG.SYS, contains commands that affect how MS OS/2 runs on your computer. Each time you start MS OS/2, it searches the root directory of the drive in which it was started for the CONFIG.SYS file.

You can edit your CONFIG.SYS file to change your system's default configuration settings. For example, you can add installable device drivers to your system by including special commands in your CONFIG.SYS file.

If you want to change your CONFIG.SYS file, you can use a line editor. Remember that MS OS/2 performs the commands in the CONFIG.SYS file only when you first start the system. If you make changes to this file or if you create a new CONFIG.SYS file, you must restart MS OS/2 to take advantage of the new or changed file.

Note You need only one CONFIG.SYS file to configure your system to operate in both real mode and protected mode. MS OS/2 reads this file only when the you start the system by turning on your computer or when you restart the system by pressing CONTROL+ALT+DELETE. However, some of the CONFIG.SYS commands work only in real mode. The real-mode commands are noted in their command descriptions, included later in this chapter.

Configuration Commands

Frequently used commands

There are three types of configuration commands. The first group of commands includes those that you are most likely to need. Feel free to experiment with these commands in your CONFIG.SYS file to see which settings work best for you. This set of configuration commands includes the following:

break	pauseonerror
buffers	rem
device	rmsize
diskcache	run
fcbs	swappath
libpath	

The second group includes commands that the average MS OS/2 user should not adjust unless required to by an application. In most cases, the existing default values for these commands are appropriate. These commands include the following:

iopl	protectonly
maxwait	shell
memman	threads
priority	timeslice
protshell	

Little-used commands

The last set of commands are country- or language-dependent. If you live in, or work with, a country other than the United States, you may need to include one or more of these commands in your CONFIG.SYS file:

codepage
country
devinfo

Country-dependent commands

Note Two MS-DOS configuration commands, **files** and **lastdrive**, are ignored in your MS OS/2 CONFIG.SYS file.

The following table briefly describes the purpose of each configuration command:

Command	Purpose
break	Sets a CONTROL+C check in real mode only.
buffers	Sets the number of disk buffers that MS OS/2 allocates in memory when you start the system.
codepage	Selects the system code pages that MS OS/2 creates for code-page switching.
country	Allows MS OS/2 to use international time, date, currency, and date conversions.
device	Installs a specified device driver on the system list.

Defining CONFIG.SYS commands

devinfo	Prepares a device for code-page switching.
diskcache	Allocates space for a disk cache.
fcbs	Determines the number of file-control blocks (FCB) that can be open concurrently in real mode only.
iopl	Allows data input/output privilege to a process that requests it.
libpath	Specifies the location of dynamic-link modules.
maxwait	Sets the maximum allowable time that a process must wait before it runs.
memman	Selects memory-management options.
pauseonerror	Determines whether the system should pause if an error is found in the CONFIG.SYS file.
priority	Defines how MS OS/2 sets priorities for process scheduling.
protectonly	Selects the operation mode.
protshell	Loads and starts an alternate program-selector program.
rem	Allows comments in the CONFIG.SYS file.
rmsize	Sets the amount of memory to be reserved for real-mode applications.
run	Loads and starts a system (or background) process during system initialization.
shell	Loads an alternate real-mode command processor.
swappath	Specifies the location of the swap file.
threads	Determines the maximum number of threads for your computer.
timeslice	Sets timeslice values for process scheduling.

Note All commands in this section except **break** must be run from within the CONFIG.SYS file and therefore cannot be executed from the MS OS/2 prompt.

A Sample CONFIG.SYS File

A typical configuration file might look like this:

```

buffers=30
device=c:\dev\network.sys
break=on
shell=c:\bin\command.com c:\bin /p
protectonly=no

```

The following explains how this sample CONFIG.SYS file configures the system:

What this file does

- **buffers=30** Sets the number of buffers in memory to 30.
- **device=c:\dev\network.sys** Sets the searchpath C:\DEV\NETWORK.SYS to find the device driver being added to the system. In this case, it is network software. Generally, you receive a disk with installation software when you purchase a new device. Make sure that the device driver is in the directory that you specify in the **device** command.
- **break=on** Makes it possible to use CONTROL+C in real mode to stop a program, depending on the program that you are running.
- **shell=c:\bin\command.com c:\bin /p** Sets the MS OS/2 command processor to the COMMAND.COM file located in the BIN directory on the disk in drive C.

c:\bin tells the command processor where to look for COMMAND.COM if it needs to reread the disk.

The **/p** switch is used by the COMMAND.COM shell. For more information about COMMAND.COM and its switches, see the *Microsoft Operating System/2 User's Reference*.

- **protectonly=no** Allows both real-mode and protected-mode applications to run.

The next several pages discuss more fully how each command can be used. The commands are listed alphabetically.

**CONFIG.SYS:
Break command**

Break

Real Mode only

Purpose:

Sets CONTROL+C check.

Syntax:

break=on

or

break=off

Default:

break=off

Comments:

Depending on the program you are running, you may use CONTROL+C to stop a process. (For example, you may use CONTROL+C to stop the sorting of a file.) Normally, MS OS/2 checks to see whether you have pressed CONTROL+C only while it is reading from your keyboard or sending something to your screen or printer. If you set **break** to **on**, MS OS/2 checks for CONTROL+C in real mode whenever a program issues a system call.

Notes:

The **break** command is a real-mode configuration command. It works only if the **protectonly** command in your CONFIG.SYS file is set to **no**, or if there is no **protectonly** command in the CONFIG.SYS file.

Example:

To turn on CONTROL+C checking, put the following line in your CONFIG.SYS file:

```
break=on
```

Buffers

**CONFIG.SYS:
Buffers
Command**

Purpose:

Sets the number of disk buffers that MS OS/2 allocates in memory at the time you start the system.

Syntax:

buffers=*x*

where:

x is a variable, ranging from 1 to 100, that identifies the number of disk buffers.

Default:

buffers=10

Comments:

A disk buffer is a block of memory that MS OS/2 uses as a work space when it is reading or writing data.

For applications such as word processors, a number between 10 and 20 provides the best performance. If you plan to create many subdirectories, you may want to increase the number of buffers to between 20 and 30. Remember, though, that buffers take up 512 bytes of space. Thus, if you have more buffers, you have less memory available for applications.

Notes:

Feel free to experiment with different **buffer** settings to see how different values affect the way your computer operates.

Example:

To create 20 disk buffers, put the following line in your CONFIG.SYS file:

```
buffers=20
```

**CONFIG.SYS:
Codepage
command**

Codepage**Purpose:**

Selects the system code pages that MS OS/2 creates for code-page switching.

Syntax:

codepage=*xxx*[,*yyy*]

where:

xxx is a three-digit primary system code.

yyy is a three-digit secondary system code.

Default:

If you do not specify a value for the **codepage** command in the CONFIG.SYS file, MS OS/2 uses default values for your console screen, keyboard, and printer. MS OS/2 also uses default keyboard and country-code information unless these are otherwise specified with **codepage**.

Comments:

A code page is a table that defines the character set you are using. A character set is a country-specific or language-specific group of characters that are translated from the code-page table and displayed by your monitor or printer. An example of a character set is the set of letters, numbers, and symbols (such as accent marks) used by French-speaking Canadians.

The following code pages may be selected for *xxx* or *yyy*:

Value	Code Page
437	United States
850	Multilingual
860	Portuguese
863	French-Canadian
865	Nordic

Notes:

- For each device for which the selected code pages are to be prepared, you must include a corresponding **devinfo** command in the CONFIG.SYS file. See the section on **devinfo** later in this chapter for information on that command.

- For more information about code pages, see Appendix B, "Using Code Pages."

Example:

The following example sets the primary code page to Portuguese and the secondary code page to Multilingual:

```
codepage=860,850
```

If you included this statement in your CONFIG.SYS file, MS OS/2 would support applications that used either of these two character sets. MS OS/2 automatically loads the primary code page (860 in this case) and allows you to select the secondary code page (850) when you use the MS OS/2 **chcp** command. For more information about the **chcp** command, see Appendix B, "Using Code Pages."

**CONFIG.SYS:
Country
command**

Country
Purpose:

Customizes the time, date, and currency conventions for a particular country or language.

Syntax:

country=xxx

where:

xxx is a variable country code.

Default:

country=001

Comments:

This configuration command identifies which country's character set you intend to use. You must include a **country** command in your CONFIG.SYS file to use a character set for a country other than the United States (country=001).

The following list contains the valid country codes and their corresponding country or language:

<u>Code/Country</u>	<u>Code/Country</u>
001 United States	047 Norway
002 Canada (French)	049 Germany
003 Latin America	061 Australia (English)
031 Netherlands	081 Japan
032 Belgium	082 Korea
033 France	086 China (PRC)
034 Spain	088 Taiwan
039 Italy	099 Asia (English)
041 Switzerland	351 Portugal
044 United Kingdom	358 Finland
045 Denmark	785 Arabic-speaking countries
046 Sweden	972 Hebrew-speaking countries

Notes:

- Some countries and languages listed are not yet supported by MS OS/2.

- Swiss-French and Swiss-German both use country code 041.
- For more information about using code pages and country codes, see Appendix B, “Using Code Pages.”

Example:

The following example sets the **country** value to 033 (France) and converts international currency, time, date, and case to French conventions:

```
country=033
```

CONFIG.SYS:
Device command

Device**Purpose:**

Installs an additional device driver and identifies it to MS OS/2.

Syntax:

device=[*drive:*][*path*] *filename*[*arguments*]

where:

arguments are valid switches or other variables for the device driver.

Default:

None.

Comments:

The **device** command allows you to install device drivers in addition to the standard default drivers loaded by MS OS/2. These default drivers support the console, printer, standard floppy-disk and hard-disk drives, and the clock.

MS OS/2 provides the following device drivers that can be installed with the **device** command:

ANSI.SYS	MOUSE xx .SYS
COM.SYS	POINTDD.SYS
EGA.SYS	VDISK.SYS
EXTDSKDD.SYS	

These device drivers are described in Chapter 5, "Installing Device Drivers."

Notes:

- You may include more than one **device** command line in the CONFIG.SYS file. These **device** command lines will be processed by MS OS/2 in the order that they appear in CONFIG.SYS.
- All **device** commands are processed before any **run** commands in CONFIG.SYS.
- Generally, you receive a disk with an installable device driver when you purchase a new device. Make sure that the device driver is in the directory that you specify in the **device** command.

Example:

If you use the Microsoft InPort mouse with your computer, you must include the following command in your CONFIG.SYS file:

```
device=mousea04.sys
```

For more information about the installable device driver MOUSEA04.SYS, see Chapter 5, "Installing Device Drivers."

**CONFIG.SYS:
Devinfo
command**

Devinfo
Purpose:

Prepares a device for code-page switching.

Syntax:

devinfo=*devtype*,*subtype*,[*path*]*filename*,
[**ROM**=[(*xxx*,*yyy*)]]

where:

devtype is the type of device.

subtype is the style or model of the device.

filename identifies the code page table to be used for this device.

xxx and *yyy* are primary and secondary code pages, respectively.

Default:

None.

Comments:

The **devinfo** command provides MS OS/2 with information about the devices you are using with your computer. This command allows a device to accept code page switching.

The following list contains the *devtype* and *subtype* values you can use to prepare your keyboard, screen, or printer for code-page switching:

Device	Devtype	Subtype
Keyboard	KBD	<i>keyboard code</i>
Screen	SCR	EGA, VGA, LCD
Printer	PRN, LPT1, LPT2, LPT3	4201, 5202

For the keyboard (KBD), the *subtype* value is the selected keyboard code, which can also be used with the MS OS/2 **keyb** command. For more information, see the list of keyboard codes in Appendix B, "Using Code Pages."

Four device code-page files are supplied with MS OS/2:

- The 4201.DCP file contains a printer translation table for the model 4201 printer.

