Windows Resource Kit

Microsoft® Windows™ Version 3.1

MICROSOFT® WINDOWS™

Complete Technical Information for the Support Professional



Microsoft® WINDOWS™ RESOURCE KIT

For Operating System Version 3.1

Microsoft® Windows™ Resource Kit

Complete Technical Information
for the Support Professional
for the Microsoft Windows Operating System
Version 3.1

Microsoft Corporation

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. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure 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, recording, or information recording and retrieval systems, for any purpose other than the purchaser's personal use, without the express written permission of Microsoft.

© 1992 Microsoft Corporation. All rights reserved.
Printed in the United States of America.

Microsoft, MS, MS-DOS, Bookshelf, QuickC, and QuickPascal are registered trademarks of, and QuickBasic, QBasic, Windows, and the Windows logo are trademarks of Microsoft Corporation.

This document was prepared with Microsoft® Word for Windows™ 2.0 and TrueType®.

Document No. 0030-31645

1-2-3, Freelance, Graphwriter, Lotus, Lotus Agenda, Magellan, and Symphony are registered trademarks of Lotus Development Corporation. 3+, 3Com, and EtherLink are registered trademarks of and 3+Open, 3Plus, and 3Station are trademarks of 3Com Corporation. ACCPAC and SuperCalc are registered trademarks of Computer Associates International, Inc. Adobe, Adobe Type Manager, and PostScript are registered trademarks of Adobe Systems, Inc. AIX, AT, DB2, DisplayWrite, Filing Assistant, IBM, Micro Channel, OS/2, Personal System/2, Proprinter, PS/2, Quickwriter, and Quietwriter are registered trademarks of and DisplayWrite Assistant, ExecJet, Graphics Assistant, PC/XT, Professional Editor, and XT are trademarks of the International Business Machines Corporation. Aldus and PageMaker are registered trademarks of Aldus Corporation. Apricot is a registered trademark of Apricot Computers, PLC. Arial and Times New Roman are registered trademarks of The Monotype Corporation PLC. Ashton-Tate, dBASE, dBASE III, dBASE IV, Framework, and Framework III are registered trademarks of and APPLAUSE and Framework II are trademarks of Ashton-Tate Corporation. AST and RAMpage! are registered trademarks of AST Research, Inc. AT&T is a registered trademark of American Telephone and Telegraph Company. AutoCAD is a registered trademark of Autodesk, Inc. Banyan and VINES are registered trademarks of Banyan Systems, Inc. Bitstream is a registered trademark of and Fontware is a trademark of Bitstream, Inc. The Fontware mark is licensed to Bitstream, Inc., in the UK, France, and West Germany by Electronic Print Systems Ltd. Borland, Borland International, Quattro, Reflex, SideKick, and Turbo Pascal are registered trademarks of Borland International, Inc. BRIEF is a registered trademark of SDC Software Partners II L.P. Bring the World to Your Senses is a trademark of Multimedia PC Marketing Council, Inc. Canon is a registered trademark of Canon USA Inc. cc:Mail is a registered trademark of cc:Mail, Inc. Centronics is a registered trademark of Centronics Data Computer Corp. COMPAQ DESKPRO 386 is a registered trademark of Compaq Computer Corporation. CompuServe is a registered trademark of CompuServe, Inc. Crosstalk is a registered trademark of Digital Communications Associates, Inc. DataEase is a registered trademark of DataEase International. DEC, DECnet, and Pathworks are registered trademarks of Digital Equipment Corporation. DeskJet, Hewlett-Packard, HP, JetSeries, LaserJet, PaintJet, PCL, and ThinkJet are registered trademarks of Hewlett-Packard Company. DrawPerfect, PlanPerfect, and WordPerfect are registered trademarks of WordPerfect Corporation. Epson is a registered trademark of Seiko Epson Corporation, Inc. Everex is a trademark of Everex Systems, Inc. Flight Simulator is a registered trademark of Bruce A. Artwick. Foxbase+ is a trademark of Fox Software, Inc. Fujitsu is a registered trademark of Fujitsu Limited. Harvard, Harvard Graphics, OfficeWriter, PFS, PFS:, PFS: Access, PFS: First Choice, PFS: Plan, Professional File, and Professional Write are registered trademarks of and Harvard GeoGraphics is a trademark of Software Publishing Corporation. Helvetica and Linotype are registered trademarks of and Linotronic is a trademark of Linotype AG and its subsidiaries. Hercules is a registered trademark of Hercules Computer Technology. Insight Development is a trademark of Insight Development Corporation. Intel is a registered trademark of and 386SX, Genius, and Ripplemode are trademarks of Intel Corporation. Interleaf is a registered trademark of Interleaf, Inc. KnowledgePro is a registered trademark of Knowledge Garden Inc. Lap-Link is a registered trademark of Traveling Software, Inc. LaserWriter and TrueType are registered trademarks of Apple Computer, Inc. Logitech is a trademark of Logitech, Inc. Managing Your Money is a registered trademark of MECA Software. Microrim and R:BASE are registered trademarks of and R:BASE 5000 is a trademark of Microrim, Inc. MultiMate is a registered trademark of MultiMate International Corporation, an Ashton-Tate company. NEC is a registered trademark of NEC Corporation. Net/One and Ungermann-Bass are registered trademarks of Ungermann-Bass, Inc. NetWare, Novell, and XTRIEVE are registered trademarks of Novell, Inc. Norton Utilities is a registered trademark of Peter Norton Computing. Okidata is a registered trademark of Oki America, Inc. Olivetti is a registered trademark of Ing. C. Olivetti. Panasonic is a registered trademark of Matsushita Electric Co., Ltd. Paradox is a registered trademark of Ansa Software, a Borland company. PC Paintbrush IV and Publisher's Type Foundry are trademarks of ZSoft Corporation. PC Tools and PC Tools Deluxe are trademarks of Central Point Software, Inc. Phaser is a trademark of Tektronix, Inc. Prodigy is a trademark of Prodigy Services Company. Qemm is a trademark of Quarterdeck Office Systems. QMS is a registered trademark of QMS, Inc. Qualitas is a registered trademark of and 386-Max and Blue Max are trademarks of Qualitas, Inc. Quicken is a registered trademark of Intuit. Ready! is a trademark of Living Videotext, Inc. Rolodex is a registered trademark of Insilco Corporation. SCSI is a registered trademark of Security Control Systems, Inc. Smartcom II is a registered trademark of Hayes Microcomputer Products, Inc. Sun and Sun Microsystems are registered trademarks of Sun Microsystems, Incorporated. Thanks for the Memory is a registered trademark of Tehsys Memory Products Corporation. Toshiba is a registered trademark of Kabushiki Kaisha Toshiba. Triumph is a registered trademark of Triumph Adler AG. TurboTax is a registered trademark of ChipSoft, Inc. Type Director is a trademark of Agfa Corporation. Unisys is a registered trademark of Unisys Corporation. UNIX is a registered trademark of UNIX Systems Laboratories. Vectra is a registered trademark of H-P Company Ventura Publisher is a registered trademark of Ventura Software, Inc. VISI ON is a registered trademark of VISICORP. Volkswriter is a registered trademark of Lifetree Software, Inc. WordStar is a registered trademark of WordStar International Incorporated. Wyse is a registered trademark of Wyse Technology. WYSIfonts! is a registered trademark of SoftCraft, Inc. Xcellenet and Design are registered trademarks of Xcellenet, Inc. XyWrite is a registered trademark of XYQUEST, Inc. Zenith is a registered trademark of Zenith Radio Corporation.

THE WINDOWS RESOURCE KIT CONTAINS MICROSOFT AND OTHER THIRD PARTY SOFTWARE. UNLESS NOTED OTHERWISE IN ANY PORTION OF THE SOFTWARE, THE FOLLOWING TERMS AND CONDITIONS SHALL APPLY TO THE USE OF THE SOFTWARE.

MICROSOFT SOFTWARE LICENSE: Windows Resource Kit

1. GRANT OF LICENSE. Microsoft grants to you the right to use one copy of the enclosed Microsoft software program (Software) on a single terminal connected to a single computer (i.e., with a single CPU). You may not network the Software or otherwise use it on more than one computer or computer terminal at the same time.

2. COPYRIGHT. The Software is owned by Microsoft or its suppliers and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the Software like any other copyrighted material (e.g., a book or musical recording) except that you may either (a) make one copy of the Software solely for backup or archival purposes, or (b) transfer the Software to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the Software.

3. OTHER RESTRICTIONS. This Microsoft License Agreement is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent or lease the Software. You may not reverse engineer, decompile or disassemble the Software.

NO WARRANTY. ANY USE BY YOU OF THE SOFTWARE IS AT YOUR OWN RISK. THE SOFTWARE IS PROVIDED FOR USE AS IS WITHOUT WARRANTY OF ANY KIND. MICROSOFT AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall Microsoft or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this Microsoft product, even if Microsoft has been advised of the possibility of such damages. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You may have additional rights, which vary from jurisdiction to jurisdiction.

U.S. GOVERNMENT RESTRICTED RIGHTS

The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

If you acquired the Software in the United States, this Agreement shall be governed by the laws of the State of Washington.

Should you have any questions concerning the Microsoft software, please write your local Microsoft subsidiary or: Microsoft Customer Sales and Service, One Microsoft Way, Redmond, WA 98052-6399.

Should you have any questions concerning any third party software, please contact the respective owner of such software.

Contents

Microsoft Software License Agreement ..... *iii*

Welcome **1**

About the Windows Resource Kit 2
Conventions in This Manual 4
The Troubleshooting Flowcharts for Windows 3.1 7

Part 1 Installation and Setup

Chapter 1 **Windows 3.1 Installation** **57**

About Windows Setup 58
Windows Setup: A Technical Discussion 62
Setting Up Windows for a Network 71
Using Automated Setup: Setup /h 75
Maintaining Windows with Setup 82

Chapter 2 **The Windows Setup Information Files** **85**

About the Setup Information Files 86
SETUP.INF: The Setup Information File 88
APPS.INF: The Application Installation Information File 121
CONTROL.INF: The Control Panel Information File 126
Modifying .INF Files for Custom Installations 128

Chapter 3 **The Windows Files** **135**

About the Windows Files 136
WIN.COM 136
The Core Files 137
Drivers, Fonts, and International Support Files 137
MS-DOS Support Components of Windows 148
Windows Applications, Setup, and Other Files 151
Files You Can Delete 155

Chapter 4 **The Windows Initialization Files** **157**

About the Initialization Files 158
WIN.INI: The Windows Initialization File 163
SYSTEM.INI: The System Initialization File 183
Other Windows Initialization Files 219

Part 2 Configuring Windows 3.1

Chapter 5 **Windows 3.1 and Memory Management** 227

About Memory....228
Windows Standard Mode and Memory....235
Windows 386 Enhanced Mode and Memory....238
Other Memory Management Issues....247
SMARTDrive 4.0: A Technical Discussion....252

Chapter 6 **Tips for Configuring Windows 3.1** 257

Tips for Optimizing Performance....258
Tips for Configuring the Windows Shell....272

Part 3 Non-Windows–Based Applications

Chapter 7 **Setting Up Non-Windows Applications** 283

About Applications....284
Setting Up PIFs and Program Items....285
Starting Non-Windows Applications with Windows....288
Running Non-Windows Applications....290
Tips for Running Non-Windows Applications....294

Chapter 8 **PIFs and PIF Editor** 299

About Creating PIFs....300
About PIF Editor....300
Modifying the Default PIF....305
Creating Multiple PIFs for a Single Application....306

Part 4 Using Windows 3.1

Chapter 9 Fonts 309

About Fonts ... 310
TrueType and Windows 3.1 ... 318
Printer Fonts and Windows 3.1 ... 320
Changing Typeface Names for Soft Fonts ... 337

Chapter 10 Printing 339

About Windows Printing ... 340
Printer Ports ... 342
Determining Print Method ... 349
Printing with a Network Driver ... 350

Chapter 11 Compound Documents in Windows 3.1 351

About Object Linking and Embedding ... 352
Windows Support for Compound Documents ... 354
Managing the Registration Database ... 359

Part 5 Networks and Windows 3.1

Chapter 12 Networks and Windows 3.1 367

About Networks and Windows 3.1 ... 368
3Com Networks ... 374
Artisoft LANtastic ... 376
Banyan VINES ... 377
DEC Pathworks ... 379
IBM OS/2 LAN Server ... 383
IBM PC LAN Program ... 383
Invisible Software —NET/30 For Windows ... 384
Microsoft LAN Manager Networks ... 384
Microsoft Network and Compatibles ... 387
Novell NetWare ... 387
TCS 10Net ... 391
Ungermann-Bass Net/One ... 392

Part 6 Troubleshooting Windows 3.1

Chapter 13 **Troubleshooting Windows 3.1** 395

About Troubleshooting 396
Troubleshooting Setup 399
Troubleshooting Windows Configuration 403
Troubleshooting Windows Running in Standard Mode 418
Troubleshooting Windows Running in 386 Enhanced Mode 420
Troubleshooting for Non-Windows Applications 423
Troubleshooting Fonts 425
Troubleshooting Printing 426
Troubleshooting Networks 430
Troubleshooting GP Faults 431
Troubleshooting Multimedia 434

Chapter 14 **The Windows Resource Kit Disk** 439

The Windows Resource Kit Utilities 440
Other Windows Resource Kit Files 443

Part 7 References, Resources, and Appendixes

Glossary 447

Appendix A **Windows Resource Directory** 457

Appendix B **Hardware Compatibility List** 487

Appendix C **Windows 3.1 Disks and Files** 499

Appendix D **Articles** 505

Windows for Pens 506
QEMM-386 and Windows 3.1 515
PC-NFS and Windows 3.1 519
FastDisk: An Introduction to 32-Bit Disk Access 520

Index 525

Introduction



Welcome

Welcome to the *Windows Resource Kit: Complete Technical Information for the Support Professional for the Microsoft Windows Operating System.* This manual is designed for people who are, or who want to become, expert users of Microsoft® Windows™ 3.1.

We prepared the *Windows Resource Kit* in response to your requests for a detailed, easy-to-read technical view of Windows, so that you can better manage how Windows is used at your site. The *Windows Resource Kit* also contains specific information for system administrators who are responsible for installing and managing Windows in a network or multiuser environment.

This introductory chapter presents three kinds of information you can use to get started:

- The first section of this chapter outlines the contents of the *Windows Resource Kit,* so you can quickly find technical details about specific elements of Microsoft Windows.
- The second section contains an overview of the conventions used to present information in the *Windows Resource Kit.*
- The third section of this chapter presents a series of troubleshooting flowcharts, so you can quickly find details and procedures for solving problems you might have installing or running Windows.

The *Windows Resource Kit* is a technical supplement to the documentation that is included in your Windows package and does not replace that information as the source for learning how to use Windows features and Windows applications.

Contents of this chapter

About the Windows Resource Kit 2
Conventions in This Manual 4
    Document Conventions 4
    Syntax Conventions 5
The Troubleshooting Flowcharts for Windows 3.1 7

About the Windows Resource Kit

This guide is organized in seven parts to present specific details about Windows installation, memory management, non-Windows applications, Windows fonts and printing, networks, and other issues.

Part 1: Installation and Setup

- **Chapter 1, "Windows 3.1 Installation,"** contains a technical discussion of the Windows Setup program, details about setting up Windows on a network, and instructions for creating a custom installation routine for automated Setup.
- **Chapter 2, "The Windows Setup Information Files,"** contains the details you need to create custom Windows setup information files (SETUP.INF, CONTROL.INF, and APPS.INF) for multiple installations.
- **Chapter 3, "The Windows Files,"** describes the purpose for each file in the WINDOWS directory and the Windows SYSTEM subdirectory.
- **Chapter 4, "The Windows Initialization Files,"** describes the contents of WIN.INI and SYSTEM.INI files, plus other Windows initialization files, and explains how you can change entries in these files.

Part 2: Configuring Windows 3.1

- **Chapter 5, "Windows 3.1 and Memory Management,"** contains information about how Microsoft Windows interacts with memory.
- **Chapter 6, "Tips for Configuring Windows 3.1,"** presents tips about configuring your system, both for gaining optimal performance and for creating custom Windows configurations.

Part 3: Non-Windows–Based Applications

- **Chapter 7, "Setting Up Non-Windows Applications,"** presents details about how to set up and run non-Windows applications under Microsoft Windows 3.1.
- **Chapter 8, "PIFs and PIF Editor,"** presents details related to creating PIFs for non-Windows applications.

Part 4: Using Windows 3.1

- **Chapter 9, "Fonts,"** examines technical issues related to fonts in Windows, focusing on TrueType, the new font technology available in Microsoft Windows 3.1. This chapter also presents details about using printer fonts with specific types of printers.
- **Chapter 10, "Printing,"** examines technical issues related to printing in Microsoft Windows 3.1.
- **Chapter 11, "Compound Documents in Windows 3.1,"** presents an overview of how Object Linking and Embedding (OLE) works in Windows applications. This chapter also describes how to update and add to the system registration database using REGEDIT.EXE.

Part 5: Networks and Windows 3.1

- **Chapter 12, "Networks and Windows 3.1,"** contains general tips about using a network with Microsoft Windows, plus notes about specific networks.

Part 6: Troubleshooting Windows 3.1

- **Chapter 13, "Troubleshooting Windows 3.1,"** provides specific information for troubleshooting problems with Windows, showing the key steps for isolating and solving common problems.
- **Chapter 14, "The Windows Resource Kit Disk,"** contains information about the memory utilities and files on the diskette that accompanies the *Windows Resource Kit.*

Part 7: References, Resources, and Appendixes

This part of the *Windows Resource Kit* contains a glossary, a directory of training and other information resources, a list of hardware supported by Microsoft Windows 3.1, instructions for creating a minimum Windows "footprint," and a comprehensive index. There is also a special section at the end of this appendix with technical articles describing Microsoft Windows for Pens, FastDisk, and other technical topics.

Conventions in This Manual

This document assumes that you have the complete Windows 3.1 documentation set and that you are familiar with using the menus, dialog boxes, and other Windows features. It also assumes that you have installed Windows on your system, and that you are using a mouse with Windows. For keyboard equivalents to actions described here, see the Microsoft Windows online help.

This document uses several conventions to help you identify information.

Document Conventions

The following table describes the typographical conventions used in the *Windows Resource Kit.*

Type style	Used for
bold	MS-DOS command names such as **copy** or **dir**; switches such as **/?** or **/3**; section and entry names in .INI and .INF files such as **[386enh]** or **emmexclude=**; and any text that you type to carry out actions at the command prompt.
italic	Parameter values for which you can supply specific values. For example, to supply a value for a parameter that calls for a *filename*, you must type a specific filename such as MYFILE.EXE.
ALL CAPITALS	Directory names, filenames, and acronyms. For example, "WINDOWS" is used to represent the Windows main directory, and "SYSTEM" represents the Windows System subdirectory. When you type directory names and filenames at the command prompt or in a dialog box, you can use lowercase letters.

Other conventions in this document include:



- "Windows" refers to Microsoft Windows version 3.1 or later.
- "MS-DOS" refers to Microsoft MS-DOS® version 3.1 or later.
- The Microsoft Windows logo appears in the margin to indicate features that are new in Windows 3.1.
- "Windows application" is used as a shorthand term to refer to an application that is designed to run with Windows and does not run without Windows. All Windows applications follow similar conventions for arrangement of menus, style of dialog boxes, and keyboard and mouse use.

- "Non-Windows application" is used in this document as a shorthand term to refer to an application that is designed to run with MS-DOS but not specifically with Windows and that may not be able to take full advantage of all Windows features (such as memory management).
- "Command prompt" refers to the command line where you type MS-DOS commands. Typically, you see characters such as "C:\>" to show the location of the command prompt on your screen. When Windows is running, you can double-click the MS-DOS Prompt icon in Program Manager to use the command prompt.
- An instruction to "type" any information means to press a key or a sequence of keys, then press the ENTER key.
- Mouse instructions in this document, such as "Click the OK button" or "Drag an icon in File Manager," use the same meanings as the descriptions of mouse actions in the *Windows User's Guide* and the Windows online tutorial.

Check the glossary at the end of the *Windows Resource Kit* for definitions of terms that you are unfamiliar with.

Syntax Conventions

"Syntax" refers to the order in which you must type an item such as an MS-DOS command with its switches or an entry in a Windows initialization (.INI) file. Elements that appear in bold must be typed exactly as they appear in the syntax example. Elements that appear in italic are placeholders for parameter values, for which you must supply specific information.

Unless specified otherwise, you can type commands, keynames, parameters, and switches in either uppercase or lowercase letters.

This example shows the syntax for a SYSTEM.INI entry, with each item in the sample explained in the following table.

[*section*]
keyname=value, string, source, destination

Entry item	Meaning
[*section*]	The name of a section in an initialization file or setup information file. The enclosing brackets ([]) are required, and the left bracket must be in the leftmost column on the screen. For example, **[standard]**.
keyname	The name of an entry, which usually can consist of any combination of letters and digits. For many entries described in this document, the *keyname* must be followed immediately by an equal sign (=). For example, **run=**.
value	An integer, a string, or a quoted string, depending on the entry.
string	A group of characters to be treated as a unit. A string can include letters, numbers, spaces, or any other characters. Sometimes the syntax definition will indicate that the string must be enclosed in double quotation marks (" ").
source	The location of data to be transferred to a specific destination or to be used as input to a command. *Source* can consist of a drive letter and colon, a directory name, a filename, or a combination of these elements.
destination	A location to which the data specified by *source* is to be transferred. *Destination* can consist of a drive letter and colon, a directory name, a filename, or a combination of these elements.

Note If you have MS-DOS version 5.0 on your system, you can get help for any MS-DOS commands such as **mem** and for many of the drivers such as SMARTDRV.EXE by typing the command name and **/?** at the command prompt (for example, **mem /?**). Or type **help** plus the command name. Type **help** at the command prompt to see a list of all MS-DOS commands with a brief description of command syntax, parameters, and switches.

Troubleshooting Charts



The Troubleshooting Flowcharts for Windows 3.1

The troubleshooting flowcharts in this introduction to the *Windows Resource Kit* provide an orientation and quick introduction to problem-solving for Windows 3.1. Each chart presents a brief statement of the problem with a decision tree for determining the source and solution. Each chart also includes pointers to related technical details in the *Windows Resource Kit.*

Flowchart 1.1
System Requirements

This icon appears throughout the *Windows Resource Kit* to refer to a flowchart that shows a specific solution to a problem.

The flowcharts are organized in the following sequence.

Flowchart Series 1: Troubleshooting Windows Setup ***page***

Chart 1.1: System Requirements for Windows 9
Chart 1.2: Windows Hangs During Setup 10
Chart 1.3: Setup /n Fails 11
Chart 1.4: Known Problems with Computer BIOS 12
Chart 1.5: Creating 'Clean Boot' Floppy for MS-DOS 4 or Earlier 13
Creating 'Clean Boot' Floppy for MS-DOS 5 14
Chart 1.6: Reinstalling Windows Without Losing Settings 15
Chart 1.7: Expanding Files from the Windows Disks 16
Chart 1.8: Stacker 17

Flowchart Series 2: Troubleshooting Hardware Problems ***page***

Chart 2.1: Mouse Doesn't Work in Windows 18
Chart 2.2: Serial Port Problems (Communications and Terminal) 19
Chart 2.3: Display Adapter Problems (Distorted Video) 20
Chart 2.4: Cannot Access CD-ROM in 386 Enhanced Mode 21
Chart 2.5: Cannot Access Floppy Drives in 386 Enhanced Mode 22
Chart 2.6: Cannot Access Hard Disk in 386 Enhanced Mode 23
Chart 2.7: Common IRQ Usage for PC-AT and 80386 Computers 25
Chart 2.8: Mouse Doesn't Work in Windowed Non-Windows Applications .. 26

Flowchart Series 3: Troubleshooting Windows Fonts ***page***

Chart 3.1: TrueType Fonts 27
Chart 3.2: Disabling Adobe Type Manager 28
Chart 3.3: Disabling Bitstream FaceLift 29
Chart 3.4: Disabling HP Intellifont 29

Flowchart Series 4: Troubleshooting Windows Printing ***page***

Chart 4.1: Cannot Print on Stand-Alone PC 30
Chart 4.2: Cannot Print on Network 31
Chart 4.3: Garbled Output 32
Chart 4.4: Garbled Output on Novell Network 33

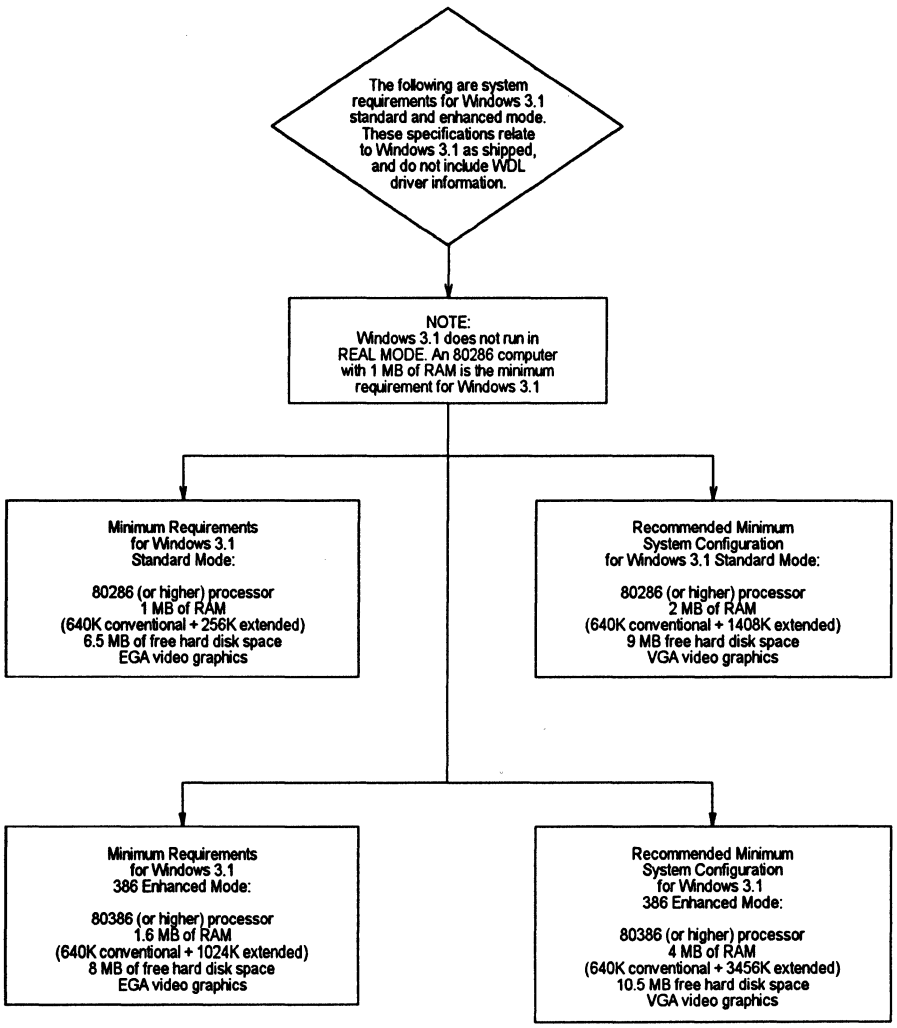
Flowchart Series 5: Troubleshooting Windows Operating Problems ***page***

Chart 5.1: Cannot Run 386 Enhanced Mode 34
Chart 5.2: Cannot Run 386 Enhanced Mode on Network Workstation 36
Chart 5.3: General Protection Faults 37
Chart 5.4: Application Execution Error 39
Chart 5.5: System Integrity Violation in 386 Enhanced Mode 40
Chart 5.6: Disabling Norton Desktop for Windows 41
Chart 5.7: Disabling PC-Tools Version 7 42
Chart 5.8: Out-of-Memory Errors in Windows 43
Chart 5.9: Cannot Run Windows in Standard Mode 44
Chart 5.10: Problems with Object Linking and Embedding 45

Flowchart Series 6: Troubleshooting Multimedia Windows ***page***

Chart 6.1: Multimedia Drivers 46
Chart 6.2: Media Player 47
Chart 6.3: MIDI Mapper 48
Chart 6.4: Music Box 49
Chart 6.5: Sound 50
Chart 6.6: Sound Recorder 51

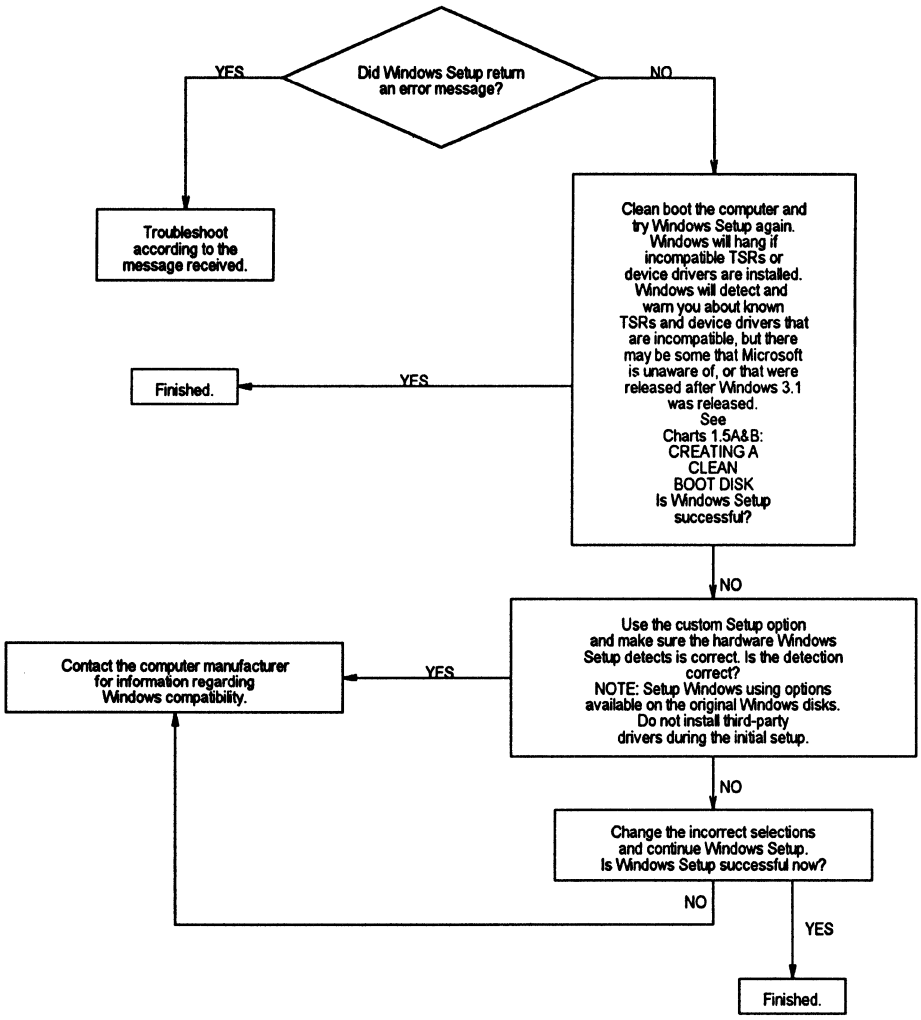
Chart 1.1 System Requirements for Windows



Windows Resource Kit Reference ***page***

"Memory and Windows Startup Requirements" in Chapter 5 249

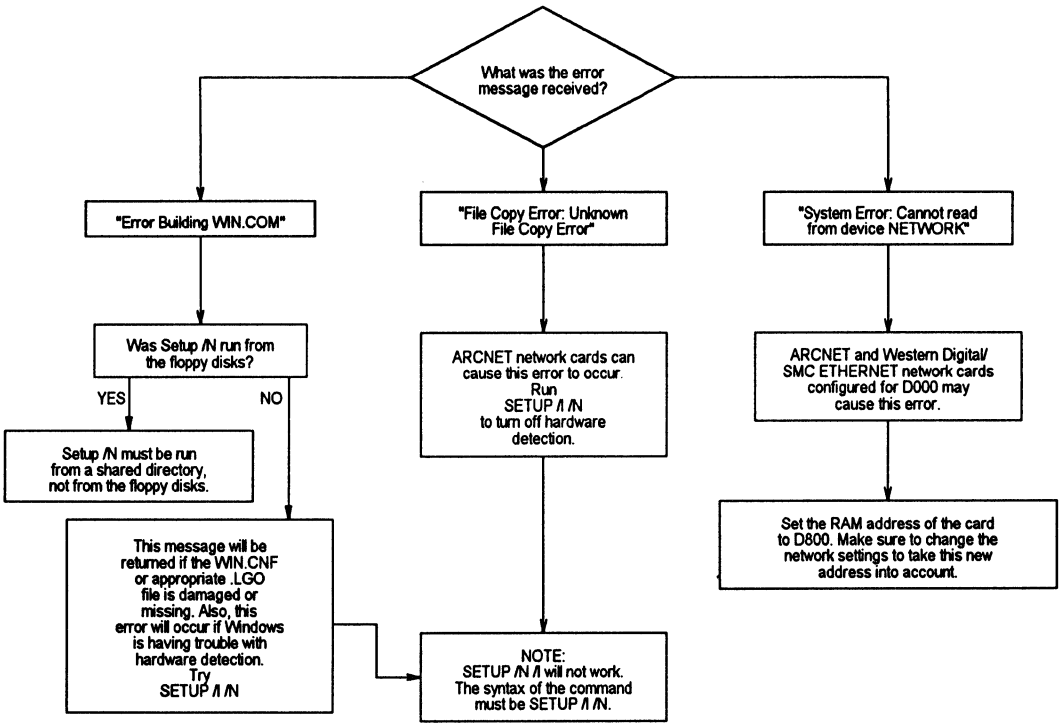
Chart 1.2 Windows Hangs During Setup



Windows Resource Kit Reference ***page***

"Windows Setup: A Technical Discussion" in Chapter 1 62
"Troubleshooting Setup" in Chapter 13 399

Chart 1.3 Setup /n Fails



Windows Resource Kit Reference ***page***

"Setting Up Windows for a Network" in Chapter 1 71

Chart 1.4 Known Problems with Computer BIOS

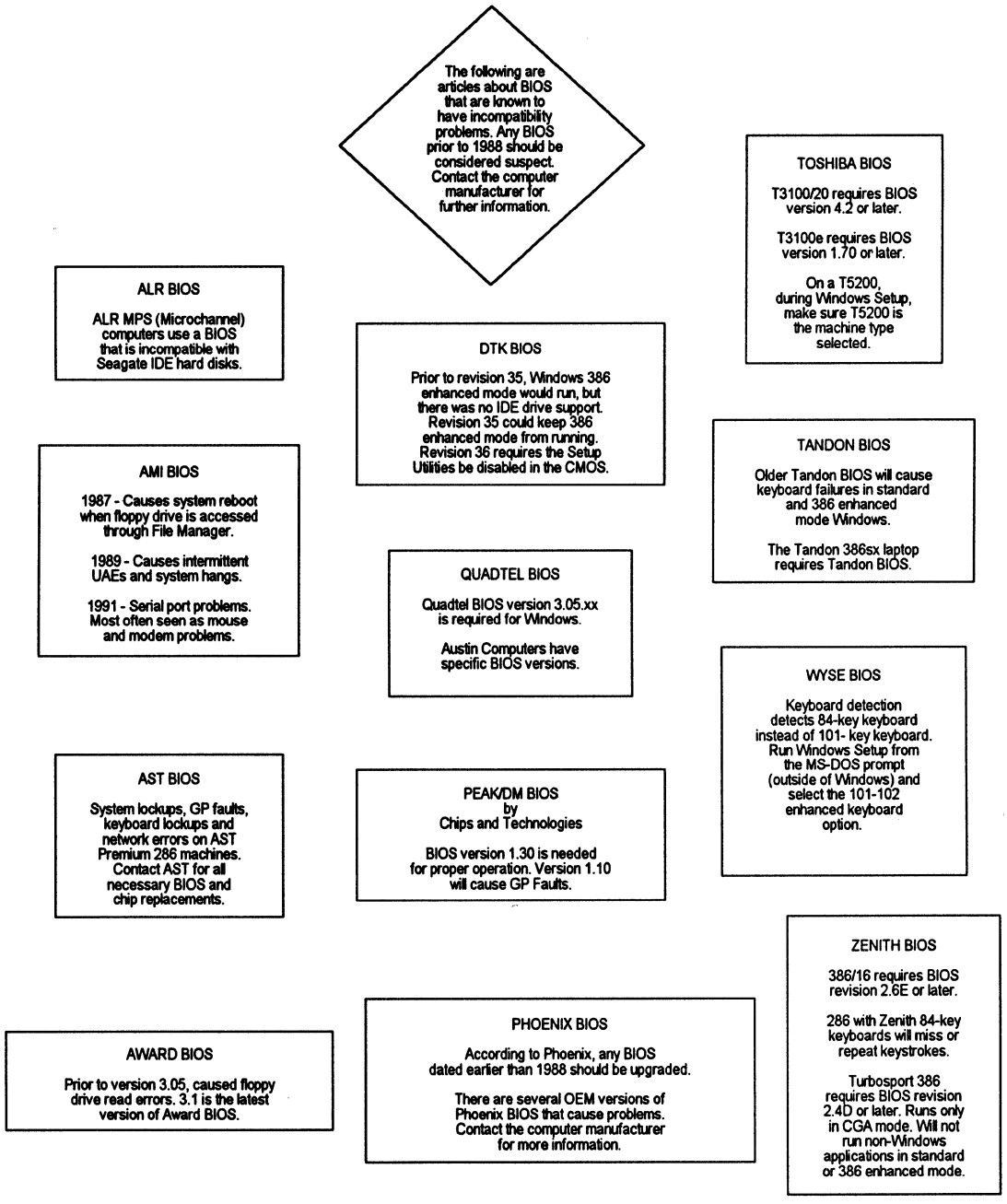
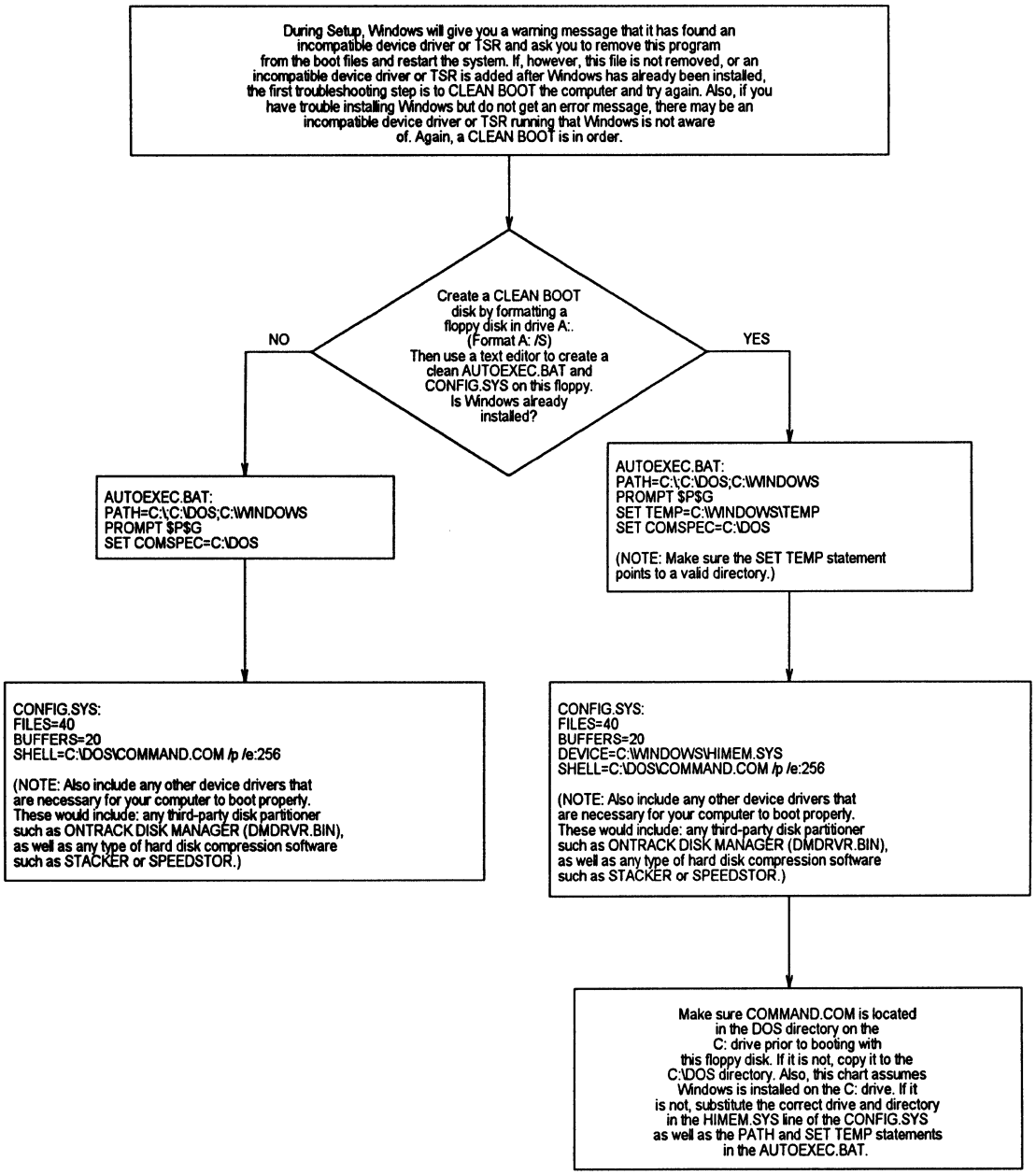


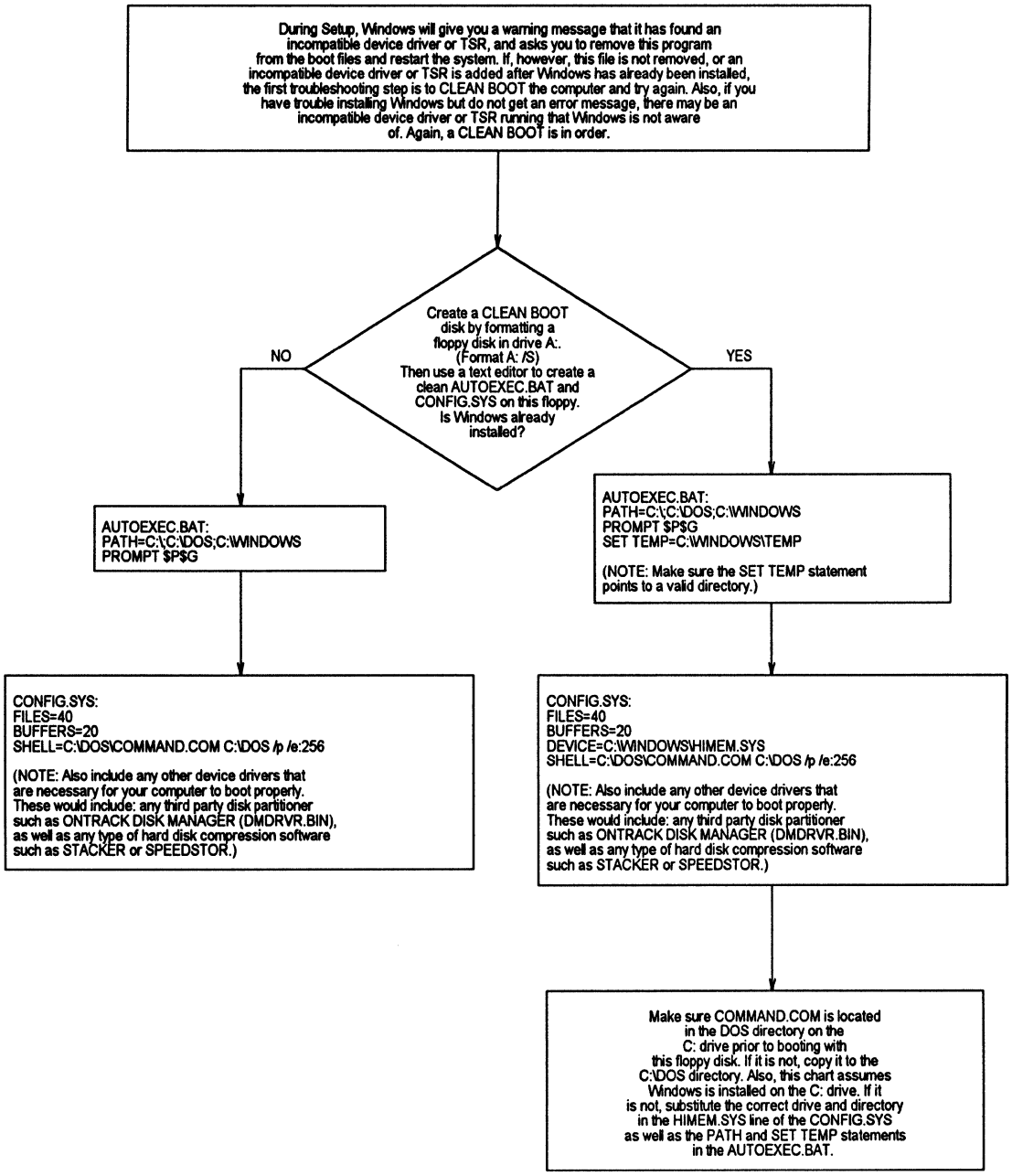
Chart 1.5A Creating 'Clean Boot' Floppy for MS-DOS 4 or Earlier



Windows Resource Kit Reference ***page***

"Creating a 'Clean Boot' for Troubleshooting" in Chapter 13 398

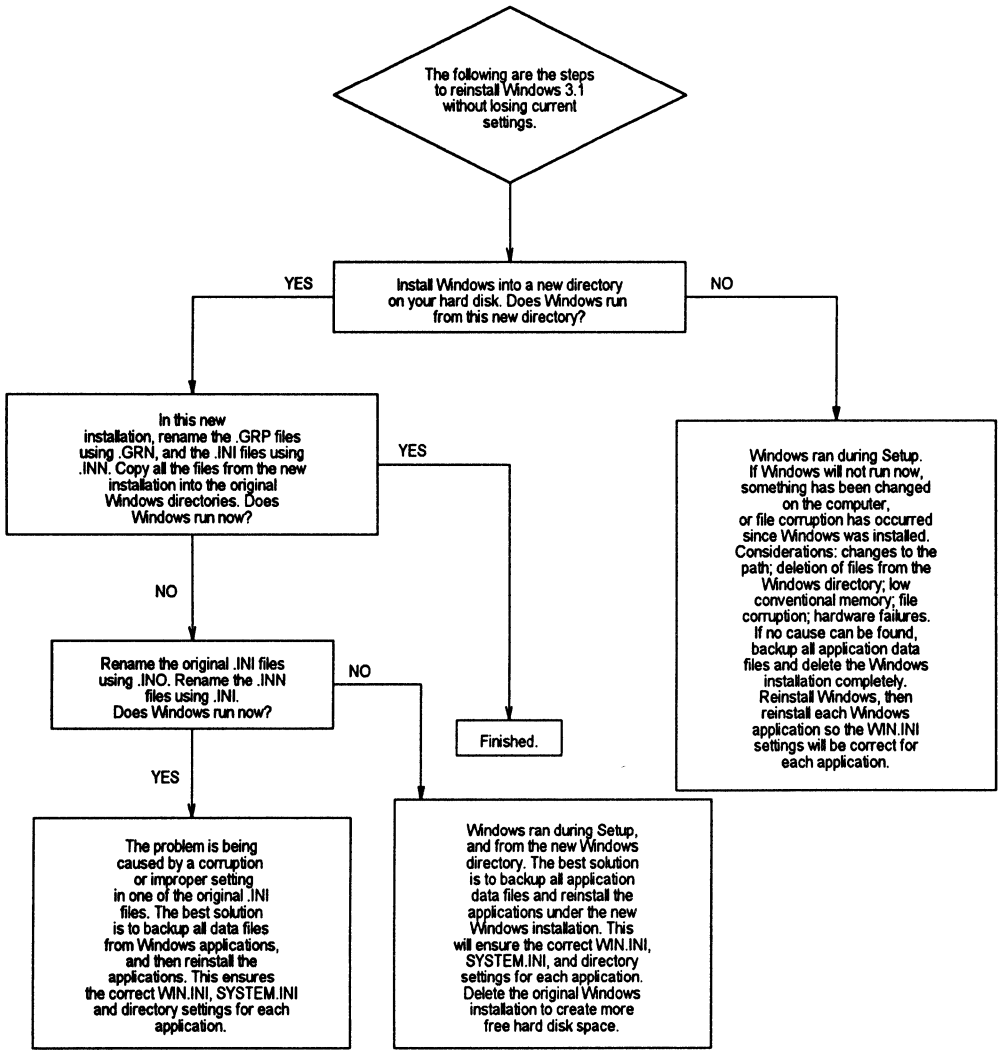
Chart 1.5B Creating 'Clean Boot' Floppy for MS-DOS 5



Windows Resource Kit Reference ***page***

"Creating a 'Clean Boot' for Troubleshooting" in Chapter 13..... 398

Chart 1.6 Reinstalling Windows Without Losing Settings



Other References

Chapter 15, "Maintaining Windows with Setup," in *Windows User's Guide*

Chart 1.7 Expanding Files from the Windows Disks

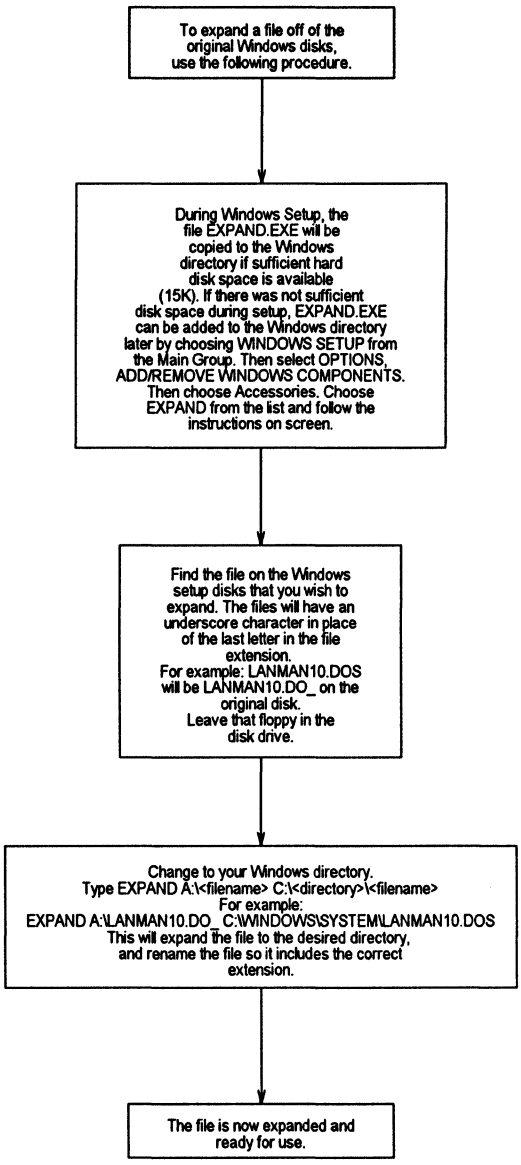


Chart 1.8 Stacker

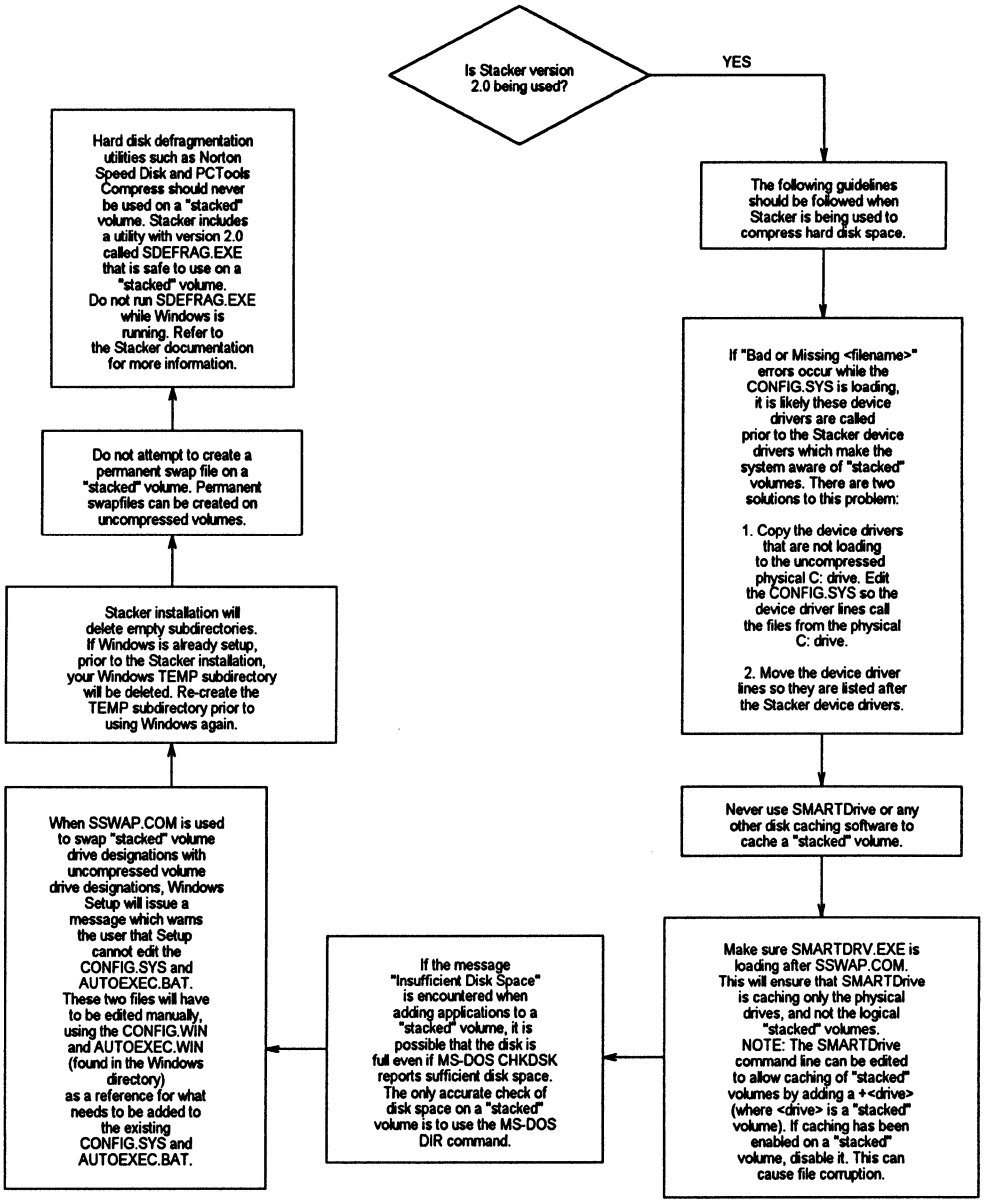
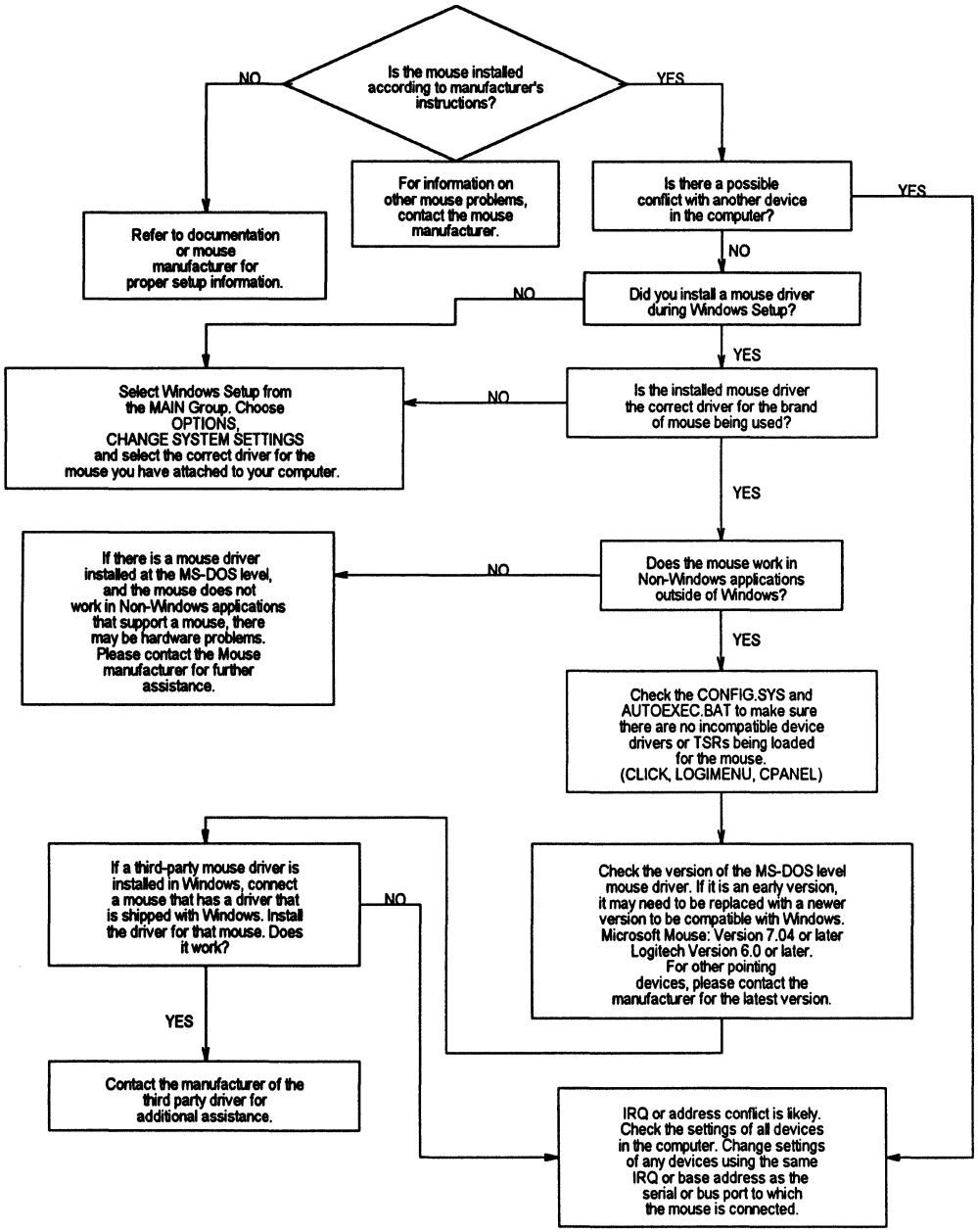


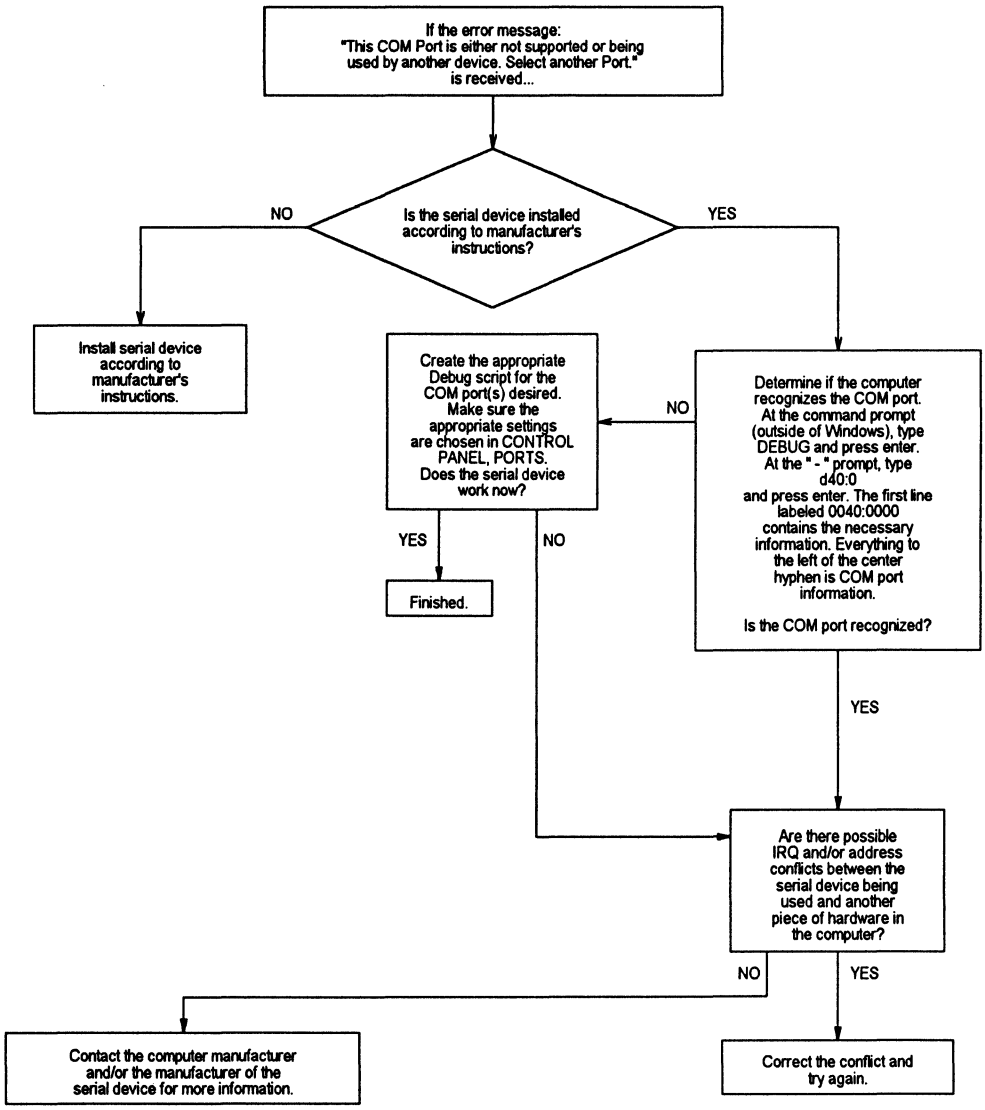
Chart 2.1 Mouse Doesn't Work in Windows



Windows Resource Kit Reference ***page***

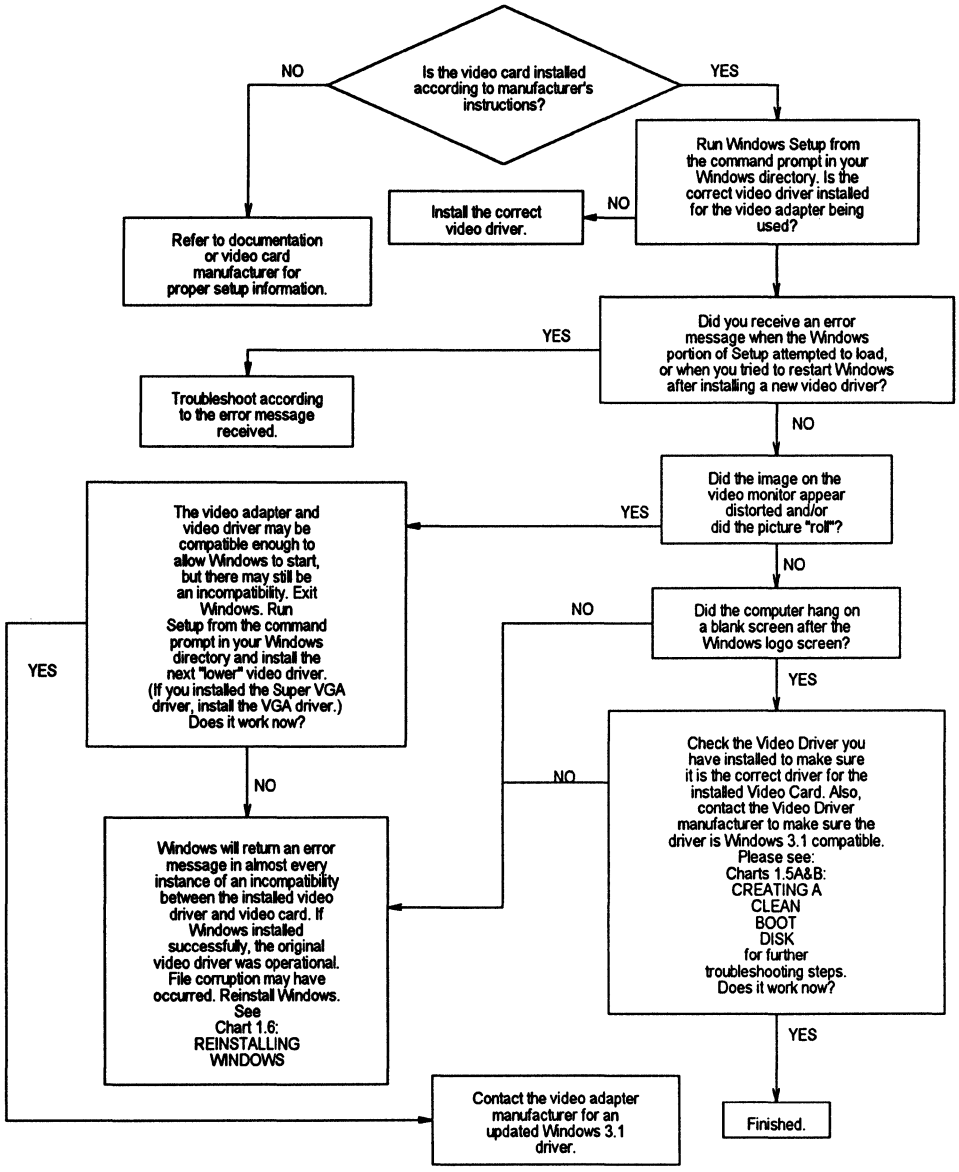
Flowchart 2.8: "Mouse Doesn't Work in Non-Windows Applications" 426

Chart 2.2 Serial Port Problems (Communications and Terminal)



Windows Resource Kit Reference	*page*
"Printing from Serial Ports" in Chapter 10	343
"Tips for Specific Printing Problems" in Chapter 13	428

Chart 2.3 Display Adapter Problems (Distorted Video)



Windows Resource Kit Reference ***page***

"Troubleshooting Display Problems" in Chapter 13 408

Chart 2.4 Cannot Access CD-ROM in 386 Enhanced Mode

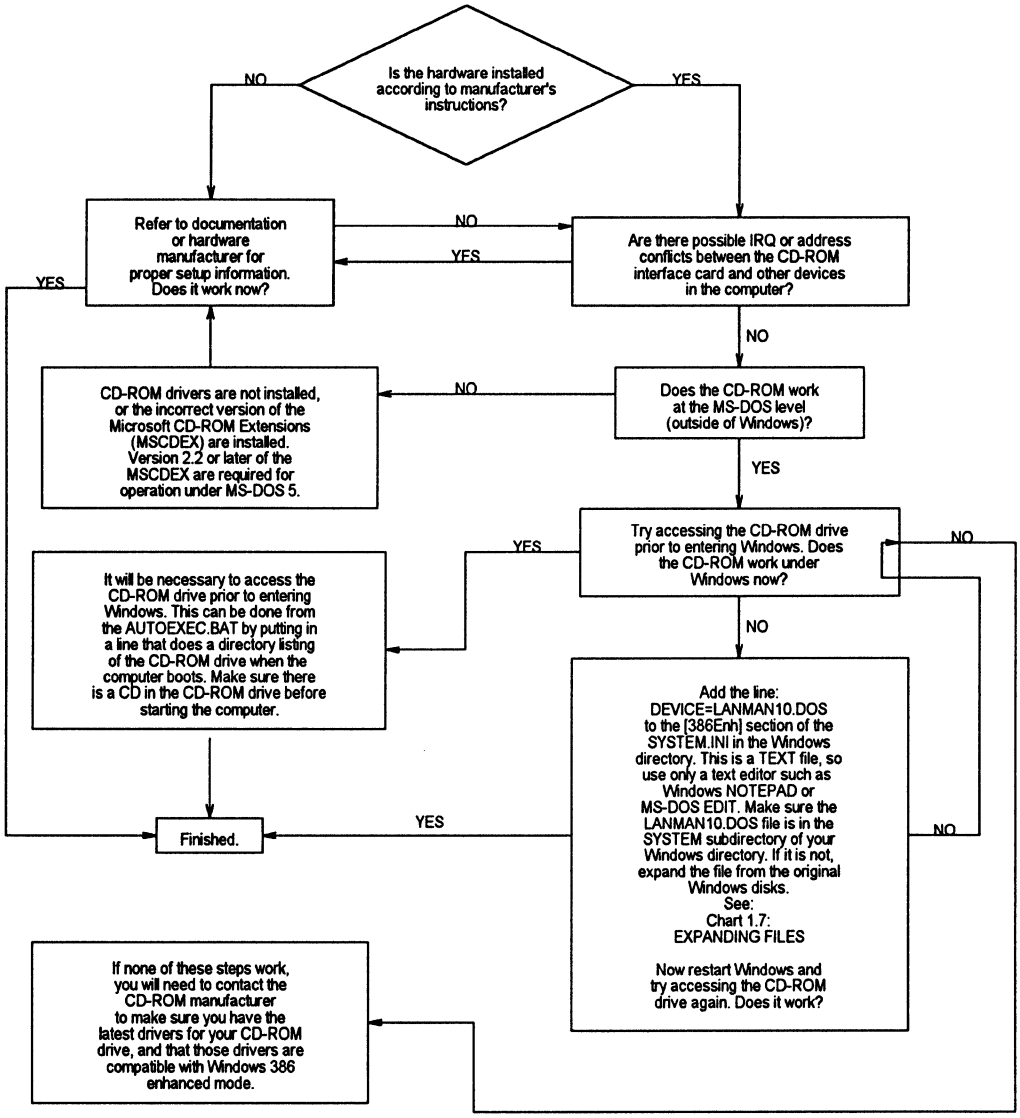


Chart 2.5A Cannot Access Floppy Drives in 386 Enhanced Mode

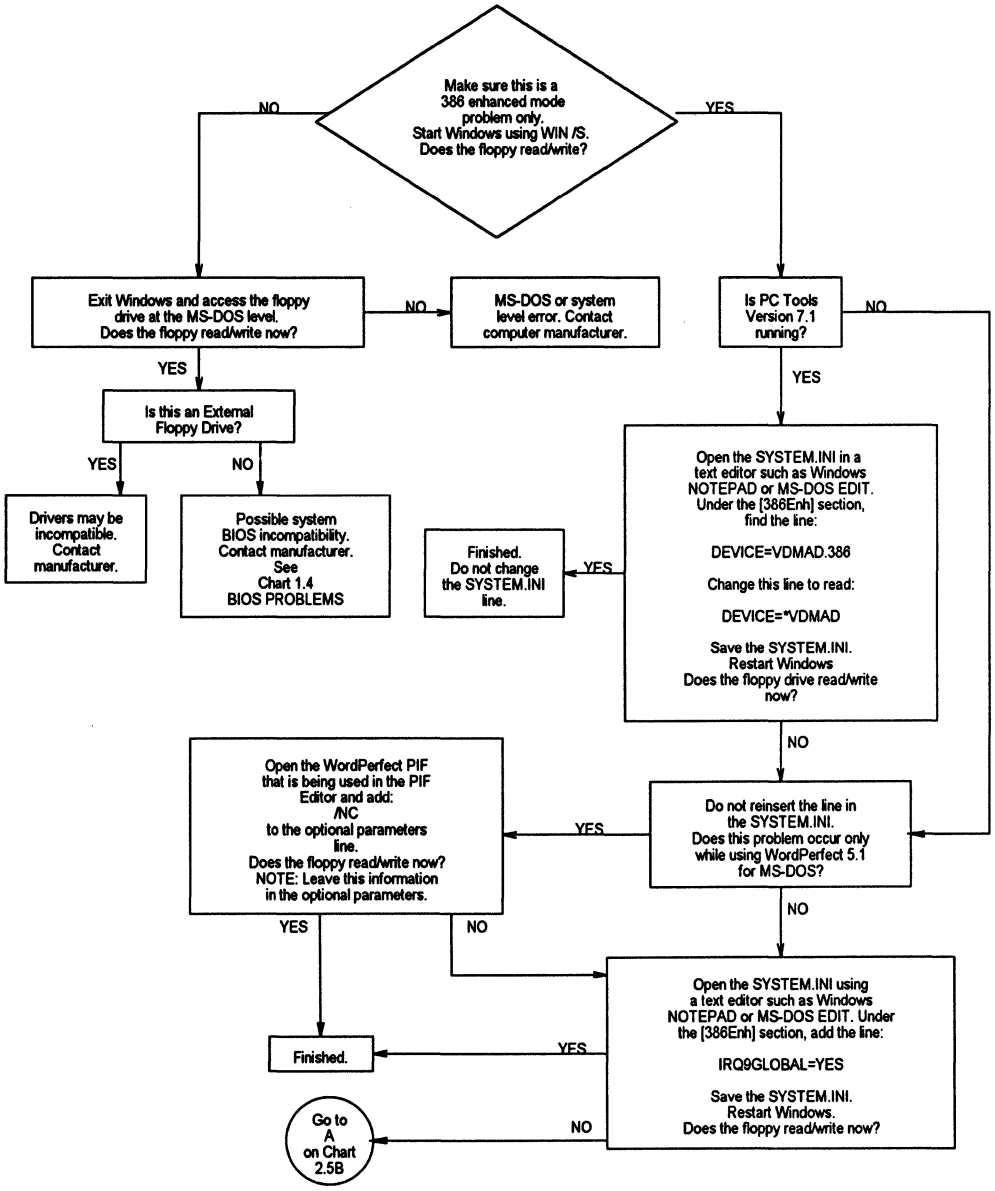


Chart 2.5 B Cannot Access Floppy Drives in 386 Enhanced Mode

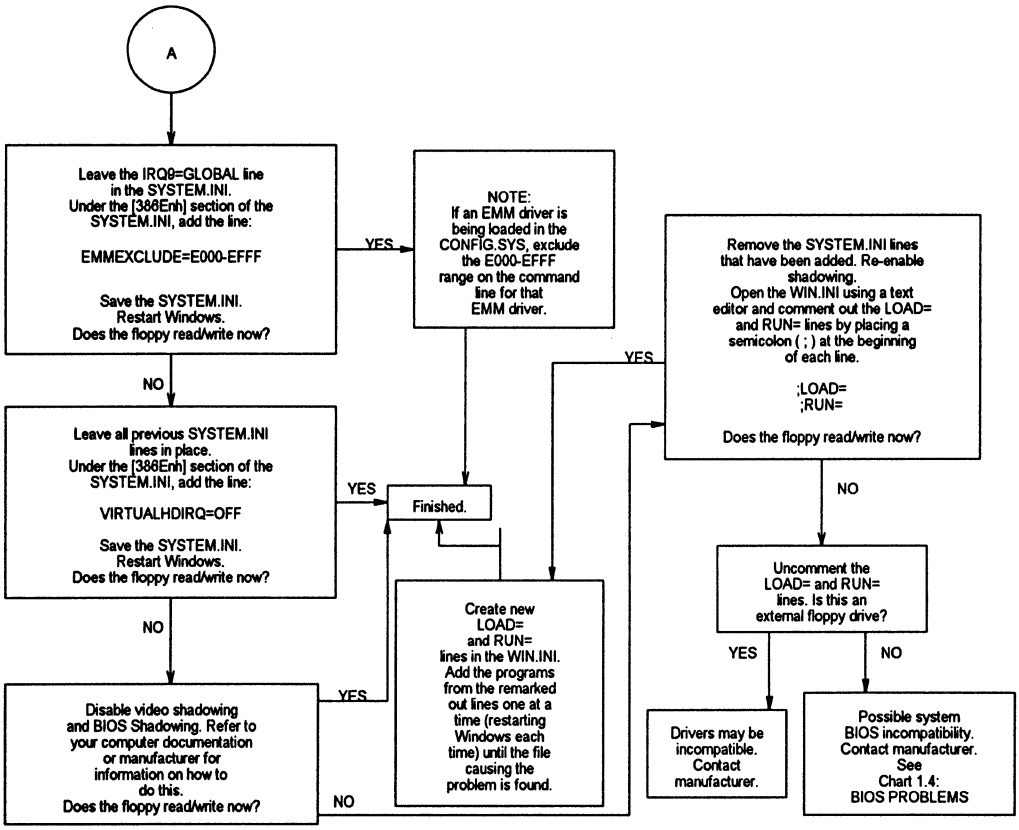
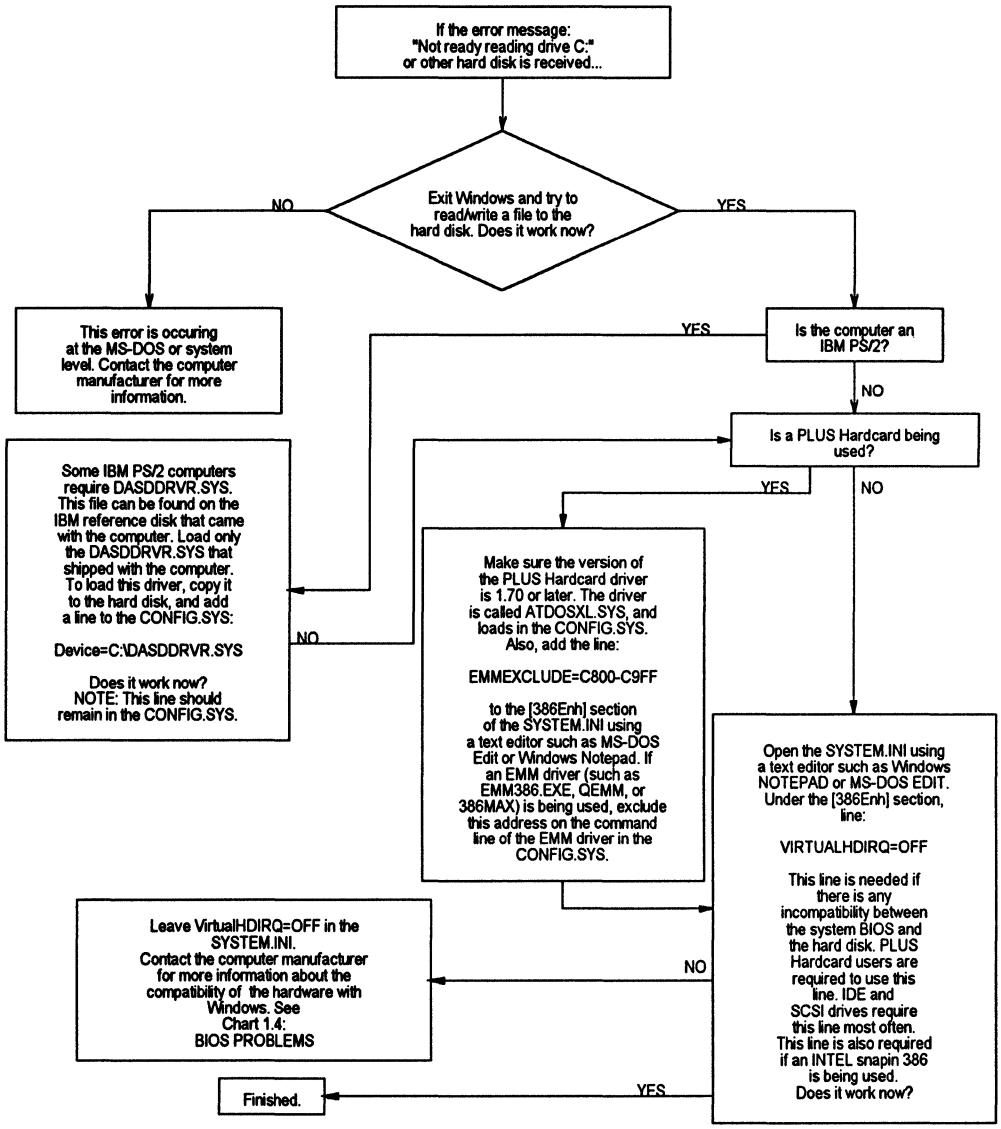


Chart 2.6 Cannot Access Hard Disk in 386 Enhanced Mode



Windows Resource Kit Reference ***page***

"Troubleshooting Windows in 386 Enhanced Mode" in Chapter 13 420

Chart 2.7 Common IRQ Usage for PC-AT and 80386 Computers



PC AT and 386 Computers

IRQ 0 - Timer Interrupt

IRQ 1 - Keyboard Interrupt

IRQ 2 - Programmable Interrupt Controller Cascade (to IRQ 9)

IRQ 3 - COM 2 and COM 4

IRQ 4 - COM 1 and COM 3

IRQ 5 - LPT 2 (Parallel Printer Port)

IRQ 6 - Floppy Disk

IRQ 7 - LPT 1 (Parallel Printer Port)

IRQ 8 - Real Time Clock

IRQ 9 - Cascade to IRQ 2

IRQ 10 - Reserved for future expansion (Open)

IRQ 11 - Reserved for Future Expansion (Open)

IRQ 12 - Reserved for Future Expansion (Open)
Mouse Port on PS/2

IRQ 13 - 80x87 Math Coprocessor

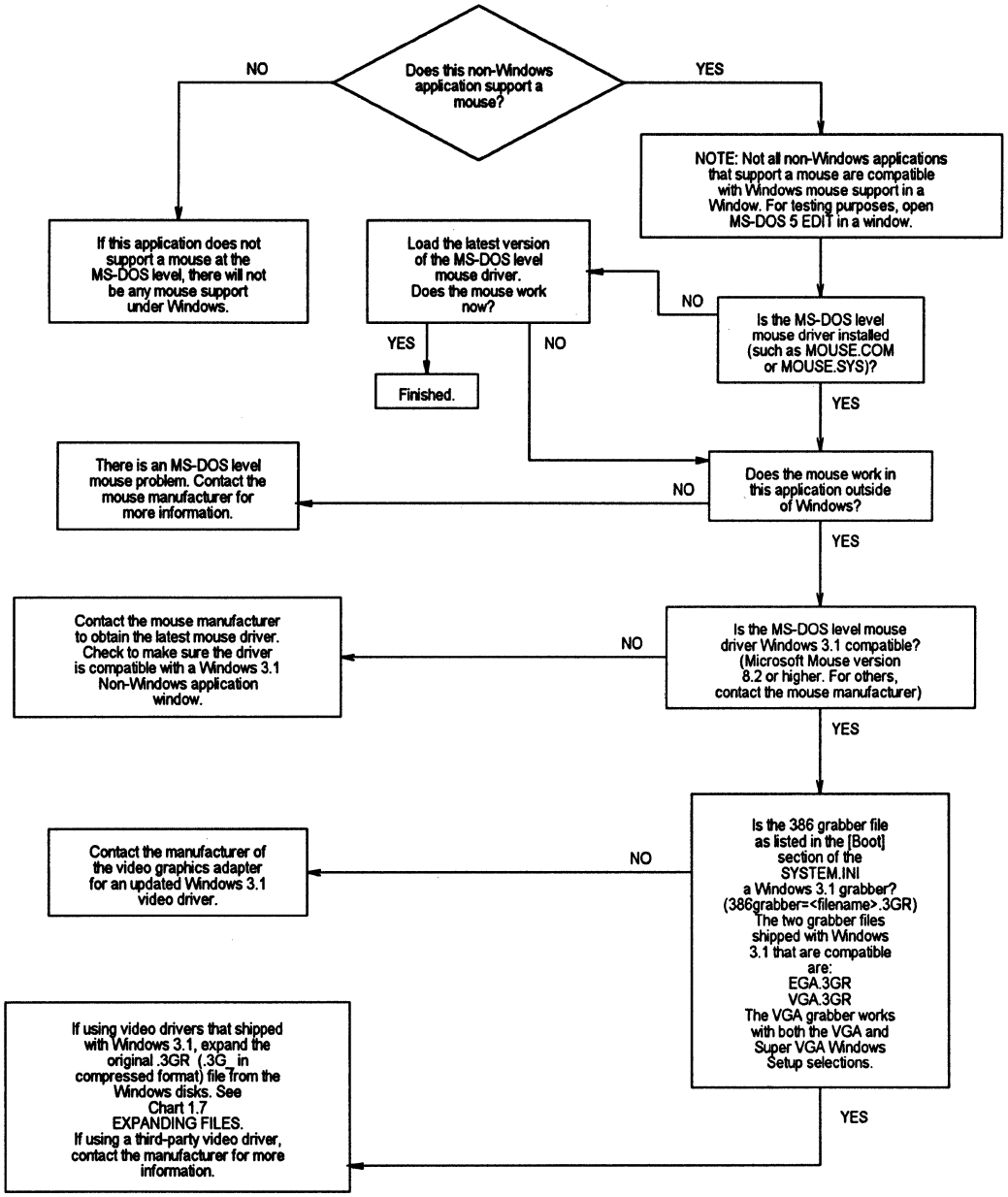
IRQ 14 - Hard Disk

IRQ 15 - Reserved for Future Expansion (Open)