**Device-type
options**

- The 5202.DCP file contains a printer translation table for the model 5202 printer.
- The KEYBOARD.DCP file contains a set of keyboard translation tables.
- The VIOTBL.DCP file contains a screen translation table.

These files contain tables of screen or printer code pages, or keyboard translation tables, as applies to the specific device. Other device code-page files may be provided by your hardware device vendor.

If the file does not contain tables for the physical device (*subtype*) and for the code pages selected with the **codepage** command, the device is not prepared for that code page. If the file's path is omitted, MS OS/2 assumes that the file is installed in the root directory of the drive from which you started MS OS/2.

The **ROM** keyword is used when the computer or printer includes code-page information in the read-only memory, or in the case of certain printers, in the font cartridge. This keyword is valid only for parallel-printer (PRN, LPT1, LPT2, LPT3) and screen (SCR) device drivers.

For the printer, each **ROM** specification may be a pair of values enclosed in parentheses, where the first value is the code page and the second is the font identification number for each ROM or printer cartridge. If multiple fonts or font identification numbers are omitted, **devinfo** assumes that the font identification number is zero.

Printer ROM options

Notes:

- You may include more than one **devinfo** command line in
- For more information about using code pages and keyboard codes, see Appendix B, "Using Code Pages."

Example:

To prepare an EGA monitor for code-page switching, you might include a command line like this in your CONFIG.SYS file:

```
devinfo=scr,ega,c:\os2\viotbl.dcp
```

**CONFIG.SYS:
Diskcache
command**

Diskcache**Purpose:**

Allocates space for a disk cache.

Syntax:

diskcache=*n*

where:

n is the number of kilobytes of storage to be allocated to the disk cache, ranging from 64 to 7200 kilobytes.

Default:

No cache space is allocated.

Comments:

Disk caching is a method of data storage that allows the system to retrieve data instantly. When a cache is allocated, the system recalls what it read from hard disks, usually on a sector-by-sector basis, and consults that data prior to actually reading the disk. When the system finds the data it needs in the cache, its response rate is much quicker than if it reads the data from the disk.

Part of the space allocated for the cache is used by MS OS/2 for control information. The size of this control-information space varies based on the size of the cache itself and the amount of hard-disk storage on the system.

If there is no **diskcache** command in CONFIG.SYS, MS OS/2 will not enable caching and no cache space is allocated.

The cache size is not dynamically adjustable.

Example:

To allocate a 512K disk cache on your system, you would put the following line in your CONFIG.SYS file:

```
diskcache=512
```

Fcbs

CONFIG.SYS:
Fcbs command

Real Mode Only

Purpose:

Sets the number of file control blocks (FCBs) that can be open concurrently.

Syntax:

fcbs=*x,y*

where:

x specifies the number of files that MS OS/2 can open at any one time; the allowed values range from 1 to 255. *y* specifies the number of files opened by FCBs that MS OS/2 cannot close automatically; the allowed values range from 0 to 255, where *y* is less than or equal to *x*.

Default:

fcbs=16,8

Comments:

A file control block is a data structure in real mode that is used to control open files.

If an application tries to open more than *x* files by FCBs, all but the first *y* files may be closed by MS OS/2.

Notes:

- The preferred method of accessing files is to use file handles instead of file control blocks. However, some older applications may require you to use the **fcbs** command in your CONFIG.SYS file. You should only use the **fcbs** command if a real-mode application requires you to do so.
- The **fcbs** command is a real-mode configuration command. It works only if the **protectonly** command in your CONFIG.SYS file is set to **no**. **Protectonly=no** lets you run both real-mode and protected-mode applications on your computer.

Example:

To open as many as four files by FCBs and to protect the first two files from being closed, put the following line in your CONFIG.SYS file:

```
fcbs=4, 2
```

**CONFIG.SYS:
iopl command****iopl****Purpose:**

Gives data input/output privilege to a process that requests it.

Syntax:

iopl=yes

or

iopl=no

Default:

iopl=yes

Comments:

The **iopl** command lets you control read-and-write activity to certain programs and associated data.

Iopl=yes grants data input/output privilege; **iopl=no** denies it.

Notes:

Use the **iopl** command only if an application requires you to do so.

Example:

If you want to run an old application that is linked with IOPL segments and uses I/O instructions, you must include this line in your CONFIG.SYS file:

```
iopl=yes
```

Libpath

CONFIG.SYS:
Libpath
command

Purpose:

Specifies the location of dynamic-link (MS OS/2 interface) modules.

Syntax:

libpath=*drive:pathname*;*drive:pathname*[...]

Default:

The root directory of the disk from which you started MS OS/2.

Comments:

The **libpath** command specifies the directories that should be searched for loading dynamic-link libraries. Unlike the **path** environment variable, the default directory is not searched first.

MS OS/2 will search more than one directory if you specify several pathnames separated by semicolons. MS OS/2 will search the directories in the order specified.

Example:

If you wanted to put all dynamic-link library modules in the DYNLIB directory rather than in the root directory, you would include the following line in your CONFIG.SYS file to set the path that will allow your application access to the modules it needs:

```
libpath=c:\dynlibs
```

**CONFIG.SYS:
Maxwait
command**

Maxwait**Purpose:**

Sets the maximum time that a process has to wait before it runs.

Syntax:

maxwait=*x*

where:

x is the number of seconds that must elapse before a process may receive a boost in priority because it hasn't yet been recognized by the scheduler. The allowed values for *x* range from 1 to 255.

Default:

maxwait=3

Comments:

A process must receive permission from the MS OS/2 scheduler to run. When an active process is unable to run for the number of seconds defined by **maxwait**, the process receives a temporary increase in priority for one execution cycle (timeslice). For more information, see the section on the **timeslice** command in this chapter.

Example:

If you wanted waiting processes to get a priority boost every two seconds, you would include this line in your CONFIG.SYS file:

```
maxwait=2
```

Memman

CONFIG.SYS:
Memman
command

Purpose:

Selects memory-management options.

Syntax:

memman=noswap,nomove

or

memman=swap,move

or

memman=noswap,move

Default:

If the boot device is not a floppy disk:

memman=swap,move

If the boot device is a floppy disk:

memman=noswap,move

Comments:

Swap and **noswap** turn memory swapping on or off respectively. If memory swapping is on, MS OS/2 can use a time-sharing method to provide memory resources to a maximum number of processes. For example, if are running two processes, MS OS/2 may move some data from the first process into a temporary swap file to free storage for the second process. When the second process can be interrupted, MS OS/2 swaps the first process back. When swapping is off, system performance decreases.

Move and **nomove** refer to the ability to relocate data segments.

To ensure that MS OS/2 is responsive in some time-critical environments, you may need to disable swapping and/or segment motion. If you select swapping (**swap**), segment motion (**move**) is automatically enabled as well.

Notes:

Noswap and **nomove** should be specified only for dedicated systems, and should never be used in an office environment.

Example:

If your computer is controlling part of a factory's assembly line, the application program may require this line to be added to the CONFIG.SYS file:

```
memman=noswap,nomove
```

Pauseonerror

**CONFIG.SYS:
Pauseonerror
command**

Purpose:

Determines whether the system pauses if an error is found in the CONFIG.SYS file.

Syntax:

pauseonerror=yes

or

pauseonerror=no

Default:

pauseonerror=yes

Comments:

If you include **pauseonerror=yes**, MS OS/2 displays messages to tell you about any errors it finds in the CONFIG.SYS file. The system pauses and displays this message following a message about the error itself:

Press ENTER to continue

If you do not want the system to pause when it finds errors in CONFIG.SYS, type **pauseonerror=no**.

Example:

If you do not want a message to appear if MS OS/2 finds an unrecognizable command in your CONFIG.SYS file, you should put the following line in your CONFIG.SYS file:

pauseonerror=no

**CONFIG.SYS:
Priority
command****Priority****Purpose:**

Determines how a process receives enough priority over other processes to run.

Syntax:

priority=absolute

or

priority=dynamic

Default:

priority=dynamic

Comments:

Within MS OS/2 are three priority classes for processes: time-critical, normal, and idle-time. Each of these priority classes has 32 priority levels used by MS OS/2 to schedule processes. Within the normal class, the priority level may be adjusted.

If you type **priority=absolute** in your CONFIG.SYS file, MS OS/2 allots CPU resources on a first-come, first-served basis. If you type **priority=dynamic**, MS OS/2 tries to determine which process needs CPU resources most at any given interval of time (timeslice). If MS OS/2 finds a process that seems to have a lower priority than it ought to, and if you included the command line **priority=dynamic**, MS OS/2 increases the priority level of that process.

Notes:

The typical user will never need to use the command line **priority=absolute**.

Example:

If you want MS OS/2 to run processes on a first-come, first-served basis, you would put this line in your CONFIG.SYS file:

```
priority=absolute
```

Protectonly

**CONFIG.SYS:
Protectonly
command**

Purpose:

Selects the operation mode.

Syntax:

protectonly=yes

or

protectonly=no

Default:

protectonly=no

Comments:

The **protectonly** command lets you configure your system so that it allows only protected-mode applications to run.

Setting **protectonly** to **yes** means you can run only protected-mode applications; setting **protectonly** to **no** means you can run both real-mode and protected-mode applications. If you include the **protectonly=yes** command line, you will not be able to run old DOS applications written for DOS version 3.x or earlier.

Example:

If you know that you will run only protected-mode applications, include the following line in your CONFIG.SYS file:

```
protectonly=yes
```

**CONFIG.SYS:
Protshell
command**

Protshell**Purpose:**

Loads and starts a program selector other than the MS OS/2 program selector included with the system.

Syntax:

```
protshell=[drive:][path]filename [argument]
```

where:

argument gives the drive, path, and filename of the command interpreter for protected-mode sessions.

Default:

```
protshel=shell.exe
```

Comments:

A program selector lets you start and switch between sessions. The **protshell** command in your CONFIG.SYS file tells MS OS/2 to use a program selector other than the program selector included with MS OS/2.

You must include *filename* to specify the filename of the program selector.

For more information about sessions and the MS OS/2 program selector, see the *Microsoft Operating System/2 User's Reference*.

Example:

Suppose you created a new program selector called NEWSHELL.EXE to replace the MS OS/2 program selector. To use your new program selector, which is located on the disk in drive D, you would include the following line in your CONFIG.SYS file:

```
protshell=d:\newshell.exe
```

Rem

**CONFIG.SYS:
Rem command****Purpose:**

Allows comments in the CONFIG.SYS file.

Syntax:

rem *text*

where:

text is any remark you would like to add to the CONFIG.SYS file.

Default:

None.

Comments:

Remarks included in CONFIG.SYS with **rem** are not displayed on the screen when MS OS/2 reads the CONFIG.SYS file. Remarks are not commands; they are meant as documentation aids within the file itself.

Example:

If you create an alternative CONFIG.SYS file for use in special situations, you might want to include **rem** command lines to describe when and how the alternative file should be used. The following **rem** lines might be included in an alternative CONFIG.SYS file to be used when a particular printer is connected to COM1:

```
rem This file (config.alt) is the alternate
rem configuration file used when the
rem LaserZoom printer is connected to COM1.
rem Rename the primary configuration
rem file (config.sys) to config.pri, then
rem rename this file to config.sys
rem and reboot.
```

**CONFIG.SYS:
Rmsize
command****Rmsize****Purpose:**

Sets the amount of memory to be reserved for real-mode applications.

Syntax:

rmsize=*x*

where:

x is a number ranging from 0 to 640 that represents the memory boundary in kilobytes.

Default:

The default depends on the memory size of your computer. See the following comments.

Comments:

The default memory boundary depends on your computer's total memory. If the total memory is 640K or less, the default size is the total memory minus the minimum memory required for protected-mode operations. If the total memory is 1024K or greater, the default size is the amount of memory installed below 1024K, either 512K or 640K.