Windows Resource Kit Reference ***page***

"Troubleshooting Networks" in Chapter 13 430

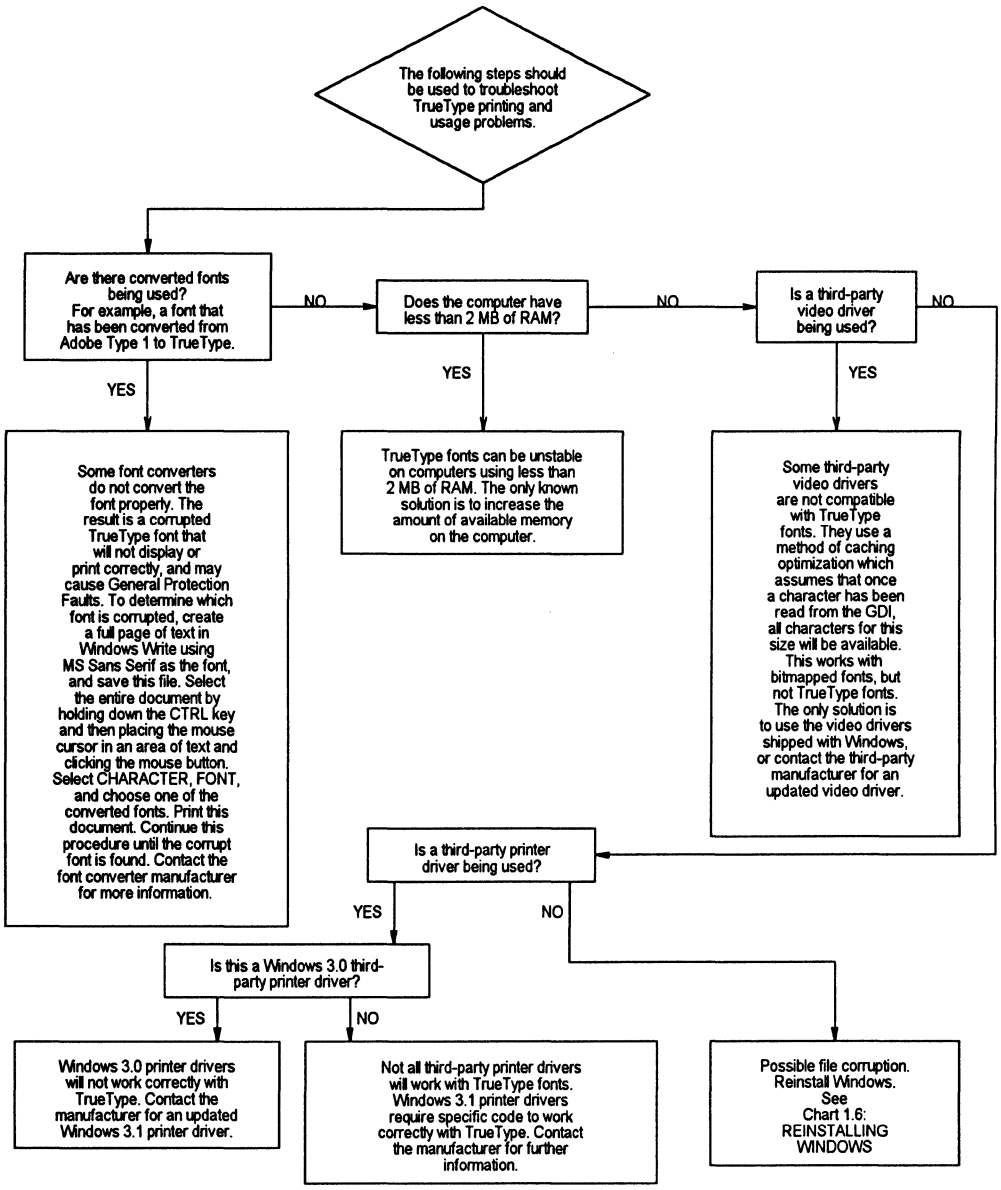
Chart 2.8 Mouse Doesn't Work in Windowed Non-Windows Applications



Windows Resource Kit Reference ***page***

"Troubleshooting Non-Windows Applications" in Chapter 13 423

Chart 3.1 TrueType Fonts



Windows Resource Kit Reference ***page***

"TrueType and Windows 3.1" in Chapter 9 318
"Troubleshooting Fonts" in Chapter 13 425

Chart 3.2 Disabling Adobe Type Manager

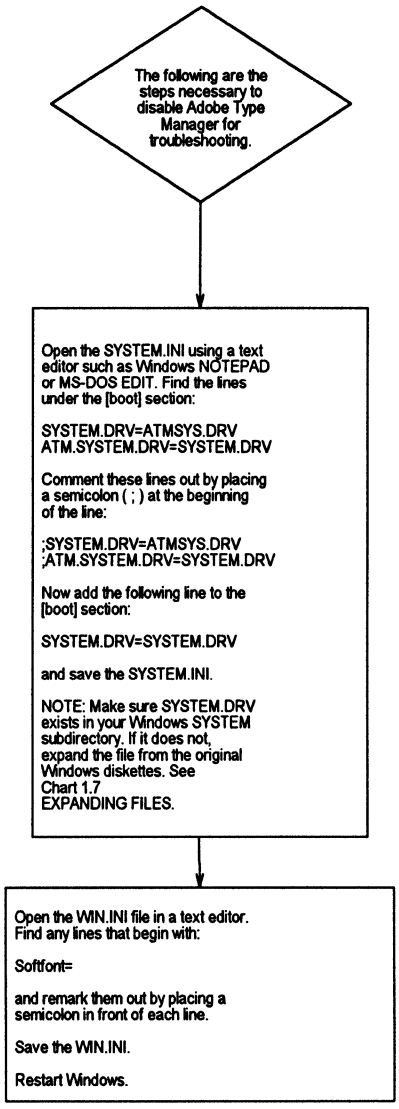


Chart 3.3 Disabling Bitstream FaceLift

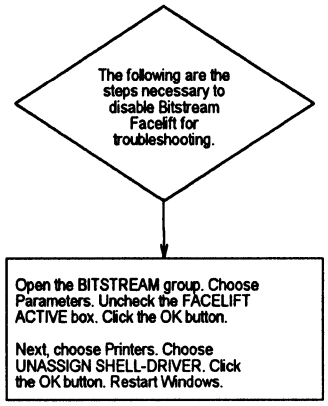


Chart 3.4 Disabling HP Intellifont

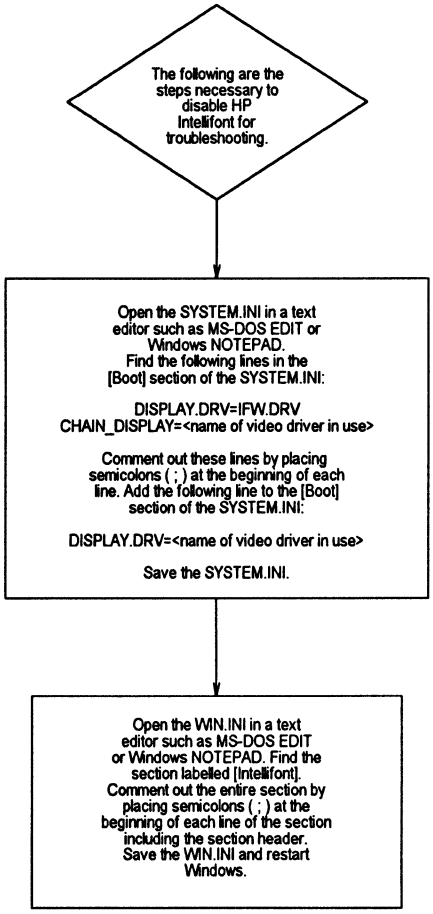
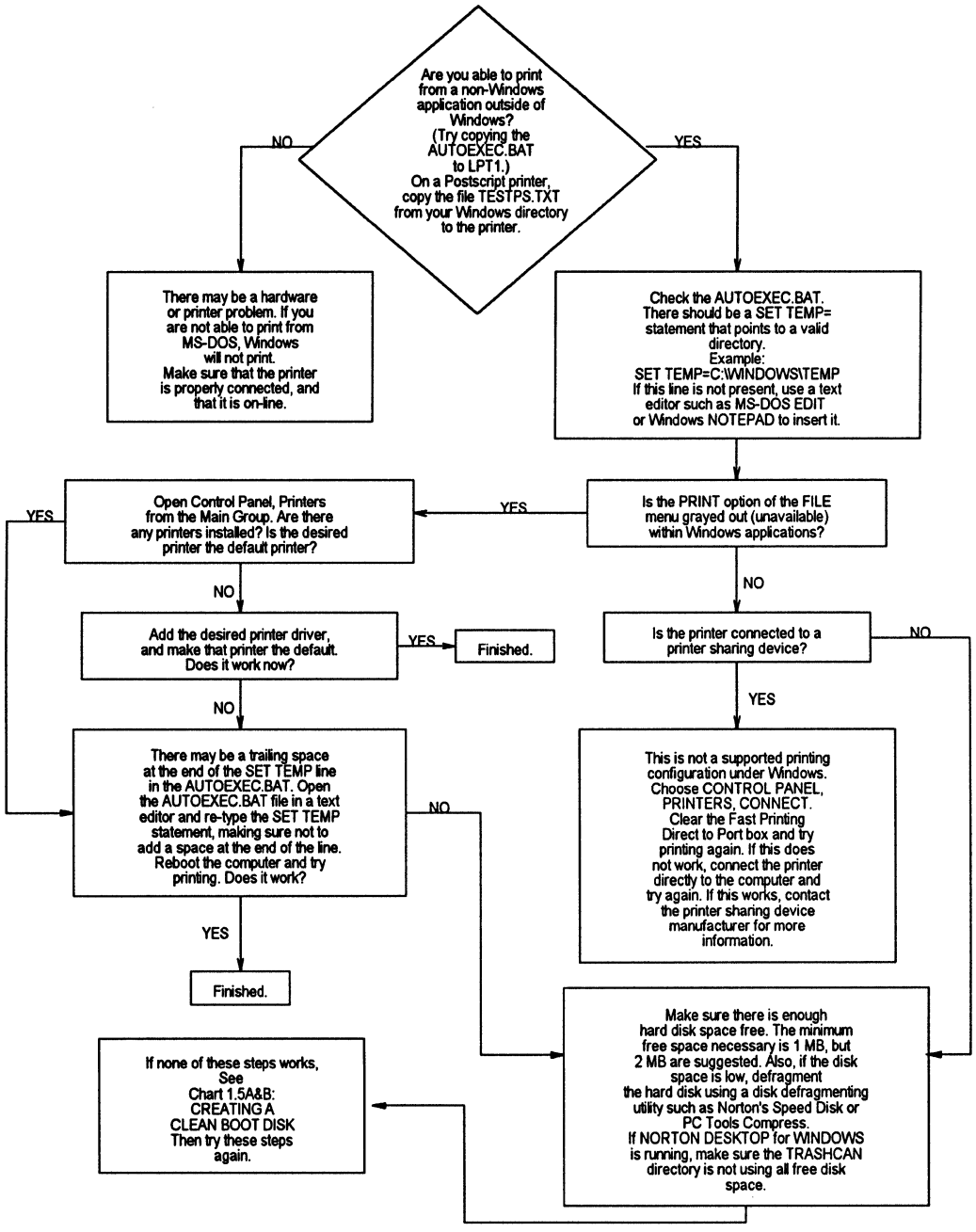
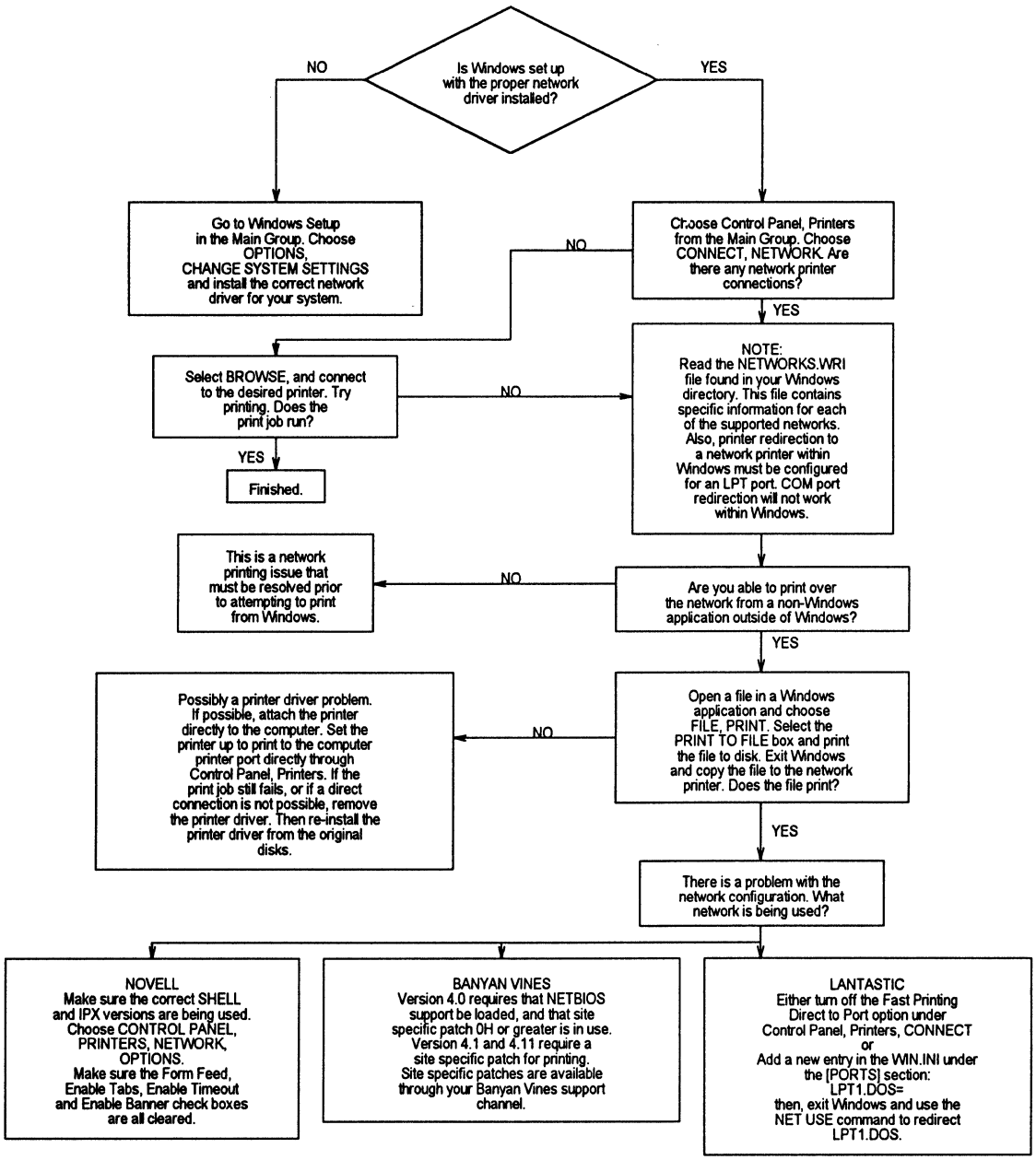


Chart 4.1 Cannot Print on Stand-Alone PC



Windows Resource Kit Reference	***page***
"Determining Print Method" in Chapter 10	349
"Troubleshooting Printing" in Chapter 13	426

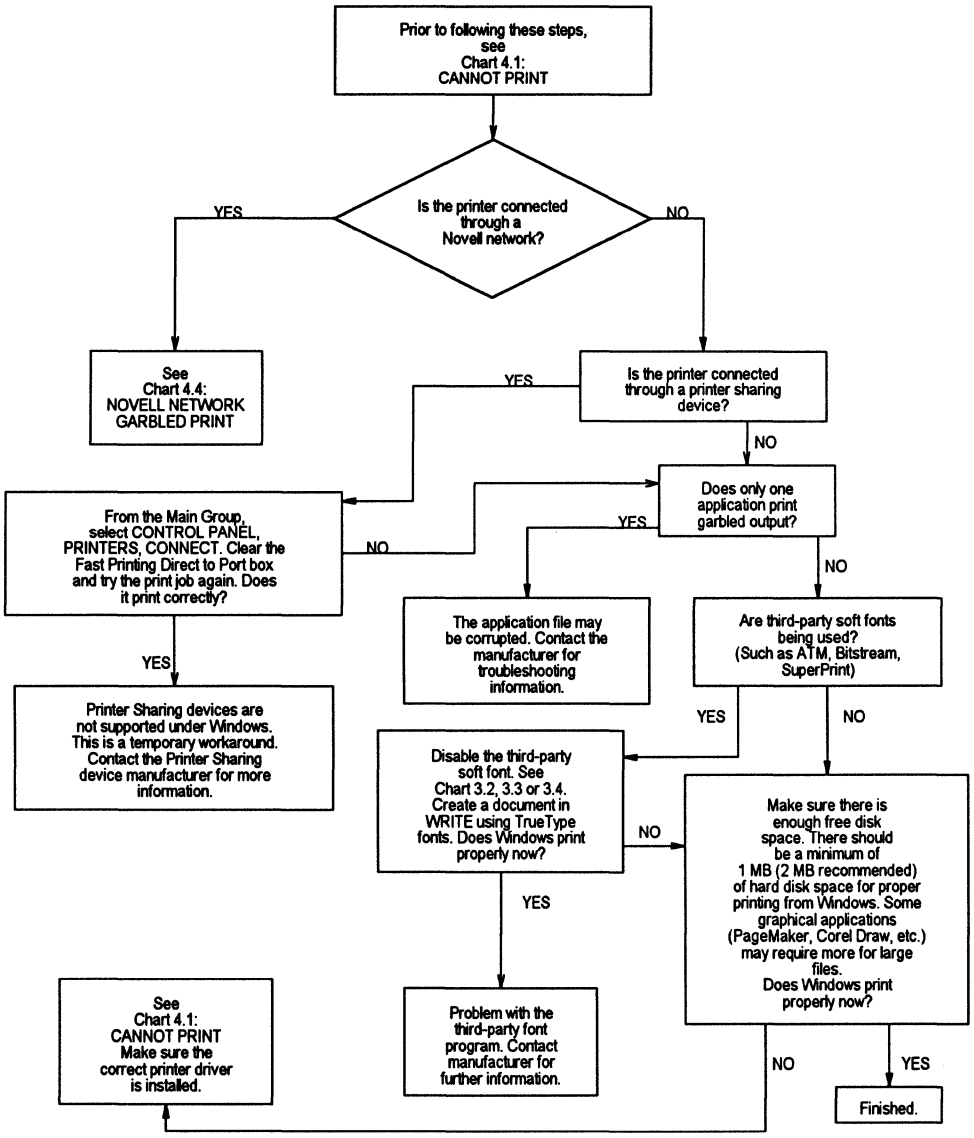
Chart 4.2 Cannot Print on Network



Windows Resource Kit Reference ***page***

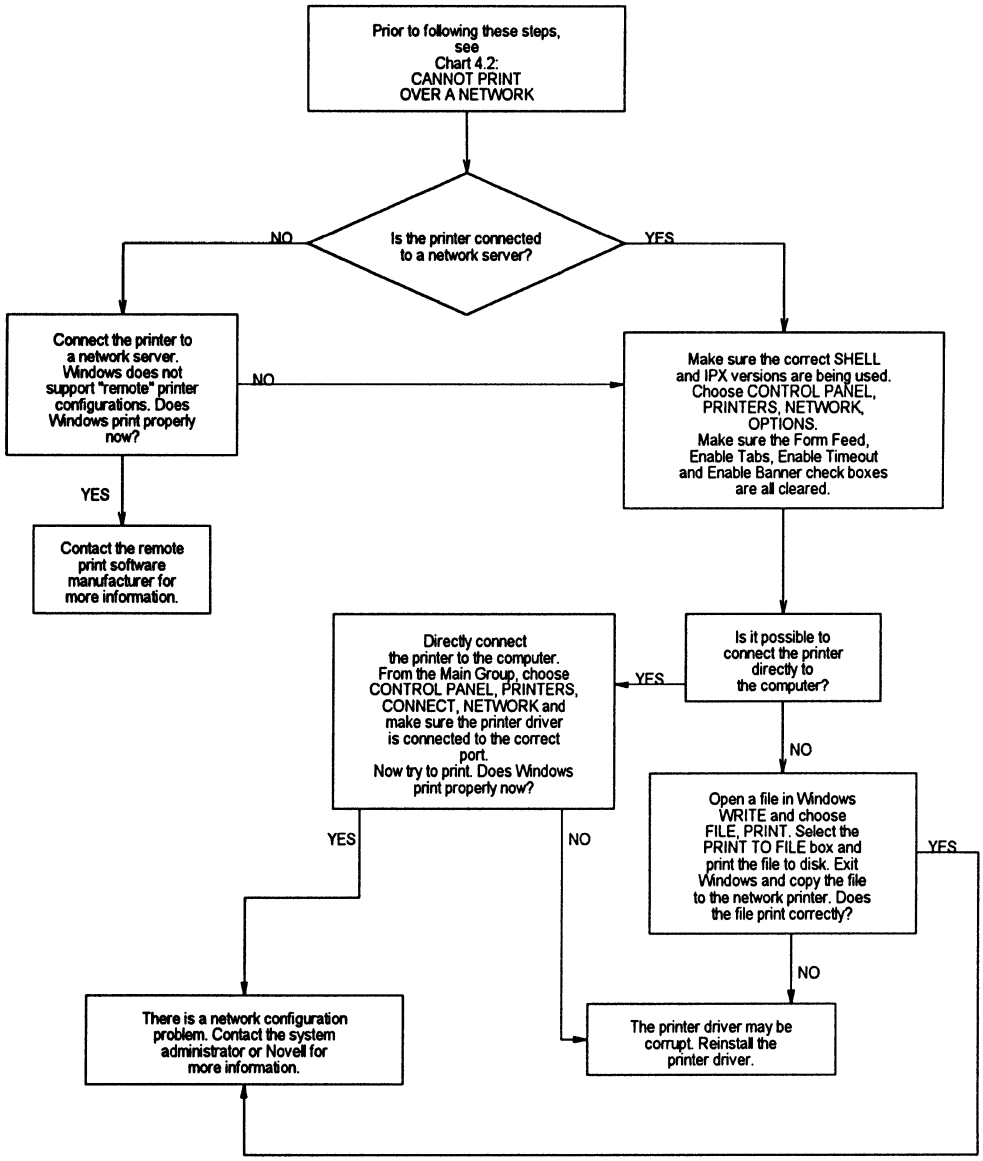
"Printing with a Network Driver" in Chapter 10 350
"Troubleshooting Printing" in Chapter 13..... 426
"Troubleshooting Networks" in Chapter 13 430

Chart 4.3 Garbled Output



Windows Resource Kit Reference	*page*
"Printer Ports" in Chapter 10	342
"Troubleshooting Printing" in Chapter 13	426

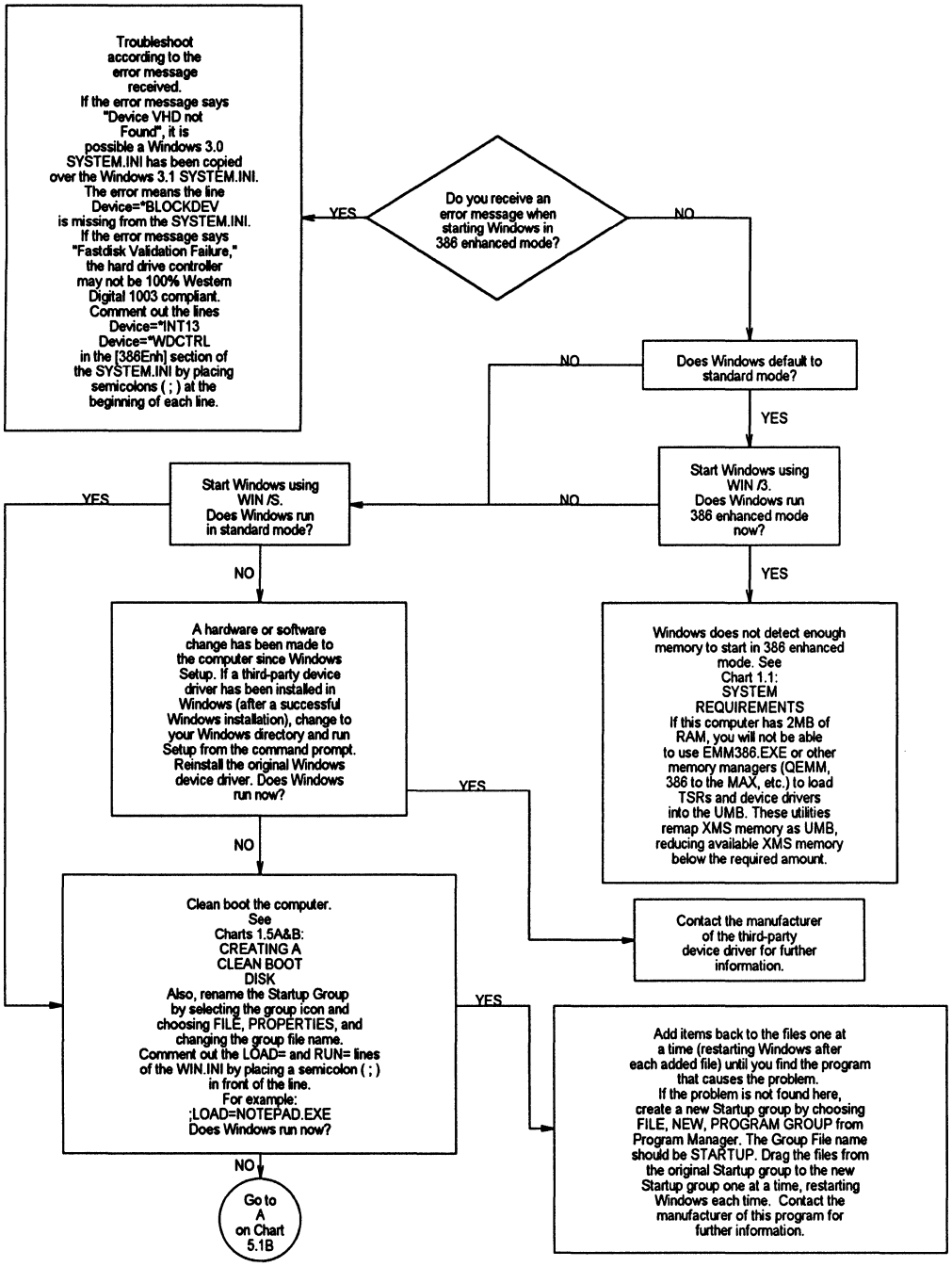
Chart 4.4 Garbled Output on Novell Network



Windows Resource Kit Reference ***page***

"Printing with a Network Driver" in Chapter 10 350
"Novell NetWare" in Chapter 12 387
"Troubleshooting Printing" in Chapter 13 426
"Troubleshooting Networks" in Chapter 13 430

Chart 5.1A Cannot Run 386 Enhanced Mode



Windows Resource Kit Reference page

"Windows 386 Enhanced Mode and Memory" in Chapter 5 238
"Troubleshooting Windows in 386 Enhanced Mode" in Chapter 13 420

Chart 5.1B Cannot Run 386 Enhanced Mode

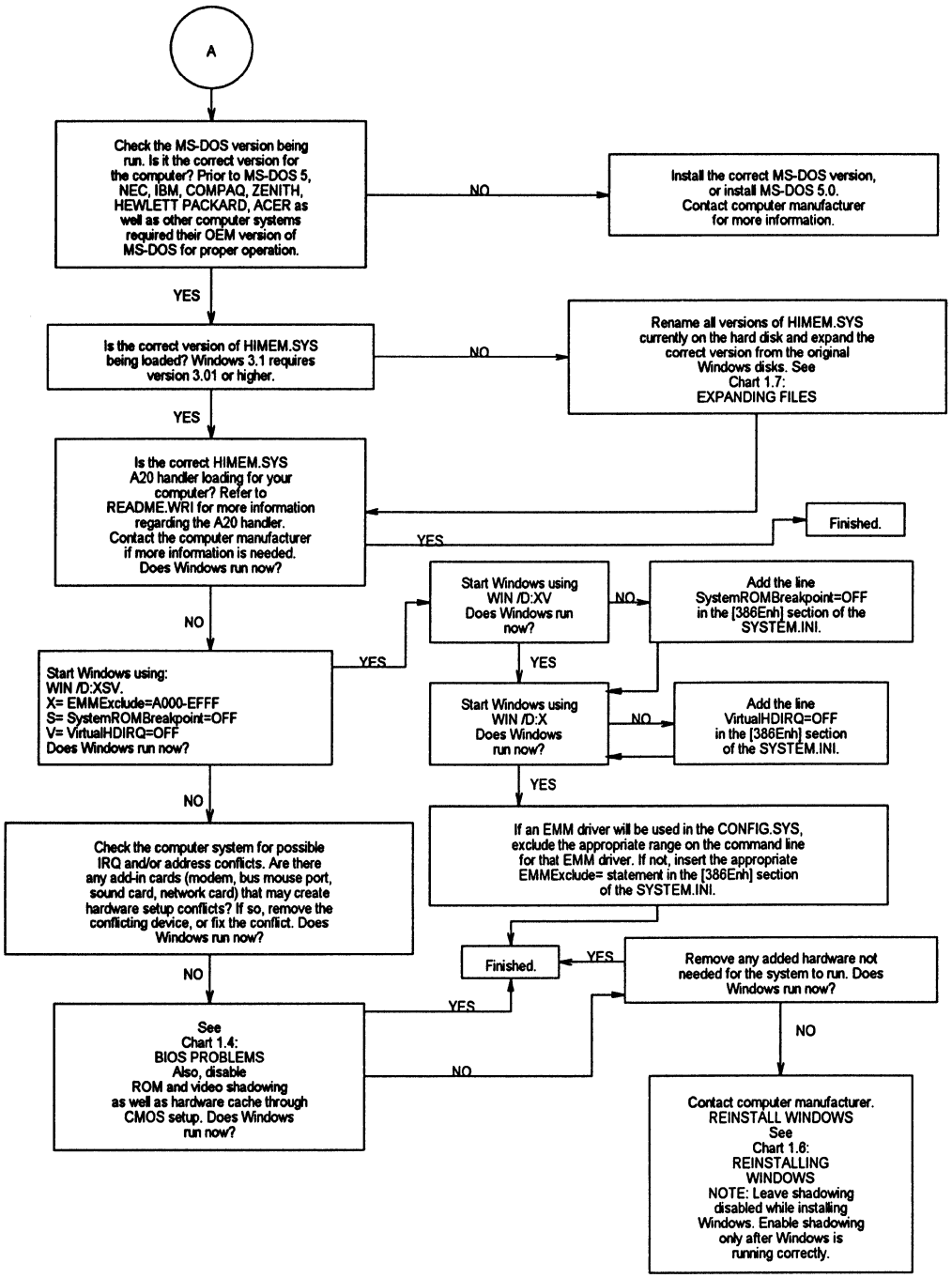
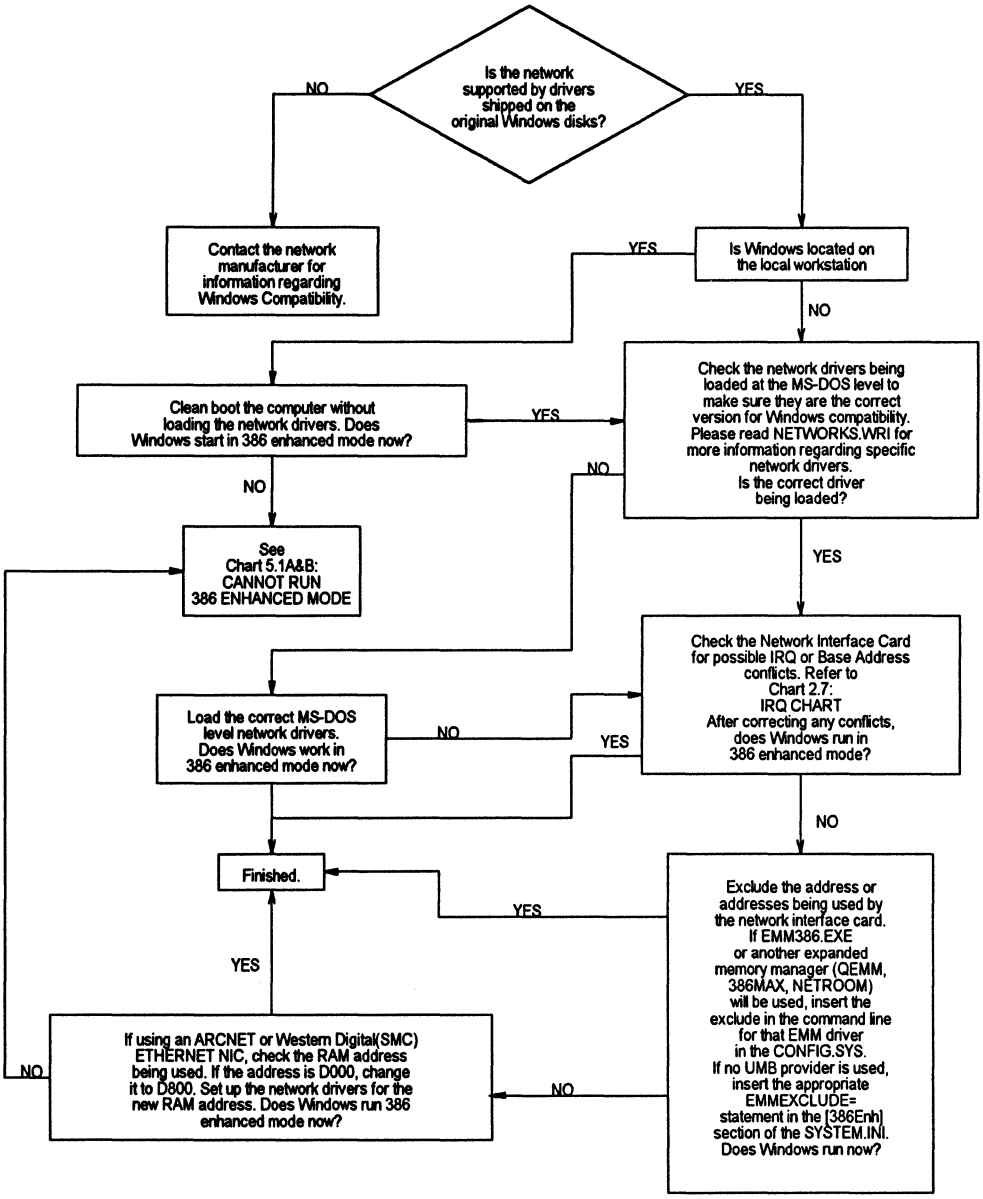


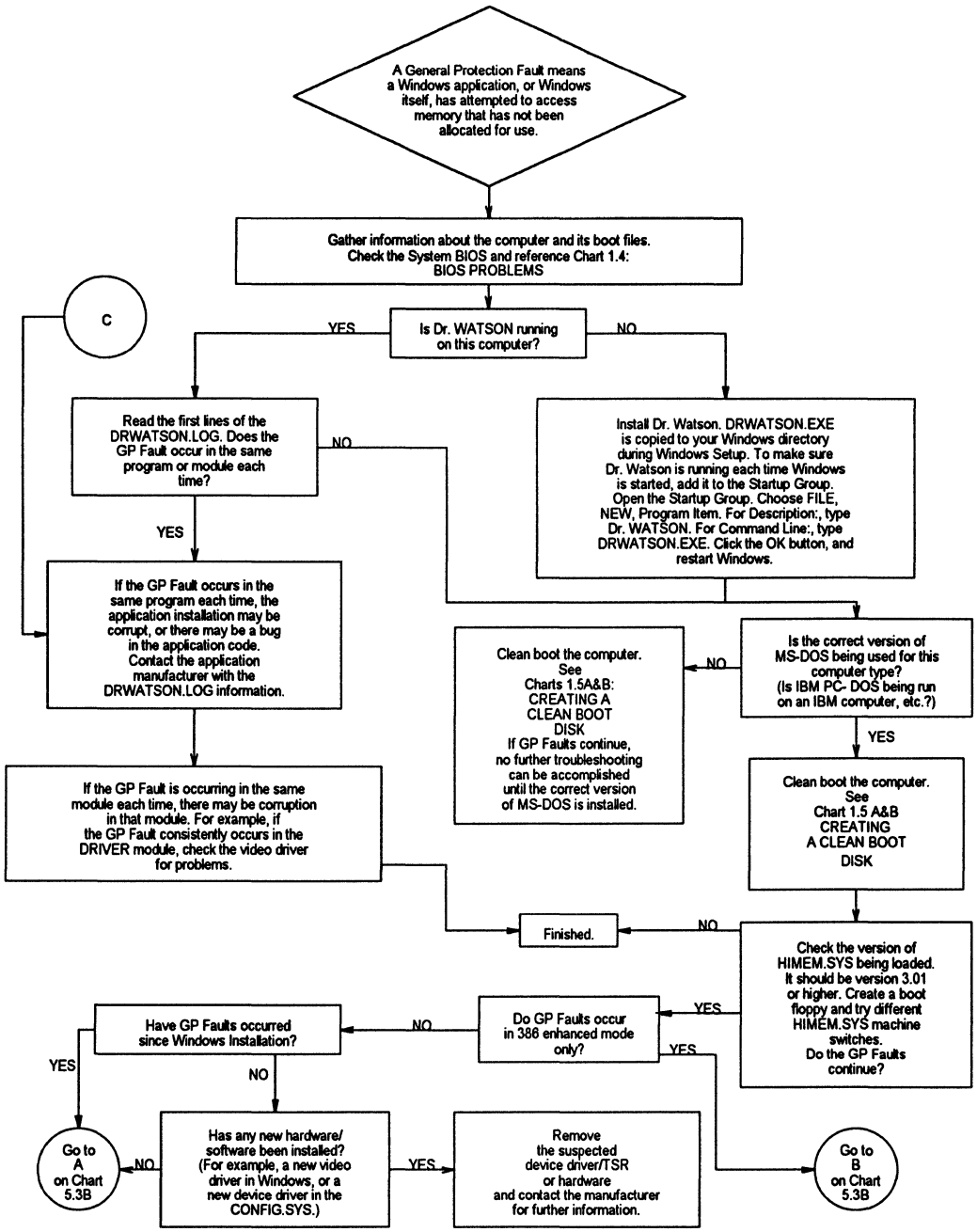
Chart 5.2 Cannot Run 386 Enhanced Mode on Network Workstation



Windows Resource Kit Reference ***page***

"Memory, Windows, and the Network" in Chapter 12 370
"Troubleshooting Networks" in Chapter 13 430

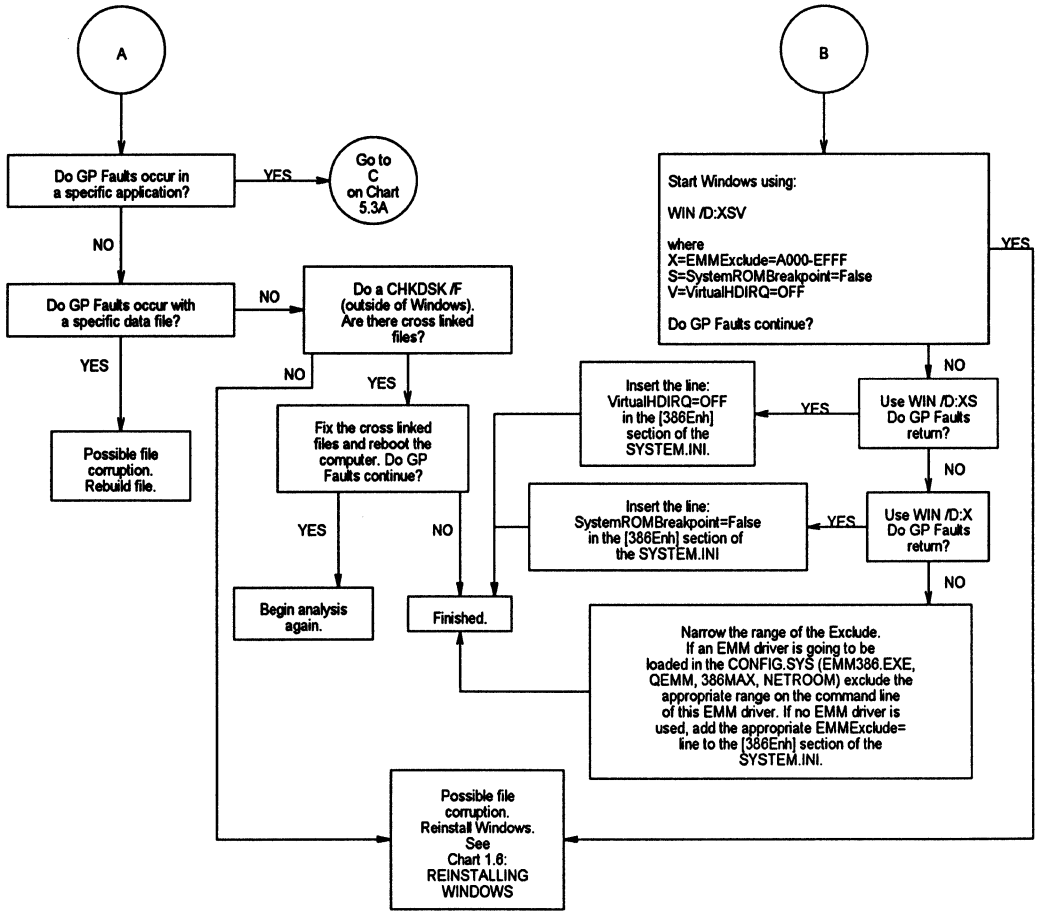
Chart 5.3A General Protection Faults



Windows Resource Kit Reference ***page***

"Troubleshooting GP Faults" in Chapter 13 431

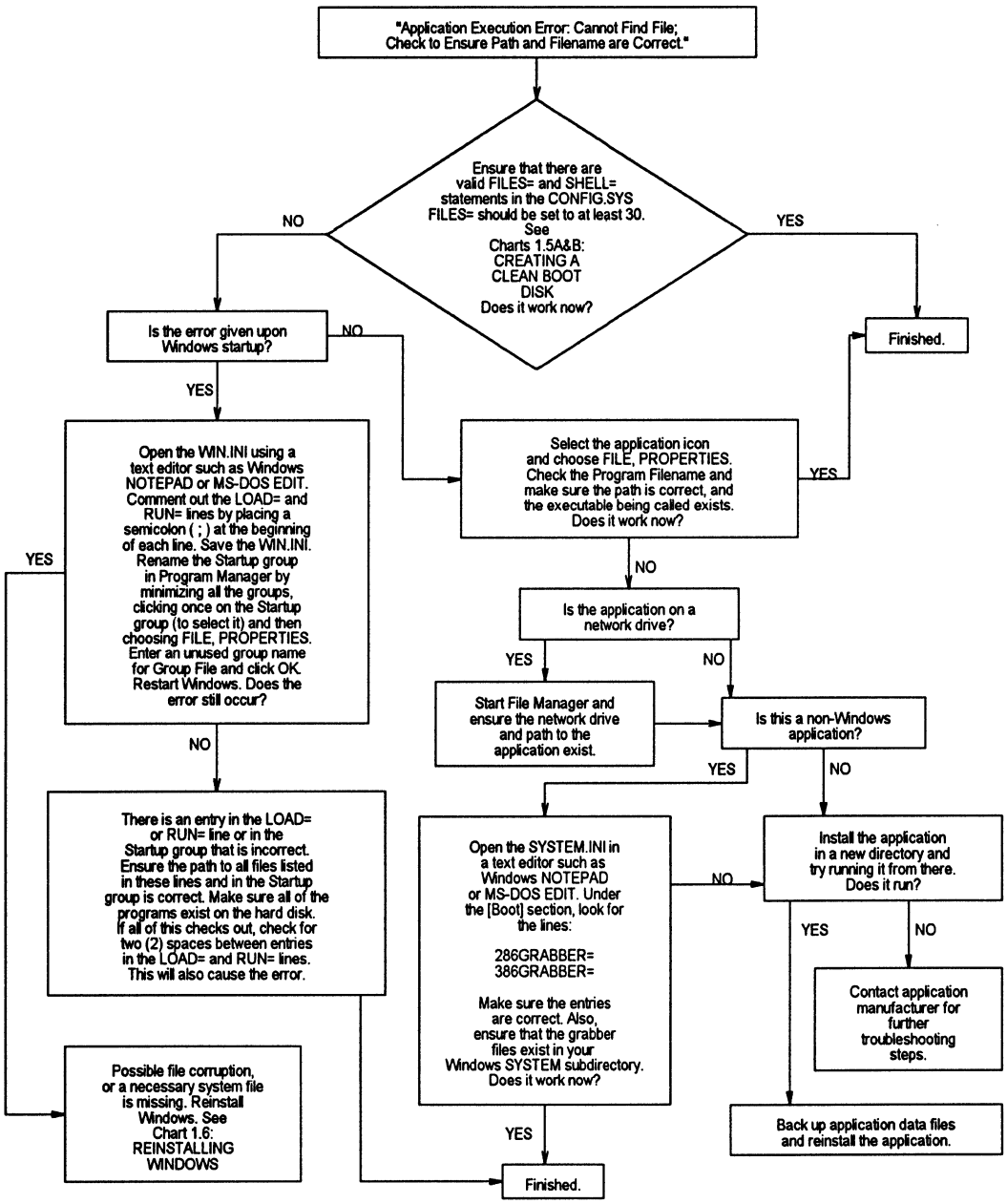
Chart 5.3B General Protection Faults



Windows Resource Kit Reference ***page***

"Troubleshooting GP Faults" in Chapter 13 431

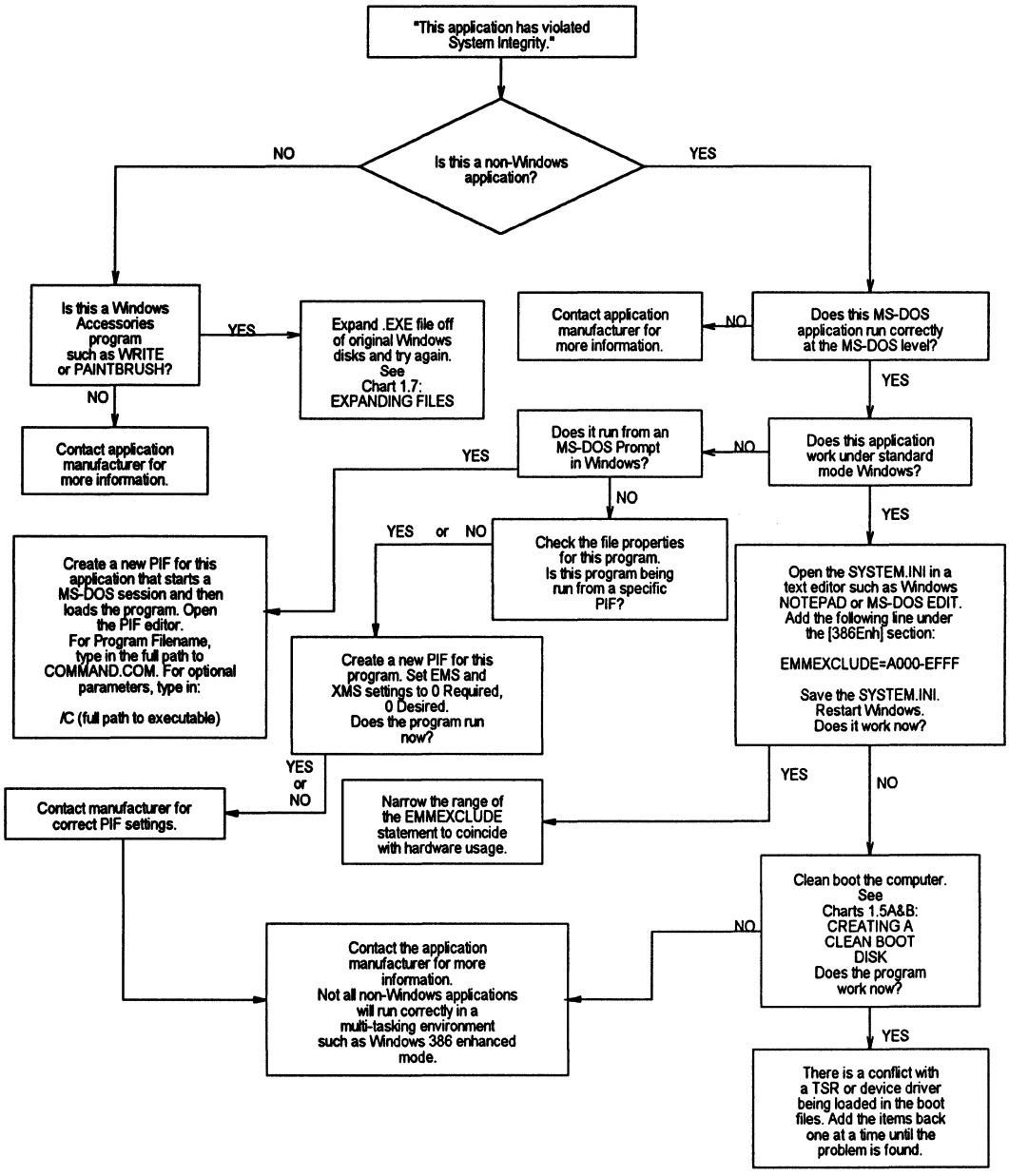
Chart 5.4 Application Execution Error



Windows Resource Kit Reference ***page***

"Tips for Running Non-Windows Applications" in Chapter 7 294
"Troubleshooting for Non-Windows Applications" in Chapter 13..... 423

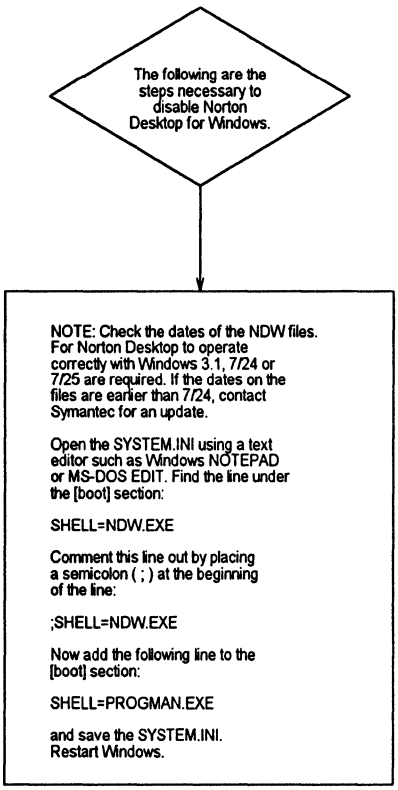
Chart 5.5 System Integrity Violation in 386 Enhanced Mode



Windows Resource Kit Reference ***page***

"Tips for Running Non-Windows Applications" in Chapter 7..... 294
"Troubleshooting for Non-Windows Applications" in Chapter 13 423

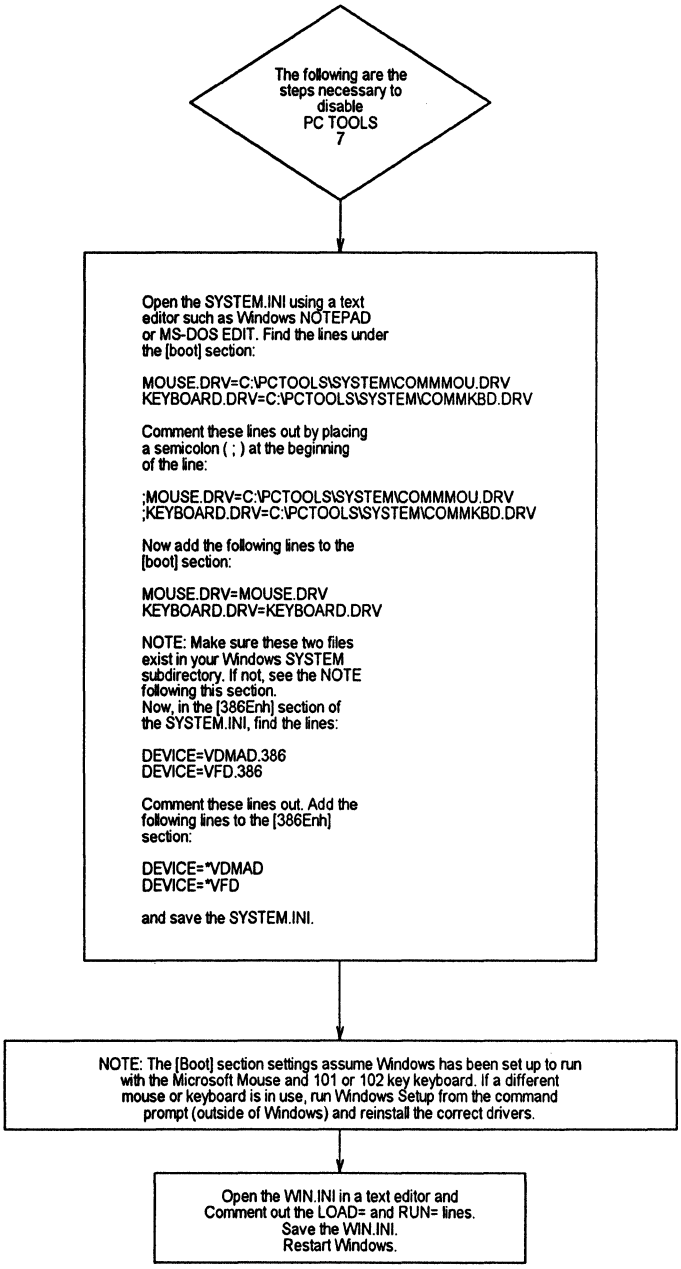
Chart 5.6 Disabling Norton Desktop for Windows



Windows Resource Kit Reference ***page***

"Troubleshooting TSRs During Setup" in Chapter 13 400
"Troubleshooting TSR Compatibility Problems" in Chapter 13 404

Chart 5.7 Disabling PC-Tools Version 7



Windows Resource Kit Reference ***page***

"Troubleshooting TSRs During Setup" in Chapter 13 400
"Troubleshooting TSR Compatibility Problems" in Chapter 13 404

Chart 5.8 Out-of-Memory Errors in Windows

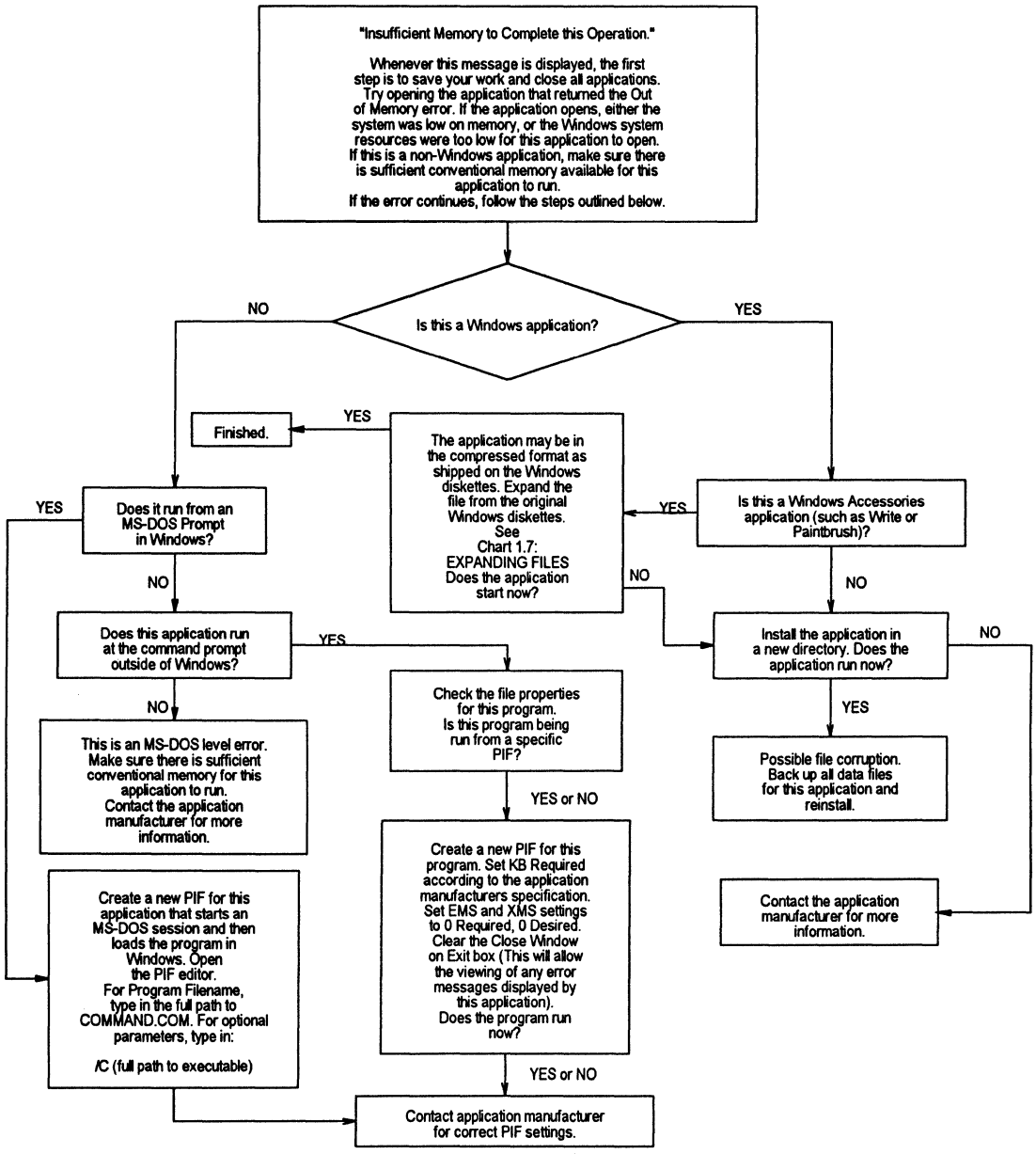
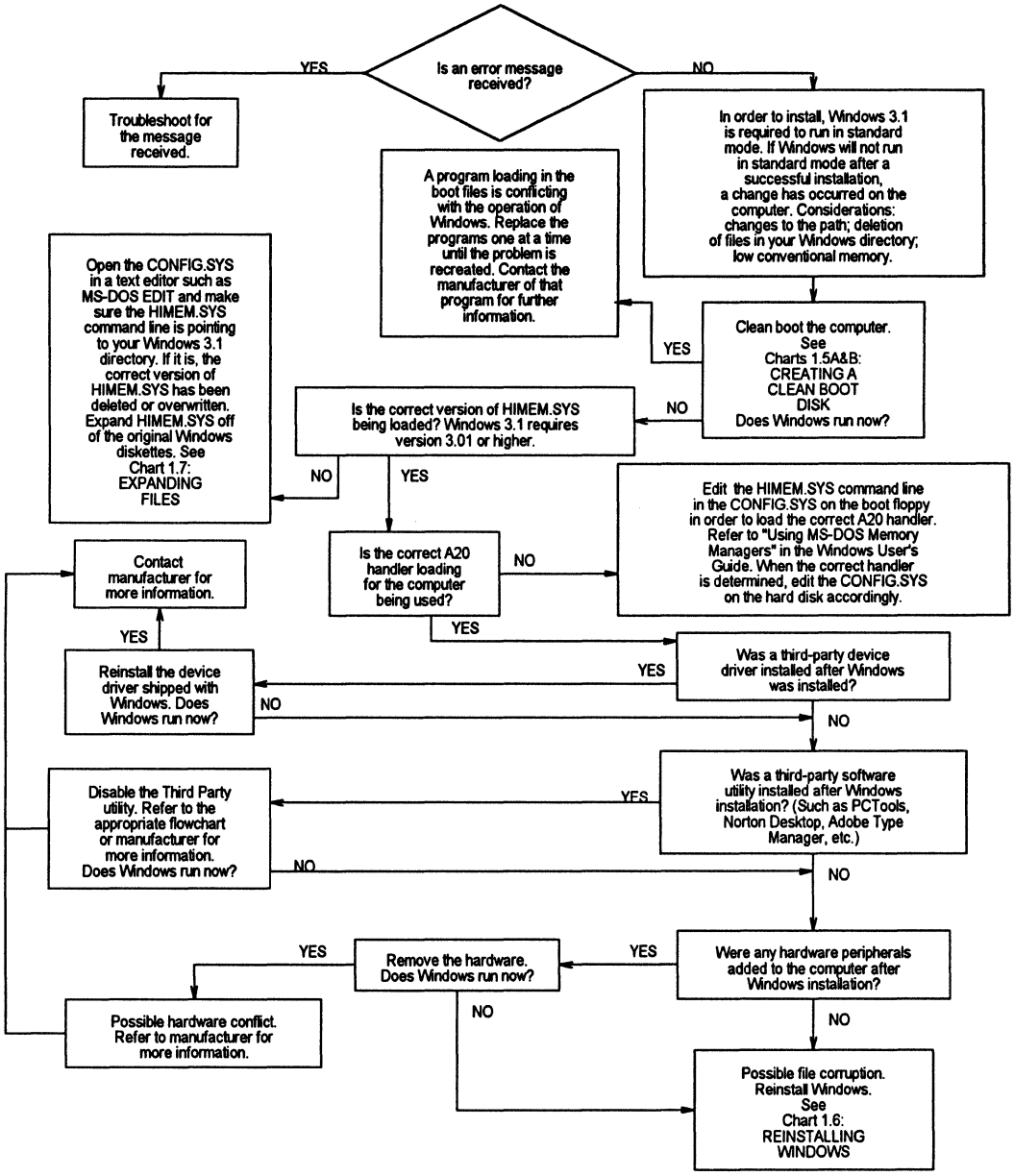


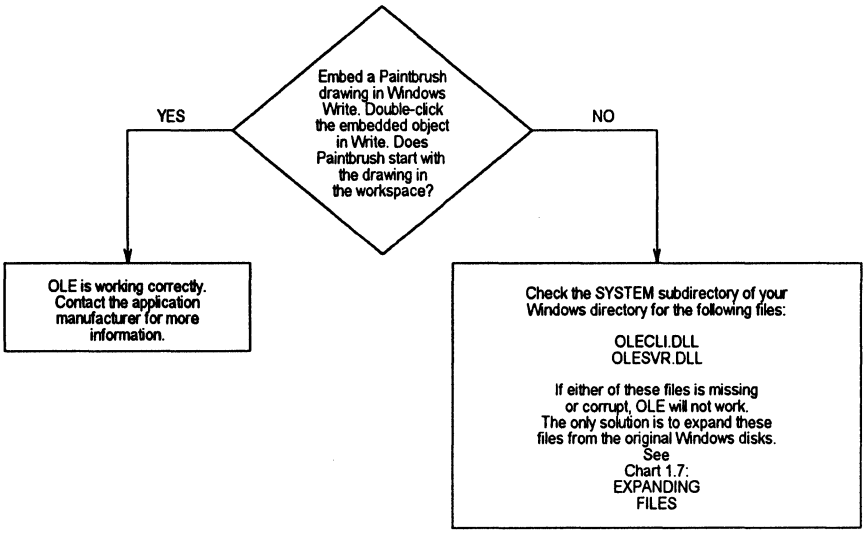
Chart 5.9 Cannot Run Windows in Standard Mode



Windows Resource Kit Reference	page

"Troubleshooting Windows Running in Standard Mode" in Chapter 13 418

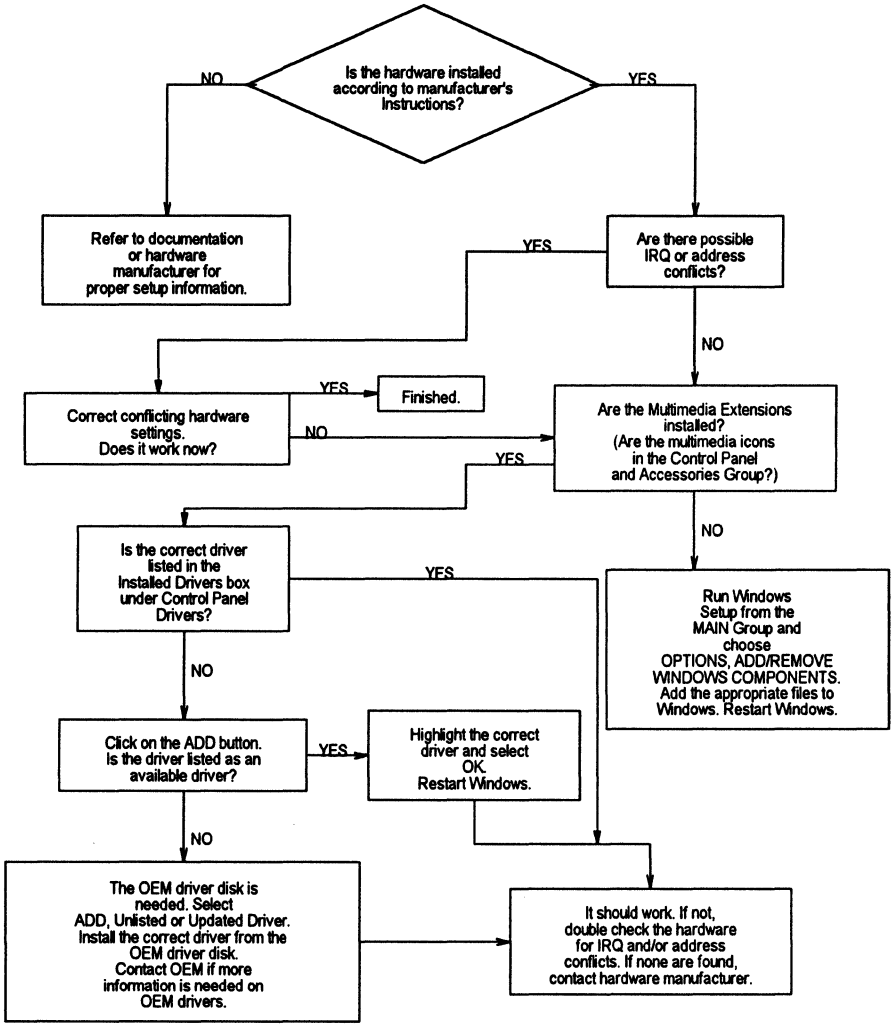
Chart 5.10 Problems with Object Linking and Embedding



Windows Resource Kit Reference ***page***

Chapter 11, "Compound Documents in Windows 3.1" 351

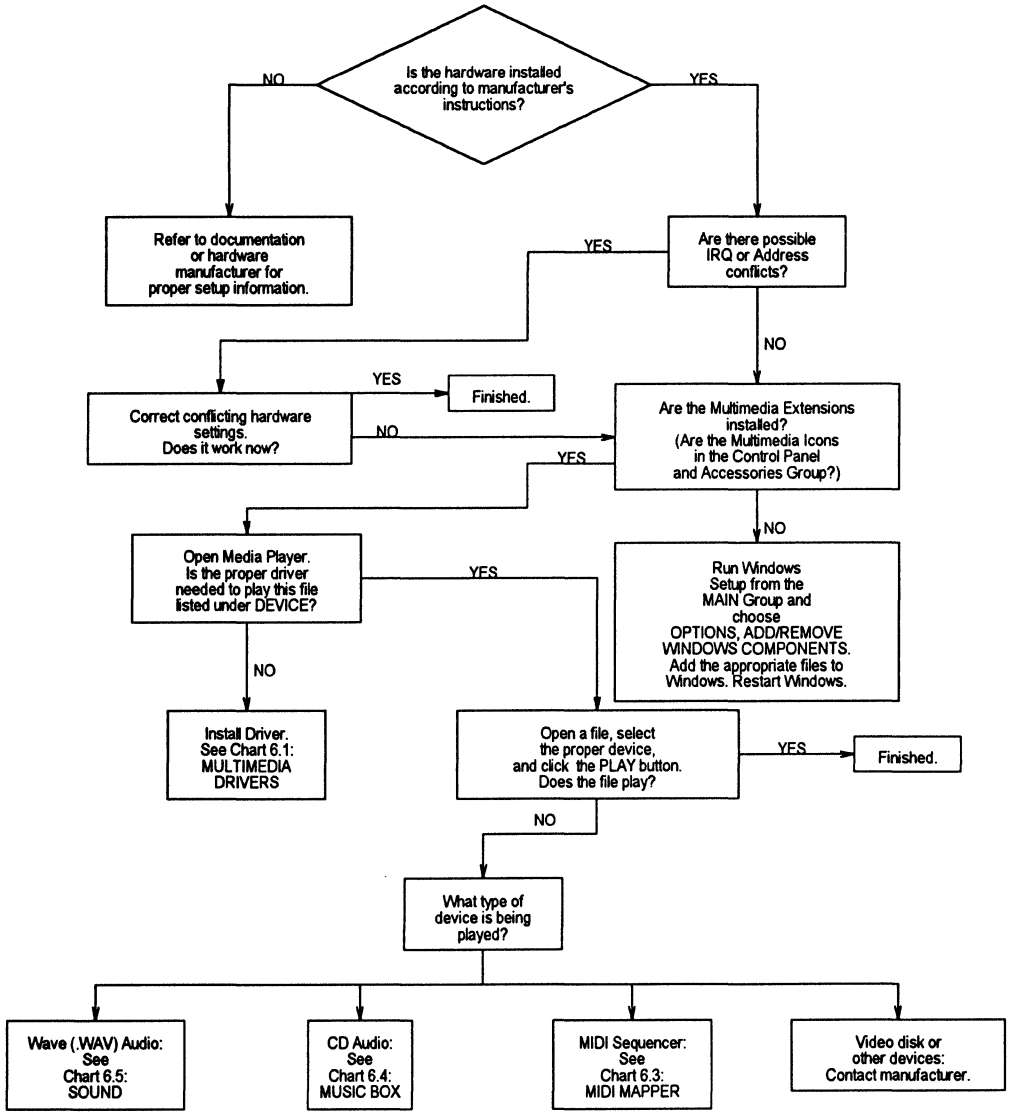
Chart 6.1 Multimedia Drivers



Windows Resource Kit Reference ***page***

"Troubleshooting Multimedia" in Chapter 13 434

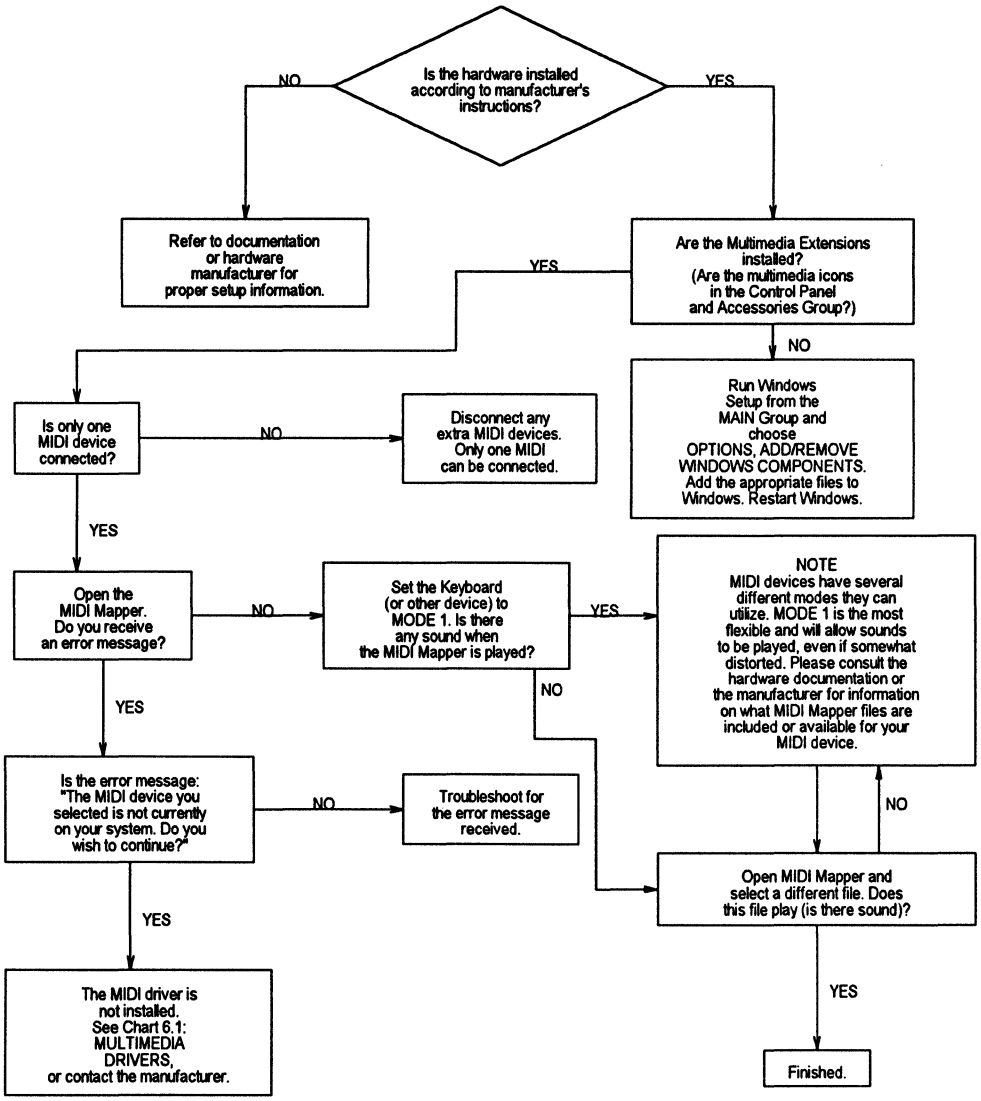
Chart 6.2 Media Player



Windows Resource Kit Reference ***page***

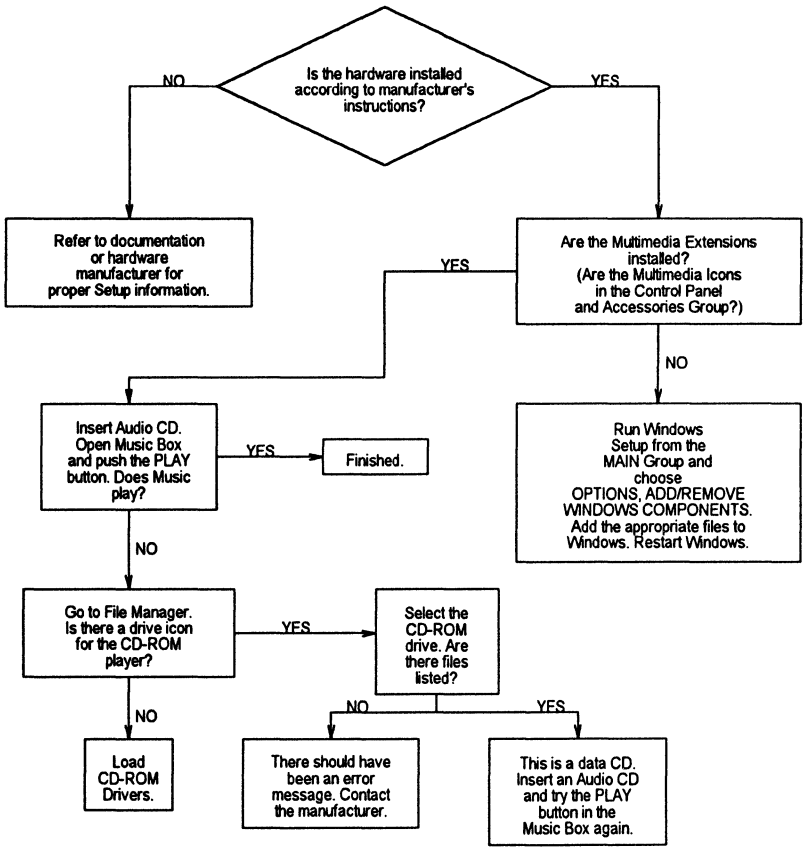
"Troubleshooting Multimedia" in Chapter 13 434

Chart 6.3 MIDI Mapper



Windows Resource Kit Reference	page
"Troubleshooting Multimedia" in Chapter 13	434

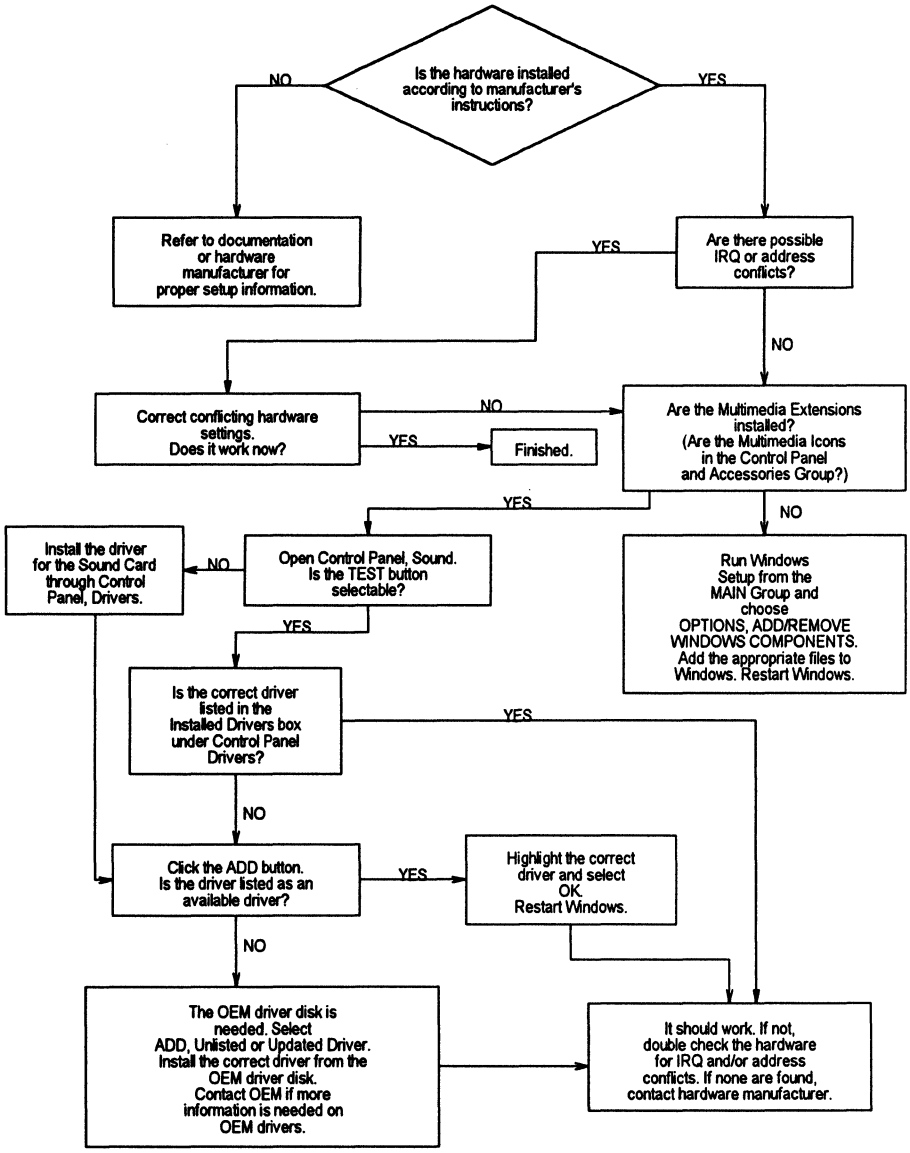
Chart 6.4 Music Box



Windows Resource Kit Reference ***page***

"Troubleshooting Multimedia" in Chapter 13 434

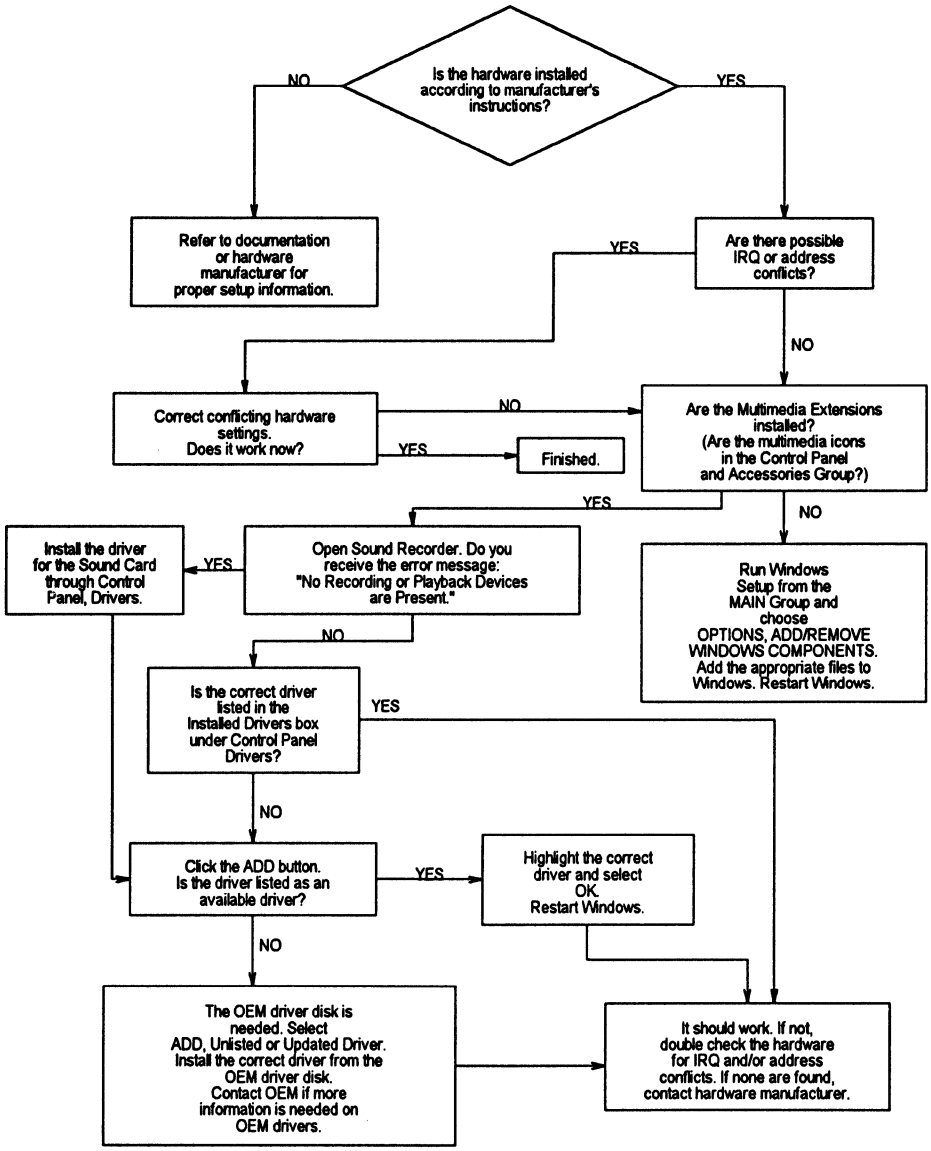
Chart 6.5 Sound



Windows Resource Kit Reference ***page***

"Troubleshooting Multimedia" in Chapter 13 434

Chart 6.6 Sound Recorder



Windows Resource Kit Reference ***page***

"Troubleshooting Multimedia" in Chapter 13 434

More About MS-DOS, Microsoft Windows, and Personal Computers

Print References:

- *Windows User's Guide* and the *Getting Started* booklet in your Windows 3.1 package.
- *Microsoft MS-DOS 5.0 User's Guide and Reference* and the *Getting Started* booklet in your MS-DOS 5.0 package.
- *Computer Dictionary*. Microsoft Press, 1991.
- Gookin, Dan. *The Microsoft Guide to Managing Memory with DOS 5.* Microsoft Press, 1991.

Online References:

- Windows 3.1 online help contains instructions and definitions of terms related to the screen elements in Microsoft Windows. To get online help, press F1 or choose a command from the Help menu in Program Manager or File Manager.

Support Services and Training:

- See the information in your Windows package about how to contact Microsoft Product Support Services and how to use the Microsoft Forums on CompuServe.
- See Appendix A, "Windows Resource Directory," in the *Windows Resource Kit* for information about Microsoft University, the Microsoft Authorized Training Centers, and other sources of information and training.

Also, in Appendix A, "Windows Resource Directory," there is an extensive list of reference materials.

Installation and Setup



Part
1 Installation and Setup

Chapter 1 **Windows 3.1 Installation** 57

About Windows Setup ... 58
Windows Setup: A Technical Discussion ... 62
Setting Up Windows for a Network ... 71
Using Automated Setup: Setup /h ... 75
Maintaining Windows with Setup ... 82

Chapter 2 **The Windows Setup Information Files** 85

About the Setup Information Files ... 86
SETUP.INF: The Setup Information File ... 88
APPS.INF: The Application Installation Information File ... 121
CONTROL.INF: The Control Panel Information File ... 126
Modifying .INF Files for Custom Installations ... 128

Chapter 3 **The Windows Files** 135

About the Windows Files ... 136
WIN.COM ... 136
The Core Files ... 137
Drivers, Fonts, and International Support Files ... 137
MS-DOS Support Components of Windows ... 148
Windows Applications, Setup, and Other Files ... 151
Files You Can Delete ... 155

Chapter 4 **The Windows Initialization Files** 157

About the Initialization Files ... 158
WIN.INI: The Windows Initialization File ... 163
SYSTEM.INI: The System Initialization File ... 183
Other Windows Initialization Files ... 219

Chapter

1 Windows 3.1 Installation

This chapter contains an overview of installation for Microsoft Windows 3.1, plus a detailed technical discussion of the Windows Setup program. This chapter concludes with details about setting up Windows 3.1 over a network and creating a custom installation routine for automated Setup.

Related information

- *Windows User's Guide:* Part 1, "Windows Fundamentals," and the *Getting Started* booklet
- *Windows Resource Kit:* Chapter 2, "Windows Setup Information Files"; Chapter 4, "The Windows Initialization Files"; Chapter 13, "Troubleshooting Windows 3.1"

Contents of this chapter

About Windows Setup 58
Running Setup 59
Choosing Setup Options 60
Choosing Express Setup vs. Custom Setup 62
Windows Setup: A Technical Discussion 62
About the Windows Operating Modes 63
MS-DOS Mode Setup 63
Windows Mode Setup 68
Setting Up Windows for a Network 71
General Guidelines 72
Placing Windows Files on a Network Server: Setup /a 73
Setting Up a Shared Copy of Windows: Setup /n 74
Using Automated Setup: Setup /h 75
Creating the System Settings File 76
Sample System Settings File 81
Maintaining Windows with Setup 82

About Windows Setup

The Windows Setup program is on Disk #1 of the Microsoft Windows disk set. The Setup program guides you through a straightforward series of steps, in two basic stages:

1. In the first stage of installation, called MS-DOS Mode Setup, Setup evaluates the computer system and asks you to verify this information. Then Setup copies the basic Windows files onto the system's hard disk.
2. In the second stage of installation, called Windows Mode Setup, Setup loads Windows and installs the remaining files, then prompts you to make a series of choices about installing printer drivers and other applications.

New in Setup for Windows 3.1



Look for these new features in Windows Setup:

- Setup warns you about the presence of any incompatible drivers known to be incompatible with either Setup or Windows. Setup looks for earlier versions of Windows 3.x and various drivers to update driver files. For details about related changes in SYSTEM.INI and WIN.INI, see Chapter 4, "The Windows Initialization Files."
- Whether you choose Express Setup or Custom Setup, Setup detects the machine for installation, so you seldom have to choose from the list of specific machines and devices. For details about the machines and drivers that Setup detects, see the entries for the **[display]**, **[keyboard.types]**, **[pointing.device]**, and **[machine]** sections in SETUP.INF, as described in Chapter 2, "The Windows Setup Information Files."
- Setup automatically creates a Startup Group in Program Manager, containing all the applications you want to start each time Windows runs.
- You can automate Windows installation by creating a script. For details, see "Using Automated Setup" later in this chapter.

Note In this manual, *WINDOWS directory* refers to the directory where you choose to install Windows, and *SYSTEM subdirectory* refers to the subdirectory where the Windows system files are stored.

Running Setup



You can run Windows Setup in two ways, for two different purposes:

- Type **setup** at the command prompt to install and configure Microsoft Windows from floppy disks or from a network drive. Do this to upgrade your Windows installation or to install Windows for the first time on your system.
- Run Setup from Program Manager after Windows is installed. Do this to change your system configuration or to add or delete files.

Follow these basic steps to install Windows 3.1 from the installation disks.

To install Windows 3.1 from the disk drive:

1. Insert the Windows 3.1 disk #1 into a floppy drive, and change to the drive letter of the floppy drive.

   For example, to switch to the A drive, at the command prompt type **A:** and press ENTER.

2. At the command prompt, type **setup** and press ENTER.

   You can include any switches with the **setup** command, as described in "Choosing Setup Options" later in this chapter.

3. Follow the instructions on your screen.

   If you have questions about any procedures or options, press F1 to get online help.

Important Do not use the MS-DOS **copy** command to copy the Windows files from the Setup disks. These files are compressed and cannot be used unless you run the Setup program. Setup renames the compressed files during installation, so if you have files on your system that end with an underscore (_), that file is compressed and has not been installed properly.

For details about installing Windows over a network, see "Setting Up Windows for a Network" later in this chapter.

Choosing Setup Options

You can include various switches when you type **setup** at the command prompt, as described in the following table.

Switch	Purpose
/i	Ignores automatic hardware detection. The user will have to check settings and possibly make corrections on the System Information screen during Setup.
/n	Sets up a shared copy of Windows from a network server. For details, see "Setting Up Windows for a Network" later in this chapter.
/a	Begins Administrative Setup, by expanding and copying all files from the Windows installation disks onto a network server, and also marking the files as read-only. For details, see "Setting Up Windows for a Network" later in this chapter.
/b	Sets up Windows with monochrome display attributes.
/t	Searches the drive for incompatible software that should not run at the same time as Setup or Windows 3.1. (For maintenance only.)
/h:*filename*	Runs Batch Mode Setup to install Windows with little or no user interaction. The *filename* is the name of the system settings file that contains information about the user's configuration. Include the path if *filename* is not in the directory that contains the Windows Setup files. For details, see "Using Automated Setup" later in this chapter.
/o:*filename*	Specifies the SETUP.INF file, including the path if necessary.
/s:*filename*	Specifies the SETUP.INF file, including a path for the Windows installation disks.

Tip To see a list of the switches available for Windows Setup, at the command prompt type **setup /?** and press ENTER.

You can customize the installation procedure for Windows using the command-line switches. You can also customize the Windows information (.INF) files and the initialization (.INI) files to install a custom version of Windows at your site.

The following decision tree will help you decide how you might want to customize Windows for installation at your site.

Windows Custom Installation Decision Tree for a Single System

- **You want to customize how Windows is installed:**
  → Choose Custom Setup at the first Setup prompt.
- **You want to customize how Windows works after installation:**
  → Use Control Panel, Program Manager, and Windows Setup, as described in Chapter 15 in the *Windows User's Guide*.

Windows Custom Installation Decision Tree for Multiple Systems

- You want to set up Windows on a network:
  → Use **setup /a** to copy files to the network server, then use **setup /n** to install a shared copy of Windows on the workstations. See "Setting Up Windows for a Network" later in this chapter.
- **You want to automate installation for multiple users:**
  → Create custom system settings files and use **setup /h**. See "Using Automated Setup" later in this chapter.
- **You want custom choices available in the System Information screen:**
  → Edit device entries in SETUP.INF for hardware, and edit printer entries in CONTROL.INF. See "Removing Hardware Devices from Lists in Setup" in Chapter 2.
- **You want to customize the applications installed with Windows:**
  → Install custom applications, as described in "Modifying .INF Files to Install Custom Applications" in Chapter 2.

  → Customize the Windows applications installed, as described in "Modifying .INF Files to Install Custom Windows" in Chapter 2.

  → Install custom PIFs during installation, as described in "Adding Custom PIF Information in APPS.INF" in Chapter 2.
- **You want to customize Program Manager for multiple systems:**
  → Restrict user capabilities or set up shared Program Manager groups as described in "Other Windows Initialization Files" in Chapter 4
- **You want a custom Windows startup environment for multiple systems:**
  → Edit WIN.SRC and SYSTEM.SRC before installing Windows, as described in "Editing the .INI Source Files" in Chapter 4.

Choosing Express Setup vs. Custom Setup



At the beginning of Setup, you are prompted to choose Express Setup or Custom Setup. Choose Express Setup for most standard installations. You only have to provide information about the printers and port connections for your system.

Choose Custom Setup for precise control over how Windows is set up. Setup asks you to verify information detected about the hardware and software in your system, such as:

- The directory where you want the Windows files installed.
- The computer system hardware, including the monitor, mouse, and keyboard. Windows Setup detects most kinds of hardware, but with Custom Setup, you can make more detailed choices about your system configuration.
- The preferred language (you can run Windows in several languages, including French, German, and Spanish).
- The network, if any.
- The printers and printer ports for the computer system.
- The other applications you want to run with Windows.
- The Windows applications and supporting files that you want to install.
- The specific Windows components you want to install.

Windows Setup: A Technical Discussion

This section is a step-by-step examination of how the Windows Setup program installs Windows. This information will help you to diagnose and solve problems with Windows installation, and will also help you to make custom modifications. For additional up-to-date information, be sure to review the README file in the WINDOWS directory.

During the installation procedure (if you choose Custom Setup), Setup asks you to confirm that the hardware settings are correct or modify the settings if necessary. At a more technical level, Setup continually checks the system along the way to make sure it can complete the installation procedure, and performs several automatic steps to upgrade from a previous version of Windows that may be installed on your system. Setup also installs printer drivers, applications in Program Manager groups, and third-party drivers, and it performs various maintenance functions.

The Setup program has two parts: MS-DOS Mode Setup and Windows Mode Setup.

About the Windows Operating Modes

Flowchart 1.1 System Requirements

Windows can run in two different operating modes: standard mode and 386 enhanced mode. For both of these operating modes, the processor runs in protected mode. When you type **win** at the command prompt, Windows automatically runs in the appropriate mode, depending on the computerhardware and the amount of memory the system has:

- **Standard mode** requires a PC with an 80286 processor (or higher) and 256K of free conventional memory and 192K of free extended memory. This mode provides access to extended memory using application swap files. Windows does not use virtual memory. Non-Windows applications run in full screen.

  If you have an 80386 or higher processor, you can force Windows to run in standard mode by typing **win /s** at the command prompt.

- **386 enhanced mode** requires a PC with an 80386 processor (or higher) and 2 megabytes or more of memory (256K of free conventional memory and 1024K extended memory).

  Windows 386 enhanced mode provides access to the virtual memory capability of the 80386 processor, so that Windows applications can use more memory than is physically available. Windows 386 enhanced mode also allows multitasking of non-Windows applications and allows you to have more control over those applications. For example, you can run a non-Windows application in a window.

For technical details about the Windows operating modes, see Chapter 5, "Windows 3.1 and Memory Management."

MS-DOS Mode Setup

In the MS-DOS Mode part of Setup, the Setup program:

- Confirms the directory where the Windows files will be stored.
- Identifies the type of hardware installed in the system.
- Installs the Windows files required for Windows Mode Setup.

Note Throughout the Setup procedure, Setup checks that the files it copies are newer than any previous versions on the system. Setup will not copy the version of a file on the installation disks over a newer version of a file that's already on your system.

Getting Ready for MS-DOS Mode Installation

MS-DOS Mode Setup begins by drawing the Setup welcome screen, then saves for later use any command-line parameters (such as **/n**), the type of processor, and the free memory available. Setup also checks to see if it can read the SETUP.INF file and that the file is not corrupted, and it checks whether the MS-DOS version is greater than or equal to MS-DOS 3.10. If either the MS-DOS version check or the SETUP.INF check fails, Setup displays a message box explaining the failure, and then exits to MS-DOS.

If the MS-DOS version and the SETUP.INF are okay, Setup checks the path for a previous version of Windows. If it cannot find Windows on the path, Setup searches every local fixed disk for Windows. If Setup finds that the same version of Windows 3.1 has already been installed on the system, Setup switches to Maintenance Mode and performs updates of the system drivers.

If the same version of Windows is not found, Setup checks that at least 376K of conventional memory is available. If the required memory is not free, Setup fails. If the memory is free, then Setup begins installation.

Setting the Windows Directory and Updating the Version

For Express Setup, if no previous version of Windows is found on the local drives, Setup creates the default directory listed in SETUP.INF (the **defdir=** entry in **[data]**) and begins installation.

For Custom Setup, Setup displays a prompt showing the suggested directory to install Windows (usually either C:\WINDOWS or the directory containing the previous version of Windows). If you modify the directory for installing Windows, Setup checks to make sure that the path is valid and that you're not installing Windows in the same directory as the source files. If you try to do this, a warning prompts you to modify the directory path.

If Windows 3.0 is in the destination directory, Setup asks if you want to upgrade the earlier Windows installation or enter a different destination directory.

If you answer Yes to upgrade an earlier version of Windows, Setup performs this version update:

- The files listed in the **[win.copy]**, **[win.copy.net]**, or **[win.copy.win386]** sections and the **[update.files]** sections of SETUP.INF are copied to the appropriate directories.
- Files listed in the **[delFiles]** section of SETUP.INF are deleted.
- Various entries in WIN.INI and SYSTEM.INI are replaced, based on entries in the **[ini.upd.31]** section of SETUP.INF.
- Program Manager groups are redefined based on entries in the **[new.groups]** section of SETUP.INF.
- The old WINVER.EXE file is deleted and replaced with the current version.
- The closing screen is displayed, asking you to reboot Windows or return to MS-DOS. (If you are installing from a network, Setup will only return to MS-DOS, not reboot.)

If you specify another directory, Setup continues installing Windows 3.1 as though it were a first-time installation.

Identifying Hardware

If you run Setup with the **/i** switch (**setup /i**), then Setup displays the System Configuration screen so that you can choose the machine type, display, mouse driver, and network used in your system. For details about making choices in this screen, press F1 during Setup, or see the *Getting Started* booklet.

If Setup is not performing an update and if you did not use the **setup /i** switch, then Setup performs Auto-detection to identify the specific type of machine, display, keyboard, mouse, and network devices. After the detection routines, Setup checks the type of display against the type of processor to make sure the display type is supported for running Windows 386 enhanced mode. (For example, EGABW and EGAMONO are not supported in 386 enhanced mode.)

After the detection routines are completed, Custom Setup displays the configuration information so you can modify the settings if necessary. For example, you might want to specify a display type that requires an OEM-supplied disk with the necessary drivers.

Figure 1.1

System Configuration screen for Windows Setup

```
Windows Setup
=============
If your computer or network appears on the Hardware Compatibility List with
an asterisk next to it, press F1 before continuing.

      System Information
      Computer:         MS-DOS System
      Display:          Super VGA (800x600,16 colors)
      Mouse:            Microsoft, or IBM PS/2
      Keyboard:         Enhanced 101 or 102 key US and Non US keyboards
      Keyboard Layout:  US
      Language:         English: (American)
      Codepage:         English (437)
      Network:          No Network Installed

Complete Changes:  Accept the configuration shown above.

To change a system setting, press the UP or DOWN ARROW key to move the
highlight to the setting you want to change. Then press ENTER to see
alternatives for that item. When you have finished changing your settings,
select the "Complete Changes" option to quit Setup.

ENTER=Continue     F1=Help     F3=Exit
```

Setup also searches for TSRs and drivers that are known to interfere with installation. Where possible, Setup automatically removes these drivers. You may see a warning at this point that you must uninstall incompatible drivers or TSRs for Setup to proceed. Setup also checks for disk compression software such as Stacker. For more information, see "Troubleshooting TSRs During Setup" in Chapter 13, "Troubleshooting Windows 3.1."

When you confirm the system configuration settings, Setup copies the appropriate files required to start Windows in standard mode and prepares a minimum SYSTEM.INI file to start Windows Mode Setup. If you are installing Windows on a local drive, Setup copies SETUP.EXE to the WINDOWS directory and SETUP.INF to the Windows SYSTEM subdirectory. Next, it copies files listed in the **[winexec]**, **[windows]**, and **[windows.system]** sections of SETUP.INF plus the appropriate Windows logo code and the logo data files. Then Setup renames the SYSTEM.SRC file as SYSTEM.INI.

Modifying SYSTEM.INI to Support Your Hardware

To prepare for Windows Mode Setup, two sections in the SYSTEM.INI file are modified to support the hardware configuration:

- The **[boot]** section of SYSTEM.INI is modified to update the following entries: **system.drv=**, **display.drv=**, **keyboard.drv=**, **mouse.drv=**, **fonts.fon=**, **oemfonts.fon=**, **sound.drv=**, **comm.drv=**, **fixedfon.fon=**, **language.dll=**, and **network.drv=**.

- The **[keyboard]** section of SYSTEM.INI is modified, updating the following entries: **keyboard.dll=**, **oemansi.bin=**, **type=**, and **subtype=**. For most U.S. installations, the **type=** entry is the only entry that is modified.

The **shell=** entry in the **[boot]** section of SYSTEM.INI becomes **shell=setup** (which is specified in the **[shell]** section of SETUP.INF). The **[boot.description]** section of SYSTEM.INI is filled with the text strings used by Setup to show the current configuration in a format that can be understood by nontechnical Windows users.

The files listed in the **[boot]** section of SYSTEM.INI are now copied to the Windows SYSTEM subdirectory. The names of the actual drivers will depend on the machine configuration. Usually, the following files are copied: SYSTEM.DRV, DISPLAY.DRV, KEYBOARD.DRV, MOUSE.DRV, FONTS.FON, OEMFONTS.FON, SOUND.DRV, COMM.DRV, and FIXEDFON.FON.

After these system files are copied, the KEYBOARD.DLL file and the OEMANSI.BIN file (if necessary) are copied to the Windows SYSTEM subdirectory.

MS-DOS Mode Setup next modifies SYSTEM.INI for settings to complete installation. The **286grabber=** entry in the **[boot]** section is modified first, and if the system configuration will support 386 enhanced mode, the **386grabber=** entry is modified. For 80386 and higher processors, entries are placed in the **[386enh]** section of SYSTEM.INI to support the virtual display device, EBIOS, and any machine-specific settings such as **emmexclude=** values.

Last Steps for MS-DOS Mode Setup

Before Windows runs, Setup builds WIN.COM in the WINDOWS directory by joining the WIN.CNF file with the correct logo code and logo data files from the Windows installation disks.

The last step before starting the Windows Mode part of Setup is to build the command line to pass to Setup. (The command-line parameters are switches following an executable name that tell a program to execute in a specific way.) First, the name of the application to execute (**win.com**) is read from the **startup=** entry in the **[data]** section of SETUP.INF.

Next, Setup reads the default command line from the **execcmd=** entry in the **[winexec]** section of SETUP.INF, which specifies the Windows kernel to use for Windows Mode Setup.

Finally, Setup calls the MS-DOS **exec** command to start WIN.COM using the Setup command-line parameters. At the completion of MS-DOS Mode Setup, WIN.COM loads, first displaying the logo screen and determining whether Windows can run in standard mode to complete Setup. If HIMEM.SYS is present and there is at least 256K of conventional memory and 192K of extended memory, Windows starts in standard mode. If HIMEM.SYS is not present, then XMSMMGR.EXE is also executed, to serve as the extended memory manager for Setup.

Flowchart 1.2 Windows Hangs During Setup

If MS-DOS Mode Setup fails, or if Windows Mode Setup hangs, Setup probably failed to detect the correct hardware on your system (or you provided Windows with incorrect hardware settings). For more information about Setup failure, see Chapter 13, "Troubleshooting Windows 3.1."

Windows Mode Setup

When MS-DOS Mode Setup starts Windows, all of the essential Windows files are loaded, including the core executable files (KRNL286.EXE or KRNL386.EXE, User, and GDI), dynamic-link libraries, the drivers, and the fonts. The Windows background color is drawn on the screen, followed by the hourglass mouse cursor (if a mouse or pointing device is present), and finally by the Windows Setup screen. During Windows Mode Setup, Windows automatically runs in standard mode.

Getting Ready for Windows Mode Installation

While Windows is starting for the first time, Setup is loading SETUP.INF into memory and searching for the **[blowaway]** section. It trims down the file at that section, discarding unnecessary sections to minimize the memory overhead for the file. For more information on **[blowaway]**, see Chapter 2, "The Windows Setup Information Files."

Setup then checks the **shell=** entry in the **[boot]** section of SYSTEM.INI to make sure that it is the Windows boot shell. If Setup is not the boot shell, then Setup performs a series of maintenance tasks and prompts you to change the system settings. If Setup is the boot shell, then it gets the path of the WINDOWS directory, reads the command-line parameters from MS-DOS Mode Setup, and sets flags for later use.

After the Windows Setup screen is drawn and if your system has a mouse, you can choose to run a tutorial to practice using a mouse in Windows and to identify key elements of a window.

Next, Setup prompts you for a user name and company name, which is the identifying name that will appear with the registration serial number in the About Program Manager dialog box. If you have previously installed Windows using the same disks, a reminder appears instead of the registration dialog box.

Completing Custom Application Installation

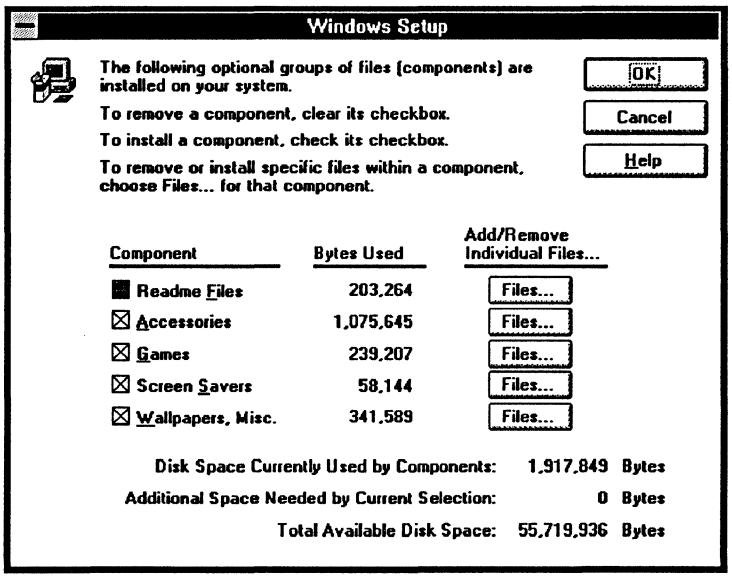


During Custom Setup, the Add/Remove Files dialog box is displayed at this point, asking if Setup should install printers and optional applications, and if you want to create a swap file. Setup saves the response to these options for later use. In Windows 3.1, the dialog box shows the total amount of disk space required for each option and allows you to further customize each choice.

Figure 1.2

Add/Remove dialog box for Windows Setup

This dialog box does not appear if you use ***setup /n*** *for installation*



Setup now uses information from the "copy files" sections of SETUP.INF to copy the Windows files and applications to the hard disk. These sections in SETUP.INF list the files to be copied, depending on the system:

- **[win.copy]**, if the system has an 80286 processor
- **[win.copy.net]**, if installing from a network drive
- **[win.copy.win386]**, if the system has an 80386 processor

Updating the System Files

Setup updates the system files at this point. Depending on settings in SETUP.INF, Setup will:

- Copy the virtual mouse drivers and update the virtual mouse driver entry in the **[386enh]** section of SYSTEM.INI.
- Copy the virtual network drivers and the related help files, plus any optional network files.
- Update system support sections of the WIN.INI and SYSTEM.INI files.
- Copy the virtual display device files, if required, along with the matching application fonts (such as EGA80WOA.FON), and modify SYSTEM.INI, if necessary. For more information see "Display Driver Sections" in Chapter 2, "The Windows Setup Information Files."



Setup also copies the TrueType fonts and other screen fonts, matching the resolution of the display driver with those of a given font set. The desired font sets are copied to the Windows SYSTEM subdirectory, and the WIN.INI font entries are updated, if necessary. For details, see "Fonts Files" in Chapter 3, "The Windows Files." When all other files are copied, Setup installs the specified applications and printers.

After completing these steps, Setup modifies the **shell=** entry in the **[boot]** section of SYSTEM.INI, changing it from **setup** to **progman** (the Windows Program Manager). Then Setup starts Program Manager and sends DDE messages to it to create groups and install group items as defined in the **[progman.groups]** and **[new.groups]** sections of SETUP.INF.

Checking for SMARTDrive and Incompatible Drivers



Setup searches CONFIG.SYS and removes any incompatible drivers and adds or updates entries for drivers such as HIMEM.SYS to load the desired drivers.

If SMARTDrive is not installed in the system, and the system has at least 512K of extended memory, Setup installs SMARTDRV.EXE in the AUTOEXEC.BAT file. If your system requires double buffering, then Setup also installs SMARTDRV.EXE in your CONFIG.SYS file. If RAMDRIVE.SYS is installed in CONFIG.SYS, Setup will update it; but Setup will not install RAMDrive if it isn't already installed.

Setup also checks the list of incompatible drivers in SETUP.INF. If any of these incompatible drivers are listed in CONFIG.SYS or if the MS-DOS **append**, **subst**, or **join** commands are in AUTOEXEC.BAT, Setup displays a message box explaining that incompatible drivers or commands have been found. Also, Setup checks the **files=** entry in CONFIG.SYS to make sure that the number of file handles is at least 30.

After these modifications, Custom Setup displays a dialog box, asking if the changes to CONFIG.SYS and AUTOEXEC.BAT should be made automatically or if you want to review and edit the file. Depending on your response, Setup takes the appropriate action and automatically saves the previous versions in backup files. For Express Setup, the changes are always made automatically.

Updating from Earlier Version of Windows

For updating from Windows 3.0, Setup makes changes to WIN.INI and SYSTEM.INI based on entries in **[ini.upd.31]** and other sections of SETUP.INF. For more information, see "Miscellaneous Sections" in Chapter 2, "The Windows Setup Information Files."

Completing Setup

To complete the installation of the Windows files, Setup renames WINVER to WINVER.EXE. Windows is now installed on your system.

Setup displays a final message, asking you to reboot Windows or to return to MS-DOS so that all changes can take effect. If you are installing Windows from a network, Setup always returns to MS-DOS rather than rebooting Windows.

Setting Up Windows for a Network

The rest of this chapter describes how to place Windows files on a network server, how to set up workstations to access a shared copy of Windows, and how to use the Automated Setup option to efficiently set up Windows on multiple computers.

If you want to set up Windows over a network, set up shared copies of Windows, or you are setting up multiple workstations, read these sections.

Note If you want to use the Windows Setup disks to set up Windows on a single workstation's hard disk, you do not need to read these sections.

General Guidelines

You have many options for setting up Windows over a network, depending on your environment:

- Do you want to set up a separate copy of Windows on each workstation? (This requires 8 to 10.5 MB on each hard disk.) Or do you want users to share most of the Windows files on a network server? (This requires 16 MB on shared network server disk and 300K on each workstation hard disk.)
- Do you want to run Setup for each workstation from the Windows setup disks or from a network server?
- How interactive do you want the setup process to be? Do you want the user to configure options, or do you want to automate the process so that all options are predetermined?
- Do you want to set up a customized version of Windows? You can first copy Windows files to the network, then make modifications described in Chapter 2, and finally set up Windows for users' workstations. For details, see "Modifying .INF Files for Custom Installations" in Chapter 2, "The Windows Setup Information Files."

Note You must have a license to use Microsoft Windows on more than one machine.

After deciding the answers to the questions above, you can decide which Setup options to use:



- **setup /a** (Windows Administrative Setup) copies the Windows files to a network server so that users can run the Setup program from the network. (This option replaces EXPALL.BAT that was used in Windows 3.0.)
- **setup /n** (Windows Network Setup) sets up a workstation to run a shared copy of Windows. Only a few Windows files are copied to the workstation's hard disk (or the user's personal network directory) so that the user can customize the Windows environment.



- **setup /h** (Windows Automated Setup) uses an automated Setup routine for easy installation. For more information, see "Using Automated Setup" later in this chapter.

The rest of this section describes Administrative Setup and Network Setup. The next section in this chapter describes Automated Setup.

Placing Windows Files on a Network Server: Setup /a



You can use **setup /a** (the Windows Administrative Setup option) to place Windows files on a network server. The administrative option does not set up a usable copy of Windows, but simply transfers the files from the Windows Setup disks to a network drive. The files are automatically expanded, renamed, and marked as read-only when they are copied to the network drive. The shared Windows files must be marked as read-only (even on a read-only network drive) so that they can be used by more than one user or application at a time.

Important Before you run the Setup program, turn off any network messaging services or TSRs that allow the network or another user to send a message to your screen. These popup programs can cause Windows Setup to fail if a message is received while Windows is being installed.

To place Windows files on a network server:

1. From your workstation, connect to the network drive where you want to copy the Windows files. Ensure that you have write access in that directory.
2. Insert the Windows 3.1 disk #1 into a floppy drive, and change to the drive letter of the floppy drive.
3. At the command prompt, type **setup /a** and press ENTER.
4. Follow the instructions on your screen.

   You are prompted to specify the network drive and directory you want the Windows files copied to (for example, W:\WINSHARE). You're also prompted to specify group registration information (group name and company name), which is stored and used when individual workstations are set up.
5. If your network requires it, make sure that all of the Windows files in this directory are marked as shareable. For more information, see your network documentation.

If you want to copy Windows files to other shared network servers in your organization, run **setup /a** from the shared directory you just copied Windows files to. This will copy files and mark them as read-only in the new directory. You'll be prompted for the company name and group name again, allowing you to specify a different company or group name for each shared WINDOWS directory in your organization.

Note Do not use the MS-DOS **copy *.*** command at the command prompt to copy Windows files to another network server, and do not delete the 8-byte files named **disk1**, **disk2**, **disk3**, and so forth, from the shared network directory. These files are required to run **setup /a** from the network.

After you place the Windows files on a network server, users can connect to this server and run **setup /n** to set up Windows for their workstations.

Before you instruct users to set up Windows, decide whether you want them to share Windows files on the network server or copy all Windows files to their workstations, and whether you want them to use Automated Setup. You may also want to customize Setup to add custom Program Manager groups and applications to the users' Windows environment. Or you might customize Setup to modify the Program Manager capabilities available to users. For more information, see "PROGMAN.INI: The Program Manager Initialization File" in Chapter 4, "The Windows Initialization Files."

Setting Up a Shared Copy of Windows: Setup /n

Flowchart 1.3 Setup /n Fails

Users can set up their workstations to access a shared copy of Windows by running **setup /n** (the Network Setup option). **Setup /n** copies some of the Windows files to the user's personal WINDOWS directory. The user can customize Windows according to personal preferences while still sharing most of the Windows files with other users.

To set up a shared copy of Windows on a workstation:

1. From the workstation, connect to the network drive that contains the Windows files.
2. Change to the directory where the Windows files are located.
3. Type **setup /n** at the command prompt and press ENTER.
4. Follow the instructions on the screen.

   You are prompted for the directory where you want to place the user files. This directory can either be on the local hard disk (for example, C:\WINDOWS) or on a private network directory (for example, U:\USERNAME). Specify a private directory on a network server if you are setting up a diskless workstation.

Setup copies only the files that are relevant to the user's system and desktop preferences—for example, the group and initialization files (.GRP and .INI files). All other Windows files are found in the shared network directory. Setup adds both the private WINDOWS directory and the shared network directory to the path in the user's AUTOEXEC.BAT file.

Note You cannot run **setup /n** from the installation disks. You can modify Setup so that it always sets up a shared copy of Windows on a workstation, even if a user specifies a different Setup option. To do this, specify **netsetup=true** in the **[data]** section of SETUP.INF. For more information, see "Forcing the Network Setup Option" in Chapter 2, "The Windows Setup Information Files." There are no custom options available to the user when you force Network Setup in this way.

Using Automated Setup: Setup /h



The Windows Automated Setup option (also called Batch Mode Setup) uses information defined in a system settings file to install Windows quickly and easily, with little or no user interaction. Automated Setup is useful if you have to set up many workstations, or if you want users to be able to set up on their own workstations without having to make any system configuration choices such as hardware components and printers, or choices about which optional Windows components to install.

Automated Setup can be used for setting up a shared copy of Windows or a nonshared copy, and you can use it if you are setting up Windows from a network drive or from the Windows Setup disks.

Note Before you run the Setup program, turn off any network messaging services or TSRs that allow the network or another user to send a message to your screen. These popup programs can cause Windows Setup to fail if a message is received while Windows is being installed.

To use the Automated Setup option:

1. Create a system settings file for each workstation configuration and place the system settings files in a directory where users have rights to open and read files, or copy it to the workstation disk.

   For details, see the following section, "Creating the System Settings File."

2. From the workstation, either you or a user can set up Windows by using the Automated Setup option. If you are setting up Windows from a network, connect to the network drive and change to the directory where the Windows Setup files are located. If you are setting up Windows from the Windows Setup disks, change to the letter of the floppy drive.

3. If you want to run a shared copy of Windows from the network, type **setup /h:***[drive:\path]filename* **/n**

   If you want to run a nonshared copy of Windows from a local drive or a network drive, type **setup /h:***[drive:\path]filename*

   The *filename* is the name of the system settings file that contains details about the system configuration. Include a path if the system settings file is not in the directory from which you are setting up Windows.

For example, you might create a system settings file named SETUPVGA.SHH in the same directory as the Setup program. To set up a complete copy of Windows on the workstation, at the command prompt type **setup /h:setupvga.shh** and press ENTER.

If you want to run a shared copy of Windows from the network, using this same example, connect to the network drive and directory where the Windows Setup files are located. At the command prompt, type **setup /h:setupvga.shh /n** and press ENTER.

Creating the System Settings File



Windows comes with a system settings template file called SETUP.SHH, which you can copy or modify to create a custom system settings file. The SETUP.SHH file is located on Microsoft Windows 3.1 disk #1 and is copied to the shared Windows directory when you use **setup /a**. You can modify this file with any text editor that can save unformatted text (ASCII) files.

Note When you create the .SHH file, use only ANSI characters. The .SHH file will be interpreted as containing characters from the ANSI character set, not from the code page character set currently installed on your system.

The system settings file is summarized in the following table.

Section	Purpose
[sysinfo]	Species whether System Configuration screen will appear during Setup.
[configuration]	Specifies the various devices on the user's system.
[windir]	Specifies where to put Windows files.
[userinfo]	Specifies the user and company name.
[dontinstall]	Specifies components that should not be installed.
[options]	Specifies various Setup options such as viewing README files, setting up existing applications, and starting the Windows tutorial.
[printers]	Specifies printers to set up.
[endinstall]	Specifies whether Setup modifies the CONFIG.SYS and AUTOEXEC.BAT files and whether the system is rebooted after Windows is installed.

The entries for these sections are described later in this section. Many of the entries require values from the Windows SETUP.INF, CONTROL.INF, and WIN.INI files. When you use **setup /a** to place Windows files on a network server, these files are copied to the shared WINDOWS directory. The master WIN.INI and SYSTEM.INI files are called WIN.SRC and SYSTEM.SRC in this directory. You can also find SETUP.INF on Microsoft Windows 3.1 disk #1. You can print or view all of these text files with any text editor.

For details about specific items in SETUP.INF, see the related entries in Chapter 2, "The Windows Setup Information Files." For details about specific items in WIN.SRC and SYSTEM.SRC, see the related entries in Chapter 4, "The Windows Initialization Files."

[sysinfo]

This section specifies whether the System Information screen is displayed during MS-DOS Mode Setup, so that the user can confirm or change the settings before continuing with Setup. Specify **showsysinfo=yes** to display the screen, or **showsysinfo=no** if you don't want to display the screen. (The default is **no**.)

For details about the System Information screen, see "MS-DOS Mode Setup" earlier in this chapter.

[configuration]

This section specifies the various devices on the user's system. If an entry for **[configuration]** is not included in the system settings file, the detected or default device will be used.

If you are updating Windows, Setup ignores some of these entries and uses the devices that are already installed. If you want to force the update and override the installed device, precede the **machine=**, **display=**, **mouse=**, and **network=** values with an exclamation point (!), for example, **display=!vga**. Any **keyboard=**, **language=**, or **kblayout=** values specified will automatically override the installed device.

Use	To
machine=	Specify the type of machine, using a profile string from the **[machine]** section of SETUP.INF. Example: **machine=ibm_compatible.**
display=	Specify the type of display, using a profile string from the **[display]** section of SETUP.INF. Example: **display=vga.**
mouse=	Specify the type of pointing device, using a profile string from the **[pointing.device]** section of SETUP.INF. Example: **mouse=ps2mouse.**
network=	Specify the type of network and the version number, using a profile string from the **[network]** section of SETUP.INF to specify the network. Example: **network=lanman** sets up the detected version of Microsoft LAN Manager. If there is a *[network_version]* section in SETUP.INF, for example **[lanman.versions]**, then use a number from that section to specify the version of network you want to set up. Separate the version number from the network type with a slash (/). Example: **network=lanman/01020000** sets up Microsoft LAN Manager version 2.0 Enhanced.
keyboard=	Specify the type of keyboard, using a profile string from the **[keyboard.types]** section of SETUP.INF. Example: **keyboard=t4s0enha.**
language=	Specify the language, using a profile string from the **[language]** section of SETUP.INF. Example: **language=enu.**
kblayout=	Specify the keyboard layout, using a profile string from the **[keyboard.tables]** section of SETUP.INF. Example: **kblayout=nodll.**

[windir]

This section specifies where to put the user's Windows files. If Windows is already installed in the specified directory, Setup updates it. If you do not specify a directory or if the specified directory is not valid, a dialog box appears during Setup, asking for the directory in which to set up Windows.

For example, if you want to install Windows in the WINDOWS directory on drive C, specify **c:\windows** in this section. This setting overrides the value of **defdir=** in the **[data]** section of SETUP.INF.

[userinfo]

This section specifies the user and company name. The first line specifies the user name. This line is required unless you are setting up a shared copy of Windows (**setup /n**). The second line specifies the company name, and is optional. Both names can be up to 30 characters long and must be enclosed in double quotation marks (" ") if they include blank spaces. For example:

```
"John Smith"
"My Company"
```

If you do not specify a user name, a dialog box appears during Setup to ask for the name.

If you use **setup /n** to set up a shared copy of Windows, the **[userinfo]** section is ignored because the information was supplied when you used **setup /a** to copy Windows files to the server.

[dontinstall]

This section specifies optional Windows components that you do not want to set up. If this section is not present or is empty, all optional components are set up. You can choose not to install the components listed in the following table.

Specify	*If you do not want to install*
readmes	Readme files
accessories	Accessories (However, WRITE.EXE is always installed.)
games	Games
screensavers	Screen savers
bitmaps	Background wallpapers

[options]

This section specifies whether a user can set up applications during Setup and whether to start the Windows Tutorial at the end of Setup. If you don't want any of these options, omit this section. You can specify in this section the entries shown in the following table.

Specify	To
setupapps	Set up applications interactively during Setup. The Add/Remove File dialog box appears during Setup so the user can choose which applications to install.
autosetupapps	Automatically set up all applications found on the user's hard disk.
tutorial	Start the Windows Tutorial at the end of Setup.

You should not specify both **setupapps** and **autosetupapps**, but if you do, all applications on the hard disk will be set up.

[printers]

This section specifies which printers to install during Setup. To install a printer, you specify a printer name and a port. Use one of the descriptive strings found in the **[io.device]** section of the CONTROL.INF file to specify the printer name. The printer name must be enclosed in double quotation marks (" ") if it contains blank spaces. Use one of the values found in the **[ports]** section of WIN.INI to specify the port. The printer name and the port must be separated by a comma. For example, to specify the HP LaserJet III printer connected to port LPT1, include this entry:

```
"HP LaserJet III",LPT1:
```

If you don't want to install any printers during Setup, omit this section.

[endinstall]

This section specifies what should happen after Windows is successfully installed. You can specify whether Setup should modify CONFIG.SYS and AUTOEXEC.BAT and then exit to MS-DOS, restart Windows, or restart the system.

Specify	If you want Setup to
configfiles=modify	Make all changes to CONFIG.SYS and AUTOEXEC.BAT for the user. When Setup changes these files, the previous CONFIG.SYS and AUTOEXEC.BAT files are saved as CONFIG.OLD and AUTOEXEC.OLD in the root directory. (If CONFIG.OLD or AUTOEXEC.OLD are already present, the previous file will be renamed CONFIG.000 or AUTOEXEC.000, or the next number that does not conflict with an existing file.) If you are setting up Windows on a diskless workstation, Setup disregards this entry.
configfiles=save	Save proposed changes to the CONFIG.SYS and AUTOEXEC.BAT files in alternate files (CONFIG.WIN and AUTOEXEC.WIN) stored in the WINDOWS directory. The CONFIG.SYS and AUTOEXEC.BAT files are not changed; you must make these changes later. (If CONFIG.WIN or AUTOEXEC.WIN are already present, the previous file will be renamed CONFIG.000 or AUTOEXEC.000, or the next number that does not conflict with an existing file.)
endopt=exit	Return to MS-DOS on completion of Setup.
endopt=restart	Restart Windows on completion of Setup.
endopt=reboot	Restart the user's system on completion of Setup. If this setting is specified and the user uses **setup /n** to install Windows, Setup will return to MS-DOS without rebooting after completing the installation.

Sample System Settings File

The following system settings file installs Windows on a PC-compatible computer with these characteristics:

- A VGA display, Microsoft mouse, LAN Manager 2.0 Enhanced network, and 101-key U.S. keyboard.
- The System Information screen is displayed for confirmation, and Windows files are put in the C:\WINDOWS directory.
- The games are not installed.
- An HP LaserJet III printer is installed on LPT1.
- The CONFIG.SYS and AUTOEXEC.BAT files are modified by Setup.
- After Windows is set up, Windows is restarted.

```
[sysinfo]
showsysinfo=yes

[configuration]
machine=ibm_compatible
display=vga
mouse=ps2mouse
network=lanman/01020000
keyboard=t4s0enha
language=enu
kblayout=nodll

[windir]
c:\windows

[userinfo]
"user's name"
"company name"

[dontinstall]
games

[options]

[printers]
"HP LaserJet III",LPT1:

[endinstall]
configfiles=modify
endopt=restart
```

Maintaining Windows with Setup

You can use Setup to maintain Windows after the initial installation. When you run Setup from within Windows, you can review the same options as during original installation for choosing system settings, installing Windows applications, and adding or removing Windows components. For details about using Setup to make changes or additions, see Chapter 15, "Maintaining Windows with Setup," in the *Windows User's Guide*.

To make other changes to a system from within Windows you can:

- Choose the 386 Enhanced icon in Control Panel to define device contention settings, task scheduling for multitasking in 386 enhanced mode, and virtual memory settings.
- Choose the Ports icon or Printers icon in Control Panel to define settings for ports and devices.

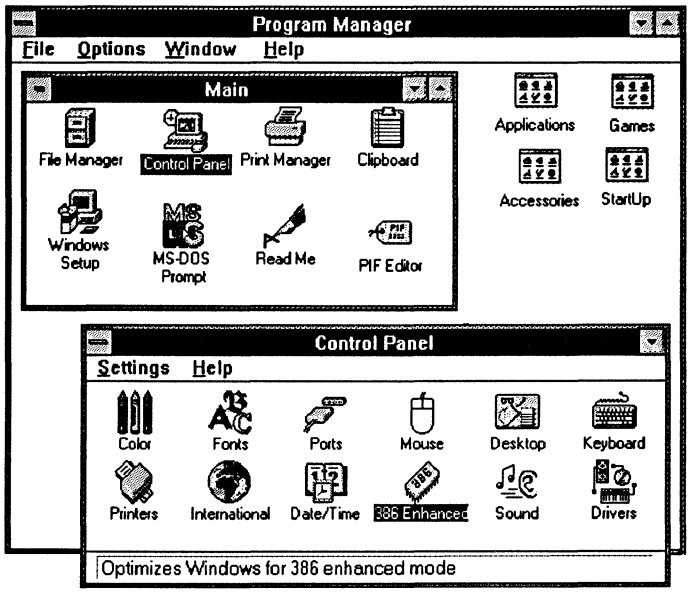


- Drag icons to the Startup Group in Program Manager to specify which applications will run whenever you start Windows.
- Use the Program Manager commands to create new groups. For example, use the File Properties command to change icons and specify other elements of an application's appearance on the desktop.

For details about using these Windows components, see the related chapters in the *Windows User's Guide.*

Figure 1.3

Program Manager and Control Panel



Chapter

2 The Windows Setup Information Files

This chapter contains information that can be used to create custom Microsoft Windows setup information files (SETUP.INF, CONTROL.INF, and APPS.INF) for multiple installations.

Note You should only modify the setup information files if you want to specify custom drivers, applications, and machine types for multiple systems.

Related information

- *Windows User's Guide:* Chapter 15, "Maintaining Windows with Setup"
- *Windows Resource Kit:* Chapter 3, "The Windows Files"; Chapter 4, "The Windows Initialization Files"; "Troubleshooting Setup" in Chapter 13, "Troubleshooting Windows 3.1"

Contents of this chapter

About the Setup Information Files 86
SETUP.INF: The Setup Information File 88
- General Installation Sections 89
- Display Driver Sections 94
- Keyboard and Code Page Sections 97
- Mouse Driver Sections 100
- Network Installation Sections 101
- System Fonts Sections 104
- Copy-Files Sections 105
- Program Manager Groups Sections 109
- Fonts Sections 112
- Incompatible Driver Sections 113
- Miscellaneous Sections 114
- System Configuration Sections 117

APPS.INF: The Application Installation Information File 121
- [dontfind] 121
- [pif] 122

CONTROL.INF: The Control Panel Information File 126
- Printer Driver Sections 126
- [country] 128

Modifying .INF Files for Custom Installations 128
- Modifying .INF Files to Install Custom Windows 129
- Modifying .INF Files to Install Custom Applications 130
- Adding Custom PIF Information in APPS.INF 133

About the Setup Information Files

Three files supply information used by the Windows Setup program: SETUP.INF, CONTROL.INF, and APPS.INF. The entries in each of these files are discussed in detail in separate sections of this chapter.

- SETUP.INF is the key file that Setup uses to install the Windows system files and application files, and to create program groups that you can use to run applications from Program Manager. This file is on the Windows installation disk #1 and is not compressed.

  Setup checks information in SETUP.INF during both MS-DOS Mode Setup and Windows Mode Setup. The SETUP.INF entries specify which files to copy based on the system hardware configuration and the choices specified during installation. You can modify the SETUP.INF entries to accommodate specific hardware and software configurations for installing Windows on multiple machines.



- APPS.INF is the file that contains information for installing applications with Windows. You can modify this file to install PIF information for custom applications during Windows installation. Setup places the APPS.INF file in the SYSTEM subdirectory during installation.



- CONTROL.INF is the file that contains specific information about printers and international settings. You can modify this file to install custom printer drivers. Setup places the CONTROL.INF file in the SYSTEM subdirectory during installation.

For other ways to customize Setup to meet the installation needs for your site, see the Custom Installation Decision Tree in Chapter 1, "Windows 3.1 Installation."

The setup information (.INF) files are text files in a format similar to WIN.INI and SYSTEM.INI. The .INF files are broken into logical groups called sections. Each section has a header that is an identifier in square brackets, such as **[shell]**, with values assigned under each section header. Brief comments are included below many section headers to explain the contents of that section.

Each section has a format similar to this:

[section]
keyname = value ; comment

Entry	Meaning
[section]	The name of a section. The enclosing brackets ([]) are required, and the left bracket must be in the leftmost column on the screen.
keyname	The name of an entry, which can consist of any combination of letters and digits. Some *keyname* entries must be followed immediately by an equal sign (=).
value	The value of each entry, which can be an integer, a string, or a quoted string, depending on the entry. Any entry listed in this chapter that does not appear in your .INF file has the default value.
; comment	Some entries include brief comments below the header or on the same line as an entry. You can include comments anywhere in an .INF file by prefacing the comment with a semicolon (;).

In this chapter, the items that appear in bold in an entry are actual words included in the entry. For example: **[win.net]**. Values that must be substituted in the entry are shown in italics. For example:

keyname = profile, description, filesize

Case (capital or lowercase letters) does not matter for values, unless specified for a particular item. Some items must be enclosed in double quotation marks (" "). For example: **caption= "Windows Setup"**.

Note The order of the sections in an .INF file is not important, except for the placement of the **[blowaway]** section in SETUP.INF. Also, the entries do not necessarily appear alphabetically as they are listed in this chapter. So to change an entry, you must search for it within the section.

To change entries in a setup information file:

1. Create a backup copy of the .INF file and save the original version in its unedited form.
2. Open the copy of the .INF file with a text editor.
3. Edit the specific entries and save the file in ASCII (text-only) format.

Important Always back up the .INF file before you open it, and use care when making changes with a text editor. Incorrect changes can lead to unexpected results when you run Setup. Also, some editors can damage characters with ANSI values of greater than 127. Use a nonformatting text editor to change these files, or make sure to save the files in ASCII (text-only) formats.

SETUP.INF: The Setup Information File

This discussion describes details about each section in SETUP.INF, and how and when to modify entries. The discussion follows roughly the same order as sections appear in SETUP.INF, in these topical categories:

- General installation sections
- Display driver sections
- Keyboard and code page sections
- Mouse driver sections
- Network installation sections
- System fonts and other font sections
- Copy-files sections
- Program Manager group sections
- Incompatible driver sections
- Miscellaneous sections
- System configuration sections

The following table summarizes the purpose of the sections in SETUP.INF.

Type of section	*Section name*	*Purpose*
General installation sections	**[setup]**	Defines the Windows Setup Help file
	[run]	Defines programs to run after Setup
	[dialog]	Defines installation screen text
	[data]	Defines general data to support Setup
	[winexec]	Defines core fields for Windows Mode Setup
	[disks]	Defines disks that Setup can use
	[oemdisks]	Defines extra disks that Setup can use
	[user]	Defines the location of TEMP file containing user information
	[windows]	Defines the minimum files for WINDOWS directory
	[windows.system]	Defines the minimum files for SYSTEM subdirectory
	[windows.system.386]	Defines the minimum files for 386 machine; plus **[bluemax]** and **[386max]** subentries for 386 expanded memory managers
	[shell]	Defines the shell entry in SYSTEM.INI for Setup
Display driver sections	**[display]**	Defines a set of display drivers
	[.3gr]	Defines font files for specific 386 grabbers
Keyboard and code page sections	**[keyboard.drivers]**	Maps keyboard short names and descriptors
	[keyboard.types]	Defines keyboard short names
	[keyboard.tables]	Maps keyboard names for international support
	[codepages]	Maps code pages for international support
Mouse driver sections	**[pointing.device]**	Defines mouse driver keynames
	[dos.mouse.driver]	Maps Windows mouse driver names with MS-DOS mouse drivers
	[lmouse]	Defines data for Logitech MS-DOS mouse drivers
Network installation sections	**[network]**	Defines names for network files and .INI file changes for networks
	[Network_Version]	Defines information for specific versions of network drivers
	[Network_Specific]	Defines entries for .INI files for specific networks
System fonts sections	**[sysfonts]**	Defines system font files by display resolution
	[fixedfonts]	Defines fixed-pitch font files by display resolution
	[oemfonts]	Defines OEM system font files by display resolution

Type of section	Section name	Purpose (continued)
Copy-files sections	**[win.copy]**	Defines files to copy for 80286 PCs
	[win.copy.net]	Defines files to copy for network workstations
	[win.copy.net.386]	Defines files to copy for 80386 and higher PCs for network workstation
	[win.copy.386]	Defines files to copy for 80386 and higher PCs
	[DelFiles]	Defines files to delete when upgrading to Windows 3.1
	[RenFiles]	Defines files to rename when upgrading to Windows 3.1
	[win_copyFiles]	Sections that define files to copy for specific items
Program Manager Groups sections	**[new.groups]**	Defines Program Manager groups to modify for Windows 3.1 upgrade
	[progman.groups]	Defines Program Manager groups to be created
	[group#]	Defines properties for items in program groups
Fonts sections	**[fonts]**	Defines vector and raster screen font files based on display resolution
	[ttfonts]	Defines TrueType font files to copy
Incompatible driver sections	**[compatability]**	Defines drivers to remove in CONFIG.SYS
	[incompTSR1]	Defines TSRs and drivers that cause problems during Setup
	[incompTSR2]	Defines TSRs and drivers that cause problems if running with Windows
	[block_devices]	Defines incompatible block devices
Miscellaneous sections	**[installable.drivers]**	Defines data for multimedia drivers
	[translate]	Defines filename translation data for version 3.0 OEM disks
	[update.files]	Defines installable drivers to update to version 3.1
	[ini.upd.patches]	Temporarily maps .INI file updates
	[blowaway]	Defines end of installation and start of configuration sections
	[ini.upd.31]	Defines .INI file updates for Windows 3.1
System configuration sections	**[system]**	Maps system short names and descriptors
	[machine]	Defines names for specific computers
	[special_adapter]	Defines special data for EtherLink MC cards
	[ebios]	Defines extended BIOS names
	[language]	Defines language DLL names

General Installation Sections

The following sections in SETUP.INF define the installation options and specify the filenames and locations of related files for MS-DOS Mode Setup:

[setup]	**[data]**	**[windows.system]**
[run]	**[disks],[oemdisks]**	**[windows.system.386]**
[dialog]	**[user]**	**[386max], [bluemax]**
[winexec]	**[windows]**	**[shell]**

[setup]

This section contains the location of the help file for the Setup program. If you want online help to be available during installation, there must be an entry similar to this:

```
[setup]
help= setup.hlp
```

[run]

The entries in this section specify which programs should run at the end of Setup. If you want to run an application when Setup is completed, enter its name here. For example, to run Control Panel after Setup is completed, include this entry:

```
[run]
"control"
```

[dialog]

This section contains the text strings used in the Setup dialog boxes. Some of the keynames and the default text assigned are:

```
[dialog]
caption = "Windows Setup"
exit = "Exit Windows Setup"
title = "Installing Windows 3.1"
options = "In addition to installing Windows 3.1, you can:"
printwait = "Please wait while Setup configures your printer(s)..."
copywait = "Welcome to Microsoft Windows 3.1! . . ."
(other copywait entries)
```

[winexec]

This section provides data needed by MS-DOS Mode Setup so that Setup can copy the correct Kernel file for Windows Mode Setup. You probably do not need to change these entries.

```
[winexec]
execstd= "dosx.exe "                ; MS-DOS Extender for standard mode
execcmd= " krnl286.exe /b /q:"      ; command and switches for kernel
exechimem= "xmsmmgr.exe"            ; extended memory manager for Setup
himemcmd= ""                        ; switches for HIMEM.SYS
Krnl386= 2:krnl386.exe              ; kernel for 386 enhanced mode
Krnl286= 2:krnl286.exe              ; kernel for standard mode
dosx= 2:dosx.exe                    ; disk location for DOSX.EXE file
```

[data]

This section defines a series of default settings for Setup. The first half of these entries specifies the hard disk space and minimum memory required for basic running without options. If you plan to install additional files during Setup, you must increase the minimum disk space required for installation. The second half of these entries refers to the default files and other data used by Setup.

These are the standard entries for the **[data]** section:

```
[data]
netadmin= 16000000                  ; 16.0 MB
netadminupd= 16000000               ; 16.0 MB
upd2x386full= 10500000,8000000      ; 10.5 MB, 8.0 MB
upd2x386net= 300000                 ;.3 MB
upd3x386full= 5500000,5000000       ; 5.5 MB, 5.0 MB
upd3x386net= 300000                 ;.3 MB
upd2x286full= 9000000,6500000       ; 9.0 MB, 6.5 MB
upd2x286net= 300000                 ;.3 MB
upd3x286full= 5500000,5000000       ; 5.5 MB, 5.0 MB
upd3x286net= 300000                 ;.3 MB
new386full= 10500000,8000000        ; 10.5 MB, 8.0 MB
new286full= 9000000,6500000         ; 9.0 MB, 6.5 MB
new386net= 300000,300000            ; .3 MB, .3 MB
new286net= 300000,300000            ; .3 MB, .3 MB

startup= WIN.COM                    ; file referred to when starting Windows
defdir= C:\WINDOWS                  ; default directory for installation
short name= Windows
welcome= "Windows 3.1"
deflang= enu                        ; default based on [language]
defxlat= 437                        ; default based on [codepage]
defkeydll= usadll                   ; default based on [keyboard.tables]
register= "regedit /s /u setup.reg" ; default file for registration database
tutor= "wintutor.exe "              ; tutorial to run at end of installation
NetSetup= FALSE                     ; TRUE allows only SETUP /N
MouseDrv= TRUE                      ; Setup uses the mouse
Version= "3.1"                      ; current Windows version
```

Note If you want to force users to set up using network installation (**setup /n**), make sure **netsetup=true** in **[data]**. For more information, see "Setting Up Windows for a Network" in Chapter 1, "Windows 3.1 Installation."

[disks] and [oemdisks]

These sections describe the disks that might be requested during installation. The format for entries in **[disks]** and **[oemdisks]** is:

n= path, disk name, disk tag

Entry	Meaning
n	A disk "number" as 1–9 and A–Z (single character only). The disk number 0 is reserved to represent the Windows Setup directory.
path	The path of disk *n*. The symbol "." tells Setup to look for the disk in the same drive as the Windows Setup disks.
disk name	The label of disk *n*, enclosed in quotes. There must be a disk name for each disk number used.
disk tag	The volume name for disk *n* (used to determine if the correct disk is in the drive).

This is a sample entry for Windows installation disks:

```
[disks]
1=. ,"Microsoft Windows 3.1 Disk #1",disk1
2=. ,"Microsoft Windows 3.1 Disk #2",disk2
3=. ,"Microsoft Windows 3.1 Disk #3",disk3
4=. ,"Microsoft Windows 3.1 Disk #4",disk4
5=. ,"Microsoft Windows 3.1 Disk #5",disk5
6=. ,"Microsoft Windows 3.1 Disk #6",disk6

[oemdisks]
Z=. ,"HP DeskJet Series v2.0 disk (from printer box or contact HP)",diskz
```

If you want to install other applications from disk during Windows installation, you can add an entry in the **[disks]** section. For details, see "Modifying .INF Files for Custom Installations" later in this chapter.

Note In other entries in SETUP.INF, a number followed by colon, such as **1:**, identifies which Windows installation disk contains that file, based on the entries in the **[disks]** section. If Windows is installed from a network server where all files exist in a single directory, Setup ignores the disk numbers.

[user]

This section describes which installation disk contains the user and company identification. This is a temporary file.

[windows]

This section identifies which files MS-DOS Mode Setup will copy into the WINDOWS directory. The assignment of **Net** for the **win.src** and **system.src** entries tells Setup to copy these files only for the Windows Administrative Setup option (**setup /a**), to prepare a shared copy of Windows for running Setup over a network.

```
[windows]
1:setup.hlp
1:setup.txt
3:control.hlp
2:win.src,          Net          ; for shared Windows setup
1:system.src,       Net          ; for shared Windows setup
1:winhelp.exe
```

[windows.system]

This section defines which files MS-DOS Mode Setup will copy into the Windows SYSTEM subdirectory. For details about the files copied during Windows Mode Setup, see "Copy-Files Sections" later in this chapter.

```
[windows.system]
1:gdi.exe
1:user.exe
1:win.cnf
1:lzexpand.dll
2:ver.dll
```

[windows.system.386] and Related Sections

These sections describe files to be copied into the SYSTEM subdirectory for 80386 machines:

```
[windows.system.386]
1:cpwin386.cpl

[386max]                         ; files copied if system is using 386Max
2:386max.vxd
2:windows.lod

[bluemax]                        ; files copied if system is using BlueMax
2:bluemax.vxd
2:windows.lod
```

[shell]

This section tells Setup which application to install as the Windows shell. The default shell is Program Manager.

```
[shell]
progman.exe, "Windows Program Manager"
```

Display Driver Sections

Windows display drivers require three kinds of files:



- **The Windows display driver** (a .DRV file; for example, VGA.DRV), which contains code for communication between the display adapter board and Windows. These files are specified in the **[display]** section.
- **The grabber file** (.2GR for standard mode; .3GR for 386 enhanced mode), which contains code to support data exchange between Windows and non-Windows applications. These files are specified in entries in the **[display]** section. The fonts for 386 grabbers are specified in *[.3gr]* sections. In Windows 3.0, these files had .GR2 and .GR3 filename extensions.
- **The Virtual Display Driver** (VDD*x*.386), which provides virtual display support in 386 enhanced mode. The related VDD files are specified in entries in the **[display]** section.

The naming conventions listed parenthetically are not required. However, most drivers use this naming convention.

If you are using a display driver other than those in the Windows package, you can add a custom entry that contains the display driver data in **[display]**. For Setup to correctly install third-party display drivers, you must copy the driver and the appropriate grabber files, and you must also update the **286grabber=** or **386grabber=** entries in the **[boot]** section of SYSTEM.INI.

During Custom Setup, you can specify the display type. Setup copies the matching system fonts for that display type. For information about the related system font files, see "System Fonts Sections" later in this chapter.

[display]

This section specifies information about various display devices. This is the format for entries in this section:

keyname= driver, description, resolution, 286grabber, logo code, VDD, 386grabber, ega.sys, logo data, optional section

Entry	Meaning
keyname	The profile of the driver, used to refer to the driver in other sections of SETUP.INF.
driver	The filename of the display device driver.
description	A description of the driver, enclosed in quotes, which appears in the Change System Settings dialog box in Setup.
resolution	The aspect ratio for the display, as three comma-separated numbers enclosed in quotes.
286grabber	The filename of the appropriate 286 grabber, which supports data exchange between Windows and non-Windows applications in standard mode.
logo code	The filename for the Microsoft Windows logo code.
VDD	The filename for the Virtual Display Driver.
386grabber	The filename for the appropriate 386 grabber, which supports data exchange in a virtual machine under 386 enhanced mode.
ega.sys	The filename for the data that determines whether an EGA.SYS driver needs to be installed.
logo data	The filename for the appropriate Microsoft logo bitmap.
optional section	The name of a section that contains additional information about the display driver.

Important Do not change description strings for drivers, because they are used during installation to make a decision to upgrade files, depending on which driver is already installed. Changing description string can cause errors in updating drivers.

These *optional sections* in SETUP.INF define details about display drivers:

[v7vga]	**[8514]**	**[tiga2]**	**[v7d]**	**[xgasm]**
[vgamono]	**[8514s]**	**[v7b]**	**[v7f]**	**[xgalg]**
	[tiga1]	**[v7c]**	**[v7e]**	**[xgalo]**

The *optional sections* have this format:

file, destination, ini file, section, OldEntry, NewEntry

Entry	Meaning
file	An optional file to be copied; this item can be left null.
destination	The directory where the file is to be copied; this is null if *file* is null. The value is **0:** if the file is to be copied to the WINDOWS directory or **0:system** if the file is copied to the SYSTEM subdirectory.
ini file	Any .INI file to be modified or created, such as WIN.INI or SYSTEM.INI; this item is optional.
section	The section of the .INI file that is to be modified.
OldEntry	Any entry, enclosed in quotes, to be removed from the .INI file; this item is optional.
NewEntry	Any new entry, enclosed in quotes, to be added in the .INI file.

This example shows the default entry for the 8514 display, which is a high-resolution IBM display:

```
[display]
8514= 2:8514.drv, "8514/a", "100,120,120", 3:vgacolor.2gr, 2:vgalogo.lgo,2:vdd8514.386,
    3:vgadib.3gr,, 2:vgalogo.rle, 8514
```

This is the *optional section* for the 8514 driver, which modifies the **dpi=** entry in the **[8514.drv]** section in SYSTEM.INI:

```
[8514]
,,system.ini,8514.DRV,"dpi=","dpi=120"
```

[.3gr] Font Sections

The 386 grabber supports copying text and graphics in a non-Windows application and translating it for Windows to use. The **[vga.3gr]**, **[vga30.3gr]**, **[v7vga.3gr]**, **[ega.3gr]**, **[vgadib.3gr]**, **[plasma.3gr]**, **[herc.3gr]**, and **[tiga.3gr]** sections specify which font files to copy for the 386 grabber described in the section name. (The 386 grabber itself is specified in an entry for the specific display device in **[display]**.)

Each entry in a **[*.3gr*]** section is a list of filenames for the appropriate fonts, including the installation disk number. For example, this entry specifies fonts for the 386 grabber that supports Video 7 VGA:

```
[V7VGA.3gr]
2:CGA40WOA.FON,2:CGA40850.FON
2:CGA80WOA.FON,2:CGA80850.FON
2:EGA40WOA.FON,5:EGA40850.FON
2:EGA80WOA.FON,2:EGA80850.FON
```

If code page 437 is used (the default for character support in the U.S.), then Setup copies the *WOA.FON files. If the code page is not 437, Setup copies the *WOA.850 files.

Keyboard and Code Page Sections

These sections specify the files for the keyboard drivers and code page support for Windows:

[keyboard.drivers] **[keyboard.types]**
[keyboard.tables] **[codepages]**

You might want to change these sections of SETUP.INF if you want to use a different keyboard driver, or if you want to use different foreign language support than is specified in SETUP.INF. If you change the keyboard support in these sections, you should also modify the **deflang=**, **defxlat=**, and **defkeydbl=** entries in the **[data]** section earlier in SETUP.INF.

[keyboard.drivers]

This section maps the actual keyboard driver filenames to the keyboard short names defined in **[keyboard.types]**:

```
[keyboard.drivers]
kbd= 2:keyboard.drv
kbdhp= 2:kbdhp.drv
```

[keyboard.types]

This section creates the keyboard short names used in the **[machine]** section. Each entry in this section also specifies whether a keyboard-specific DLL must be installed. The format for entries in this section is:

short name=description, DLL filename

Entry	Meaning
short name	The abbreviated descriptor to be used in the **[machine]** section.
description	A description, enclosed in quotes, that appears in the Change System Settings dialog box in Setup.
DLL filename	The filename for any supporting DLL required; this can be null.

The standard entries for this section are:

```
[keyboard.types]
t3s0alat= "All AT type keyboards (84 - 86 keys)" ,nodll
t1s2at&t= "AT&T '301' keyboard",nodll
t1s4at&t= "AT&T '302' keyboard",nodll
t4s0enha= "Enhanced 101 or 102 key US and Non US keyboards" ,nodll
t3s0hp1= "Hewlett-Packard Vectra keyboard (DIN)",nodll
t4s40oliv= "Olivetti 101/102 A keyboard" ,nodll
t1s0oliv= "Olivetti 83 key keyboard" ,nodll
t3s10oliv= "Olivetti 86 Key keyboard" ,nodll
t2s1oliv= "Olivetti M24 102 key keyboard" ,usadll
t1s42oliv= "PC-XT 83 key keyboard" ,nodll
t1s0pcxt= "PC/XT - Type keyboard (84 keys)",nodll
```

[keyboard.tables]

This section maps short names for DLLs to disk locations and filenames for the DLLs required by specific keyboard types. This section is useful only if you need foreign language support for Windows.

Note Changes to this section can affect **defkeydll=** in **[data]**.

```
[keyboard.tables]
beldll= 2:kbdbe.dll , "Belgian"
bridll= 2:kbduk.dll , "British"
cafdll= 2:kbdfc.dll , "Canadian Multilingual"
dandll= 2:kbdda.dll , "Danish"
dutdll= 2:kbdne.dll , "Dutch"
findll= 2:kbdfi.dll , "Finnish"
fredll= 2:kbdfr.dll , "French"
candll= 2:kbdca.dll , "French Canadian"
gerdll= 2:kbdgr.dll , "German"
icedll= 2:kbdic.dll , "Icelandic"
itadll= 2:kbdit.dll , "Italian"
latdll= 2:kbdla.dll , "Latin American"
nordll= 2:kbdno.dll , "Norwegian"
pordll= 2:kbdpo.dll , "Portuguese"
spadll= 3:kbdsp.dll , "Spanish"
swedll= 2:kbdsw.dll , "Swedish"
swfdll= 2:kbdsf.dll , "Swiss French"
swgdll= 2:kbdsg.dll , "Swiss German"
nodll= , "US"
usadll= 2:kbdus.dll , "US"
usddll= 2:kbddv.dll , "US-Dvorak"
usxdll= 2:kbdusx.dll, "US-International"
```

[codepages]

Setup refers to this section when any code page other than code page 437 is installed in the system. If Setup detects another code page, it installs the correct files for the translation table and (for VGA displays) the correct OEM font.

Note Changes to this section can affect **defxlat=** in **[data]**.

This is the format for entries in this section:

codepage = xlat table, OEM font, WOA font, description

Entry	Meaning
codepage	The table number used to associate the keys on the keyboard with the appearance of characters on the display.
xlat table	The filename for the translation table used to associate the OEM font file with the code page.
OEM font	The filename, with installation disk number, containing the correct OEM font.
WOA font	The filename, with installation disk number, containing the correct font to be used by the Windows grabber.
description	A string, enclosed in quotes, showing the country for which the code page is used.

```
[codepages]
863= 1:xlat863.bin, 2:vga863.fon, 2:app850.fon, "Canadian-French (863)"
861= 2:xlat861.bin, 2:vga861.fon, 2:app850.fon, "Icelandic (861)"
865= 2:xlat865.bin, 2:vga865.fon, 2:app850.fon, "Nordic (865)"
850= 2:xlat850.bin, 2:vga850.fon, 2:app850.fon, "Multi-Lingual (850)"
860= 3:xlat860.bin, 2:vga860.fon, 2:app850.fon, "Portuguese (860)"
437= , , 2:dosapp.fon, "English (437)"
```

Mouse Driver Sections

The **[pointing.device]**, **[dos.mouse.driver]**, and **[lmouse]** sections define information about the pointing device used while running Windows. If you want to use a mouse driver that is not included with Windows, you can add custom mouse driver references in these sections.

[pointing.device]

This section provides a table of information for the pointing devices supported by Windows 3.1. This is the format for each entry in this section:

keyname= mouse driver, mouse descriptor, VMD, optional section

Entry	Meaning
keyname	The profile name.
mouse driver	The filename for the mouse driver, including the installation disk number.
mouse descriptor	A description of the driver, enclosed in quotes, that appears in the Change System Settings dialog box in Setup.
VMD	The filename with installation disk number for the Virtual Mouse Device support file (for using the mouse with non-Windows applications in 386 enhanced mode). If multiple VMDs are specified, they must be separated by commas and enclosed in quotes. Example: **"x:*vmd,2:1vmd.386"**.
optional section	The name of a section that contains additional information about a mouse driver.

The assignment of ***vmd** as the *VMD* value indicates that this file is built into WIN386.EXE (and therefore is not a separate file on the installation disks); **x:** refers to the disk where WIN386.EXE is located.

```
[pointing.device]
hpmouse= 2:hpmouse.drv, "HP Mouse (HP-HIL)", x:*vmd
lmouse= 3:lmouse.drv, "Logitech", 2:lvmd.386, lmouse
ps2mouse= 2:mouse.drv,"Microsoft, or IBM PS/2",x:*vmd
genius1= 2:mscmouse.drv,"Genius serial mouse on COM1", 1:mscvmd.386
genius2= 2:msc3bc2.drv, "Genius serial mouse on COM2", 1:mscvmd.386
msmouse2= 2:mscmouse.drv,"Mouse Systems serial or bus mouse", 1:mscvmd.386
msmouse1= 2:msc3bc2.drv, "Mouse Systems serial mouse on COM2",1:mscvmd.386
nomouse= 2:nomouse.drv, "No mouse or other pointing device", x:*vmd
kbdmouse= 2:kbdmouse.drv,"Olivetti/AT&T Keyboard Mouse",x:*vmd
```

[dos.mouse.driver]

This section maps the Windows mouse driver to the appropriate MS-DOS mouse driver. Setup updates the MS-DOS mouse driver for compatibility with Windows 3.1.

```
[dos.mouse.drivers]
mouse.sys= 4:mouse.SYS, "MS DOS Mouse driver .SYS ver 7.XX"
mouse.com= 4:mouse.com, "MS DOS Mouse driver .COM ver 7.XX"
mousehp.sys= 3:mousehp.SYS,"HP DOS Mouse driver .SYS ver 7.XX"
mousehp.com= 3:mousehp.com,"HP DOS Mouse driver .COM ver 7.XX"
```

[lmouse]

This section provides data to support the Logitech MS-DOS mouse driver.

```
[lmouse]
2:lmouse.com,0:,,,,
```

Network Installation Sections

The network installation sections in SETUP.INF are in three parts:

- The basic **[network]** section.
- [*Network_Version*] sections that provide information for specific versions of network drivers.
- [*Network_Specific*] sections that describe SYSTEM.INI entries and other information for specific networks.

The information that you might add to customize these sections for your site could include entries for updated network drivers in the [*Network_Version*] sections, or entries in the [*Network_Specific*] sections to specify information for SYSTEM.INI, such as **emmexclude=** entries.

[network]

This section is used to associate the network *keyname* with the appropriate files that must be installed to support the network from within Windows.

This is the format for entries in this section:

network keyname= driver, description, HelpFile, optional file, WininiSectName, SysiniSectName, VDD, VDD, . . .

Entry	Meaning
network keyname	The profile for the specific network driver.
driver	The network driver filename and installation disk number.
description	A description of the driver, enclosed in quotes, that appears in the Change System Settings dialog box in Setup.
HelpFile	The filename for any associated help file.
optional file	The filename for any optional file to be installed.
WininiSectName	The section name to add or modify in WIN.INI.
SysiniSectName	The section name to add or modify in SYSTEM.INI.
VDD	The filenames for the Virtual Display Device support files for non-Windows applications in 386 enhanced mode. Any number of filenames can be specified.

These are the standard entries for the **[network]** section:

```
[network]
nonet= ,"No Network Installed"
3open= ,"3Com 3+Open"
3share= ,"3Com 3+Share" ,,,msnet_stuff
lantastic= ,"Artisoft LANtastic"
banyan= ,"Banyan Vines"
dlr= ,"IBM OS/2 LAN Server"
pclp= ,"IBM PC LAN Program"
lanman= ,"Microsoft LAN Manager"
msnet= ,"Microsoft Network (or 100% compatible)", ,,,msnet_stuff
novell= ,"Novell Netware"
pathworks= ,"DEC Pathworks"
10net= ,"TCS 10Net"
```

[Network_Version] Sections

These sections provide information about specific network versions.

Section name	Network
[10net.versions]	TCS 10Net
[3open.versions]	3Com 3+Open
[banyan.versions]	Banyan Vines
[dlr.versions]	IBM OS/2 LAN Server
[lanman.versions]	Microsoft LAN Manager
[lantastic.versions]	Artisoft LANtastic
[novell.versions]	Novell Netware
[pathworks.versions]	DEC Pathworks
[pclp.versions]	IBM PC LAN Program

The format for these sections is:

version #=description, optional files, optional sections

Entry	Meaning
version #	The version number for the network driver.
description	A descriptive string, enclosed in quotes.
optional files	Any optional files to be installed.
optional sections	One or more [*Network_Specific*] sections in SETUP.INF.

For example, this is the entry describing Microsoft LAN Manager versions:

```
[lanman.versions]
xx000000="versions 1.X",,lmbasic
00020000="version 2.00 Basic",,lmbasic
01020000="version 2.00 Enhanced",,lmenh20
00021000="version 2.10 Basic",,lmbasic21
01021000="version 2.10 Enhanced",lanman21.drv,lmenh21nodrv,lmenh21
```

This entry describes additional details for five different versions of LAN Manager, including references to the **[lmbasic]**, **[lmenh20]**, and **[lmbasic21]** network-specific sections.

[Network_Specific] Sections

These [*Network_Specific*] sections specify data that Setup must add to WIN.INI or SYSTEM.INI, as indicated by the *WininiSectName* and *SysiniSectName* entries in the **[network]** section described earlier in this chapter.

These are the [*Network_Specific*] sections in SETUP.INF:

[10net41]
[10net41a]
[10net42]
[10net42def]
[10net50]
[3open1]
[ban4]
[ban41]
[ban41def]
[dec4]
[dec41]
[dlr12]
[dlr12csd]
[dlr13]
[lant3]
[lant4]
[lant4def]
[lmbasic]
[lmbasic21]
[lmenh20]
[lmenh21]
[lmenh21nodrv]
[msnet_stuff]
[novell301]
[novell321]
[old_pclp]
[pclp132]
[unsupported_net]

This is the format for these sections:

[network keyname]
ini-keyname=section, entry-keyname, value

Entry	Meaning
network keyname	The keyname defined in the **[network]** section.
ini-keyname	Initialization filename (such as SYSTEM.INI).
section	The section name in the .INI file.
entry-keyname	The keyname for the entry in the .INI file.
value	The value to be assigned to the .INI entry.

For example, this is the entry for Novell NetWare version 3.01:

```
[novell301]
warning=2
driver=2:netware.drv
vxd=x:*vnetbios,2:vnetware.386,2:vipx.386
sysdir=2:netware.hlp,2:nwpopup.exe
windir=2:ipx.obj,2:ipxodi.com,2:netx.com,2:lsl.com
windir=2:tbmi2.com
```

System Fonts Sections

Setup refers to the **[sysfonts]**, **[fixedfonts]**, and **[oemfonts]** sections to install a set of fonts to use as the system font. Setup determines the appropriate fonts to install by finding the nearest match for the display resolution defined in the **[display]** section. This is the format for entries in the system fonts sections:

font filename, description, resolution

These are the standard system fonts sections:

```
[sysfonts]
1:egasys.fon,"EGA (640x350) resolution System Font", "133,96,72"
1:vgasys.fon,"VGA (640x480) resolution System Font", "100,96,96"
1:8514sys.fon,"8514/a (1024x768) resolution System Font", "100,120,120"
1:egasys.fon,"AT&T (640x400) resolution System Font", "120,96,80"

[fixedfonts]
2:egafix.fon,"EGA (640x350) resolution Fixed System Font", "133,96,72"
1:vgafix.fon,"VGA (640x480) resolution Fixed System Font", "100,96,96"
1:8514fix.fon,"8514/a (1024x768) resolution Fixed System Font", "100,120,120"
2:egafix.fon,"AT&T (640x400) resolution Fixed System Font", "120,96,80"
```

```
[oemfonts]
1:egaoem.fon,"EGA (640x350) resolution Terminal Font (USA/Europe)", "133,96,72",1
1:vgaoem.fon,"VGA (640x480) resolution Terminal Font (USA/Europe)", "100,96,96",1
1:8514oem.fon,"8514/a (1024x768) resolution Terminal Font (USA/Europe)",
    "100,120,120",1
1:egaoem.fon,"AT&T (640x400) resolution Terminal Font (USA/Europe)","120,96,80",1
```

For more information about fonts, see "Fonts Sections" later in this chapter, and see Chapter 9, "Fonts."

Copy-Files Sections

Windows Mode Setup uses these sections to identify which files to copy to the WINDOWS directory or to the SYSTEM subdirectory. Setup uses these copy-files section names to determine which groups of files to copy for the appropriate configuration of Windows:

- **[win.copy]** for 80286 machines
- **[win.copy.net]** and **[win.copy.net.win386]** for network workstations
- **[win.copy.win386]** for 80386 machines

Under each copy-files section is a set of entries in this format:

from, to

Entry	*Meaning*
from	A file or a reference to a group of files.
to	The destination for the files, as either **0:** to indicate the WINDOWS directory, or **0:system** to indicate the SYSTEM subdirectory.

For example, the entry **1:myfile.txt,0:** copies the file MYFILE.TXT from disk #1 to the WINDOWS directory. If the **[win.copy.win386]** section contains an entry **#pwin386,0:system**, then Setup copies all files in the **[pwin386]** section to the SYSTEM subdirectory. If the name of an entry in any copy-files section is preceded by a "#" character, the entry refers to a group of files in a separate entry in SETUP.INF. For example, the **[win.copy]** section is:

```
[win.copy]
#net, 0:
#win.shell, 0:
#win.other, 0:system
```

The **#net, 0:** entry indicates that Setup should copy the files listed in the **[net]** section into the user's WINDOWS directory. For all items in the copy-files sections, a text string, enclosed in quotes, shows the text that Setup displays to indicate which files are currently being installed.

Basic Copy-Files Sections

The main copy-files sections include:

- **[win.copy]**, which specifies that the files listed in **[net]**, **[win.shell]**, and **[win.other]** be copied for 80286 machines.
- **[win.copy.net]**, which specifies that the files listed in **[net]** be copied for network setup.
- **[win.copy.net.win386]**, which specifies that the files listed in **[net.win386]** be copied for network setup for 80386 machines..
- **[win.copy.win386]**, which specifies that the files listed in **[net]**, **[win.shell]**, **[pwin386]**, and **[win.other]** be copied for 80386 machines.

The following describes the contents of the supporting copy-files sections:

- The entries in **[net]** specify Control Panel source files and memory driver files to copy for network setup:

```
[net]
5:CONTROL.SRC, "Windows User Files"
4:WINVER
```

- The entries in **[win.devices]** and **[win.devices.386]** specify the memory drivers to install:

```
[win.devices]
; These devices will be copied on all machines
5:HIMEM.SYS,    "XMS Memory Manager"
3:SMARTDRV.EXE, "Disk Caching Program"
3:RAMDRIVE.SYS, "RAM Drive Program"

[win.devices.win386]
; These devices will be copied on 386 machines only
5:HIMEM.SYS,    "XMS Memory Manager"
4:EMM386.EXE,   "LIM Expanded Memory Manager"
3:SMARTDRV.EXE, "Disk Caching Program"
3:RAMDRIVE.SYS, "RAM Drive Program"
```

- The **[win.other]** entries specify various drivers, DLLs, and font files to copy, plus the Windows core files. Example:

```
[win.other]
4:WINOLDAP.MOD,"Windows Drivers"
4:WIN87EM.DLL
5:SYSEDIT.EXE
5:COMMDLG.DLL
(other entries)
```

- The **[win.shell]** entries specify files for applications that are installed with the Windows shell. Example:

```
[win.shell]
4:PROGMAN.EXE, "Windows Program Manager"
5:TASKMAN.EXE, "Windows Task Manager"
4:WINFILE.EXE, "Windows File Manager"
3:CLIPBRD.EXE, "Windows Clipboard Viewer"
3:CLIPBRD.HLP, "Windows Clipboard Viewer help"
(other entries)
```

- The **[pwin386]** entries specify files that provide alternate executable files for 386 enhanced mode:

```
[pwin386]
4:WINOA386.MOD,"386 Enhanced Mode files"
5:WIN386.PS2
4:WIN386.EXE
5:VTDAPI.386
```

[DelFiles] and [RenFiles] Sections

These sections lists files that will be deleted or renamed when Setup upgrades Windows version 3.0 to version 3.1. You can add or remove items from these lists, using wildcards in filenames wherever appropriate. For example, some of the standard entries in these sections are:

```
[DelFiles]
; The first files in the list are deleted from Windows System directory (for local setup).
; Note that DelFiles section always is processed before RenFiles section.
tmsr?.fon
helv?.fon
swapfile.exe
kernel.exe
(other entries)

[RenFiles]
; Similar section but for file renaming to create .bak if needed
MIDIMAP.CFG,MIDIMAP.OLD
ADLIB.DRV,MSADLIB.DRV
```

Copy-Files Sections for Applications

The entries in the **[win.apps]**, **[win.games]**, **[win.dependents]**, **[win.scrs]**, **[win.bmps]**, and **[win.readme]** sections describe the Windows applications, screensaver, and README files to be copied if the user requests these files during Custom Setup. All files listed are copied in Express Setup.

For details about adding information for custom applications, see "Modifying .INF Files for Custom Installations" later in this chapter.

These sections have this format:

#:filename, description, filesize, profile string

Entry	Meaning
#:filename	The disk number and filename of the application or other element to be copied.
description	The string, enclosed in quotes, that will be displayed in the Windows Setup dialog box for selecting applications to copy during Custom Setup.
filesize	The expanded size of the file, in bytes.
profile string	The string used by Setup to find the item in the appropriate **[*group#*]** section, to determine if and how the item is added to a Program Manager group with DDE. The *profile string* is also used to find any needed .DLL files in the **[win.dependents]** section.

The end of each section contains an entry that specifies the disk space required for all the files in that section. If you add an entry to any section, you must also increment the disk space number.

The following summarizes the contents of the application copy-file sections:

- The **[win.apps]** entries specify the Windows applications, utilities, and help files. Example:

```
[win.apps]
3:CALC.EXE, "Calculator" , 42944, calc
3:CALENDAR.EXE, "Calendar", 58432, calendar
3:CARDFILE.EXE, "Cardfile", 88576, cardfile
(other entries)
diskspace=1075780
```

- The **[win.dependents]** entries specify the DLL files for utilities. Example:

```
[win.dependents]
pbrush= 3:PBRUSH.DLL
recorder= 3:RECORDER.DLL
```

- The **[win.games]** entries specify the game files and their help files. Example:

```
[win.games]
3:SOL.EXE,"Windows Solitaire" , 180688, sol
3:WINMINE.EXE, "Windows Minesweeper", 27728, winmine
3:SOL.HLP,"Windows Solitaire help", 18072
3:WINMINE.HLP, "Windows Minesweeper help" , 18072
diskspace=242008
```

- The **[win.scrs]** entries specify the screensaver files. Example:

```
[win.scrs]
3:SSMARQUE.SCR, "Marquee Screen Saver" , 16752
(other entries)
```

- The **[win.bmps]** entries specify the bitmap files for wallpaper, wave-form audio files, and MIDI sample files. Example:

```
[win.bmps]
3:ARCADE.BMP,"Arcade Wallpaper", 630
3:tada.wav, "Trumpet Sound", 27804
5:passport.mid, "Passport MIDI Song" , 40294
(other entries)
```

- The **[win.readme]** entries specify the README text files:

```
[win.readme]
3:readme.wri,"General Readme" , 80768, readme
3:networks.wri, "Networks Readme" , 45568
3:winini.wri,"Win.ini Readme" , 85120
3:sysini.wri,"System.ini Readme" , 135040
3:printers.wri, "Printers Readme" , 17920
diskspace=306176
```

Program Manager Groups Sections

After Setup copies files to the appropriate Windows directories, it tells Program Manager to create the groups listed in the **[new.groups]** or **[progman.groups]** section. Related *[group#]* sections contain entries that describe the contents of the groups that Program Manager will create.

This is the format for entries in the **[new.groups.]** and **[progman.groups]** sections:

section = Progman group, Min/Max

Entry	*Meaning*
section	The name of the *[group#]* section that contains the group definition.
Progman group	The profile name that identifies the group in Program Manager.
Min/Max	A switch defining whether to minimize or maximize the group when loading Windows, where 1=maximize, nothing=minimize.

For details about installing a custom group in Program Manager, see "Modifying .INF Files for Custom Installations" later in this chapter.

[new.groups]

If you are updating Windows from 3.0 to 3.1, Setup uses the **[new.groups]** section instead of the **[progman.groups]** section.

This is the standard entry for this section:

```
[new.groups]
group7=Main,1
group1=StartUp
group2=Accessories
group6=Games
```

[progman.groups]

This section contains a list of the Program Manager groups that are built for new installations of Windows 3.1. These are the standard entries in this section:

```
[progman.groups]
group3=Main,1
group4=Accessories
group5=Games
group1=StartUp
```

[group#] Sections

After the Program Manager groups are created, program items are installed in the groups based on the [*group#*] section entries. This is the format for entries in these sections:

description, filename.exe, icon filename, icon n, profile

Entry	Meaning
description	A string, enclosed in quotes, that defines the text that appears below the icon in Program Manager.
filename.exe	The executable filename, including the extension, to be represented in the group. If *filename.exe* is null, the item will be deleted from the group.
icon filename	The filename of the icon extraction file that contains the icon for the program item. If this is blank, the first icon in *filename.exe* is used.
icon n	The number from the icon extraction index, which defines the offset of the icon specified in *icon filename*. If this item is null, the first icon is used. If you want to use the *n*th icon, use the number *n*–1. That is, for the third icon, specify **2**.
profile	A string that defines whether the item is added to the group. If this value is null, the item will always be added to the group. Otherwise, the item will only be added if it was installed with the Add/Remove Files dialog box in Custom Setup.