Notes:

- MS OS/2 uses some of the low memory set aside by the **rmsize** command. Therefore, if you type **rmsize=640**, the amount of space actually left for real-mode applications is 640K minus the amount of space used by MS OS/2.
- The **rmsize** command is a real-mode configuration command. It works only if the **protectonly** command in your CONFIG.SYS file is set to **no**. This setting allows both real-mode and protected-mode applications to run on your computer.
- For information about setting the real-mode environment size, see the section on the **shell** command in this chapter.

Example:

If you want to reserve 512K of memory for use by processes running in real mode, type this line in your CONFIG.SYS file:

```
rmsize=512
```

**CONFIG.SYS:
Run command****Run****Purpose:**

Loads and starts an application or a system process (or background process) during system initialization.

Syntax:

```
run=[drive:][path]filename [arguments]
```

where:

arguments are valid switches or other variables for the application.

Default:

None.

Comments:

The filename of the program must be specified exactly; if the filename includes an extension, that extension must be included.

Notes:

- You may include more than one **run** command line in your CONFIG.SYS file.
- Before MS OS/2 processes **run** command lines, it processes all **device** command lines in the CONFIG.SYS file.
- CONFIG.SYS parsing preserves the case of filenames and parameters. This is important for some programs that are case-sensitive.

Examples:

To start an alarm-clock program called Alarm each time you start MS OS/2, you would include the following line in your CONFIG.SYS file:

```
run=alarm
```

The **run** command also allows you to include command-line arguments. If the alarm-clock application lets you choose a flashing screen or a beep as the alarm, you might include a line like this in your CONFIG.SYS file:

```
run=alarm /flash
```

Shell

CONFIG.SYS:
Shell command

Real Mode Only

Purpose:

Loads the real-mode command processor.

Syntax:

shell=[*drive:*]*pathname*[*arguments*]

where:

pathname specifies the program that MS OS/2 uses as a command processor.

arguments are valid switches or other variables for the real-mode command-processor program.

Default:

shell=command.com

Comments:

If you do not use the default, MS OS/2 starts the real-mode command processor specified by *pathname* instead of starting the standard COMMAND.COM command processor.

Notes:

- The **shell** command is a real-mode configuration command. It works only if the **protectonly** command in your CONFIG.SYS file is set to **no**. This setting allows both real-mode and protected-mode applications to run on your computer.
- The **shell** command itself does not accept switches. However, if the command processor accepts switches, they may be included in the command line as arguments. For example, if you are using the default setting, **shell=command.com**, you can include any of the switches accepted by command.com in the **shell** command line. Thus the following is a valid command:

```
shell=command.com /p
```

See the *Microsoft Operating System/2 User's Reference* for more information about COMMAND.COM and the switches it uses.

- The CONFIG.SYS **shell** command line does not affect any batch **set** environment commands, nor does it affect the **shell** command in BASIC.

Examples:

Suppose you had another command processor called NEWCMD.COM. The following command loads the \BIN\NEWCMD.COM file as the command processor:

```
shell=\bin\newcmd.com
```

The next example loads the default real-mode command processor, COMMAND.COM, with an environment size of 512 bytes. This environment size is used to load values from MS OS/2 commands like **prompt** and **path**.

```
shell=command.com c:\os2 /e:512
```

Swappath

**CONFIG.SYS:
Swappath
command**

Purpose:

Specifies the location of the disk swap file.

Syntax:

```
swappath=drive:[path]
```

Default:

```
swappath=c:\
```

Comments:

A swap file is a temporary file used to store data that has been moved to disk out of memory while another process runs. The **swappath** command designates the drive and path where the swap file will be written. MS OS/2 erases the swap file when the process using that file is completed.

Notes:

- For best results, and better processing time, specify a hard disk in the swap path. Writing swap files to floppy disks slows processing time.
- In order to use the **swappath** command, the **memman** command in the CONFIG.SYS file must allow swapping. For more information, see the section on the **memman** command earlier in this chapter.
- If you intend to use file swapping, be sure your disk has ample space available. The minimum swap-file size is 512K.

Example:

If you want MS OS/2 to write the swap file to a directory called DATA on drive C, include this line in your CONFIG.SYS file:

```
swappath=c:\data
```

**CONFIG.SYS:
Threads
command****Threads****Purpose:**

Determines the maximum number of threads for your computer.

Syntax:

threads=*x*

where:

x is the maximum number of threads that can be created at one time, ranging from 16 to 255.

Default:

threads=50

Comments:

A thread is a part of an application or other process that can be scheduled by MS OS/2 to run on its own. A process generally contains multiple threads, which act like small programs that perform a particular part of the process.

The default of 50 threads is sufficient for MS OS/2 system threads and a small number of simple applications. The **threads** value may need to be increased when you run more programs at the same time.

Notes:

- MS OS/2 creates and maintains approximately 14 threads as soon as you start the system.
- When the value of **threads** is increased, the amount of system resources (including memory) is decreased.

Example:

If you wanted to increase the number of threads from 50 (the default amount) to 64, you would include the following line in your CONFIG.SYS file:

```
threads=64
```

Timeslice

**CONFIG.SYS:
Timeslice
command**

Purpose:

Sets the minimum and maximum timeslice values for scheduling the threads of a process.

Syntax:

timeslice=*x*[,*y*]

where:

x is the minimum timeslice in milliseconds; it must be greater than 31.

y is the maximum timeslice value in milliseconds; it must be greater than or equal to *x*.

Default:

The timeslice is set by MS OS/2.

Comments:

A timeslice is an interval of time used by MS OS/2 to schedule the threads of a process. A process, such as an application, is often made up of several threads. Each one of these threads is like a small program that can be scheduled as a separate unit.

The **timeslice** command identifies the minimum and maximum amount of time that MS OS/2 may dedicate to a given process before it must check on other processes.

If you specify only one value, **timeslice** sets both minimum and maximum timeslice values.

Notes:

Unless an application specifically requires that the **timeslice** value be changed, it is not recommended that you do so.

Examples:

If you are running an application that advises you to change the **timeslice** value to 64 milliseconds, you would add the following line to your CONFIG.SYS file:

```
timeslice=64
```

If you included the following command line, the scheduler would wait one second (1000 milliseconds) before checking other threads to see if any has priority higher than the threads currently running:

```
timeslice=1000
```

3 Using Startup Files

When you start Microsoft Operating System/2, three files set up your computer's working environment: AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM. In this chapter, you will learn the following:

- How to create or modify AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM files
- How to use these files
- How these files differ from the CONFIG.SYS file

The following figure shows what happens when you start MS OS/2:

[Figure: What happens when you start MS OS/2]

How Startup Files Work

When you first use the Install program, MS OS/2 creates the three startup files AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM in your root directory. These three files are batch files: they contain a series of commands that tell MS OS/2 what to do. For more information about batch files, see the *Microsoft Operating System/2 User's Reference*.

When you start MS OS/2, it reads the STARTUP.COM file. MS OS/2 reads the AUTOEXEC.BAT file the first time you switch to the real-mode session; it reads the INITENV.COM file the first time you start a protected-mode session. MS OS/2 carries out the commands listed in these files.

If you want MS OS/2 to start a program or command automatically when you first start the real-mode session, include the command line in the AUTOEXEC.BAT file. Likewise, if you want a program or command to run when you start your first protected-mode session, enter that command line in the INITENV.COM file.

Note The **protectonly** command in your CONFIG.SYS file must be set to **no** in order for your AUTOEXEC.BAT file to run in real mode. For more information on changing your CONFIG.SYS file, see Chapter 2, "Customizing Your System with CONFIG.SYS."

The AUTOEXEC.BAT, INITENV.COM, and STARTUP.COM files let you use MS OS/2 more efficiently. For example, when MS OS/2 created these three files, it inserted the **path** command into each of them. **Path** lists the directories most likely to contain the command files MS OS/2 needs. By automating the search path, MS OS/2 has reduced the amount of interaction necessary with the user and so is able to work more efficiently.

Creating Startup Files

Viewing the startup files

MS OS/2 creates the AUTOEXEC.BAT, INITENV.CMD, and STARTUP.CMD files when you install the system with the Install program. You can look at the contents of these files by using your word processor, or by using the **type** command:

```
type autoexec.bat
```

or

```
type initenv.cmd
```

or

```
type startup.cmd
```

You can customize these files by using an editor such as Edlin, the real-mode line editor included with MS OS/2, or by using a word processor such as Microsoft Word. By adding new commands or changing the existing contents, you can customize how MS OS/2 starts real-mode and protected-mode sessions.

The following sections illustrate some of the methods for creating and modifying these files, and include some sample AUTOEXEC.BAT, INITENV.CMD, and STARTUP.CMD files.

Note The AUTOEXEC.BAT, INITENV.CMD, and STARTUP.CMD files must be located in the root directory of your MS OS/2 disk.

Modifying AUTOEXEC.BAT with Edlin

Using Edlin

You can use Edlin to customize your AUTOEXEC.BAT file. Edlin does not work in protected mode.

For example, if you want to load GW-BASIC and run a program called Menu automatically each time you start a real-mode session, you could use Edlin to modify your AUTOEXEC.BAT file as follows:

- 1 From the root directory, type the following command and then press the ENTER key:

```
edlin autoexec.bat
```

This command tells Edlin to edit AUTOEXEC.BAT.

- 2** When you see the Edlin prompt, an asterisk (*), type the letter **l** and press the ENTER key. Edlin lists all of the lines currently in the AUTOEXEC.BAT file. Your screen might look something like this:

```
[Edlin autoexec/'l' one line w/path command]
```

Note that Edlin uses line numbers to help you edit. These numbers are not actually part of your file.

- 3** Type **i** followed by the next line number and press the ENTER key. (For example, if the file already has three lines in it, type **i4**.) This allows you to insert new command lines following the existing lines in the file. Edlin will display a new line number:

```
[Edlin: add line 2]
```

- 4** Type the following lines. Use the BACKSPACE key to erase mistakes. Press the ENTER key at the end of each line, including the last line. Edlin will give you a new line number for each new line:

```
prompt [$p]
path c:\;c:\msdos;c:\basic
cls
gwbasic menu
```

- 5** Press CONTROL+C to return to the Edlin prompt.
6 Type the letter **e** to end the Edlin program and return to MS OS/2.

The AUTOEXEC.BAT file shown in this example does the following when you start the real-mode session:

- **prompt [\$p]** Sets your prompt to display the current drive and directory within brackets. You can use the **prompt** command to include other information in your real-mode prompt (or protected-mode prompt within INITENV.CMD). For example, the **prompt [\$t]** command line displays the time, and the **prompt [\$d]** command line displays the date. For more information about the **prompt** command, see the *Microsoft Operating System/2 User's Reference*.
- **path c:\;c:\msdos;c:\basic** Sets your command search path. MS OS/2 will first look in your working directory to find the program or other executable file when you type a command. If it is not there, MS OS/2 searches the root directory on drive C, then C:\MSDOS, and finally C:\BASIC to find the program you want to run.
- **cls** Clears the screen, except for the real-mode prompt.
- **gwbasic menu** Loads GW-BASIC and runs a program called Menu. (This is a fictitious program name.) To run your own BASIC program, type the name of your program in place of Menu.

Creating a new STARTUP.CMD with the Copy command

Using the Copy command

You can use the **copy** command to create a startup file. The following list explains how you can create a new STARTUP.CMD file that automatically runs three programs when you start your computer:

- 1 Rename your existing STARTUP.CMD file to STARTUP.BAK by typing the following command:

```
ren startup.cmd startup.bak
```

This way, you will have a backup copy of your original STARTUP.CMD file should you have any problem with the new file. If you do not rename your original STARTUP.CMD file, using the **copy** command to create a new STARTUP.CMD file will overwrite the existing file.

- 2 Type the following command and then press the ENTER key:

```
copy con startup.cmd
```

This command tells MS OS/2 to copy what you type from the keyboard into the STARTUP.CMD file. Remember that you must put STARTUP.CMD in the root directory of your MS OS/2 disk.

- 3 Type the following lines. Press the ENTER key after each line:

```
start c:\network\net
start c:\finance\market
c:\plan\agenda
```

- 4 After the last line, press CONTROL+Z and then press the ENTER key to copy these lines into the STARTUP.CMD file.

With the STARTUP.CMD file shown in this example, MS OS/2 does the following when you start your computer (the Net, Market, and Agenda programs are fictitious):

- **start c:\network\net** Begins a program called Net, located on drive C in the NETWORK directory. Net will run in its own session.
- **start c:\finance\market** Begins a program called Market in its own session. Because no **path** command is used in this STARTUP.CMD file, you must specify the location of these programs.
- **c:\plan\agenda** Begins a program called Agenda in the first session you see. This is called the foreground session. Other active sessions (for example, those started by the first two lines of this STARTUP.CMD file) are called background sessions, because you are not currently viewing them.

For more information about foreground and background processes, see the *Microsoft Operating System/2 User's Reference*.

Using INITENV.CMD

The INITENV.CMD file is initiated each time you start a new protected-mode session from the program-selector screen. Just as the AUTOEXEC.BAT file sets the environment in real mode, the INITENV.CMD file sets the environment in protected mode. Here is an example of an INITENV.CMD file:

**A sample
INITENV.CMD file**

```
path c:\;c:\os2\bin;c:\os2\pbin
prompt [%d $t $p]
dir /w
```

This INITENV.CMD file does the following:

- **path c:\;c:\os2\bin;c:\os2\pbin** Sets the command search path. After MS OS/2 searches your working directory to find the program you want to run, it searches the directories in the order listed in your **path** command line until the program is found.
- **prompt [%d \$t \$p]** Creates a custom protected-mode prompt that shows you the date (**%d**), time (**\$t**), and your working drive and directory (**\$p**). The prompt this command line creates looks something like this:


```
[Fri 1-22-1988 9:59:06.94 c:\os2]
```
- **dir /w** Displays a wide-format listing of the current directory.

How CONFIG.SYS Differs from These Files

MS OS/2 uses the AUTOEXEC.BAT, INITENV.CMD, and STARTUP.CMD files in a different way than it does the CONFIG.SYS file because they perform different types of commands. While AUTOEXEC.BAT, INITENV.CMD, and STARTUP.CMD can contain any MS OS/2 command or program, the CONFIG.SYS file can contain only a special set of configuration commands.

In addition, you must restart MS OS/2 to perform the commands in the CONFIG.SYS file. But to perform the commands in the AUTOEXEC.BAT file, for example, you simply type the following from the root directory in the real-mode session (at the prompt C>):

```
autoexec
```

And, to perform commands in the INITENV.CMD file, you would type the following from the root directory in a protected-mode session (at the prompt [C:]):

```
initenv
```

4 Creating Partitions with Fdisk

A hard disk (also known as a fixed disk) can be organized into one to four separate sections called partitions. Partitions separate your hard disk into individual areas, and each partition may contain a different operating system.

To prepare your hard disk for Microsoft Operating System/2, you must create a partition for it. You can create an MS OS/2 partition on your hard disk by using a menu-driven program called Fdisk (a fixed-disk setup program). You must use Fdisk if you want to do one of the following:

- Create a primary MS OS/2 partition
- Create an extended MS OS/2 partition
- Create a logical drive
- Change the primary partition
- Delete an MS OS/2 partition or logical drive
- Display information about an MS OS/2 partition or logical drive
- Review or modify the configuration of another hard disk on your computer

Warning Reconfiguring your disk with Fdisk destroys all existing files. Be sure to have a backup of all files on your disk before you create an MS OS/2 partition with Fdisk.

Checking for a Configured and Formatted Disk

Has Fdisk been run already?

Many computer stores configure hard-disk computers for MS OS/2, so you may not need to use Fdisk. They may also format your hard disk to start MS OS/2 when you turn the power on. To find out whether this has been done, try to start MS OS/2 from your hard disk. If it starts, your hard disk is both configured and formatted, and the MS OS/2 system files are on the disk. If MS OS/2 does not start, your disk is not formatted to start MS OS/2, but may have been configured.

If MS OS/2 did not start, check to see if the disk has been configured with Fdisk by doing the following:

- ❶ Insert the MS OS/2 program disk in drive A and press CONTROL+ALT+DELETE to start MS OS/2.
- ❷ Run Fdisk and select the Display Partition Data to see if any MS OS/2 partitions exist.
If any do exist, your disk has been configured. If no partitions exist, follow the instructions in this chapter to configure your disk.
- ❸ After you have configured your hard disk, be sure to format your disk by using the **format /s** command line before you copy files onto the disk. Otherwise, your files will be unreadable.

Starting Fdisk

Running Fdisk

The Fdisk program is easy to use because it uses menus to lead you through each procedure. To start Fdisk, follow these steps:

- ❶ Place the MS OS/2 program disk in drive A.
- ❷ Turn on your computer to start MS OS/2.
- ❸ Press the ENTER key to select the Start a Program option from the program-selector screen. For more information about the program selector, see the *Microsoft Operating System/2 User's Reference*.
- ❹ Type the following command and press the ENTER key:

```
fdisk
```

In response, Fdisk displays its main menu on your screen. This menu lists five options, as shown in the following example (if your computer has only one hard disk, option 5 will not appear on your screen):

Fdisk main menu

Disk Options

Choose one of the following:

1. Create Microsoft Operating System/2 partition or Logical Drive
2. Change primary partition
3. Delete Microsoft Operating System/2 partition
4. Display Partition Data
5. Select Next Fixed Disk Drive

Enter choice: [1]

Press ESC to return to Microsoft Operating System/2

The following sections describe each of these options and show the menus and other information they display. To return to MS OS/2 from the main menu, just press the ESCAPE key. You can also use the ESCAPE key to return to the main menu from any of the Fdisk menus.

Most of the Fdisk menus displays a default value. To choose the default value, press the ENTER key. To choose another value, just type the value you want, and press the ENTER key.

Note In some Fdisk screens, the primary MS OS/2 partition is referred to as the primary DOS partition (PRI DOS).

Creating an MS OS/2 Partition

If you choose the first option on the main menu, and if your hard disk is not yet completely partitioned, Fdisk displays a screen like the following (if no extended partitions exist, the third option on this screen is not displayed):

Fdisk main menu: option 1

Create Microsoft Operating System/2 Partition

1. Create Primary Microsoft Operating System/2 partition
2. Create Extended Microsoft Operating System/2 partition
3. Create logical drive(s) in the Extended Microsoft Operating System/2 partition

Enter choice: [1]

Press ESC to return to Fdisk Options

You must create a primary MS OS/2 partition before you can create any extended MS OS/2 partitions on your disk. The primary partition is the one from which MS OS/2 starts. In most cases, you will need only one MS OS/2 partition for your entire disk.

Creating an MS OS/2 partition

To create the primary MS OS/2 partition, press the ENTER key to accept the default selection (option 1).

The Create Primary Microsoft Operating System/2 Partition menu appears next:

Create Primary Microsoft Operating System/2 Partition

Do you wish to use the maximum size for a Microsoft Operating System/2 partition and make the Microsoft Operating System/2 partition active (Y/N).....? [Y]

Press ESC to return to Fdisk Options

If you use your entire hard disk for MS OS/2, you will use the Fdisk program only once to create the primary MS OS/2 partition.

If you want to use the entire hard disk (up to 32 megabytes) for MS OS/2, press the ENTER key to accept the default selection (Yes).

Fdisk then displays the following message:

The system will now restart.
Insert the Microsoft Operating System/2 diskette in drive A: and press Ctrl+Alt+Del.

Put your MS OS/2 disk in drive A and press any key to restart MS OS/2.

If you want to create a primary MS OS/2 partition that is smaller than the maximum size, type **n** and press the ENTER key. See the following section, “Creating More than One MS OS/2 Partition,” for more information.

Now that the MS OS/2 partition is created, you must format your hard disk so that MS OS/2 can use it. If you want to start MS OS/2 from your hard disk, remember to use the **/s** switch with the **format** command. For example, if you are formatting the disk in drive C and want to start MS OS/2 from that disk, type the following command:

```
format c: /s
```

For more information about the **format** command, see the *Microsoft Operating System/2 User's Reference*.

Creating More than One MS OS/2 Partition

You may choose to create a primary MS OS/2 partition that is smaller than the maximum size and then use the remainder of the disk as an extended MS OS/2 partition. To do this, type **n** in response to the question on the Create Primary Microsoft Operating System/2 Partition menu. Fdisk then displays a second Create Primary Microsoft Operating System/2 Partition menu like the following. From this menu, you can specify the size of the primary MS OS/2 partition:

```
Create Primary Microsoft Operating System/2
Partition
```

```
Partition  Status  Type  Start  End  Size
```

```
Total disk space is 732 cylinders.
Maximum space available for partition
is 732 cylinders.
```

```
Enter partition size..... [ 732]
```

```
Press ESC to return to Fdisk Options
```

The space available on your hard disk is measured in cylinders (also called tracks). This menu shows the total number of cylinders available for a hard-disk partition, and prompts you to enter the size of your new partition. The default size for the partition is the maximum available space on the hard disk. Press the ENTER key if you

Formatting your hard disk

Creating multiple MS OS/2 partitions

want the default size; otherwise, type the size (in cylinders) that you want for the partition, and press the ENTER key.

Any part of the disk that you do not use for the primary MS OS/2 partition may be used for an extended MS OS/2 partition, as discussed in the following section.

Creating an Extended MS OS/2 Partition

Configuring the rest of your disk

You can use Fdisk to create an extended partition if your disk is larger than 32-megabytes (the maximum partition size), or if you want to designate one or more logical drives for the disk.

To create an extended partition, you should select option 2 from the Create Microsoft Operating System/2 Partition menu, shown as follows:

Create Microsoft Operating System/2 Partition

1. Create Primary Microsoft Operating System/2 partition
2. Create Extended Microsoft Operating System/2 partition
3. Create logical drive(s) in the Extended Microsoft Operating System/2 partition

Enter choice: [1]

Press ESC to return to Fdisk Options

Type **2**, then press the ENTER key. In response, Fdisk displays a menu like this one:

Create Extended Microsoft Operating System/2 Partition

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	599	600

Total disk space is 1263 cylinders.
Maximum space available for partition is 663 cylinders.

Enter partition size..... [663]

Press ESC to return to Fdisk Options

This screen shows the total number of cylinders available for an extended partition. The default for the partition

size is the maximum available space on the hard disk. Press the ENTER key if you want the default; otherwise, type the size (in cylinders) that you want for the partition, and press the ENTER key.

Note If Fdisk finds any defective tracks at the start of the partition, it adjusts the partition boundaries to avoid those bad tracks.

Creating Logical Drives on Extended Partitions

When you have created an extended partition, you must specify one or more drive letters for that area of the disk. Fdisk automatically displays the following menu after you create an extended partition (Fdisk also displays a screen like the following if you select option 3 from the Create Microsoft Operating System/2 Partition menu):

Create Logical Drive(s)

Drv	Start	End	Size
D:	650	1049	400

Total partition space is 1000 cylinders.
Maximum space available for logical drive is 600 cylinders.

Enter logical drive size..... [600]

Press ESC to return to Fdisk Options

You may designate the entire partition as one logical drive, or you may choose to divide it into two or more logical drives. You may, for example, want to create a second logical drive on the partition if you want to segregate a particular application and its data files to its own drive.

Because you cannot use an MS OS/2 extended partition without a drive letter, Fdisk continues to prompt you for logical disk-drive information until the whole partition has been assigned to logical drives.