These are the standard entries for the [*group#*] sections:

```
[group3]                                    ; for [progman.groups]
"File Manager", WINFILE.EXE
"Control Panel", CONTROL.EXE
"Print Manager", PRINTMAN.EXE
"Clipboard Viewer", CLIPBRD.EXE
"MS-DOS Prompt", DOSPRMPT.PIF, PROGMAN.EXE, 9
"Windows Setup", SETUP.EXE
"PIF Editor", PIFEDIT.EXE
"Read Me", README.WRI,,,readme

[group4]                                    ; for [progman.groups]
"Write",WRITE.EXE
"Paintbrush", PBRUSH.EXE,,,pbrush
"Terminal",TERMINAL.EXE,,, terminal
"Notepad", NOTEPAD.EXE,,, notepad
"Recorder",RECORDER.EXE,,, recorder
"Cardfile",CARDFILE.EXE,,, cardfile
"Calendar",CALENDAR.EXE,,, calendar
"Calculator", CALC.EXE,,, calc
"Clock",CLOCK.EXE,,, clock
"Object Packager", PACKAGER.EXE,,, packager
"Character Map", CHARMAP.EXE,,, charmap
"Media Player", MPLAYER.EXE,,, mplayer
"Sound Recorder",SOUNDREC.EXE,,, soundrec

[group5]                                    ; for [progman.groups]
"Solitaire", SOL.EXE,,,sol
"Minesweeper",WINMINE.EXE,,, winmine

[group1]                                    ; [progman.groups] and [new.groups]

[group2]                                    ; for [new.groups]
"Object Packager", PACKAGER.EXE,,, packager
"Character Map", CHARMAP.EXE,,, charmap
"Media Player", MPLAYER.EXE,,, mplayer
"Sound Recorder",SOUNDREC.EXE,,, soundrec
"PIF Editor"
"Write",WRITE.EXE
"Control Panel"

[group6]                                    ; for [new.groups]
"Solitaire", SOL.EXE,,,sol
"Minesweeper",WINMINE.EXE,,, winmine

[group7]                                    ; for [new.groups]
"PIF Editor", PIFEDIT.EXE
"Read Me"
"Read Me", README.WRI,,,readme
"Tutorial"
```

Fonts Sections

The last files that Setup copies are the raster, vector, and TrueType font files, based on the resolution of the display as defined in the **[display]** section. Three vector fonts (ROMAN.FON, MODERN.FON, and SCRIPT.FON) are resolution independent and are installed by default.

This is the format of entries in the **[fonts]** section:

filename, description, resolution

Entry	Meaning
filename	The filename of the file containing the font.
description	The font description, enclosed in quotes, that will appear in the Fonts dialog box in the Control Panel, plus the point sizes that are installed.
resolution	Three comma-separated numbers, enclosed in quotes, that represent the font aspect ratio.

For example, the entries for VGA displays in the **[fonts]** section are:

```
[fonts]
(other entries)
6:SSERIFE.FON, "MS Sans Serif 8,10,12,14,18,24 (VGA res)", "100,96,96"
(other entries)
6:COURE.FON, "Courier 10,12,15 (VGA res)", "100,96,96"
(other entries)
6:SERIFE.FON, "MS Serif 8,10,12,14,18,24 (VGA res)", "100,96,96"
(other entries)
6:SYMBOLE.FON, "Symbol 8,10,12,14,18,24 (VGA res)", "100,96,96"
(other entries)
6:SMALLE.FON, "Small (VGA res)", "100,96,96"
```

This is the format for entries in the **[ttfonts]** section:

font header file, description, font file, fontfamily/flags

Entry	Meaning
font header file	The filename for the TrueType header source file.
description	The string, enclosed in quotes, that will appear in the Fonts dialog box. The first letter of each word in the string needs to be capitalized.
font file	The filename for the TrueType font.
fontfamily/flags	The name of the font family plus a flag, where **0000** is a normal-weight font, **0100** is a bold font, and **1000** is an italic font. The *fontfamily/flag* value must be enclosed in quotes.

These are the entries for the **[ttfonts]** section:

```
[ttfonts]
5:ARIAL.FOT, "Arial (TrueType)", 5:arial.ttf, ""
5:ARIALBD.FOT, "Arial Bold (TrueType)", 5:arialbd.ttf, "Arial0100"
5:ARIALBI.FOT, "Arial Bold Italic (TrueType)", 5:arialbi.ttf, "Arial1100"
5:ARIALI.FOT, "Arial Italic (TrueType)", 5:ariali.ttf, "Arial1000"
5:COUR.FOT, "Courier New (TrueType)", 5:cour.ttf, ""
5:COURBD.FOT, "Courier New Bold (TrueType)", 5:courbd.ttf, "Courier0100"
5:COURBI.FOT, "Courier New Bold Italic (TrueType)", 5:courbi.ttf, "Courier1100"
5:COURI.FOT, "Courier New Italic (TrueType)", 5:couri.ttf, "Courier1000"
3:TIMES.FOT, "Times New Roman (TrueType)", 4:times.ttf, ""
3:TIMESBD.FOT, "Times New Roman Bold (TrueType)", 4:timesbd.ttf,
    "Times New Roman0100"
3:TIMESBI.FOT, "Times New Roman Bold Italic (TrueType)", 4:timesbi.ttf,
    "Times New Roman1100"
4:TIMESI.FOT, "Times New Roman Italic (TrueType)", 4:timesi.ttf, "Times New Roman1000"
5:SYMBOL.FOT, "Symbol (TrueType)", 5:symbol.ttf, ""
```

Incompatible Driver Sections



Some MS-DOS drivers and TSRs are not compatible with Windows 3.1. Setup checks these sections in SETUP.INF for names of incompatible drivers and TSRs: **[compatibility]**, **[incompTSR1]**, **[incompTSR2]**, and **[block_devices]**. You can add to the lists of drivers and TSRs in any of these sections.

Some manufacturers are updating their drivers for compatibility with Windows 3.1. To determine whether you have the latest version of a driver, check with the driver manufacturer.

If Setup finds a command in the CONFIG.SYS file that loads any of the files listed in **[compatibility]**, it removes the command and leaves a blank line. You can reinstall these files at a later time.

Setup checks three other sections: **[incompTSR1]**, **[incompTSR2]**, and **[block_devices]**. The TSRs and drivers listed in **[incompTSR1]** prevent Setup from running and should be removed from memory and from AUTOEXEC.BAT and CONFIG.SYS before running Setup. The TSRs and device drivers listed in **[incompTSR2]** can cause problems if they are running during Setup or when you start Windows. The block devices listed in the **[block_devices]** section are not compatible with Windows 3.1.

Note Check the current entries in SETUP.INF on your disk, and check also the information in the Readme file in your Windows package. Microsoft will update the entries in the incompatible driver sections when new information is known about these and other programs. For more information about these TSRs and Setup, see "Troubleshooting TSRs During Setup" in Chapter 13, "Troubleshooting Windows 3.1."

Miscellaneous Sections

Several sections provide information about files to update and translate filenames for OEM installation disks.

[installable.drivers]

This section specifies drivers that can be installed for multimedia Windows, including drivers for MIDI, sequencers, waveform audio, SoundBlaster boards, and device timers. The format for entries in this section is:

keyname= filename, types, description, VxDs, default parameters

Entry	Meaning
keyname	The profile name of the driver.
filename	The filename of the device driver, with disk number.
types	The kinds of multimedia device, enclosed in quotes.
description	A descriptive string, enclosed in quotes.
VxDs	Filenames for any supporting files for virtual devices.
default parameters	The default parameters for the device, if any.

The standard entries for this section are:

```
[installable.drivers]
adlib= 3:adlib.drv,"MIDI","Ad Lib",4:vadlibd.386,
lapc1= 3:mpu401.drv,"MIDI","Roland LAPC1",,
midimapper= 4:midimap.drv,"MidiMapper","MIDI Mapper",,
mpu401= 3:mpu401.drv,"MIDI","Roland MPU401",,
sequencer= 4:mciseq.drv,"Sequencer","[MCI] MIDI Sequencer",,
soundblaster= 3:sndblst.drv,"Wave,MIDI","Creative Labs Sound Blaster 1.0",
    3:vsbd.386,,adlib
soundblaster2= 3:sndblst2.drv,"Wave,MIDI","Creative Labs Sound Blaster 1.5",
    3:vsbd.386,,adlib
timer= 5:timer.drv,"Timer","Timer",5:vtdapi.386,
thunder= 3:sndblst2.drv,"Wave,MIDI","Media Vision Thunder Board",3:vsbd.386,,adlib
wave= 4:mciwave.drv,"WaveAudio","[MCI] Sound",,"4"
```

[translate]

This section specifies translated OEM filenames, with their installation disk filenames and locations. Some examples of entries in this section include:

```
[translate]
WinDiskName=25,"Microsoft Windows 3.0 Disk"
WinDiskName=14,"Windows Disk #"
*vddvga=2:vddvga30.386
cti.lgo=2:herclogo.lgo
herclogo.rle=1:herclogo.rle
vddega=2:vddega.386
vgacolor.gr2=3:vgacolor.2gr
vgalogo.lgo=2:vgalogo.lgo
vgalogo.rle=2:vgalogo.rle
rgdi.lgo=2:vgalogo.lgo
rgdi.rle=2:vgalogo.rle
vga.gr3=1:vga30.3gr
(other entries for font, keyboard, DLL, and code page filename translations)
```

[update.files]

This section describes installable drivers to be updated if earlier versions of the files are found on the user's hard disk. The format for this section is:

location, filename

Entry	Meaning
location	The location for the file to be updated, as either **0:** to indicate the WINDOWS directory, or **0:system** to indicate the SYSTEM subdirectory.
filename	The filename of the file to be updated, with the disk number.

These are the standard entries for this section:

```
[update.files]
0:system,3:sndblst.drv
0:system,3:sndblst2.drv
0:system,3:adlib.drv
0:system,3:mpu401.drv
0:system,5:vtdapi.386
0:system,3:vsbd.386
0:system,4:vadlibd.386
```

Setup uses the **[Update.Dependents]** section to update dependents of files listed in **[Update.Files]**. These are the standard entries for this section:

```
[Update.Dependents]
msadlib.drv = 4:vadlibd.386
sndblst.drv = 4:vsbd.386
sndblst2.drv = 4:vsbd.386
```

[ini.upd.patches]

This section is used by MS-DOS Mode Setup and Windows Mode Setup to temporarily rename profile strings for .INI entries during Setup if the original profile string exists in the .INI file and has a value defined. The format for this section is:

ini file, section, temporary profile, original profile

Entry	Meaning
ini file	The initialization file.
section	The section name in the .INI file.
temporary profile	The profile to be replaced, enclosed in quotes.
original profile	The value to be added in the .INI file, enclosed in quotes.

These are example entries for this section:

```
[ini.upd.patches]
system.ini, Boot, "olddrivers" ,"drivers"
win.ini, Desktop, "oldwallpaper" ,"wallpaper"
system.ini, Boot, "oldtaskman" ,"taskman.exe"
```

[blowaway]

This section marks the end of the installation information and the beginning of the configuration sections. This section tells Windows Mode Setup where to stop reading the SETUP.INF file, because Windows Mode Setup does not need this information.

[ini.upd.31]

This section tells Setup which lines in SYSTEM.INI and WIN.INI are to be replaced with new values in the upgrade from Windows version 3.0 to version 3.1. This is the format for entries in this section:

ini file, section, OldEntry, NewEntry

Entry	Meaning
ini file	The .INI file that is to be changed.
section	A section name in the .INI file.
OldEntry	The old entry to be replaced, enclosed in quotes. If this value is null, Setup assumes that the *NewEntry* is to be added.
NewEntry	The new entry to be added, enclosed in quotes.

Some of the entries in this section include:

```
[ini.upd.31]
system.ini, 386Enh, "keyboard=bpvkd.386", "keyboard=*vkd"
system.ini, 386Enh, "device=vpicda.386", "device=*vpicd"
system.ini, 386Enh, "device=*vhd" , "device=*blockdev"
system.ini, 386Enh, "device=*vpd" ,
system.ini, 386Enh, , "device=*pagefile"
win.ini , FontSubstitutes, , "Helv=MS Sans Serif"
win.ini , FontSubstitutes, , "Tms Rmn=MS Serif"
win.ini , FontSubstitutes, , "Courier=Courier New"
win.ini , FontSubstitutes, , "Times=Times New Roman"
win.ini , FontSubstitutes, , "Helvetica=Arial"
(other entries for fonts that have been replaced or removed for Windows 3.1)
```

System Configuration Sections

These sections provide information for installing appropriate system files for various machines.

[system]

The **[system]** section is used to map system short names used in the **[machine]** section to the appropriate system files. This section lists drivers that are selected only by the **[machine]** settings; these drivers can vary from system to system, and do not have special selections in the Setup menus:

```
[system]
system= 2:system.drv
sound= 1:mmsound.drv
comm= 1:comm.drv
hpsystem= 2:hpsystem.drv
```

[machine]

This section is used by Setup to install the appropriate system files for various machines. Most machines use the "MS-DOS System" option, but some specific original equipment manufacturers (OEMs) need special drivers or specific switches to be set in the SYSTEM.INI file for compatibility.

A system description for each entry in **[machine]** appears in the System Settings list in Setup. An asterisk (*) in the menu means that Setup will use the device specified in this section, overriding the description of the detected device for keyboards, mouse drivers, displays, and other elements.

This is the format for the entries in this section:

keyname= description, system drv, kbd drv, kbd type, mouse drv, disp drv, sound drv, comm drv, himem switch, ebios, Cookz

Entry	Meaning
keyname	The profile string for the system.
description	A string describing the system, enclosed in quotes, that will appear in the system-description dialog box in Setup.
system drv	The short name for the system driver, identified in **[system]**.
kbd drv	The short name for the keyboard driver, as defined in **[keyboard.drivers]**.
kbd type	The short name for the keyboard type, as defined in **[keyboard.types]**.
mouse drv	The name of the mouse driver, which is the *keyname* defined in **[pointing.device]**.
disp drv	The name of the display driver, which is the *keyname* defined in **[display]**.
sound drv	The name of the sound driver, as defined in **[system]**.
comm drv	The name of the communications driver, as defined in **[system]**.
himem switch	A number (1–8) that identifies the A20 handler, which allows HIMEM.SYS to access extended memory. For a list of these numbers, see Chapter 13, "Troubleshooting Windows 3.1."
ebios	The EBIOS support, defined in **[ebios]**.
Cookz	A *[Cookz]* section (called "cookies" in the SETUP.INF comments) that contains information for modifying entries in the .INI files, usually to define **[386enh]** entries in SYSTEM.INI.

Important Do not change the order of entries in the **[machine]** section.

The list of drivers in **[machine]** is a default list. If Setup detects a different display adapter than the type listed for the machine, the detected display driver will be installed.

For example, this shows the first three entries in the **[machine]** section:

```
[machine]
ibm_compatible= "MS-DOS System",system,kbd,t4s0enha,nomouse,egahires,
    sound,comm,,ebios,
ast_386_486= "AST: all 80386 and 80486 based machines",system,kbd,
    t4s0enha,nomouse,egahires,sound,comm,,ebios,ast_cookz
at_and_t= "AT&T PC",system,kbd,t4s0enha,nomouse,!olibw,sound,comm,,ebios,
(other entries)
```

These are the default [*Cookz*] specified in **[machine]** entries:

[apm_cookz]	**[everex_cookz]**	**[nec_pm_cookz]**
[apm_sl_cookz]	**[ibml40_cookz]**	**[t5200_cookz]**
[ast_cookz]	**[ncr386sx_cookz]**	**[zen386_cookz]**

The entries in the [*Cookz*] sections have this format:

ini file, section, cookie, needed file

Entry	*Meaning*
ini file	The .INI file that is to be modified.
section	The section to be changed in that .INI file.
cookie	The specific entry to be added to the section, enclosed in quotes.
needed file	Any related file that might be required.

For example, this [*Cookz*] section for the AST system specifies that the entry **emmexclude=E000-EFFF** should be added to the **[386enh]** section of SYSTEM.INI:

```
[ast_cookz]
system.ini,386enh,"emmexclude=E000-EFFF",
```

The [*Cookz*] section for MS-DOS systems with Advanced Power Management (APM) defines special drivers to be copied plus SYSTEM.INI additions:

```
[apm_cookz]
specialdriver,,,5:power.drv
system.ini,386enh,"device=vpowerd.386",5:vpowerd.386
,,,5:power.hlp
```

[special_adapter]

This section is used by MS-DOS Mode Setup to provide support for special adapters that require additional files or other changes to the standard Windows installation. For example, for the EtherLink MC adapter, the entry **DMABufferSize=32** is placed in the **[386enh]** section of SYSTEM.INI.

The default entries in **[special_adapter]** are:

```
[special_adapter]
eitherlink= DMAbuffersize,32          ; Needed if EtherLink MC is detected.
cdrom= 2:lanman10.386                 ; Needed for CD ROM extensions.
```

[ebios]

This section is used by the **[machine]** section to indicate which files are copied for extended BIOS support on the specific configuration. EBIOS means Extended Basic Input/Output System.

```
[ebios]
ebios= x:*ebios                    ; Required for most 386 machines.
hpebios= 2:hpebios.386,x:*ebios    ; Required for HP 386 machines.
```

[language]

Setup uses this section to install the language libraries that come with Windows. A language DLL must be installed for systems needing language support other than U.S. English. Each entry in this section has this format:

keyname=language DLL, DLL description, language ID

Entry	Meaning
keyname	The profile string for the short name for the language.
language DLL	The filename of the language DLL, including the installation disk number.
DLL description	A string, enclosed in quotes, that will be used to describe the language in Control Panel.
language ID	A special identifying code.

The default entries for this section are:

```
[language]
dan= 2:langsca.dll, "Danish",1030
nld= 2:langdut.dll, "Dutch" ,1043
enu=, "English (American)" ,1033
eng= 2:langeng.dll, "English (International)" ,2057
fin= 2:langsca.dll, "Finnish",1035
fra= 2:langfrn.dll, "French",1036
frc= 2:langeng.dll, "French Canadian" ,3084
deu= 2:langger.dll, "German",1031
isl= 2:langsca.dll, "Icelandic",1039
ita= 2:langeng.dll, "Italian",1040
nor= 2:langsca.dll, "Norwegian",1044
ptg= 2:langeng.dll, "Portuguese" ,2070
esp= 2:langspa.dll, "Spanish (Classic)" ,1034
esn= 2:langeng.dll, "Spanish (Modern)",3082
sve= 2:langsca.dll, "Swedish" ,1053
```

Note Changes to this section can affect **deflang=** in **[data]**.

APPS.INF: The Application Installation Information File



Setup places the APPS.INF file in the SYSTEM subdirectory during Windows installation. The APPS.INF file contains information in these basic sections:

- **[dialog]** contains the title text used when Setup displays the Setup Applications dialog box.
- **[base_PIFs]** defines a batch file for creating the _DEFAULT.PIF file, and specifies settings for COMMAND.COM.
- **[enha_dosprompt]** defines memory requirements for the MS-DOS Prompt when it runs under 386 enhanced mode Windows.
- **[dontfind]** lists Windows applications to be ignored during the Setup Existing Applications stage of installation. This section is described later in this section.
- **[pif]** contains PIF parameters for non-Windows applications. Additional sections define information for optimizing PIFs. This section is described later in this section.

If you have a custom PIF you want to install for multiple systems, or if you want to tailor application setup in other ways, you can change entries in APPS.INF.

Note Installing non-Windows applications with Setup helps to avoid conflicts in setting up PIFs for applications that use the same executable filename.

[dontfind]

This section is used when Setup is searching the hard disk for Windows applications. The entries in this section are overlooked by Setup, and these items are not placed in the Program Manager Applications Group. This prevents Setup from creating duplicate copies of standard Windows applications.

Some of the files in the **[dontfind]** section are listed in the following table. Check SETUP.INF in your system for the most up-to-date list.

Filenames

APM.EXE
CALC.EXE
CALENDAR.EXE
CARDFILE.EXE
CHARMAP.EXE
CLIPBRD.EXE
CLOCK.EXE
CONTROL.EXE
CVPACKW.EXE
DDESPY.EXE
DEWPROJ.EXE
DIALOG.EXE
DLGEDIT.EXE
DRWATSON.EXE
EQNEDIT.EXE
FONTEDIT.EXE
FTP.EXE
GRAPH.EXE
GRAFLINK.EXE
HEAPWALK.EXE
IMAGEDIT.EXE
IMPLIBW.EXE
JFPRINT.EXE
LIBW.EXE
LINKW.EXE
LWINHELP.EXE
MACRODE.EXE
MCWIN.EXE
MPLAYER.EXE
MSDOS.EXE
MSDRAW.EXE
MXWIN.EXE
NOTE-IT.EXE
NOTEPAD.EXE
PACKAGER.EXE
PAINT.EXE
PBRUSH.EXE
PIFEDIT.EXE
PLAY.EXE
PPTGRAPH.EXE
PRINTMAN.EXE
PROGMAN.EXE
QT.EXE
RCPPW.EXE
RCW.EXE
RECORDER.EXE
REGEDIT.EXE
REGLOAD.EXE
REVERSI.EXE
SDKPAINT.EXE
SETUP.EXE
SHAKER.EXE
SHED.EXE
SOL.EXE
SOUNDREC.EXE
SPOOLER.EXE
SPY.EXE
STRESS.EXE
SWAPFILE.EXE
SYSEDIT.EXE
TASKMAN.EXE
TBOOK.EXE
TERMINAL.EXE
TRANS.EXE
WHELP.EXE
WIN2WRS.EXE
WINBBS.EXE
WINCBT.EXE
WINFILE.EXE
WINHELP.EXE
WINMINE.EXE
WINTUTOR.EXE
WINVER.EXE
WORDART.EXE
WPCDLL.EXE
WPWINFIL.EXE
WRITE.EXE
ZOWOMIN.EXE

[pif]



Complete setup information for PIFs in Windows 3.1 is stored in APPS.INF (rather than a subset of options in SETUP.INF, as in earlier versions of Windows). Several related sections also appear in APPS.INF, the details of which are described below.

A PIF is always created for any non-Windows application when you install it to run under Windows, and a program item is added to the Applications group in Program Manager. You can add PIFs during Setup by specifying the disk location of a custom PIF in a [*win_copyFile*] section in SETUP.INF, or by letting Setup create a PIF from custom entries in APPS.INF.

For a list of the applications for which PIF information is included in APPS.INF, see "Applications with PIF Data in APPS.INF" in Chapter 7, "Setting Up Non-Windows Applications," in the *Windows Resource Kit.*

This is the format for each entry in the **[pif]** section:

exe file= pif name, window title, startup directory, close-window flag, icon filename, icon n, standard pif, 386 pif, ambiguous exe, optimized pifs

Entry	Meaning
exe file	The filename, with extension, for the application's executable file.
pif name	The filename for the PIF, without the extension.
window title	The string, enclosed in quotes, that will appear in the title bar.
startup directory	The directory that is the current directory when the application starts.
close-window flag	The flag that indicates whether the application window is automatically closed when the user quits the application. The value is **cwe** if the application window should be closed on exit; the value is null if the window should not be closed on exit.
icon filename	The filename from which to extract the program icon. The default is PROGMAN.EXE; in Windows 3.1, many PIFs list icons from MORICONS.DLL.
icon n	The number from the icon extraction table; the default is **0**.
standard pif	The APPS.INF section that contains the PIF settings for standard mode; the default is **[std_dflt]**.
386 pif	The APPS.INF section that contains the PIF settings for 386 enhanced mode; the default is **[enha_dflt]**.
ambiguous exe	The APPS.INF section that lists applications with same executable filename specified in *exe file*.
optimized pifs	Other APPS.INF sections that contain optimized PIF settings.

Important The list in **[pif]** must remain in alphabetical order by executable filename.

These are the entries for **[std_dflt]** and **[enha_dflt]**, which specify the values for the default standard mode and 386 enhanced mode PIFs:

```
[std_dflt]
; Default is a text mode application that does not directly modify COM ports.
; Other possible options are given for reference.
; (or) means the entry corresponds to a radio button group
minconvmem= 128
videomode= txt                  ; (or) gra
xmsmem= 0,0                     ; ##,, ## (min, max)
checkboxes=                     ; c1,c2,c3,c4,kbd,nse,pps,ata,aes,ces,psc,aps,nss
```

```
[enha_dflt]
; Other possible options are given for reference.
; (or) means entry corresponds to radio button group
convmem= 128,640                    ; ##,## (Required, Limit)
emsmem= 0,1024                      ; ##,## (Required, Limit)
xmsmem= 0,1024                      ; ##,## (Required, Limit)
dispusage= fs                       ; (or) win
execflags= ; bgd, exc
multaskopt= 50,100                  ; ##,## (Bgd Pri, Fgd Pri)
procmemflags= dit,hma               ; eml,xml,lam
dispoptvideo= txt                   ; (or) lgr,hgr
dispoptports= hgr                   ; txt,lgr
dispflags= emt                      ; rvm
otheroptions= afp                   ; cwa,ata,aes,ces,psc,aps,asp,aen
```

The options for [*std_**] and [*enha_**] entries are the same kinds of settings you make in the PIF Editor dialog boxes for standard and 386 enhanced mode.

The following example shows the basic **[pif]** entry for an application that uses LEARN.EXE as the executable filename:

```
LEARN.EXE= LEARN ,"Learning Microsoft Works",,cwe,moricons.dll,5,
     std_LEARN,enha_LEARN,amb_learn
```

These are the *optimized pifs* sections for Learning Microsoft Works:

```
;Learning Works (LEARN.EXE)
[std_LEARN]
minconvmem= 185
videomode= gra
xmsmem= 0,1024
checkboxes= kbd,pps

[enha_LEARN]
dispoptvideo= hgr
convmem= 185,520
emsmem= 0,4096
xmsmem= 0,4096
dispusage= win
execflags= exc
procmemflags= hma
dispoptports=
dispflags= emt,rvm
```

When you choose to have Setup install other applications to run under Windows, Setup displays a list of applications that use the same *exe file* so that you can choose which application you want to install. For example, these are the default *ambiguous exe* entries in APPS.INF for other applications besides Microsoft Works that use LEARN.EXE as the *exe file:*

```
[amb_learn]
;MS quick pascal, learning works, learning word (LEARN.EXE)
LEARN.EXE= LEARN ,"MS Quick Pascal Express 1.0",,cwe,moricons.dll,5,
    std_PASEXPES,enha_PASEXPES
LEARN.EXE= LEARN ,"Learning Microsoft Word 5.0",,cwe,moricons.dll,5,
    std_LEARN5,enha_LEARN5
LEARN.EXE= LEARN ,"Learning Microsoft Word 5.5 ",,cwe,moricons.dll,5,
    std_LEARN55,enha_LEARN55
```

And, as an example, these are the *optimized pifs* sections for Learning Microsoft Word 5.5:

```
[std_LEARN55]
minconvmem= 180
videomode= gra

[enha_LEARN55]
dispoptvideo= hgr
convmem= 180,195
emsmem= 0,0
xmsmem= 0,0
dispoptports=
```

You might want to include custom entries in APPS.INF if, for example, you want to use different PIFs for the same application. During Setup, users would be prompted to specify which PIF to install. This example adds PIF information for a custom application and uses a custom *optimized pif* section:

```
XYZAPP.EXE= xyzapp,"XYZ - Example APP",C:\XYZAPP,cwe,moricons.dll,25,
    std_dflt,enha_dflt,opt_xyz

[opt_XYZ]
XYZAPP.EXE= xyzapp2,"XYZ - Enhanced",C:\XYZAPP,cwe,moricons.dll,27,,enha_XYZ2,

[enha_XYZ2]
minconvmem= 204
videomode= gra
xmsmem= 320,1024
dispoptports= txt
```

Note Be sure you thoroughly understand the settings in PIF Editor before you attempt to edit APPS.INF to incorporate new standard settings. For details, see Chapter 8, "PIFS and PIF Editor."

CONTROL.INF: The Control Panel Information File



The CONTROL.INF file contains information that Control Panel uses during setup to install printer drivers and to provide international support. The three sections are:

- **[io.device]**, which defines the printers supported in Windows.
- **[io.dependent]**, which defines supplementary files required for some printer drivers.
- **[country]**, which defines international formats used by Control Panel.

Printer Driver Sections

The **[io.device]** and **[io.dependent]** sections provide information about printer drivers. If you want to install a printer driver during Setup other than a driver that is supplied with Windows, add references in these two sections.

A custom printer driver might be able to use the fonts already installed in Windows, or it might require additional files to be installed with it and specified in the **[sysfonts]**, **[fixedfonts]**, and **[oemfonts]** sections of SETUP.INF.

[io.device]

This section lists all of the supported printers for Windows 3.1. This section is used by Control Panel to install printer support. This is the format for each entry in this section:

filename, description, scaling string, scaling string

Entry	Meaning
filename	The filename and extension, with the installation disk number.
description	The string, enclosed in quotes, that appears in Control Panel and that is used in WIN.INI.
scaling string	One or two strings, enclosed in quotes, that indicate the scaling for the device.

A driver can have more than one entry in **[io.device]** to specify settings that correspond to different printers.

Some of the entries in this section include:

```
[io.device]
6:TTY.DRV,"Generic / Text Only","DEVICESPECIFIC"
6:HPPCL.DRV,6:unidrv.dll,"Agfa Compugraphic Genics","DEVICESPECIFIC"
6:pscript.DRV,"Apple LaserWriter II NTX","DEVICESPECIFIC"
6:HPPCL.DRV,6:unidrv.dll,"Apricot Laser","DEVICESPECIFIC"
6:HPPLOT.DRV,"AT&T 435","CONTINUOUSSCALING"
6:CITOH.DRV,6:unidrv.dll,"AT&T 470/475","DEVICESPECIFIC"
6:EPSON9.DRV,6:unidrv.dll,"Epson DFX-5000","DEVICESPECIFIC"
6:HPPCL.DRV,6:unidrv.dll,"HP LaserJet IID","DEVICESPECIFIC"
(other entries)
```

[io.dependent]

Control Panel checks this section to see if the printer driver requires other files to be copied to the SYSTEM subdirectory. The files to be copied might be soft font installers, font files, help files, or other support for the printer. This is the format for each entry in this section:

driver filename, files to be copied with the driver

The default entries in this section are:

```
[io.dependent]
pscript.drv = 6:pscript.hlp, 6:testps.txt
tty.drv = 6:tty.hlp
ibm4019.drv = 6:sf4019.exe
lbpiii.drv = 6:can_adf.exe
lbpII.drv = 6:can_adf.exe
pg306.drv = 6:sfinst.exe
hppcl.drv = 6:finstall.dll, 6:finstall.hlp, 6:unidrv.hlp
hpdskjet.drv = 6:finstall.dll, 6:finstall.hlp, 6:unidrv.hlp
hppcl5a.drv = 6:finstall.dll, 6:finstall.hlp, 6:hppcl5a.hlp, 6:hppcl5op.hlp
paintjet.drv = 6:dmcolor.dll, 6:unidrv.hlp
thinkjet.drv = 6:unidrv.hlp
canon130.drv = 6:unidrv.hlp
canon10e.drv = 6:unidrv.hlp
epson9.drv = 6:dmcolor.dll, 6:unidrv.hlp
epson24.drv = 6:dmcolor.dll, 6:unidrv.hlp
escp2.drv = 6:unidrv.hlp
fuji9.drv = 6:dmcolor.dll, 6:unidrv.hlp
fuji24.drv = 6:dmcolor.dll, 6:unidrv.hlp
ibmcolor.drv = 6:dmcolor.dll
oki9.drv = 6:unidrv.hlp
oki9ibm.drv = 6:unidrv.hlp
oki24.drv = 6:dmcolor.dll, 6:unidrv.hlp
nec24pin.drv = 6:dmcolor.dll, 6:unidrv.hlp
panson24.drv = 6:dmcolor.dll, 6:unidrv.hlp
panson9.drv = 6:dmcolor.dll, 6:unidrv.hlp
proprint.drv = 6:unidrv.hlp
proprn24.drv = 6:unidrv.hlp
```

[country]

This section is used by Control Panel for international formats. We recommend that you only make international modifications by choosing the Country icon in the Control Panel.

For example, these are the two entries for Canada. The "!" symbol is a field delimiter.

```
"Canada (English)", "2!1!0!2!1!1!0!2!1!1!!!$!,!.!/!:!,!dd/MM/yy!MMMM d, yyyy!ENG"
"Canada (French)", "2!2!3!2!1!1!0!2!8!1!!!$! !,!/!:!;!yy-MM-dd!d MMMM, yyyy!FRC"
```

The **[country]** section contains entries for these countries:

Australia	Germany	Spain
Austria	Iceland	Sweden
Belgium (Dutch)	Ireland	Switzerland (French)
Belgium (French)	Italy	Switzerland (German)
Brazil	Mexico	Switzerland (Italian)
Canada (English)	Netherlands	Taiwan
Canada (French)	New Zealand	United Kingdom
Denmark	Norway	United States
Finland	Portugal	Other Country
France	South Korea	

Modifying .INF Files for Custom Installations

You can create custom Windows installations for multiple systems by changing entries in the Setup information files to modify the following:

- The hardware options available for installation on individual workstations.
- The Setup options available for network installation.
- The files copied into the user's Windows directories.
- The applications that are installed automatically during Setup.

This section summarizes the activities related to modifying the .INF files for custom installations.

Modifying .INF Files to Install Custom Windows

If you want the users at your site to run a customized version of Windows, you can modify the SETUP.INF file before setting up Windows for the workstations.

You can change default Program Manager groups, create other groups, and add applications to these groups. You should only modify SETUP.INF if you want to set up a customized Windows desktop; you do not need to modify this file if you want the default Windows configuration.

Removing Hardware Devices from Lists in Setup

During Setup, users might be required to specify hardware devices from a list of supported devices. For example, when setting up a printer, the user is presented with a list of printers. The user must know what kind of printer the system is connected to and select that printer from the list. Users might also have to select their particular display, pointing device, network, or other hardware devices.

You can remove from these lists any devices that your users do not have by modifying the SETUP.INF and CONTROL.INF files. Be careful not to remove a device that a user might have, because this might cause the Setup program to run incorrectly.

To remove a printer from the list of printers:

1. Make a backup copy of the original .INF files, and edit the copies.
2. In the **[io.device]** section of CONTROL.INF, delete the entry for any printers you do not want to display in the Setup lists.

To remove other hardware devices from the lists in Setup:

- In SETUP.INF, delete the entry for the hardware device you do not want to display in the Setup lists.

  You can delete entries from **[display]**, **[system]**, **[keyboard.types]**, **[network]**, **[machine]**, and **[pointing.device]**.

If you remove an entry from SETUP.INF or CONTROL.INF, the entry will not appear when users install a printer from Control Panel or change system settings by running Setup from within Windows.

Specifying Windows Applications to Install

You can create a custom list of Windows applications, bitmaps, and screen-savers to be installed with Windows if you want to restrict or customize the installation for multiple systems. To do this, add or remove entries in the **[win.apps]**, **[win.dependents]**, **[win.games]**, **[win.scrs]**, and **[win.bmps]** sections of SETUP.INF.

Forcing the Network Setup Option

If you want to make sure that users set up a shared copy of Windows (the **setup /n** option), you can modify SETUP.INF so that even if the user types **setup** without the **/n** option, the Setup program installs a shared copy of Windows on the workstation.

To do this, make sure **netsetup=true** in the **[data]** section of SETUP.INF. The default entry is **netsetup=false**, which allows users to run any of the Setup options.

Modifying .INF Files to Install Custom Applications

If you want Windows to install a custom application for multiple systems, you can make changes in .INF files to install the application during Setup.

The examples in the following discussion show the steps for installing a custom application, QB.EXE, with its help file and PIF.

Copying Additional Files to a User's Directory

If you want the application files to be copied to the user's WINDOWS directory during Setup, you can specify additional files to copy during Setup in the **[net]** section of SETUP.INF for network setup, or specify files in **[win.copy]** to copy for stand-alone systems.

The files you want to copy must be either on a floppy disk or in the same directory as the Windows files on the network server. If the files are on a floppy disk, you must first assign a disk number in the **[disks]** section of SETUP.INF and then specify that disk number in the **[net]** or **[win.copy]** sections.

To specify an additional disk that Setup can prompt for:

- Add an entry to the **[disks]** section of SETUP.INF, including the disk number, path of the disk, name of the disk, and disk tag (used to determine if the disk is in the drive). Example:

```
[disks]
(other entries)
7=. ,"Quick Blibbet Application",disk7
```

The disk number refers to the Windows Setup disk where the file is located. Make sure the values for the *disk name* and *disk tag* items in the **[disks]** entry match the actual disk label and volume name.

To copy additional files to a user's system:

1. Add one or more entries to the **[win.copy]** or **[win.net]** sections in SETUP.INF for the files that Setup should copy. Or add a reference in one of these sections to an optional *[win_copyFiles]* section.

   You can also specify that the files be copied to a subdirectory of the WINDOWS directory.

   For example, to place the file in the user's C:\WINDOWS\APPDIR subdirectory, you might change the **[win.net]** section as follows:

```
[win.net]
(other entries)
#custom.apps,0:appdir                ; files are specified in [custom.apps]
```

2. If you specify a *[win_copyFiles]* section as the value in **[win.net]** or **[win.copy]**, add that section to SETUP.INF with a list of the files to copy. For each file you want to copy, include a disk number, a colon, the filename, and a description (optional).

   If the files are on the network, you can specify any disk number; the number will be ignored, but you must still specify a number. If the files are on a floppy disk, use the number specified in the **[disks]** section.

   For example, to install the custom application QB.EXE and its related PIF and help file, the **[custom.apps]** section should contain these entries:

```
[custom.apps]
7:QB.EXE,"Quick Blibbet",165000,qblibbet
7:QB.PIF,"QBlibbet",545
7:QB.HLP,"QBlibbet Help",15000
diskspace=180545
```

3. If you have a custom Windows application that uses a DLL, add an entry to the **[win.dependents]** section to install the DLL file. Example:

```
[win.dependents]
qblibbet= 7:QBLIBBET.DLL
(other entries)
```

Adding Custom Groups to Program Manager

You can edit SETUP.INF to change the content of the default Program Manager groups or to add your own custom groups. You should only modify SETUP.INF for this purpose if you want to create a custom Windows desktop for multiple systems. Otherwise, individual users can use the menu commands in Program Manager to customize their personal desktops.

If this SETUP.INF file will be used both for updating and for new installations, add the group to both the **[program.groups]** and the **[new.groups]** sections.

To add an application to a Program Manager group:

1. Add entries to the **[progman.groups]** and **[new.groups]** sections of SETUP.INF to install a custom Program Manager group. Example:

```
[progman.groups]                [new.groups]
(other entries)                 (other entries)
group9=QBlibbet                 group9=QBlibbet
```

2. Add one or more [*group#*] sections to SETUP.INF with an entry that describes the program item.

   Type the program title in double quotes (" ") followed by a comma (,) and then the application filename and icon information. If the application is not in the WINDOWS directory, include the path. Example:

```
[group9]
"Quick Blibbet",k:\qb\qblibbet.pif,moricons.dll,24,qblibbet
```

This example specifies the icon extraction file as MORICONS.DLL, a file that comes with Windows 3.1.

A custom group for your application will be created in Program Manager, with a program item in that group, when the other Program Manager groups are installed during Setup.

If the application is on a network, make sure each user has access to the appropriate network drive. After Setup, you might also want to specify a working directory for user files so that personal files don't remain in the shared application directory. You or a user can specify a working directory by modifying the Program Item Properties in Program Manager. For details, see Chapter 3, "Program Manager," in the *Windows User's Guide.*

Adding Custom PIF Information in APPS.INF



If your custom application uses the same executable filename as other applications listed in the **[pif]** section in APPS.INF, add entries in that file to define specific information about your application.

To add custom PIF information in APPS.INF:

1. In the basic entry in **[pif]**, add an *ambiguous exe* item. Example:

```
QB.EXE = QB,"Microsoft QuickBASIC",,cwe,moricons.dll,14,
    std_QB,enha_QB,amb_QB
```

2. Add a new *[amb_*]* section to define the PIF information for your application. Example:

```
[amb_QB]
;"Microsoft QuickBASIC" and "Quick Blibbet" both use QB.EXE
QB.EXE= QBlibbet,"Quick Blibbet",,cwe,moricons.dll,24,std_QB1,enha_QB1
```

3. Add *[std_*]* and *[enha_*]* sections if you want to define custom PIF information. Example:

```
[std_QB1]
minconvmem= 192
videomode= gra

[enha_QB1]
convmem= 380,640
execflags= bgd
```

Chapter

3 The Windows Files

A question that technical people often ask about Microsoft Windows is: *What does this file do?* This chapter describes the purpose for each file in the WINDOWS directory and the SYSTEM subdirectory.

For information about how to add to the list of files that are installed automatically with Windows, see "Modifying .INF Files for Custom Installations" in Chapter 2, "The Windows Setup Information Files."

Related information

- *Windows Resource Kit:* Chapter 2, "The Windows Setup Information Files"; Chapter 4, "The Windows Initialization Files"; Appendix C, "Windows 3.1 Disks and Files"
- *Glossary terms:* code page, EMS, XMS, protected mode, virtual device

Contents of this chapter

About the Windows Files 136
WIN.COM 136
The Core Files 137
Drivers, Fonts, and International Support Files 137
    Driver Files 137
    Font Files 144
    International Support Files 147
MS-DOS Support Components of Windows 148
    MS-DOS Driver Files 148
    WinOldAp and the Grabber Files 148
    Files for Standard Mode 150
    Files for 386 Enhanced Mode 150
Windows Applications, Setup, and Other Files 151
    Files for Windows Applications 151
    Setup-Related Files 152
    Other Files 153
Files You Can Delete 155

About the Windows Files

When Microsoft Windows runs, it performs all operating system duties except file system management, which MS-DOS still performs. Windows calls functions that are stored in a variety of executable files, driver files, and other dynamic-link libraries to manage the display, keyboard, and other devices, and to manage memory and execute programs.

The kinds of files that make up Windows 3.1 include:

- The WIN.COM file.
- The core dynamic-link libraries (kernel files, USER, and GDI) that contain the code and data for the Windows functions.
- The font files and the drivers for keyboard, display, system, mouse, printers, networks, multimedia, and other devices.
- The files that provide MS-DOS support components for Windows.
- The Windows applications files and other files such as shells, utilities, and accessories.

Flowchart 1.7 Expanding Files from the Windows Disks

For instructions on how to expand any files from the Windows installation disks, see Flowchart 1.7 on page 16. For technical information about the Windows 3.1 files, see the manuals for the Microsoft Windows Software Development Kit and Driver Development Kit.

WIN.COM

WIN.COM is the loader for Windows. It checks the machine type, memory configuration, and device drivers to determine which mode is appropriate to start Windows. To start Windows, there needs to be sufficient memory, an XMS driver present (such as HIMEM.SYS), and processor support for standard mode (80286 or higher) or 386 enhanced mode (80386 or higher).

After WIN.COM determines the appropriate operating mode, it uses the MS-DOS **exec** command to execute one of the following files, which in turn loads Windows:

- DOSX.EXE for standard mode
- WIN386.EXE for 386 enhanced mode

To build Windows, WIN.COM brings together a number of files:

- The core files
- The drivers
- The fonts and language support files
- Support files for non-Windows application
- MS-DOS support and various mode-specific files

The Core Files

Three files make up the Windows core components: Kernel, User, and GDI.

- The kernel files (KRNL286.EXE or KRNL386.EXE) control and allocate all the machine resources to manage memory, load applications, and schedule program execution and other tasks.
- USER.EXE creates and maintains windows on the screen, carrying out all requests to create, move, size, or destroy a window. User also handles requests regarding the icons and other components of the user interface. User directs input to the appropriate application from the keyboard, mouse, and other input sources.
- GDI.EXE controls the Graphics Device Interface, which executes graphics operations that create images on the system display and other devices.

Drivers, Fonts, and International Support Files

Driver Files

Drivers make device independence possible for Windows applications, providing the hardware-specific interface between the physical devices and Windows. Setup can install several kinds of drivers for Windows, such as:

Comm drivers	Mouse drivers	Printer drivers
Display drivers	Multimedia drivers	Sound drivers
Keyboard drivers	Network drivers	System drivers

The network, multimedia, and printer drivers are optional. Also, drivers can be installed to support virtual machines in 386 enhanced mode, as described in "Files for 386 Enhanced Mode" later in this chapter.

System Driver Files

The system driver provides support for the system timer, information about system disks, and access to OEM-defined system hooks. There are two system drivers shipped with Windows:

- SYSTEM.DRV, the driver for most hardware systems
- HPSYSTEM.DRV, the HP Vectra system driver for standard mode

Keyboard Driver Files

The keyboard drivers shipped with Windows support keyboard input:

- KEYBOARD.DRV for standard keyboards, installed by default
- KBDHP.DRV for all Hewlett-Packard machines
- KBDMOUSE.DRV, the Olivetti/AT&T keyboard mouse driver

The keyboard driver is a standard driver for all systems worldwide. Windows can also handle international keyboards, including foreign symbols, by using the keyboard tables to refer to a language library.

Keyboard table	*Language library*
KBDBE.DLL	Belgian keyboard
KBDCA.DLL	French-Canadian keyboard
KBDDA.DLL	Danish keyboard
KBDDV.DLL	U.S.-Dvorak keyboard
KBDFC.DLL	Canadian multilingual keyboard
KBDFI.DLL	Finnish keyboard
KBDFR.DLL	French keyboard
KBDGR.DLL	German keyboard
KBDIC.DLL	Icelandic keyboard
KBDIT.DLL	Italian keyboard
KBDLA.DLL	Latin American keyboard
KBDNE.DLL	Dutch keyboard
KBDNO.DLL	Norwegian keyboard
KBDPO.DLL	Portuguese keyboard
KBDSF.DLL	Swiss-French keyboard
KBDSG.DLL	Swiss-German keyboard
KBDSP.DLL	Spanish keyboard
KBDSW.DLL	Swedish keyboard
KBDUK.DLL	British keyboard
KBDUS.DLL	U.S. keyboard
KBDUSX.DLL	U.S.-International keyboard

The .DLL filename extension indicates that the file is a dynamic-link library.

Mouse Driver Files

The mouse drivers shipped with Windows support pointing devices for use with Windows and Windows applications.

Driver	Supported mouse or pointing device
HPMOUSE.DRV	Hewlett-Packard mouse (HP-HIL)
KBDMOUSE.DRV	Olivetti/AT&T keyboard mouse
LMOUSE.DRV	Logitech Serial mouse
MSC3BC2.DRV	Mouse Systems COM2/3 button mouse
MSCMOUSE.DRV	Mouse Systems Serial/Bus mouse
MOUSE.DRV	Logitech Bus or PS/2 style, Microsoft, or IBM PS/2 mouse
NOMOUSE.DRV	No mouse attached to system

For information about the related MS-DOS mouse drivers, see "MS-DOS Support Components of Windows" later in this chapter.

Display Driver Files

The display drivers shipped with Windows support the system display and the cursor for the pointing device. The display driver, however, does not support non-Windows applications running in full screen, because such applications write directly to video.

Driver	Supported display adapter
8514.DR	8514/a
EGA.DRV	EGA
EGAHIBW.DRV	EGA with 128K RAM
EGAMONO.DRV	EGA monochrome
HERCULES.DRV	Hercules monochrome
OLIBW.DRV	Olivetti/AT&T monochrome or PVC display
PLASMA.DRV	Compaq Portable plasma
SUPERVGA.DRV	Super VGA (800x600 - 16 colors)
TIGA.DRV	TIGA
VGA.DRV	VGA
VGAMONO.DRV	VGA monochrome, MCGA
V7VGA	Video Seven VGA with 512K (FastWrite, VRAM, 1024i, and compatibles)
XGA.DRV	XGA

Other Driver Files

The communications driver, COMM.DRV, supports serial and parallel device communications.

The Advanced Power Management device driver, POWER.DRV, supports the power management features of laptop and notebook PCs.

Printer Driver Files

Printer drivers support output to the printer device. Some of the printer drivers shipped with Windows have a soft font installation utility. The related files also include help files for the printer drivers and soft font installers. In Windows 3.1, many of the dot-matrix drivers have been replaced by a universal printer driver. Other drivers have been updated for performance and to support TrueType fonts.

Printer driver	*Representative printer*
CANON10E.DRV	Canon Bubble-Jet BJ-10e
CANON130.DRV	Canon Bubble-Jet BJ-130e
CANON330.DRV	Canon Bubble-Jet BJ-300/330
CIT24US.DRV	Citizen 24-pin
CIT9US.DRV	Citizen 9-pin
CITOH.DRV	C-Itoh 8510 or AT&T 470/475
DICONIX.DRV	Kodak Diconix
DM309.DRV	Olivetti DM 309
DMCOLOR.DLL	Universal color printing support libary
EPSON24.DRV	Epson 24-pin
EPSON9.DRV	Epson 9-pin
ESCP2.DRV	Epson ESCP2 dot matrix
EXECJET.DRV	IBM ExecJet
FUJI24.DRV	Fujitsu 24-pin
FUJI9.DRV	Fujitsu 9-pin
GENDRV.DLL	Generic library
HPDSKJET.DRV	Hewlett-Packard DeskJet Series
HPPCL.DRV	HP LaserJet II Series
HPPCL5A.DRV	HP LaserJet III Series (HPPCL5A.HLP and HPPCL5OP.HLP are the help files)
HPPLOT.DRV	HP Plotter
IBM4019.DRV	IBM Laser Printer 4019
IBM5204.DRV	IBM Quickwriter 5204

Printer driver	*Representative printer* *(continued)*
IBMCOLOR.DRV	IBM Color
LBPII.DRV	Canon LBP-8 II
LBPIII.DRV	Canon LBPIII

NEC24PIN.DRV	NEC 24-pin
OKI24.DRV	Okidata 24-pin
OKI9.DRV	Okidata 9-pin
OKI9IBM.DRV	Okidata 9-Pin IBM Model
PAINTJET.DRV	HP PaintJet
PANSON24.DRV	Panasonic 24-pin
PANSON9.DRV	Panasonic 9-pin
PG306.DRV	PG 306
PROPRINT.DRV	IBM Pro series
PROPRN24.DRV	IBM Pro 24 pin series
PS1.DRV	IBM PS/1
PSCRIPT.DRV	Postscript (PSCRIPT.HLP is the help file)
QWIII.DRV	IBM QuietWriter III
THINKJET.DRV	HP ThinkJet (2225 C-D)
TI850.DRV	TI 850/855
TOSHIBA.DRV	Toshiba p351/1351
TTY.DRV	Generic / Text only (TTY.HLP is the help file)
UNIDRV.DLL	Microsoft universal library (UNIDRV.HLP is the help file)

The following files are soft font installers for specific printers.

Soft font installer	*Related printer*
CAN_ADF.EXE	Canon LBP-8 II or LBPIII
SF4019.EXE	IBM Laser Printer 4019
SFINST.EXE	PG 306
FINSTALL.DLL	HPPCL5/A (FINSTALL.HLP is the help file)

The following files provide additional PostScript description information for specific printers.

PostScript description	*Related printer*
40291730.WPD	IBM LaserPrinter 4029 (17 fonts)
40293930.WPD	IBM LaserPrinter 4029 (39 fonts)
EPL75523.WPD	Epson EPL-7500
HERMES_1.WPD	Hermes H 606 PS (13 Fonts)
HERMES_2.WPD	Hermes H 606 PS (35 Fonts)
HPELI523.WPD	HP LaserJet IIISi PostScript
HPIID522.WPD	HP LaserJet IID PostScript
HPIII522.WPD	HP LaserJet III PostScript
HPIIP522.WPD	HP LaserJet IIP PostScript

PostScript description	*Related printer*	*(continued)*
HP_3D522.WPD	HP LaserJet IIID PostScript	
HP_3P522.WPD	HP LaserJet IIIP PostScript	
IBM17521.WPD	IBM 4019 (17 fonts)	
IBM39521.WPD	IBM 4019 (39 fonts)	
MT_TI101.WPD	Microtek TrueLaser	
N2090522.WPD	NEC Silentwriter2 90	

N2290520.WPD	NEC Silentwriter2 290
N2990523.WPD	NEC Silentwriter2 990
OL840518.WPD	Oki OL840/PS
Q2200510.WPD	QMS-PS 2200
Q820_517.WPD	QMS-PS 820
SEIKO_04.WPD	Seiko ColorPoint PS Model 04
OLIVETI1.WPD	Olivetti PG 306 PS (13 fonts)
P4455514.WPD	Panasonic KX-P4455
TRIUMPH1.WPD	Triumph Adler SDR 7706 PS (13 fonts)
N890X505.WPD	NEC Silentwriter LC890XL
N890_470.WPD	NEC Silentwriter LC890
O5241503.WPD	OceColor G5241 PS
O5242503.WPD	OceColor G5242 PS
OLIVETI2.WPD	Olivetti PG 306 PS (35 fonts)
PHIIPX.WPD	Phaser II PX
SEIKO_14.WPD	Seiko ColorPoint PS Model 14
TIM17521.WPD	TI microLaser PS17
TRIUMPH2.WPD	Triumph Adler SDR 7706 PS
U9415470.WPD	Unisys AP9415
TIM35521.WPD	TI microLaser PS35
TKPHZR21.WPD	Phaser II PX I
TKPHZR31.WPD	Phaser III PX I
DEC1150.WPD	Digital DEClaser 1150
DEC2150.WPD	Digital DEClaser 2150
DEC2250.WPD	Digital DEClaser 2250
DEC3250.WPD	Digital DEClaser 3250
DECCOLOR.WPD	Digital ColorMate PS
DECLPS20.WPD	Digital LPS Print Server
NCM40519.WPD	NEC Colormate PS/40
NCM80519.WPD	NEC Colormate PS/80
L200230&.WPD	Linotronic 200/230
L330_52&.WPD	Linotronic 330
L530_52&.WPD	Linotronic 530
L630_52&.WPD	Linotronic 630

Network Driver Files

The network drivers provide a network interface to the Windows File Manager, Control Panel, Print Manager, and system utilities.

Driver	Support file	Supported network
LANMAN.DRV	LANMAN.HLP	Microsoft LAN Manager 2.0 Extended (and 100% compatible)
	LANMAN.HLP	Microsoft LAN Manager 2.0 driver help
	NETAPI20.DLL	Microsoft LAN Manager API library
	PMSPL20.DLL	Microsoft LAN Manager printer API library
MSNET.DRV		Generic network driver*
PCSA.DRV		DEC Pathworks network driver
NETWARE.DRV	NETWARE.HLP	Novell NetWare 2.10 or above; Novell NetWare386
	NWPOPUP.EXE	Supports pop-up messages
	NETX.COM	Workstation shell
	IPX.OBJ	Workstation comm driver (dedicated)
	IPXODI.COM	Workstation comm driver (ODI model)
	LSL.COM	Workstation link support layer (ODI)
	TBMI2.COM	Workstation task switch support (IPX/SPX)

* MSNET.DRV supports 3Com 3+Share, 3Com 3+Open LAN Manager (XMS only), Banyan VINES 4.0, Microsoft LAN Manager 1.x (and compatibles), Microsoft LAN Manager 2.0 Basic (and compatibles), Microsoft Network (and compatibles), and IBM PC LAN Program.

For a list of the supporting virtual device files, see "Files for 386 Enhanced Mode" later in this chapter. For information about networks, see Chapter 12, "Networks and Windows 3.1."

Multimedia Driver Files



The following drivers support the multimedia capabilities of Windows 3.1.

Filename	Purpose
MCICDA.DRV	MCI CD-audio driver
MCISEQ.DRV	MCI driver for MIDI driver
MCIWAVE.DRV	MCI driver for waveform audio
MIDIMAP.DRV	Driver for MIDI Mapper Control Panel extension
MPU401.DRV	MIDI driver for MPU401 compatibles
MMSOUND.DRV	Multimedia sound driver
MSADLIB.DRV	MIDI driver for Adlib compatibles
SNDBLST.DRV	SoundBlaster 1.5 DSP driver
SNDBLST2.DRV	SoundBlaster 2.0 DSP driver
TIMER.DRV	Multimedia timer driver

Font Files

Windows has several fonts for supporting the Windows system and Windows applications, and for non-Windows applications running in Windows and data copied to the Clipboard from those applications. For details about Windows fonts, see Chapter 9, "Fonts."

Font files usually have a .TTF, .FON, or .FOT filename extension.

System Font Files

Three basic types of fonts are installed to support display and output devices:

- **System** is a proportional font used by default to draw menus, dialog box controls, and other text in Windows 3.x.
- **Fixed** is a fixed-width font used in Windows 2.x and earlier versions as the system font (for menus and dialog boxes).
- **OEM font**, or Terminal, is a fixed-width font used to display the OEM text in the Windows Clipboard Viewer. The OEM font also provides an OEM character set used by some Windows applications.

The system, fixed, and OEM fonts that are shipped with Windows 3.1 are listed in the following tables.

System font file	*Supported display resolution*
8514SYS.FON	8514/a (1024x768) resolution system font
EGASYS.FON	EGA (640x350) resolution system font
VGASYS.FON	VGA (640x480) resolution system font

Fixed font file	*Supported display resolution*
8514FIX.FON	8514/a (1024x768) resolution fixed system font
EGAFIX.FON	EGA (640x350) resolution fixed system font
EGAFIX.FON	AT&T (640x400) resolution fixed system font
VGAFIX.FON	VGA (640x480) resolution fixed system font

OEM font file	Supported display resolution
8514OEM.FON	8514/a (1024x768) resolution Terminal font (U.S./Europe)
EGAOEM.FON	EGA (640x350) resolution Terminal font (U.S./Europe)
EGAOEM.FON	AT&T (640x400) resolution Terminal font (U.S./Europe)
VGAOEM.FON	VGA (640x480) resolution Terminal font (U.S./Europe)

Raster Font Files

Six resolutions of raster screen fonts are shipped with Windows. If used for printing, raster fonts print text and graphics as bitmaps or raster lines. The resolutions are identified by a letter appended to the filename of the font as described in the following table.

Letter	Output device	Resolution	x size*	y size*
A**	CGA display	2:1	96	48
B	EGA display	1.33:1	96	72
C**	Printer	1:1.2	60	72
D**	Printer	1.66:1	120	72
E	VGA display	1:1	96	96
F	8514 display	1:1	120	120

* *x,y* indicates the height/width aspect ratio, in pixels per inch.
** These fonts are not included on the Windows 3.1 installation disks.

By appending the letter that identifies the resolution to the raster font filenames in the following table, you can see the files that Windows installs for a given display or printer. For example, the files for the 8514 raster fonts are COURF.FON, SSERIFF.FON, SERIFF.FON, SMALLF.FON, and SYMBOLF.FON.

Font	Filename	Character set	Font description
Courier	COURx.FON	ANSI	Fixed-width with serifs
MS Sans Serif	SSERIFx.FON	ANSI	Sans serif proportional-width
MS Serif	SERIFx.FON	ANSI	Serif proportional-width
Small	SMALLx.FON	ANSI	Proportional small size
Symbol	SYMBOLx.FON	Symbol	Math symbols

Vector Font Files

Windows provides these vector font files: ROMAN.FON, SCRIPT.FON, and MODERN.FON. For vector fonts, characters are stored as sets of relative coordinate pair points with connecting lines. Vector fonts are fully scalable fonts, so the font can be created in any size desired, although applications or printing devices might have limits on the font sizes they support.

TrueType Font Files



The TrueType downloadable fonts shipped with Windows 3.1 support the Arial, Courier, Symbol, and Times New Roman font families. Each family requires two files, a .TTF file and an .FOT file.

TrueType filenames	Font name
ARIAL.FOT, ARIAL.TTF	Arial
ARIALBD.FOT, ARIALBD.TTF	Arial Bold
ARIALBI.FOT, ARIALBI.TTF	Arial Bold Italic
ARIALI.FOT, ARIALI.TTF	Arial Italic
COUR.FOT, COUR.TTF	Courier
COURBD.FOT, COURBD.TTF	Courier Bold
COURBI.FOT, COURBI.TTF	Courier Bold Italic
COURI.FOT, COURI.TTF	Courier Italic
TIMES.FOT, TIMES.TTF	Times New Roman
TIMESBD.FOT, TIMESBD.TTF	Times New Roman Bold
TIMESBI.FOT, TIMESBI.TTF	Times New Roman Bold Italic
TIMESI.FOT, TIMESI.TTF	Times New Roman Italic
SYMBOL.FOT, SYMBOL.TTF	Symbol
WINGDING.FOT, WINGDING.TTF	Wingding

Font Files for Non-Windows Applications

Some fonts are installed for displaying non-Windows applications in a window when Windows is running in 386 enhanced mode. By default, code page 437 (U.S.) fonts are installed. Other font files are included for international language support. These are identified by the code page number appended to the filename.

The following font files are provided with the associated code page translation table files.

Font file	Translation table	Code page	Configuration
APP850.FON		850	U.S., 386 enhanced mode
DOSAPP.FON		437	U.S., 386 enhanced mode
CGA40850.FON	XLAT850.BIN	850	Multilingual
CGA40WOA.FON	-	437	U.S.
CGA80850.FON	XLAT850.BIN	850	Multilingual
CGA80WOA.FON	-	437	U.S.
EGA40850.FON	XLAT850.BIN	850	Multilingual
EGA40WOA.FON	-	437	U.S.
EGA80850.FON	XLAT850.BIN	850	Multilingual
EGA80WOA.FON	-	437	U.S.
HERC850.FON	XLAT850.BIN	850	Multilingual
HERCWOA.FON	-	437	U.S.
VGA850.FON	XLAT850.BIN	850	Multilingual
VGA860.FON	XLAT860.BIN	860	Portuguese
VGA861.FON	XLAT861.BIN	861	Icelandic
VGA863.FON	XLAT863.BIN	863	French Canadian
VGA865.FON	XLAT865.BIN	865	Norwegian/Danish

International Support Files

Windows provides language libraries to support a number of languages.

Filename	Supported languages
LANGDUT.DLL	Dutch language driver
LANGENG.DLL	General International language driver
LANGFRN.DLL	French language driver
LANGGER.DLL	German language driver
LANGSCA.DLL	Finnish/Icelandic/Norwegian/Swedish language driver
LANGSPA.DLL	Spanish language driver

MS-DOS Support Components of Windows

Two kinds of files provide MS-DOS support for Windows: MS-DOS drivers and the grabber files that support data exchange between Windows and non-Windows applications.

MS-DOS Driver Files

Several MS-DOS driver files are included with Windows. The following drivers are the recommended versions to use with Windows 3.1.

Driver	Purpose
EGA.SYS	EGA MS-DOS driver
EMM386.EXE	Microsoft MS-DOS 386 EMS manager
HIMEM.SYS	Microsoft MS-DOS XMS manager
RAMDRIVE.SYS	Microsoft MS-DOS RAMDrive utility
SMARTDRV.EXE	Microsoft MS-DOS SMARTDrive 4.0 disk caching utility
LMOUSE.COM	MS-DOS Level Logitech mouse driver
MOUSE.COM	MS-DOS mouse driver
MOUSE.SYS	MS-DOS mouse driver (installed at MS-DOS boot time)
MOUSEHP.COM	MS-DOS mouse driver for Hewlett-Packard systems
MOUSEHP.SYS	MS-DOS mouse driver for Hewlett-Packard systems

WinOldAp and the Grabber Files

Two primary parts of Windows support non-Windows applications under standard mode Windows: WinOldAp and the grabber. When Windows runs in 386 enhanced mode, the limited resources on the machine are virtualized to provide virtual memory, virtual displays, and virtual communications along with a number of other services. The related files are discussed in "Files for 386 Enhanced Mode" later in this chapter.

WinOldAp and the grabber files support data exchange between non-Windows applications and Windows. The support for non-Windows applications varies, depending on the capabilities of the system CPU and the mode in which Windows is running.

WinOldAp comes in two versions for the two Windows operating modes:

- WINOLDAP.MOD for standard mode
- WINOA386.MOD for 386 enhanced mode

The grabber for your system is specific to the display driver.

The 286 grabbers used for standard mode only support PrintScreen and copying and pasting text between Windows applications and non-Windows applications. The 386 grabbers that support Windows 386 enhanced mode provide the following capabilities:

- Copying text from non-Windows applications
- Displaying data in a windowed virtual machine
- Selecting data in a windowed virtual machine
- Copying graphics to the Windows Clipboard
- PrintScreen

The files that provide font support for the grabbers are listed below, with descriptions of the kinds of display drivers that the grabbers support.

286 grabber support file	*Display device supported*
CGA.2GR	CGA
EGACOLOR.2GR	EGA
EGAMONO.2GR	EGA monochrome
HERCULES.2GR	Hercules monochrome
OLIGRAB.2GR	Olivetti/AT&T PVC
VGACOLOR.2GR	VGA
VGAMONO.2GR	VGA monochrome

386 grabber support file	*Display device supported*
EGA.3GR	EGA 386
HERC.3GR	Hercules monochrome
PLASMA.3GR	Compaq Portable plasma
V7VGA.3GR	Video 7
VGA.3GR	VGA
VGA30.3GR	VGA (version 3.0)
VGADIB.3GR	DIB (8514/a monochrome)

Files for Standard Mode

When Windows is running in standard mode, the processor is switched into 80286 protected mode, allowing access to extended memory through XMS support. The DOSX.EXE file, required for standard mode, is the MS-DOS Extender for Windows. When Windows runs in standard mode, WIN.COM executes DOSX.EXE. Then the Kernel file is loaded (KRNL286.EXE for 80286 machines, or KRNL386.EXE for 80386 machines), which in turn loads the other parts of Windows. Two more files support task swapping for standard mode:

- WSWAP.EXE supports Windows applications in standard mode.
- DSWAP.EXE supports non-Windows applications in standard mode.

Files for 386 Enhanced Mode

In 386 enhanced mode, Windows can use virtual memory. Much of the virtual support is provided by WIN386.EXE, which is executed by WIN.COM. When WIN386.EXE begins to load, it looks for the files identified in the **[386enh]** section of SYSTEM.INI. Some of the standard files are built into WIN386.EXE (designated with the "*" symbol in SYSTEM.INI entries). The other files that WIN386.EXE loads to support virtual devices are listed in the following table.

Filename	*Virtual device supported*
BANINST.386	Banyan VINES 4.0 instancing virtual device
DECNB.386	DEC Pathworks
DECNET.386	
LANMAN10.386	LAN Manager version 1.0 support
HPEBIOS.386	EBIOS virtual device for Hewlett-Packard machines
LVMD.386	Logitech virtual mouse device
MSCVMD.386	Mouse Systems virtual mouse device
V7VDD.386	Video Seven virtual display device
VADLIBD.386	Virtual DMA device for Adlib
VDD8514.386	8514/a virtual display device
VDDCGA.386	CGA virtual display device
VDDCT441.386	82C441 VGA virtual display device
VDDEGA.386	EGA virtual display device
VDDHERC.386	Hercules monochrome virtual display device
VDDTIGA.386	TIGA virtual display device
VDDVGA30.386	VGA virtual display device (version 3.0)
VDDXGA.386	XGA virtual display device
VIPX.386	Novell NetWare virtual IPX support
VNETWARE.386	NetWare virtual support
VPOWERD.386	Advanced Power Management virtual device
VSBD.386	SoundBlaster virtual device
VTDAPI.386	MultiMedia virtual timer device
WIN386.PS2	Support for PS/2 architecture

Windows Applications, Setup, and Other Files

Files for Windows Applications

The Windows files also include applications, shells, utilities, accessories, and games. The following table lists the applications and associated files, with a brief description of each application.

Filename	Associated files	Application name and description
CALC.EXE	CALC.HLP	Calculator (general/scientific)
CALENDAR.EXE	CALENDAR.HLP	Calendar
CARDFILE.EXE	CARDFILE.HLP	Cardfile (desktop Rolodex)
CHARMAP.EXE	CHARMAP.HLP	Character Map
CLIPBRD.EXE	CLIPBRD.HLP	Clipboard Viewer
CLOCK.EXE		Clock (analog/digital)
CONTROL.EXE	CONTROL.HLP	Control Panel
	CONTROL.INI	Initialization file
	CPWIN386.CPL	386 enhanced mode extension for Control Panel
	DRIVERS.CPL	Installable drivers extension for Control Panel
	LZEXPAND.DLL	File expansion utility for Control Panel
	MAIN.CPL	Main Control Panel extension
	MIDIMAP.CFG	MIDI Mapper extension file for Control Panel
	SND.CPL	Sound extension for Control Panel
DRWATSON.EXE		Windows fault detection utility
MPLAYER.EXE	MPLAYER.HLP	Media Player
	MMSYSTEM.DLL	Multimedia system library
	MMTASK.TSK	Multimedia background task
MSD.EXE	MSD.INI	Microsoft Diagnostics utility and initialization file
NOTEPAD.EXE	NOTEPAD.HLP	Notepad (desktop text editor)
PACKAGER.EXE	PACKAGER.HLP	Object Packager
PBRUSH.EXE	PBRUSH.DLL	Paintbrush
	PBRUSH.HLP	
PIFEDIT.EXE	PIFEDIT.HLP	PIF Editor
POWER.HLP	SL.DLL, SL.HLP	Advanced Power Management supporting files
PRINTMAN.EXE	PRINTMAN.HLP	Print Manager (Windows print spooler)
PROGMAN.EXE	PROGMAN.INI	Program Manager (Windows 3.1 shell)
	PROGMAN.HLP	
RECORDER.EXE	RECORDER.HLP	Recorder (desktop macro recorder)
	RECORDER.DLL	
REGEDIT.EXE	REGEDIT.HLP	Registration Editor and supporting files
	REGEDITV.HLP	
	DDEML.DLL	DDE management library
	OLECLI.DLL	Client library and server
	OLESVR.DLL	for Object Linking and Embedding

Filename	Associated files	Application name and description (continued)
SHELL.DLL		Shell library
SOL.EXE	SOL.HLP	Solitaire (most-tested game)
SMARTDRV.EXE		Disk-caching utility
SOUNDREC.EXE	SOUNDREC.HLP	Sound Recorder
SYSEDIT.EXE		Windows System Editor
TASKMAN.EXE		Task Manager (application switcher)
TERMINAL.EXE	TERMINAL.HLP	Terminal (desktop communications)
TOOLHELP.DLL		Windows Tool Helper library
WINFILE.EXE	WINFILE.HLP	File Manager (Windows 3.1 shell)
WINHELP.EXE	WINHELP.HLP GLOSSARY.HLP	Help (Windows help engine) Windows Help glossary
WINMINE.EXE	WINMINE.HLP	MineSweeper (game)
WINTUTOR.EXE	WINTUTOR.DAT	Windows Tutorial
WRITE.EXE	WRITE.HLP	Write (desktop word processor)

Control Panel uses LZEXPAND.DLL to decompress files from the Windows installation disks. Because most of the files on the Windows installation disks are compressed (except SETUP.INF, SETUP.EXE, and EXPAND.EXE), Control Panel must decompress the files to install a new printer or to add fonts. LZEXPAND is a Windows library counterpart to EXPAND.EXE.

Setup-Related Files

Setup has a number of files for its exclusive use. For example, the *.LGO files contain the code for displaying the opening screen logo, and the *.RLE files contain the actual logo bitmap (in Run Length Encoded format). Setup combines the .LGO and .RLE files with the WIN.CNF file to create WIN.COM. Setup also uses the files listed in the following table.

Filename	Purpose
SETUP.SHH	Automated Setup template
SETUP.EXE	Windows Setup application file
SETUP.HLP	Setup Help
SETUP.INF	Setup information file
SETUP.INI	Initialization file for Setup
SETUP.REG	Registration Database template
SETUP.TXT	Windows Readme file
VER.DLL	Version Resource and File Installation library
WINVER	Windows-version utility
XMSMMGR.EXE	Setup XMS Manager
EXPAND.EXE	MS-DOS-based file expansion utility

Filename	Purpose (continued)
Startup logo files:	
CGALOGO.LGO	CGA startup logo code
CGALOGO.RLE	CGA display logo screen
EGALOGO.LGO	EGA display logo screen
EGALOGO.RLE	EGA display logo screen
EGAMONO.LGO	EGA mono startup logo code
EGAMONO.RLE	EGA monochrome logo screen
HERCLOGO.LGO	Hercules mono startup logo code
HERCLOGO.RLE	Hercules display logo screen
VGALOGO.LGO	VGA startup logo code
VGALOGO.RLE	VGA display logo screen
Initialization and information source files:	
APPS.INF	Information file for non-Windows applications
CONTROL.INF	Information file for Control Panel and printer installation
CONTROL.SRC	CONTROL.INI template
PRTUPD.INF	Information for printer driver updates
SYSTEM.SRC	SYSTEM.INI template
WIN.CNF	Windows startup code
WIN.SRC	WIN.INI template

Other Files

These files serve a wide range of functions, including support for PS/2 architectures and README files for general information.

Filename	Purpose
MORICONS.DLL	Icons for non-Windows applications
Bitmaps files for wallpaper:	
256COLOR.BMP	256-color wallpaper
ARCADE.BMP	Arcade wallpaper
ARCHES.BMP	Arches wallpaper
ARGYLE.BMP	Argyle wallpaper
CARS.BMP	Cars wallpaper
CASTLE.BMP	Castle wallpaper
CHITZ.BMP	Chitz wallpaper
EGYPT.BMP	Egypt wallpaper
FLOCK.BMP	Flock wallpaper
HONEY.BMP	Honey wallpaper
LEAVES.BMP	Leaves wallpaper
MARBLE.BMP	Marble wallpaper
REDBRICK.BMP	Redbrick wallpaper
RIVETS.BMP	Rivets wallpaper
SQUARES.BMP	Squares wallpaper

Filename	Purpose (continued)
TARTAN.BMP	Tartan wallpaper
THATCH.BMP	Thatch wallpaper
WINLOGO.BMP	Logo wallpaper
ZIGZAG.BMP	Zigzag wallpaper
Screensaver files:	
SSSTARS.SCR	Stars screen saver
SCRNSAVE.SCR	Generic screen saver
SSMYST.SCR	Mystify screen saver
SSMARQUE.SCR	Marquee screen saver
SSFLYWIN.SCR	Flying Windows
MIDI sound file:	
CANYON.MID	Canyon MIDI sound
Wave-form sound files:	
CHORD.WAV	Question Sound
DING.WAV	Default Beep
CHIMES.WAV	Exit Sound
TADA.WAV	Start Sound
README files:	
NETWORKS.WRI	README file for networks
PRINTERS.WRI	README file for printers
README.WRI	README file
SYSINI.WRI	README file for SYSTEM.INI
WININI.WRI	README file for WIN.INI
Miscellaneous hardware support and other supporting files:	
386MAX.VXD	Qualitas 386MAX virtual device for standard mode
BLUEMAX.VXD	Qualitas BlueMAX virtual device
COMMDLG.DLL	Windows Common Dialogs library
TIGAWIN.RLM	TIGA firmware code
WIN87EM.DLL	80x87 math coprocessor emulation library
WINDOWS.LOD	Qualitas 386MAX/BlueMAX loadable module
TESTPS.TXT	PostScript test text file

Files You Can Delete

Because of the large number of files that come with Windows 3.1, you might want to delete some of the files to free disk space.

Note Do not delete any of these files while Windows is running. Instead, exit Windows, then delete the files from the command prompt.

You can delete these files when Windows is not running without degrading Windows performance:

- EMM386.EXE (expanded memory emulator) if you don't need to provide EMM support for non-Windows applications
- Any files in the TEMP directory
- Any files that start with the characters ~WOA or ~GRB
- Any files named WIN386.SWP (a temporary Windows swap file)

You can choose the Windows Setup icon in Control Panel, then choose Add/Remove Components from the Options menu to remove any of these files from your system:

- Any accessories you do not use (such as Paintbrush, Write, Calendar, Cardfile) with their related .HLP and .DLL files
- Games
- Screen savers
- Wallpapers (.BMP files) and sound files (.WAV files)

For a list of the files for a minimum Windows configuration, see "Minimizing the Windows 'Footprint'" in Appendix C, "Windows 3.1 Disks and Files."

Chapter

4 The Windows Initialization Files

This chapter describes the contents of the key Microsoft Windows initialization files, WIN.INI and SYSTEM.INI, plus other Windows initialization files. This chapter also describes how you can change entries in the files.

Note If you want to reinstall Windows and preserve the settings in the current initialization files on your system, follow the procedures outlined in Flowchart 1.5, "Reinstalling Windows without Losing Settings."

Related information

- *Windows User's Guide:* Chapter 5, "Control Panel"; Chapter 15, "Maintaining Windows with Setup"
- *Windows Resource Kit:* Chapter 2, "The Windows Setup Information Files"

Contents of this chapter

About the Initialization Files 158
Format of .INI Files 158
Changing Entries in .INI Files 159
Editing the .INI Source Files 161
Windows 3.1 .INI File Changes 161
WIN.INI: The Windows Initialization File 163
SYSTEM.INI: The System Initialization File 183
Other Windows Initialization Files 219
PROGMAN.INI: The Program Manager Initialization File 219
CONTROL.INI: Control Panel Initialization File 222
WINFILE.INI: The File Manager Initialization File 223

About the Initialization Files

The Windows initialization files contain information that defines your Windows environment. Microsoft Windows and the Windows applications use the information in these files to configure themselves according to your needs and preferences.

The standard Windows initialization files are CONTROL.INI, PROGMAN.INI, SYSTEM.INI, WIN.INI, and WINFILE.INI. Other Windows applications may also add .INI files in the WINDOWS directory.

.INI File	File contents
WIN.INI	Entries that you can set to alter the Windows environment according to your preferences.
SYSTEM.INI	Entries that you can set to customize Windows to meet your system's hardware needs.
CONTROL.INI	Entries that describe the color schemes and patterns used in Windows, and the settings for printers and installable drivers.
PROGMAN.INI	Entries that define content of program groups.
WINFILE.INI	Entries that define the appearance and behavior of items in File Manager.
application.INI	Entries created and maintained by a specific Windows application. The file is created when the application is installed.

Important Errors made by editing initialization files can lead to undesirable results when you run Windows. Before changing any entry, make a backup copy of the file. Read "Changing Entries in .INI Files" later in this chapter for guidelines.

Format of .INI Files

Windows initialization files are broken into logical groups called sections. Each section has this format:

[*section*]
keyname=value

Item	Meaning
[*section*]	The name of a section. The enclosing brackets ([]) are required, and the left bracket must be in the leftmost column on the screen.
keyname	The name of an entry, which can consist of any combination of letters and digits. For many entries, the *keyname* must be followed immediately by an equal sign (=).
value	An integer, a string, or a quoted string, depending on the entry.

For many entries, the *value* is shown as *Boolean.* To enable an entry that requires a Boolean value, you can enter **True**, **Yes**, **On**, or **1**. To disable such an entry, you can enter **False**, **No**, **Off**, or **0**. Case doesn't matter.

To include comments in .INI files, begin each line of comment with a semicolon (;).

The order of the sections in an .INI file is not important. Some Windows applications may add or insert other sections. The entries in each section are listed alphabetically in this chapter, but they do not appear in this order in the .INI files. So to change an entry, you must search for it within the section.

In this chapter, the items that apppear in bold are actual words included in the entry. Values that must be substituted in the entry are shown in italics. When the value can be one of several choices, the choices are separated with a pipe character (|). For example:

Beep=*Yes* | *No*

Any entry listed here that does not appear in your .INI file has the default value. Default values are shown for all the entries in this chapter.

Changing Entries in .INI Files

Windows creates the initialization files during installation and assigns default values. Some entries are added or changed when you install or configure a Windows application. You can edit these entries to change the appearance or performance of Windows. There are several ways to change entries in the .INI files:

- Use Control Panel, Program Manager, or File Manager to change many entries with menu commands and dialog box options.
- Run Windows Setup again to change system settings, the keyboard or mouse configuration, or network options, and to add or remove printers and fonts.
- Use a text editor such as Notepad to edit the file directly.
- Choose a command such as Printer Setup from the File menu in Print Manager, and specify new options.

The entries in this chapter show which entries in WIN.INI and SYSTEM.INI you can change with Control Panel or Setup to avoid editing the .INI file directly, where it is easy to make editing mistakes. For the entries you cannot change with Control Panel or Setup, you must edit the .INI file with a text editor.

Important Always back up the .INI file before you open it, and use care when making changes. Incorrect changes can lead to unexpected results when you run Windows. Also, some editors can damage characters with ANSI values of greater than 127. We recommend that you use Control Panel or Setup whenever possible to make changes. If you must edit the file, use a text editor such as Notepad, System Editor, or Edit (the MS-DOS 5.0 text editor).

To change .INI file entries with a text editor:

1. ***Important*** Create a backup copy of the .INI file you want to edit.
2. Open the .INI file with a text editor such as Notepad.
3. Edit the specific entries and save the file.
4. Restart Windows so the changes will take effect.

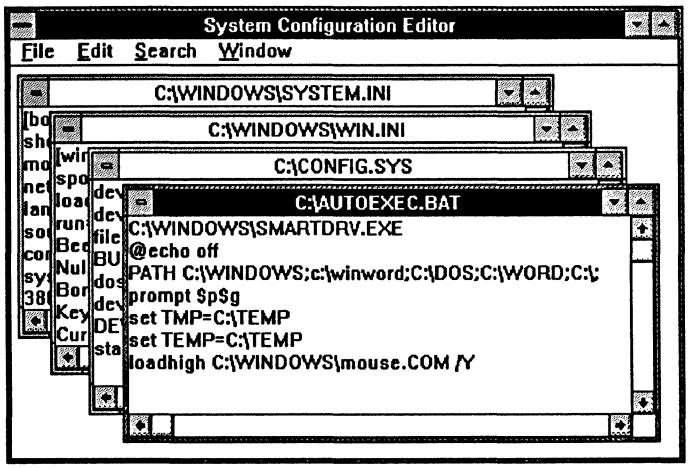
You can use the System Editor utility in Windows to edit SYSTEM.INI, WIN.INI, CONFIG.SYS, and AUTOEXEC.BAT at the same time. The System Editor is installed in your WINDOWS directory by Windows Setup.

To edit system files with System Editor:

1. ***Important*** Create a backup copy of the .INI file you want to edit.
2. Choose Run from the File Menu in Program Manager. Type **sysedit** and press ENTER.
3. Click the window in System Editor that contains the file you want to edit.

Figure 4.1

The System Editor in Windows 3.1



4. Edit the file, using the same text editing techniques as in Notepad.
5. Save the file, and choose Exit from the File menu to close System Editor.
6. Restart Windows so that the changes you made can take effect. If you edit CONFIG.SYS or AUTOEXEC.BAT, you must reboot your system for the changes to take effect.

Editing the .INI Source Files

If you copy the Windows files to a network server using the **setup /a** option (to install a shared copy of Windows), Setup uses WIN.SRC, SYSTEM.SRC, and CONTROL.SRC to build WIN.INI, SYSTEM. INI, and CONTROL.INI when a user installs Windows on a workstation with the **setup /n** option. To create custom initialization files for multiple installations, you can change the .SRC files.

To edit the .SRC files, first save a backup copy of the original file, then make editing changes in the file, following the guidelines in this chapter. Save the file with an .SRC filename extension in the WINDOWS directory on the network server. And, of course, test the new .INI file on a single system before installing Windows on multiple systems.

Windows 3.1 .INI File Changes



If you update to Windows, Setup automatically updates the .INI files with the following changes. Setup leaves all other sections and entries in your existing WIN.INI and SYSTEM.INI unchanged when you upgrade to Windows 3.1.

For the WIN.INI file:

- In the **[Windows]** section, Setup adds three new entries:

  ```
  KeyboardDelay=
  ScreenSaverTimeout=
  DosPrint=
  ```

  If the following entries are present in the 3.0 version of WIN.INI, Setup does not change them. If they are not present, Setup adds them:

  ```
  device=
  documents=
  programs=com exe bat pif
  ```

- In the **[extensions]** section, Setup adds **bmp=prush.exe ^.bmp**, and also adds **pcx=pbrush.exe ^.pcx** and **rec=recorder.exe ^.rec** if they are not already in WIN.INI:
- In the **[intl]** section, Setup changes **sLanguage=English (American)** to **sLanguage=enu**.
- In the **[ports]** section, Setup adds **LPT1.DOS** and **LPT2.DOS**. If they are not already present in WIN.INI, Setup adds these three entries:

```
file:=
com3:=9600,n,8,1,x
com4:=9600,n,8,1,x
```

- Setup adds **[fontSubstitutes]** and **[TrueType]** to support TrueType fonts. The default **[fontSubstitutes]** entries are:

```
Helv=MS Sans Serif
Tms Rmn=MS Serif
Times=Times New Roman
Helvetica=Arial
```

- Setup adds the **[mci extensions]** section, with these entries to support multimedia:

```
wav=waveaudio
mid=sequenccer
rmi=sequencer
```

- Setup adds the **[sound]** section, with these entries to support multimedia:

```
SystemDefault=ding.wav, Default Beep
SystemExclamation=chord.wav, Exclamation
SystemStart=tada.wav, Windows Start
SystemExit=chimes.wav, Windows Exit
SystemHand=chord.wav, Critical Stop
SystemQuestion=chord.wav, Question
SystemAsterisk=chord.wav, Asterisk
```

- Setup adds the **[embedding]** section, with these entries to support Object Linking and Embedding:

```
SoundRec=Sound,Sound,SoundRec.exe,picture
Package=Package,Package,packager.exe,picture
PBrush=Paintbrush Picture,Paintbrush Picture,pbrush.exe,picture
```

For the SYSTEM.INI file:

- In the **[386enh]** section, Setup adds **device=** entries if it encounters Western Digital–compatible protected mode block devices in your system.
- One new section, **[drivers]**, contains a list of aliases for installable drivers, and another new section, **[mci]**, lists the Media Control Interface drivers.
- Many other changes in communications and enhanced mode capabilities for Windows 3.1 have created new entries in the **[386enh]** section.

WIN.INI: The Windows Initialization File

The WIN.INI file contains several sections, each of which consists of a group of related entries that you can use to customize the Windows environment. A specific WIN.INI file might not have all of these sections, or it might have additional sections, depending on your system's hardware and software requirements. For example, many Windows applications add entries in WIN.INI to define user preferences and other items.



For Windows 3.1, the size of WIN.INI is no longer limited to 32K.

These basic sections can appear in WIN.INI after you first install Windows.

Section	Purpose
[Windows]	Affects several elements of the Windows environment.
[desktop]	Controls the appearance of the desktop and the position of windows and icons.
[extensions]	Associates specified types of files with corresponding applications.
[intl]	Describes how to display items for countries other than the United States.
[ports]	Lists all available output ports.
[fonts]	Describes the screen font files that are loaded by Windows.
[fontSubstitutes]	Lists pairs of fonts that are recognized by Windows as interchangeable.
[TrueType]	Describes options for using and displaying TrueType fonts.
[mci extensions]	Associates specified types of files with Media Control Interface devices.
[network]	Describes network settings and previous network connections.
[embedding]	Lists the server objects used in Object Linking and Embedding (OLE).
[Windows Help]	Lists settings used to specify the default size, placement, and text colors of the Help window and dialog boxes.
[sound]	Lists the sound files assigned to each system event.
[printerPorts]	Lists active and inactive output devices to be accessed by Windows.
[devices]	Lists active output devices that provide compatibility with earlier versions of Windows applications.
[programs]	Lists additional paths that Windows will search to find a program file when you try to open an associated data file.
[colors]	Defines colors for the Windows display.

More sections can be inserted in WIN.INI by other Windows applications, although some applications set up their own .INI files. For information about those sections, see the application's documentation.

[Windows] Section

This section contains entries that affect these parts of the Windows environment:

- Applications that start when you start Windows
- Warning beep
- Printing
- Window border width
- Keyboard speed
- Mouse settings
- Definition of files as documents or programs

The **[Windows]** section can contain the following entries:

Beep=*Yes* | *No*

This entry specifies whether Windows sounds a warning beep when you attempt to do something that is not allowed. The default is **Yes**. To change this entry, choose the Sound icon from Control Panel. For entries that control multimedia sound, see the description of **[sound]** section later in this chapter.

BorderWidth=*number*

This entry sets the width of the borders around all the windows that have sizeable borders. The possible range is **1** (narrowest) to **49** (widest). The default is **3**. To change this entry, choose the Desktop icon from Control Panel.



CoolSwitch=*0* | *1*

This entry turns fast task switching on or off. The default is **1**. To change this entry, choose the Desktop icon from Control Panel and check or clear the Fast ALT+TAB Switching option.

CursorBlinkRate=*milliseconds*

This entry indicates how much time elapses between each blink of the selection cursor. The default is **530**. To change this entry, choose the Desktop icon from Control Panel.



DefaultQueueSize=*number*

This entry specifies the number of messages an application's message queue can hold. The default is **8**. To change this entry, you must edit WIN.INI. (You should never need to change this entry.)

Device=*output-device-name, device-driver, port-connection*

This entry defines the default printer. The *output-device-name* can be any device name listed in the **[devices]** section. An explicit port and driver must be assigned to the device. The *device-driver* is the filename (without the extension) of the device driver file. The *port-connection* is any port in the **[ports]** section. The default is none. To change this entry, choose the Printers icon from Control Panel or choose Printer Setup from the Option menu in Print Manager.

DeviceNotSelectedTimeout=*seconds*

This entry specifies how much time Windows waits for a device to be switched on. An entry is saved for each printer listed in the **[printerPorts]** section. If the device is not switched on during this time, Windows won't print to the device. For some devices, Windows immediately posts an error message if the device is not already switched on. The default is **15**. To change this entry, you must edit WIN.INI. This entry only defines the system default value. To change the value for a particular printer, choose the Printers icon from Control Panel, which changes the corresponding value only in the **[printerPorts]** section.



Documents=*extensions*

This entry defines files to be considered "documents" by Windows. Use this entry to define only document file extensions not listed in the **[extensions]** section, because those extensions are automatically considered documents. The extensions listed in this entry are not associated with any application. Separate the filename extensions with a space, and do not include the preceding periods. The default is none. To change this entry, you must edit WIN.INI.



DosPrint=*Yes* | *No*

This entry specifies whether Windows should use MS-DOS interrupts when printing. When this entry is **Yes**, MS-DOS interrupts are used; if the value is **No**, printing output is sent directly to the port that the printer is assigned to. The default is **No**. To change this entry, choose the Printers icon from Control Panel or the Printer Setup command from the Print Manager Options menu, then clear the Fast Printing Direct To Port option in the Connect dialog box.



DoubleClickHeight=*pixels*
DoubleClickWidth=*pixels*

These entries specify the height and width (in pixels) that the mouse pointer can move between clicks in a double-click. If the mouse pointer is moved farther up or down (or left or right), the two clicks will be treated as separate single clicks. For example, if **DoubleClickHeight=10**, the mouse pointer cannot move vertically more than 5 pixels between clicks for the two clicks to be recognized as a double-click. If you enter an odd number, it is rounded to the nearest even number. The default is **4** for both entries. To change these entries, you must edit WIN.INI.

DoubleClickSpeed=*milliseconds*

This entry sets the maximum time between clicks of the mouse button that the system will permit for one double-click. The lower the value for this entry, the less time you have to click twice to double-click. The default is **452**. To change this entry, choose the Mouse icon from Control Panel.



KeyboardDelay=*milliseconds*

This entry establishes how much time elapses after you hold down a key before the key starts to repeat. The default is **2**. To change this entry, choose the Keyboard icon from Control Panel.

KeyboardSpeed=*milliseconds*

This entry sets how much time elapses between repetitions of a character on the display when you hold down a keyboard key. This entry does not work for all keyboards. The default is **31**. To change this entry, choose the Keyboard icon from Control Panel. If you delete this entry from WIN.INI, Windows Setup will not reset the speed, but the speed will still be set if you choose the Keyboard icon in Control Panel, even if you cancel the dialog box.

Load=*filename(s)*

This entry specifies the applications to be run as icons when Windows is started. This entry is a list of application filenames, or documents associated with an application, with each filename separated by a space. Make sure to specify the path if the file is not located in the WINDOWS directory. The default is none. To change this entry, add the application to the Startup group in Program Manager, then check Minimize On Use in the Properties dialog box.



MenuDropAlignment=*0* | *1*

This entry specifies whether menus open so that they are right aligned or left aligned with the menu title. If this entry is **0**, menus open so that they are left aligned. If it is **1**, menus open so that they are right aligned. The default is **0**. To change this entry, you must edit WIN.INI.



MenuShowDelay=*milliseconds*

This entry specifies how long to wait before displaying a cascading menu. The default is **0** for 80386 computers; **400** for 80286 computers. To change this entry, you must edit WIN.INI.

MouseSpeed=*0* | *1* | *2*

This entry sets the relationship between mouse and cursor movement when the value of either **MouseThreshold1=** or **MouseThreshold2=** is exceeded. When this occurs, Windows causes cursor movement to accelerate according to the value of **MouseSpeed=**. If the value is **0**, there is no acceleration. If the value is **1**, the cursor is moved twice the normal speed when mouse movement exceeds the value of **MouseThreshold1=**. If the value is **2**, the cursor is moved twice the normal speed when the mouse movement exceeds the value of **MouseThreshold1=**, or four times the normal speed if mouse movement exceeds **MouseThreshold2=**. The default is **1**. To change this entry, choose the Mouse icon from Control Panel.

MouseThreshold1=*pixels*
MouseThreshold2=*pixels*

These entries set the maximum number of pixels that the mouse can move between mouse interrupts before Windows alters the relationship between mouse and cursor movement. If the mouse movement exceeds the threshold defined by **MouseThreshold1=** and if **MouseSpeed=** is greater than **0**, Windows moves the cursor at twice the normal speed. If the mouse movement exceeds the threshold defined by **MouseThreshold2=** and if **MouseSpeed=2**, Windows moves the cursor at four times the normal speed. The defaults are **MouseThreshold1=5** and **MouseThreshold2=10**. To change these entries, choose the Mouse icon from Control Panel.



MouseTrails=*number*

This entry specifies the number of pointers shown on the screen when the MouseTrails option is selected in the Mouse dialog box in Control Panel. You can specify a number between **1** and **7**. This entry is supported only if you are using the EGA, VGA, or SuperVGA display driver. The default is none. To change this entry, choose the Mouse icon from Control Panel. If the Mouse dialog box does not include the Mouse Trails option, you need to add this entry to WIN.INI to set mouse trails.

NetWarn=*0* | *1*

This entry specifies whether Windows should display a warning message if the system is configured to run a network and the network is not running or the wrong network is running. All Windows network-related options will be disabled if the network is disabled or incorrect. Setting this value to **0** turns off the warning message. The default is **1**. To change this entry, choose the Network icon from Control Panel.

NullPort=*string*

This entry specifies the name used for a null port. This name appears in the Printers Connect dialog box in Control Panel when a device is installed (that is, the device driver is present) but is not connected to any port. This value also appears for such a device in the **[devices]** section of WIN.INI. The default is **"None"**. To change this entry, you must edit WIN.INI.

Programs=*extensions*

This entry defines which files Windows regards as applications. Separate the filename extensions with a space and do not include the preceding periods. The default is **com exe bat pif**. To change this entry, you must edit WIN.INI.

Run=*filename(s)*

This entry tells Windows to run the specified applications when Windows is started. The value is a list of application filenames or documents associated with applications, with each filename separated by a space. Make sure you specify the complete path if the file is not in the WINDOWS directory. The default is none. To change this entry, add the application to the Startup group in Program Manager.



ScreenSaveActive=*0* | *1*

This entry specifies whether a screen saver should be displayed if Windows is not actively being used. Set this value to **1** to use a screen saver; **0** turns off the screen saver. The default is **0**. To change this entry, choose the Desktop icon from Control Panel.



ScreenSaveTimeOut=*seconds*

This entry specifies the amount of time that Windows must be idle before the screen saver appears. The default is **120**. To change this entry, choose the Desktop icon from Control Panel.

Spooler=*Yes* | *No*

This entry specifies whether output to the printer is to be sent through Print Manager. Changing this value to **No** turns off Print Manager. The default is **Yes**. To change this entry, choose the Printers icon from Control Panel and check or clear the Use Print Manager option.



SwapMouseButtons=*0 | 1*

This entry specifies whether to swap the right and left mouse buttons. If the value is **1**, the buttons are swapped. The default is **0**. To change this entry, choose the Mouse icon from Control Panel.

TransmissionRetryTimeout=*seconds*

This entry specifies the default amount of time to give Windows for attempted transmission retries. A value is saved for each printer in the **[printerPorts]** section. If a successful transmission does not occur during this time, Print Manager displays a message stating that the printer is not receiving characters. The default is **90** for PostScript printers; **45** for all other printers. To change this entry, you must edit WIN.INI. This setting serves only as the system default value. To change the value for a particular printer, choose the Printers icon from Control Panel or choose Printer Setup from the Options menu in Print Manager. Changing this setting with Control Panel or Print Manager changes the corresponding value only in the **[printerPorts]** section.

[desktop] Section

The **[desktop]** section contains entries that control the appearance of the screen background and the position of windows and icons on the screen. This section can contain the following entries:

GridGranularity=*number*

This entry specifies the size of the grid that Windows uses to position windows on the screen. The possible range is **0** through **49**, in units of 8 pixels. The default is **0**. To change this entry, choose the Desktop icon from Control Panel.

IconSpacing=*pixels*

This entry specifies the number of pixels that appear horizontally between icons. A larger number increases the space between icons. The default is **77**. To change this entry, choose the Desktop icon from Control Panel.



IconTitleFaceName=*fontname*

This entry specifies the font used to display icon titles. Change this value if the icon title is difficult to read. The default is **MS Sans Serif**. To change this entry, you must edit WIN.INI.



IconTitleSize=*number*

This entry specifies the size of the font used to display icon titles. Change this value if the icon title is difficult to read. The default is **8**. To change this entry, you must edit WIN.INI.



IconTitleWrap=*0 | 1*

This entry specifies whether to wrap icon titles. A value of **1** allows icon title wrapping and increases icon vertical spacing by three lines; **0** turns off icon title wrapping. The default is **1**. To change this entry, choose the Desktop icon from Control Panel.



IconVerticalSpacing=*pixels*

This entry specifies the number of pixels that will appear vertically between icons. Increasing this number increases the amount of vertical space between icons. Windows determines the default value based on the icon title font and the display you are using. To change this entry, you must edit WIN.INI.



Pattern=*b1 b2 b3 b4 b5 b6 b7 b8*

This entry specifies a pattern for the screen background. The 8 numeric values define a bitmap 8 pixels wide and 8 pixels high. Each decimal value represents a byte, and each byte represents a row of 8 pixels, where **0** sets the corresponding pixel to the background color and **1** sets the corresponding pixel to the foreground color (specified by the **Background=** and **WindowText=** entries in the **[colors]** section, respectively). For example, if you set the *b1* value to the decimal value 175, the top row of pixels in the bitmap appears as the binary equivalent (10101111). The default is **"(None)"** (this string appears when no pattern is specified). To change this entry, choose the Desktop icon from Control Panel.

TileWallpaper=*0 | 1*

This entry specifies that the desktop wallpaper is tiled across the screen if this value is **1**. The wallpaper is centered if this value is **0**. The default is **0**. To change this entry, choose the Desktop icon from Control Panel.

Wallpaper=*bitmap-filename*

This entry supplies the filename for the bitmap on the screen background. Include the path if the file is not in the WINDOWS directory or SYSTEM subdirectory. The default is **"(None)"** (this string appears when no pattern is specified). To change this entry, choose the Desktop icon from Control Panel.

[extensions] Section

The **[extensions]** section contains entries that identify document files with corresponding command lines, so that opening a document file automatically starts the application. The format for each entry is:

extension=command-line

Value	Meaning
extension	A filename extension of one to three characters.
command-line	An ordinary MS-DOS command line. The command line always begins with an application filename (including the .EXE extension). This can be followed by any command parameters the application needs, usually including the document filename. The caret (^) character can be used when the original document filename is needed in the command line; the caret is replaced by the document filename without any extension.

The most typical **[extensions]** entries simply list the application name and the document name; for example:

```
CAL=CALENDAR.EXE ^.CAL
```

If you run a file called MYFILE.CAL after this association has been made, the Calendar program starts and automatically opens the file MYFILE.CAL.

You can enter only one line per extension (.TXT, .INI, and others), but more than one line for any program (for example, separate lines to indicate that Write should be used for .TXT, .INI, and .WRI files). To change an entry in **[extensions]**, choose Associate from the File menu in File Manager. Windows also updates this entry from any changes made in the Registration Info Editor. If an entry in **[extensions]** duplicates information in the registration database, File Manager uses the information from the database.

[intl] Section

The **[intl]** section describes how to display dates, times, currency amounts, and other items for countries other than the United States. The default values reflect U.S. options. To change any entry in this section, choose the International icon from Control Panel. For details about the specific values for individual entries, see the WININI.WRI file in your WINDOWS directory.

The **[intl]** section can contain the entries shown in the following table. If the keyword begins with an "i" then the value is an integer; if the keyword begins with "s" then the value is a string.

Entry	Value
iCountry=*country*	Specifies the country code. This number matches the country's international telephone code, except for Canada, which is 2. The default is **1**.
iCurrDigits=*number*	Specifies the number of digits to put after the decimal separator in currency. The default is **2**.
iCurrency=*number*	Specifies a currency format, where **0** = $2, **1** = 2$, **2** = $ 2, and **3** = 2. The default is **0**. The actual currency symbol is specified by the **sCurrency=** value.
iDate=*number*	Specifies a numerical date format for compatibility with Windows 2.x, where **0** = 12/31/90, **1** = 31/12/90, and **2** = 90/12/31. The default is **0**. The actual date divider is specified by the **sShortDate=** value.
iDigits=*number*	Specifies the number of digits to display after the decimal separator in numbers. The default is **2**.
iLZero=*0 | 1*	Specifies whether to put leading zeros in decimal numbers, where **0** = .7 and **1** = 0.7. The default is **0**. The actual decimal separator is specified by the **sDecimal=** value.
iMeasure=*0 | 1*	Specifies the measurement system as metric or English, where **0** = metric and **1** = English. The default is **1**.
iNegCurr=*number*	Specifies a negative number format, where: **0** = ($1) **3** = $1- **6** = 1-$ **9** = -$ 1 **1** = -$1 **4** = (1$) **7** = 1$- **0** = $ 1- **2** = $-1 **5** = -1$ **8** = -1 $ The default is **0**. The actual currency symbol is specified by the **sCurrency=** value.
iTime =*number*	Specifies whether to format time using a 12-hour or 24-hour clock, where **0** = 1:00 (12-hour clock) and **1** = 13:00 (24-hour clock). The default is **0**. The actual time separator is specified by the **sTime=** value.
iTLZero=*number*	Specifies whether to put leading zeros in time, where **0** = 9:15 and **1** = 09:15. The default is **0**. The actual time separator is specified by the **sTime=** value.

Entry	Value (continued)
s1159=*string*	Specifies the string that follows times before noon in the 12-hour time format. The default is **AM**.
s2359=*string*	Specifies the string that follows times after noon in the 12-hour format or that follows all times in the 24-hour format. The default is **PM**.
sCountry=*string*	Specifies the name of the country whose standard value you want to use. The default is **United States**.
sCurrency=*string*	Specifies the currency symbol you want to use. The default is **$**.
sDecimal=*string*	Specifies the punctuation used to separate the fractional part of a decimal number from the whole number part. The default is **.** (a period).
sLanguage=*string*	Specifies the language you want to work in. Windows applications that provide language specific tasks, such as sorting or spell checking, use this entry. The default is **enu**. Values are: **dan** = Danish **deu** = German **eng** = International English **enu** = U.S. English **esn** = Modern Spanish **esp** = Castilian Spanish **fin** = Finnish **fra** = French **frc** = French Canadian **isl** = Icelandic **ita** = Italian **nld** = Dutch **nor** = Norwegian **ptg** = Portuguese **sve** = Swedish
sLis=*string*	Specifies the character used to separate items in a list. In U.S. English, the most common separator is a comma. The default is **,** (a comma).
sLongDate=*format*	Specifies your choices for the long date formats, including abbreviations for the words and separators. Control Panel will accept only certain *format* combinations. Therefore, you should use Control Panel to change these entries. The default is **dddd, MMMM dd, yyyy** (that is, Friday, June 1, 1990). Values are: **d** = Day (1–31) **dd** = Day (01–31) **ddd** = Day (Mon–Sun) **dddd** = Day (Monday–Sunday) **M** = Month (1–12) **MM** = Month (01–12) **MMM** = Month (Jan–Dec) **MMMM** = Month (January–December) **yy** = Year (00–99) **yyyy** = Year (1900–2040)
sShortDate=*format*	Specifies a choice for the short date format, including abbreviations for the words and separators, according to the list described for **sLongDate=**. Control Panel will accept only certain *format* combinations. Therefore, you should use Control Panel to change this setting. The default is **M/d/yy** (that is, 6/1/90).
sThousand=*string*	Specifies the symbol used to separate thousands. For example, if the value is a comma (,), the number appears as 3,000. The default is **,** (a comma).
sTime=*string*	Specifies the character used to separate the hours, minutes, and seconds in time. For example, if the value is a colon (:), the time appears as 15:29:31. The default is **:** (a colon).

[ports] Section

This section lists the available communications and printer ports, defines default values, and lists files to which printer output can be sent. You can specify up to 10 ports. This is similar to the **mode** command for MS-DOS.

This section can contain one or more entries in this format:

portname:=*baud-rate, parity, word-length, stop-bits* **[[, p]]**

Value	Meaning
portname	The name of an output port as it is recognized by MS-DOS. This can be: **COM***x*:= for a communications port, where *x* represents the port number. **LPT***x*:= for a parallel port, where *x* represents the port number. **EPT:=** for a specific IBM printer. **FILE:=** for a filename. **LPT***x***.DOS=** for a parallel port, where *x* represents the port number. Connecting to LPT2.DOS can bypass Windows handling or print output. Choose this option if you have trouble printing.
baud-rate	Specifies a COM port's baud rate.
parity	Specifies the parity setting for a COM port.
word-length	Specifies the length of a word (in bits) for a COM port.
stop-bits	Specifies the number of stop bits to be used for a COM port.
[[,p]]	Specifies that hardware handshaking is in effect.

Values for LPT, EPT, FILE, and LPT*x*.DOS ports do not take any parameters.

Specify a filename for the *portname* value to direct output to that file. A filename, unlike the other port names, must not be followed by a colon (:). If you specify a value for **FILE:=**, Windows will prompt you to specify an output file each time you print.

To change COM port settings, choose the Ports icon from Control Panel. To change or add other ports, you must edit WIN.INI.

LPT1.DOS=*path*
LPT2.DOS=*path*
LPT3.DOS=*path*
LPT4.DOS=*path*

Flowchart 4.2 Cannot Print on Network

These settings are used (instead of the corresponding LPT ports) to specify the path of the printer you are using. This allows you to print directly to a parallel port. Use these settings to bypass Windows special handling for parallel port output. The default is none. To change this entry, you must edit WIN.INI.

[fonts] Section

The **[fonts]** section describes the screen font files that Windows loads at startup. To change this entry, choose the Fonts icon from Control Panel. This section can contain one or more entries in this format:

font-name=font-filename

Value	Meaning
font-name	The descriptive name of a font.
font-filename	The filename of a file that contains font resources.

Note Adding entries to the **[fonts]** section will not install a font. You must choose the Fonts icon in Control Panel to install new fonts.

[fontSubstitutes] Section

The **[fontSubstitutes]** section describes fonts recognized by Windows as identical to another typeface. This is useful if you want to work on documents that include screen fonts that are not supported by Windows 3.1, but were supported by Windows 3.0.

The **[fontSubstitutes]** section contains one or more occurrences of the following entry:



font-name=*font-name*

These entries specify the font that Windows will use in place of another font, only if that font is not installed on your system. For example, if you are viewing a document that is formatted in Helvetica, but Helvetica is not installed on your system, Windows will use Arial to display the document. To change this entry, you must edit WIN.INI. The default entries for this section are:

```
[fontSubstitutes]
Helv=MS Sans Serif
Tms Rmn=MS Serif
Times=Times New Roman
Helvetica=Arial
```

[TrueType] Section

The **[TrueType]** section describes options that affect the use and display of TrueType fonts in your Windows applications. The **[TrueType]** section can contain the following entries:



OutlineThreshold=*number-of-pels-per-em*

This entry specifies the number of pels-per-em at which Windows will render TrueType fonts as outline fonts instead of as bitmap fonts. Using bitmap fonts is faster, but requires more memory. If your system is low on memory, decrease this value. Do not specify a value over **300**. If you do, you might encounter problems with TrueType fonts. The default is **256**. To change this entry, you must edit WIN.INI.



TTEnable=*0* | *1*

This entry controls whether TrueType fonts are available. Setting this value to **1** makes TrueType fonts available in your Windows applications. Setting this value to **0** turns off TrueType fonts so they are unavailable in applications. The default is **1**. To change this entry, choose the Fonts icon from Control Panel.



TTIfCollisions=*0* | *1*

This entry specifies whether to use TrueType fonts in place of other fonts if both types of fonts are installed on your system and have the same font name. If this value is set to **1**, the TrueType font will be used. For example, Windows provides both a bitmap and a TrueType version of the Symbol font. If you set this value to **0**, Windows will use the TrueType version. The default is **0**. To change this entry, you must edit WIN.INI.



TTOnly=*0* | *1*

This entry specifies whether to make only TrueType fonts available in Windows applications. If this value is set to **1**, only TrueType fonts are available. If this value is set to **0**, all fonts installed on your system are available. The default is **0**. To change this entry, choose the Fonts icon from Control Panel.

[mci extensions] Section



The **[mci extensions]** section contains entries that associate different types of media files with Media Control Interface drivers. Whenever a media file is selected, Windows will use the associated driver to play it. This section can contain one or more occurrences of the following entry:

extension=mcidevice-type

[network] Section

The **[network]** section describes network settings and previous network connections. The **[network]** section can contain the following entries:



drive=*network-server-and-share*

This entry shows the network connections that are restored each time Windows is started. To change this entry, choose the Network Connections command from the Disk menu in File Manager.



InRestoreNetConnect=*0 | 1*

This entry specifies whether to reconnect to the network servers you were connected to in your previous Windows session. This setting works for MS-Net, LAN Manager Basic, and LAN Manager 2.0. If this value is set to **1**, you will be reconnected to the network servers each time you start Windows. If this value is set to **0**, you will not be reconnected. The default is **1**. To change this entry, choose the Networks option from Control Panel, or if the Network dialog box for your network does not include a reconnect option, edit WIN.INI.



port=*network-printer-path*

This entry defines the path for a network printer and the port the printer is assigned to. This connection is restored each time Windows is started. To change this entry, choose the Printers option from Control Panel, or the Network Connections command from the Options menu in Print Manager.

[embedding] Section



The **[embedding]** section lists the Object Linking and Embedding (OLE) objects, their description, the program that is used to create them, and their file format. This information also appears in the Registration Info Editor. We recommend that you use the Registration Info Editor to register applications or edit application information for OLE. The **[embedding]** section is included in WIN.INI only to maintain compatibility with Windows 3.0 and 3.0 applications. The **[embedding]** section can contain one or more entries in the following format to specify the objects used as servers for OLE:

object=description, description, program file, format

Value	Meaning
object	Specifies the name of the server object.
description	Provides a description of the server object. The second description appears in the list of registered file types in the Registration Info Editor.
program file	Identifies the path and name of the executable file for the application that is used to create the object.
format	Identifies the format of the file. The value for this entry is most often **picture**, which is the Windows Metafile format. To change this entry, use the Registration Info Editor.

[Windows Help] Section

The **[Windows Help]** section includes settings that specify the size and placement of the Help window and dialog boxes, as well as the color of text that, when chosen, displays a macro, pop-up window, or new panel. The **[Windows Help]** section can include the following entries:



M_WindowPosition=*number,number, number,number,number*
H_WindowPosition=*number,number, number,number,number*
A_WindowPosition=*number,number, number,number,number*
C_WindowPosition=*number,number, number,number,number*

These entries define the default size and position of the main Help window, and the History, Annotate, and Copy dialog boxes. The first and second *number* values define the *x* and *y* coordinates of the upper-left corner for the default position of the main Help window and the dialog boxes. The third and fourth *number* values define the default width and height (in pixels) of the Help window and dialog boxes. The fifth *number* specifies whether the main Help window is maximized, where 1 is maximized, and 0 is the default height and width of the window. Changing this value has no effect on the size of the dialog boxes, because they cannot be maximized.

To change this entry, move or size the main Help window, or the History, Annotate, or Copy dialog boxes. The values are updated in WIN.INI automatically.



IFJumpColor=*red-value green-value blue-value*

This entry specifies the color of text that, when chosen, leads to a new panel of information in another help file. To change this entry, you must edit WIN.INI.



IFPopupColor=*red-value green-value blue-value*

This entry specifies the color of text that, when chosen, displays a pop-up panel (the type of panel used for glossary definitions) located in a different help file. To change this entry, you must edit WIN.INI.

JumpColor=*red-value green-value blue-value*

This entry specifies the color of text that leads to a new panel of help information. For example, entering 000 000 000 as values results in black text on a white background. To change this entry, you must edit WIN.INI.



MacroColor=*red-value green-value blue-value*

This entry specifies the color of text that, when chosen, runs a Help macro. To change this entry, you must edit WIN.INI.



PopupColor=*red-value green-value blue-value*

This entry specifies the color of text that, when chosen, displays a pop-up panel (the type of panel used for glossary definitions). For example, entering 000 000 000 as values results in black text on a white background. To change this entry, you must edit WIN.INI.

[sound] Section

The **[sound]** section lists the system events that support sound and the sound files that are assigned to each event. The **[sound]** section can contain one or more occurrences of the following entry:



system event=filename,description

This entry specifies the sound file assigned to and a description of a *system event*. The *filename* keyname is the name of the sound file. The *description* keyname is a text string that describes the system or application event. To change this entry, choose the Sound icon in Control Panel.

[printerPorts] Section

The **[printerPorts]** section lists the active and inactive printers that can be accessed by Windows device drivers, specifies the communications port or ports to which the devices are connected, and specifies their corresponding timeout values. This section can contain one or more entries in this format:

device=driver, port, DeviceTimeout, RetryTimeout [, other ports . . .]

Value	Meaning
device	The name of the device.
driver	The driver filename, with path if necessary.
port	Specifies the port the device is connected to. If a device is not currently connected, the value is the **NullPort=** entry of the **[Windows]** section. There is a set of *port, DeviceTimeout,* and *RetryTimeout* values for each port that has a printer of the specified type.
DeviceTimeout	Determines the number of seconds that Print Manager will wait for the device to be switched on.
RetryTimeout	Determines the number of seconds before attempted transmission retries. This value might need to be increased for network printers.

If the timeout values are not specified for a particular port, Windows uses the default values listed in the **[Windows]** section for that port.

To change this entry, choose the Printers icon from Control Panel or choose Printer Setup from the Options menu in Print Manager, then choose Connect.

Important When changing an entry in this section, be sure it does not conflict with a corresponding entry in the **[devices]** section. Problems can occur if a connected device does not appear in the **[printerPorts]** section.

[devices] Section

The **[devices]** section lists the active printers, and is necessary only for compatibility with Windows version 2.x applications. The entries in this section are identical to those in the **[printerPorts]** section, without the timeout values. The **[devices]** section can contain one or more entries in this format:

device=driver, port [, other ports . . .]

The values specified are identical to the first three items in the **[printerPorts]** section. To change this entry, choose the Printers icon from Control Panel.

Important When changing an entry in the **[devices]** section, be sure it does not conflict with a corresponding entry in the **[printerPorts]** section. Problems can occur if a connected device does not appear in the **[printerPorts]** section.

[programs] Section

The **[programs]** section identifies the paths, in addition to those in the AUTOEXEC.BAT file, that Windows will search to start an application. If you try to open a data file associated with an application, and Windows cannot find the program file needed to start the application, Windows will prompt you to supply the path for the program file. The path you specify is automatically added to the **[programs]** section of WIN.INI. Windows will search the path for the program file the next time you want to start an application by opening an associated data file. The **[programs]** section can contain one or more occurrences of the following entries to specify the drive, directory, and executable program file needed to start an application:



program file=drive:directory\program file

Value	Meaning
program file	Specifies the name of the program file that Windows was searching for when you tried to open the associated data file.
drive	Specifies the drive on which the program file is located.
directory	Specifies the directory in which the program file is located. The second *program file* keyname specifies the name of the executable program file, located in the specified drive and directory, needed to start the associated application.

To change this entry, complete the dialog box in File Manager that prompts you for the path if you try to open an application that it can't find.

[colors] Section

The **[colors]** section defines the colors for components of the Windows display. To change these entries, choose the Color icon from Control Panel. This section can contain entries in this format:

component=red-value green-value blue-value

Value	Meaning
component	Any one of these keynames: **ActiveBorder** (the border of the active window) **ActiveTitle** (the title bar of the active window) **AppWorkspace** (the application workspace for Windows applications) **Background** (refers to the desktop) **ButtonFace** **ButtonShadow** **ButtonText** **GrayText** (text that is dimmed, as in an unavailable command name) **Hilight** (the background of highlighted text) **HilightText** (the highlighted text itself) **InactiveBorder** (the border of an inactive window) **InactiveTitle** (the title bar of an inactive window) **InactiveTitleText** (text in an inactive title bar) **Menu** (the background of menus) **MenuText** **Scrollbar** **TitleText** (the text in the title bar) **Window** (the workspace in a window) **WindowFrame** **WindowText**
red-value *blue-value* *green-value*	Integers that specify the relative intensities of the respective colors. These values can range from **0** (minimum intensity) to **255** (maximum intensity).

SYSTEM.INI: The System Initialization File

When you install Windows, Setup creates the SYSTEM.INI file, which contains global system information that Windows uses when it starts. The following sections can appear in SYSTEM.INI.

Section	Purpose
[boot]	Lists drivers and Windows modules.
[boot.description]	Lists the names of devices you can change using Windows Setup.
[drivers]	Contains a list of aliases (or names) assigned to installable driver files.
[keyboard]	Contains information about the keyboard.
[mci]	Lists Media Control Interface (MCI) drivers.
[NonWindowsApp]	Contains information used by non-Windows applications.
[standard]	Contains information used by Windows in standard mode.
[386enh]	Contains information used by Windows in 386 enhanced mode.

Setup assigns a value to each entry in the **[boot]** and **[keyboard]** sections, and to the **Device=** entry and its synonyms in the **[386enh]** section. These entries must appear in SYSTEM.INI for Windows to function properly. These entries do not have built-in values, but have specific values that Setup assigns when you install Windows.

Many of the other entries described in this chapter are rarely needed and will not appear in the SYSTEM.INI file unless you add them yourself. Most of these entries have built-in values that are present whether the entry appears in SYSTEM.INI. You might need to change one or more of these values to improve the performance of Windows or of a specific application.

The possible methods for changing values are noted for each entry in this chapter. You cannot change most SYSTEM.INI entries through Control Panel, but you can change many settings by running Setup. Other values in SYSTEM.INI can be changed only by opening the file and editing it with a text editor such as Notepad.

Any changes you make to the SYSTEM.INI file do not take effect until you restart Windows.

Important Always back up the SYSTEM.INI file before you open it. Use extreme care when making changes with a text editor, because you can disable your system if you edit the file incorrectly. If the entry can be changed without editing SYSTEM.INI, use that recommended method.

This section describes the contents of the SYSTEM.INI file and how to change values for entries in the file.

[boot] Section

The **[boot]** section contains a list of the drivers and Windows modules that Windows uses to configure itself each time you start it.

Important All entries in this section are required. If you modify or delete one of these entries, Windows might not start or operate properly. Setup assigns all values based on your system configuration; there are no built-in values for these entries.

The **[boot]** section can contain the following entries:

286grabber=*filename*

This entry specifies the filename of the grabber, which is the device driver that makes a non-Windows application visible when you run Windows in standard mode. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window and change the Display setting. If you are installing a device driver that is not included with Windows, exit Windows and run Setup from MS-DOS.

386grabber=*filename*

This entry specifies the filename of the grabber, which is the device driver that makes a non-Windows application visible when you run Windows in 386 enhanced mode. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window and change the Display setting. If you are installing a device driver that is not included with Windows, exit Windows and run Setup from MS-DOS.



CachedFileHandles=*number*

This entry specifies the number of the most recently-used executable files (.EXE) and dynamic-link library files (.DLL) that can remain open. Windows keeps these files open so that they can be accessed quickly, ensuring optimal performance. However, some networks have a limit on the number of files that can be open on a server at a time. If you have problems running Windows from a network server, use a lower number for this value. Valid numbers are **2** through **12**. The default is **12**. To change this entry, you must edit SYSTEM.INI.

comm.drv=*filename*

This entry specifies the filename of the serial communications driver you are using. The default is none (blank). To change this entry, you must edit SYSTEM.INI.

display.drv=*filename*

This entry specifies the filename of the display driver you are using. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window. If you are installing a device driver that is not included with Windows, exit Windows and run Setup from MS-DOS.



drivers=*filename* | *aliasname*

This entry specifies the filenames or aliases of the installable drivers that load when Windows is started. An installable driver is a dynamic-link library that Windows treats like a device driver. If the drivers you want to use include parameters, you must specify them by alias name as defined in the **[drivers]** section of SYSTEM.INI, where you can list several filenames or aliases. The default is none (blank). Most Setup programs for Windows applications add these settings to the SYSTEM.INI file when installing drivers, so you shouldn't need to change the values. To change this entry, you must edit SYSTEM.INI.

fixedfon.fon=*filename*

This entry specifies the filename of the fixed system font used by Windows version 2.x applications. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window. This entry also changes when you change the Display setting in Setup.

fonts.fon=*filename*

This entry specifies the filename of the proportionately spaced system font used by Windows 3.1. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window. This entry also changes when you change the Display setting in Setup.

keyboard.drv=*filename*

This entry specifies the filename of the keyboard driver you are using. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window and change the setting for the Keyboard option.

language.dll=*library-name*

This entry specifies the name of the dynamic-link library that supplies language-specific functions. If no language library is specified, Windows uses the built-in U.S. English library. The default is none (blank). To change this entry, choose the International icon from Control Panel and change the setting for the Language option.

mouse.drv=*filename*

This entry specifies the filename of the mouse driver you are using. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window and change the setting for the Mouse option.

network.drv=*filename*

This entry specifies the filename of the network driver you are using. The default is none (blank). To change this entry, choose the Windows Setup icon from the Main Group window and change the setting for the Network option.

oemfonts.fon=*filename*

This entry supplies the name of the font file for the OEM character set, which is associated with the Display setting. The default is none (blank). To change this entry, exit Windows and run Setup from MS-DOS. This entry also changes when you change the Codepage or Display setting in Setup.

shell=*filename*

This entry specifies the Windows shell program that runs when you start Windows. Unless you change this entry, Program Manager will run when you start Windows. The default is none (blank). Setup initializes this value as **progman.exe**. To change this entry, you must edit SYSTEM.INI.

sound.drv=*filename*

This entry specifies the filename of the system sound driver you are using. The default is none (blank). To change this entry, you must edit SYSTEM.INI.

system.drv=*filename*

This entry specifies the filename of the system hardware driver you are using. The default is none (blank). To change this entry, exit Windows and run Setup from MS-DOS, then choose a new System setting.



TaskMan.Exe=*filename*

This entry specifies the task-switching application that appears when you press CTRL+ESC. The default is **taskman.exe**. To change this entry, you must edit SYSTEM.INI.

[boot.description] Section

The **[boot.description]** section is a list of strings used to describe the devices you can change when you run Setup. There is no reason to change these entries. If you do change values, you cannot use Windows Setup to update drivers to newer versions.

[drivers] Section



This section contains a list of aliases (or names) assigned to installable driver files. The **[drivers]** section can contain this entry:

alias=driver filename[parameters]

This entry assigns an alias name to an installable driver and specifies any parameters used by the driver. This entry is associated with the **drivers=** entry in the **[boot]** section of SYSTEM.INI. If the driver includes parameters, you must assign an alias to it and then specify the driver by alias name in the **[boot]** section. You can define multiple aliases by specifying multiple alias lines in this entry. The default is none. Most Setup programs for Windows applications add these settings to SYSTEM.INI when installing drivers, so you shouldn't need to change the values. To change this entry, choose the Drivers icon in Control Panel or use the installation program provided by the manufacturer.

[keyboard] Section

The **[keyboard]** section provides information about the keyboard.

Important All entries in this section are required. If you modify or delete one of these entries, Windows will not operate properly. There are no built-in default values for these entries; Setup assigns values based on the system configuration.

The **[keyboard]** section can contain the following entries:

keyboard.dll=*filename*

This entry specifies the name of a dynamic-link library that defines the layout for non-U.S. keyboards and keyboards not compatible with IBM-compatible systems. This entry is required for all keyboards except these U.S. keyboards: IBM XT, PC/AT, or enhanced; AT&T type 301 or 302; and Olivetti 83-key. To change this entry, choose the Windows Setup icon from the Main Group window or the International icon from Control Panel.

oemansi.bin=*filename*

This entry specifies the name of a file that defines OEM/ANSI code-page translation tables for systems not using code page 437 (the U.S. OEM character set). To change this entry, exit Windows and run Setup from MS-DOS. This setting also changes when you change the Codepage or Display settings.

subtype=*number*

This entry distinguishes, for some drivers, special features for keyboards that otherwise have identical layouts. This value can also be used by other drivers. To change the **subtype=** entry, choose the Windows Setup icon from the Main Group window. See the **type=** entry for information about type values. These values are defined for **subtype=**.

Type value	*Subtype value*	*Meaning*
1	2	Olivetti M24 83-key or AT&T 6300 type 301 83-key.
1	4	AT&T type 302 sometimes used on the 6300 Plus.
2	1	Olivetti 102-key ICO used on M24 systems.

type=*number*

This entry specifies the keyboard type. It can be any one of these values.

Value	*Meaning*
1	IBM PC or XT compatible (83 keys)
2	Olivetti 102-key ICO
3	IBM AT compatible (84 or 86 keys)
4	IBM compatible, enhanced (101 or 102 keys)

If this entry is blank or missing, the driver selects a default type. For IBM-compatible keyboards (using the KEYBOARD.DRV driver), the default type is determined by the BIOS. To change this entry, choose the Windows Setup icon from the Main Group window.

[mci] Section



The **[mci]** section contains a list of the drivers that use the Media Control Interface (MCI) to play media files. These drivers are installed automatically when you run Setup. To change these values, choose the Drivers icon in Control Panel.

[NonWindowsApp] Section

The **[NonWindowsApp]** section contains entries that affect the performance of non-Windows applications. This section can contain the following entries:



CommandEnvSize=*bytes*

This entry specifies the size of the COMMAND.COM environment. Since running batch files with the extension .BAT starts COMMAND.COM, this setting also applies to batch files. The value for this setting must be either **0** or between **160** and **32768**. A value of **0** disables this setting. If the value is not valid, it will be rounded up to **160** or down to **32768**. If the value is less than the current size of the actual environment, this setting will be disabled, as if it were set to **0**. If you specified the environment size in a PIF for COMMAND.COM, the PIF setting overrides this setting. The default is **0** with MS-DOS versions earlier than 3.2. Otherwise, the value is the **/e:** parameter in the **shell=** line in CONFIG.SYS. To change this value, you must edit SYSTEM.INI.



DisablePositionSave=*Boolean*

When this entry is **0**, the position and font used in a non-Windows application is saved in the DOSAPP.INI file when you quit the application. If this entry is **1**, any settings that weren't saved previously in DOSAPP.INI will not be saved. If this entry is **1**, the setting can be overridden for each non-Windows application by selecting the Save Settings On Exit checkbox in the Fonts dialog box. The default is **0**. To change this entry, you must edit SYSTEM.INI.



FontChangeEnable=*Boolean*

This entry, if **1**, provides the ability to change the fonts when running non-Windows applications in a window on systems that use version 3.0 of the grabbers (usually in version 3.0 display drivers). Windows version 3.1 of the video grabbers (used in version 3.1 display drivers) include built-in support for changing fonts when running non-Windows applications in a window. If you are using a 3.0 grabber that has not been updated to include the ability to change fonts and you want to use this feature, set this value to **1**. However, when this entry is **1**, your screen may lose characters and the cursor may change size and position slightly. The default is **1** on systems that use Windows 3.1 grabbers, and **0** on systems that use Windows 3.0 grabbers. To change this entry, you must edit SYSTEM.INI.



GlobalHeapSize=*kilobytes*

This entry specifies the size of a buffer in conventional memory that Windows allocates when running in standard mode. This buffer can be used to share information among all non-Windows applications that are started from Windows. This setting has no effect when Windows is running in 386 enhanced mode. The default is **0**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this setting.)



LocalTSRs=*list-of-TSR-applications*

This entry specifies which terminate-and-stay-resident (TSR) programs work properly if they are copied to each instance of a virtual machine. When you start Windows, it detects any TSR programs that are currently running. If the TSR is on the **LocalTSRs=** list, Windows will place a copy of the TSR in each virtual machine you run. Many TSRs will not run properly if they are added to this list. Make sure your TSR is fully compatible with Windows and can be copied to a virtual machine before adding it to the list. The default is **dosedit, pced, ced**. To change this value, you must edit SYSTEM.INI.



MouseInDosBox=*Boolean*

This entry specifies whether the mouse is supported when running a non-Windows application in a window. Mouse support for non-Windows applications running in a window is available automatically if you are using a Windows 3.1 version of the grabbers. If you are using a Windows 3.0 version of the grabbers and you want mouse support, enable this setting. If you do not want mouse support, disable this setting. The default is **1** if an MS-DOS mouse driver is loaded that has the extension .COM or .SYS and that supports using a mouse with a non-Windows application. Otherwise, the default is **0**. To change this entry, you must edit SYSTEM.INI.

NetAsynchSwitching=*0 | 1*

This entry indicates whether you can switch away from an application running in Windows standard mode after it has made an asynchronous network BIOS call. The default value of **0** specifies that such task switching is not possible. Switching away from some applications that make these calls might cause the system to fail. Once Windows detects an asynchronous NetBIOS call, it will not allow switching away from the application even if no more of these calls are made. Set this value to **1** if you are sure the applications you use will not receive network messages while you are switched away from them. To change this entry, you must edit SYSTEM.INI.

ScreenLines=*number*

This entry specifies the number of lines displayed on the screen when a non-Windows application runs. An application that specifies a different screen mode can override this entry. The default is **25**. To change this entry, you must edit SYSTEM.INI.

SwapDisk=*drive*:*directory*

This entry provides the name of the disk drive and directory to which Windows standard mode swaps non-Windows applications. The default is the directory pointed to by the TEMP environment variable; if there is no TEMP variable, then the default is the boot directory of your first hard disk (usually C:). If your system does not have a hard disk, the default is the root directory on the first floppy drive (usually A:). To change this entry, you must edit SYSTEM.INI.

[standard] Section

The **[standard]** section contains entries that are specific to running Windows in standard mode. This section can contain the following entries:



FasterModeSwitch=*0* | *1*

This entry, if set to **1**, causes Windows standard mode to use a faster method of switching from protected to real mode on many 80286-based computers. When this entry is **1**, Windows responds more quickly to hardware interrupts, allowing better throughput for interrupt-intensive applications, such as communications applications. Also, you should set this entry to **1** if you are using a Zenith Z-248 system and are losing characters while typing, or if you are using an Olivetti M-250-E and lose control of the mouse. The default is **0**. To change this entry, you must edit SYSTEM.INI.

Note The value for **FasterModeSwitch=** has no effect on 80386-based computers. Some early IBM-AT and compatible computers do not have the BIOS support necessary to use this setting. Setting this entry to **1** on these computers may cause them to hang when starting Windows.

Int28Filter=*number*

This entry specifies the percentage of INT 28h interrupts, generated when the system is idle, that are visible to software loaded before Windows. Windows will reflect every *n*th interrupt, where *n* is the value of this entry. For example, a value of **1** reflects every INT 28h interrupt, a value of **2** reflects every second INT 28h interrupt, a value of **3** reflects every third INT 28h interrupt, and so on. Increasing this value might improve Windows performance, but may interfere with some memory-resident software such as a network.

Set this value to **0** to prevent INT 28h interrupts. But setting this value too low adds to system overhead that might interfere with communications applications. The default is **10**. To change this entry, you must edit SYSTEM.INI.



MouseSyncTime=*milliseconds*

This entry specifies the number of milliseconds that can elapse between mouse data bytes before Windows running in standard mode assumes that a mouse data packet is complete. This setting only affects Windows standard mode on computers with an IBM PS/2 mouse interface. The default is **500**. To change this value, you must edit SYSTEM.INI.

NetHeapSize=*kilobytes*

This entry specifies the size of the data-transfer buffers that Windows standard mode allocates in conventional memory for transferring data over a network. If an application is not running correctly, your network may require a larger buffer than the default. Increasing this value will decrease the amount of memory available to applications. If no network software is running, this entry is ignored and no memory is allocated. The default is **8**. To change this entry, you must edit SYSTEM.INI.

PadCodeSegments=*0* | *1*

Setting the value of this entry to **1** causes Windows Kernel to pad code segments with 16 bytes. This prevents the last instruction in the segment from being too close to the segment limit for 80286 C2 stepping. Set this value to **1** only for this 80286 stepping. Unfortunately, there is no easy way of telling what stepping a 80286 chip is. Try setting this value to **1** if your 80286 system hangs in standard mode. The default is **0**. To change this entry, you must edit SYSTEM.INI.



Stacks=*number*

This entry specifies the number of interrupt reflector stacks used by the standard mode MS-DOS Extender (DOSX) to map an MS-DOS or BIOS API from real mode to protected mode. If you receive a "Standard Mode: Stack Overflow" message, try increasing this number. You can specify a number between **8** and **64**. The default is **12**. To change this entry, you must edit SYSTEM.INI.



StackSize=*kilobytes*

This entry specifies the size of the interrupt reflector stacks used by the standard mode MS-DOS Extender. The default is **384**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)

[386enh] Section

The **[386enh]** section contains information specific to running Windows in 386 enhanced mode, including information used for virtual-memory page swapping.

For entries in this section that specify virtual devices, the value can appear in two ways: either the filename of a specific virtual device driver (with path if necessary), or as an asterisk (*) followed immediately by the device name to refer to a virtual device built into the WIN386.EXE file.

This section can contain the following entries:



32BitDiskAccess=*Boolean*

This entry turns 32-bit disk access on or off. The default is **Off** if your hard disk is capable of supporting 32-bit disk access. Otherwise, this setting is not present in SYSTEM.INI. To change this entry, choose the 386 Enhanced icon in Control Panel. If this setting is not present in SYSTEM.INI, the 32-Bit Disk Access check box will not appear in Control Panel Virtual Memory dialog box.



A20EnableCount=*number*

This entry specifies the initial A20 enable count, which identifies the A20 handler that HIMEM.SYS uses to access extended memory. (For more information about the A20 enable count, see the documentation for your extended memory manager.) The default is the value computed by the Win386 loader. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



AllEMSLocked=*Boolean*

This entry, if **On**, locks the contents of expanded memory used by all Windows and non-Windows applications into memory (instead of swapping it to disk) and overrides PIF settings for EMS Memory Locked. Set this value to **On** if you are using a disk cache program that uses expanded memory. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

AllVMsExclusive=*Boolean*

This entry, when **On**, forces all applications to run in exclusive full-screen mode, overriding all other settings in the application PIFs. Setting **AllVMsExclusive=on** might help avoid UAEs when you are running network or memory-resident software that is not compatible with Windows. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



AllXMSLocked=*Boolean*

This entry, if **On**, locks the contents of the extended memory used by all Windows and non-Windows applications into memory (instead of swapping it to disk), and overrides PIF settings for XMS Memory Locked. The default is **Off**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry).

AltKeyDelay=*seconds*

This entry specifies how much time Windows waits to process a keyboard interrupt after it processes an ALT interrupt. Some applications expect a slower processing rate than Windows 386 enhanced mode usually uses. Increase this value if such an application has trouble handling the ALT key. The default is **.005**. To change this entry, you must edit SYSTEM.INI.

AltPasteDelay=*seconds*

This entry specifies how much time Windows waits before pasting any characters after the ALT key has been pasted. Some applications may require more time for recognition of the ALT keystroke. The default is **.025**. To change this entry, you must edit SYSTEM.INI.



AutoRestoreScreen=*Boolean*

This entry specifies whether the display for non-Windows applications should be restored by Windows or by the applications themselves when they become active. If this entry is **On**, Windows saves the display information for a non-Windows application in memory when you switch away from the application. When you switch back to the application, Windows restores the screen. If this entry is **Off**, the application must restore its own display by repainting the screen. This requires less memory, but can slow down performance since Windows can usually restore the screen faster. This entry only applies to VGA displays and affects only applications that notify Windows that they can update their screens automatically when Windows sends a display update call. The default is **On**. To change this entry, you must edit SYSTEM.INI.



BkGndNotifyAtPFault=*Boolean*

This entry specifies when a notification should be sent to a non-Windows application or to Windows to prevent the application from attempting to access the display (which might corrupt the display of a different application). This notification can be sent while switching to a different application or when attempting to access the actual display. If this entry is **Off**, Windows sends a notification when switching between applications. Set this entry to **Off** if you are using a display that has special hardware (for example, 8514 and TIGA). If you are using a VGA display, setting this entry to **On** should work in most cases. The default is **On** for VGA displays; **Off** for 8514 displays. To change this entry, you must edit SYSTEM.INI.

CGA40WOA.FON=*filename*

This entry specifies the filename of the fixed-pitch display font used for non-Windows applications with a display of 40 columns and 25 or fewer lines. The default is none. To change this entry, exit Windows and run Setup from MS-DOS, then change the Codepage or Display setting.

CGA80WOA.FON=*filename*

This entry specifies the filename of the fixed-pitch display font used for non-Windows applications with a display of 80 columns and 25 or fewer lines. The default is none (blank). To change this entry, exit Windows and run Setup from MS-DOS, then change the Codepage or Display setting.

CGANoSnow=*Boolean*

This entry, when **On**, causes Windows to do special handling to avoid snow appearing on an IBM CGA display device. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

COM1AutoAssign=*number* | *seconds*
COM2AutoAssign=*number* | *seconds*
COM3AutoAssign=*number* | *seconds*
COM4AutoAssign=*number* | *seconds*

These entries indicate the contention detection values for each connected communications port. These values are used by Windows to determine how to arbitrate requests for the use of a device by more than one application, at least one of which is a non-Windows application. If the value is **-1**, Windows will display a warning message that asks you which application should be given control of the port. If the value is **0**, any application can use the device at any time. If the value is a positive integer less than 1000, this value represents the number of seconds after an application stops using the device before another application can use the same device. The default is **2**. To change this entry, choose the 386 Enhanced icon from Control Panel.

COM1Base=*address*
COM2Base=*address*
COM3Base=*address*
COM4Base=*address*

Flowchart 2.2 Serial Port Problems

These entries specify the base (first) port for the serial port adapter you are using for both standard mode and 386 enhanced mode. Check your hardware documentation for the appropriate value. The defaults are **COM3Base=3E8h**, and the port address values in the BIOS data area for COM1, COM2, and COM4. To change this entry, choose the Ports icon in Control Panel, and click the Advanced button for the selected COM port.

COMBoostTime=*milliseconds*

This entry specifies the time to allow a virtual machine to process a COM interrupt. If a communications application is losing keyboard characters on the display, you can try increasing this value. The default is **2**. To change this entry, you must edit SYSTEM.INI.

COM1Buffer=*number*
COM2Buffer=*number*
COM3Buffer=*number*
COM4Buffer=*number*

These entries specify the number of characters that the device will buffer on the corresponding communications port. Before changing one of these entries, make sure the corresponding **COMxProtocol=** entry has the proper value. Buffering may slow down communications on a port, but might be necessary to prevent some communications applications from losing characters at high baud rates. The size of the buffer required will depend on the speed of the machine and the application's needs. The default is **128**. Before increasing this value, see **COMxProtocol=**. To change this entry, you must edit SYSTEM.INI.



COMdrv30=*Boolean*

This entry, if **On**, indicates that the Virtual COM Driver (VCD) will use its own copy of the serial communications driver's interrupt handler. This will improve performance of COM ports. Set this entry to **On** if you are using a Windows 3.0 serial communications driver. Set this entry to **Off** if you are using the standard Windows 3.1 serial communications driver. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



COM1FIFO=*Boolean*
COM2FIFO=*Boolean*
COM3FIFO=*Boolean*
COM4FIFO=*Boolean*

These entries specify whether the FIFO buffer of a COM port's 16550 Universal Asynchronous Receiver Transmitter (UART) should be enabled (**On**) or disabled (**Off**). If a serial port does not have a 16550 UART, this setting is ignored. These values are used by Windows for both standard and enhanced modes. The default is **On**.

COM1Irq=*number*
COM2Irq=*number*
COM3Irq=*number*
COM4Irq=*number*

These entries specify which interrupt line is used by the device on the specified serial port. Check your hardware documentation for the correct value. If there is a hardware conflict between ports, set a value of **-1** to disable input for that COM port. The defaults for ISA and EISA machines are **COM1Irq=4**, **COM2Irq=3**, **COM3Irq=4**, and **COM4Irq=3**; for MCA machines, the defaults are **COM1Irq=4**, **COM2Irq=3**, **COM3Irq=3**, and **COM4Irq=3**. To change this entry, choose the Ports icon in Control Panel and click the Advanced button.

COMIrqSharing=*Boolean*

This entry specifies whether COM interrupt lines will be sharable between mulitiple serial ports or with other devices. Set this switch if your machine uses the same interrupt for COM3 or COM4 as it does for COM1 or COM2. The default is **On** for Micro Channel and EISA machines; **Off** for all other machines. To change this entry, you must edit SYSTEM.INI.

COM1Protocol=*XOFF* | *blank*
COM2Protocol=*XOFF* | *blank*
COM3Protocol=*XOFF* | *blank*
COM4Protocol=*XOFF* | *blank*

These entries specify whether Windows 386 enhanced mode should stop simulating characters in a virtual machine after the virtual machine sends an *XOFF* character. Set the value for a port to **XOFF** if a communications application using that port is losing characters while doing text transfers at high baud rates. Windows will resume simulating characters when the virtual machine sends another character after the *XOFF* character.

Leave this entry blank if the application does binary data transfers; setting this switch might suspend binary transmissions. The default is no entry, which is the same as any entry other than **XOFF**. Windows will not check for XOFF characters if this entry is blank or set to anything other than **XOFF**. If the application continues to lose characters after this entry is set, try increasing the related **COMxBuffer=** value. To change this entry, you must edit SYSTEM.INI.

Device=*filename* | ***devicename***

This entry specifies which virtual devices are being used with Windows 386 enhanced mode. This value can appear in two ways: either the name of a specific virtual device file (with path if necessary), or an asterisk (*) followed immediately by the device name to refer to a virtual device built into the WIN386.EXE file. Synonyms for **Device=** are **Display=**, **EBIOS=**, **Keyboard=**, **Network=**, and **Mouse=**. Filenames usually include the .386 extension. Multiple device lines are required to run Windows 386 enhanced mode. The default is none.

Setup assigns appropriate values based on your system configuration. To change this entry, you must edit SYSTEM.INI.



If Setup detects protected mode block devices for Western Digital–compatible controllers, it adds the entries **device=*int13** and **device=*wdctrl** to **[386enh]**. A block device is a virtual device used in 386 enhanced mode to talk directly to disk drive controllers, bypassing MS-DOS and BIOS. To remove block device support, delete these entries in SYSTEM.INI. You do not need to delete files from your hard disk, as these devices are appended to WIN386.EXE.

Display=*filename* | ******devicename*

This entry specifies the display device being used with Windows 386 enhanced mode. This entry is a synonym for **Device=**. The default is none. Setup assigns an appropriate value based on your system configuration. To change this entry, choose the Windows Setup icon from the Main Group window.

DMABufferIn1MB=*Boolean*

This entry, if set to **On**, indicates that the direct memory access (DMA) buffer memory should be in the first 1 MB of memory (above 640K, if possible) to be compatible with 8-bit bus master cards. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

DMABufferSize=*kilobytes*

This entry specifies the amount of memory to be reserved for the DMA buffer. This memory will be allocated above 640K, if possible. Windows 386 enhanced mode will default to a DMA buffer size that will handle disk access. The default is **16**. To change this entry, you must edit SYSTEM.INI.



DOSPromptExitInstruc=*Boolean*

This entry, if **On**, causes a message box to appear when you start the MS-DOS Prompt, displaying instructions on how to exit and switch away from the MS-DOS Prompt. Set this entry to **Off** if you do not want to see the message. The default is **On**. To change this entry, you must edit SYSTEM.INI.

DualDisplay=*Boolean*

Usually, when Windows is running in 386 enhanced mode, the memory between B000:0000 and B7FF:000F will be used by the general system unless a secondary display is detected. If this entry is **On**, this memory will be left unused and available for display adapters, or if you do not have a monochrome display card installed and want EMM386.EXE to include this address space as an upper memory block (UMB). If this entry is **Off**, the address range will be available on EGA systems but not under VGA systems, because the VGA display device supports monochrome modes, which use this address space. To change this entry, you must edit SYSTEM.INI.

EBIOS=*filename* | ***devicename***

This entry specifies the extended BIOS device being used with Windows 386 enhanced mode. This entry is a synonym for **Device=**. The default is the value that Setup assigns, based on your system configuration. To change this entry, you must edit SYSTEM.INI.

EGA40WOA.FON=*filename*

This entry specifies the filename of the fixed-pitch display font used for non-Windows applications with a display of 40 columns and more than 25 lines. The default is none. To change this entry, exit Windows and run Setup from MS-DOS, then change the Codepage or Display setting.

EGA80WOA.FON=*filename*

This entry specifies the filename of the fixed-pitch display font used for non-Windows applications with a display of 80 columns and more than 25 lines. The default is none. To change this entry, exit Windows and run Setup from MS-DOS, then change the Codepage or Display setting.

EISADMA=*Boolean* | *channel,size*

This entry specifies the mode of operation of an extended DMA channel for Extended Industry Standard Architecture (EISA) machines only. This entry's value can take one of two forms. If you set this entry to **Off**, Windows treats the machine as non-EISA, avoiding all EISA-related logic. Set this value to **Off** if you cannot run Windows 386 enhanced mode on an EISA machine. If you are using an EISA machine, you can specify the default transfer size for one or more DMA channels. The channels can operate in these modes: 8-bit (**8**), 16-bit specified in words (**16w**), 16-bit specified in bytes (**16b**), or 32-bit (**32**). If you are not using an EISA machine, Windows ignores this entry. The default is **0,8**; **1,8**; **2,8**; **3,8**; **5,16w**; **6,16w**; **7,16w**. Each pair goes with its own **EISADMA=** entry. To change this entry, you must edit SYSTEM.INI.

EMMExclude=*paragraph-range*

Flowchart Series 5 Windows Operation and Memory Problems

This entry specifies a range of memory that Windows will not scan to find unused address space. This has the side effect of turning off the RAM and ROM search code for the range. The range (two paragraph values separated by a hyphen) must be between A000 and EFFF. This scanning can interfere with some adapters that use the same memory area. The starting value is rounded down and the ending value is rounded up to a multiple of 16K. For example, you could set **EMMExclude=C800-CFFF** to prevent Windows from scanning the addresses C800:0000 through CFFF:000F (the block that some VGA cards use to enhance performance). You can specify more than one range by including more than one **EMMExclude=** entry. The default is none. To change this entry, you must edit SYSTEM.INI.

EMMInclude=*paragraph-range*

This entry specifies a range of memory that Windows will scan for unused address space regardless of what may be there. **EMMInclude=** takes precedence over **EMMExclude=** if you specify ranges that overlap. The range (two values separated by a hyphen) must be between A000 and EFFF. The starting value is rounded down and the ending value is rounded up to a multiple of 16K. For example, you could set **EMMInclude=C800-CFFF** to ensure that Windows scans the addresses C800:0000 through CFFF:000F. You can specify more than one range by including more than one **EMMInclude=** entry. The default is none. To change this entry, you must edit SYSTEM.INI.

EMMPageFrame=*paragraph*

This entry specifies the starting paragraph where the 64K page frame will begin when Windows cannot find a suitable page frame, allowing an EMS page frame in an area containing some unused RAM or ROM. For example, you could set **EMMPageFrame=C400** to start the page frame at C400:0000. The default is none. To change this entry, you must edit SYSTEM.INI.

EMMSize=*kilobytes*

This entry specifies the total amount of memory to be made available for mapping as expanded memory. The default allocates the maximum possible amount of system memory as expanded memory. Specify a value for this entry if you run an application that allocates all of the available expanded memory. This will be apparent if, when you run the application, you can never create any new virtual machine. If this value is **0**, then no expanded memory will be allocated. This entry does not prevent the EMS driver from being loaded; use **NoEMMDriver=on** to turn off EMS. The default is **65,536**. To change this entry, you must edit SYSTEM.INI.

FileSysChange=*Boolean*

This entry indicates whether File Manager automatically receives messages anytime a non-Windows application creates, renames, or deletes a file. When this entry is **Off**, a virtual machine can be run exclusively even when it modifies files. Setting this entry to **On** can slow down system performance significantly. The default is **On** for 386 enhanced mode; **Off** in standard mode. To change this entry, you must edit SYSTEM.INI.

Global=*device-name*

This entry defines MS-DOS devices loaded in CONFIG.SYS that need to be global to the system. The default is that all devices loaded in CONFIG.SYS are global to the system. But certain virtual devices might specify that an MS-DOS device be local (for example, MS$MOUSE). Use this entry to override that local specification. The *device-name* value must exactly match the case of the device name in CONFIG.SYS, or this entry will not work. Most device names are in all capital letters, so this value must usually be in all caps. To change this entry, you must edit SYSTEM.INI.



HardDiskDMABuffer=*kilobytes*

This entry specifies the amount of memory used for the Direct Memory Access (DMA) buffer. If you are using SMARTDrive and double buffering is turned on, this amount is determined automatically. You may need to change this value if you are using a hard disk that supports DMA and are not using SMARTDrive, or if you do not have double buffering turned on. The default is **0** for AT architecture computers; **64** for Micro Channel computers or computers that use DMA channel 3 (unless you use SMARTDrive with double buffering, in which case the default is **0**). To change this entry, you must edit SYSTEM.INI.



IdleVMWakeUpTime=*seconds*

This entry causes timer interrupts to periodically "wake up" idle virtual machines after the specified number of seconds has elapsed. If a virtual machine does not use timer interrupts (INT 8 or INT 1Ch), Windows will not usually force timer interrupts into a virtual machine unless it is active. This entry forces the timer interrupts to occur. The value for this entry is rounded down to the lowest power of 2 (for example, 1, 2, 4, 8, 16, 32, 64). The default is **8**. To change this entry, you must edit SYSTEM.INI.

IgnoreInstalledEMM=*Boolean*

This entry, if set to **On**, allows Windows to start in 386 enhanced mode even when there is an unknown expanded memory manager (EMM) running. Starting Windows with an unknown EMM running can cause the system to fail if memory-resident software was using expanded memory before Windows started. Set this entry only if no such software is installed or you are sure it will not be active when you run Windows. This entry applies only to EMMs servicing physical EMS hardware; Windows will not disable unrecognized 80386 expanded memory emulators. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

InDOSPolling=*Boolean*

This entry, if set to **On**, prevents Windows from running other applications when memory-resident software has the **InDOS** flag set. Setting this entry to **On** is necessary if the memory-resident software needs to be in a critical section to do operations off an INT 21 hook. Setting this entry to **On** will slow down system performance slightly. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

INT28Critical=*Boolean*

This entry specifies whether a critical section is needed to handle INT 28h interrupts used by memory-resident software. Some network virtual devices do internal task switching on INT 28h interrupts. These interrupts might hang some network software, indicating the need to add an **INT28hCritical=** entry. If you are not using such software, setting this entry to **Off** might improve Windows task switching. The default is **On**. To change this entry, you must edit SYSTEM.INI.

IRQ9Global=*Boolean*

Flowchart 2.5 Cannot Access Floppy Drives in 386 Enhanced Mode

This entry, if set to **On**, converts IRQ 9 masks to global. Set this entry to **On** if your system hangs when it reads from a floppy drive, or to make sure the system reads the floppy drive before starting Windows. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

Keyboard=*filename* | ***devicename***

This entry specifies the keyboard driver for Windows 386 enhanced mode. This entry is a synonym for **Device=**. The default is none. Setup assigns an appropriate value based on your system configuration. To change this entry, choose the Windows Setup icon from the Main Group window.

KeyBoostTime=*seconds*

This entry specifies how much time an application gets to run with increased priority when it receives a keystroke. Use this entry to increase the response to keystrokes when several background applications are running. The default is **.001**. To change this entry, you must edit SYSTEM.INI.

KeyBufferDelay=*seconds*

This entry specifies the time to delay pasting keyboard input after the keyboard buffer is full. Some applications might require more than .2 seconds. The default is **.2**. To change this entry, you must edit SYSTEM.INI.

KeyIdleDelay=*seconds*

This entry specifies how long Windows ignores idle calls after simulating a keystroke into a virtual machine. You can set this value to **0** to speed up keyboard input, but some applications might respond sluggishly if you do. The default is **.5**. To change this entry, you must edit SYSTEM.INI.



KeyPasteCRSkipCount=*number*

This entry specifies the number of times that a read-status INT 16 call should return a status of "empty" for the keyboard buffer after pasting a carriage return and before pasting another character. When Windows pastes data from the Clipboard to a non-Windows application, it must first paste the data to the BIOS keyboard buffer before pasting it into the application. This setting is used to slow down fast pasting from the Clipboard to the keyboard buffer so that the application can handle all incoming characters from the buffer. If you seem to lose characters, or if the screen does not update often enough while pasting information from the Clipboard, increase this value. This setting is related to **KeyPasteSkipCount=**. The default is **10**. To change this entry, you must edit SYSTEM.INI.

KeyPasteDelay=*seconds*

This entry specifies how long to wait before pasting any characters after a key has been pasted. Some applications might require more time than .003 seconds for recognition of a keystroke. The default is **.003**. To change this entry, you must edit SYSTEM.INI.



KeyPasteSkipCount=*number*

This entry specifies the number of times that a read-status INT 16 call should return a status of "empty" for the keyboard buffer before pasting another character. When Windows pastes data from the Clipboard to a non-Windows application, it must first paste the data to the BIOS keyboard buffer before pasting it into the application. This setting is used to slow down fast pasting from the Clipboard to the keyboard buffer so that the application can handle all incoming characters from the buffer. If you seem to lose characters, or if the screen does not update often enough while pasting information from the Clipboard, increase this value. The default is **2**. This setting is related to **KeyPasteCRSkipCount=**. To change this entry, you must edit SYSTEM.INI.

KeyPasteTimeout=*seconds*

This entry specifies how much time to allow an application to make the necessary BIOS calls for reading keyboard input before Windows will change from the fast paste (INT 16h) to the slow paste (INT 9h) mechanism. The default is **1**. To change this entry, you must edit SYSTEM.INI.



KybdPasswd=*Boolean*

This entry specifies whether the Virtual Keyboard Device (VKD) should support PS/2 8042 commands that implement password security. This entry only applies to 8042 keyboard controllers that are compatible with the PS/2 computer. The default is **On** for IBM PS/2 computers; **Off** for all other computers. To change this entry, you must edit SYSTEM.INI.



KybdReboot=*Boolean*

If this entry is **On**, Windows will attempt to reboot the computer using a keyboard controller command. On some computers this method is unreliable. If your computer hangs while rebooting, then set this entry to **Off**, in which case Windows will quit and display a prompt to press CTRL+ALT+DEL a second time if you attempt to reboot from the keyboard while Windows is running. The default is **On**. To change this entry, you must edit SYSTEM.INI.

Local=*device-name*

This entry defines MS-DOS device drivers that need to be local to each virtual machine. The *device-name* value must exactly match the case of the device name, or this entry will not work. Most device names are in all capital letters, so this value must usually be in all caps. If a device driver is local, a separate copy of it will exist in each virtual machine, and each will keep different state information. Most device drivers will not function properly when this entry is used. An exception is CON, the MS-DOS console device, which should be local to avoid filling a buffer with input from multiple virtual machines. The default is none; but in a standard SYSTEM.INI file, Setup will set **Local=CON**. To change this entry, you must edit SYSTEM.INI.



LocalLoadHigh=*Boolean*

This entry determines how the upper memory blocks (UMBs) are used when Windows in running on MS-DOS version 5.0. If this entry is **Off**, Windows uses all of the upper memory area, leaving no extra UMBs available for virtual machines. If this entry is **On**, then Windows does not use all of the upper memory area, so the UMBs are available locally to each virtual machine. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



LocalReboot=*On* | *Off*

This entry specifies whether you can press CTRL+ALT+DEL to quit applications that cause an unrecoverable error in 386 enhanced mode, without restarting Windows. If this setting is **On**, you can quit the application. If this setting is **Off**, pressing CTRL+ALT+DEL will restart your entire system (as it usually does). To change this entry, you must edit SYSTEM.INI.

LPT1AutoAssign=*seconds*
LPT2AutoAssign=*seconds*
LPT3AutoAssign=*seconds*
LPT4AutoAssign=*seconds*

These entries apply to Windows version 3.0 only. If you have upgraded to Windows 3.1, these settings may still appear in your SYSTEM.INI file, but will have no effect. The default is 60.



LRULowRateMult=*number*

This entry specifies the value used to determine the Least Recently Used (LRU) low paging rate sweep frequency, which is computed by multiplying the value for **LRUSweepFreq=** by the value specified for **LRULowRateMult=**. Values between **1** and **65535** can be used. The default is **10**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



LRURateChngTime=*milliseconds*

This entry specifies the length of time that the Virtual Memory Manager (VMM) stays at high rate with no paging before switching to low rate, and the length of time the VMM stays at low rate with no paging before turning the LRU sweep off. The default is **10,000**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



LRUSweepFreq=*milliseconds*

This entry specifies the time between LRU sweep passes. This is also the high paging rate sweep frequency. The default is **250**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



LRUSweepLen=*length-in-pages*

This entry specifies the length in pages of the region swept on each pass. Windows computes this value by dividing the value of **LRUSweepReset=** by the value of **LRUSweepFreq=**. The value must be at least **1**. The default is **1024**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



LRUSweepLowWater=*number*

This entry specifies when the LRU sweeper should be turned on. When the number of free pages drops below this value, the sweeper is turned on. The default is **24**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)

LRUSweepReset=*milliseconds*

This entry specifies the time desired for an ACC bit reset divided by 4 MB of pages. Therefore, the time to reset all ACC bits is *number of pages in system+1023/1024*, where 1024 pages = 4 MB. The minimum value is **100**. The default is **500**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)

MapPhysAddress=*range*

This entry specifies the address range (in megabytes) in which the memory manager will preallocate physical page-table entries and linear address space. Set a value for this entry if you are using an MS-DOS device driver that needs this contiguous memory (such as an older version of RAMDrive that uses extended memory). The default is none. To change this entry, you must edit SYSTEM.INI.

MaxBPs=*number*

This entry specifies the maximum number of break points (a method for transferring control to Windows 386 enhanced mode) that can be used by the VMM. You may need to increase this value if you are using a third-party virtual device driver that requires more break points than the default value. The default is **200**. To change this entry, you must edit SYSTEM.INI.

MaxCOMPort=*number*

This entry specifies the maximum number of COM ports supported in 386 enhanced mode. Change this value if you have more than four COM ports in your system. The default is **4**. To change this entry, you must edit SYSTEM.INI.

MaxDMAPGAddress=*address*

This entry specifies the maximum physical page address that can be used for DMA. This entry works only with hard drives that support DMA. The default is **0FFFh** for non-EISA computers; **0FFFFFh** for EISA computers. To change this entry, you must edit SYSTEM.INI.

MaxPagingFileSize=*kilobytes*

This entry specifies the maximum size for a temporary swap file. The default is 50 percent of the available disk space. To change this entry, choose the 386 Enhanced icon in Control Panel, then choose Virtual Memory to set a value.

MaxPhysPage=*hexadecimal-page-number*

This entry specifies the maximum physical page number that the VMM can manage as a useable page, allowing pages to be added at a physical address beyond what the VMM recognized during initialization. If the value specified is less than what the VMM determines, the VMM will ignore several physical pages that it would usually use, preventing the use of memory. This is useful if

you are using a hardware device that cannot recognize all of the physical memory in your computer (for example, ISA DMA network cards cannot access physical memory above 16 MB). The default is determined by Windows based on the highest physical page number detected by the VMM during initialization. To change this entry, you must edit SYSTEM.INI.



MCADMA=*Boolean*

This entry specifies whether Windows should use the Micro Channel (MCA) extensions to DMA. This entry only applies to MCA computers. Set this entry to **Off** if you are using an MCA computer but the DMA extensions are not implemented. The default is **On** for most MCA computers; **Off** for all other computers. To change this entry, you must edit SYSTEM.INI.



MessageBackColor=*vga-color-attribute*

This entry specifies the background color of message screens, such as the screen that appears when you press CTRL+ALT+DEL. The default is **1** (for blue). To change this entry, you must edit SYSTEM.INI.



MessageTextColor=*vga-color-attribute*

This entry specifies the color used to display text in message screens, such as the screen that appears when you press CTRL+ALT+DEL. The default is **F** (for white). To change this entry, you must edit SYSTEM.INI.

MinTimeSlice=*milliseconds*

This entry specifies the minimum time a virtual machine will be allowed to run before other virtual machines can take over. A smaller value (such as 10 milliseconds) will make multitasking appear smoother, but will diminish the overall system performance. The default is **20**. To change this entry, choose the 386 Enhanced icon from Control Panel.



MinUnlockMem=*kilobytes*

This entry specifies the amount of memory that must remain unlocked and available for use when resuming a virtual machine if more than one virtual machine is running. The default is **40**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)

MinUserDiskSpace=*kilobytes*

This entry tells Windows how much disk space to leave free when creating a temporary swap file. Use this entry if your system's paging drive has less available space than Windows can use for paging. This entry has no effect if a permanent swap file exists. The default is **500**. To change this entry, you must edit SYSTEM.INI.

Mouse=*filename* | ***devicename***

This entry specifies the virtual device that handles the mouse hardware in 386 enhanced mode. This entry is a synonym for **Device=**. The default is none. Setup assigns an appropriate value based on your system configuration. To change this entry, choose the Windows Setup icon in the Main Group window.

MouseSoftInit=*Boolean*

This entry specifies whether Windows should convert INT 33h function 0 hard initialization calls to function 33 soft initialization calls, which do not reset the mouse hardware. Set this entry to **On** if you want to use a mouse with a non-Windows application started in a window. Set this entry to **Off** if the cursor and screen information appear distorted when you are using the mouse with an application. If you set this entry to **Off**, you will not be able to use the mouse with a non-Windows application started in a window. The default is **On**. To change this entry, you must edit SYSTEM.INI.

NetAsynchFallback=*Boolean*

This entry, if set to **On**, tells Windows to attempt to save a failing NetBIOS request. When an application issues an asynchronous NetBIOS request, Windows will attempt to allocate space in its global network buffer to receive the data. If there is insufficient space in the global buffer, Windows will usually fail the NetBIOS request. If this entry is **On**, Windows will attempt to save such a request by allocating a buffer in local memory and preventing any other virtual machines from running until the data is received and the timeout period specified by **NetAsynchTimeout=** expires. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

NetAsynchTimeout=*seconds*

This entry specifies the timeout period when Windows needs to enter a critical section to service an asynchronous NetBIOS request. This value is used only when **NetAsynchFallback=** is set. This value can include a decimal, such as 0.5. The default is **5.0**. To change this entry, you must edit SYSTEM.INI.

NetDMASize=*kilobytes*

This entry specifies the DMA buffer size for NetBIOS transport software if a network has been installed. In this case, the buffer size is the larger value between this value and the value of **DMABufferSize=**. The default is **32** on Micro Channel machines, and **0** on non–Micro Channel machines. To change this entry, you must edit SYSTEM.INI.

NetHeapSize=*number*

This entry specifies the size, in 4K increments, of the data-transfer buffers that Windows 386 enhanced mode allocates in conventional memory for transferring data over a network. All values are rounded up to the nearest 4K. The default is **12**. To change this entry, you must edit SYSTEM.INI.

Network=*filename* | ***devicename***

This entry specifies the type of network used with Windows 386 enhanced mode. This entry is a synonym for **Device=**. The default is none. Setup assigns an appropriate value based on your system configuration. To change this entry, choose the Windows Setup icon from the Main Group window.

NMIReboot=*Boolean*

This entry, if set to **On**, causes a reboot to occur when a nonmaskable interrupt is received. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

NoEMMDriver=*Boolean*

This entry, if set to **On**, prevents Windows 386 enhanced mode from installing the expanded memory driver. This differs from setting **EMMSize=0**, which prevents UMBs from being allocated, but does not prevent the EMS driver from being loaded. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



NoWaitNetIO=*On* | *Off*

This entry specifies whether to convert synchronous NetBIOS commands to asynchronous commands. When this entry is **On**, synchronous NetBIOS commands are converted. This can improve overall system performance when running several applications. The default is **On**. To change this value, you must edit SYSTEM.INI.



OverlappedIO=*Boolean*

This entry, if **On**, allows several virtual machines to make read and write requests to a disk before the first request has been completed. If this entry is **Off**, virtual machines cannot issue a request to read or write to a disk until any previous read and write requests have been completed. The default is **Off** if **InDOSPolling=on**; otherwise, **On**. To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry.)



PageBuffers=*number*

This entry specifies the number of 4K page buffers used to store asynchronous read and write pages. This setting is used only if you are using 32-Bit Disk Access and a permanent swap file. You can specify a value between **0** and **32**. Increasing this value can improve Windows performance. The default is **4**. To change this entry, you must edit SYSTEM.INI.



PageOverCommit=*megabytes*

This entry specifies the multiplier that determines the amount of linear address space the VMM will create for the system, which is computed by rounding up the amount of available physical memory to the nearest 4 MB and then multiplying that value by the value specified for **PageOverCommit=**. Increasing this value increases the amount of available linear address space, causing the size of data structures to increase. This also increases paging activity proportionately and can slow down the system. You can specify a value between **1** and **20**. The default is **4**. To change this entry, you must edit SYSTEM.INI.

Paging=*Boolean*

This entry sets demand paging (virtual memory). Set this entry to **Off** only if you need the disk space that would be used for a temporary swap file. The default is **On**. To change this entry, you must edit SYSTEM.INI.

PagingDrive=*drive-letter*

This entry specifies the disk drive where Windows 386 enhanced mode will allocate a temporary swap file. This entry is ignored if you have a permanent swap file. If you don't have a permanent swap file and no drive is specified or the specified drive does not exist, Windows attempts to put the temporary swap file on the drive containing the SYSTEM.INI file. If the specified drive is full, paging will be turned off. The default is none. To change this entry, you must edit SYSTEM.INI.



PagingFile=*path-and-filename*

This entry specifies the path and filename for the temporary swap file that is created when you start Windows in 386 enhanced mode. This file is deleted when you quit Windows. This setting overrides the **PagingDrive=** setting. The default is **WINDOWS\WIN386.SWP**. To change this entry, you must edit SYSTEM.INI.



PerformBackfill=*Boolean*

This entry specifies whether to allocate a full 640K of memory to a computer that has less than 640K of base conventional memory. Setting this entry overrides the automatic checking done by Windows. The default is **On** (Windows automatically checks to see if it needs to fill in memory based on how MS-DOS is occupying memory.) To change this entry, you must edit SYSTEM.INI. (You should never need to change this entry, because Windows can automatically detect whether to perform a backfill.)



PermSwapDOSDrive=*drive-letter*

This entry specifies the disk drive where Windows in 386 enhanced mode will create a permanent swap file. This setting is ignored if you are using a temporary swap file. The default is none. To change this entry, choose the 386 Enhanced icon from Control Panel.



PermSwapSizeK=*kilobytes*

This entry specifies the desired size of a permanent swap file. The default is none. To change this entry, choose the 386 Enhanced icon from Control Panel.

PerVMFILES=*number*

This entry specifies the number of private file handles Windows should allocate to each virtual machine. Increase this value if an application does not have enough file handles to run. The total number of file handles, including the global handles specified in the **files=** statement in CONFIG.SYS, cannot exceed 255. If it does exceed 255, this value will be rounded down. Set this value to **0** to prevent the allocation of any private file handles. The default is **10**; if the MS-DOS Share utility is installed, this setting in SYSTEM.INI is ignored. To change this entry, you must edit SYSTEM.INI.

PSPIncrement=*number*

This entry specifies the amount of additional memory, in 16-byte increments, that Windows should reserve in each successive virtual machine when **UniqueDosPSP=on**. The value that will work best for your machine might vary depending on the memory configuration and the applications you are running. Valid values are **2** through **64**. The default is **2**. See **UniqueDosPSP=** for more information. To change this entry, you must edit SYSTEM.INI.

ReflectDosInt2A=*Boolean*

This entry indicates whether Windows should consume or reflect MS-DOS INT 2A signals. If the value is **Off**, Windows will consume these signals and therefore run more efficiently. Set this entry to **On** if you are running memory-resident software that relies on detecting INT 2A messages. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



ReservedHighArea=*paragraph-range*

This entry specifies a range of memory that Windows will not scan to find unused address space. This also turns off the RAM and ROM search code for the range. This scanning can interfere with some adapters that use the same memory area. The range (two paragraph values separated by a hyphen) must be between **A000** and **EFFF**. The starting value is rounded down and the ending value is rounded up to a multiple of 4K. For example, you could set **ReservedHighArea=E100-E3FF** to prevent Windows from scanning the first 12K of memory starting at E100. You can specify more than one range by including more than one **ReservedHighArea=** entry. The default is none. To change this entry, you must edit SYSTEM.INI.

ReservePageFrame=*Boolean*

This entry tells Windows whether to give preference to EMS page frame space or conventional memory when it has to use one of them to allocate MS-DOS transfer buffers. This choice is necessary when Windows cannot find space between 640K and 1 MB other than EMS page frame space. If set to **On**, this entry will preserve EMS page frame space at the expense of conventional memory. If you are not going to run non-Windows applications that use expanded memory, set this entry to **Off** to give non-Windows applications more conventional memory. The default is **On**. To change this entry, you must edit SYSTEM.INI.



ReserveVideoROM=*Boolean*

This entry, if **On**, tells Windows that video ROM exists in pages C6 and C7. If the text font appears scrambled when you start non-Windows applications, set this entry to **On**. The default is **Off**. To change this entry, you must edit SYSTEM.INI. (You should not need to change this entry, because the default value causes Windows to detect video ROM correctly.)



ROMScanThreshold=*number*

This entry specifies a parameter used to determine whether an area of memory in the adapter area (usually between C000-EFFF) is ROM when the area has no ROM header or incorrect header information. The number for this entry specifies how many transitions (value changes) must occur within the memory area to determine whether it is ROM. If the number of transitions is greater than the value for this entry, Windows recognizes the memory as ROM. If the number of transitions is less than the value, Windows recognizes the memory as usable memory. Specifying **0** causes Windows to recognize the memory area as ROM and then reserve it as unusable. The default is **20**. (You should never need to change this entry.)



ScrollFrequency=*number*

This entry specifies the number of lines you can scroll in a non-Windows application that is running in Windows before its display is updated. The default is **2**. You should never need to change this setting.

SGrabLPT=*port-number*

This entry routes all printer interrupts on the specified port to the system virtual machine rather than to the current virtual machine. The default is none. To change this entry, you must edit SYSTEM.INI.



SyncTime=*Boolean*

This entry, if **On**, causes Windows to periodically synchronize its time with the computer's CMOS clock. If this entry is **Off**, Windows usually maintains the correct time, unless **TrapTimerPorts=off** when you are running applications that can cause the system time to run faster or slower than the actual time. The default is **On**. To change this entry, you must edit SYSTEM.INI.