When the entire partition is assigned to logical drives, Fdisk displays this message:

```
All available space in the Microsoft
Operating System/2 Extended partition is
assigned to logical drives.
```

Press the ESCAPE key to return to the main Fdisk menu. From there, you can restart MS OS/2, or select another option.

Changing the Active Partition

Fdisk main menu: option 2

If you choose the second option on the main menu, Fdisk displays a screen showing the status of each partition on your hard disk. The active partition, indicated in the Status column by the letter "A," shows the partition from which MS OS/2 starts when you turn on or reset your computer. If you have created a partition on your disk with another operating system, this menu allows you to make that partition the active partition. Only one partition can be active at a time.

For example, if you have partitions for both XENIX and MS OS/2 on your disk, the Change Primary Partition screen might look like this:

Change Primary Partition

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	401	402
2		non DOS	402	731	330

Total disk space is 732 cylinders.

Enter the number of the partition you want to make active.....:[1]

Press ESC to return to Fdisk Options

Type the number of the partition that you want to activate, and press the ENTER key. The default setting is the active partition number.

If your hard disk contains only MS OS/2 partitions, Fdisk displays the following message instead of prompting you for the partition that you want to activate:

The only startup partition on Drive 1 is already marked active.

Press ESC to return to FDISK Options.

Deleting an MS OS/2 Partition

If you select option 3 on the main menu, Delete Microsoft Operating System/2 Partition, Fdisk displays the following screen, which asks you to identify whether the partition you want to delete is a primary or an extended MS OS/2 partition:

**Fdisk main menu:
option 3**

Delete Microsoft Operating System/2 Partition

1. Delete Primary Microsoft Operating System/2 partition
2. Delete Extended Microsoft Operating System/2 partition
3. Delete logical drive(s) in the Extended Microsoft Operating System/2 partition

Enter choice: []

Press ESC to return to Fdisk Options

Type the number next to the option you've selected and press the ENTER key. The next menu, whether you've chosen to delete a primary or extended MS OS/2 partition, shows the status of that partition. When you delete an MS OS/2 partition, Fdisk deletes the partition boundaries and any data that existed in that partition. Once you delete the partition, *you cannot recover the data that was on it.*

Note You cannot use Fdisk to delete a partition other than an MS OS/2 partition. In addition, to start a different operating system in another partition of your hard disk, you must change the active partition to that number *before* you delete the MS OS/2 partition.

Deleting a Primary MS OS/2 Partition

Removing a primary partition

The Delete Primary Microsoft Operating System/2 Partition menu looks similar to the following:

Delete Primary Microsoft Operating System/2 Partition.

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	399	400
2		EXT DOS	400	731	332

Total disk space is 732 cylinders.

Warning! Data in Primary Microsoft Operating System/2 partition will be lost. Do you wish to continue.....? [N]

Press ESC to return to Fdisk Options

If you do not want to delete the MS OS/2 partition, press the ENTER key to accept the default value (No).

To delete the MS OS/2 partition, do the following:

- 1 Type y.
- 2 Press the ENTER key.

Deleting a Logical Drive

Removing an extended partition

If you choose to delete an extended partition, you must first delete the logical drives associated with that partition.

To delete a logical drive, select option 3, Delete Logical Drive(s) in the Extended Microsoft Operating System/2 Partition, from the Delete Microsoft Operating System/2 Partition menu. Fdisk then displays a menu like the following:

Delete Logical Drive(s)

Drv	Start	End	Size
D:	400	999	600
E:	1000	1399	400

Total partition space is 1000 cylinders.

Warning! Data in the logical drive will be lost. What drive do you want to delete.....? []

Press ESC to return to Fdisk Options

Type the letter of the drive you want to delete and press the ENTER key. Fdisk displays this message:

Are you sure.....? [N]

If this logical drive contains valuable data you have not backed up, press the ENTER key. This stops Fdisk from deleting the logical drive.

Note Be sure to back up all files you need from the logical drive *before* you delete the drive. When Fdisk deletes a logical drive or partition, the data is destroyed.

If you are sure you want to delete the drive, type **y**.

Displaying Partition Data

If you choose option 4 on the main menu, Display Partition Data, Fdisk displays a screen that contains information about each of the partitions on your hard disk.

The Display Partition Information screen might look like the following:

Display Partition Information

Partition	Status	Type	Start	End	Size
C: 1	A	PRI DOS	0	399	400
2		EXT DOS	400	731	332

Total disk space is 732 cylinders.

The Extended Microsoft Operating System/2 partition contains logical drives.

Do you want to display logical drive information? [Y]

Press ESC to return to Fdisk Options

This screen identifies the partitions on your disk. It shows each partition's number, status, and type; its starting and ending cylinder numbers; and its size in cylinders.

**Fdisk main menu:
option 4**

If you have an extended partition, Fdisk asks whether you want to see information about that partition's logical drives. Type **y** and press the ENTER key to display a screen like the following:

Display Logical Drive(s)

Drv	Start	End	Size
D:	400	999	600
E:	1000	1399	400

Press ESC to return to Fdisk Options

Press the ESCAPE key to return to the main menu.

Selecting the Next Fixed-Disk Drive

Fdisk main menu: option 5

Option 5, Select Next Fixed Disk Drive, appears on the Fdisk main menu only if you have more than one hard disk attached to your computer. If you choose this option, Fdisk changes the current disk drive to the next fixed-disk drive.

For example, if the current disk drive is drive C, and if you choose option 5 on the main menu, Fdisk changes the current disk drive to drive D. You could then choose any of the Fdisk options (1-4) to prepare the second fixed disk for MS OS/2. Or, you could select option 5 once again to select the next drive. If there is not a third fixed disk, Fdisk changes the current disk drive from D back to C.

After you have selected the next drive, Fdisk displays the main menu again. This time the information reported is for your second hard disk.

5 Installing Device Drivers

Device drivers are programs that let the operating system recognize devices that are not part of the computer, such as a modem, printer, mouse, or external disk drive. Some device drivers are already installed with MS OS/2. Other device drivers, called installable device drivers, come with MS OS/2 for you to install if you need them.

This chapter describes the installable device drivers provided with MS OS/2:

- ANSI.SYS loads the ANSI character set into real mode.
- COM.SYS supports asynchronous communications ports.
- EGA.SYS supports a mouse used with certain modes on the IBM Enhanced Graphics Adapter (EGA).
- EXTDSKDD.SYS supports external floppy disk drives.
- MOUSEAxx.SYS supports a mouse.
- POINTDD.SYS draws the mouse pointer on the screen.
- VDISK.SYS supports one or more RAM (virtual) drives.

For more information about the **device** configuration command, which is used to install these device drivers, see Chapter 2, "Customizing Your System with CONFIG.SYS."

ANSI.SYS

The ANSI.SYS installable device driver lets you use ANSI escape sequences in real mode. An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) developed by the American National Standards Institute (ANSI). These sequences are used to define functions for MS OS/2. Specifically, you can change graphics functions and affect the movement of the cursor.

To install ANSI.SYS, include a command line of the following form in your CONFIG.SYS file:

ANSI.SYS syntax

device=[drive][path]**ansi.sys**

The escape sequences used in the ANSI.SYS file are listed in Appendix A, "ANSI Escape Sequences."

Note In protected mode, you can use the **ansi** command to turn ANSI support on and off. For example, the following command line turns ANSI support on:

```
ansi on
```

For more information about using the **ansi** command in protected mode, see the *Microsoft Operating System/2 User's Reference*.

COM.SYS

The COM.SYS installable device driver supports asynchronous communication to other computers using a serial port (COM1 or COM2).

To install COM.SYS, include a command line of the following form in your CONFIG.SYS file:

```
device=[drive:][path]com.sys[/COMn:baud,parity,
data,stop,recv,xmit][,p]
```

where *n* is a number from 1 to 8.

The following list describes how each variable is used:

COM.SYS syntax

Variable	Function
<i>baud</i>	Defines the decimal rate baud. Acceptable values are 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200.
<i>parity</i>	Represents the parity used for this port. Acceptable values are n (no), o (odd), e (even), m (mark), or s (space).
<i>data</i>	Defines the number of data bits: 5, 6, 7, or 8.
<i>stop</i>	Defines the number of stop bits. Acceptable values are 1 (one stop bit), 2 (two stop bits), or 5 (one and a half stop bits).
<i>recv</i>	Defines the receive buffer size in bytes.
<i>xmit</i>	Defines the transmit buffer size in bytes.
,p	Sets timeouts for RLS (Receive Line Signal), CTS (Clear to Send), and DSR (Data Set Ready) to infinite.

Note If you have a serial mouse, you must install the mouse device driver MOUSEA02.SYS before you install COM.SYS.

EGA.SYS

EGA.SYS is an installable device driver that supports a mouse device used with certain graphics modes of the IBM Enhanced Graphics Adapter (EGA). The EGA.SYS device driver supports a mouse with the following graphics video modes:

- 640 × 200 pixels, 16-color display (Mode 14)
- 640 × 350 pixels, monochrome display (Mode 15)
- 640 × 350 pixels, 64-color display (Mode 16)
- 640 × 480 pixels, 2-color display (Mode 17)
- 640 × 480 pixels, 16-color display (Mode 18)
- 320 × 200 pixels, 256-color display (Mode 19)
- 1024 × 768 pixels, 256-color display (BGA)
- 640 × 480 pixels, 256-color display (BGA)
- 1024 × 768 pixels, 16-color display (BGA)
- 640 × 480 pixels, 16-color display (BGA)

The program that uses this installable device driver must also interact with the graphics adaptor through the Microsoft Mouse EGA Register Interface. This interface allows a program to read to and write from write-only registers on the graphics adaptor.

EGA.SYS syntax

To install EGA.SYS, insert a command line of the following form in your CONFIG.SYS file:

device=[drive:][path]ega.sys

For example, suppose you purchased a graphics application that requires an EGA card and a mouse. If that application's documentation instructs you to install the EGA.SYS device driver, you would include the following line in the CONFIG.SYS file:

```
device=ega.sys
```

EXTDSKDD.SYS

EXTDSKDD.SYS is an installable device driver that supports external floppy-disk drives.

To install EXTDSKDD.SYS, insert a command line of the following form in your CONFIG.SYS file:

```
device=[drive:][path]extdiskdd.sys /d:ddd [/c] [/f:f]
[/h:hh] [/n] [/s:ss] [/t:ttt]
```

The following list describes how each switch is used:

Switch Function

- | | |
|---------------|---|
| /d:ddd | Specifies the physical drive number, ranging from 0 to 255. The first physical floppy disk drive is 0, otherwise known as drive A. Drive 1 is the second physical floppy-disk drive. Drive 2 is the third, which must be external. |
| /c | Indicates that change-line (doorlock) support is available for the drive. This means that the device driver can find out whether the door of a floppy-disk drive was opened since the last time the drive was accessed. If the door is open, the device driver will assume that the drive does not have a disk in it yet. |
| /f:f | Specifies the device type (form factor). The following values are allowed for f:

0, specifies 160/180K, or 320/360K
1, specifies 1.2 megabytes
2, specifies 720K

The default is 2. |
| /h:hh | Specifies the maximum number of disk read/write heads, ranging from 1 to 99. The default is 2. |
| /n | Specifies a nonremovable block device. A fixed disk is an example of a nonremovable block device. |

EXTDSKDD.SYS

syntax

- `/s:ss` Specifies the number of sectors per track, ranging from 1 to 99. The default is 9.
- `/t:ttt` Specifies the number of tracks per side on the block device, ranging from 1 to 999. The default is 80.

For example, if you want to add an external 720-kilobyte drive to your computer, you would include the following line in the CONFIG.SYS file:

```
device=extdskdd.sys /d:2 /f:2
```

MOUSEA xx .SYS

The MOUSEA xx .SYS installable device drivers support pointing (mouse) devices. Each of these device drivers works with a specific type of Microsoft mouse. These device drivers will work with both real-mode and protected-mode applications.

To install one of the MOUSEA xx .SYS device drivers, include a command line of the following form in your CONFIG.SYS file:

```
device=mousea $xx$ .sys [,serial=device][,mode=m]  
[,qsize=q]
```

where xx is 02, 03 or 04.

Each of the supported devices is supplied with a named device driver containing both real-mode and protected-mode functions. The installable device drivers for mice are as follows:

Driver	Mouse
MOUSEA02.SYS	Microsoft Serial Mouse for IBM Personal Computers (model 039-199)
MOUSEA03.SYS	Microsoft Bus (parallel) Mouse for IBM Personal Computers (model 037-299)
MOUSEA04.SYS	Microsoft InPort (parallel) Mouse for IBM Personal Computers (model 037-399)

The following list describes how each variable is used:

Variable	Function
<i>device</i>	Specifies the communications port to which the mouse is connected. Valid values are COM1 or COM2. With the MOUSEA02.SYS device driver, COM1 is the default. This argument is not used for parallel mice (MOUSEA03.SYS and MOUSEA04.SYS).

MOUSEA xx .SYS syntax

<i>m</i>	Specifies whether the mouse support is used for one or both modes. Acceptable values are r (real mode), p (protected mode), or b (both). The default is b.
<i>q</i>	Specifies the size in bytes of the queue buffer to be allocated for each protected-mode screen group. This number ranges from 1 to 100. The default is 10.

Identifying your mouse

If you purchased your Microsoft mouse a while ago, you may have forgotten what type of mouse you own. To find out which kind of Microsoft mouse you have, follow these steps:

- 1 Find out whether the mouse is connected to a parallel or serial port.

If it is connected to a parallel port, it is either a bus mouse or an InPort mouse. If it is connected to a serial port (i.e., COM1 or COM2), it is a serial mouse.

- 2 Check the mouse's connector.

The connector of a serial mouse has holes designed to hold the pins from the COM1 or COM2 port. The serial mouse is rectangular and will have either a 9-hole or 25-hole configuration.

The connector of a parallel mouse has nine pins, or at least space for nine pins. When a connector has less than nine pins, the holes of the remaining pin positions are visible.

- 3 If you have a parallel mouse, check to see if the connector is round or rectangular. If the connector is rectangular, you have a bus mouse. If it is round, and you have a bus mouse card, you have a bus mouse. If the connector is round, but is not connected to a bus mouse card, you have an InPort mouse.

Note If you have a serial mouse, be sure to install the MOUSEA02.SYS device driver before the COM.SYS driver. Be sure to install the POINTDD.SYS device driver in addition to this installable device driver. See the section on POINTDD.SYS, which follows.

POINTDD.SYS

POINTDD.SYS is the installable device driver that creates the pointer image that you see on your screen when you use a mouse. This device driver must be installed in addition to one of the MOUSEAxx.SYS device drivers.

To install POINTDD.SYS, include a command line of the following form in your CONFIG.SYS file:

device=[drive][path]pointdd.sys

The POINTDD.SYS device driver accepts no switches.

Be sure to install the mouse device driver in addition to this installable device driver. See the section on MOUSEAxx.SYS earlier in this chapter.

POINTDD.SYS
syntax

VDISK.SYS

VDISK.SYS is an installable device driver that lets you use a portion of your computer's memory as if it were a hard disk. This memory area is called a RAM disk and is sometimes referred to as a virtual disk.

RAM disks are faster than hard disks because the information they contain is always loaded into memory. If your computer has extended memory installed (starting at the 1-megabyte boundary), or if you have an extended memory board that meets the Lotus/Intel/Microsoft Expanded Memory Specification, you can use this extended memory for one or more RAM disks. Otherwise, VDISK.SYS places RAM disks in low memory.

Note The `device=vdisk.sys` command line increases the amount of memory used by resident MS OS/2 files.

VDISK.SYS syntax

To install VDISK.SYS, include a command line of the following form in your CONFIG.SYS file:

```
device=[drive][path]vdisk.sys [bbbb] [ssss] [dddd]
```

Warning When you reset or turn off the power on your computer, the information stored in RAM disks is lost.

The following list describes how each variable is used:

Variable	Function
<i>bbbb</i>	Specifies the disk size in kilobytes. The default value is 64; the minimum is 16.
<i>ssss</i>	Specifies the sector size in bytes. The default value is 128. The following values are allowed: 128, 256, 512, and 1024.
<i>dddd</i>	Specifies the number of root-directory entries. The default value is 64; the minimum is 2; the maximum is 1024.

VDISK.SYS adjusts the value of *dddd* to the nearest multiple of 16. For example, if you specify a value of 25 when the sector size is 512 bytes, the 25 will be rounded up to 32 (there are 16 32-byte directory entries in 512 bytes).

If there is not enough memory to create the RAM disk you specify, MS OS/2 will try to create a RAM disk of 16 directory entries. The result may be a RAM disk with a different number of directories than specified by *ddd*.

VDISK.SYS also adjusts the dimensions of the RAM disk to fit within the maximum root-directory limit of 255 sectors. For example, suppose you included the following command line in your CONFIG.SYS file:

```
device=vdisk.sys 64 128 1024
```

The RAM disk specified by this command line requires a root directory of 256 sectors, which exceeds the maximum limit. VDISK.SYS automatically adjusts the size of the RAM disk to fit within the system limits by changing the value of *ddd* from 1024 to 1020.

1

2

3

Appendix A

ANSI Escape Sequences

This appendix lists all of the escape sequences that can be used when you use the ANSI.SYS installable device driver in real mode or the **ansi** command in protected mode.

Escape sequences used in the ANSI.SYS file affect three general areas:

- Cursor positioning
- Erase functions
- Screen graphics

The ANSI escape sequences listed in this appendix use the following variables:

Variable	Description
<i>n</i>	A number, as in the number of rows or columns a cursor moves
<i>col</i>	A column number
<i>row</i>	A row number

For information about the ANSI.SYS file, see Chapter 5, "Installing Device Drivers." For information about the protected-mode **ansi** command, see the *Microsoft Operating System/2 User's Reference*.

Cursor Functions

Changing the cursor position

Cursor Position

ESC [*row* ; *col* H or ESC [*row* ; *col* f

These two escape sequences move the cursor to the position specified by the parameters. When no parameters are provided, the cursor moves to the home position (the upper-left corner of the screen).

Cursor Up

ESC [*n* A

This sequence moves the cursor up *n* rows without changing columns. If the cursor is already on the top line, MS OS/2 ignores this sequence.

Cursor Down

ESC [*n* B

This sequence moves the cursor down *n* rows without changing columns. If the cursor is already on the bottom row, MS OS/2 ignores this sequence.

Cursor Forward

ESC [*n* C

This sequence moves the cursor forward *n* columns without changing lines. If the cursor is already in the far right column, MS OS/2 ignores this sequence.

Cursor Backward

ESC [*n* D

This escape sequence moves the cursor back *n* columns without changing lines. If the cursor is already in the far left column, MS OS/2 ignores this sequence.

Save Cursor Position

ESC [s

This sequence saves the current cursor position. This position can be restored with the Restore Cursor Position sequence.

Restore Cursor Position

ESC [u

This sequence restores the cursor position to the Save Cursor Position value.

Erase Functions

Erase Display

ESC [2 J

This sequence erases the screen and moves the cursor to the home position.

Erase Line

ESC [K

This sequence erases from the cursor to the end of the line (including the cursor position).

Affecting erase functions

Screen Graphics Functions

Set Graphics Rendition

ESC [*g*; ... ; *g*

This escape sequence calls the graphic functions specified by the following numeric parameters. These functions remain until the next occurrence of this escape sequence. This escape sequence works only if the screen device supports graphics.

The *g* variable may be any of the following values:

Affecting screen graphics

Value	Function
0	All attributes off
1	Bold on
5	Blink on
7	Reverse video on
8	Concealed on
30	Black foreground
31	Red foreground
32	Green foreground

33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background
48	Subscript
49	Superscript

Parameters 30 through 47 meet the ISO 6429 standard.

Affecting screen mode

Set Mode

ESC = *s* h

This escape sequence changes the screen width or type. The *s* variable can be one of the following numeric parameters:

Value	Function
0	40 × 25 black and white
1	40 × 25 color
2	80 × 25 black and white
3	80 × 25 color
4	320 × 200 color
5	320 × 200 black and white
6	640 × 200 black and white
7	Wraps at the end of each line

Reset Mode

ESC = *s* l

Parameters for this escape sequence are the same as for Set Mode except parameter 7 resets the mode that causes wrapping at the end of each line.

Appendix B Using Code Pages

Microsoft Operating System/2 provides national language support through the use of language-specific code pages. If you live in, or work with, a country other than the United States, you may choose to use the MS OS/2 commands that support code-page switching.

A code page is a table that defines the character set you are using. A character set is a country-specific or language-specific group of characters that are translated from the code-page table and displayed by your screen or printer. Each code-page character set contains 256 characters. An example of a character set is the set of letters, numbers, and symbols (such as accent marks) used by French-Canadians.

MS OS/2 supports five different code pages:

- United States code page
- Multilingual code page, which includes all characters for most languages of European, North American, and South American countries
- Portuguese code page
- French-Canadian code page
- Nordic code page, which includes all characters for the Norwegian and Danish languages

MS OS/2 also provides national language support through the use of two other codes:

- A country code defines the country in which you live or work. MS OS/2 uses this code to prepare and assign default code pages for your system. MS OS/2 recognizes 19 country codes.
- A keyboard code defines the type of keyboard you are using. MS OS/2 recognizes 17 different keyboard codes.

What is a code page?

Country and keyboard codes

National Language Support Codes

The following table lists each country (or language) supported by MS OS/2. The table also lists the related country codes, default code-page assignments, and related keyboard codes. The code pages shown are automatically prepared by MS OS/2 when you load the corresponding country code by using the CONFIG.SYS **country** command. If you do not specify a country code, MS OS/2 loads the default United States code-page (437). For more information about the CONFIG.SYS file, see Chapter 2, "Customizing Your System with CONFIG.SYS."

Country or language	Country code	Code pages	Keyb code
United States	001	437,850	US
French-Canadian	002	863,850	CF
Latin America	003	437,850	LA
Netherlands	031	437,850	NL
Belgium	032	437,850	BE
France	033	437,850	FR
Spain	034	437,850	SP
Italy	039	437,850	IT
Switzerland	041	437,850	SF,SG
United Kingdom	044	437,850	UK
Denmark	045	865,850	DK
Sweden	046	437,850	SV
Norway	047	865,850	NO
Germany	049	437,850	GR
English (International)	061	437,850	—
Portugal	351	860,850	PO
Finland	358	437,850	SU
Arabic	785	437	—
Hebrew	972	437	—

Note Both Swiss-French and Swiss-German use country code 041. Code pages for Arabic and Hebrew languages are not available. Country codes 785 (Arabic) and 972 (Hebrew) assume the United States code page (437), but include country-specific date and time conventions.

National Language Support Commands

Two MS OS/2 commands support code-page selection for national languages:

Code-page selection commands

Command	Action
chcp	Displays or changes the current code page for the system and all prepared devices.
keyb	Selects a country-specific keyboard code for the keyboard you are using, and a code page for the character set you prefer. You may also select an alternate file that defines keyboard codes (other than the default KEYBOARD.SYS file) with this command, if such a file exists.

Four CONFIG.SYS commands also support country-specific information:

CONFIG.SYS commands

Command	Action
codepage	Prepares primary and secondary code pages for the system.
country	Identifies the country in which you work or live. This command also defines country-specific conventions to be used, such as date and time formats and sorting sequence for the character set.
device	Installs device drivers, including those that support code-page switching.
devinfo	Specifies code pages supported by devices.

Four other MS OS/2 commands — **date**, **backup**, **restore**, and **time** — use country-specific date and time conventions, based on the code pages you choose to use.