SystemROMBreakPoint=*Boolean*

This entry specifies whether Windows should use ROM address space between F000:0000 and 1 MB for a break point. Windows 386 enhanced mode usually searches this space to find a special instruction that is used as a system break point. If this address space contains something other than permanently available ROM, set this entry to **Off**. The default is **On** if Windows is started in real mode of the 80386/486 CPU. **Off** if Windows is started in virtual mode of the 80386/486 CPU. Most 386 memory managers, such as QEMM and 386MAX, require this value to be set to **Off**. If a 386 memory manager is running, Windows is started in virtual mode. Otherwise, Windows is started in real mode. To change this entry, you must edit SYSTEM.INI.

SysVMEMSLimit=*number* | *kilobytes*

This entry specifies how many kilobytes of expanded memory Windows should be permitted to use. Setting this value to **0** prevents Windows from gaining access to any expanded memory. Setting it to **-1** gives Windows all the available expanded memory that it requests. The default is **2048**. To change this entry, you must edit SYSTEM.INI.

SysVMEMSLocked=*Boolean*

This entry indicates whether to swap to the hard disk the expanded memory being used by non-Windows applications. Locking expanded memory can improve the performance of a non-Windows application that uses it, but locking it slows down the rest of the system. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

SysVMEMSRequired=*kilobytes*

This entry specifies how much expanded memory must be free to start Windows. Leave this value at **0** if no non-Windows application running under Windows will require expanded memory. The default is **0**. To change this entry, you must edit SYSTEM.INI.

SysVMV86Locked=*Boolean*

This entry, if set to **On**, causes the virtual-mode memory being used in the system virtual machine to remain locked in memory rather than being swappable out to disk. Because Windows handles this process, there is no known reason to set this entry. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

SysVMXMSLimit=*number* | *kilobytes*

This entry specifies the maximum amount of memory the extended memory driver will allocate to MS-DOS device drivers and memory-resident software in the system virtual machine. Set the value to **-1** to give an application all the available extended memory that it requests. The default is **2048**. To change this entry, you must edit SYSTEM.INI.

SysVMXMSRequired=*kilobytes*

This entry specifies how many kilobytes of extended memory must be reserved by the XMS driver to start Windows. Leave this value at **0** if there are no XMS users in the system virtual machine. The default is **0**. To change this entry, you must edit SYSTEM.INI.

TimerCriticalSection=*milliseconds*

This entry instructs Windows to go into a critical section around all timer interrupt code, and specifies a timeout period. Specifying a positive value will ensure that only one virtual machine at a time receives timer interrupts. Some networks and other global memory-resident software may fail unless this entry is used. However, using this entry will slow down performance and can make the system sluggish or seem to stop for short periods of time. The default is **0**. To change this entry, you must edit SYSTEM.INI.

TokenRingSearch=*Boolean*

This entry, if **On**, tells Windows whether to search for a token ring network adapter on machines with IBM PC/AT architecture. Set this entry to **Off** if you are not using a token ring card and the search interferes with another device. The default is **On**. To change this entry, you must edit SYSTEM.INI.

TranslateScans=*Boolean*

This entry, if **On**, indicates whether Windows will translate a keyboard's scan codes to make them compatible with standard IBM scan codes. This entry is necessary only for keyboards that generate nonstandard scan codes. This entry is used only for the Switcher Screen and full-screen message boxes. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



TrapTimerPorts=*Boolean*

If this entry is set to **Off**, Windows allows applications to read from and write to the system timer ports. In this case, Windows can usually detect when an application has changed the timer interrupt interval and then make any adjustments to the time. However, some applications may prevent Windows from keeping accurate time. If your system's time appears to be running fast or slow, set this entry to **On**, so that Windows will deny access to the timer ports. This ensures that the system time will be accurate, but may cause applications that frequently access the timer ports to run slowly. If you do not want to set this entry to **On**, set **SyncTime=On**, which causes Windows to restore the correct time periodically regardless of whether applications access the timer ports. The default is **On**. To change this entry, you must edit SYSTEM.INI.

UniqueDOSPSP=*Boolean*

This entry, if set to **On**, tells Windows to start every application at a unique address (PSP). Each time Windows creates a new virtual machine to start a new application, Windows reserves a unique amount of memory (*i* bytes) below the application. For example, the first application would be loaded at address M, the second at address M+*i*, the third at M+2*i*, and so forth. The amount of memory (*i*) is determined by **PSPIncrement=**. These entries ensure that applications in different virtual machines all start at different addresses. Some networks use applications' load addresses to identify the different processes using the network. On such networks, setting this entry to **Off** might cause one application to fail when you exit another, because the network interprets them as the same application. However, setting this entry will leave slightly less memory for non-Windows applications. The default is **On** if you are running a network based on Microsoft Network (MS-Net) or LAN Manager; **Off** for all other networks. See Chapter12, "Networks and Windows 3.1," to find out if your network is based on MS-Net or LAN Manager. To change this entry, you must edit SYSTEM.INI.



UseableHighArea=*paragraph-range*

This entry specifies the range of memory that Windows will scan for unused address space regardless of what may be there. **UseableHighArea=** takes precedence over **ReservedHighArea=** if you specify ranges that overlap. The range (two values separated by a hyphen) must be between **A000** and **EFFF**. The starting value is rounded down and the ending value is rounded up to a multiple of 4K. For example, you could set **UseableHighArea=E100-E3FF** to ensure that Windows scans the first 12K of memory starting at E100.

You may specify more than one range by including more than one entry for **UseableHighArea=**. The default is none. To change this entry, you must edit SYSTEM.INI.

UseInstFile=*Boolean*

This entry specifies whether Windows should look in the INSTANCE.386 file for information it can use to determine whether data structures within MS-DOS need to be local. Two other methods give Windows this information: internal tables within the device, and an INT 2Fh call documented in the OEM Adaptation Kit. Both methods are preferable to using this setting; this entry is provided only for compatibility with Windows/386 version 2.x. The default is **Off**. To change this entry, you must edit SYSTEM.INI.



UseROMFont=*Boolean*

This entry, if **On**, causes Windows to use the soft font stored in video ROM for displaying messages that appear when non-Windows applications are running in a full screen and for displaying the text that appears when switching away from a non-Windows application. Set this entry to **Off** if the font used to display messages is different from the font used in the application, or if random dots and shapes appear on your screen. The default is **On**. To change this entry, you must edit SYSTEM.INI.



VGAMonoText=*Boolean*

This entry tells Windows to ignore the video memory address space in VGA displays, usually used for monochrome adapters. When this entry is **Off**, Windows can use the B000h through B7FF range in upper memory if no hardware device is using these addresses and other applications do not use the monochrome display mode of the VGA adapter. This entry applies only to VGA displays. The default is **On**. To change this entry, you must edit SYSTEM.INI.



VideoBackgroundMsg=*Boolean*

This entry, if **On**, causes Windows to display a message when a background application is suspended, or if its display cannot be updated properly because video memory is low. Setting this entry to **Off** turns off the warning message. This entry affects all non-Windows applications currently running. The default is **On**. To change this entry, you must edit SYSTEM.INI.



VideoSuspendDisable=*Boolean*

This entry specifies whether to suspend an application running in the background if its display becomes corrupted. If this entry is **On**, the application will continue running. If it is **Off**, applications will be suspended, and a warning message will appear if the **VideoBackgroundMsg=On**. This entry applies only to VGA displays. The default is **Off**. To change this entry, you must edit SYSTEM.INI.

VirtualHDIrq=*Boolean*

Flowcharts 2.5-2.6 Cannot Access Floppy Drives or Hard Disk

This entry allows Windows 386 enhanced mode to terminate interrupts from the hard disk controller, bypassing the ROM routine that handles these interrupts. Some hard drives might require that this entry be set to **Off** for interrupts to be processed correctly. If this entry is set to **Off**, the ROM routine handles the interrupts, which slows the system's performance. The default is **On** for AT-compatible computers; **Off** for other computers. To change this entry, you must edit SYSTEM.INI.

WindowKBRequired=*kilobytes*

This entry specifies how much conventional memory must be free to start Windows. The default is **256**. To change this entry, you must edit SYSTEM.INI.

WindowMemSize=*number* | *kilobytes*

This entry limits the amount of conventional memory Windows can use for itself. The default value indicates that Windows can use as much conventional memory as it needs. Try entering a positive value less than 640 if there is not enough memory to run Windows in 386 enhanced mode. The default is **-1**. To change this entry, you must edit SYSTEM.INI.

WindowUpdateTime=*milliseconds*

This entry specifies the time Windows takes between updates of the display for a non-Windows application running in a window. The default is **50**. To change this entry, you must edit SYSTEM.INI.

WinExclusive=*Boolean*

This entry, if **On**, indicates whether Windows always gets all of the computer's processing time whenever a Windows application is in the foreground. If this entry is **On** and a Windows application is in the foreground, no non-Windows application will run. The default is **Off**. To change this entry, choose the 386 Enhanced icon from Control Panel.

WinTimeSlice=*number,number*

This entry contains two numbers. The first number specifies the relative processing time (based on **MinTimeSlice=**) given to all Windows applications running in the foreground relative to the time allocated to all non-Windows applications running in the background. The second number specifies the relative processing time given to all Windows applications running in the background when a non-Windows application is running in the foreground. The range of numbers that can be entered is **1** through **10000**. The important value is the ratio of these values to the corresponding settings in the PIFs of the active non-Windows applications. The default is **100,50**. To change this entry, choose the 386 Enhanced icon from Control Panel.



WOAFont=*font filename*

This entry specifies which font files are loaded into memory when running non-Windows applications. These fonts are also available to Windows applications as long as a non-Windows application is running. If you want to make these fonts always available to Windows applications, add them to your system with the Fonts option in Control Panel. The default is **DOSAPP.FON**. To change this entry, exit Windows and run Setup from MS-DOS, then change the Codepage or Display setting.



XlatBufferSize=*kilobytes*

This entry specifies the size of the low-memory buffer used to map MS-DOS calls from protected mode to virtual 386 enhanced mode. Specifying a higher value can improve the performance of protected-mode Windows applications that read or write information in a large number of bytes (for example, some database applications). However, this may decrease the amount of memory available for running non-Windows applications and may cause problems on networks that use named pipes. These networks may require a value of **4**. The value you specify is rounded to the next 4K increment. For example, if you set this value to **5**, Windows rounds it up to **8**. The default is **8**. To change this entry, you must edit SYSTEM.INI.



XMSUMBInitCalls=*Boolean*

This entry specifies whether Windows should call the extended memory driver's UMB management routines. The default is usually **On**, but may be **Off** depending on the extended memory manager you are using. You should never need to change this entry.

Other Windows Initialization Files

Your WINDOWS directory includes at least three other initialization files: PROGMAN.INI, CONTROL.INI, and WINFILE.INI. These files contain the initial settings for elements defined by Program Manager, Control Panel, and File Manager, respectively.

Many other Windows applications create initialization files to store specific settings for the application. These files also appear in the WINDOWS directory with .INI filename extensions.

PROGMAN.INI: The Program Manager Initialization File

The Program Manager initialization file, PROGMAN.INI, usually has entries for **[settings]** and **[groups]**, which describe what should appear in the Program Manager window when you run Program Manager. To change the appearance and behavior of Program Manager, use the menu commands as described in Chapter 3, "Program Manager," in the *Windows User's Guide.*

A system administrator might also choose to add a third section, **[restrictions]**, for custom installations. For more information about creating custom entries for PROGMAN.INI for multiple systems during Windows installation, see "Configuring Program Manager for a Network Workgroup" in Chapter 6, "Tips for Configuring Windows 3.1."

[settings]

The **[settings]** section usually has these entries:

```
[Settings]
Window=60 28 565 388 1
SaveSettings=1
MinOnRun=0
AutoArrange=1
Startup=
```

Entry	Value
Window=	Four numbers that indicate the position of the window when Program Manager is opened, followed by a **1** if the window is maximized.
SaveSettings=	**1** if the Save Settings On Exit command on the Options menu is checked in Program Manager. In this case, Program Manager will save the current configuration when you close Windows.
MinOnRun=	**1** if the Minimize On Use command is checked on the Options menu in Program Manager. In this case, the Program Manager will be iconized when you run another application.
AutoArrange=	**1** if the AutoArrange command is checked on the Options menu in Program Manager. In this case, the icons in each group will be automatically arranged when you run Program Manager.
Startup=	Name of the group that serves as the "startup" group. If this entry is blank, the Startup group created in Windows Setup is the startup group.

[groups]

The PROGMAN.INI file will also have a section named **[groups]**, which has entries such as these:

```
[Groups]
Group1=C:\WINDOWS\ACCESSOR.GRP
Group2=C:\WINDOWS\GAMES.GRP
Group3=C:\WINDOWS\ALDUS.GRP
Group5=C:\WINDOWS\WORDFORW.GRP
Group6=C:\WINDOWS\MAIN.GRP
Group7=C:\WINDOWS\STARTUP.GRP
Order= 8 7 2 3 5 1 6
Group8=C:\WINDOWS\APPLICAT.GRP
```

Entry	Value
Groupx=	A filename (with full path) for the .GRP file that contains the group created during Setup, or created when an application was installed, or that you created in Program Manager.
Order=	A list of numbers separated with spaces, indicating the order in which the groups are drawn in the window.

To change these settings, we recommend that you use the commands on the Program Manager menu.

[restrictions]



For network system administration, you can also add a **[restrictions]** section to PROGMAN.INI to restrict user actions. The **[restrictions]** section can have these entries:

```
[restrictions]
NoRun=
NoClose=
NoSaveSettings=
NoFileMenu=
EditLevel=
```

Entry	Value
NoRun=	**1** disables the Run command on the File menu. The Run command will appear dimmed on the File menu and the user will not be able to run applications from Program Manager unless the applications are set up as icons in a group.
NoClose=	**1** disables the Exit Windows command on the File menu. Users will not be able to quit Program Manager through the File Menu or the Control menu (the Exit Windows and Close commands will be dimmed), or by using ALT+F4.
NoSaveSettings=	**1** disables the Save Settings on Exit command on the Options menu. The Save Settings command will appear dimmed on the Options menu and any changes that the user makes to the arrangement of windows and icons will not be saved when Windows is restarted. This setting overrides the **SaveSettings=** entry in the **[Settings]** section of the PROGMAN.INI file.
NoFileMenu=	**1** removes the File menu from Program Manager. All of the commands on that menu will be unavailable. Users can start the applications in groups by selecting them and pressing ENTER, or by double-clicking the icon. Unless you have also disabled the Exit Windows command, users can still quit Windows by using the Control menu or ALT+F4.
EditLevel=*n*	Sets restrictions for what users can modify in Program Manager. You can specify one of the following values for *n*: **0** allows the user to make any change. (This is the default value.) **1** prevents the user from creating, deleting, or renaming groups. If you specify this value, the New, Move, Copy, and Delete commands on the File menu are not available when a group is selected. **2** sets all restrictions in **EditLevel=1**, plus prevents the user from creating or deleting program items. If you specify this value, the New, Move, Copy, and Delete commands on the File menu are not available at all. **3** sets all restrictions in **EditLevel=2**, plus prevents the user from changing command lines for program items. If you specify this value, the text in the Command Line box in the Properties dialog box cannot be changed. **4** sets all restrictions in **EditLevel=3**, plus prevents the user from changing any program item information. If you specify this value, none of the areas in the Properties dialog box can be modified. The user can view the dialog box, but all of the areas are dimmed.

To enable any of the commands or remove any of the **EditLevel=** restrictions, either remove the entry from the PROGMAN.INI file, or set the value to **0**.

CONTROL.INI: Control Panel Initialization File

The CONTROL.INI file contains several sections that specify the options you set by choosing icons in Control Panel.

Section	Values specified
[Current]	Specifies the current color scheme.
[Color Schemes]	Defines the colors for each element of specific color schemes, as set by choosing the Color icon.
[Custom Colors]	Defines the custom colors in the color palette, as set by choosing the Color icon.
[Patterns]	Defines the color values for the bitmap patterns, as set by choosing the Desktop icon.
[MMCPL]	Specifies values related to the multimedia items in Control Panel.
[Screen Saver.*]	Specifies the density, warp speed, and password-protection values for the screensaver.
[Userinstallable.drivers]	Specifies values related to installable drivers used for multimedia.
[Drivers.Desc]	Specifies the MIDI Mapper and Time, the multimedia control devices.
[Installed]	Specifies the current version of Microsoft Windows and the installed printers.

To change any of these items, choose the related icons in Control Panel, as described in Chapter 5, "Control Panel," in the *Windows User's Guide.*

WINFILE.INI: The File Manager Initialization File

The WINFILE.INI file contains one section, **[settings]**, to specify the options you can set by choosing menu commands in File Manager. For example, these settings might appear in WINFILE.INI:

```
[Settings]
Window=0,6,800,600, , ,1
dir1=48,7,543,236,-1,-1,3,30,201,1808,150,B:\*.*
Face=Small Fonts
Size=8
LowerCase=1
ConfirmDelete=0
ConfirmSubDel=0
ConfirmReplace=1
ConfirmMouse=1
ConfirmFormat=1
```

Entry	Value
Window=	The size and position of the window and whether it is maximized when opened.
dir1=	The current directory settings.
Face=	The name of the typeface used for desktop items. The default is **Small Fonts**.
Size=	The point size for the typeface. The default is **8**.
LowerCase=	**1** if filenames are to appear in lowercase.
ConfirmDelete=	**1** if the user is to be prompted to confirm file deletion requests.
ConfirmSubDel=	**1** if the user is to be prompted to confirm subdirectory deletion requests.
ConfirmReplace=	**1** if the user is to be prompted to confirm file replacement requests.
ConfirmMouse=	**1** if the user is to be prompted to confirm mouse drag-and-drop requests.
ConfirmFormat=	**1** if the user is to be prompted to confirm formatting requests.

To change values for these settings, use the menu commands in File Manager, as described in Chapter 4, "File Manager," in the *Windows User's Guide.*

Configuring Windows 3.1



Part
2 Configuring Windows 3.1

Chapter 5 **Windows 3.1 and Memory Management** 227

About Memory 228
Windows Standard Mode and Memory 235
Windows 386 Enhanced Mode and Memory 238
Other Memory Management Issues 247
SMARTDrive 4.0: A Technical Discussion 252

Chapter 6 **Tips for Configuring Windows 3.1** 257

Tips for Optimizing Performance 258
Tips for Configuring the Windows Shell 272

Chapter

5 Windows 3.1 and Memory Management

This chapter contains information about how Microsoft Windows 3.1 interacts with memory. You can use this information to manage memory while running Windows and to troubleshoot various problems related to memory management. For specific information about optimizing your system configuration, see Chapter 6, "Tips for Configuring Windows 3.1."

Related information

- *Windows User's Guide:* Chapter 5, "Control Panel," and Chapter 14, "Optimizing Windows"
- *Windows Resource Kit:* Chapter 7, "Setting Up Non-Windows Applications," and Chapter 13, "Troubleshooting Windows 3.1"
- *Glossary terms:* Expanded Memory Specification, Extended Memory Specification, multitasking, page frame, protected mode, virtual memory

Contents of this chapter

About Memory 228
Types of Memory: An Overview 228
The Windows 3.1 Memory Device Drivers 230
Expanded Memory: A Technical Discussion 230
Windows Standard Mode and Memory 235
Extended Memory and Standard Mode 236
Expanded Memory and Standard Mode 237
Windows 386 Enhanced Mode and Memory 238
WINA20.386 and 386 Enhanced Mode 238
Extended Memory and 386 Enhanced Mode 239
Expanded Memory and 386 Enhanced Mode 240
Virtual Memory and 386 Enhanced Mode 243
Other Memory Management Issues 247
DPMI and VCPI Specifications 247
MS-DOS 5.0 and Windows 3.1 248
Memory and Windows Startup Requirements 249
Memory and the Windows System Resources 250
SMARTDrive 4.0: A Technical Discussion 252
About SMARTDrive 4.0 252
SMARTDrive 4.0: Frequently Asked Questions 254

About Memory

Your computer's random access memory (RAM) is a volatile medium where applications and data are stored while you are working with them. When you finish working, the information is transferred back to permanent storage on the hard disk or floppy disks.

Windows applications such as Microsoft Word and Microsoft Excel have to be loaded into memory before they can run. Generally, when you run Windows, the more memory your system has available, the more applications you can run at the same time, and the faster the applications will run.

This section describes the types of memory in a PC and provides a technical overview of expanded memory. Other sections in this chapter present issues specific to using memory in Windows standard mode and 386 enhanced mode.

Note This chapter supplements, rather than repeating, the information in Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

Types of Memory: An Overview

Your computer system can have three different kinds of memory: conventional memory, extended memory, and expanded memory. Windows also creates a fourth type of memory, virtual memory, which is discussed in "Virtual Memory and 386 Enhanced Mode" later in this chapter.

- **Conventional memory** consists of the first 640K of memory available on your machine. Most PCs have at least 256K of conventional memory. Your system must have 640K of conventional memory to run Windows.

  When you boot your machine, MS-DOS runs the utilities and applications listed in the CONFIG.SYS and AUTOEXEC.BAT files. These files often use conventional memory to function. The remaining memory is available for running other applications such as Windows.

- **Extended memory** is essentially a seamless upward extension of the original one megabyte address space available in the memory of 80286 and 80386 machines. Extended memory always starts exactly at 1024K, where the upper memory area ends. The first 64K of extended memory is referred to as the high memory area (HMA).

- **Expanded memory** can be installed as an expanded memory card or, on an 80386 machine, emulated by an expanded memory manager (EMM). The EMM software maps pages of expanded memory onto the system's upper memory area (from 640K to 1024K). Applications must be designed to interact with EMM software to take advantage of expanded memory.

Expanded memory is slower and more cumbersome to use than extended memory, because the expanded memory manager gives applications access to only a limited amount of expanded memory at a time.

An 80286 processor can address 16 megabytes of total memory, and an 80386 processor can address 4 gigabytes of memory. The 8086 and 8088 machines have hardware limitations that exclude use of extended memory. For such machines, expanded memory is the only option for extra memory.

Working with conventional and extended memory under Windows 3.1 is a straightforward process that seldom demands your attention, because Windows handles it automatically through HIMEM.SYS. If you want to run non-Windows applications that require expanded memory, you need a deeper understanding of how it is structured and how to access it. For more details, see "Expanded Memory: A Technical Discussion" later in this chapter.

Figure 5.1 shows the relative addresses of conventional memory, the upper memory area (used by expanded memory), and extended memory.

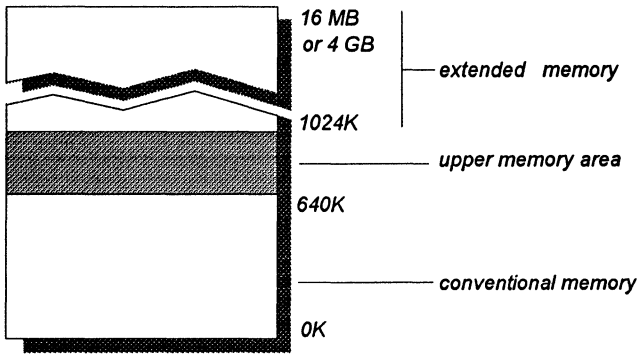


Figure 5.1

Extended memory goes from 1024K to 16 MB for 80286 PCs, or to 4 GB for 80386 PCs

Expanded memory uses page frames in the upper memory area

Conventional memory goes from 0K to 640K

To see the kind and amount of memory in your system:

- Type **msd** at the command prompt to run the Microsoft Diagnostics utility that is installed with Windows 3.1. This utility prepares an extensive report on the memory and drivers installed in your system.
- Or, if MS-DOS 5.0 is installed on your system, type **mem** at the command prompt to see a brief summary of your system's memory.

You can also use parameters with the MS-DOS 5.0 **mem** command. The **mem /c** command reports free memory for both conventional and upper memory addresses, and lists the decimal and hexadecimal sizes of all programs loaded into conventional and upper memory. The **mem /p** command presents a detailed listing of memory use, including load order, size, and

address information for programs (listed separately from the environment). For more information, see your MS-DOS 5.0 documentation.

Note If you use the **mem** command to view memory from within a Windows 3.1 virtual machine, you will only see the amount of memory available to the virtual machine.

The Windows 3.1 Memory Device Drivers

Both Windows 3.1 and MS-DOS 5.0 provide these device drivers for managing memory:

- HIMEM.SYS provides access to extended memory and the HMA. The extended memory it provides conforms to XMS 3.0.
- EMM386.EXE takes the XMS 3.0 memory provided by HIMEM.SYS and uses it to emulate expanded memory or to provide UMBs, or both. EMM386.EXE is DPMI-compliant and emulates expanded memory that conforms to LIM 3.2 or LIM 4.0.
- RAMDRIVE.SYS creates a RAM disk.
- SMARTDRV.EXE is a disk caching utility.

In addition, Windows 3.1 has two built-in memory managers:

- Swapdisk, which manages the disk swap process for caching Windows application code in both standard and 386 enhanced modes.
- Virtual Memory Manager (VMM), which manages swapping to a temporary or permanent swap file in 386 enhanced mode.

Expanded Memory: A Technical Discussion

Note This discussion is relevant if you want to take advantage of expanded memory installed in your system, or if you are running Windows 3.1 with non-Windows applications that require expanded memory. If your system doesn't have expanded memory, or if you are not running applications designed to use expanded memory, you can skip this discussion.

The original IBM PC design, based on the Intel 8086/8088 CPU, restricts usable memory to about 640K. A Lotus/Intel/Microsoft (LIM) collaboration developed a technique for adding memory to PC systems. The LIM Expanded Memory Specification (EMS) bypasses the memory limits by supporting memory cards that contain 16K pages (or banks) of RAM that must be enabled or disabled by

software, but that cannot normally be addressed by the CPU. Instead, each page is mapped into the address space of the processor.

Applications can use expanded memory to get around the memory limitations of 8086/8088 processors by using a special area of the machine's memory called the upper memory area. This upper memory area is always located in the same area of the computer's address space: from 640K to 1024K (A000 to FFFF hexadecimal). The upper memory area (sometimes referred to as the "adapter segment" because that part of memory is used by hardware adapters such as display adapters) is also where the ROM BIOS read-only memory segment is located. Figure 5.2 shows the upper memory area, its address range, and the items that occupy fixed portions of it.

An application must be specifically written to take advantage of expanded memory. Many non-Windows applications use expanded memory:

- To gain more effective performance from large non-Windows applications such as spreadsheets and CAD programs.
- To run memory-resident programs or applications that use shared data.

This section describes the LIM expanded memory specifications and discusses two operations for using expanded memory: bank switching and backfilling.

Note The 80386 and higher microprocessors can emulate EMS hardware by using extended memory with memory managers and special software such as EMM386.EXE. For more information about using this device driver with Windows 3.1, see Chapter 6, "Tips for Configuring Windows 3.1."

The Expanded Memory Specifications

The two kinds of expanded memory are differentiated from one another by their LIM expanded memory specification version numbers:

- **LIM 3.2** This expanded memory specification moves data in 64K blocks, each made up of four contiguous 16K pages to form a 64K page frame. The LIM 3.2 standard works well for storing data from spreadsheets in expanded memory, but it is insufficient for multitasking.
- **LIM 4.0** This expanded memory specification allows data to be moved in blocks of 1 to 64 pages (overcoming the LIM 3.2 limitations on size and flexibility). LIM 4.0 also removes the restriction that the pages must be contiguous, essentially abandoning the page frame requirement.

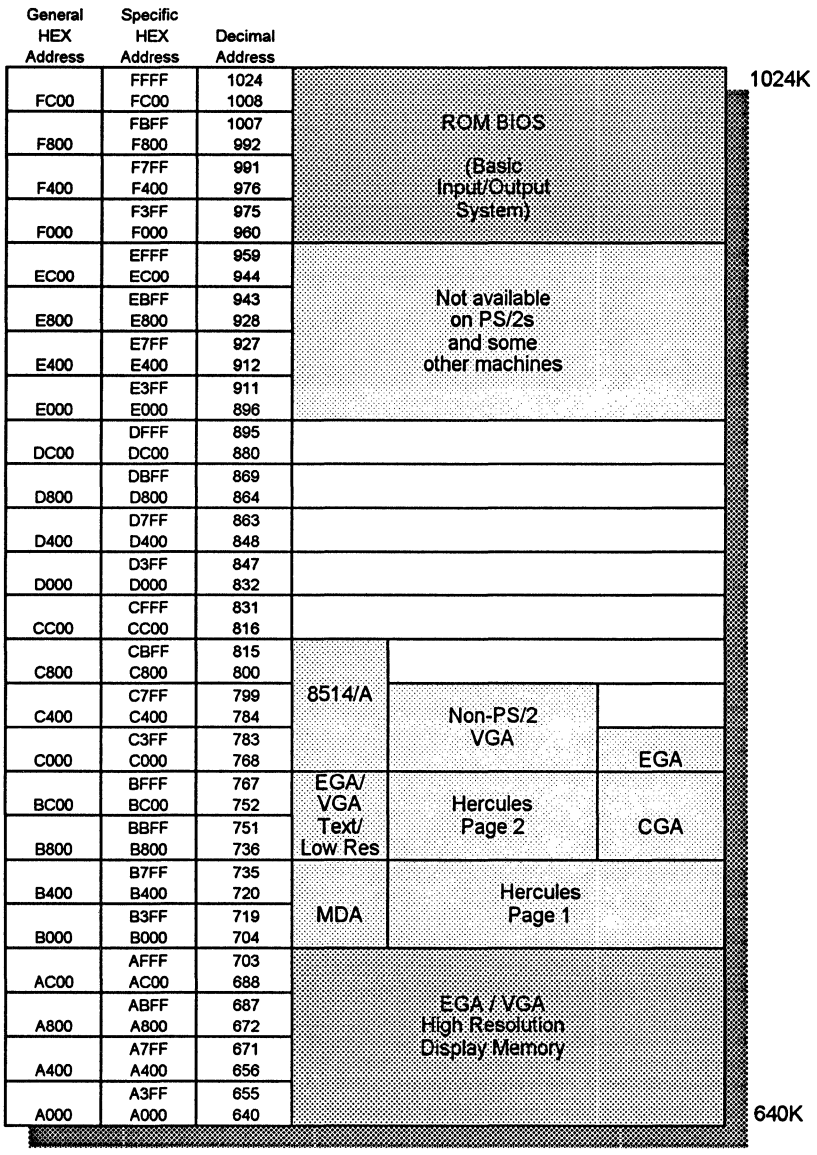


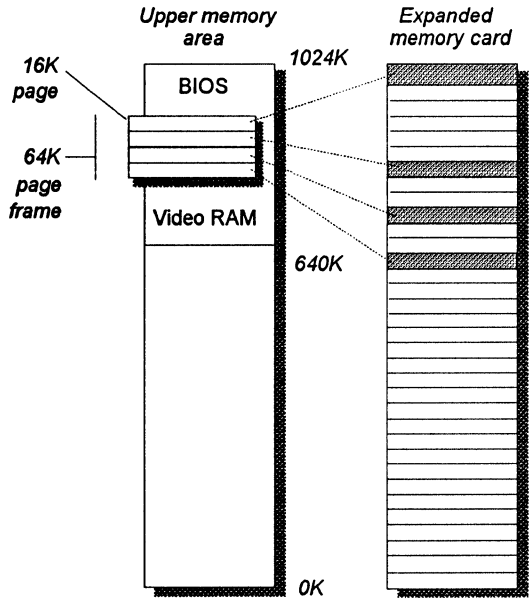
Figure 5.2

The upper memory area (640K to 1024K)

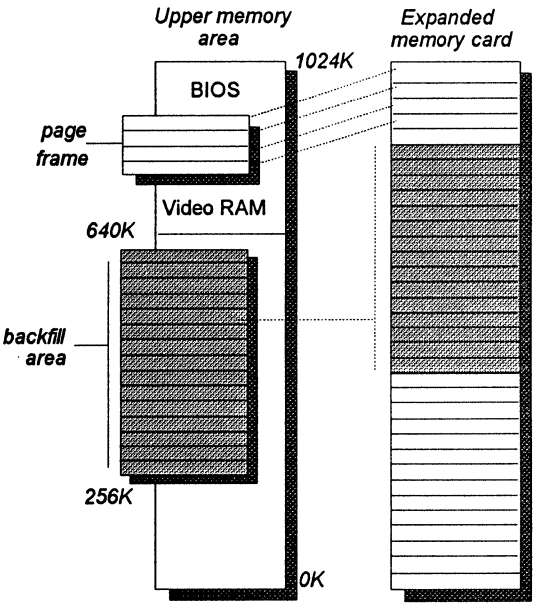
Figure 5.3 shows some key differences between the LIM 3.2 and LIM 4.0 specifications. If a non-Windows application requires expanded memory, usually its documentation will describe the LIM version under which it works.

Figure 5.3

LIM 3.2 EMS allows applications to move four continguous 16K pages of data, forming a 64K page frame



LIM 4.0 EMS allows applications to move up to 64 16K pages and allows backfilling in conventional memory from 640K down to 256K



Bank Switching in Expanded Memory

Application programs take advantage of expanded memory by making calls to the expanded memory manager (EMM) to request blocks of expanded memory. Bank switching is the process of mapping or temporarily assigning memory from a pool of expanded memory to an empty address space in the upper memory area.

For example, if an application needs more memory for data, it contacts the EMM, which allocates expanded memory 16K at a time. The application writes up to 64K of data to the address allocated by the EMM, then requests another 64K allocation. This request is the bank switch request. The EMM allocates another 64K of memory at a different address while tracking where the first 64K of data was placed.

The EMM continues this bank switching activity, 16K at a time, until the application makes no more request for memory space. With bank switching, the EMM can manipulate several megabytes of data through the single 64K space.

Backfilling in Expanded Memory

With LIM 4.0, the expanded memory manager can backfill conventional memory addresses from 640K down to approximately 256K, using memory that is usually reserved for MS-DOS, utilities, and TSRs. With backfilling, the system can manipulate large amounts of data, which is particularly important for multitasking. A rotating pool of backfilled memory allows entire applications and considerable data to be loaded into memory at once, greatly speeding operations.

One trick to maximize the benefit of backfilling on 80286 machines is to disable as much of the motherboard memory as possible (down to the 256K level) and let the expanded memory card supply that memory. Because the expanded memory card is supplying the memory, it can bank switch freely and make the best use of available memory.

For 80386 machines, you can convert extended memory into expanded memory by using an expanded memory manager such as EMM386.EXE. If you are using an expanded memory card on an 80386 machine, read your manual carefully before trying to backfill. Not all memory cards have the register support to supply more than four 16K pages. Also, because Windows 3.1 supports extended memory directly, backfilling expanded memory will not give you any advantage for running Windows.

Difficulties with Expanded Memory

Expanded memory uses the upper memory area, which it shares with hardware adapters such as SuperVGA video cards, network cards, 3270 emulation cards, and ESDI disk controllers. Several potential difficulties can arise when expanded memory contends for address space with the adapters:

- **Lack of free space.** The primary problem for the EMM is finding at least 64K of contiguous free space in which to place the page frame. Even though LIM 4.0 technically doesn't require a 64K page frame, most expanded memory applications will not use expanded memory unless the 64K page frame is present. Frequently, the address areas of various adapter cards need to be shuffled around to free enough space for a contiguous 64K page frame. This process gets complicated with boards such as the IBM 3270 emulator, which has a nonmovable address in most machines.
- **Mapping conflicts.** Most expanded memory managers such as EMM386 use a search algorithm to find unused memory between the hexadecimal addresses C000 through DFFF. The EMM then uses this memory as page frames. Some cards don't reserve their address space until the card is accessed, so the EMM can inadvertently map expanded memory pages on top of the adapter, causing crashes and intermittent operation. This problem is fairly rare, because the page search routine can locate almost all popular adapters.

If problems occur, start by disabling expanded memory to see if a page conflict is causing the problem. If the problem goes away, then the EMM needs to be told to exclude the adapter address from consideration as a page location. The adapter might also have to be moved. You do this in different ways with different EMMs. For information on how to exclude an address range, consult the documentation for your expanded memory manager.

For more information about taking advantage of memory in the UMBs, see "Optimizing Use of the UMBs" in Chapter 6, "Tips for Configuring Windows 3.1." For related troubleshooting tips, see "Troubleshooting EMS Memory Problems" in Chapter 13, "Troubleshooting Windows 3.1."

Windows Standard Mode and Memory

Standard mode is the normal operating mode for Windows on 80286 machines, providing direct access to extended memory.

Conventional memory takes no special consideration for Windows in standard mode. Windows running in standard mode treats the total free conventional and extended memory as one contiguous memory block. This section describes how standard mode Windows takes advantage of extended memory.

Standard mode does not use expanded memory for Windows operations, but can work with expanded memory for non-Windows applications running under Windows, as described at the end of this section.

For information about entries in SYSTEM.INI that affect performance in Windows standard mode, see the description of the entries in the **[standard]** section in Chapter 4, "The Windows Initialization Files."

Extended Memory and Standard Mode

Flowchart 5.9 Cannot Run Windows in Standard Mode

Standard mode Windows accesses extended memory directly through HIMEM.SYS (or through a third-party XMS driver), providing the total free conventional and extended memory for Windows applications to use. Any non-Windows applications that uses extended memory can run under standard mode Windows.

Standard mode Windows takes advantage of extended memory by:

- Performing code caching with Windows applications.
- Swapping non-Windows applications to extended memory.

Using Extended Memory for Code Caching

Standard mode Windows can speed up its operations by caching code in extended memory through HIMEM.SYS. To perform code caching, Windows takes advantage of certain attributes of Windows applications. The code for each Windows application is divided into segments with specific attributes, including:

- The Movable code segment, which means that the segment can be moved around in memory.
- The Discardable code segment, which means that the segment can be overwritten and then reloaded from disk when necessary.
- The Swapable code segment, which can be swapped to the hard disk.

Each Windows application keeps a minimum amount of code (its "swapsize") loaded in memory. If Windows runs out of memory when a new application executes, Windows discards part of an old application from active memory and overwrites it with new code from the new executable file.

Using Extended Memory with Non-Windows Applications

Non-Windows applications that use extended memory can run under standard mode Windows. The amount of extended memory that the application requires should be specified in the application's program information file (PIF).

For example, your system might have 2048K of extended memory, and you might specify in the PIF that the non-Windows application requires 1024K of extended memory. If standard mode Windows is already using all of the extended memory when you run the non-Windows application, then the information in the first megabyte of extended memory is swapped to disk, and the non-Windows application is given access to the newly free extended memory. When you switch back to Windows from the non-Windows application, the data in the original 1024K of extended memory is reloaded from disk.

Because this file-swapping process can be slow, don't request any more extended memory in a PIF than is absolutely necessary to run the non-Windows application.

Some non-Windows applications use MS-DOS Extender technology such as VCPI or DPMI to run in protected mode. If you are running such applications under standard mode Windows, you must allocate extended memory in the application's PIF, as described in Chapter 7, "Setting Up Non-Windows Applications." For more information about VCPI, see "DPMI and VCPI Specifications" later in this chapter.

Expanded Memory and Standard Mode

Windows running in standard mode does not use expanded memory at all for its operations. But non-Windows applications running under standard mode Windows can access expanded memory if the system has a physical EMS card such as AST RAMPage! or the Intel Above Board.

The expanded memory manager for the EMS card uses upper memory blocks (UMBs) in the upper memory area. If you suspect a UMB conflict is causing a problem in your system, remove the expanded memory manager to see if that solves the problem. For more information, see "Troubleshooting EMS Memory Problems" in Chapter 13, "Troubleshooting Windows 3.1."

Because non-Windows applications running under standard mode can only use expanded memory with a physical EMS card, an external 386 expanded memory manager such as EMM386.EXE cannot provide the required expanded memory support in Windows. However, an external 386 memory manager can provide expanded memory support for non-Windows applications when you aren't running Windows.

If you want to use memory in the upper memory area, or if you have an application that requires expanded memory but you don't have a physical EMS card, you must run Windows in 386 enhanced mode. For more information about using EMM386.EXE, see Chapter 6, "Tips for Configuring Windows 3.1."

Windows 386 Enhanced Mode and Memory

Windows 386 enhanced mode is the normal operating mode for systems with 80386 and higher processors.

Flowcharts 5.1 and 5.2 Cannot Run 386 Enhanced Mode

When Windows runs in 386 enhanced mode, it adds up the amount of free conventional and extended memory and treats the total amount as an available block of memory, in much the same way as standard mode. The entries in the **[386enh]** section of SYSTEM.INI that control how Windows allocates conventional memory are **PerformBackfill=**, **ReservePageFrame=**, **WindowKBRequired=**, and **WindowMemSize=**. For more information, see these entries in Chapter 4, "The Windows Initialization Files."

Although 386 enhanced mode doesn't use expanded memory for Windows operations, it can simulate expanded memory for use by non-Windows applications, as described later in this section.

You can also create a swap file so that Windows 386 enhanced mode can take advantage of the virtual memory capabilities of the 80386 and higher processors, as described at the end of this section.

WINA20.386 and 386 Enhanced Mode

The MS-DOS 5.0 Setup program installs a virtual device driver that resolves conflicts between Windows 3.0 and MS-DOS 5.0 when both try to access the HMA. This driver is a read-only file named WINA20.386, which is automatically installed in your root directory. Windows 3.0 will not run in 386 enhanced mode without this file.

- If you upgrade to Windows 3.1, or if you never run 386 enhanced mode with Windows 3.0, you can remove the WINA20.386 file by changing its read-only attribute, then deleting it in the usual way.

- If you are running Windows 3.0 and you move the WINA20.386 file to a different directory, you must ensure that Windows can find this file by adding a **switches=/w** command to CONFIG.SYS and also adding a **device=[*new path*]\wina20.386** entry to the **[386enh]** section of SYSTEM.INI.

Extended Memory and 386 Enhanced Mode

Windows 386 enhanced mode uses HIMEM.SYS, the extended memory device driver, to load itself and its drivers into extended memory. As in standard mode, Windows 386 enhanced mode provides the total free conventional and extended memory for Windows applications to use directly. So code caching for Windows applications can also be performed in 386 enhanced mode.

Windows 386 enhanced mode also allows non-Windows applications to run in protected mode if the application uses the DOS Protected Mode Interface (DPMI) specification (as described later in this chapter). Lotus 1-2-3 version 3.1 is an example of a DPMI application.

Windows 386 enhanced mode provides access to extended memory for non-Windows applications by creating virtual machines up to 640K in size, or the size defined by the **CommandEnvSize=** entry in the **[NonWindowsApp]** section of SYSTEM.INI.

Each virtual machine inherits the environment present before you started Windows. This means that every driver and terminate-and-stay-resident program (TSR) loaded before running Windows consumes memory in every subsequent virtual machine. The memory available within each virtual machine under 386 enhanced mode is slightly less than the free memory available at the command prompt before you start Windows, depending on your system configuration.

When creating virtual machines for non-Windows applications, Windows 386 enhanced mode uses the upper memory area for two purposes:

- To allocate translation buffers for protected-mode API calls for MS-DOS and the network.
- To place the page frame for expanded memory (if required).

Frequently, all of the free pages in the upper memory area are used by 386 enhanced mode. The memory conflicts that can result are discussed in the next section.

Expanded Memory and 386 Enhanced Mode

Windows 386 enhanced mode does not use expanded memory for itself, and Windows applications don't need expanded memory, because they run in protected mode and can access extended memory directly. However, Windows 386 enhanced mode can create expanded memory for use by non-Windows applications such as Lotus 1-2-3 that require or can take advantage of expanded memory.

Windows 386 enhanced mode automatically provides expanded memory for non-Windows applications that require it when you run such applications under Windows. It cannot provide this memory, however, if you load EMM386.EXE with the **noems** switch. Use the **ram** switch when loading EMM386.EXE in CONFIG.SYS, or use the **x=mmmm-nnnn** parameter to allocate enough space in the upper memory area for Windows to create an EMS page frame.

Note The expanded memory required by a non-Windows application should be allocated with PIF parameters as described in Chapter 7, "Setting Up Non-Windows Applications."

Page-Frame Conflicts in 386 Enhanced Mode

Windows 386 enhanced mode provides additional page frames for LIM 4.0 expanded memory in all virtual machines. But most non-Windows applications use only the 64K page frame itself, not the additional bankable pages in conventional memory that LIM 4.0 EMS supplies. So to use expanded memory to run non-Windows applications, you must have a contiguous 64K page frame, made up of four contiguous 16K pages in the upper memory area. The key issue, therefore, for expanded memory under Windows is page-frame conflicts.

The adapters installed on a system can break up the free area in the upper memory area so that there is no 64K contiguous area to place the page frame, and hence no free expanded memory for running non-Windows applications. If this problem occurs, you might have to rearrange the adapter memory locations.

This is easiest to do on a Micro Channel machine such as the IBM Personal System/2, which allows you to change adapter memory locations by booting with the PS/2 Reference Disk and choosing Change Configuration. A similar procedure is available on most Extended Industry Standard Architecture (EISA) bus machines, such as the Compaq SystemPro and HP Vectra 486.

For Industry Standard Architecture (ISA) bus machines, such as the IBM AT and Compaq 386, you might have to open the case and flip DIP switches on the cards to change their memory addresses. Refer to your hardware manual,

and use Figure 5.2 (page 232) as a reference when readdressing adapters to open a 64K page frame.

You can also disable expanded memory entirely (and 64K page frame support) in Windows 386 enhanced mode by setting **NoEMMDriver=yes** in the **[386enh]** section of SYSTEM.INI.

Other related entries in the **[386enh]** section of SYSTEM.INI are **EMMSize=** and **IgnoreInstalledEMM=**. For more information, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files."

Placing Translation Buffers in the Upper Memory Area

In 386 enhanced mode, Windows allocates buffers in the upper memory area to translate MS-DOS and network application program interface (API) calls from Windows protected mode to MS-DOS real mode. (Because the upper memory area is within the first megabyte of address space, it can be accessed by MS-DOS in real mode on an 80386 and higher processor.) Ideally, there will be enough free space in the upper memory area to place both the translation buffers and any expanded memory page frame required. But on many systems there isn't enough room, and you must choose to eliminate the expanded memory page frame or allocate the translation buffers in conventional memory (instead of in the upper memory area).

If the translation buffers are allocated in conventional memory, they take up memory in every virtual machine Windows creates, leaving less space in the virtual machines to run non-Windows applications. To compound the problem, the translation buffers can be allocated either in the upper memory area or in conventional memory, but never half-and-half.

To specify a preference, set the value for the **ReservePageFrame=** entry in the **[386enh]** section of SYSTEM.INI. If **ReservePageFrame=true** (the default), then Windows allocates the page frame first and the translation buffers second. Usually, on machines with full UMBs, the translation buffers are forced into conventional memory, but this lets you use expanded memory for non-Windows applications.

If **ReservePageFrame=false**, the translation buffers are allocated in the UMBs first, followed by the page frame if there is still room. This setting gives you the most free memory in virtual machines, but you might not have enough expanded memory for non-Windows applications.

Controlling UMB Mapping in 386 Enhanced Mode

You can control the placement of expanded memory page frames and the translation-buffer mapping with the **EMMExclude=** or **ReservedHighArea=** entries in the **[386enh]** section of SYSTEM.INI. To explicitly exclude an area of the upper memory area from mapping by Windows 386 enhanced mode, set a value for **EMMExclude=**, for example, **EMMExclude=E000-EFFF**. Because there is no standard for hardware implementation of the E000-EFFF area, it is frequently necessary to exclude this range, so that 386 enhanced mode can function properly. Windows 386 enhanced mode detects and excludes the area for most adapter cards automatically.

Any values set with the **x=** switch in the line that loads EMM386.EXE in CONFIG.SYS will override the value set for **EMMExclude=** in SYSTEM.INI.

The **ReservedHighArea=** entry provides the same support, but for 4K ranges, rather than 16K. Other entries related to UMB mapping in the **[386enh]** section of SYSTEM.INI are **EMMInclude=**, **EMMPageFrame=**, **EMMSize=**, and **UseableHighArea=**. If you suspect conflicting use of the upper memory area, use **EMMExclude=**. Because Windows will use all free pages in the upper memory area automatically, there are few uses for **EMMInclude=**, **EMMPageFrame=**, and **UseableHighArea=**. For more information, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files," and see also "Troubleshooting EMS Memory Problems" in Chapter 13, "Troubleshooting Windows 3.1."

Windows/386 2.x did not use the E000-EFFF area of the adapter segment unless specifically instructed to do so. Windows 386 enhanced mode uses this segment unless the machine identifies itself as a PS/2. As a side note, most of the entries in the **[386enh]** section of SYSTEM.INI that begin with the letters "EMM" control both placement of the expanded memory page frames and the translation-buffer mapping. The letters "EMM" are used only for backward compatibility; except for the **EMMPageFrame=** entry, they no longer apply only to the expanded memory page frame. The **LastEMMSeg=** parameter used in Windows/386 2.x has been dropped for Windows 3.x.

About 386 Expanded Memory Managers

An expanded memory manager such as EMM386.EXE can provide expanded memory for non-Windows applications on 80386 and higher machines without a physical EMS card when you aren't running Windows. For more information about EMM386.EXE, see Chapter 6, "Tips for Configuring Windows 3.1," in the *Windows Resource Kit;* see also "Freeing Expanded Memory" in Chapter 12 of the *MS-DOS 5.0 User's Guide and Reference*.

Some 386 expanded memory managers such as EMM386.EXE and CEMM.EXE allow Windows to turn them off when Windows is run. CEMM.EXE requires that no expanded memory be in use when you start Windows. (That is, set **NoEMMDriver=yes** in the **[386enh]** section of SYSTEM.INI.)



Both Windows 3.1 and MS-DOS 5.0 support the upper memory area mechanisms defined for the LIM 3.2 and LIM 4.0, so 386 expanded memory managers such as EMM386.EXE, 386MAX.SYS, and QEMM.SYS can load network drivers and other software devices in the upper memory area and still run with Windows 386 enhanced mode.

Virtual Memory and 386 Enhanced Mode

Virtual memory has been widely used for years with mainframes, but first came to PCs with the introduction of the IBM/Microsoft OS/2 operating system. Windows 386 enhanced mode goes beyond OS/2 to offer virtual memory using the special demand-paging capabilities of the Intel 80386 processor.

When virtual memory is used with Windows 386 enhanced mode, some of the program code and data are kept in physical memory while the rest is swapped to the hard disk in a swap file. Whenever a reference is made to a memory address, it can be used without interruption if the information is currently in physical memory. If the information isn't in physical memory, a page fault occurs and the Windows Virtual Memory Manager (VMM) takes control, pulling the required information back into physical memory and, if necessary, swapping other information to the disk. All of this activity is invisible to the user, who only sees some hard disk activity.

Windows applications can use virtual memory without being specially written to take advantage of it, because Windows handles the memory management, allocating however much memory the application requests. With Windows managing virtual memory, you will see much more memory available than is installed in your machine when you choose About Program Manager or About File Manager from the Help menu.

A major benefit of using virtual memory is that you can run more programs simultaneously than your system's physical memory would usually allow. The drawbacks are the disk space required for the virtual memory swap file and the decreased execution speed when page swapping is required. However, it's usually better to run a program slowly in virtual memory than to not be able to run it at all.

Creating Swap Files for Virtual Memory



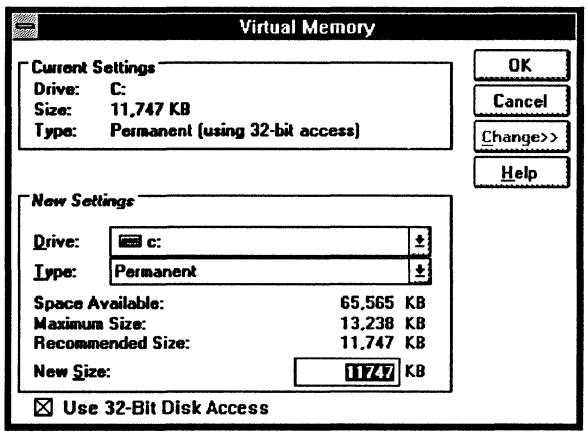
In Windows 3.1, you can create a swap file for virtual memory during Setup, or you can choose the 386 Enhanced icon in Control Panel to change the swap file at any time. For details about using the 386 Enhanced Mode dialog box, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

Figure 5.4

Virtual Memory dialog box

To display this dialog, click the Virtual Memory button in the 386 Enhanced dialog

Click the Change button to display this extended dialog box



You can create either a temporary or a permanent swap file. A permanent swap file improves the speed of the Windows virtual memory system because the file is contiguous, so accessing it requires less overhead than the normal MS-DOS file created for a temporary swap file.

- A temporary swap file named WIN386.SWP is created on the hard disk while Windows is running, then deleted automatically when you exit Windows. This swap file is not a hidden or system file, and it can shrink or grow in size as necessary. The entry for **PagingFile=** in the **[386enh]** section of SYSTEM.INI defines the filename and path for the temporary swap file. You need about 1.5 MB of free hard disk space on the paging drive for a temporary swap file.
- A permanent swap file is a hidden file named 386PART.PAR, which has a system attribute and is always created in the root directory of the specified drive. Windows also creates a read-only SPART.PAR file in the WINDOWS directory that tells Windows where and how large the permanent swap file is. Because a permanent swap file must be contiguous, you cannot create a permanent swap file bigger than the largest contiguous free segment of your hard disk. You cannot create a permanent swap file if Stacker is running on your system.

You can specify the type and size of a swap file and the drive where it's located in the Virtual Memory dialog box. A permanent swap file is always created in the

root directory. But you can specify a subdirectory as the location for a temporary swap file as a **PagingFile=** value in the **[386enh]** section of SYSTEM.INI.

When you install Windows, Setup checks whether your hard-disk controller is compatible with 32-bit disk access. If so, the 32-Bit Disk Access check box appears. Select this check box if your system has only a small amount of free memory and you want to increase performance for the MS-DOS Prompt. When this option is checked, you can run more instances of MS-DOS Prompt and switch between them faster. If you have multiple instances of MS-DOS Prompt running and the applications in them all access disk drives, the access time is faster with 32-bit disk access. For a technical discussion of this features see "FastDisk: An Introduction to 32-Bit Disk Access" in Appendix D, "Articles."

The Windows virtual memory utility that creates swap files supports only hard disks that use 512-byte sectors. If 512-byte sectors are not being used, this indicates a nonstandard configuration, such as a third-party driver. Never create a permanent swap file on a drive that uses a partitioning driver, with the exception of the Compaq ENHDISK.SYS utility.

Before you can create a large permanent swap file, you should compact your hard disk with a disk compacting utility. If an error message reports that your swap file is corrupted, delete the current swap file and create a new one.

The related entries for swap files in the **[386enh]** section of SYSTEM.INI are **MaxPagingFileSize=**, **MinUserDiskSpace=**, **PagingDrive=**, and **PagingFile=**. For more information, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files."

Note Do not attempt to create a swap file on a RAM disk. A swap file created on a RAM disk is self-defeating, because you sacrifice physical memory to create virtual memory.

Demand Paging and Virtual Memory Management

Windows 386 enhanced mode manages virtual memory as a demand-paged system that uses a virtual memory manager (VMM) and a pageswap device (which is built into the WIN386.EXE file). This means that pages of data are brought into physical memory when they are referenced, and the system does not try to predict which pages will be required in the future.

The Windows VMM maintains the virtual memory page table that lists the pages currently in physical memory and those swapped to disk. Because 386 enhanced mode is a multitasking environment, the VMM page table also lists which memory pages belong to which processes. When the VMM needs a page

not currently in physical memory, it calls the pageswap device, which allocates virtual memory and maps pages into and out of physical memory.

Some virtual memory systems rely on program segmentation to do their work. Although the code for a Windows application is segmented as described earlier, Windows virtual memory management is not related to this segmentation. All virtual and physical memory is divided into 4K pages, and the system is managed on this basis. Page mapping starts at 0000K and works its way up.

Two kinds of pages can be allocated: physical pages and virtual pages. The possible amount of physical pages is the amount of physical memory in the machine divided by 4K. Memory allocated to an application is made up of virtual pages, and at any time a virtual page can be in physical memory or swapped to the hard disk.

The entries in the **[386enh]** section of SYSTEM.INI that control paging for virtual memory are:

LocalLoadHigh=	**Paging=**	**SysVMEMSLimit=**
MaxBPs=	**PagingDrive=**	**SysVMEMSLocked=**
MaxPhysPage=	**PagingFile=**	**SysVMV86Locked=**
MinUnlockMem=	**PageOverCommit=**	**SysVMXMSLimit=**

For more information, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files."

The LRU Algorithm for Virtual Memory Management

The Windows VMM swaps pages using a Least Recently Used (LRU) page replacement algorithm. This means pages that have not been accessed for the longest period of time are the first ones to be swapped to the disk.

The VMM page table contains flags for the Accessed and Dirty attributes of each page. *Accessed* means that a process has made a reference to the page since it was originally loaded. *Dirty* means that a "write" has been made to the page since it was loaded. Because a "write" qualifies as an "access," the Dirty attribute implies the Accessed attribute.

If physical memory space cannot be found when a process requests additional memory, Windows uses the LRU algorithm to decide which pages to swap to the hard disk to fulfill the request. This decision is a three-step process:

1. Windows scans the VMM page table looking for pages with neither an Accessed nor a Dirty attribute. During the scanning process, Windows clears the Accessed attribute from all the pages.

2. If Windows finds enough pages meeting the Not Accessed/Not Dirty requirement, it swaps those pages to the hard disk and gives the resulting free memory to the process.
3. If Windows cannot find enough pages the first time, then it repeats the scan. Theoretically, more pages will meet the requirements because the Accessed attribute was cleared in the first scan. If the second scan doesn't find the required pages, then Windows swaps pages to the hard disk regardless of their attributes.

The related entries in the **[386enh]** section of SYSTEM.INI are **LRULowRateMult=**, **LRURateChngTime=**, **LRUSweepFreq=**, **LRUSweepLen=**, **LRUSweepLowWater=**, and **LRUSweepReset=**. For more information, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files."

Swapping Pages to a Network Drive

We recommend that you do not swap pages to a network drive. Paging to a network drive is possible, but it's extremely slow. If you must page to a network drive, use a permanent swap file. Also, the directory must not have an MS-DOS read-only attribute, and you must have both create and write access to the directory. Do not set the value for **PagingDrive=** or **PagingFile=** in SYSTEM.INI to a Novell network drive, because Novell networks are not compatible with MS-Net Redirector.

Other Memory Management Issues

This section provides brief information on various memory management issues:

- DPMI and VCPI specifications
- MS-DOS and Windows 3.1
- Memory and startup requirements
- System resources and memory

DPMI and VCPI Specifications

The DOS Protected Mode Interface (DPMI) was developed by a group of industry leaders. Several members of the DPMI committee also helped create the Virtual Control Program Interface (VCPI). DPMI is primarily a creation of Microsoft, and VCPI was formulated primarily by Phar Lap Systems. DPMI and VCPI solve two different problems.

Applications that use MS-DOS Extenders can execute code in the protected mode of the 80286 or 80386 processor. DPMI provides a standard method for

such applications to switch the 80286 processor to protected mode and to allocate extended memory. Hundreds of applications use various types of MS-DOS Extenders, and those that do not already support DPMI require minor modifications to do so.

VCPI provides an interface that allows applications using MS-DOS Extenders on 80386 machines to run simultaneously with 386 expanded memory managers. For example, QEMM.EXE, 386MAX.EXE, and CEMM.EXE support the VCPI specification. Windows 3.1 supports VCPI in both standard mode and 386 enhanced mode. Windows 3.0 does not support VCPI.

MS-DOS 5.0 and Windows 3.1



Many of the changes in MS-DOS 5.0 make it a more robust platform for Microsoft Windows, enhancing its ability to make the best use of your system's memory. For the best performance from Windows 3.1 (and other applications), we suggest that you upgrade your PC's operating system to MS-DOS 5.0 if you haven't already done so.

On 80386 and 80486 PCs, with MS-DOS 5.0 you can load memory-resident programs such as device drivers, TSRs, and network software into the upper memory area, thereby freeing conventional memory. To do this, you must load HIMEM.SYS and EMM386.EXE, then load the memory-resident programs using the **devicehigh** command in CONFIG.SYS and the **loadhigh** command in AUTOEXEC.BAT.

For more information about taking advantage of MS-DOS 5.0 to optimize your system configuration, see Chapter 6, "Tips for Configuring Windows 3.1."

Running Windows Standard Mode with MS-DOS 5.0

- Use the Windows Task List to swap applications (or press ALT+TAB). Using the MS-DOS 5.0 Task Swapper while running Windows is redundant, incurring unnecessary conventional memory overhead.
- Load MS-DOS 5.0 into the HMA, regardless of whether your PC is an 80286, 80386, or 80486. This will provide more conventional memory for non-Windows applications running under Windows standard mode.

Running Windows 386 Enhanced Mode with MS-DOS 5.0

- Load MS-DOS 5.0 into the HMA, and load whatever else fits into the upper memory area. The conventional memory you free by loading MS-DOS 5.0 into the HMA is also free in every virtual machine, giving more memory to every non-Windows application you run in Windows 386 enhanced mode.
- Use the Task List in Windows 3.1 to switch between applications (or press ALT+TAB). Windows 386 enhanced mode allows multitasking, so that multiple applications can run in the background. The MS-DOS 5.0 Task Swapper can only task-switch, which means that the swapped application is suspended.

Memory and Windows Startup Requirements

Flowchart 1.1 System Requirements

Windows starts automatically in the appropriate Windows operating mode, depending on your system configuration. However, you can start Windows in a particular operating mode by using one of these command-line switches:

- **win /s** to run in standard mode
- **win /3** to run in 386 enhanced mode

Standard Mode Startup Requirements

For Windows to start automatically in standard mode, the system must have:

- 80286 or higher processor
- 256K of free conventional memory
- 192K of free extended memory
- An XMS driver such as HIMEM.SYS already loaded

Conventional and extended memory requirements are mutually dependent for standard mode and are not fixed.

386 Enhanced Mode Startup Requirements

For Windows to start in 386 enhanced mode, the system must have:

- 80386 or higher processor
- 256K of free conventional memory
- 1024K of free extended memory
- An XMS driver such as HIMEM.SYS already loaded

Windows 386 enhanced mode requires between 580K and 624K combined conventional and extended memory to run. A typical installation requires a minimum of 192K free conventional memory at the command prompt plus sufficient extended memory available to run in 386 enhanced mode. Windows can start under low memory in 386 enhanced mode because it provides virtual memory support, but it can be extremely slow because of the extra disk swapping that Windows must perform.

All numbers are approximate and can vary widely depending on the Windows device drivers present, the MS-DOS version, the display adapter, and other factors. For example, on Compaq 386 machines, 128K of extended memory is recovered from shadow RAM. Memory requirements take into account memory that can be recovered from SMARTDrive, down to the minimum cache size specified.

Windows checks for a minimum of 1 MB of free extended memory before it automatically starts in 386 enhanced mode. If it finds less, it tries to run in standard mode. On an 80386 with 2 MB or less system memory, if Windows doesn't find enough free extended memory, it runs in standard mode. To free more extended memory so that you can run in 386 enhanced mode, you might try reducing the amount of extended memory that SMARTDrive uses by setting its **MinCacheSize** parameter to **0**.

The entries in the **[386enh]** section of SYSTEM.INI that control how Windows allocates conventional memory for startup and for use in 386 enhanced mode are **SysVMEMSRequired=**, **SysVMXMSRequired=**, **WindowKBRequired=**, and **WindowMemSize=**. For details, see the descriptions of these entries in Chapter 4, "The Windows Initialization Files."

Memory and the Windows System Resources

Flowchart 5.8
Out-of-Memory Errors

The Help About dialog box in Program Manager and File Manager (and in many other Windows applications) shows the percentage of free system resources and the amount of free memory. The system resources percentage reflects the memory used by the core Windows internal structure.

Three core files make up the part of Windows that runs Windows applications:

- The kernel file (KRNL286.EXE or KRNL386.EXE) loads and executes Windows applications and handles their memory management.
- GDI.EXE manages graphics and printing.
- USER.EXE controls user input and output, including the keyboard, mouse, sound driver, timer, communications ports, and window management.

In Windows 3.x, these files are in the Windows SYSTEM subdirectory. (In Windows 2.x these modules were linked into the files WIN200.BIN and WIN200.OVL, so you didn't see them in the WINDOWS directory.)



Both GDI and User have storage areas called heaps, which are limited to 64K in size. GDI has a local heap; User has a menu heap and a user heap, each with 64K storage space. The available system resources reflect the remaining free percentage after combining the GDI local heap and the user and menu heaps in User. This increases by 64K the amount of heap space that was available in Windows 3.0. In Windows 3.1, Program Manager icons are handled separately and do not use the User heap space.

To see how much of the system resources a particular application uses, note the amount of system resources available before and after the application runs. (Choose About Program Manager from the Help menu and check the amount of system resources listed in the dialog box.)

Although Windows 3.1 allows you to run many more simultaneous Windows applications than any previous Windows version, you may get an out-of-memory message that indicates your system is low on system resources. This is because every window and sub-window created requires User and GDI local heap space. The system resources can be exhausted if enough objects are created by the Windows applications.

An important element of memory management for Windows applications that is not included in the system resources percentage is the number of selectors. A selector is a memory pointer that is consumed with each memory allocation made by a Windows application. If a Windows application allocates many small data objects, it is possible to run out of selectors, resulting in an out-of-memory message.

The efficiency with which a Windows application handles its data objects can help in this situation. If you experience a chronic problem with a particular application while few other applications are loaded, contact the application vendor so the company becomes aware of the problem and corrects it.

For more information about low-memory and out-of-memory conditions, see Chapter 13, "Troubleshooting Windows 3.1."

SMARTDrive 4.0: A Technical Discussion



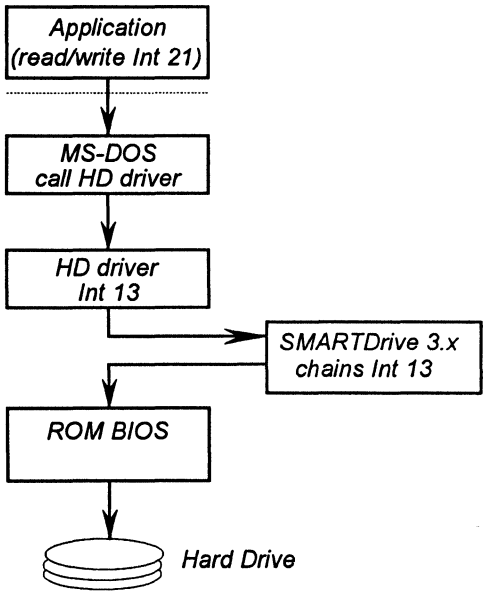
This section describes the new MS-DOS disk cache, SMARTDrive 4.0, which replaces SMARTDrive version 3.x. The section is written for experienced technical users who want a deeper understanding of the utility.

About SMARTDrive 4.0

SMARTDrive 3.x is a read-only track cache that caches on a track basis and for read operations only. The internal data structures are tied to the logical geometry of the disk. It caches at the ROM BIOS Int 13 level and uses the BIOS specified disk geometry to decide the size of its caching element (that is, track size). This leads to problems (described below), so Microsoft chose to implement a new cache in SMARTDrive 4.0.

Figure 5.5

SMARTDrive 3.x



SMARTDrive 4.0 is designed as a block-oriented disk cache. It hooks into the system at the MS-DOS device driver level rather than the ROM BIOS Int 13 level. Each block device driver on the MS-DOS device driver chain is "front-ended" by a SMARTDrive 4.0 module that provides the actual caching. This yields the following benefits:

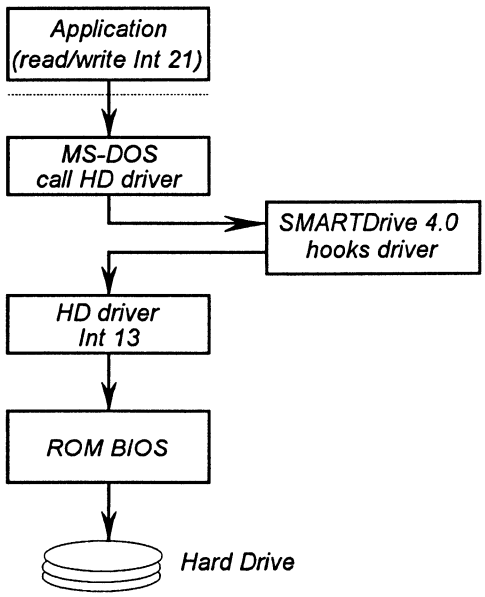
- **Independent of Int 13 interface.** Many device drivers do not use the Int 13 interface. This means that SMARTDrive 4.0 can cache these devices where SMARTDrive 3.x could not. Examples are Bernoulli, some hard cards, and many SCSI and WORM drives.

- **Independent of disk geometry.** Some disk managers and disk controllers use a disk geometry mapping scheme that causes the "logical" geometry (that is, what MS-DOS sees) to be different from the physical geometry. Examples are many PS/2 systems, Ontrack Disk Manager, and several disk controllers. Int 13-based caches are sensitive to this and often have problems. For example, Ontrack Disk Manager will actually change the ROM BIOS-specified disk geometry on the fly and thus confuse SMARTDrive 3.x.

  An API determines the true geometry, but this requires disk-manager detection and generally complicates matters. Often logical tracks will actually cross physical track boundaries, which then causes track caches to incur performance penalties (intertrack seeks and rotational latencies). Also, to get around the ROM BIOS 1024 cylinder limitation, disk managers and controllers will "fold" multiple tracks into one logical track. This yields the above problem, as well as forcing track caches to have a very large track buffer. In some cases, this is as large as 31.5K and must reside in low memory. The design of SMARTDrive 4.0 eliminates the geometry mismatch problem.

Figure 5.6

SMARTDrive 4.0



SMARTDrive 4.0 is also a write-behind cache. It adds significant performance improvement where files are being written and implements a "valid" bits scheme to avoid thrashing the disk in case of random access I/O.

SMARTDrive 4.0 currently uses a FIFO replacement algorithm and implements a shrink algorithm that frees memory for Windows in a way similar to SMARTDrive 3.x. The difference is that SMARTDrive 4.0 watches for the Windows startup broadcast while SMARTDrive 3.x provides an IOCTL

interface. The net effect is identical, but SMARTDrive 4.0 is much simpler. When Windows quits, the process is reversed and the memory is reacquired by SMARTDrive 4.0.

The following table summarizes the SMARTDrive 3.x design problems, what other disk-cache software does, and the solutions employed in SMARTDrive 4.0.

Feature	*SMARTDrive 3.x*	*Other disk caches*	*SMARTDrive 4.0*
Read-Only Cache	Causes writes to proceed at the same or slower speed as with no cache.	All others solve this.	Lazy write caching. Valid bits to avoid thrashing.
Track Cache	Tied to "logical" disk geometry, which disk managers and some disk controllers change, causing a "single" track read to actually be multiple — performance hit.	All others have configurable cache element size.	Support configurable cache element size.
	Some disks have very large track sizes (56 on some mapping controllers), which forces a very large track buffer into low memory and thus a large low memory footprint.	Same	Same
	Internal data structures and algorithms are tied to the size of track and thus it is very hard to retrofit variable block size.	Same	Same
Int 13 based	Many popular device drivers do not go through Int 13 interface and therefore don't get cached. Bernoulli, WORM drives, and SCSI are some examples of non-Int 13 devices that SMARTDrive does not cache.	Mixed. Some cache Int 13; some cache at device driver level.	Front-end MS-DOS device drivers. This means that block device in MS-DOS will be cached.
	Disk manager drivers and some controllers cause a geometry mismatch.	Some DM drivers provide DDI, which Int 13 caches need to call (PC-KWIK).	Above solution eliminates need to know true geometry.

SMARTDrive 4.0: Frequently Asked Questions

Q: What is double buffering?

A: Certain disk controllers support a concept called bus mastering, where the actual disk controller takes over the bus to transfer data to or from system RAM. Some SCSI controllers have this feature. A problem occurs when running in the virtual 8086 mode that Windows 3.x virtual machines provide. Popular memory managers also use virtual 8086 mode. The read or write address that is passed to MS-DOS is often not the same as the actual physical memory address. This can cause data to be read from the wrong location or, worse, can cause data to be written to the wrong RAM. The result can be erratic system behavior.

Microsoft created a standard called Virtual DMA Services, which provides an interface that allows these bus master controllers to get the correct address and avoid the problem. However, some older bus master controller cards do not support this standard. So we have added a feature to SMARTDrive that provides a memory buffer with physical and virtual addresses that are the same, so we avoid the problem at the cost of 2.5K of conventional memory and a small amount of performance (the cost of moving the data to and from the buffer). This feature is used by placing the line **device=smartdrv.exe /double_buffer** in CONFIG.SYS. This only installs the double-buffer driver, not the cache. (The cache must be installed in AUTOEXEC.BAT.)

Q: Do I need double buffering?

A: The Windows 3.1 Setup program tries to determine whether your system needs double buffering, and if so, installs SMARTDRV.EXE in your CONFIG.SYS file for double buffering. Most disk controllers do not need double buffering. These include all MFM, RLL, and IDE controllers as well as many ESDI and SCSI devices. In the cases where Setup is unable to determine whether double buffering is needed, it will install the driver in CONFIG.SYS, possibly erring on the side of safety. We have added a feature to SMARTDrive to help you determine if double buffering is unneeded and to allow you to remove the driver. After your system is running with SMARTDrive loaded, type **smartdrv** at the command prompt and press ENTER. You will see something similar to the following screen.

```
Microsoft SMARTDrive Disk Cache version 4.0
Copyright 1991,1992 Microsoft Corp.
Cache size: 1,048,576 bytes
Cache size while running Windows: 1,048,576 bytes
Disk Caching Status
drive     read cache   write cache   buffering
A:           yes          no           no
B:           yes          no           no
C:           yes          yes          yes
D:           yes          yes           -
For help, type smartdrv /? at the MS-DOS prompt.
```

Notice the column labeled "buffering." For each drive that is being cached, it can have one of three values: **Yes** to indicate that double buffering is needed. **No** to indicate that buffering is not needed, and the "–" symbol to indicate that SMARTDrive has not yet determined the necessity of double buffering. If the buffering column has all No's in it, the double buffer driver is not needed.

Q: Why is SMARTDrive in both my CONFIG.SYS and AUTOEXEC.BAT files?

A: There are really two device drivers in a single file: a disk cache and a double buffer driver. See above for a description of double buffering. The cache component of SMARTDRV.EXE is installed in AUTOEXEC.BAT and the double buffer driver is installed in CONFIG.SYS.

Q: Does SMARTDrive work with Stacker?

Flowchart 1.8 Stacker

A: Yes. SMARTDrive works quite well with Stacker. SMARTDrive is aware of Stacker and will automatically cache the underlying drive that Stacker uses. This provides significantly better cache utilization by increasing the effective size of the cache by the compression ratio of Stacker. However, do not cache the actual "stacked" volume. Only the underlying (uncompressed) drive should be cached.

Q: Why doesn't my Stacker Volume show up in the SMARTDrive status screen?

A: This is because SMARTDrive is caching underneath Stacker. You should see the underlying drive letter listed.

Q: How can I make sure that data written to the disk is really on the disk and not still in the cache when I reboot my machine? Won't my data get lost when I reboot?

A: SMARTDrive goes to great lengths to avoid data loss. When it detects the CTRL+ALT+DEL reboot key sequence, SMARTDrive takes control and makes sure that all data has been written to the actual disk. You might see a box in the upper left corner of the display asking you to wait while this happens. SMARTDrive also writes all data to the disk when an application calls the MS-DOS reset disk function to make sure that all data in the MS-DOS buffers gets written to disk.

To force all data to be written to the disk, type **smartdrv /c** at the command prompt and press ENTER.

If you use a third-party program to reboot your machine from a batch file, you should make sure that you have the above line in the batch file before the reboot program. Failure to do so may cause loss of data.

Chapter

6 Tips for Configuring Windows 3.1

This chapter presents several tips for configuring Microsoft Windows 3.1, both for gaining optimal performance and for setting up Program Manager and File Manager.

Related information

- *Windows User's Guide:* Chapter 3, "Program Manager"; Chapter 4, "File Manager"; Chapter 14, "Optimizing Windows"
- *Windows Resource Kit:* Chapter 4, "The Windows Initialization Files," and Chapter 5, "Windows 3.1 and Memory Management"
- *Glossary terms:* upper memory area, upper memory blocks, Expanded Memory Specification, Extended Memory Specification, paging, virtual machines

Contents of this chapter

Tips for Optimizing Performance 258
Checklist for Optimizing Performance 258
Trading Speed for Hard Disk Space 262
Optimizing Use of the UMBs 263
Adjusting EMM386 for Optimal Performance 266
Tips for Using Third-Party Memory Managers 269
Optimizing with SYSTEM.INI 271
Tips for Configuring the Windows Shell 272
Customizing Program Manager 272
Customizing File Manager 277

Tips for Optimizing Performance

This section presents a summary of tips for achieving the best performance for Windows 3.1, plus specific information about using EMM386.EXE.

Before making changes to your system configuration, create a backup boot disk, as described in "Creating a 'Clean Boot' for Troubleshooting" in Chapter 13, "Troubleshooting Windows 3.1." For information that will help you solve specific performance problems with your system, see "Troubleshooting Windows Configuration" in Chapter 13.

Checklist for Optimizing Performance

Here is a summary of items related to optimizing your hardware and software configuration for both Windows applications and non-Windows applications running under Windows 3.1.

For information about the parameters discussed in this chapter for HIMEM.SYS, EMM386.EXE, SMARTDRV.EXE, and RAMDRIVE.SYS, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

Tip To see a list of switches that can be used with Microsoft system drivers that have .EXE filename extensions, at the command prompt, type the driver name and the **/?** switch, then press ENTER. For example, type **emm386 /?** at the command prompt to see a list of the switches you can use with EMM386.EXE.

Configuring Hardware for Optimal Performance

- Install additional extended memory.
- Install the largest hard disk that you can afford, as discussed in "Trading Speed for Hard Disk Space" later in this chapter.
- Set the optimal hard disk interleave for your system, using third-party software such as Steve Gibson's SpinRite. Hard disks are often formatted with the wrong interleave at the dealer or factory. Some utilities can correct your interleave without reformatting the hard disk.
- If your system has a memory expansion board and the board can be configured as either expanded or extended memory, configure all of this memory as extended memory, then use EMM386.EXE to emulate expanded memory only as needed by non-Windows applications that require such memory to run. In such a case, place the expanded memory driver first in CONFIG.SYS, followed by the lines that load HIMEM and EMM386.

For example:

```
device=c:\emmbord.sys
device=c:\dos\himem.sys
device=c:\dos\emm386.exe 1024 x=C000-C7FF
```

For information about configuring the memory in your add-on board, see its documentation. Some expanded memory boards may be incompatible with EMM386.EXE.

Configuring MS-DOS for Optimal Performance

- Upgrade to MS-DOS version 5.0 and load MS-DOS into the high memory area (HMA) by setting **dos=high** or **dos=high,umb** in CONFIG.SYS. For more information, see "Loading MS-DOS in the HMA" later in this chapter.
- Make sure the most recent versions of HIMEM.SYS, EMM386.EXE, RAMDRIVE.SYS, and SMARTDRV.EXE are stored in the directories specified in the command lines that load these drivers in your CONFIG.SYS and AUTOEXEC.BAT files. (Placing any of these drivers in the root directory will not improve performance.)
- Load HIMEM.SYS in CONFIG.SYS and make sure the line that loads HIMEM comes before commands that load any other applications or drivers that use extended memory.
- Load SMARTDRV.EXE in AUTOEXEC.BAT and allocate the largest amount of memory possible, because the SMARTDrive disk caching driver can produce the biggest Windows 3.1 performance improvement.
- Set **files=30** in CONFIG.SYS unless you have a software application that requires a higher setting.
- Set **buffers=10** in CONFIG.SYS if you use SMARTDRV.EXE. Using a higher number of buffers with SMARTDrive will decrease efficiency.

  If you are not using SMARTDrive, set **buffers=20** in CONFIG.SYS. More buffers may improve disk access times, but will use more conventional memory.
- Install RAMDrive to use as the TEMP environment variable if you have a diskless workstation or if you have limited hard disk space and ample extended memory (more than 6 MB). Performance improves for Print Manager and for applications that frequently create .TMP files, because access is faster for a RAM disk than for a physical disk.
- Load EMM386.EXE if you are running non-Windows applications that require expanded memory or if you want to allocate UMBs for loading memory-resident TSRs and drivers. For more information, see "Optimizing Use of the UMBs" later in this chapter.

- Load only the necessary TSRs, drivers, or software programs. Wherever possible, load TSRs and device drivers into the UMBs.
- If the environment space is set by the **shell** command in CONFIG.SYS, you can specify a smaller environment.
- Remove any commands for mouse drivers in AUTOEXEC.BAT and CONFIG.SYS if you only use the mouse in Windows, and don't want mouse support while running non-Windows applications in 386 enhanced mode.

Maintaining Optimal Hard Disk Performance

- Delete unnecessary application and system files, including .BAK and .TMP files. Make sure that Windows is not running when you delete .TMP files. For a description of the minimum Windows files, see Appendix C, "The Windows 3.1 Disks and Files."
- Run **chkdsk** frequently to find lost clusters. If lost chains or clusters are detected, then run **chkdsk /f** and delete *.CHK files.
- Use a utility program to compact your hard disk regularly. A fragmented hard disk greatly impacts Windows performance, especially if SMARTDrive is installed or if you are using a temporary swap file. Make sure Windows is not running whenever you run the disk compacting utility.

Configuring Windows for Optimal Performance

- Use a color or a pattern for the desktop background instead of wallpaper if you need to free memory for running applications.
- For standard mode Windows, set the application swap file to the fastest hard drive by setting the **SwapDisk=** entry in the **[NonWindowsApp]** section of SYSTEM.INI.
- For 386 enhanced mode, create a permanent swap file on your fastest hard drive by choosing the 386 Enhanced icon in Control panel and filling in the options in the Virtual Memory dialog box.

  The only reason not to use a permanent swap file is if hard disk space is at a premium. For details about permanent swap files, see "Virtual Memory and 386 Enhanced Mode" in Chapter 5, "Windows 3.1 and Memory Management."
- If you do not create a permanent swap file for 386 enhanced mode, set your temporary swap file to the fastest hard drive by setting a value for **PagingDrive=** or **PagingFile=** in the **[386enh]** section of SYSTEM.INI.
- Run Setup and choose the lowest-resolution display driver that will meet your needs. For VGA-compatible displays that can use an extended mode driver, such as Video 7 or 8514, using the standard VGA driver will ensure faster display performance (but lower resolution and less color support).

Optimizing for Non-Windows Applications

- Allocate the expanded and extended memory required by non-Windows applications by using the PIF Editor to create a custom PIF, as described in Chapter 7, "Setting Up Non-Windows Applications."
- Turn off all entries in the Monitor Ports option in the PIF Editor Advanced dialog box. These options are not required for most non-Windows applications.
- Set **FileSysChange=no** in the **[386enh]** section of SYSTEM.INI to avoid sending update messages to File Manager from non-Windows applications.
- Set **NoEMMDriver=yes** in the **[386enh]** section of SYSTEM.INI if your non-Windows applications do not need expanded memory. This ensures that the most memory possible is available in virtual machines.

Optimizing for TSRs

- Include only the TSRs in AUTOEXEC.BAT that are required for the network, or that must be present for other purposes.
- If a TSR is required only with a single non-Windows application, start the TSR in the virtual machine after launching that application. For more information about creating batch files for running TSRs from Windows, see Chapter 7, "Non-Windows Applications," in the *Windows User's Guide.*
- For all TSRs that you want to be available in Windows, add their .EXE names to the file WINSTART.BAT, which is executed whenever Windows runs in 386 enhanced mode. These TSRs are not available in the virtual machines (VMs), leaving more room in conventional memory for the non-Windows applications. For more information about WINSTART.BAT, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

The following table shows where a TSR is available (and using memory), depending on where it is launched.

Where TSR is launched	*TSR visible in Windows?*	*TSR visible in virtual machines?*
From MS-DOS	Yes	Yes, all virtual machines
From WINSTART.BAT	Yes	No
In a single virtual machine	No	Only that virtual machine

Optimizing for Windows on a Network

- Set **lastdrive** in CONFIG.SYS to **J** or **K** if you can forgo access to some of the network disk drives.
- For a diskless workstation, use a RAM drive as the TEMP variable.
- Do not use a network drive for a temporary swap file, and avoid using a net drive for a permanent swap file if at all possible.
- If a user has problems running Windows or other applications with network software loaded in the upper memory area or in the high memory area, try disabling the network's HMA option. Consult your network documentation for more information.
- Some memory managers, such as EMM386, 386MAX, and QEMM, can load networks and other device drivers in the upper memory area. If the network driver is loaded in the upper memory area and a user has problems running Windows, try loading the network in conventional memory.

Trading Speed for Hard Disk Space

To optimize Windows performance, you must make trade-offs between speed and hard disk space, depending on your system's hardware capabilities. In either case, make sure that as much free memory as possible is available.

To ensure maximum free memory in your system:

- Remove unnecessary TSRs from AUTOEXEC.BAT and CONFIG.SYS.
- Load MS-DOS high and use all possible elements of MS-DOS 5.0 memory management, such as using EMM386.EXE to manage UMBs.

To ensure optimal speed performance:

- Free as much conventional and extended memory as possible.
- Install SMARTDrive for disk caching.
- Set the TEMP variable to the fastest hard disk or to a RAM disk.
- Create a permanent swap file to use in 386 enhanced mode.
- Set **SwapDisk=** in the **[NonWindowsApp]** section of SYSTEM.INI to the hard disk with the most free space.
- Use third-party software to organize the location of files on your hard disk so that the Windows files are located near the file allocation table (FAT).

To ensure maximum hard disk space:

- Create a RAM drive to use as the TEMP variable.
- Delete unnecessary files after exiting Windows.
- Decrease the size of your permanent swap file, or use a temporary swap file in 386 enhanced mode. Or prevent swapping by typing **none** in the New Settings Type edit box of the Virtual Memory dialog box, or by setting **paging=no** in the **[386enh]** section of SYSTEM.INI.
- Follow the steps in Appendix C, "The Windows 3.1 Disks and Files," to create a minimum Windows "footprint."