Language-specific date and time formats

The following table lists the date and time formats related to each country (or language group). These formats are determined by the **country** code set in your CONFIG.SYS file.

For each country, the Date Format column shows how MS OS/2 would display January 3, 1989, and the Time Format column shows how MS OS/2 would display 5:35 p.m. (with zero seconds and zero hundredths of seconds).

Country or language	Country code	Date format	Time format
United States	001	1-03-1989	17:35:00.00
French-Canadian	002	1989-01-03	17:35:00,00
Latin America	003	03/01/1989	17:35:00.00
Netherlands	031	03-01-1989	17:35:00,00
Belgium	032	03/01/1989	17:35:00,00
France	033	03/01/1989	17:35:00,00
Spain	034	03/01/1989	17:35:00,00
Italy	039	03/01/1989	17:35:00,00
Switzerland	041	03.01.1989	17.35.00.00
United Kingdom	044	03-01-1989	17:35:00.00
Denmark	045	03/01/1989	17.35.00,00
Sweden	046	1989-01-03	17.35.00,00
Norway	047	03/01/1989	17.35.00,00
Germany	049	03.01.1989	17.35.00,00
English (International)	061	03-01-1989	17:35:00.00
Portugal	351	03/01/1989	17:35:00,00
Finland	358	03.01.1989	17.35.00,00
Arabic	785	03/01/1989	17:35:00,00
Hebrew	972	03 01 1989	17:35:00.00

Setting the System Code Page

How to use code pages

Unless you specify otherwise, MS OS/2 assumes that you want to use the United States character set. To set your system to support another character set, you need to do the following:

- Set the country code with the CONFIG.SYS **country** command. This code identifies the country in which you live or work.

- Set the system code page with the CONFIG.SYS **codepage** command. This command lets you designate two code pages that MS OS/2 prepares for your system. The first code page listed is the primary code page, which MS OS/2 automatically loads for your system. The **chcp** command lets you select the secondary code page.
- Set the keyboard code with the **keyb** command.

Note Remember that when you change your CONFIG.SYS file, you must restart MS OS/2 to enable the new settings.

As an example, suppose you live in Quebec, Canada. You would follow these steps to use the French-Canadian character set with your system:

Changing character sets

- 1 Add the following line to your CONFIG.SYS file:

```
country=002
```

- 2 Restart MS OS/2 so that MS OS/2 will read your revised CONFIG.SYS file.

MS OS/2 will automatically select the French-Canadian code page (863) for you to use. Because your country code is 002, MS OS/2 has also prepared the Multilingual code page (850) for your system.

- 3 If you would like to change to the Multilingual code page, type the following:

```
chcp 850
```

- 4 Select the French-Canadian keyboard by typing the following command:

```
keyb cf
```

Note In place of steps 3 and 4, you could add the following lines to your AUTOEXEC.BAT file (for more information about AUTOEXEC.BAT, see Chapter 3, “Using Startup Files”). Then you would not have to type these commands each time you started MS OS/2:

```
chcp 850
keyb cf
```

Now your computer is set up to use the French-Canadian character set. Since your screen and printer are independent devices, you will also need to set them up for national language support. The section called "Preparing Device Code Pages," later in this appendix, explains how to do this.

Switching Between Code Pages

If you work in more than one language, you may need to switch between code pages. For example, suppose you work for an international company with offices in New York, London, and Oslo. You may need to use two different code pages to read or work with the correspondence you receive from your other offices.

To switch between code pages, you can use the **chcp** command. For example, if you have included this line in your CONFIG.SYS file, you can switch between the Multilingual and Nordic code pages:

```
codepage=850, 865
```

Changing code pages

In this example, MS OS/2 selects 850 (Multilingual) as the primary code page. If you want to change to the Nordic code page (865) to work with some information you receive from the Oslo office, you would type this command:

```
chcp 865
```

The display on your screen may flicker slightly as MS OS/2 loads a new code page.

Later, if you want to use the Multilingual code page again, you would type this command:

```
chcp 850
```

Preparing Device Code Pages

MS OS/2 lets you define code pages for individual devices with the CONFIG.SYS **devinfo** command.

To install a device driver, use the **device** command in your CONFIG.SYS file.

Note There is no limit to the number of times you can use the **device** command in your CONFIG.SYS file.

As an example, if you are using an EGA display, and want to use the Multilingual code page 850, you could include lines like the following in your CONFIG.SYS file:

```
device = c:\os2\dev\ega.sys  
devinfo = scr,ega,c:\os2\viotbl.dep,rom=850
```

1

2

3

Index

- * (asterisk) 47
- [] (brackets) 7
- ... (ellipsis) 7
- + (plus sign) vii
- 4201.DCP file 20
- 5202.DCP file 21

- Active session 49
- Alarm program 36
- Ansi command 64, 75
- ANSI.SYS device driver 63, 64, 75
- Asterisk (*) 47
- Asynchronous communication 63
- AUTOEXEC.BAT file
 - code page 83
 - creating 2, 5, 46
 - editing 46
 - options 47
 - Path command 5, 45
 - starting MS OS/2 45

- Background session 49
- BACKSPACE key 47
- Backup command 81
- Backup file 2
- BASIC (Microsoft) 48
- Batch file 45
- Baud rate 65
- Bold vii
- Bracket ([]) 7
- Break command 8, 9, 12
- Buffer 65, 70
- Buffers command 8, 9, 13
- Bus mouse (Microsoft) 69

- /c switch 67
- Capital letters vii
- Case conversion 14, 16
- Character set 14

- Chcp command 81, 83, 84
- Code page
 - available 14
 - Backup command 81
 - character set 79
 - command 9, 14
 - CONFIG.SYS file 80
 - country codes 79, 80
 - Date command 81
 - default 80
 - defined 14, 79
 - national language support 79
 - primary 15, 20, 81
 - Restore command 81
 - secondary 15, 20, 81
 - setting primary code pages 14
 - setting secondary code pages 14
 - setting the system 82
 - switching between code pages 84
 - Time command 81
- Codepage command 9, 14
- COM.SYS device driver 63, 65
- Command
 - Ansi 64, 75
 - Backup 81
 - Break 8, 9, 12
 - Buffers 8, 9, 13
 - Chcp 81, 83, 84
 - Codepage 9, 14, 21
 - configuration 8
 - Copy 48
 - Country 9, 16, 80, 81, 83
 - country dependent 9
 - Date 81
 - Device 8, 9, 11, 18, 36, 81, 84
 - Devinfo 9, 10, 14, 20, 21, 81
 - Diskcache 8, 10, 22
 - Fcbs 8, 10, 23, 51
 - Fdisk
 - creating extended partitions 56
 - deleting drive message 61
 - finding defective tracks 57

Command *(continued)*Fdisk *(continued)*

- main menu 53, 58
- message 54
- options 2, 53, 58
- starting 52
- uses 51
- warning 51

Files 9

- Format 52, 55

- frequently-used commands 8

- ignored 9

- interpreter 32

- Iopl 9, 10, 24

- Keyb 20, 81, 83

- language dependent 9

- Lastdrive 9

- Libpath 8, 10, 25

- little-used commands 9

- Maxwait 9, 10, 26

- Memman 9, 10, 27, 39

- notational conventions vii

- Path 5, 10, 38, 45, 49, 50

- Pauseonerror 8, 29

- Priority 9, 10, 30

- processor 10, 37, 38

- Prompt 38

- Protectonly 9, 10, 12, 23, 31, 34, 45

- Protshell 9, 10, 32

- Rem 8, 10, 33

- Ren 48

- Restore 81

- Rmsize 8, 10, 34

- Run 8, 10, 18, 36

- search path 5, 50

- Shell 9, 10, 37

- starting automatically 45

- STARTUP.COM file 5

- Swappath 8, 39

- Threads 9, 10, 40

- Time 81

- Timeslice 9, 10, 26, 30, 41

- uppercase letters vii

- COMMAND.COM file 11, 37, 38

- Command processor 11

- Communications port 69

- CONFIG.SYS file

- adding external disk drives 68

- Break command 12

- Buffers command 13

- Chcp command 81

CONFIG.SYS file *(continued)*

- code pages 80, 81

- Codepage command 14

- command line processing order 18

- Country command 16, 81

- country dependent 9

- creating 5

- customizing your system 5

- default settings 7

- defined 8

- Device command 18, 36, 81

- device drivers 8

- Devinfo command 14, 20, 21, 81

- differences between Startup files 43, 50

- Diskcache command 22

- error handling 10

- Fcbs command 23

- InPort mouse 19

- Install program 2

- installing device drivers 66, 84

- Iopl command 24

- Keyb command 81

- language dependent 9

- line editor 8

- Maxwait command 26

- Memman command 27, 39

- POINTDD.SYS device driver 71

- Priority command 30

- Protectonly command 12, 23, 34, 45

- Rem command 33

- remarks in 10, 33

- Rmsize command 34

- Run command 18, 36

- sample file 11

- Shell command 37

- Swappath command 39

- switching between code pages 84

- system code page 82

- Threads command 40

- Timeslice command 41

- VDISK.SYS device driver 72, 73

- Control block 23

- CONTROL+ALT+DELETE key 8, 52, 54

- CONTROL+C key 11, 12, 47

- CONTROL+Z key 49

- Codepage command 9, 14, 21

- Copy command 48

- Country codes 16, 79

- Country command 9, 16, 80, 81, 83

- Currency format 14, 16

- Cursor 75, 76, 77

Cylinder 55, 56, 57, 60, 61

Data 22, 27, 65

DATA directory 39

Date command 81

Date conversion 9

Date format 14, 16, 82

/d:ddd switch 67

Decimal rate baud 65

Device command 8, 9, 11, 18, 36, 81, 84

Device

boot 27

code page 20, 81, 84

Devinfo command 14

drivers 8, 11, 18, 63-73

nonremovable block 67

protected-mode function 69

real-mode function 69

type 67

Device drivers

See also individual device driver

ANSI.SYS 63, 64, 75

COM.SYS 63, 65

default 18

EGA.SYS 63, 66

EXTDSKDD.SYS 63, 67

installing 11

MOUSEA.SYS 63

MOUSEA02.SYS 65, 69

MOUSEA03.SYS 69

MOUSEA04.SYS 69

POINTDD.SYS 63, 71

DISK.SYS 63, 72, 73

Devinfo command 9, 10, 14, 20, 21, 81

Directory

command processor 11

creating 2

DATA 39

DYNLIBS 25

Install program 5

locating programs 49

notational conventions vii

\OS2 4

\OS2\BIN 4

\OS2\DEV 4

\OS2\PBIN 4

\OS2\RBIN 4

Path command 5

pathname 25

RAM disk 73

Directory (*continued*)

root

AUTOEXEC.BAT file 5, 50

Codepage command 21

CONFIG.SYS file 5, 8

editing Startup files 46

File Allocation Table (FAT) 4

files created using the Install program

45

INITENV.CMD file 5

Libpath command 25

STA_TUP.CMD file 5

VDISK.SYS device driver 72

Disk

allocating space 22

buffer 13

caching data 22

command processor 11

configuration 52

drive 32

Fdisk command warning 51

floppy

boot device 27

default device drivers 18

Install program 2

setup vi

formatting 52

hard

available space 55

caching data 22

configuration 52

creating partitions 51, 53

default device drivers 18

diagram 3

drive letter 3

extended partition 3

formatting 2, 55

logical drive 3

/n switch 67

organization 4, 51

partitioning 2, 3

primary partition 3

sector 3

setup vi

side 3

swap file 39

switching between hard disks 62

system files 2

track 3

updating the operating system 2

DISK.SYS device driver 72

Disk (*continued*)

- initialization software 11
- minimum swap size 39
- RAM virtual drive 63, 72
- read/write heads 67
- sector 68
- setting buffers 9
- size 72
- space 55-57, 60, 61
- supplemental vi, 2, 4
- swap file 39
- track 68
- VDISK.SYS device driver 63, 72, 73

Disk drive

- adding external disk drives 68
- deleting logical drives 60
- floppy vi, 63
- hard vi
- logical 51, 56, 57, 62
- program disk 4
- RAM (virtual) 63, 72
- supplemental disk 4

- Diskcache command 8, 10, 22

- Doorlock 67

- DOS 4, 9, 31

- See also* MS-DOS

- Dynamic-link library 25

- Dynamic-link module 10

- DYNLIB directory 25

- /e switch 38

- Edlin program 46, 47

- EGA (Enhanced Graphics Adapter) 63, 66

- EGA.SYS device driver 63, 66

- Ellipsis (...) 7

- Enhanced Graphics Adapter (EGA) 63, 66

- Erase function 75

- Escape sequence 64, 75

- EXTDSKDD.SYS device driver 63, 67

- Extension (filename) 36

- FAT (File Allocation Table) 4

- Fcbs command 8, 10, 23

- Fdisk command

- creating extended partitions 56

- deleting drive message 61

- finding defective tracks 57

- main menu 53, 58

- message 54

Fdisk command (*continued*)

- options 2, 53, 58

- starting 52

- uses 51

- warning 51

- /f:f switch 67

File

- 4201.DCP 20

- 5202.DCP 21

- accessing 23

- ANSI.SYS 63, 64, 75

- AUTOEXEC.BAT

- code page 83

- creating 2, 5, 46

- editing 46

- options 47

- Path command 5, 45

- starting MS OS/2 45

- backup 2

- batch 45

- closing 23

- COMMAND.COM 11, 37, 38

- CONFIG.SYS

- adding external disk drives 68

- Break command 12

- Buffers command 13

- Chcp command 81

- Codepage command 14

- code pages 80, 81

- command line processing order 18

- Country command 16, 81

- country dependent 9

- creating 5

- customizing your system 5

- default settings 7

- defined 8

- Device command 18, 36, 81

- device drivers 8

- Devinfo command 14, 20, 21, 81

- difference between Startup files 43, 50

- Diskcache command 22

- error handling 10

- Fcbs command 23

- InPort mouse 19

- Install program 2

- installing device drivers 66, 84

- Iopl command 24

- Keyb command 81

- language dependent 9

- line editor 8

- Maxwait command 26

File (*continued*)

CONFIG.SYS (*continued*)
 Memman command 27, 39
 POINTDD.SYS device driver 71
 Priority command 30
 Protectonly command 12, 23, 34, 45
 rebooting 33
 Rem command 33
 remarks in 10, 33
 Rmsize command 34
 Run command 18, 36
 sample file 11
 Shell command 37
 Swappath command 39
 switching between code pages 84
 system code page 82
 Threads command 40
 Timeslice command 41
 VDISK.SYS device driver 72, 73
 configuration 7
 control blocks 10, 23
 editing 8
 extension 36
 Fcbs command 8, 10, 23
 handles 23
 INITENV.COMD
 creating 2, 5, 46
 editing 46
 example 49
 MS OS/2 5
 Path command 45
 starting MS OS/2 45
 logical drive 3
 naming vi, 32
 NEWCMD.COM 38
 NEWSHELL.EXE 32
 pathname 25
 remarks in CONFIG.SYS file 10, 33
 search path 5
 segregating application files 57
 Startup 46
 STARTUP.COMD
 creating 2, 5, 48
 editing 46
 starting MS OS/2 45
 swap file 10, 39
 VIOTBL.DCP 21
 File Allocation Table (FAT) 4
 File control blocks 10, 23
 Filename extension 36
 Filename vii, 32

Files command 9
 Fixed disk *See* Hard disk
 Floppy disk
 boot device 27
 default device drivers 18
 Install program 2
 setup vi
 Foreground session 49
 Form factor 67
 Format command 52, 55

Graphics video modes 66

Hard disk
 available space 55
 caching data 22
 configuration 52
 creating partitions 51
 default device drivers 18
 diagram 3
 drive letter 3
 extended partition 3
 formatting 2, 55
 logical drive 3
 /n switch 67
 organization 4, 51
 partitioning 2, 3
 primary partition 3
 sector 3
 setup vi
 side 3
 swap file 39
 switching between hard disks 62
 system files 2
 track 3
 updating the operating system 2
 VDISK.SYS device driver 72

Hardware

console 18
 Devinfo command 14
 Enhanced Graphics Adapter (EGA) 63
 extended memory board 72
 external disk drive 63
 floppy-disk drive 18
 hard disk drive 18
 identifying device drivers 18
 InPort mouse 19
 keyboard
 changing default settings 7

Hardware *(continued)*keyboard *(continued)*

code page 80, 81, 83
 CONFIG.SYS file 14
 country specific 81
 creating a startup file 49
 Keyb command 20, 83
 KEYBOARD.DCP file 21
 selecting alternate code pages 20
 81-82
 translation table 21
 modem 63
 monitor 14, 20, 21, 84
 mouse (Microsoft) 63, 69-70
 printer 14, 18, 20, 21, 63, 84
 screen 64, 76, 77, 78

/h:hh switch 67

How to use this guide vii

INITENV.CMD file

creating 2, 5, 46
 editing 46
 example 49
 MS OS/2 5
 Path command 45
 starting MS OS/2 45

InPort mouse (Microsoft) 19, 69-70

Install program 2, 4, 5, 7, 45, 46

Intel 72

International conversions 16

Iopl command 9, 10, 24

Italic vii

Key

BACKSPACE 47
 CONTROL+ALT+DELETE 8, 52, 54
 CONTROL+C 11, 12, 47
 CONTROL+Z 49
 keyboard settings 7
 notational conventions vii
 plus sign (+) vii
 reassigning 64
 small caps vii

Keyb command 20, 81, 83

Keyboard

changing default settings 7
 code page 80, 81, 83
 CONFIG.SYS file 14
 country specific 81

Keyboard *(continued)*

creating a startup file 49
 Keyb command 20, 83
 KEYBOARD.DCP file 21
 selecting alternate code pages 20, 81-82
 translation table 21

Language support 79

Lastdrive command 9

Libpath command 8, 10, 25

Lotus 72

Maxwait command 9, 10, 26

Memman command 9, 10, 27, 39

Memory

512 kilobytes vi
 allocating disk buffers 13
 boundary 34
 caching data 22
 creating a RAM (virtual) disk drive 72,
 73
 creating an MS OS/2 partition 51, 54
 expanded specification 72
 extended 72
 low 72
 management 27
 Memman command 10, 27
 read only 21
 real-mode applications 10
 recommended vi
 setting disk buffers 9
 swap file 10, 27, 39
 time sharing 27
 workspace 13

Microsoft BASIC 48

Microsoft Mouse 66, 69-70

Microsoft Word 46

Modem 63

Monitor 14, 20, 21, 84

Monospace vii

Mouse 19, 63, 66, 69-70

MOUSEA.SYS device driver 63

MOUSEA02.SYS device driver 65, 69

MOUSEA03.SYS device driver 69

MOUSEA04.SYS device driver 69

MS OS/2

alternate program selector 10
 central processing unit (CPU)
 resources 30
 command processor 11

- MS OS/2 (*continued*)
 - configuring your system 7
 - CONTROL+C key 12
 - copying to hard disk 2
 - CPU (central processing unit)
 - resources 30
 - creating partitions 51, 54
 - currency conversion 9
 - cursor movement 64, 76
 - date conversion 9
 - defining function 64
 - deleting disk partitions 59
 - disk buffers 13
 - dynamlink libraries 4
 - dynamic-link library 25
 - earlier versions 2
 - Edlin program 46, 47
 - identifying device drivers 18
 - Install program 2, 4, 5, 7, 45, 46
 - national language support 79
 - opening and closing files 23
 - partition 3
 - performance 27
 - priority classes 30
 - process scheduling 10
 - program disk vi, 2, 4, 52, 55
 - program protection 4
 - program selector 31
 - prompt 47, 48, 49, 50
 - protected mode 4, 5, 8, 31
 - real mode 4, 8, 48
 - removing an old operating system 2
 - restarting the program 8
 - retaining an old operating system 2
 - running several programs 4
 - scheduler 26, 42
 - search path 5
 - setting disk buffers 9, 13
 - setup program 51
 - starting the program 44, 52
 - stopping programs 11
 - switching code pages 20
 - system code pages 14
 - system files 2
 - threads 40, 41
 - unrecognized commands 29
 - updating the operating system 2
 - utilities 4
 - writing swap files 39
- MS-DOS 4, 9, 31
- /n switch 67
- National language support 79
- NEWCMD.COM file 38
- NEWSHELL.EXE file 21, 23, 32
- Nonremovable block device 67
- Notational conventions vii
- Operating system *See* MS OS/2
- Option notational conventions vii
- OS/2 *See* MS OS/2
- \OS2 directory 4
- \OS2\BIN directory 4
- \OS2\DEV directory 4
- \OS2\PBIN directory 4
- \OS2\RBIN directory 4
- /p switch 11
- Parallel mouse (Microsoft) 69
- Parity 65
- Partition
 - active 58
 - deleting partitions 59, 60
 - displaying information 61
 - extended 51, 53, 56, 59, 60, 62
 - maximum space message 55, 56, 57
 - multiple 55
 - primary 51, 54, 55, 59
- Partitioning a hard disk 3
- Path command 5, 10, 38, 45, 49, 50
- Pauseonerror command 8, 29
- Plus sign (+) vii
- POINTDD.SYS device driver 63, 71
- Pointing device *See* Mouse
- Printer *See* Hardware
- Priority classes 30
- Priority command 9, 10, 30
- Program disk (MS OS/2) vi, 2, 4, 52, 55
- Program selector 10, 31, 52
- Prompt 47, 48, 50
- Prompt command 38
- Protected mode 5, 8
- Protectonly command 9, 10, 12, 23, 31, 34, 45
- Protshell command 9, 10, 32
- RAM virtual disk drive 63, 72
- Real mode 4, 8, 48
- Reboot 8, 33

Register Interface (Microsoft Mouse EGA)

66

Rem command 8, 10, 33

Remarks in CONFIG.SYS file 10, 33

Ren command 48

Restarting MS OS/2 8

Restore command 81

Reverse video 77

Rmsize command 34

Root directory *See* Directory

Run command 8, 10, 18, 36

/s switch 55

Scheduler (MS OS/2) 26, 42

Screen 64, 66, 75, 76, 77, 78

Search path 5, 11, 50

Sector 22, 72

Serial port 69, 70

Session 49

Session Manager 37

Setup program 51

Shell 37

Shell command 9, 10, 37

Small caps vii

/s:ss switch 68

Starting MS OS/2 8, 44, 52

Startup files defined 46

STARTUP.CMD file

creating 2, 5, 48

editing 46

starting MS OS/2 45

Stop bit 65

Stopping a program 11

Stop variable 65

Swappath command 8, 39

Switch

/c 67

/d:ddd 67

device drivers 67

/e 38

/f:f 67

/h:hh 67

/n 67

notational conventions vii

/p 11

POINTDD.SYS device driver 71

/s 55

Shell command 37

/s:ss 68

/t:ttt 68

Syntax notational conventions vii

System code 14

System configuration (MS OS/2) 7

Threads 40, 41

Threads command 9, 10, 40

Time command 81

Time conversion 9

Time format 14, 16, 82

Timeslice command 9, 10, 26, 30, 41

Track 55, 57

Translation table (keyboard) 21

/t:ttt switch 68

Unrecognized commands 29

Updating MS OS/2 2

Uppercase letters vii

Utilities (MS OS/2) 4

Variable notational conventions vii

VDISK.SYS device driver 63, 72, 73

VIOTBL.DCP file 21

Word (Microsoft) 46

XENIX 58