Optimizing Use of the UMBs

Related information

- *Microsoft MS-DOS User's Guide and Reference:* "Running Programs in the Upper Memory Area" in Chapter 12
- *Windows Resource Kit:* "Expanded Memory: A Technical Discussion" in Chapter 5, "Windows 3.1 and Memory Management"; "Troubleshooting EMS Memory Problems" in Chapter 13, "Troubleshooting Windows 3.1"

On 80386 and higher PCs that use MS-DOS 5.0, you can free conventional memory by loading memory-resident programs such as device drivers, TSRs, and network software into the upper memory blocks (UMBs). You can optimize the way your system uses the UMBs by following the guidelines in this section:

- Load MS-DOS 5.0 in the HMA with the **umb** switch.
- Load EMM386.EXE with the **noems** or the **ram** switch.
- Find the best order for loading drivers and programs into UMBs.
- Adjust the command line for EMM386.EXE in CONFIG.SYS to set aside fewer UMBs for expanded memory, so that there is more room for loading other programs into the UMBs. For details about how to do this, see "Adjusting EMM386 for Optimal Performance" later in this chapter.

Loading MS-DOS in the HMA

You must load MS-DOS 5.0 in the HMA and use the **umb** switch to link the UMBs to conventional memory if you want to load programs into the UMBs.

To load MS-DOS 5.0 in the HMA:

- Add these lines to CONFIG.SYS, in this order:

```
device=<path>\himem.sys
device=<path>\emm386.exe ram | noems
dos=high,umb
```

The parts of MS-DOS 5.0 that are loaded into the HMA include most of the kernel and COMMAND.COM, code pages, and the MS-DOS buffers (from the **buffers=** command in CONFIG.SYS). To remain compatible with software created for earlier versions of MS-DOS, about 19K of MS-DOS 5.0 code stays in conventional memory.

You can only load one program at a time into the HMA, so if you load MS-DOS 5.0 into the HMA, then you cannot load a network redirector, for example, into it as well. If you must choose what to load into the HMA, always choose to load MS-DOS 5.0, because it uses the HMA efficiently and leaves little behind in conventional memory.

Note Use the **device** command to load HIMEM.SYS and EMM386.EXE, since loading them with **devicehigh** has no effect.

Loading EMM386.EXE to Use the UMBs

After loading HIMEM.SYS in CONFIG.SYS, load EMM386.EXE with the **ram** switch or the **noems** switch, plus any other parameters as required for your system. The **ram** switch provides both UMBs and a 64K EMS page frame; **noems** provides UMBs only and no page frame. If none of your applications require EMS, then use **noems**, which will provide an extra 64K in the UMBs.

If you load EMM386 with the **noems** switch and find that some programs fail to run or display out-of-memory messages, it's because they require expanded memory to run. If this happens, replace the **noems** switch with the **ram** switch to make expanded memory available again.

Setting the Load Order for Drivers in UMBs

You must experiment to find the best order for loading drivers and TSRs in the UMBs.

To find the best order for loading drivers into UMBs:

---

1. In CONFIG.SYS, load the largest device drivers with the **devicehigh** command. Use the **device** command to start any drivers that are smaller than the largest memory-resident program loaded from AUTOEXEC.BAT.
2. In AUTOEXEC.BAT, start the largest memory-resident programs with the **loadhigh** command.
3. After rebooting, use the MS-DOS **mem /c** command to see what is loaded into the UMBs and what is in conventional memory. If everything didn't fit into the UMBs, change the order in which the programs are loaded in CONFIG.SYS and AUTOEXEC.BAT.

With **devicehigh** and **loadhigh**, each device and TSR is loaded into the largest available slot, so the order in which programs are loaded can affect how many programs can fit into the UMB. Also, how and when a program allocates memory determines what size UMB it needs. Some programs need a UMB as large as the program size while running, and some need a UMB as large as the program file size.

The following table lists some Microsoft device drivers and memory-resident programs that can run in the UMBs. This table shows whether the program's file size or size while running determines the UMB size the program needs.

Program	*UMB size needed*	*Program*	*UMB size needed*
ANSI.SYS	File size	MOUSE.COM	File size
DISPLAY.SYS	Size while running	MOUSE.SYS	File size
DOSKEY.COM	Size while running	PRINTER.SYS	Size while running
DRIVER.SYS	File size	RAMDRIVE.SYS	Size while running
EGA.SYS	File size	SETVER.EXE	File size
FASTOPEN.EXE	Size while running	SHARE.EXE	Size while running
MIRROR.COM	Size while running		

Note You cannot use **loadhigh** or **devicehigh** to load HIMEM.SYS, EMM386.EXE, or SMARTDRV.EXE.

If your mouse comes with both MOUSE.SYS and MOUSE.COM files, you can load the mouse driver at any point after HIMEM.SYS and EMM386.EXE. Use **devicehigh** in CONFIG.SYS to load MOUSE.SYS, or use **loadhigh** in AUTOEXEC.BAT to load MOUSE.COM.

For example, suppose you have an 80386 computer with 2 MB of extended memory that you want to configure as follows:

- 512K of expanded memory for a non-Windows CAD application
- As much free conventional memory as possible

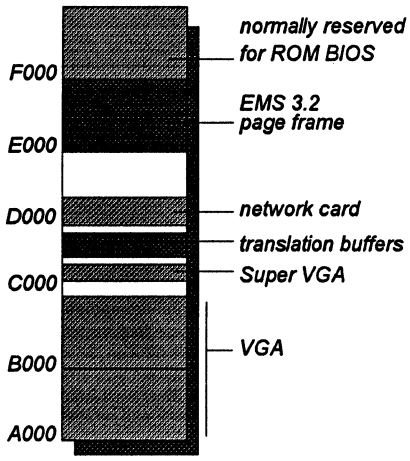
Your CONFIG.SYS file should contain the following commands:

```
device=c:\dos\himem.sys
device=c:\dos\emm386.exe 512 ram
dos=high,umb
devicehigh=c:\dos\mouse.sys
```

Figure 6.1

Example of UMBs in the upper memory area

For a VGA or V7VGA display adapter, a network card at D000-D7FF, and an EMS 3.2 page frame at E000-EFFF



For more information about determining memory requirements and load order for programs, see the *Microsoft MS-DOS Getting Started* booklet and see page 519 of the *Microsoft MS-DOS User's Guide and Reference.*

Adjusting EMM386 for Optimal Performance

The EMM386.EXE driver takes a conservative approach when scanning the UMBs for free space, because it must remain compatible with software that was created for earlier versions of MS-DOS. For example, EMM386.EXE does not map over any unused ROM. If you have space below C800 or above DFFF that you know is unused and you want it used for the UMBs, you must explicitly instruct EMM386.EXE to include it by using the **i=** switch to specify a range of segment addresses.

For more information about EMM386, see pages 287–290 and 605–609 of the *Microsoft MS-DOS User's Guide and Reference;* see also Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

Adjusting EMM386 to Provide More UMBs

You can adjust EMM386 to allocate space in the UMBs, especially if you start EMM386 with the **ram** switch to make expanded memory available to programs. First, carry out Procedure 1 below; then, if you still cannot load programs and device drivers into UMBs, carry out Procedure 2.

Procedure 1
To allocate additional space for use as UMBs:

Flowchart 1.5
Clean Boot Disk

1. Create a backup boot disk, as described in "Creating a 'Clean Boot' for Troubleshooting" in Chapter 13, "Troubleshooting Windows 3.1."
2. Add the **i** switch in front of the **ram** or the **noems** switch in the **device=emm386.exe** line of CONFIG.SYS.

   The **i** switch specifies a range of addresses to be allocated in the UMBs. The value you specify for the **i** switch depends on your computer and monitor type.

   If your computer is not an IBM PS/2 and does not have a monochrome monitor, add the **i=c000-efff** and the **i=b000-b7ff** switches before the **ram** or **noems** switch. For example:

   ```
   device=c:\dos\emm386.exe i=e000-efff i=b000-b7ff ram
   ```

   If your computer is not a PS/2 and has a monochrome monitor, add the **i=e000-efff** switch before the **ram** or **noems** switch. For example:

   ```
   device=c:\dos\emm386.exe i=e000-efff ram
   ```

   If your computer is an IBM PS/2 and does not have a monochrome monitor, add the **i=b000-b7ff** switch before the **ram** or **noems** switch. For example:

   ```
   device=c:\dos\emm386.exe i=b000-b7ff ram
   ```

   If your computer is an IBM PS/2 with a monochrome monitor, see Procedure 2.
3. If your computer fails when you restart it after this change, the memory range you specified for EMM386 is probably being used by hardware or video display drivers. In this case, use your backup boot disk to restart your computer. Then remove the **i** switches you added for EMM386 in CONFIG.SYS.

   If you start EMM386 with the **ram** switch, follow Procedure 2.
4. After your computer starts, use the MS-DOS **mem /c** command to check whether the programs were loaded into UMBs successfully.

If you still cannot load programs into UMBs, your system might be using much of the UMBs for hardware code. Some computers use the UMBs for ROM shadowing, which can improve your computer's speed. (In ROM shadowing, the computer copies hardware code from the slower read-only memory into faster RAM.) To find out more about how your computer uses the UMBs, see the documentation that came with your system.

If you cannot load programs into UMBs, and you are starting EMM386 with the **ram** switch, carry out Procedure 2.

Note Use Procedure 2 only if your non-Windows EMS programs use expanded memory according to LIM EMS version 4.0. Programs that follow LIM EMS 3.2 need the full 64K page frame and are unable to use expanded memory if you follow this procedure.

Procedure 2
To set aside fewer UMBs for expanded memory:

1. Type **emm386** at the command prompt.

   The information that appears shows current memory-management activities. Find the line that reads *Page frame segment,* and write down the hexadecimal address that appears on that line. For example:

   ```
   Page frame segment . . . E000 H
   ```

2. Add the **p0** switch in front of the **ram** switch in the **device=emm386.exe** line of CONFIG.SYS.

   The **p0** switch should specify the address you found in Step 1. For example, if the page-frame segment was E000h, the command would be:

   ```
   device=c:\dos\emm386.exe p0=e000 ram
   ```

3. If your computer fails when you restart it, insert your backup boot disk in drive A and restart your computer. Then remove the **p0** switch you added to CONFIG.SYS in Step 2. Restart your computer and repeat Steps 1 and 2, checking the page-frame segment address carefully.

4. After your computer starts, use the MS-DOS **mem /c** command to check whether the programs were loaded into UMBs successfully.

If your programs still do not run in UMBs, contact Microsoft Product Support Services for assistance.

Freeing Conventional Memory in 386 Enhanced Mode

When you run non-Windows applications in 386 enhanced mode, Windows allocates translation buffers to pass data between the applications and drivers executing in real mode, as described in Chapter 5, "Windows 3.1 and Memory Management." On a stand-alone PC, Windows allocates two translation buffers for each virtual machine (8K total); on a networked PC, Windows allocates six buffers (24K total).

If it finds space, Windows places the translation buffers in the UMBs; otherwise, they are placed in conventional memory, where they take up space in every virtual machine.

To find out if you still have UMBs free while running EMM386, at the command prompt type **mem /p** to see what is free in the hexadecimal addresses between A000 and FFFF. Then add an **x=** switch to the EMM386 line in CONFIG.SYS to exclude the unused UMB ranges.

If you don't have at least 28K of free UMB, you can still partially optimize the use of this area. Because each translation buffer is 4K, each 4K chunk you exclude will save conventional memory. For example, this line will let Windows load one buffer into the UMB, freeing 4K of conventional memory:

```
device=c:\dos\emm386.exe noems x=df00-dfff
```

Excluding memory, however, can change how MS-DOS 5.0 loads programs into UMBs, so you may need to experiment to get the optimal configuration.

Tips for Using Third-Party Memory Managers

For complete compatibility, Windows 3.1 and MS-DOS 5.0 both provide HIMEM.SYS and EMM386.EXE as memory management drivers. The overriding objective Microsoft has for these drivers is to avoid hanging any computer during or after memory configuration. Optimization, while important, is secondary.

You can also use a third-party memory manager to manage UMBs in an 80386 computer. Such memory managers have different objectives, and there are pros and cons to using them in place of the built-in Microsoft memory managers.

To allocate UMBs through a third-party memory manager, you need both a UMB provider and a UMB loader:

- A UMB provider is the memory manager that makes UMBs available. For MS-DOS 5.0, the UMB provider is the combination of EMM386.EXE plus HIMEM.SYS. In QEMM386 and 386MAX, for example, the UMB providers are QEMM386.SYS and 386MAX.SYS, respectively.
- A UMB loader loads a program into the proper UMB. The UMB loader in MS-DOS 5.0 is in the MS-DOS kernel, accessed through the **devicehigh** and **loadhigh** commands. The equivalents in 386MAX are 386LOAD.SYS and 386LOAD.COM; in QEMM386, they are LOADHI.SYS and LOADHI.COM.

The arguments for using third-party memory managers include:

- Most third-party memory managers have a wider range of options for loading into the UMBs. For example, you can specify a particular UMB for loading a program.
- Third-party memory managers typically require 3K to 5K of conventional memory, while HIMEM.SYS with EMM386.EXE may take up to 10K. However, some memory managers also require overhead in the UMBs, but neither HIMEM.SYS nor EMM386.EXE require any.

The arguments against using third-party memory managers include:

- Third-party memory managers can be more difficult to configure properly than EMM386.EXE, and they are often geared toward more sophisticated users.
- Third-party memory managers are supplemental to HIMEM.SYS and EMM386.EXE, so they require additional costs and resources to support. Unless you absolutely require 100 percent optimization, a third-party memory manager may be superfluous.

If you decide to use a third-party memory manager, follow these guidelines:

- Don't use either HIMEM.SYS or EMM386.EXE.
- Use **dos=high** to load MS-DOS 5.0 into the HMA.
- Don't use the **dos=umb**, **devicehigh**, or **loadhigh** commands.
- Don't use the third-party commands that relocate the MS-DOS buffers into the UMBs. MS-DOS 5.0 automatically puts its buffers into the HMA, saving the UMBs for other programs.
- Use both the third-party UMB provider (such as 386MAX.SYS) and the third-party UMB loader (such as 386LOAD.SYS and 386LOAD.COM).
- Use the third-party commands to relocate MS-DOS file handles and drive space (from the **files** and **lastdrive** commands in your CONFIG.SYS) into the UMBs.

You can also use third-party memory drivers as alternatives to HIMEM.SYS, SMARTDRV.EXE, and RAMDRIVE.SYS. However, the memory drivers provided with Windows 3.1 have been specifically designed and tested to provide maximum compatibility between MS-DOS 5.0 and Windows 3.1.

Optimizing with SYSTEM.INI

Many entries in SYSTEM.INI can be set to optimize performance for your system. Some entries are for specific hardware configurations. For details about any of these entries, see Chapter 4, "The Windows Initialization Files."

The **[standard]** section of SYSTEM.INI contains these entries for configuring Windows performance in standard mode: **FasterModeSwitch=** (for specific Zenith and Olivetti systems), **Int28Filter=**, and **Stacks=**. Also, the **SwapDisk=** entry in the **[NonWindowsApp]** section of SYSTEM.INI affects performance for running non-Windows applications in standard mode.

The **[386enh]** section contains entries for optimizing performance in 386 enhanced mode. These entries specify values for virtual memory and paging:

LRU*entries*=	**MinUnlockMem=**	**PagingFile=**
MaxBPs=	**MinUserDiskSpace=**	**PerVMFILES=**
MaxPagingFileSize=	**PageOverCommit=**	**PSPIncrement=**
MaxPhysPage=	**Paging=**	**SysVMV86Locked=**
MinTimeSlice=	**PagingDrive=**	**XlatBufferSize=**

These entries in **[386enh]** specify values for controlling UMBs and EMS management for VMs:

AllEMSLocked=	**IgnoreInstalledEMM=**	**ReserveVideoROM=**
EMMExclude=	**LocalLoadHigh=**	**SysVMEMSLocked=**
EMMInclude=	**NoEMMDriver=**	**SysVMEMSRequired=**
EMMPageFrame=	**ReservedHighArea=**	**UseableHighArea=**
EMMSize=	**ReservePageFrame=**	**XMSUMBInitCalls=**

These entries in **[386enh]** specify values for XMS management and VMs:

AllXMSLocked=	**SysVMEMSLimit=**	**SysVMXMSRequired=**
MapPhysAddress=	**SysVMXMSLimit=**	

These entries in **[386enh]** specify values for DMA buffers:

DMABufferIn1MB=	**EISADMA=**	**MaxDMAPGAddress=**
DMABufferSize=	**HardDiskDMABuffer=**	**MCADMA=**

Several entries in various sections of SYSTEM.INI are specifically related to optimizing Windows performance on a network.

Section	*Network-related entries*	
[Boot]	**CachedFileHandles=**	**Network.drv=**
[Standard]	**Int28Filter=**	**NetHeapSize=**
[NonWindowsApps]	**NetAsynchSwitching=**	
[386enh]	**FileSysChange=**	**Network=**
	InDOSPolling=	**PSPIncrement=**
	Int28Critical=	**ReflectDOSInt2A=**
	NetAsynchFallback=	**TimerCriticalSection=**
	NetAsynchTimeout=	**TokenRingSearch=**
	NetDMASize=	**UniqueDOSPSP=**
	NetHeapSize=	

Tips for Configuring the Windows Shell

This section describes how you can customize Program Manager and File Manager, the principal shells for managing Windows services and files. For information about the system registration database that supports many of these services, see "Managing the Registration Database" in Chapter 11, "Compound Documents in Windows 3.1."

Customizing Program Manager



The Program Manager is installed automatically by Setup as the main Windows shell. Use the menu commands in Program Manager to control the appearance and behavior of items in the window. Use the applications and accessories in the Program Manager groups to control the configuration and appearance of all elements of Windows, including the drivers, memory management, and peripherals such as printers and multimedia devices.

You might choose to use another shell (by changing the **shell=** entry in the **[boot]** section of SYSTEM.INI). However, Setup for most Windows applications looks for Program Manager and cannot complete installation unless Program Manager is the current Windows shell. This is because Program Manager facilitates many services, including the addition of new entries in the system registration database.

You can also specify an entry for **TaskMan.exe=** in the **[boot]** section of SYSTEM.INI if you want to use a third-party task manager.

For details about the Program Manager initialization file, see the entries for PROGMAN.INI in Chapter 4, "The Windows Initialization Files." For details

about working with any items in Program Manager, see Chapter 3, "Program Manager," in the *Windows User's Guide.*

New in Program Manager in Windows 3.1



In Windows 3.1, Program Manager makes fewer demands on system memory resources than in earlier Windows versions. This means that you can have more groups defined without using up memory, and that the window can refresh itself faster. Other changes in Program Manager in Windows 3.1 include:

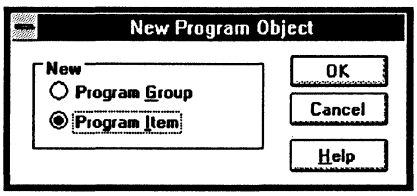
- Context-sensitive help is available for all items.
- Menu settings and window positions are saved automatically, unless you choose otherwise.
- When you use File Run to launch an application, Windows searches for executable programs using every listing in the **programs=** entry of WIN.INI. If you type a filename in the Command Line box, Windows uses the registration database and the **[extensions]** setting in WIN.INI to find the associated program.

Defining Program Manager Groups

The Main, Applications, Accessories, Games, and Startup groups are created in Program Manager automatically during Setup. You can move icons from one group to another by dragging, and you can add to any group by dragging the icon for a file or application from File Manager to the Program Manager group. To create a new group in Program Manager, choose New from the File menu and complete the dialog box. Choose Properties from the File menu to define the appearance and behavior of any group item.

Figure 6.2

New Program Object dialog box





Drag icons into the Startup group for any application or file that you want to run each time you run Windows. (This means you no longer have to edit the **run=** and **load=** entries in WIN.INI to define the startup applications; these entries are independent from the Startup group.)

Information about the groups defined in Program Manager appears in the **[groups]** section of the PROGMAN.INI file.

Tip If you want to rename the Startup group to customize your desktop, add a **startup=***other group name* entry to the **[settings]** section of PROGMAN.INI.

Defining Properties for Program Groups

The Properties command on the Program Manager File menu provides options for controlling the appearance of icons and the behaviors related to launching applications from Program Manager. Make a group window active, or select a group item, then choose Properties to define:

- The icon caption.
- The program and (if you want) the document filename to be launched when you double-click the icon.
- The working directory for the application. By default, this is the directory that contains the application.
- The shortcut key you want to reserve for switching to the application when it's running.
- Whether you want the application to appear as an icon or in a window whenever you run it.
- The icon you want to represent the program item.

Any changes you make take effect immediately. The **[group]** section in PROGMAN.INI and the related .GRP file are updated to save the changes.

For information about customizing the group items, and controlling other aspects of how items appear on the desktop, see Chapter 5, "Control Panel," in the *Windows User's Guide.*

Setting Up Shared Program Manager Groups

Program Manager groups can be shared over the network by making them available to users as read-only files. A user cannot change the contents of a shared group. The Move and Delete commands on the File menu in Program Manager are not available for shared groups. A user can choose the Properties command to review file properties of items in a shared group, but cannot change them.

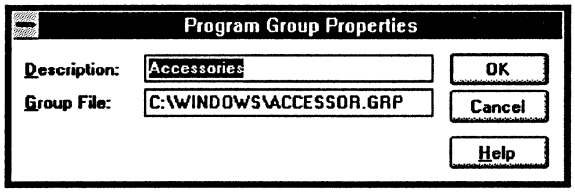
To share a Program Manager group, you must mark the group file (*.GRP) as read-only and place it in a shared network directory. At the workstation, you or a user can create a group and use the group file from the shared network directory.

To set up a shared Program Manager group:

1. Create a group file by using Program Manager to create a group and add program items to it.
2. Copy the group file (.GRP) to a shared network directory and mark it as read-only.
3. From each workstation, use Program Manager to set up the group. In the Group File box in the Program Group Properties dialog box, specify the location and name of the group file in the shared network directory.

Figure 6.3

Program Group Properties dialog box



If you don't want to use Program Manager to set up the group, you can modify the PROGMAN.INI file for the workstation. In the **[groups]** section of the PROGMAN.INI file, add a unique variable plus the name and path of the group file. For example:

```
group8=z:\winshare\programs.grp
```

If you want a shared Program Manager group to be set up when users install Windows for their workstations, you must create a PROGMAN.INI file that points to the shared group, and then have that PROGMAN.INI file copied to the user's WINDOWS directory during Setup.

For more information, see "Modifying .INF Files for Custom Installations" in Chapter 2, "The Windows Setup Information Files."

Configuring Program Manager for a Network Workgroup



If you are the system administrator for a network, you might want to restrict user actions for a workgroup that uses Windows on a network. You can add a **[restrictions]** sections to PROGMAN.INI to:

- Remove the Run command from Program Manager, by setting **NoRun=1**.
- Remove the File menu from Program Manager, by setting **NoFileMenu=1**.
- Not allow workgroup users to exit Windows, by setting **NoClose=1**.
- Not allow workgroup users to save settings, by setting **NoSaveSettings=1**.
- Not allow workgroup users to create, delete, or rename groups or group-items, or change properties for items or groups, by setting values for **EditLevel=**.

For example, if you want users to have a standard work environment where they cannot run programs from Windows other than ones you have set up in Program Manager groups, and you do not want any of their window or icon positions saved when Windows is restarted, specify the following entries in PROGMAN.INI:

```
[restrictions]
NoRun=1
NoSaveSettings=1
EditLevel=4
```

If you do not want users to be able to exit Windows, specify the following entry in PROGMAN.INI:

```
[restrictions]
NoClose=1
```

If you don't want users to run any programs from Windows other than ones that you set up in Program Manager groups, specify the following entries in PROGMAN.INI:

```
[restrictions]
NoRun=1
EditLevel=4
```

Note Some applications allow users to start other applications, for example File Manager. So if you do not want users to start other applications, make sure you do not include them in Program Manager groups.

For more information, see the description of the **[restrictions]** section in "PROGMAN.INI: The Program Manager Initialization File" in Chapter 4, "The Windows Initialization Files."

Customizing File Manager



The Windows 3.1 File Manager provides the tools you need to organize files and directories and to take advantage of drag-and-drop features for creating compound documents.

For details about working with any items in File Manager, see Chapter 4, "File Manager," in the *Windows User's Guide.*

New in File Manager for Windows 3.1



The File Manager works more quickly and efficiently in Windows 3.1. For example:

- The window updating, drive searching, and background processing capabilities have been enhanced. Also, you can cancel time-consuming processes (such as complete searches of a CD-ROM drive).
- You can view more than one directory tree at a time, and browse the file system in a single window. Also, more information appears, such as disk space usage, network share names, and more.
- Context-sensitive help is available for all items.
- Several inconsistent behaviors have been changed, so that File Manager provides the capabilities you expect and need for file management. For example, menu settings and window positions are saved automatically, unless you choose otherwise.
- You can open and print files by dragging icons in File Manager, and drag icons to embed and link one file in another, as described in Chapter 13, "Integrating Your Windows Applications," in the *Windows User's Guide.*
- For networks, the previous network connections are made automatically when you run Windows, and related menu items appear automatically to assist you with network activities related to file management.
- You can use long file names on network servers, if the driver for your network supports this feature.

Defining File Manager Actions

To use File Manager services or to define the behavior and appearance of information in the window:

- Choose File and Disk menu commands for file, disk, and drive management activities.
- Choose Tree and View menu commands to arrange the appearance of directories and files in the File Manager windows.
- Choose Confirmation from the Options menu to specify whether you want warning messages to appear when you delete files or perform other actions.



- Choose other Options menu commands to define the font, status bar, and other behaviors for running and quitting File Manager.

Associating Extensions with File Types

If the filename extension for a file is associated with a file type defined in the registration database, you can:

- Open the file by double-clicking its icon in File Manager.



- Print the file by dragging its icon to the Print Manager.
- Place the file as a group item in any Program Manager group.



- Embed the file as a packaged object by dragging its icon to an application icon or to another document icon.

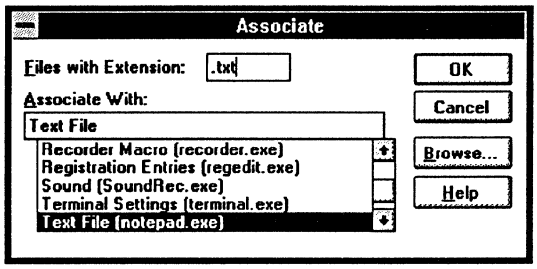
Many filename extensions are predefined when you install Windows. You can use the File Manager to add or change the association of a filename extension with any file type that is defined in the registration database.

To associate a filename extension with a file type:

1. In the File Manager window, select the icon for the file you want to associate with a file type.
2. Choose Associate from the File menu in File Manager.
3. If the extension of the selected file doesn't appear, type the filename extension you want in the Files With Extension box.
4. Choose the application in the Associate With list that you want associated with the extension. If the application name isn't in the list, type an executable program filename in the box, or click Browse and select a file from the list. Then click the OK button.

Figure 6.4

Associate dialog box



When you change the associated filename extension in the File Manager Associate dialog box, Windows updates the related entry in the registration database and updates the list in the **[extensions]** section of WIN.INI.

Note If you want to use drag-and-drop in File Manager to create packages, the server application must have been written for Windows 3.1. Otherwise, you must use the Object Packager to create the package, as described in Chapter 13, "Integrating Your Windows Applications," in the *Windows User's Guide.*

More About Memory Management

Gookin, Dan. *The Microsoft Guide to Managing Memory with DOS 5.* Microsoft Press, 1991.

"Running Programs in the Upper Memory Area" in Chapter 12 of the *Microsoft MS-DOS User's Guide and Reference.*

Non-Windows–based Applications and PIFS



Part
3 Non-Windows–Based Applications

Chapter 7 Setting Up Non-Windows Applications 283

About Applications 284
Setting Up PIFs and Program Items 285
Starting Non-Windows Applications with Windows 288
Running Non-Windows Applications 290
Tips for Running Non-Windows Applications 294

Chapter 8 PIFs and PIF Editor 299

About Creating PIFs 300
About PIF Editor 300
Modifying the Default PIF 305
Creating Multiple PIFs for a Single Application 306

Chapter

7 Setting Up Non-Windows Applications

This chapter presents details about how to set up and run non-Windows applications under Microsoft Windows 3.1. For information about creating program information files, see Chapter 8, "PIFs and PIF Editor." For information about specific problems with running non-Windows applications, see Chapter 13, "Troubleshooting Windows 3.1."

Related information

- *Windows User's Guide:* Chapter 7, "Non-Windows Applications," and Chapter 8, "PIF Editor"; see also the background information on creating program groups and managing files in Chapter 3, "Program Manager," and Chapter 4, "File Manager"
- *Windows Resource Kit:* "APPS.INF" in Chapter 2, "The Windows Setup Information Files"; Part 2, "Configuring Windows 3.1"
- *Glossary terms:* application, expanded memory, extended memory, grabber, PIF, virtual memory

Contents of this chapter

About Applications 284
Setting Up PIFs and Program Items 285
    Setting Up a PIF 285
    Setting Up Icons in Program Manager 287
Starting Non-Windows Applications with Windows 288
    Starting an Application When Starting Windows 288
    Starting an Application from the Desktop 289
Running Non-Windows Applications 290
    Running Applications in Standard Mode 290
    Running Applications in 386 Enhanced Mode 291
    SYSTEM.INI Entries for Non-Windows Applications 293
Tips for Running Non-Windows Applications 294
    General Tips 294
    Rebooting a Non-Windows Application 294
    Notes on Running Non-Windows Applications 295

About Applications

You are likely to encounter two basic types of applications while running Microsoft Windows 3.1: Windows applications and non-Windows applications.

Windows applications are any applications that are designed to run with Windows and that do not run without Windows. All Windows applications follow similar conventions for arrangement of menus, style of dialog boxes, and keyboard and mouse use. You might encounter these types of Windows applications:

- **Applications designed for Windows 3.1.**
- **Applications designed for Windows 3.0 or later versions.** Windows applications can take advantage of Windows 3.0 features and interact with other Windows applications.
- **Applications designed for older Windows versions.** Applications designed for Windows 2.x or earlier cannot run under Windows 3.1. When you attempt to run such an application, Windows displays a message advising you to get an updated version of that application. Your application manufacturers will be able to tell you if a Windows 3.x version of their software is available.

"Non-Windows application" is used in the *Windows Resource Kit* as a shorthand term to refer to an application that is designed to run with MS-DOS but not specifically with Windows. Such applications may not be able to take full advantage of all Windows features (such as memory management). There are two basic kinds of non-Windows applications that you might encounter under Windows 3.1:

- **Application software.** Applications such as Lotus 1-2-3 and WordPerfect can be run under Windows in either standard or 386 enhanced mode. A program information file (PIF) gives Windows the information it needs to run a non-Windows application.
- **TSRs and other memory-resident software.** Memory-resident programs such as mouse drivers, network software, or pop-up programs like Borland Sidekick are loaded into memory when you start them and are available while other applications are running. You can start memory-resident programs before you start Windows or while Windows is running. Such non-Windows programs also use a PIF if you run them under Windows.

The next section of this chapter contains details about setting up non-Windows applications that you want to run under Windows 3.1.

Setting Up PIFs and Program Items

To run a non-Windows application, Windows uses a program information file (PIF) to run the application in the most efficient way. Windows needs to refer to PIF information because non-Windows applications don't use memory cooperatively in the ways that Windows applications do, which can lead to insufficient-memory problems.

A custom PIF contains specific information about running the application under Windows, including how much memory it needs and how it uses system components such as the communications ports. If there isn't a custom PIF for an application, Windows uses _DEFAULT.PIF.

You can set up a PIF for a non-Windows application in three ways:

- Install a PIF with Windows Setup.
- Use the PIF that came with your non-Windows application.
- Use PIF Editor to create a new PIF or to modify _DEFAULT.PIF, the default program information file. For details, see Chapter 8, "PIFs and PIF Editor."

Setting Up a PIF

The Windows Setup program can install PIFs and icons for your non-Windows applications, either automatically during Windows setup or after Windows is installed. This is the best method for specifying a PIF, because Windows 3.1 includes data for creating a large variety of PIFs specially for popular non-Windows applications, as listed at the end of this chapter.

The Setup program gets the information it needs for a PIF by checking a file called APPS.INF, which is placed in the Windows SYSTEM subdirectory during installation. APPS.INF contains the recommended settings for many popular non-Windows applications. For details about the contents of APPS.INF, see Chapter 4, "The Windows Initialization Files."

If the APPS.INF file doesn't contain information about your application, then Windows uses the _DEFAULT.PIF file to run the application.

To set up a PIF during Windows installation:

1. Run Windows Setup from the command prompt.
2. In the Windows Setup dialog box, select the Set Up Applications option.
3. When the Set Up Applications dialog box appears, select the applications from the box on the left that you want to run under Windows. Then choose the Add button.
4. When you have selected all the applications to be set up, choose the OK button.

Windows creates PIFs for the applications you choose, and adds a program-item icon for each to the Applications group in Program Manager.

To set up a PIF after installing Windows:

1. Run Setup by choosing the Windows Setup icon from the Main group in Program Manager.
2. From the Options menu in Setup, choose Set Up Applications.

   If you don't want Setup to search your entire hard disk for non-Windows applications, choose the Ask You To Specify An Application option.
3. Choose the OK button in the Set Up Applications dialog box. Windows Setup searches for applications and displays a list of those it finds.
4. Select applications from the box on the left that you want to add. Then choose the Add button, and choose the OK button.

Windows Setup creates a PIF and adds a program-item icon for each application to the Applications group in Program Manager.

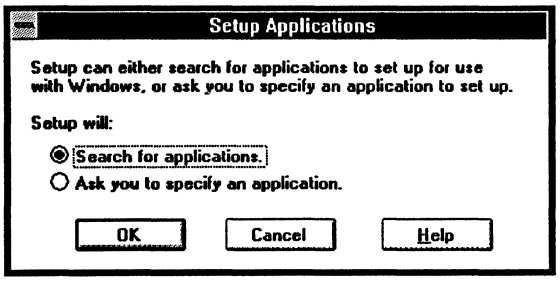


Figure 7.1

Setup Applications dialog box

To display this dialog, choose Set Up Applications from the Options menu in Windows Setup

Some application manufacturers provide PIFs for their applications. To find out if your application has a PIF, contact the manufacturer or search the application disks for a file with a .PIF filename extension. If a PIF came with your application, copy the file to the WINDOWS directory on your hard disk.

If there is a PIF for your application available through Windows Setup and another that came from your application's manufacturer, use the PIF from the manufacturer.

Setting Up Icons in Program Manager

When you use Setup to add PIFs for non-Windows applications, Setup automatically adds program-item icons to the Applications group in Program Manager. If you create or add a PIF for a non-Windows application, you can also add a program-item icon for the PIF to Program Manager. For more information about how to set up a program item, see Chapter 3, "Program Manager," in the *Windows User's Guide.*

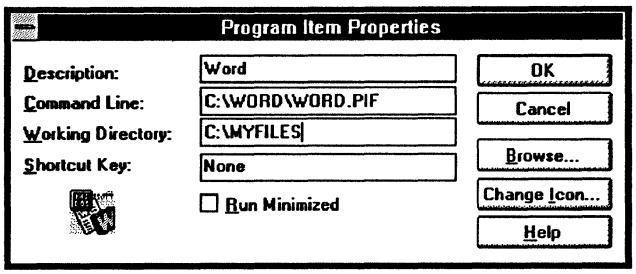
To add a PIF to a Program Manager group:

1. In Program Manager, select the program group where you want to add the icon for a PIF.
2. Choose New from the File menu. Select Program Item in the dialog box, and choose the OK button.
3. Type a description for the application in the Description box.
4. Specify the complete path and filename for the PIF in the Command Line box, including the .PIF extension. Then choose the OK button.

Figure 7.2

Program Item Properties dialog box

To display this dialog, choose File Properties in Program Manager



You can also create a new program item by dragging an application's .PIF or .EXE file from the File Manager window to the Program Manager window. The program item appears with the default MS-DOS Box icon. You must use the Properties command to change this icon.

To alter the program-item icon for a PIF:

1. Select the PIF's icon and choose Properties from the File menu in Program Manager.
2. Choose the Change Icon button in the Properties dialog box.
3. Choose an icon in the Change Icon dialog box. Or choose Browse and select other file choices.

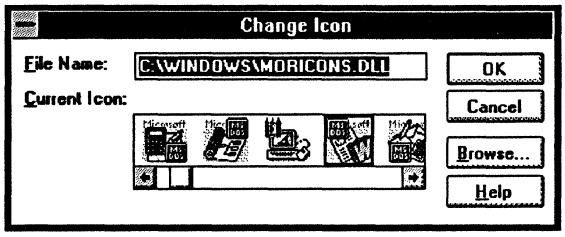
   For example, change the filename to PROGMAN.EXE to display the icons available in PROGMAN.EXE. (You can use icons from any Windows application by typing its executable filename in this field.)

   Windows 3.1 includes a wide range of icons, displayed through the MORICONS.DLL.
4. Scroll the view box and select the icon you want, then choose the OK button.

Figure 7.3

Change Icon dialog box

To display this dialog, click Change Icon in the Program Item Properties dialog box



Starting Non-Windows Applications with Windows

You can start a non-Windows application at the command prompt when you start Windows, or start the application using either Program Manager or File Manager.

Starting an Application When Starting Windows

To start a non-Windows application when you run Windows, do one of the following:

- Add an icon for the non-Windows application to the Startup group in Program Manager. All applications in this group start automatically with Windows.

- Or add the filename of the application or its PIF to the **run=** or **load=** entries in the **[Windows]** section of WIN.INI.
- Or, at the command prompt, type **win** with a command to run the application. For example, to start Windows and run Microsoft Word 5.5, at the command prompt type **win word** and press ENTER.

At the command prompt, you can use the program's .EXE command with any switches, or you can use a .BAT file that changes the environment or runs TSRs before executing the program command.

After Windows starts, it looks for a .PIF file with the same name as the executable program; if none is found, Windows uses _DEFAULT.PIF to run the non-Windows application.

Starting an Application from the Desktop

If you add an icon for a non-Windows application to a Program Manager group, you can start a non-Windows application by choosing its icon, just as you start any application.

The program-item icon can represent an .EXE, .PIF, .COM, or .BAT file that runs the non-Windows application. To see what the icon represents, select it and choose Properties from the File menu in Program Manager.

You can also use File Manager to start a non-Windows application in the same way that you start a Windows application.

Tip If you get an out-of-memory message when you start a non-Windows application, there are a number of things you can do to offset the problem:

- Close or minimize any other applications you are running.
- Clear or save the contents of Clipboard.
- Set the Wallpaper option to None for the desktop.
- If you are running Windows in 386 enhanced mode, try running the non-Windows application in full-screen display instead of in a window, and don't allow any applications to run in Background mode. To do this, edit the application's PIF to make sure that Full Screen is selected and the Execution Background box is not checked.

Running Non-Windows Applications

Windows runs non-Windows applications differently in standard mode and 386 enhanced mode.

Running Applications in Standard Mode

If you are running Windows in standard mode, a non-Windows application takes up the entire screen rather than running in a window. You can load more than one application at a time, but each application will run in full screen, and no application can process tasks in the background. This type of multitasking is called "task switching."

How Standard Mode Works with MS-DOS

When you run Windows in standard mode, it loads most of itself into extended memory. Then it creates an emulated MS-DOS real-mode environment in extended memory, using DOSX.EXE. Because there can only be one DOSX environment at a time in standard mode, only one non-Windows application can run in it at a time.

The other files that support non-Windows applications in standard mode are the .2GR grabber file for your display adapter and WINOLDAPP.MOD (WinOldApp).

Application Swap Files and Standard Mode

In standard mode, each time you start a non-Windows application, Windows creates a temporary application swap file for that application. When you switch away from the application, Windows moves some or all of that application from memory to the application swap file. When data is moved from memory to the swap file, the memory is then available for use by other applications. When you exit Windows, it deletes any application swap files created during that session.

The amount of hard disk space available determines how many application swap files Windows can create, and this in turn determines how many non-Windows applications you can start before running out of memory. For more information about application swap files, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

Running Applications in 386 Enhanced Mode

In 386 enhanced mode, you can run and view more than one non-Windows application at the same time, because you can run most non-Windows applications in a window rather than being restricted to full screen as in standard mode. You can move and size the window, shrink the window to an icon, and share information with other applications by using the Clipboard.

When a non-Windows application is running under Windows in 386 enhanced mode, you can use its Options menu for several tasks:

- Choose Edit to use commands for copying and pasting with the Clipboard.
- Choose Fonts to select a display font for text in the window.
- Choose Settings to specify Multitasking and Display Options for this session.
- Choose other standard Option menu commands to maximize and minimize the window or to switch tasks.

How 386 Enhanced Mode Works with MS-DOS

Each time you run a non-Windows application in 386 enhanced mode, Windows creates a virtual 8086 machine in which the application runs. This virtual machine is an emulation of a complete 8086 processor, including memory, I/O devices, and any TSRs, device drivers, and network software running when you started Windows.

In 386 enhanced mode, Windows remains in memory and active while running non-Windows applications. Each non-Windows application is treated as a multitasking "task" for timeslicing and device contention. Windows uses preemptive scheduling with non-Windows applications, and uses non-preemptive scheduling with Windows applications. Windows and all current Windows applications are treated as a single preemptive task, and you assign the priority for Windows in the 386 Enhanced dialog box in Control Panel. The non-Windows applications are each treated as separate tasks, and you assign their priorities in their PIFs. The percentage of time spent on an application is its assigned priority divided by the sum of all priorities for Windows and all non-Windows applications currently running.

The example in the following table shows how timeslicing is allocated between Windows and non-Windows applications.

Virtual machine	Setting	Processor time
Windows in the foreground, with two non-Windows applications running:		
System virtual machine	100	50%
MS-DOS VM #1	50	25%
MS-DOS VM#2	50	25%
Total value:	200	100%
Windows in the foreground, with four non-Windows applications running:		
System virtual machine	100	32%
MS-DOS VM #1	50	17%
MS-DOS VM#2	50	17%
MS-DOS VM #3	50	17%
MS-DOS VM #4	50	17%
Total value:	300	100%
Non-Windows application in foreground, with Windows and another non-Windows application running in the background:		
System virtual machine	50	25%
MS-DOS VM #1	100	50%
MS-DOS VM #2	50	25%
Total value:	200	100%

The files that support non-Windows applications in 386 enhanced mode are the .3GR grabber file for your display device and WINOA386.MOD (WinOldApp).

For information about virtual memory in 386 enhanced mode, see Chapter 5, "Windows 3.1 and Memory Management." For information about how Windows handles concurrent requests for devices in 386 enhanced mode, see Chapter 5, "Control Panel," in the *Windows User's Guide.*

Running Applications in a Window

You can only run non-Windows applications in a window if you run Windows in 386 enhanced mode. To get non-Windows applications to run in a window in 386 enhanced mode:

- Set the PIF's Display Usage option to Windowed (not Full Screen), and set the Display Options correctly in the Advanced dialog box.
- Or choose Settings from the Options menu while the application is running, and specify Windowed for Display Usage. This will work only for this session.
- Or press ALT+ENTER while the application is running in full screen. This will work only for this session.

However, some non-Windows applications can't run in a window, because they perform screen actions that are incompatible with the Windows display management features. Most of these applications can run successfully in the Windows environment as long as they run in a full-screen display.

SYSTEM.INI Entries for Non-Windows Applications

Several entries in SYSTEM.INI affect how non-Windows applications run. For details, see the descriptions of the specific entries in Chapter 4, "The Windows Initialization Files."

Section	*Entries related to non-Windows applications*	
[Boot]	**286grabber=**	**386grabber=**
[Standard]	**Int28Filter=**	**Stacks=**
[NonWindowsApps]	**CommandEnvSize=** **DisablePositionSave=** **FontChangeEnable=** **MouseInDosBox=**	**NetAsynchSwitching=** **ScreenLines=** **SwapDisk=**

The **[386enh]** section contains many entries that control how memory is allocated for virtual machines, and other memory management issues related to running non-Windows applications. The following entries will also affect performance and behavior for non-Windows applications, sometimes overriding settings in application PIFs.

These entries in **[386enh]** control video memory and system memory: **AllEMSLocked=**, **AllXMSLocked=**, **ReserveVideoROM=**, and **ReservePageFrame=**.

These entries in **[386enh]** control fonts in windows displaying non-Windows applications: ***WOA.FON=** entries, **UseROMFont=**, and **WOAFont=**.

These entries in **[386enh]** control devices: **COM*x*AutoAssign=** and **MouseSoftInit=**.

These entries in **[386enh]** control multitasking: **WinExclusive=** and **WinTimeSlice=**.

These entries in **[386enh]** control screen refreshing and the display of warning messages:

AutoRestoreScreen=	**BkGndNotifyAtPFault=** **FileSysChange=**	**VideoBackgroundMsg=** **WindowUpdateTime=**

Tips for Running Non-Windows Applications

This section contains tips for running non-Windows applications, in response to frequently asked questions directed to Microsoft product support specialists.

General Tips

Here are some general tips for running non-Windows applications and PIFs:

- Make sure you have properly configured the memory in your system, following the recommendations in Chapter 6, "Tips for Configuring Windows 3.1."
- If you can't start a non-Windows application, first check to make sure the application runs under MS-DOS without Windows. If it can't start or run under MS-DOS, it won't run under Windows.
- If the application doesn't access resources directly, use the _DEFAULT.PIF settings for the application's PIF.
- If the application is resource-intensive due to graphics or memory requirements, make the following settings in the application's PIF:

Full Screen:	Checked
Exclusive:	Checked
Display Options/Video Memory:	High Graphics
Retain Video Memory:	Checked

- If you get a not-enough-memory error while running Windows in 386 enhanced mode, try changing the PIF settings for both the KB Required and KB Desired to **-1**. This will allocate all available conventional memory to your application.

Flowchart Series 5 Windows Operation and Memory Problems

- Check Flowcharts 5.3, 5.4, and 5.5 for troubleshooting tips if you experience application execution errors, general protection faults, or other serious problems while running a non-Windows application.

Rebooting a Non-Windows Application



If you can't quit or complete other actions while running a non-Windows application, you can quit by pressing CTRL+ALT+DEL for a local reboot. This has the same effect as choosing the Settings Terminate command from the Control menu and, hence, can be risky.

Do not use a local reboot or the Settings Terminate command as a normal way to quit non-Windows applications. Use local reboot only in one of the circumstances described below.

If you press CTRL+ALT+DEL when an application is not responding to Windows commands to close or when the application is hung, an error message appears with options for closing the application. These are the four cases:

1. For a hung Windows application, you can press the ESC key to return to Windows, press ENTER to close the application, or press CTRL+ALT+DEL to reboot the system.
2. If a non-Windows application is hung and you press CTRL+ALT+DEL, you can then press the ESC key to cancel and return to the application, press ENTER to quit the application, or press CTRL+ALT+DEL to reboot the system.
3. If a non-Windows application is hung while running in the background and a Windows application is running in the foreground, you can press CTRL+ALT+DEL, then press any key to activate the non-Windows application or press CTRL+ALT+DEL to reboot the system.

   If you press a key to activate the hung non-Windows application, the hung application appears in full screen. If you press CTRL+ALT+DEL again to quit this application, then you will see the error message described in case #2 above.
4. If you receive a message that the application is not responding to Windows commands to close (so that you must reboot the system), you can press any key or press CTRL+ALT+DEL again to reboot the system.

If you press CTRL+ALT+DEL when the current application is active (not hung), you will get a message warning you that the application is still running and asking if you want to proceed with rebooting.

Notes on Running Non-Windows Applications

DPMI vs. VCPI for Non-Windows Applications

Windows 3.1 uses the DPMI specification to allow non-Windows applications to access extended memory on 80286 and 80386 machines.

Some non-Windows programs use the Virtual Control Program Interface (VCPI) specification to access extended memory on 80386 machines. The performance of such applications depends on how closely the application follows the VCPI specification. Most programs that use the VCPI specification can run under Windows 3.1.

For more details about DPMI and VCPI, see Chapter 5, "Windows 3.1 and Memory Management."

Running Applications That Require Expanded Memory

If your computer has an 80286 processor, you need an expanded memory board for applications that use expanded memory. Your CONFIG.SYS file must also contain a command line for the expanded memory manager that comes with the board.

If your computer has an 80386 or higher processor, you can use EMM386.EXE to simulate expanded memory for non-Windows applications when you run Windows in 386 enhanced mode. In the application's PIF, you must also specify the amount of expanded memory the application requires.

For information about using expanded memory to run non-Windows applications under Windows 3.1, see Chapter 5, "Windows 3.1 and Memory Management."

Using a Mouse with Non-Windows Applications



To use a mouse with non-Windows applications, you must have an MS-DOS mouse driver that is loaded before you run Windows.

Flowchart 2.8 Mouse Doesn't Work in Non-Windows Applications

If your mouse doesn't work for non-Windows applications, check CONFIG.SYS and AUTOEXEC.BAT to be sure the MS-DOS mouse driver is installed properly. Windows 3.1 has its own mouse driver that must be installed if you want to use the mouse with Windows and with Windows applications.

Applications with PIF Data in APPS.INF

The applications listed in the following tables have entries in the **[pif]** section of APPS.INF. Windows Setup can use the information about these applications in **[pif]** to create custom PIFs for running the applications under Windows 3.1.

Applications with PIF data in APPS.INF

Access for MS-DOS
ACCPAC BPI
ACCPAC Plus

APPLAUSE II 1.5
Ashton Tate dBase IV
AutoCAD

AutoCAD (Batch File)
Autosketch 3.0

Borland C++
BRIEF 3.1
Close-Up 4.0

Crosstalk-XVI 3.71
DataEase

DataPerfect
DECnet Job Spawner
DisplayWrite 3

DisplayWrite 4
DisplayWrite 5
DisplayWrite Assistant

DM - Directory Maintenance
DrawPerfect
DWDOS286

DWDOS386
DWINFO2
DWINFO3

Extra! for MS-DOS
Flight Simulator 3.0
Flight Simulator 4.0

Formtool
Foxbase+
FoxPro (Max. Cfg.)

FTP Utility
Framework III
Freelance Plus 4.0

FTPSRV Utility
Generic CADD
Graphwriter

GW BASIC
Harvard GeoGraphics
Harvard Graphics 2.3

Harvard Graphics 3.0
Harvard Project Manager
Harvard Total Project Manager

HotWire
IBM Filing Assistant
IBM Graphics Assistant

IBM Personal Editor
IBM Professional Editor
IBM Writing Assistant 2.0

Insight
Interleaf 5 for MS-DOS
Kid Pix

KnowledgePro (MS-DOS)
Lap-Link Pro
Learning Microsoft Works

Learning MS-DOS 3.0
Learning MS-DOS Quick Ref
LetterPerfect

Lotus 1-2-3 3.1
Lotus 1-2-3form
Lotus Access System

Lotus Agenda
LotusWorks 1.0
LPQ Utility

LPR Utility
Magellan 2.0
Managing Your Money

Manifest
Microrim R:Base 3.0
Microrim R:Base 5000

Microrim R:Base Clout
Microsoft Adv. Basic
Microsoft Basic

Microsoft Bookshelf
Microsoft C Compiler 5.1/6
Microsoft Chart

Microsoft Fortran Compiler 5.1
Microsoft Game Shop
Microsoft Macro Assembler

Microsoft Mail
Microsoft Mail-Admin
Microsoft Make Utility

Microsoft Multiplan
Microsoft Multiplan
Microsoft Online 1.0

Microsoft Pascal Compiler
Microsoft Project
Microsoft QBasic

Microsoft Quick C
Microsoft Quick Pascal
Microsoft QuickBasic

Microsoft QuickBasic Ext.
Microsoft Spell
Microsoft Word 4.0

Microsoft Word 5.x
Microsoft Works 2.0
MultiMate 4.0

Network Control Program
Norton Utilities 4.5
Norton Utilities 5/6.0

Now!
OPTune
Paradox

Paradox 3.0
Paradox 3.5
PC Paintbrush IV Plus

PC3270
PCMAIL Utility
PC Tools - FileFind

PC Tools - View
PC Tools Desktop 5.51
PC Tools PCShell 5.5

PFS: Access
PFS: First Choice 3.0
PFS: First Choice 3.1

PFS: First Graphics
PFS: First Publisher
PFS: Plan

PFS: Professional Network
PING Utility
Procomm

Procomm Plus 1.1B
Prodigy
Professional File

Professional Write
Programmer's WorkBench
Q & A Report Writer

Q-DOS 3
QModem
Quick Verse 2.0

Quicken
Ready!

Reflection 1
Reflection 2
Reflection 4

Reflection 7
Reflection 8
Reflex 2.0

Relay Gold
Remote 2 call
RightWriter

RLOGINVT Utility
RSH Utility
SAS 604

Scheduler
SEDT Editor
Sethost Terminal Emulator

SideKick 1.0
SideKick 2.0
SideKick Plus

SideKick Plus
Smartcom II
Soft Kicker

SPSS/PC+
SuperCalc 4.0

SuperCalc 5.0
Symphony 2.2
TeleMate

TN Utility
Turbo Pascal 6.0
TurboTax

Ventura Publisher
VMAIL Utility
Volkswriter 3.0

WordPerfect
WordPerfect Office
WordStar 2000

WordStar Professional
WPOffice Calculator
WPOffice Editor

WPOffice File Manager
WPOffice NoteBook
Writer Rabbit

XTree Gold
XyWrite

More applications are listed in the *ambiguous exe* sections of APPS.INF. If you specify the executable filename for any of these applications, Setup will prompt you to specify which of these applications you want to install. Setup can also create custom PIFs for these applications.

Applications with ambiguous filenames listed in APPS.INF

Ashton Tate dBase III	Microrim R:Base System V	Quattro Pro 1.0
Autosketch 2.0	Microsoft Basic Compiler	Quattro Pro 2.0/3.0
Brief 2.1/3.0	Microsoft Macro Assembler	Symphony (Access)
cc:Mail for MS-DOS	Microsoft QuickC w/ QASM	PCTools Desktop 6.0
Crosstalk Mark 4	Microsoft Works 1.x	PCTools Desktop 7.0
Framework II	MS Quick Pascal Express	PCTools PCShell 6.0
Freelance Plus 3.0	PFS: Professional Plan	Turbo Pascal 5.0
Harvard Graphics 2.0	PlanPerfect	WPOffice Calendar
Harvard Graphics 2.1	Procomm Plus	XcelleNet X/Mail-MS-DOS
Learning MS-Word 5.0	Multimate	OfficeWriter 6.2
Learning MS-Word 5.5	Norton Line Printer	Paradox SE
Lotus 1-2-3 2.3	OfficeWriter	PCSA Mail

Chapter

8 PIFs and PIF Editor

This chapter presents details related to creating program information files (PIFs) for non-Windows applications running under Microsoft Windows 3.1.

For information about specific problems related to running non-Windows applications, see Chapter 13, "Troubleshooting Windows 3.1."

Related information

- *Windows User's Guide:* Chapter 8, "PIF Editor"; see also the background information on creating program groups in Chapter 3, "Program Manager"
- *Windows Resource Kit:* Chapter 7, "Setting Up Non-Windows Applications"; Part 2, "Configuring Windows 3.1"
- *Glossary terms:* application, expanded memory, extended memory, LIM 3.2 and 4.0, PIF, virtual memory

Contents of this chapter

About Creating PIFs 300
About PIF Editor 300
The Operating Mode and Basic Parameters 300
PIF Settings for Standard Mode 301
PIF Settings for 386 Enhanced Mode 303
Modifying the Default PIF 305
Creating Multiple PIFs for a Single Application 306

About Creating PIFs

You must use the PIF Editor to create or modify PIFs. This Windows utility presents dialog boxes in which you can specify options for customizing how Windows runs an application in either standard mode or 386 enhanced mode.

The PIF Editor includes complete context-sensitive help for all dialog box options. When you are using PIF Editor, you can find the information you need to make appropriate choices for each option by pressing F1.

This section provides an overview of the options in PIF Editor in relation to memory management and other issues, and describes briefly how to use PIF Editor to modify _DEFAULT.PIF and how to create multiple PIFs for a single application.

For complete descriptions of PIFs and details about using the PIF Editor, see Chapter 8, "PIF Editor," in the *Windows User's Guide.*

About PIF Editor

This section briefly reviews the options in PIF Editor.

The Operating Mode and Basic Parameters


PIF Editor

Choose Standard or 386 Enhanced from the Mode menu in PIF Editor to specify the operating mode for which you're creating the PIF. Otherwise, PIF Editor shows the settings for the mode under which Windows is currently running.

For either Windows operating mode, you might specify:

- The program filename or batch file to be executed, including filename extension.
- The title you want to appear in the window's title bar and under the icon on the desktop. If this field is blank in PIF Editor, the title bar for the application window will contain the PIF filename with no extension.

- Any switches or other command line information. Type **?** in the Optional Parameters box if you want to be prompted for switches each time you run this application.
- The startup directory. If a different working directory is specified for the program-item icon in Program Manager, that setting overrides the Start-Up Directory setting.

For any of these basic parameter fields in PIF Editor, you can specify environment variables defined in AUTOEXEC.BAT by enclosing the variable between **%** signs. For example, **%worddir%** might specify the environment variable for the Microsoft Word 5.x directory as set in AUTOEXEC.BAT. For details about setting and using environment variables, see your MS-DOS documentation.

PIF Settings for Standard Mode

For running an application under Windows in standard mode, you can specify information about these items to control memory use, system resources, and other behaviors:

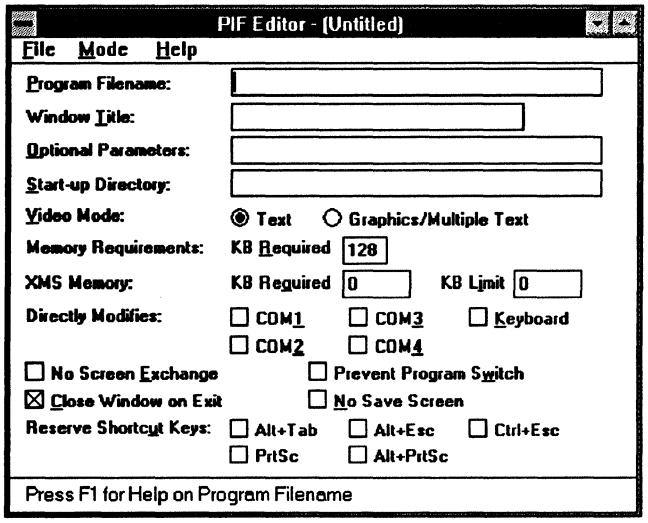
- **Video memory settings** specify memory requirements for the display adapted. Select Text if the application doesn't use graphics, because this setting uses the least memory.
- **System memory settings** specify the amount of conventional memory required to run the application. Under the XMS KB Limit option, specify **0** for no extended memory; specify **-1** for no limit if the application uses a large amount of extended memory.

  There is no option for expanded memory, because under standard mode, expanded memory must be allocated through an EMS board and driver on an 80286 machine or through an expanded memory manager such as EMM386 on an 80386 machine.
- **Settings for resource sharing** (the Directly Modifies options) prevent applications from using the same serial port simultaneously and prevent switching between ports. Check the Keyboard option to specify that the application has exclusive control of the keyboard, so that Windows cannot switch away from the application. This option also makes more memory available to the application, because Windows doesn't need to save the current state or display settings.

Figure 8.1

PIF Editor for standard mode

To display this dialog in 386 enhanced mode, choose Standard from the Mode menu in PIF Editor



The remaining options set flags and reserve shortcut keys to control how the application works in switching back to Windows:

- **No Screen Exchange** prevents copying the screen to the Clipboard with Print Screen. This is the same as reserving the PRTSC and ALT+PRTSC keys.
- **Prevent Program Switch** prevents switching back to Windows to conserve memory. You have to quit the application to return to Windows. This is the same as reserving all shortcut keys or selecting any Directly Modifies option.
- **Close Window On Exit**, if unchecked, keeps the window open when you quit the application until you press a key to return to Windows. This option displays any available messages when you quit.
- **No Save Screen** prevents Windows from updating the screen when you switch back to the application, which saves memory. Use this option only if the application saves its own screen information.
- **Reserved Shortcut Keys** saves any checked shortcut keys for use by the application, instead of carrying out Windows actions. As soon as you switch back to Windows, the keys once again work for Windows actions.

PIF Settings for 386 Enhanced Mode

For 386 enhanced mode, many of the options in PIF Editor are the same or nearly identical to those for standard mode. Another dialog box, the Advanced Options, contains additional items for specifying more information.

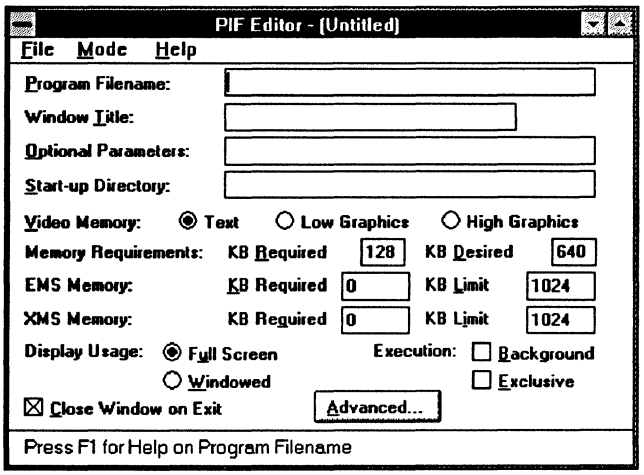
For options that are similar to those for standard mode:

- For video memory, you can differentiate Text (requiring 16K of memory), Low Graphics (equivalent to CGA resolution, requiring 32K of memory), and High Graphics (EGA or higher resolution, requiring 128K of memory). Choose High Graphics and check Retain Video Memory in the Advanced dialog box to reserve the most video memory for the application.
- For system memory, you can also specify expanded memory requirements.
- For reserved shortcut keys, you can reserve additional keys, and you can define a shortcut key for switching to the application from another window.
- The options for EMS, XMS, and video memory are in the basic Virtual Memory dialog box, rather than in the Advanced dialog box.



Figure 8.2

The PIF Editor dialog box for 386 enhanced mode



To display this dialog, in Program Manager click:



PIF Editor

The additional items in the PIF Editor and Advanced Options dialog boxes for 386 enhanced mode are settings that control how the application uses system resources. For details about how these options relate to multitasking as timeslicing, see "How 386 Enhanced Mode Works with MS-DOS" in Chapter 7, "Setting Up Non-Windows Applications."

- Display Usage options specify whether the application runs in full screen, and Execution options specify how the application shares resources. Checking Exclusive can give an application more memory and more processor time.
- Multitasking options in the Advanced Options dialog box specify timeslicing priorities for how much system time the application is given (similar to the Scheduling options for Windows in the 386 Enhanced dialog box in Control Panel). The Background Priority setting is ignored if Execution Background is not checked.

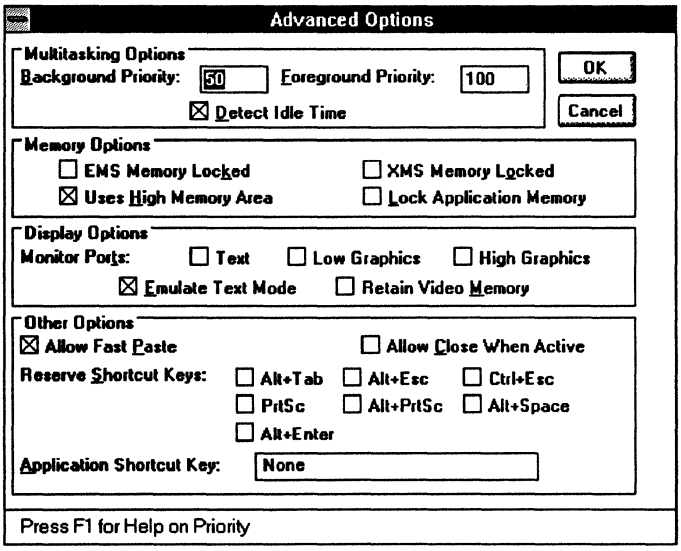
  Check Detect Idle Time to give the application's timeslices to other applications when this application is not processing information.

  Do not check Detect Idle Time for communication applications that work as 3270 or 5251 emulators. Also, clear this option when troubleshooting for an application that runs slowly or doesn't work properly.

Figure 8.3

Advanced Options dialog box for PIF Editor in 386 enhanced mode

To display this dialog, click Advanced in PIF Editor



- Memory Options control whether Windows swaps the application's use of conventional, EMS, or XMS memory to a swap file on the hard disk, and whether the application can be loaded into the HMA.

  Locking any memory option improves performance for that application, but slows down the rest of your system and reduces memory available elsewhere.

- Display Options control video memory. Checking any Monitor Ports option will increase system overhead, but lets Windows monitor whether the application is using the same values as the video adapter for display. Set one of these options if the display isn't restored properly after switching from the application. The other two options, Emulate Text Mode and Retain Video Memory, can speed up the display (at the expense of available video memory).

- Allow Fast Paste lets Windows determine whether it can find the best method for pasting into the application.

- Allow Close When Active lets you quit Windows without having to close each application.

Important Checking Allow Close When Active can cause loss of data or file damage if Windows closes the application before the application closes its files. Do not check this option for databases and word processors, or for any application that does not use standard MS-DOS file handles.

Modifying the Default PIF

Setup always installs _DEFAULT.PIF in the WINDOWS directory to be used as the PIF for applications that do not have custom PIFs. You can use _DEFAULT.PIF as a starting point for creating a PIF for an application. By using _DEFAULT.PIF, you can begin with the common optimal settings for Windows, then experiment to refine the settings for running the application on your system.

To modify the _DEFAULT.PIF file:

1. Run PIF Editor by choosing its icon in the Main group of Program Manager, then open the _DEFAULT.PIF file.
2. Make the changes you need for your application. Your application manufacturer can give you recommendations concerning these changes.
3. Choose Save As from the File menu, and type a new name for the file in the Filename box.

Use the same filename that the application uses, but add a .PIF filename extension. For example, if the application filename is SPECIAL.EXE, use the filename SPECIAL.PIF for the PIF. Save the file in the WINDOWS directory.

Creating Multiple PIFs for a Single Application

You might want to create more than one PIF for a single application. For example, you might want to specify different startup directories for different spreadsheet projects. Or you might want to use expanded memory with a word processor in some cases, but not in others.

To be able to start an application in different configurations:

1. Create two PIFs for the different configuration options.

   For example, specify a different directory in the Start-Up Directory box of PIF Editor for each PIF. Or specify different video memory and system memory options.

2. Give each PIF a distinguishing filename (for example, BUSINESS.PIF and PERSONAL.PIF).

3. Add both PIFs to Program Manager and give them each a distinguishing title (for example, WP-Business and WP-Personal).

Then you can start your application in either configuration by choosing the appropriate program-item icon.

Note If you specify a different directory in the PIF for an application than is specified as the Working Directory for the program-item icon in Program Manager, the Program Manager setting overrides the Start-Up Directory setting in PIF Editor.

Using Windows 3.1



Part
4 Using Windows 3.1

Chapter 9 **Fonts** 309

About Fonts 310
TrueType and Windows 3.1 318
Printer Fonts and Windows 3.1 320
Changing Typeface Names for Soft Fonts 337

Chapter 10 **Printing** 339

About Windows Printing 340
Printer Ports 342
Determining Print Method 349
Printing with a Network Driver 350

Chapter 11 **Compound Documents in Windows 3.1** 351

About Object Linking and Embedding 352
Windows Support for Compound Documents 354
Managing the Registration Database 359

Chapter

9 Fonts

This chapter examines technical issues related to fonts, focusing on TrueType, the new font technology available in Microsoft Windows 3.1. This chapter also presents details about using printer fonts with specific types of printers.

Related information

- *Windows User's Guide:* "Using Fonts" in Chapter 5, "Control Panel"; Appendix A, "Special Characters"

  Also see the PRINTERS.WRI file in your WINDOWS directory.

- *Windows Resource Kit:* Chapter 2, "The Windows Setup Information Files"; Chapter 3, "The Windows Files"

Contents of this chapter

About Fonts 310
- About Typography in Windows 310
- About Windows Fonts 311
- Font Installation 315
- How Windows Matches Fonts 316

TrueType and Windows 3.1 318
- How TrueType Works 318
- TrueType and Earlier Windows Applications 320

Printer Fonts and Windows 3.1 320
- Character Sets for Printer Fonts 321
- Dot Matrix Printer Fonts 322
- HPPCL Printer Fonts 323
- Printer Fonts for Other HP Printers 331
- PostScript Printer Fonts 333
- Canon Printer Fonts 336
- IBM 4019 Laser Printer Fonts 337

Changing Typeface Names for Soft Fonts 337

About Fonts

Flowchart Series 3
Font Problems

This section summarizes the typographical terms related to fonts for Windows applications. If you are already familiar with many aspects of fonts in Microsoft Windows, see the next section, "TrueType and Windows 3.1," for a detailed discussion of the new Windows fonts.

The Windows font-resource files are stored in the Windows SYSTEM subdirectory. These files are described in Chapter 3, "The Windows Files."

About Typography in Windows

A *typeface* is a set of characters that share common characteristics such as stroke width and the presence or absence of serifs. For example, Arial and Courier are each typefaces. Frequently, both the typeface and its name are copyrighted by the typeface designer or manufacturer.

In Windows, a *font* is the name of a typeface, not including attributes such as bold or italic (which is a more general definition than in traditional typography). For example, MS Serif is a font in Windows 3.1.

In Windows, a *font family* refers to a group of typefaces with similar characteristics. The families that Windows recognizes for font installation and mapping are Roman, Swiss, Modern, Script, and Decorative. For example, the sans serif typefaces Arial, Arial Bold, Arial Bold Italic, Arial Italic, Small Fonts, and MS Sans Serif are all part of the Swiss font family.

For printing and display in a computer system, each font has its own *character set* according to the ASCII, ANSI, or OEM standard or other industry standard that defines what character is represented by a specific keystroke. Windows uses the ANSI character set. Many non-Windows applications use the ASCII character set. For details about special character sets in Windows, see Appendix A, "Special Characters," in the *Windows User's Guide.*

These basic terms are used in Windows to define the appearance of a font in an application:

- **Font style** refers to specific characteristics of the font. The four characteristics you can define for fonts in Windows are *italic,* **bold,** ***bold italic,*** and roman (often called Normal or Regular in Font dialog boxes).
- **Font size** refers to the point size of a font, where a point is about 1/72 of an inch. Typical sizes for text are 10-point and 12-point.

- **Font effects** refers to attributes such as underlining, strikeout, and color that can be applied to text in many applications.

You may also encounter these terms in descriptions of fonts and typefaces:

- **Pitch** refers to type size for fixed-width fonts, specified in characters-per-inch (CPI), where 10-pitch = 12-point, 12-pitch = 10-point, and 15-pitch = 8-point type.
- **Serif** and **sans serif** describe specific characteristics of a typeface. Serif fonts such as Times New Roman or Courier have projections that extend from the upper and lower strokes of the letters. Sans serif fonts such as Arial and MS Sans Serif do not have serifs.
- **Slant** refers to the angle of a font's characters, which can be *italic* or roman (no slant).
- **Spacing** can be either fixed or proportional. In a fixed font such as Courier every character occupies the same amount of space. In a proportional font such as Arial or Times New Roman, character width varies.
- **Weight** refers to the heaviness of the stroke for a specific font, such as Light, Regular, Book, Demi, Heavy, Black, and Extra Bold.
- **Width** refers to whether the standard typeface has been extended or compressed horizontally. The variations are Condensed, Normal, or Expanded.
- **X-height** refers to the vertical size of lowercase characters.

About Windows Fonts

Windows 3.1 provides three basic kinds of fonts, which are categorized according to how the fonts are rendered for screen or print output:

- **Raster fonts** are stored in files as bitmaps and are rendered as an array of dots for displaying on the screen and printing on paper. Raster fonts cannot be scaled or rotated.
- **Vector fonts** are rendered from a mathematical model, where each character is defined as a set of lines drawn between points. Vector fonts can be scaled to any size or aspect ratio.



- **TrueType fonts** are outline fonts using new technology available in Windows 3.1. They can be scaled and rotated.

Besides the font-rendering mechanism, Windows fonts are described according to the output device:

- **Screen fonts** are font descriptions that Windows uses to represent characters on the display devices.
- **Printer fonts** are the font descriptions used by the printer to create a font. Windows applications can use three kinds of printer fonts: **device fonts, downloadable soft fonts**, and **printable screen fonts**, as described in "Printer Fonts and Windows 3.1" later in this chapter.

The rest of this section describes raster and vector screen fonts, and describes how Windows chooses which font to print or display. The next section, "TrueType and Windows 3.1," discusses details about TrueType fonts.

Windows System Screen Fonts

Windows uses special raster fonts as the system screen font for menus, window captions, messages, and other text. A set of system, fixed, and OEM terminal fonts are shipped with Windows 3.1 to match your system's display capabilities (that is, for CGA, EGA, VGA, or 8514 video displays). The default system screen font in Windows 3.1 is System, a proportionally spaced raster font.

The installed system screen fonts are listed in the **[fonts]** section of your WIN.INI file. For information about the display device resolution of specific Windows system fonts, see "Font Files" in Chapter 3, "The Windows Files."

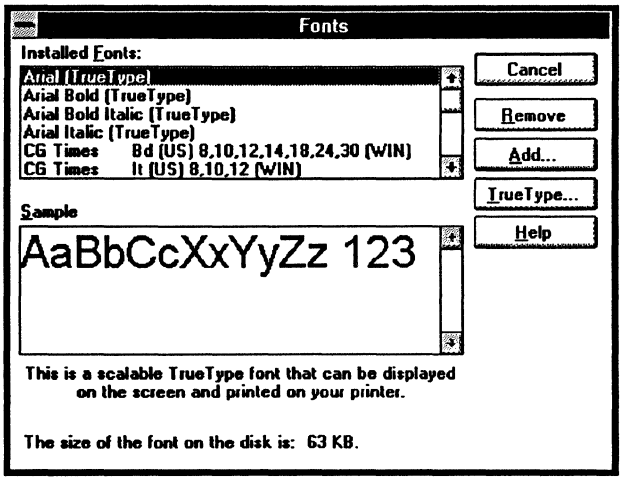
Figure 9.1

Fonts dialog box for installing TrueType and screen fonts

To display this dialog, in Control Panel, click



Fonts

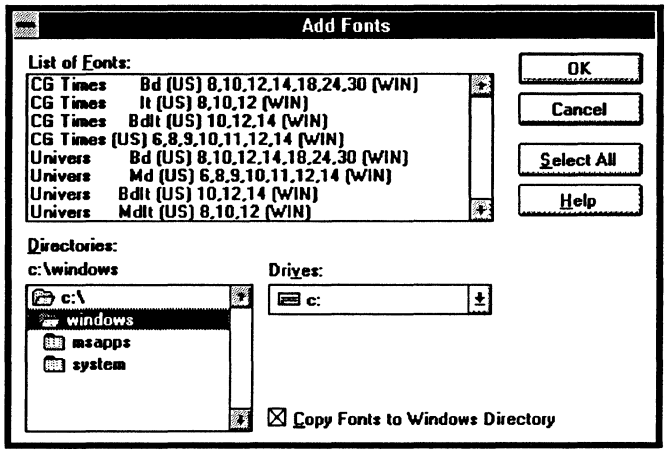


Some screen fonts are installed for displaying non-Windows applications when Windows is running in 386 enhanced mode. By default, code page 437 (U.S.) fonts are installed. Other screen font files are included for international language support, identified by the code page number appended to the file-name. For a list of these files with their associated code page translation tables, see "Font Files" in Chapter 3, "The Windows Files."

Figure 9.2

Add Fonts dialog box for installing screen fonts

To display this dialog, choose the Fonts icon in Control Panel, then click Add in the Fonts dialog box



Windows Raster Fonts

Raster fonts are bitmaps supplied in different sizes for specific video display resolutions. The Windows fonts MS Serif, MS Sans Serif, Courier, System, and Terminal are raster fonts.

A raster font file contains data that describes all the characters and style of a typeface for a specific display device. Windows provides several raster font sizes for various display devices. For example, MS Serif comes in point sizes 8, 10, 12, and 14 for CGA, EGA, VGA, and 8514 display devices.

Windows can scale raster fonts to even multiples of their supplied sizes. For example, MS Serif can be scaled to 16, 20, 24, and so on. Bold, italic, underline, and strikeout styles can also be generated from a standard raster font, but if you try to scale them too far from their original size or style, they become jagged.

The correct font sets for your display and printer are usually installed by Windows Setup. Additional raster font sets can be installed with the Control Panel. The following is a list of the raster fonts installed in Windows 3.1.

Font	Filename	Character set
Courier	COURx.FON	ANSI
MS Sans Serif	SSERIFx.FON	ANSI
MS Serif	SERIFx.FON	ANSI
Small	SMALLx.FON	ANSI
Symbol	SYMBOLx.FON	Symbol

The raster font sets for different display resolutions are distinguished by a letter suffix on the font name (represented as *x* in the previous table). Add the letter from the following table that identifies the resolution to the raster font filename to see the files that Windows installs for a given display or printer. For example, the resource file for MS Serif fonts for VGA is named SERIFE.FON.

Font set and output device	Horizontal resolution	Vertical resolution	Aspect ratio H:V
A = CGA display	96 dpi	48 dpi	2:1
B = EGA display	96 dpi	72 dpi	1.33:1
C = Printer	60 dpi	72 dpi	1:83
D = Printer	120 dpi	72 dpi	1.67:1
E = VGA display	96 dpi	96 dpi	1:1
F = 8514 display	120 dpi	120 dpi	1:1

Raster fonts can also be printed if their resolution and aspect ratio are close to what your printer requires. If you do not see raster fonts for your printer in a Fonts dialog box, check your printer's horizontal and vertical resolution and compare it with the table above. If there is a close match, choose the Fonts icon in Control Panel and make sure the appropriate font set is installed. If there is no close match, you cannot print the Windows raster fonts on your printer.

You might be able to print raster fonts in a different resolution, if the other resolution has an aspect ratio that matches your printer. Some printer drivers cannot print raster fonts, regardless of the aspect ratio.

In Windows 3.1, MS Serif and MS Sans Serif replace the identical raster fonts Tms Rmn and Helv that were installed in earlier versions of Windows. Windows matches MS Serif to Tms Rmn and MS Sans Serif to Helv through the **[FontSubstitutes]** section of WIN.INI. You will still see the Tms Rmn and Helv typeface names in a Fonts dialog box if, for example, your HPPCL printer uses the Microsoft 1Z font cartridge.

The new Windows raster font named Small Font was designed for readable screen display of small fonts. For sizes under 6 points, Small Font is a better choice than any TrueType font for screen display, because it's easier to read.

You can also purchase raster fonts as both screen and printer fonts that work with Windows. Font vendors include Bitstream Fontware, Hewlett-Packard Type Director, Adobe Type Library, and SoftCraft WYSIFonts. Windows raster fonts can also be created using the FontEdit utility from the Windows Software Development Kit or the Publisher's Type Foundry from ZSoft Corporation.

Windows Vector Fonts

Vector fonts are a set of lines drawn between points, like a pen plotter drawing a set of characters. They can be scaled to virtually any size, but generally they do not look as good as raster fonts in the sizes that raster fonts are specifically designed for.

Vector fonts are stored in Windows as collections of GDI calls and are time-consuming to generate. But these fonts are useful for plotters and other devices where bitmapped characters can't be used. Before TrueType, vectors fonts were also used in some applications to create large characters or characters that were rotated or distorted from the baseline.

Some Windows applications automatically use vector fonts at larger sizes. Some applications allow you to specify at what point size you want to use vector fonts. For example, the "Vector Above" setting in Aldus PageMaker specifies the point size at which PageMaker will switch to vector fonts.

The Windows fonts Roman, Modern, and Script are vector fonts. Although the vector fonts use the ANSI character set, they are marked internally as an OEM character set.

Font Installation

In Windows 3.1, fonts can be installed in your system in several ways:

- Windows installs TrueType and its screen fonts automatically during installation. When you specify a printer and other options in the Printer Setup dialog box, Windows includes information about font cartridges and built-in fonts for your printer.
- Install more TrueType fonts from disks by choosing the Add Fonts button in the Font Installer dialog box.
- Install more HPPCL soft fonts on your hard disk by installing the AutoFont Support files and following the instructions for adding scalable printer fonts. Then choose the Add Fonts button in the Font Installer dialog box to install the fonts in Windows.

- Install other third-party soft fonts on your hard disk by using the utility supplied by the manufacturer. Then choose the Add Fonts button in the Font Installer dialog box to install the fonts in Windows.
- Install a new font cartridge in your printer, and choose the Printer icon in Control Panel. In the Setup dialog box, choose a new item from the Cartridge list.

For more information about using the Font Installer, choose the Help button in the dialog box. Other details about using the Font Installer are discussed in "Printer Fonts and Windows 3.1" later in this chapter.

How Windows Matches Fonts

When an application requests characters to print or display, Windows must find the appropriate font to use from among the fonts installed on your system. Finding the font can be complex because, for example, your document may contain fonts that aren't available on the current printer, or there may be more than one font with the same name installed on your system.

The basic rules that Windows uses for finding a font are:

- If the font is a TrueType font, then TrueType renders the character and the result is sent to the display or to the printer.
- If the font is not a TrueType font, then Windows uses the font mapping table to determine the most appropriate device font to use.

Before TrueType, when Windows mapped fonts that had the same name, the order of the Windows internal listing of fonts determined which font was chosen. In Windows 3.1, TrueType fonts are always chosen first, then the internal listing order is followed.

When Windows uses the font mapping table to match screen fonts to printer fonts, the characteristics used to find the closest match are, in descending order of importance: the character set, variable versus fixed pitch, family, typeface name, height, width, weight, slant, underline, strikeout.

The following table shows which types of Windows fonts can be printed on different kinds of printers.

Printer type	Device fonts	Raster fonts	Vector fonts	TrueType fonts
Dot Matrix	Yes	Yes	No	Yes
HPPCL	Yes	No	Yes	Yes
PostScript	Yes	No	Yes	Yes
Plotter	Yes	No	Yes	No

The following table lists the character sets installed with Windows 3.1.

Font	Font type, spacing, and default sizes
Arial Bold Italic	TrueType, proportional, scalable
Arial Bold	TrueType, proportional, scalable
Arial Italic	TrueType, proportional, scalable
Arial	TrueType, proportional, scalable
Courier New Bold Italic	TrueType, fixed, scalable
Courier New Bold	TrueType, fixed, scalable
Courier New Italic	TrueType, fixed, scalable
Courier New	TrueType, fixed, scalable
Courier	Raster, fixed, 10,12,15
Modern	Vector, proportional, scalable
MS Sans Serif	Raster, proportional, 8,10,12,14,18,24
MS Serif	Raster, proportional, 8,10,12,14,18,24
Roman	Vector, proportional, scalable
Script	Vector, proportional, scalable
Small	Raster, proportional, 2,4,6
Symbol**	Raster, proportional, 8,10,12,14,18,24
Symbol**	TrueType, proportional, scalable
System	Raster, proportional, display-dependent size
Terminal*	Raster, fixed, display-dependent size
Times New Roman Bold Italic	TrueType, proportional, scalable
Times New Roman Bold	TrueType, proportional, scalable
Times New Roman Italic	TrueType, proportional, scalable
Times New Roman	TrueType, proportional, scalable

* OEM character set, rather than ANSI character set
** Symbol character set, rather than ANSI character set

For information about changing font names to avoid duplicate names in your system, see "Changing Typeface Names for Soft Fonts" later in this chapter.

TrueType and Windows 3.1



Windows 3.1 includes a new implementation of outline font technology called TrueType. TrueType has many benefits over other kinds of Windows fonts:

- What you see is really what you get, because Windows uses the same font for both screen and printer. You don't have to think about whether you have a specific point size for a particular printer or for your display.
- You can scale and rotate TrueType fonts, and they look good in all sizes and on all output devices that Windows supports.
- Your document will look the same when printed on different printers. And any printer that uses a Windows 3.1 universal driver can print TrueType fonts.
- Your document will look the same if you move it across platforms. For example, the text you format in Microsoft Word for Windows will look the same if you open the same document in Word for the Macintosh.
- Each TrueType typeface requires only an .FOT and a .TTF file to create fonts in all point sizes at all resolutions for all output devices. (Raster fonts need separate files for each point size, resolution, and display device).
- TrueType is integrated with the operating environment, so all Windows applications can use TrueType fonts like they do other Windows raster fonts without changes or upgrades.

The TrueType fonts installed with Windows 3.1 are Arial, Courier New, Times New Roman, and Symbol in regular, bold, bold italic, and italic.

How TrueType Works

Flowchart 3.1 TrueType Fonts

TrueType fonts are stored as a collection of points and "hints" that define the character outlines. When a Windows application requests a font, TrueType uses the outline and the hints to render a bitmap in the size requested. Hints are the algorithms that distort the scaled font outlines to improve how the bitmaps look at specific resolutions.

Each time you run Windows, the first time you select a TrueType font size, TrueType renders a bitmap of the selected characters for display or printing. Because of this, the initial font generation may be slower than with Windows raster fonts. However, Windows stores the rendered bitmaps in a font cache, so each subsequent time the font is used during that Windows session, display or printing will be just as fast as for a Windows raster font.

The Windows universal printer driver supports TrueType. Any printer that works with the universal printer driver will support TrueType automatically.

Using TrueType in Windows Applications

With TrueType, you have more choices for fonts in most Windows applications, plus the same fonts you had in earlier versions of Windows. In many applications, TrueType fonts appear in the Fonts dialog box with a "TT" logo beside the typeface name. Typefaces that are device fonts have a printer icon beside their names in the list.

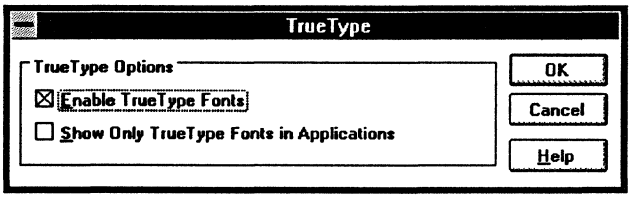
You will also notice that you can specify any size that you want for TrueType fonts, rather than choosing from a limited list of raster or vector font sizes.

You can choose the TrueType button in the Fonts dialog box in Control Panel to specify that you want to use TrueType fonts or restrict all choices to only TrueType. If you restrict all choices to TrueType, you will ensure that type styles in your documents will print on any dot matrix, HPPCL, or PostScript printer and that your documents can be moved to other platforms easily.

Figure 9.3

TrueType dialog box

To display this dialog, click TrueType in the Fonts dialog box



Windows 3.1 does not automatically change fonts in documents that were produced with earlier font technologies. If you want to update old documents to use TrueType fonts, you must update them manually. You might also contact your application vendor to see if there are new utilities available that will assist automatic upgrading of documents to use TrueType.

Disk Space, Memory Use, and Speed

You may notice a performance decrease if your document uses many fonts in many sizes. Rendering many fonts will require a large font cache, which might force more swapping to the hard disk. This same problem occurred with other fonts in earlier versions of Windows. With TrueType, less memory is used for the cache than would be required for corresponding raster fonts. So this should lead to a net performance gain. The font cache will only use more memory with TrueType if multiple logical fonts have been mapped to the same raster font. Usually, however, any additional swapping to disk is still faster than discarding the rendered bitmaps.

Hard disk space is not the problem for TrueType fonts that it can be for a comparative selection of raster fonts. When you install Windows, however, you will see that more disk space is being used to store fonts. This is because all the Windows raster fonts are still shipped for compatibility with earlier applications. Any soft fonts you already have on your hard disk will not be affected by the installation of TrueType with Windows 3.1.

The Windows limit on the number of TrueType fonts that can exist simultaneously on your system is 1,170.

TrueType and Earlier Windows Applications

TrueType uses a different character spacing (called ABC widths) than was used for raster fonts. Applications that use this spacing will be able to space characters more accurately, especially for bold and italic text. However, because of this change in spacing, text can sometimes appear inaccurately in applications written for earlier versions of Windows. For example, the end of a highlighted text line might look odd on screen.

Most applications list the font names on menus and in dialog boxes that will match the names of fonts that can be printed on the current printer. So the change in font names for Windows 3.1 will probably not affect you. You will only see more choices in most Fonts dialog boxes.

Note Windows 3.1 maps the typefaces Helvetica to Arial and Times to Times New Roman in the **[FontSubstitutes]** section of WIN.INI. You can change this section of WIN.INI to map any font names to other font names.

Printer Fonts and Windows 3.1

A printer font is any font that can be produced on your printer. There are basically three kinds of printer fonts:

- **Device fonts** are fonts that actually reside in the hardware of your printer. They can be built into the printer itself or can be provided by a font cartridge or font card.
- **Printable screen fonts** are Windows screen fonts that can be translated for output to the printer.
- **Downloadable soft fonts** are fonts that reside on your hard disk and are sent to the printer when needed.

Not all printers can use all three types of printer fonts. Plotters, for example, cannot use downloadable soft fonts. HPPCL printers cannot print Windows screen fonts.

To take advantage of TrueType and to make other improvements, the printer drivers for the many different kinds of dot matrix, HPPCL, and PostScript printers have been replaced by new universal printer drivers. Instead of seeing the specific name of your printer in the Printer Setup dialog box in Control Panel or Print Manager, you will see a generic description of your printer.

For more information about using the Font Installer and other elements of the Printer Setup dialog box, see "Installing and Configuring a Printer" in Chapter 5, "Control Panel," of the *Windows User's Guide.* Or choose the Help button in the Printer Setup dialog box.

Character Sets for Printer Fonts

Windows uses the ANSI character set. Some printers use the IBM (OEM) standard for codes above 128, such as the IBM Proprinter. Other printers might use their own proprietary set of extended character codes.

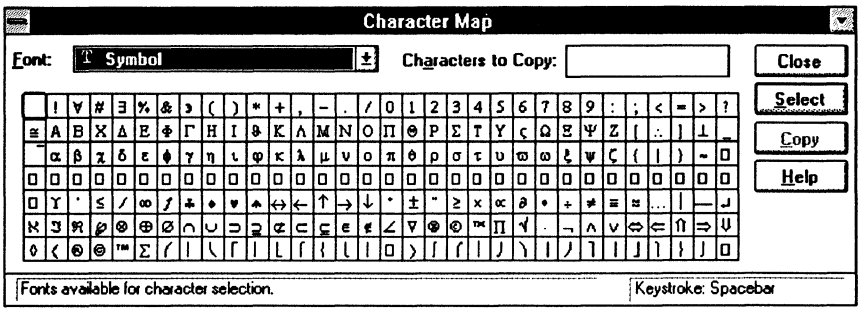
To be sure you get the characters you want, consult your printer documentation for the character set supported by the printer. Then see Appendix A, "Special Characters," in the *Windows User's Guide* for instructions on how to enter codes from the keyboard for special characters.

You can also use the Windows Character Map to select and insert special characters in your document.

Figure 9.4



Character Map



When you insert special characters in a document to print, the character you see on the screen might not be correct, because it is displayed using the ANSI character set and the best matching screen font for the current printer font. However, the printed document will contain the correct character. Conversely, if you type an ANSI character that appears on screen but is not supported in your printer fonts, some other character will be printed instead.

Dot Matrix Printer Fonts

Dot matrix printers support device fonts and printable screen fonts. Usually, a dot matrix printer includes only a limited range of internal device fonts. Typically fixed-spacing fonts are supplied in a variety of characters-per-inch (CPI) sizes, and are conventionally named *typeface xx*CPI, where *typeface* is the typeface name, and *xx* is the number of characters per inch. Distinguishing a device font on a dot matrix printer is usually as easy as checking for the CPI designation at the end of the font name, such as "Courier CPI 10."

Through the universal printer driver, dot matrix printers can also support TrueType. When you use TrueType fonts on a dot matrix printer, Windows sends a rasterized graphics image to the printer.

Dot matrix printers do not provide any landscape device fonts, but vector screen fonts can be printed in any resolution or orientation. Dot matrix device fonts are faster but less flexible than screen fonts.

Dot matrix printers are distinguished as either 9-pin or 24-pin printers:

- **9-pin dot matrix printers** such as the Epson 9-pin and IBM Proprinter series usually print in a 1.67:1 aspect ratio. The Windows Epson 9-pin driver supports resolutions of 120x72 (1.67:1 aspect ratio), 120x144 (1:1.2), and 240x144 (1.67:1). The first and last of the three resolutions can print raster screen fonts using the D font set (120x72 dpi). The same font set is available in half-point sizes for the 240x144 resolution. The aspect ratios reverse in landscape orientation, so fewer fonts will be available.
- **24-pin dot matrix printers** such as the Epson 24-pin and IBM Proprinter 24 series can print in 120x180 resolution (1:1.5 aspect ratio), 180x180 (1:1), and 360x180 (2:1). Some others, such as the NEC 24-pin, provide a 360x360 resolution. The 180x180 resolution is usually best for printing raster screen fonts. In 180x180 resolution, these printers can print 1:1 aspect ratio screen fonts, such as the E set (96x96 dpi) and the F set (120x120 dpi). E set fonts will be available at about 50 percent, and F set fonts at 75 percent of normal point sizes. A true 180x180 dpi screen font set is available by special order from Epson of America.

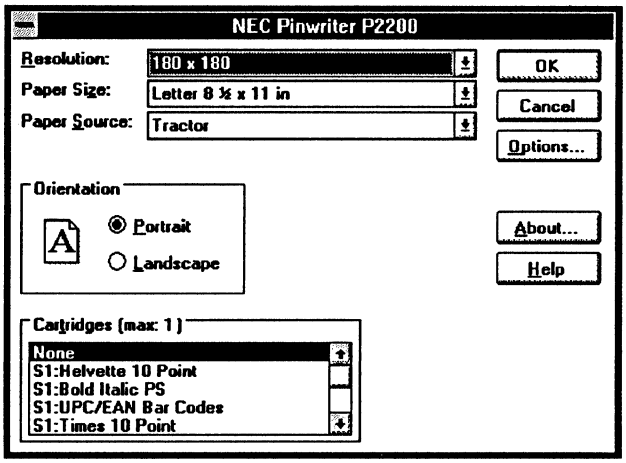
Some 24-pin dot matrix printers such as the Epson and NEC printers also support font cards or cartridges. You can use these fonts if the Windows driver for that printer supports them.

Figure 9.5

Printer Setup dialog box for dot matrix printers with cartridges

To display this dialog, click the Setup button in the Printers dialog

Choose a cartridge in the list box to match the font cartridge in your printer



HPPCL Printer Fonts

Printers that use the Hewlett-Packard Printer Control Language (HPPCL) can print several different types of fonts. HPPCL printers can use font cartridges, downloadable soft fonts, vector screen fonts, and TrueType fonts.

HPPCL printers cannot print Windows raster screen fonts.

When you use TrueType fonts on an HPPCL printer, TrueType performs all the font rendering in the computer and downloads bitmaps of the fonts to the printer. TrueType downloads only the specific characters needed in a document, not the entire font.



If you use an HP LaserJet-compatible printer, be sure to specify accurately the amount of memory installed in your printer. This is important because the Windows HPPCL driver now tracks the available memory in your printer. You may get an out-of-printer-memory error or other errors if the memory is specified incorrectly.

Font Cartridges

Hewlett-Packard LaserJet-compatible font cartridges are supplied by numerous manufacturers, including Hewlett-Packard, Pacific Data Products, and IQ Engineering. Some cartridge vendors will also produce custom font cartridges to your specifications.

Windows treats font cartridges as device fonts because they are always with the printer. Font cartridges can be selected in the Printer Setup dialog box. The HPPCL driver available with Windows 3.1 can support all HP font cartridges.

If you want to add a font cartridge that came out after the printer driver was written, you may need a printer cartridge metrics (.PCM) file. A .PCM file tells Windows the characteristics of the new font and is installed with the Font Installer in the same way as soft fonts. After a .PCM file is installed, a new entry appears in the Cartridges dialog box of the Printer Setup dialog box.

For new HP cartridges, contact Microsoft Product Support Services for the appropriate .PCM file. For third-party cartridge support, contact the cartridge vendor. If you want to develop support for your own custom font cartridges, custom .PCM files can be generated with the Windows Printer Font Metric Editor. The PFM Editor Kit is also available from Microsoft Product Support Services, but the Editor requires in-depth technical knowledge of the fonts contained in a cartridge.

Downloadable Fonts

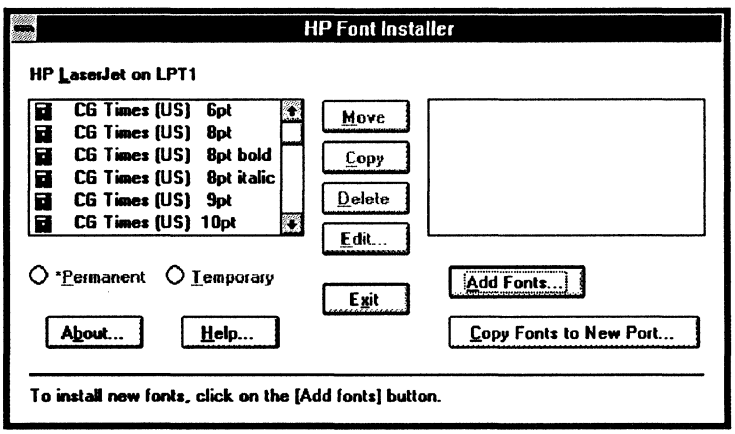
You can get HP LaserJet-compatible downloadable soft fonts from a number of sources, including Hewlett-Packard, Bitstream, SoftCraft, and CompuGraphics. Some downloadable font utilities also generate raster screen fonts for Windows. If an exact screen font match is not available, Windows uses one of its own screen fonts.

Hewlett-Packard downloadable fonts are installed with the Font Installer, while third-party HPPCL soft fonts are installed with their own installation utility. To use the Font Installer, choose the Fonts button in the Printer Setup dialog box.

Figure 9.6

HP Font Installer dialog box for LaserJet-compatible printers

To display this dialog, click the Fonts button in the Printer Setup dialog



The Font Installer places soft font entries in WIN.INI under a section specific to a driver and port, such as **[HPPCL,LPT*x*]** (where *x* is the port number) as described later in this section. Because soft fonts are installed for a printer on a specific port, the soft fonts will not appear if you change the printer. To copy the soft font listings to the other port, choose the Copy Fonts To New Port button in the Font Installer dialog box.

HPPCL fonts can be downloaded on either a temporary or a permanent basis.

Temporary fonts are downloaded only when the HPPCL driver encounters a particular font while printing. At the end of the print job, the soft font is discarded from the printer's memory. Printers such as the Apricot Laser and Kyocera F-1010, which require temporary soft fonts to be downloaded only at the start of a print job but not during the job, cannot use soft fonts with the Windows HPPCL driver.

Font entries in the WIN.INI for temporary soft fonts are similar to this:

```
[HPPCL, LPTx]
SoftFont1=C:\PCLFONTS\OPPR0090.PFM,C:\PCLFONTS\OP90BPN.USP
```

The .PFM file is the Windows Printer Font Metrics file, used by Windows applications to determine the size, weight, and spacing of the font in question. The .USP file is the soft font you want to download to the printer.

Permanent fonts stay resident in the printer until the printer is turned off. Fonts can be marked as permanently downloaded by selecting fonts in the Font Installer dialog box and choosing the Permanent option. A dialog box prompts you to choose Download Now or Download At Startup options.

If you choose the Download Now option, the selected fonts will be downloaded to the printer when you choose the Exit button to dismiss the dialog box. Windows then sends a hard reset to the printer, which forces the printer to delete all previously downloaded fonts. So if you mark some fonts as Permanent and choose Download Now and then repeat the procedure, only the fonts downloaded the second time will be resident in the printer.

When you choose the Download At Startup option, Windows performs the following tasks:

1. The PCLSF0YN.EXE program is saved on your hard disk, and a batch file named SFLPT*x*.BAT is created to run PCLSF0YN.EXE.

   This program prompts you to download permanent soft fonts:

   ```
   Download PCL fonts to ? [y/n]
   ```

   If you answer Yes, the program downloads the fonts and (if you have a network) prints the banner page.

2. A line is added to your AUTOEXEC.BAT to run the batch file whenever you boot the computer. For example:

   ```
   rem The Windows HP LaserJet/DeskJet font installer added the next line
   command /c C:\PCLFONTS\SFLPT1.BAT
   ```

   The added line uses **command/c** to run the batch file so that AUTOEXEC can continue executing after running the batch file.

3. The soft font listings in WIN.INI are modified to remove everything to the right of the comma, because the fonts will have already been downloaded whenever Windows checks this WIN.INI entry. For example, this temporary soft font entry:

   ```
   SoftFont1=c:\pclfonts\oppr0090.pfm,c:\pclfonts\op90bpn.usp
   ```

   becomes:

   ```
   SoftFont1=c:\pclfonts\oppr0090.pfm
   ```

Note The SFLPT*x*.BAT file uses the MS-DOS TEMP environment variable. If the **set TEMP=** statement in AUTOEXEC.BAT does not set a valid location for TEMP, the batch file will not work correctly, and the fonts might not be downloaded.

Recovering Soft Fonts After Reinstalling Windows

The Font Installer copies the fonts to a directory on your hard disk (usually C:\PCLFONTS) and also creates a .PFM file for the soft font. This file is used by Windows to determine the metrics of the font (such as family weight). When you reinstall Windows, the soft font entries in WIN.INI are lost. This usually means you will have to run the Font Installer and have it recopy the soft fonts and regenerate the .PFM files.

But there is an easier way. The Font Installer can generate a summary file named FINSTALL.DIR that contains the WIN.INI font entries. This file can be used to regenerate the WIN.INI entries without the time-consuming reinstall process. To keep this file up to date, create a new FINSTALL.DIR after adding or removing any soft fonts.

The process for creating FINSTALL.DIR is not documented in the *Windows User's Guide,* and is not described in the online help for the Font Installer.

To create a FINSTALL.DIR file:

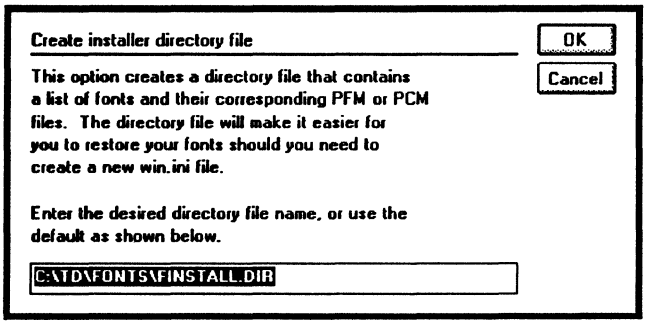
1. In the Printer Setup dialog box, choose the Fonts button to display the Font Installer.
2. In the Font Installer dialog box, hold down the CTRL and SHIFT keys and click the Exit button.

   A dialog box appears that you can use to create the FINSTALL.DIR file.
3. The edit box proposes a directory and filename for saving FINSTALL.DIR. You can accept the default or specify a different directory. The directory that contains the downloadable fonts is the best choice.
4. Choose the OK button to create the FINSTALL.DIR file.

Figure 9.7

Create Installer Directory File dialog box

To display this dialog, press CTRL+SHIFT while clicking Exit in the HP Font Installer dialog



To reinstall soft fonts with FINSTALL.DIR after reinstalling Windows:

1. In the Printer Setup dialog box, choose the Fonts button to display the Font Installer.
2. In the Font Installer dialog box, hold down the CTRL and SHIFT keys while choosing the Add Fonts button to display a special Add Fonts dialog box.
3. In the edit boxes, specify the drive, directory, and filename for the FINSTALL.DIR file. Then choose the OK button.

   The Font Installer dialog box now displays the fonts in the right list box.
4. Select the desired fonts and choose the Move button.

   When the prompt asks for a target directory, specify the same directory where the fonts are currently installed.

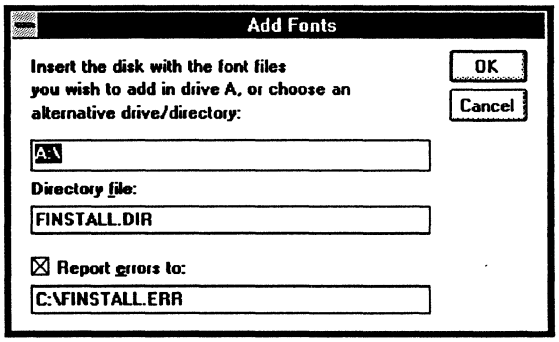
   The fonts now appear in the left list box, indicating they are installed in Windows.

In this procedure, Windows doesn't copy any fonts, but merely updates the WIN.INI file with the soft font listings contained in the FINSTALL.DIR file.

Figure 9.8

Add Fonts dialog box for reloading the fonts installer directory file

To display this dialog, press CTRL+SHIFT while clicking Add Fonts in the HP Font Installer dialog



HPPCL Driver Font Summary File

The Windows HPPCL driver maintains an internal font database in memory. Each time you choose an HPPCL printer as the current printer, the HPPCL driver must either build an internal font database or read the database from a previously created Font Summary file. Each time the Font Summary file must be rebuilt, this message appears:

Building font database. Please wait.

This building-font-database message appears when the HPPCL driver is set up, when fonts are added, or when connections are changed in Control Panel, but not for each print job.

When the HPPCL driver builds an internal font database, it attempts to write the database to a Font Summary file, so it can subsequently read the information from the file instead of rebuilding the database. If the driver can read the database from the Font Summary file, it will not display the building-font-database message. The driver saves the Font Summary file in the directory that contains HPPCL.DRV and then adds an entry in WIN.INI for this location. For example:

```
[HPPCL,LPT1]
FontSummary=C:\WINDOWS\FSLPT1.PCL
```

Do not change the path for the file in the **FontSummary=** line in WIN.INI. The driver would just continue to rebuild the Font Summary file in the directory that contains HPPCL.DRV. The only way to change the location of the Font Summary file is to change the location of HPPCL.DRV.

The driver might display the building-font-database message for each print job for these reasons:

- The HPPCL driver might be in a directory where the Font Summary file cannot be created, such as a read-only directory or a network directory for which you don't have write or create privileges.

  To correct this problem, either get access to the directory that contains HPPCL.DRV or move HPPCL.DRV to a directory that you can access.

- The WIN.INI file might be corrupt or in a directory that you can't access. Or duplicate WIN.INI files might exist. If the driver can't add the **FontSummary=** entry in WIN.INI, it can't read the Font Summary file, and therefore it will rebuild the font database each time.

  To correct this problem, verify that only one copy of WIN.INI exists, that it has the proper entries for printers, and that it is in a directory you can access. If necessary, reinstall Windows to verify the integrity of WIN.INI.

- You might have placed the entry **MaxFontSummary=0** in the driver-specific section of WIN.INI (for example, under the **[HPPCL,LPT1]** section). If **MaxFontSummary=0**, the driver will never build a Font Summary file, so it will have to build the font database each time you print.

  To correct this problem, set **MaxFontSummary=1** or remove this entry from WIN.INI.

HPPCL Soft Font Limitations

HPPCL printers have a limit of 16 soft fonts per page. (This limit applies only to soft fonts, not to cartridge fonts.) If you send a page that contains more than 16 soft fonts to an HPPCL printer, the following message appears:

PCL PRINTING WARNING: SOFT FONT PAGE LIMIT:
Some fonts will be substituted

Choose the OK button to continue printing.

An Error 20 message might appear on the front panel of the HPPCL printer when printing a document that contains soft fonts. This also indicates that you tried to download more fonts than the printer's memory can hold. You can recover from this error by pressing the Continue button on the printer control panel. The soft font that caused the error is not downloaded and will not print.

To avoid this error, reduce the number of fonts that you try to download, or add more memory to your printer. Also make sure you haven't downloaded any permanent soft fonts that are taking up memory in the printer.

The maximum number of soft fonts you can install is limited by the maximum size of WIN.INI, not by the Font Installer itself.

About Printer Fonts for HPPCL Printers

In Windows 3.1, all HPPCL printers are supported by the universal HPPCL.DRV driver. There are four basic levels of HPPCL printers: LaserJet, LaserJet Plus/LaserJet II, LaserJet IID and IIP, and LaserJet III, IIID, and IIIP.

- **HP LaserJet.** The original HP LaserJet had only one font cartridge and did not support downloadable fonts. These units are fairly rare today, as most were upgraded to the level of the LaserJet Plus.
- **HP LaserJet Plus, LaserJet II, and compatibles.** These printers are similar except the LaserJet II uses different escape codes to select the paper tray and supports two font cartridges. Most LaserJet-compatible printers are compatible with the LaserJet Plus. As a default, these printers contain only Courier and LinePrinter device fonts. To print additional fonts, cartridges or downloadable fonts are required.

  The LaserJet Plus and LaserJet II require separate soft fonts for Portrait and Landscape use. The HP-compatible downloadable fonts used by all HPPCL printers are bitmaps, so separate downloadable files must be installed for each point size you want.

- **HP LaserJet IID and IIP.** These printers have the same capabilities as previous models plus font rotation capabilities. Portrait orientation fonts can be printed in landscape mode, and vice versa. Any device font or soft font available to the printer can be rotated without installing separate portrait and landscape fonts for the IID and IIP. Also, the LaserJet IID can print double-sided. Both printers include a few more resident font styles than previous models, but these fonts are still limited to Courier and LinePrinter typefaces.
- **HP LaserJet III.** The LaserJet III has all the capabilities of previous models, plus a scalable font technology. Two scalable outline typefaces are built into the LaserJet III: CG Times and Swiss (similar to MS Serif and MS Sans Serif), in regular, bold, italic, and bold italic. The LaserJet III also has the standard nonscalable Courier and LinePrinter bitmap fonts and supports the newer generation of standard Hewlett-Packard bitmap font cartridges and standard nonscalable PCL soft fonts.

Additional LaserJet III scalable outline fonts are available from Hewlett-Packard as cartridge or downloadable soft fonts. With the HPPCL driver in Windows 3.1, downloadable outline fonts can be installed with the Font Installer.

Printer Fonts for Other HP Printers

Windows 3.1 also includes fonts for Hewlett-Packard DeskJet and PaintJet printers, and for plotters.

HP DeskJet Printer Fonts

The HP DeskJet Printer is an ink-jet printer. The Windows 3.1 driver for the Hewlett-Packard DeskJet and DeskJet Plus printers supports Windows vector screen fonts, DeskJet internal fonts, soft fonts, and TrueType. Both DeskJet printers can print at resolutions of 75, 150, and 300 dpi. Without font cartridges, the DeskJet includes only the built-in Courier and LinePrinter fonts. Cartridges can be selected in the Printer Setup dialog box. At this time, font cartridges for DeskJet printers are available only from Hewlett-Packard.

DeskJet soft fonts are installed with the Font Installer. To use downloadable fonts on the DeskJet printers, you must install either HP22707A or HP22707B RAM cartridges. When you set printer memory in the Printer Setup dialog box, make sure to specify the total amount of RAM cartridge memory if more than one cartridge is installed.

DeskJet internal, downloadable, and cartridge fonts will not work in landscape orientation. This is a hardware, not a driver, limitation. For landscape mode, print with Windows vector screen fonts such as Modern or Roman.

If you use the HP DeskJet 500 driver from Hewlett-Packard, add the entry **prtresfac=0** to the **[DJ500,port]** section of WIN.INI.

HP PaintJet Printer Fonts

The HP PaintJet is a color ink-jet printer. The Hewlett-Packard PaintJet driver in Windows 3.1 composes a full page at a time in 180x180 dpi resolution and outputs the page to the PaintJet as a large bitmap. This produces the highest possible quality of output, but results in very large temporary spool files if you use Print Manager. For acceptable printing speed, we recommend that you choose the Fast Printing Direct To Port option in the Connect dialog box in Printer Setup. This option bypasses Print Manager.

The PaintJet driver supports the printing of PaintJet internal fonts, Windows raster and vector screen fonts, PaintJet soft fonts, and TrueType. The same considerations apply for printing raster screen fonts on the PaintJet as for using the 24-pin dot matrix printers in 180x180 dpi resolution (see "Dot Matrix Printer Fonts" earlier in this chapter). PaintJet soft fonts are not downloadable fonts. They are used internally by the driver, which places them as necessary into the full-page bitmap during page composition. The font itself is never sent to the printer, except as part of the full-page bitmap.

PaintJet soft fonts, which have a .PJF filename extension, are installed with the Font Installer. Windows 3.1 supports PaintJet soft fonts for Courier 10-CPI and Letter Gothic 12-CPI and 18-CPI. Additional soft fonts can be obtained from Hewlett-Packard. Scalable PaintJet soft fonts are also available from Hewlett-Packard in the HP Color PrintKit (HP part number 17390A).

HP Plotter Printer Fonts

Because plotters are vector devices, they can print only vector fonts. Plotters cannot print any kind of bitmap, including raster screen fonts and TrueType fonts. HP plotters include one internal vector font called Plotter. The Windows vector screen fonts Modern, Roman, and Script can also be printed on HP plotters.

PostScript Printer Fonts

All PostScript fonts are scalable outlines that can be printed at any size. PostScript outline fonts can also be rotated to any angle and can be printed in both portrait and landscape modes. However, font size limits are often imposed by applications. A common PostScript font size limit in an application is 127 points.

Most PostScript printers include either the standard Apple LaserWriter Plus set of 35 scalable fonts or the earlier Apple LaserWriter set of 17 fonts.

PostScript soft fonts are installed with utilities provided by the soft font vendor. Because the fonts are scalable, if there isn't a comparable screen font, mismatches can occur between screen display and printed output.

PostScript printers cannot print Windows raster screen fonts, although they can print vector screen fonts. Printing of Windows screen fonts is not usually necessary because of the large selection of resident fonts in a PostScript printer.

LaserWriter Plus Typefaces

The LaserWriter Plus standard font set includes 11 typefaces, 8 of which are available in roman, bold, italic, and bold italic. The Symbol typeface contains mathematical and scientific symbols; Zapf Chancery is a calligraphic font; and Zapf Dingbats contains decorative bullet characters and embellishments. These typefaces are available only in roman style.

PostScript Printers and TrueType

TrueType fonts are treated as downloaded fonts by the PostScript driver. When you use TrueType fonts on a PostScript printer, scaling and hints are always performed in the computer. Scan conversion can be done in the computer or in the printer, depending on the point size. At smaller point sizes, TrueType performs scan conversion in the computer; at larger point sizes, scan conversion is done in the printer.

You can specify how to send TrueType fonts to your printer, for example, as bitmaps or in Adobe Type 1 format. To do this, in the Advanced Options dialog box of Printer Setup, choose the method in the Send To Printer As list that you want to use for sending TrueType fonts to the printer.

If your PostScript printer supports downloaded fonts, you might want to use printer fonts in place of TrueType fonts to speed up printing and to use less printer memory. To do this, in the Advanced Options dialog box of Printer Setup, select the Use Printer Fonts For All TrueType Fonts option.

You can also map a TrueType font to a PostScript font in the **[FontSubstitutes]** section of WIN.INI. This will increase printing speed, but the results on the display may not be exactly the same as the printed output.

If your PostScript printer does not support downloaded fonts, you must use printer fonts to print any TrueType fonts in your documents. There are two ways to do this: allow the PostScript driver to print using the printer fonts that most closely match the TrueType fonts, or follow the steps described in the section "Substituting PostScript Fonts" later in this chapter.

Specifying Virtual Printer Memory

You can change the amount of virtual memory that your PostScript printer has available for storing fonts. The PostScript driver uses a default setting recommended by the printer manufacturer for virtual memory.

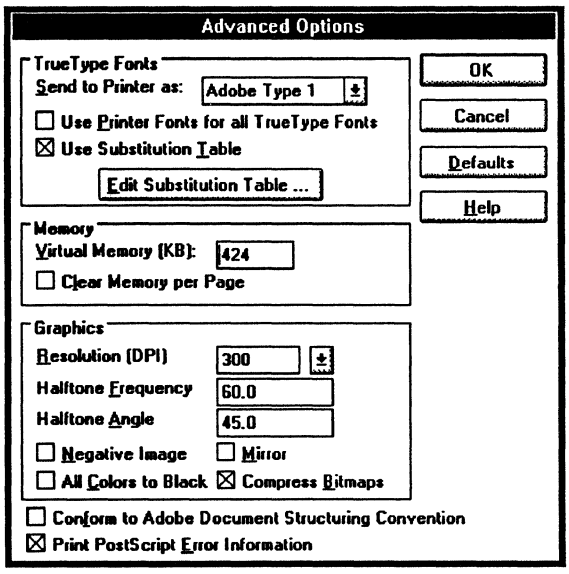
To find out how much virtual memory your printer has, print the TESTPS.TXT file in the WINDOWS directory. To adjust the amount of virtual memory, in the Advanced Options dialog box of Printer Setup, type the amount of virtual memory you want to use in the Virtual Memory (KB) box. For details, see Chapter 4, "Troubleshooting," in the *Windows Getting Started* booklet.

Figure 9.9

Advanced Options dialog box for PostScript printers

To display this dialog, click Advanced in the Options dialog

Check the Print PostScript Error Information option in this dialog to get a printout of errors



Substituting PostScript Fonts

You can edit the Substitution Table to specify which PostScript printer fonts you want to print in place of the TrueType fonts in your documents. The changes you make in the Substitution Table only affect the fonts that are printed. The fonts that appear on the screen will not change; the original TrueType fonts are still used to display TrueType text in your document. (To substitute screen fonts, change the **[FontSubstitutes]** section of WIN.INI.)

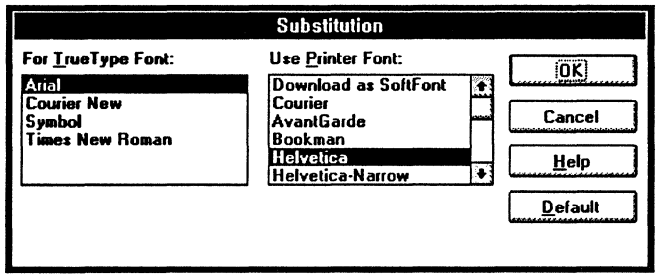
To specify which printer fonts to use, choose the Edit Substitution Table button in the Advanced Options dialog box of Printer Setup. Then select the TrueType font you want to replace from the For TrueType Font list in the Substitution dialog box. From the Use Printer Font list, select the PostScript printer font you want to use instead of the selected TrueType font.

If your printer supports downloaded fonts, you can choose the Download As Soft Font option. In this case, the selected TrueType fonts will be sent to the printer using the method you specified in the Send To Printer As list in the Advanced Options dialog box. Repeat these steps until you have selected printer fonts to use in place of all the TrueType fonts in your document.

Figure 9.10

Substitution dialog box

To display this dialog, click Edit Substitution Table in the Advanced Options dialog box



PostScript Drivers

In Windows 3.1, most PostScript printers use PSCRIPT.DRV, the universal driver. If you install a PostScript printer that does not appear in List of Printers in the Printers dialog box, you need to install a Windows PostScript Definition (.WPD) file for your printer. To do this, choose the Printers icon in Control Panel. Then choose Add Unlisted or Updated Printer from List of Printers in the Printer Setup dialog box.

Windows 3.1 requires an OEMSETUP.INF file for version 3.1 of a new .WPD file. See Chapter 6, "Print Manager," in the *Windows User's Guide* for more details about this procedure. If you have a .WPD file for Windows 3.0, then you do not need a new OEMSETUP.INF file to install the .WPD file.

In Windows 3.1, the PostScript driver can detect errors that Print Manager cannot detect. You can specify whether to print information about these errors after your document has finished printing. This information may help you or a Product Support Service representative determine what caused the error. To print out PostScript error information, in the Advanced Options dialog box of Printer Setup, choose the Print PostScript Error Information option.

PostScript Downloadable Outline Fonts

PostScript printers also accept downloadable outline fonts, which can be scaled to any size and printed in both portrait and landscape orientations. Downloadable PostScript fonts are available from several suppliers, including Adobe and Bitstream. Both Adobe and Bitstream supply utilities that install the fonts and add entries to WIN.INI. Because this capability is included with these commercial font products, the PostScript driver in Windows does not include a font installation facility.

Although PostScript downloadable outlines can be scaled to any size, Windows screen fonts cannot. You must install specific sizes of Windows screen fonts with the Adobe and Bitstream utilities. Install only the sizes you feel you will frequently use. If you specify a PostScript font size that does not have a corresponding screen font, Windows will substitute another screen font. This results in a little loss in display quality but, of course, no loss in print quality.

PostScript Cartridges

To use PostScript cartridges with Windows, you must use the PostScript printer driver. Choose the Printer icon in Control Panel and follow the steps for installing a printer, choosing the Apple LaserWriter Plus (PostScript) option or another PostScript printer from the list in the Printer Setup dialog box. PostScript cartridges are not supported directly by the Windows PostScript driver.

Canon Printer Fonts

The Canon Laser Beam Printers LBP-8 Mark III and Mark IV have built-in outline fonts that can be scaled to any size. These printers are capable of printing with scalable device fonts, bitmap device fonts, or Windows vector screen fonts. The two internal scalable typefaces are Dutch (similar to MS Serif) and Swiss (similar to MS Sans Serif). A nonscalable Courier font is also available internally in 8- and 12-point sizes. The Windows driver for the Canon Mark III and IV supports the Canon BM1, BM2, and BM3 bitmap font cartridges, as well as the SC1 scalable outline font cartridge.

TrueType fonts and downloadable fonts are not supported for the Canon Series II and III. If you are using one of these printers and want to print TrueType fonts, choose the Enable TrueType option in the Printer Setup dialog box.

IBM 4019 Laser Printer Fonts

The IBM Laser Printer model 4019 can print internal device fonts, IBM downloadable fonts, font cards, Windows vector screen fonts, and TrueType fonts. The printer includes Courier and Boldface Proportional internal fonts. Downloadable fonts are installed with the Font Installer and can be marked as Permanent or Temporary in the Font Installer dialog box. Downloadable fonts and font cards are available from IBM.

Changing Typeface Names for Soft Fonts

If you purchase a TrueType font or other soft font that has the same name as a font you are already using, you should use the Font Installer to change the name of the font. You cannot change the name of a cartridge font.

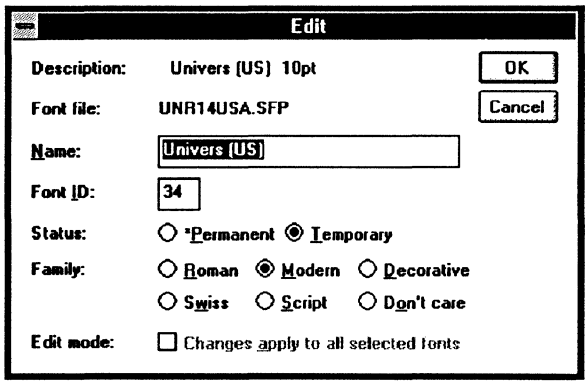
To change a soft font name:

1. In the Font Installer dialog box, select the font you want to edit from the list of installed soft fonts.
2. Choose the Edit button. In the Edit dialog box, type a new name for the selected font in the Name box.

Figure 9.11

Edit dialog box

To display this dialog, select a font name in the HP Font Installer dialog and click the Edit button



3. Choose the OK button.

   The new name for the selected font now appears in the Font Installer dialog box and in the Windows applications that use the font.

4. Choose the Exit button.

Important Do not change any Font ID or Family settings that Font Installer automatically enters unless you have experience creating fonts or manually downloading soft fonts.

The options in the Edit dialog box vary depending on whether you are installing a font that Font Installer doesn't recognize or whether you are editing the name of a currently installed font. For details about the options, choose the Help button in the dialog box.

If the Font Installer doesn't recognize the name or family of a font that you select to install, the Font Metrics dialog box appears so that you can enter the information. We strongly recommend that you do not change the settings provided by the font manufacturer. Use this option only if absolutely necessary.

To enter a font name in the Font Metrics dialog box:

1. Type the manufacturer's name for the font in the Name box.
2. Select the family the font belongs to.
3. Choose whether you want to install the soft font files permanently or temporarily.

Chapter

10 Printing

This chapter examines technical issues related to printing in Microsoft Windows 3.1. This chapter provides a technical supplement to the basic Windows documentation.

Related information

- *Windows User's Guide:* "Installing and Configuring a Printer" in Chapter 5, "Control Panel"; Chapter 6, "Print Manager"

  Also see the PRINTERS.WRI and NETWORKS.WRI files in your WINDOWS directory.

- *Windows Resource Kit:* Chapter 2, "The Windows Setup Information Files"; Chapter 3, "The Windows Files"

- *Glossary terms:* BIOS, buffers, interrupts, spooler

Contents of this chapter

About Windows Printing 340
Printer Ports 342
  Printing from Parallel Ports 342
  Printing from Serial Ports 343
  WIN.INI [Ports] Settings and the LPT*x*.DOS Method 345
  Port Entries in the ROM BIOS Data Segment 346
Determining Print Method 349
Printing with a Network Driver 350

About Windows Printing

Flowchart Series 4 Printing Problems

To print a document or other data in a Microsoft Windows application, you must have a printer properly connected to a serial or parallel port (for a stand-alone PC) or properly connected to the network. You must also have a Windows printer driver file (.DRV) installed and configured for your printer.

Most printer drivers are installed and configured automatically during Setup based on your choices about the type of printer in your system. If the Printer Setup list does not contain your printer, use the following decision tree to find how to set up unsupported printers.

Unsupported Printer Compatibility Decision Tree

- Is it a laser printer?

  HPPCL (LaserJet) compatible?
  →Set up as a Hewlett-Packard LaserJet Plus.

  PostScript compatible?
  Color PostScript?
  →Set up as a QMS-ColorScript.
  35-font Plus font set or superset thereof?
  →Set up as an Apple LaserWriter Plus.

- Is it a dot matrix printer?

  9-pin dot matrix?
  IBM compatible?
  →Set up as IBM Proprinter.
  Epson compatible?
  →Set up as Epson FX-80 for narrow, or FX-100 for wide carriage.

  24-pin dot matrix?
  Compatible with Epson or IBM 24-pin printers?
  IBM 24-pin compatible?
  →Set up as IBM Proprinter X24.
  Epson LQ compatible?
  →Set up as Epson LQ-1500.

- Is it a plotter?

  HPGL compatible?
  →Set up as Hewlett-Packard 7475A.

- Other?

  →Use Generic/Text Only/Daisywheel driver.
  Contact the manufacturer to see if custom drivers are available.

Figure 10.1

Printers dialog box

To display this dialog, in Control Panel click:



Printers

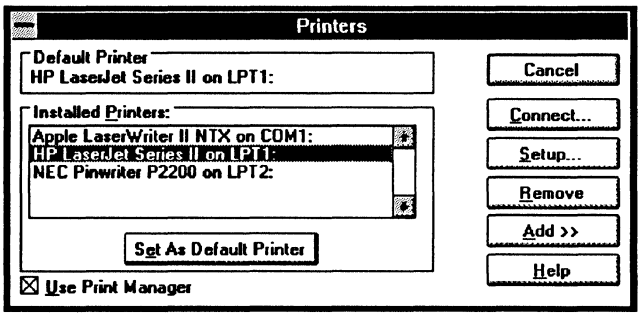
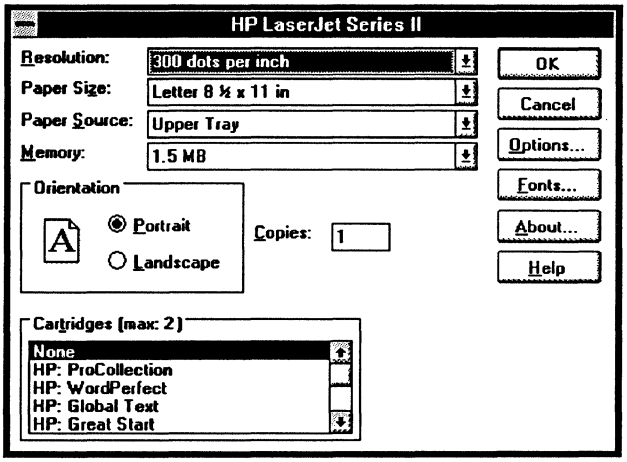


Figure 10.2

Printer Setup dialog box for HP LaserJet Series II and compatible printers

To display this dialog, click Setup in the Printer Setup dialog



For details about installing and configuring a printer, see Chapter 5, "Control Panel," in the *Windows User's Guide*. Or choose the Help button in the Printer Setup dialog box.

Printer Ports

Printers in stand-alone PC systems are usually connected either to parallel or serial ports. This section examines technical details related to LPT and COM ports for stand-alone systems.

A special case, the EPT port, is described in "WIN.INI [Ports] Settings and the LPT*x*.DOS Method" later in this chapter.

Printing from Parallel Ports

Standard IBM-compatible systems can have up to three parallel printer ports, commonly known as LPT ports. As the name suggests, parallel ports transmit data a full 8-bit character at a time. Data is transmitted to LPT ports through system I/O ports, which are individual serial data lines to the processor. Because LPT ports transmit a byte at a time, they use eight I/O ports in parallel.

The standard for the pin connections on parallel ports was developed by Centronics. Parallel port handshaking is done with hardware signals on the Centronics standard system. These signals include:

- **Select In**, which means that the printer is online.
- **Busy**, which means the printer's buffer is full or the printer is occupied.
- **Acknowledge**, which is pulsed when the printer accepts a character.
- **Paper Out**, which means the printer is out of paper or jammed.

An application can access LPT ports in three ways:

1. Non-Windows applications usually use the MS-DOS device names **LPT1:**, **LPT2:**, and **LPT3:** with standard MS-DOS Interrupt 21 file-access services. Printing with MS-DOS device names is both the slowest and most compatible printing method.
2. Applications can access the printer through the ROM BIOS Interrupt 17 printer-port services, which sends one character to the printer at a time (an inefficient process). When you access a printer at MS-DOS level, MS-DOS eventually performs LPT port output through Interrupt 17.
3. LPT ports can be addressed directly, which requires 100 percent hardware compatibility, but is the fastest by far of the three methods.

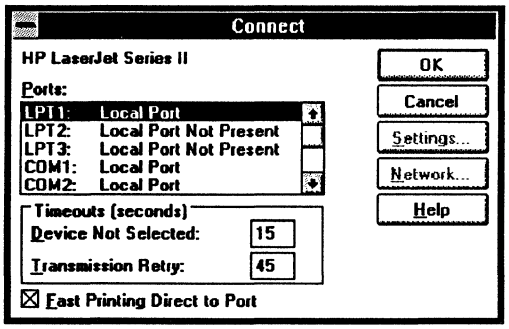
The following table lists the default IBM-compatible I/O port starting locations and hardware interrupts for parallel ports.

Port	PC-AT default	PS/2 default	Interrupt
LPT1	378h	3BCh	7
LPT2	278h	378h	5
LPT3	3BCh	278h	7

Specify a parallel port connection for your printer by choosing the Connect button in the Printer Setup dialog box.

Figure 10.3

Connect dialog box



Printing from Serial Ports

Flowchart 2.2
Serial Port Problems

Serial ports, also known as COM ports, transmit data in a 1-bit-wide stream. Windows 3.1 accesses COM ports directly. The following table lists the default IBM-compatible I/O port locations and hardware interrupts for serial ports.

Port	Default	Interrupt
COM1	3F8h	4
COM2	2F8h	3
COM3	2E8h	4
COM4	2E0h	3



For 386 enhanced mode, you can set nonstandard values by choosing the Ports icon in the Control Panel, and clicking the Advanced button in the Settings dialog box for the selected COM port. (This also changes the entries for **COM*x*BASE=** and **COM*x*IRQ=** in the **[386enh]** section of SYSTEM.INI.) However, we strongly advise you to change the hardware to fit the Windows defaults rather than changing the Windows defaults to fit the hardware.

Figure 10.4

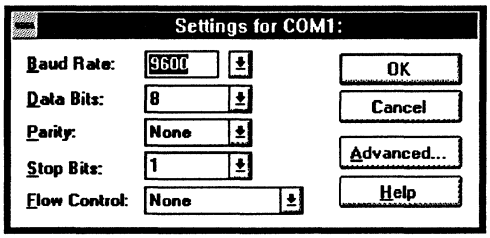
COM Port Settings dialog box

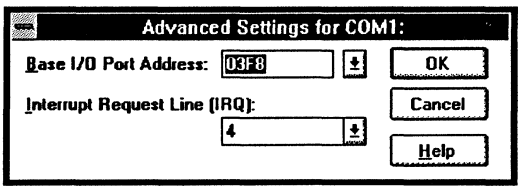
To display this dialog, in Control Panel, click



then click Settings

Click the Advanced button to display this Advanced COM Port Settings dialog box





In 386 enhanced mode, COM ports are buffered for non-Windows applications, but buffering is not necessary for Windows itself. Buffer size can be set only by changing the **COM*x*Buffer=** setting in the **[386enh]** section in SYSTEM.INI.

The key to proper serial printing is correct communications settings. Serial communication requires handshaking, which defines a method for a printer to tell Windows that its buffer is full. Serial ports can be configured for No Handshaking, XON/XOFF (software) Handshaking, or Hardware Handshaking by choosing the Ports icon in the Windows Control Panel. The settings are saved in the **[ports]** section of WIN.INI, as described in the next section.

Most serial print devices use hardware handshaking, although some can be configured for XON/XOFF. We recommend that you do not use the No Handshaking setting for a serial printer. Your printer's documentation should tell you what communications settings to use. Usually, the setting is 9600 baud, No parity, 8 bits, 1 stop bit, and Hardware Handshaking.

WIN.INI [Ports] Settings and the LPTx.DOS Method



The values that can be listed in the **[ports]** section of WIN.INI are described in Chapter 4, "The Windows Initialization Files." The default entries are:

```
[ports]
LPT1:=
LPT2:=
LPT3:=
COM1:=9600,n,8,1
COM2:=9600,n,8,1
COM3:=9600,n,8,1
COM4:=9600,n,8,1
EPT:=
FILE:=
```

Windows only recognizes the entries **LPT1:=** through **LPT3:=**, and **COM1:=** through **COM4:=** as hardware port entries. Windows performs tests to determine how to print to the LPT ports (see "Determining Print Method" later in this chapter). The COM ports are always accessed directly. The **FILE:=** entry is specially recognized by Windows. When you connect a printer using **FILE:=**, Windows will prompt you for a filename for the print job.

Any valid MS-DOS device name or filename can be placed in the **[ports]** section. Windows prints to any entry in the **[ports]** section (other than the **LPT*x*:=** and **COM*x*:=** entries) using standard MS-DOS services. The **EPT:=** entry is an example of this. EPT ports require a special card and special software and are only used by certain printers, such as the IBM Personal Pageprinter. Because the **EPT:=** entry has no special meaning to Windows, Windows simply outputs to EPT as a standard file or device through MS-DOS. If you have printer drivers that are recognized at the MS-DOS level by a device name, you can place them in the **[ports]** section and print to the device.

The **LPT*x*:=** entries are also valid MS-DOS device names. But if they have entries in the **[ports]** section in the standard format, Windows does not treat them as MS-DOS device names. However, you can force Windows to print to LPT ports through MS-DOS by changing the format of their entries in the **[ports]** section.

To make Windows stop recognizing a port as a hardware port, give it a three-letter filename extension instead of a colon in the related LPT entry in the **[ports]** section of WIN.INI. Windows will print to the port as an MS-DOS device instead. For example, you can change the entry **LPT1:=** to **LPT1.DOS=** to accomplish this. Any three-letter extension will do, but the letters "DOS" are used as an example in the *Windows Resource Kit*. If you upgraded from Windows 3.0, this entry might appear in your WIN.INI file as "LPT*x*.OS2."

You might think this type of entry would print to a file called LPT1.DOS, which is exactly what happens as far as Windows is concerned. Windows opens the file/device LPT1.DOS at MS-DOS level and sends printer output to it. At the MS-DOS level, any name beginning with the letters "LPT1" is a synonym for the LPT1 device. So MS-DOS redirects the output to the printer port. Filenames and device names are dealt with synonymously in MS-DOS. The LPT*x*.DOS method is a helpful solution when you are using borderline-compatible network software or parallel port hardware.

Port Entries in the ROM BIOS Data Segment

Flowchart 2.2 Serial Port Problems

Problems can occur if your machine does not recognize a specified COM port. This generates the message "The COM*x* port is currently assigned to a DOS application. Do you want to reassign the port to Windows?"

Information on the I/O-port starting address for each of the installed serial and parallel ports is contained in the ROM BIOS data segment at memory location 0040:0000. Each entry is a 16-bit port address stored in two hexadecimal bytes. As with all 16-bit addresses, each pair of bytes should be reversed to read correctly. For example, the entry **F8 03** yields an address of 3F8 hexadecimal (the standard starting address for COM1).

To ensure that your machine recognizes a COM port, you can examine specific ranges of the ROM BIOS data segment with the MS-DOS Debug command to find correct values for the COM port. Then you can write a batch file script or define SYSTEM.INI entries to correct the problem.

Examining the ROM BIOS Data Segment

The following steps show how to determine if the machine is recognizing the existence of the COM port.

To use the MS-DOS Debug command to examine the BIOS table:

1. At the command prompt (outside of Windows), type **debug** and press ENTER.

   This will return a hyphen prompt (-).

2. At the hyphen prompt, type **d40:0** and press ENTER.

   A listing of the BIOS data appears, and the hyphen prompt returns, where you can type **Q** to quit Debug.

Look at the table of data on the screen. The first row is the important one. It will show a line like the following:

```
0040:0000  F8 03 F8 02 E8 03 00 00-BC 03 78 03 00 00 00 00
```

Everything to the left of the center hyphen are COM ports; everything to the right are LPT ports. This example shows that the computer has COM1 at address 03F8 (listed in reverse byte order), COM2 at 02F8, and COM3 at 03E8. COM4 is not found, displaying 0000 to the left of the hyphen.

If the machine recognizes the desired COM port (identified by the 0000 entries), skip to "SYSTEM.INI Modifications for Port Entries in ROM BIOS" later in this chapter.

Batch File for Port Entries in ROM BIOS

If the machine does not recognize the COM port (identified by the 0000 entries), make a script for Debug to execute in AUTOEXEC.BAT. The following examples for COM ports show this. The second line of the examples should be edited to match the hardware used in your computer. Although the examples given are the most popular, there are exceptions to these.

To create a script for debugging a COM port:

1. Create a batch file, replacing the values shown in the second line of the following table with the values that appear on your screen:

COM1	COM2	COM3	COM4
E40:0	E40:2	E40:4	E40:6
F8 03	F8 02	E8 03	E8 02
q (ENTER)	q (ENTER)	q (ENTER)	q (ENTER)

2. Give the file a name such as C:\FIXCOM3.DEB or C:\FIXCOM4.DEB, depending on which port the file is for.
3. Add a line to the end of AUTOEXEC.BAT such as the following, for COM3:

   ```
   debug < fixcom3.deb > nul
   ```

4. Save the AUTOEXEC.BAT and reboot the machine.

The **> nul** ending added to the line in AUTOEXEC.BAT keeps the script from being echoed back to the screen. You can use Debug again to examine d40:0 to see if the change has taken effect.

SYSTEM.INI Modifications for Port Entries in ROM BIOS

If the port is recognized by the machine, note the address listed when using Debug. Then edit the SYSTEM.INI **COM*x*Base=** lines to match the base I/O address used by the hardware. The examples shown above for COM1 and COM2 will not require any changes to the SYSTEM.INI file. However, the examples below for COM3 and COM4 require the following entries to the **[386enh]** section of the SYSTEM.INI file.

Default	*Value for COMxBase=*
3F8h	COM1Base=3F8h
2F8h	COM2Base=2F8h
2E8h	COM3Base=3E8h
2E0h	COM4Base=2E8h

If you are making adjustments in order to use COM4, you might need to add both the **COM3Base=** and **COM4Base=** lines to SYSTEM.INI; otherwise, **COM4Base=** will be set to the same as the **COM3Base=** default.

If you are using a serial mouse on COM1 and you have an internal modem on COM3, this configuration won't work, because the mouse is tying up the interrupt that you want to be shared by COM3. This behavior also occurs if you use a serial mouse on COM2 and want to use an internal modem on COM4.

Programs that Change the ROM BIOS Data Segment

Port entries in the BIOS data segment can be changed by network redirector software or by the MS-DOS **mode** command. The **mode** command does not change any existing nonzero entries. It places a **1** in port entries that didn't previously exist but that have been created by **mode** redirection. For example, if you have a physical LPT1, but not an LPT3, and you execute the **mode LPT3=LPT1** command, then output sent to LPT3 will be redirected to LPT1. Because LPT3 does not exist in hardware, the BIOS data segment area contains zeroes in the LPT3 section. When **mode** creates a virtual LPT3 through its redirection, it places a **1** in the LPT3 section of the data segment.

MS-Net and LAN Manager redirector-based networks operate in a similar manner. The redirector does not modify any nonzero entries in the data segment when it performs redirection. When you redirect an existing port, the port entry is not changed. This is done so that when the redirection is removed, you can again print to the physical port. However, when you redirect a nonexistent port, the zero entry in its section is changed to a **2**. Many applications check the BIOS data segment to determine whether a port exists. Both **mode** and the MS-Net redirector place nonzero entries in this field as necessary, so the application will function correctly.

Novell networks place a 3BCh entry in the data segment locations for LPT2 and LPT3 when the NetWare shell is loaded. If you do not have a physical LPT1 in the machine, you will have a zero in the port entry for LPT1. This will not cause a problem unless you are running on a Novell network without the Windows Novell Network driver loaded.

Determining Print Method

Windows uses different methods to print depending on the types of parallel ports it detects in your machine. Windows will either access the port directly, or will print by opening the port as an MS-DOS device (with MS-DOS Interrupt 21 services). Windows 3.1 never uses the ROM BIOS Interrupt 17 services for printing.

The following decision tree shows how Windows determines which method to use for printing. This sequence applies only to ports LPT1 through LPT3. Windows always directly accesses the COM ports.

The Windows Printing Method Decision Tree

- If a Windows network driver is present:

  Query network driver to see if printer port is network redirected.
  If it is a network printer:
  →Tag as Network, go to end.

- Examine ROM BIOS data segment at 0040:0008:

  If port entry is zero:
  →Do not allow printing through this port.

  If port entry is not zero:
  If port being examined is LPT2 or LPT3:
  If the port entry is the same as any previous entries:
  →Assume port is redirected and access through MS-DOS.

  If port entry is greater than 100h, port exists:
  →Directly access this port.

  If port entry is less than 100h, port exists but is nonstandard:
  →Access this port through MS-DOS.

- Repeat for additional ports.

Printing with a Network Driver

Flowchart 4.2 Cannot Print on Network

We strongly recommend that if you are on a network, you load the proper Windows network driver. To do this, choose the Windows Setup icon in Control Panel. For basic information about printing on a network, see "Managing Network Printing" in the *Windows User's Guide.*

If the network does not place a nonzero value in a previously nonexistent port entry when it is redirected, Windows will not recognize it as a valid port. Instead, redirect an existing port, and also use the LPT*x*.DOS method as described in "WIN.INI [Ports] Settings and the LPT*x*.DOS Method" earlier in this chapter. The long-term solution in this case is for the network redirector to properly place a nonzero value in the port entry upon redirection.

Also, if the port entry for a redirected port is greater than 100h, Windows will attempt to directly access the port. Directly accessing a redirected port will not work. The method to use in this case is the LPT*x*.DOS method discussed above.

Flowchart 4.4 Garbled Output on Novell Network

Windows itself does not identify network printers. Instead, Windows queries its network driver to make this determination. The network driver then identifies network-redirected printers with whatever internal functionality that particular network contains. For example, the MS-Net driver uses the MS-DOS Interrupt 21 function 5F to detect network printer redirection.

For up-to-date details about printing with a network driver, see the NETWORKS.WRI file in the WINDOWS directory.

Chapter

11 Compound Documents in Windows 3.1

This chapter presents an overview of how Object Linking and Embedding (OLE) works when a Windows application is designed to take advantage of the OLE features of Microsoft Windows 3.1. This chapter also describes how to update and add to the system registration database using REGEDIT.EXE.

Related information

- *Windows User's Guide:* Chapter 4, "File Manager"; Chapter 9, "Write"; Chapter 10, "Paintbrush"; Chapter 13, "Integrating Your Windows Applications"
- *Windows Resource Kit:* Chapter 6, "Tips for Configuring Windows 3.1"
- *Glossary terms:* dynamic data exchange (DDE) and dynamic-link library (DLL)

Contents of this chapter

About Object Linking and Embedding 352
About Compound Documents 352
About Objects, Packages, Clients, and Servers 353
Windows Support for Compound Documents 354
OLE and File Manager 355
OLE and the Clipboard 355
OLE and the Edit Menu Commands 357
Managing the Registration Database 359
Using the Registration Info Editor 360
Restoring the Registration Database 363

About Object Linking and Embedding

You can share data between Windows applications in several ways:

- You can use the Cut and Copy commands to place data on the Clipboard, and then use the Paste command to place the data in another kind of document or file. (You can also use Cut and Paste to share data between Windows applications and many non-Windows applications.)
- In many Windows applications, you can also take advantage of dynamic data exchange (DDE) to retrieve and update information dynamically. For example, you can use the Paste Link command to link data from one Microsoft Excel spreadsheet to another spreadsheet or to a Microsoft Word for Windows document. The data is updated dynamically in the second spreadsheet or Microsoft Word document whenever it is edited in the original spreadsheet.



- For Windows applications that use Object Linking and Embedding (OLE) protocol, you can place data from one application in another kind of document as an object or as an iconic package. Whenever you want to update or view the embedded data, you can double-click it to run the application in which the data was originally created.

Flowchart 5.10 Problems with OLE

This chapter explains how OLE works in Windows 3.1 and how you can take advantage of it to create and use compound documents. The information here provides a broad overview. For details about how a specific Windows application implements OLE, see that application's documentation.

About Compound Documents

Any Windows application that uses OLE can work with other OLE applications to produce documents that contain different kinds of data. You can view or modify data in a compound document without knowing what applications were used to create the original data. Whenever you double-click a part of the compound document to work with it, the application in which it was created will start automatically.

A compound document can contain formatted text, graphics, and data from a spreadsheet or a database. It can also contain icons that run sound recordings or play multimedia devices. The more hardware you have to support creating and playing multimedia through sound boards, CD-ROM, and other media players, the more rich a compound document you can create. But even with a basic PC system, you can take advantage of OLE to create compound documents.

You may have already seen compound documents. For example, a Microsoft Excel spreadsheet with embedded graphics and detail reports is a compound document. The README files for Windows applications such as Microsoft Word for Windows sometimes contain icons in the table of contents that you click for navigation.

Compound documents created using Object Linking and Embedding in Windows is one of the first manifestations of Information At Your Fingertips™, the Microsoft vision for personal computing in the 1990s.

About Objects, Packages, Clients, and Servers

An *object* in a compound document is any encapsulated data that can be displayed and manipulated by a user. For example, a single cell, a range of cells, or an entire spreadsheet could be embedded as an object in a word processing text file. Any data can be an object if it was created in a Windows application that uses OLE.

A *package* is an embedded icon that contains an object, a file or part of a file, or a command.

The *client application* is any Windows application that can accept, display, and store objects. Usually a client application displays a distinctive border or some other visual cue so that you can identify an object embedded in a file. The client application stores some information for embedded objects, such as the target printer and page position, how the object is activated, and which server application is associated with the object.

The *server application* is any Windows application in which you can edit an object when Windows informs it that you selected the object in a client application. Some server applications also use an *object handler*, which is a dynamic-link library that acts as an intermediary between the client and server applications. The object handler for an application is usually used to improve performance, for example, to redraw a changed object in the window of the client application. Some Windows applications are capable of acting as both OLE client and server.

The discussion in this section uses two other terms: a *source document* is the file where the data or object was originally created; the *destination document* is the compound document where the data is embedded or linked.

Windows Support for Compound Documents

These key Windows elements allow you to take advantage of OLE in Windows 3.1:

- The system registration database, which lists file types and their OLE capabilities. For information about what this database contains and how it is maintained, see "Managing the Registration Database" later in this chapter.
- The associated filename extensions for file types, as defined in the Associate dialog box in File Manager. For information about how to add or change associations between extensions and file types, see Chapter 4, "File Manager," in the *Windows User's Guide.*
- The Object Packager, for creating and editing packages. For information about using the Object Packager, see "Working with Object Packager" in Chapter 13, "Integrating Your Windows Applications," in the *Windows User's Guide.*

Windows 3.1 does not rely on the initialization files to support OLE, although changes in the registration database may be reflected in the **programs=** entry in the **[Windows]**, **[embedding]**, and **[extensions]** sections of WIN.INI.

When you want to create compound documents in Windows, you use several Windows features that you are already familiar with:

- The File Manager, which supports drag-and-drop for printing, embedding data, and other activities that use DDE and OLE features.
- The Clipboard, which has enhanced capabilities in Windows 3.1 to support data from OLE applications.
- The Edit menu commands in Windows applications that support embedding and editing objects.

This section describes how File Manager, the Clipboard, and the Edit menu commands support OLE.

OLE and File Manager

In File Manager for Windows 3.1, if a file is associated with a file type in the registration database, you can:

- Double-click the file's icon in the File Manager window to open the file.
- Drag the file icon to the Print Manager to print the file.
- Drag the file icon to place the file as a group item in any Program Manager group.
- Drag the file icon to an application window to embed or link the file as a packaged object.

Note To create file packages with File Manager, the server application must have been written for Windows 3.1. For applications written for earlier versions of Windows, you must use Object Packager. Also, if the package will contain a linked document, you must use File Manager to place the file in Object Packager.

OLE and the Clipboard

When you choose Cut or Copy in any Windows application that can act as an OLE server, the data on the Clipboard can be used to create an embedded or linked object in any client application.

Clipboard Formats

In Windows 3.1, the data on the Clipboard can include a variety of formats:

- Clipboard data formats, such as **Picture** (for a metafile), **Bitmap**, **DIB Bitmap** (for 256-color palettized graphics), **RTF** (for rich-text format), or **CF_Text** (for Clipboard text format). If you run the Clipboard Viewer after copying data, you can see the list of Clipboard data formats for the data.
- The server application's own formats.
- Native format, which is data that completely defines the object and is interpreted only by the server application or its object handler DLL.
- The OwnerLink format, which identifies the owner and class of a linked or embedded object.

- The Presentation format, which the client application can use to display the object in a document. This can be the Native format if the server application uses an object handler DLL.
- The ObjectLink format (for OLE linking) or the Link format (for DDE linking, as in applications created for earlier versions of Windows). These formats identify the linked object's class and source item and document.

The class name in the ObjectLink or OwnerLink format is a unique name for a class of objects in the server application. These class names are registered in the system registration database. A client application uses this data to identify the owner of an embedded object.

Formats for Pasting from the Clipboard

When you paste data into a destination document, several possible results can occur, based on the format of the data on the Clipboard and the capabilities of the client application:

- If you choose Paste in an application that does not support OLE, then the Native, OwnerLink, and ObjectLink formats are ignored. The application pastes the data in the best Clipboard data format it supports.
- If you choose Paste in an OLE client application, the application checks the Clipboard and places the data in the first acceptable format it finds. For example, if Microsoft Word for Windows finds data in RTF format, it pastes the data that way, rather than creating an embedded object.
- If you choose Paste Special in an OLE client application, the dialog box lists the formats that you can choose for pasting the data as an embedded or linked object.
- If you choose Paste Link in an OLE client application, the application looks for ObjectLink information and makes an OLE link to the source document. If ObjectLink format isn't available, the client application looks for the Link format and creates a DDE link.

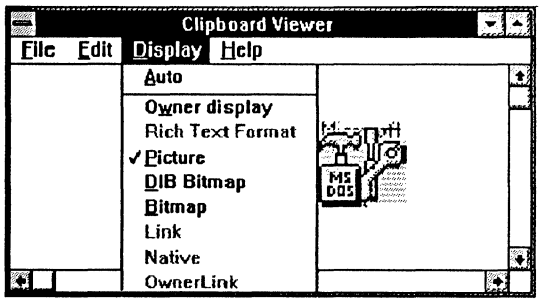
When you copy a linked object to the Clipboard and paste it into another destination document, that other document can take advantage of the original link to the source document, because the information associated with a linked object remains with it every time you copy it.

Figure 11.1

Clipboard Viewer

To display this window, in Program Manager click:





OLE and the Edit Menu Commands

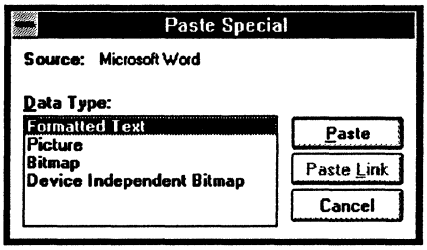
A Windows application that supports DDE links and objects has these commands (or similarly named commands) to support placing and editing embedded objects:

- **Copy** copies data from a source document to the Clipboard. A client application can use this data to create an embedded object and may be able to create a link to the source document.
- **Cut** removes data from a source document and places it on the Clipboard. A client application can use this data to create an embedded object.
- **Paste** places data from the Clipboard into the destination document.
- **Paste Link** inserts a DDE link between a document and the file that contains the object. The object appears in the destination document, but the original data that defines the object is stored in the source document.
- **<Class> Object** runs the server application so you can edit or open a link or embedded object. The text for the *<Class>* placeholder depends on the selected object such as Package Object for a selected package. This command name might also be, for example, Edit Object, depending on which actions are allowed for that object. If the object supports more than one action, choosing this command displays a cascading menu that lists other commands.
- **Links** displays the Links dialog box so you can update linked objects, cancel links, repair broken links, or edit or open the linked object.

- **Paste Special** (an optional command) displays a dialog box so you can choose the data format for the object on the Clipboard and also choose either to paste a link or paste the data without making it an object.

Figure 11.2

An example of the Paste Special dialog box, showing possible paste formats

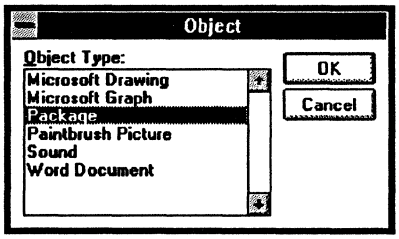


- **Insert Object** (an optional command) displays a dialog box so you can choose which server application to start, then embeds in the destination document the object that is produced by the server. (This is a shortcut for the same result you can get by running the server application, copying the data to the Clipboard, then pasting it in the destination document.)

Figure 11.3

An example of the Insert Object dialog box

Select Package to run Object Packager



After you double-click an object and edit it, you can choose Update from the File menu of the server application to save the changes.

The commands described here are the command names that Microsoft recommends to developers who implement OLE in Windows applications. Check the documentation for your OLE client application to find the actual command names it uses. For examples, see the Edit menus in Write, Paintbrush, and Cardfile.

Managing the Registration Database



This section describes how to maintain and modify the system registration database that is installed with Windows 3.1 to support OLE and other activities related to integrating applications.

The registration database (REG.DAT) provides the information that allows you to open or print files by dragging icons in File Manager and Print Manager. The registration database is also used by applications that support OLE, as described in "Windows Support for Compound Documents" earlier in this chapter.

For example, you might want to place a picture from Paintbrush into a Write file. You can do this in two ways, each of which requires information from the registration database:

- You can choose Insert Object from the Edit menu in Write. Write uses the registration database to list types of objects you can insert. When you choose from the list, Write launches the appropriate application for loading or creating that object.
- You can copy a graphic in Paintbrush, then choose Paste Special from the Edit menu in Write. Write uses the registration database to list the names of possible data types for the graphic you want to paste, and lets you choose to embed or link the graphic.

If you later double-click the embedded graphic to edit its contents, Write uses information from the registration database to launch Paintbrush.

When you first install Windows 3.1, the registration database contains information about all the OLE applications included with Windows, plus entries for Microsoft Word for Windows, Microsoft Excel charts and worksheets, Ami Professional, and Professional Write.

Using the Registration Info Editor

If you have an application that supports OLE but is not registered, you can add it to the registration database with the Registration Info Editor (REGEDIT.EXE). However, you will usually not need to use this utility unless your registration database has been corrupted or deleted.

You can use the Registration Info Editor to add, delete, modify, or restore the information that Windows needs to open and print files from File Manager. To do this, choose File Run in Program Manager, then type **regedit** and press ENTER.

If you need help with any item in Registration Info Editor, press F1 to get general help or click the Help button in any dialog box for specific help.

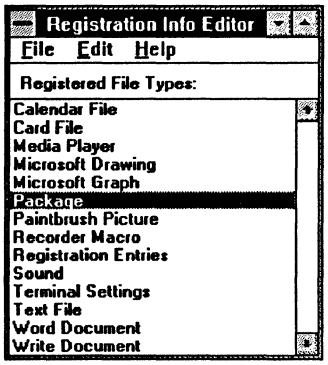
Figure 11.4

Registration Info Editor

To display this window, in Program Manager click:



Regedit



Note To remain compatible with Windows 3.0 applications that use OLE, Windows 3.1 also adds some registration information to the **[embedding]** and **[extensions]** sections of WIN.INI. However, we recommend that you use the Registration Info Editor to add or change OLE server application information.

Installing a .REG File

If a Windows application has a registration file (.REG), that information is usually added to the registration database when you install the application. But if that .REG file is not merged into the registration database, you can add it with Registration Info Editor.

To install a registration file:

1. Choose Merge Registration File from the File menu in the Registration Info Editor.
2. In the dialog box, select the .REG file for the application to be added to the database, and click the OK button.

Tip You can also double-click a .REG file in File Manager to install that registration file automatically.

Adding a New File Type

You can add a new file type to register an application that doesn't have a .REG file by either copying an existing file type or by creating a new file type.

To copy an existing file type to add registration information:

1. Select a file type from the Registered File Types list in the Registration Info Editor window.
2. Choose Copy File Type from the Edit menu.
3. In the dialog box, modify the registration information for the new file type, specifying a unique Identifier. Then click the OK button.

You cannot change the Identifier for an existing file type.

To create a new file type to add registration information:

1. Choose Add File Type from the Edit menu in the Registration Info Editor.
2. In the dialog box, modify the registration information for the new file type, specifying a unique Identifier. Then click the OK button.

Whichever procedure you use to add a file type, you must specify the following information:

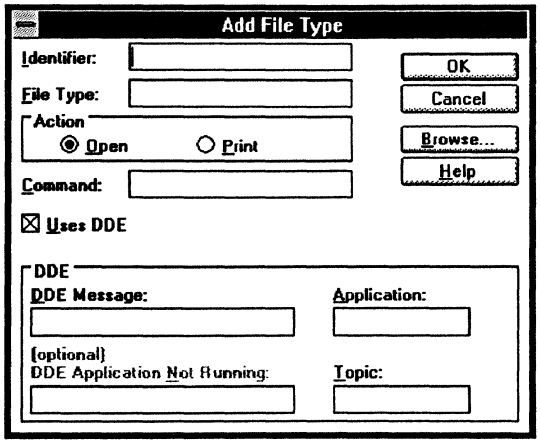
- An Identifier, which must be a unique key word of up to 63 printable ASCII characters. This Identifier is used by Windows to identify the file type in the registration database.
- A File Type, which is the text description that appears in the Registered File Types list in the Registration Info Editor window and in the Associate dialog box in File Manager.
- The action, either Open or Print.
- The command and switches to be executed (or the DDE message to be sent) to perform the action. For example, this command runs the application with the filename being opened or printed (**%1** is the filename parameter):

  pbrush.exe %1

Figure 11.5

Add File Type dialog box

To display this dialog, choose Add File Type from the Edit menu in Registration Info Editor



Check the Uses DDE option in the dialog box if the application sends DDE messages to execute the Open or Print actions. The information in the rest of the dialog box specifies the DDE message and application string, and the DDE topic associated with the command.

Modifying and Removing File Types

Use these commands from the Edit menu in Registration Info Editor to either modify or remove a file type from the registration database:

- Choose Modify File Type, and use the dialog box to change the file type information. You cannot change the Identifier.
- Choose Remove File Type to delete any file type from the database.

Restoring the Registration Database

You can restore the registration database if REG.DAT has been deleted or corrupted.

To restore the registration database:

1. Quit Windows, delete REG.DAT, and then restart Windows.
2. Run Registration Info Editor, which will display an empty list of registered file types.
3. Choose Merge Registration File from the File menu.
4. Select SETUP.REG from the File Name list in the dialog box and click the OK button. (This file is in the Windows SYSTEM subdirectory.)

A message appears to confirm that the information has been registered. The database now contains the original registration information that was installed with Windows. If any of your other applications have .REG files, you can add them to the restored database by choosing Merge Registration File from the File menu, as described earlier.

For more information about the related DDE functions and using the Registration Info Editor to create .REG files, see Chapter 7, "Object Linking and Embedding Libraries," in the *Windows Software Development Kit.*

More About Object Linking and Embedding

Heller, Martin. "Object Links Coming to Windows Applications." *Byte* (February 1991): p. 20.

Networks and Windows 3.1



Part
5 Networks and Windows 3.1

Chapter 12 Networks and Windows 3.1 367

About Networks and Windows 3.1 368
3Com Networks 374
Artisoft LANtastic 376
Banyan VINES 377
DEC Pathworks 379
IBM OS/2 LAN Server 383
IBM PC LAN Program 383
Invisible Software —NET/30 For Windows 384
Microsoft LAN Manager Networks 384
Microsoft Network and Compatibles 387
Novell NetWare 387
TCS 10Net 391
Ungermann-Bass Net/One 392

Chapter

12 Networks and Windows 3.1

This chapter contains some tips about using a network with Microsoft Windows 3.1, plus notes about using specific networks with Windows.

Related information

- *Windows User's Guide:* "Managing Network Printing" in Chapter 6, "Print Manager"; Appendix A, "Windows and Networks"

  See also Chapter 1, "Setting Up Windows," and Chapter 4, "Troubleshooting," in the *Getting Started* booklet, and the NETWORKS.WRI file in your WINDOWS directory

- *Windows Resource Kit:* "Troubleshooting Networks" in Chapter 13, "Troubleshooting Windows 3.1"; "PC-NFS and Windows 3.1" in Appendix D, "Articles"

Contents of this chapter

About Networks and Windows 3.1 368
Running a Shared Copy of Windows 368
Memory, Windows, and the Network 370
Creating Custom User Environments 374
3Com Networks 374
Artisoft LANtastic 376
LANtastic Versions 3.x 376
LANtastic Versions 4.x 377
Banyan VINES 377
Banyan VINES 4.0x 378
Banyan VINES 4.1x 378
DEC Pathworks 379
DEC Pathworks 4.1 379
Questions and Answers about DEC Pathworks 381
IBM OS/2 LAN Server 383
IBM PC LAN Program 383
Invisible Software—NET/30 For Windows 384
Microsoft LAN Manager Networks 384
Microsoft Network and Compatibles 387
Novell NetWare 387
TCS 10Net 392
Ungermann-Bass Net/One 392

About Networks and Windows 3.1



These changes affect how you can install and use Windows 3.1 with a network:

- You can use **Setup /a**, the Administrative Setup option, to easily install the files for a shared copy of Windows on a network. For details, see "Setting Up Windows for a Network" in Chapter 1, "Windows 3.1 Installation."
- You can create a script to automate Windows installation for multiple systems. For details, see "Using Automated Setup" in Chapter 1, "Windows 3.1 Installation."
- For networks, the network connection is made automatically when you run Windows, and related menu items appear automatically to assist you with network activities related to file management.
- You can use long file names on network servers, if the driver for your network supports this feature.

This chapter contain information about these specific networks:

3Com networks	Invisible Software—NET/30 for Windows
Artisoft LANtastic	Microsoft LAN Manager Networks
Banyan VINES	Microsoft Network (MS-Net) and compatibles
DEC Pathworks	Novell NetWare
IBM OS/2 LAN Server	TCS 10Net
IBM PC LAN Program	Ungermann-Bass Net/One

If your network is not listed here, it may be based on or equivalent to another network. For example, many networks are based on Microsoft Network.
For tips about using PC-NFS with Windows 3.1, see the related article in Appendix D.

Running a Shared Copy of Windows

Flowchart 5.2 Cannot Run 386 Enhanced Mode on Network Workstation

Make sure the user's personal WINDOWS directory and the shared WINDOWS directory are on the user's path, and in that order. The Setup program will automatically take care of this if the user chooses to have Setup make modifications to the AUTOEXEC.BAT file.

For some networks, MS-DOS SHARE.EXE must not be loaded on a workstation to run a shared copy of Windows. If a user has problems running a shared copy, remove the command line that loads SHARE.EXE from the user's system.

You might need to mark applications stored on a network directory as read-only to prevent sharing violations when they are run by more than one user at a time. You can set the read-only attribute from within Windows File Manager or by using the MS-DOS **attrib** command.(Your network may also have an attribute program that you can use.) Most applications require read-only status before they can be shared.

If the user chooses to have Setup make modifications to AUTOEXEC.BAT, Windows Setup modifies AUTOEXEC.BAT to make sure the WINDOWS directory appears on the path ahead of any conflicting directories. However, because of the way some networks implement their startup procedures, a network can change this path when a user logs on. If this happens, modify the network logon procedure so that the paths are set correctly in Windows.

Also, verify user rights for the directories where any Windows .INI and .GRP files are located, and verify that any shared .GRP files are marked read-only. Ensure that the mapping for all network drives is correct and that all search drives are correctly designated.

Using a Diskless Workstation

Print Manager spools print jobs to the location of the MS-DOS TEMP environment variable. If a value for TEMP is not set, Print Manager defaults to the root directory of drive C. For diskless workstations, set the TEMP variable either to a RAM disk or to the location of a network directory. The user should have full and private access to this directory. The TEMP variable is set by placing a **set TEMP=** statement in the AUTOEXEC.BAT file or network logon script. For example, to spool to the network directory X:\USERTEMP, you would add this statement to AUTOEXEC.BAT:

```
set temp=x:\usertemp
```

Each user must have a personal TEMP directory.

Maintaining Windows with a Network

If the system administrator or a user changes the network configuration by running Setup after Windows has been installed, Setup might not be able to remove all support for the previous network selection. Check the network notes later in this chapter for details about .INI file changes made for your network.

Other items to consider in maintaining Windows on a network:

- Windows does not support the execute-only permissions provided by several networks. These permission settings work for applications run outside of Windows, but neither execute-only Windows applications nor non-Windows applications can be run from within Windows.
- To view a directory where the user does not have access to the parent directory, set the current working directory to a visible directory before starting Windows. For more information, see your network documentation. File Manager displays the current directory and any subdirectories, regardless of whether the parent directory can be displayed.
- If you connect to a network drive that supports long filenames, you will be able to see the long filenames in a directory window. Using File Manager, you can move and copy files with long filenames. However, you cannot perform other operations on them, such as opening or running. If you move or copy a file with a long filename to a directory that doesn't support long filenames, File Manager will propose a shorter name. Extended attributes are preserved when you move or copy files with long filenames.

Memory, Windows, and the Network

This section presents details about specific SYSTEM.INI entries and other elements of managing memory when using Windows 3.1 with a network.

Windows Setup and Pop-up Messages

Many networks have pop-up message facilities so you can see messages sent from other servers or workstations. In most cases, these pop-up programs will cause Windows Setup to fail if a message is received while Windows is being installed. Many networks provide utilities that allow them to work with Windows after installation, but these are generally not available during Setup. You should make sure any such pop-up service is disabled before installing Windows. Please consult specific entries regarding your network in the sections later in this chapter, or consult your network documentation for instructions.

SYSTEM.INI Settings for Networks

If you are having problems running Windows in 386 enhanced mode, you may need to change the way Windows and your network are set up to use memory. Several SYSTEM.INI entries affect how Windows interacts with a network. You can change the following entries to correct most problems you might have operating Windows with a network.

Section	Entries	Entries
[boot]	Network.drv=	
[standard]	Int28Filter=	NetHeapSize=
[386enh]	AllVMsExclusive=	TokenRingSearch=
	NetAsynchFallback=	InDOSPolling=
	ReflectDOSInt2A=	Network=
	EMMExclude=	UniqueDOSPSP=
	NetAsynchTimeout=	Int28Critical=
	TimerCriticalSection=	PSPIncrement=
	FileSysChange=	NetHeapSize=
	NetDMASize=	

For information about these entries and how to edit SYSTEM.INI, see Chapter 4, "The Windows Initialization Files."

Specifying SYSTEM.INI Settings After Using Setup

If you use Setup to change your network configuration after you have already installed Windows, unnecessary files might be left in your WINDOWS directory, and entries might be left in your WIN.INI or SYSTEM.INI files that could harm your system's performance. Setup leaves these files because it cannot be sure they are not still required by other components in your system.

After changing your configuration with Setup, you should check the network-specific notes about both your previous and new system settings to see if any previous changes are no longer required. For example:

- Some networks require the **TimerCriticalSection=** entry in the **[386enh]** section of SYSTEM.INI to be set to a value of **1** or greater.
- Some network adapter cards require the **ReservedHighArea=** or the **EMMExclude=** entry in the **[386enh]** section of SYSTEM.INI to be set to the memory range used by the card. These cards will probably also require an **x=** switch on the EMM386 command line in your CONFIG.SYS file. For more information about EMM386.EXE, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide*.
- Some network cards and some network transport stacks require special handling. Consult the sections below for the manufacturer of your network and for the manufacturer of your network adapter and transport stack.

Changing Memory Settings

The network may require a larger buffer for direct memory access (DMA) data transfers than Windows usually provides. Or your network may not be able to run unless Windows is prevented from using UMBs in the upper memory area.

To change the size of the direct memory access buffer:

- In the **[386enh]** section of SYSTEM.INI, add or modify the **NetDMASize=** entry to specify how much memory (in kilobytes) Windows should reserve for direct memory access.

  The default value is **0** for ISA machines (IBM PC/AT or compatible) and **32** for Micro Channel machines (IBM PS/2 or compatible). For example, if you want Windows to provide a 32K buffer for direct memory access, specify **NetDMASize=32**.

If an application does not run correctly (in either standard or 386 enhanced mode), you may have to increase the buffer for transferring data over the network.

To increase the size of the data transfer buffer:

- In the **[standard]** or **[386enh]** section of SYSTEM.INI (depending on the mode where you're having problems), modify the **NetHeapSize=** entry to specify the maximum buffer size (in kilobytes) required for data transfers.

  The default value is **8** in the **[standard]** section and **12** in the **[386enh]** section. All values are rounded up to the nearest 4K. For example, to make the buffer size 16K, specify **NetHeapSize=16**.

To prevent Windows from scanning a part of memory:

- In the **[386enh]** section of SYSTEM.INI, set a value for **EMMExclude=** or **ReservedHighArea=** to specify a range of upper memory addresses that Windows should not scan or use.

For details about this process, see "Optimizing Use of the UMBs" in Chapter 6, "Tips for Configuring Windows 3.1."

The **PSPIncrement=** and **UniqueDOSPSP=** entries in SYSTEM.INI can also affect how Windows uses memory.

Disabling File-Handle Caching

By default, Windows keeps a minimum of 14 handles open, which may limit the number of users that can run a shared copy of Windows from a single server. You can modify SYSTEM.INI to decrease the number of cached file handles. The minimum is 2. If you decrease the number of cached file handles, Windows may slow down, but more users should be able to access Windows from a single server.

To decrease the number of cached file handles, set the value of the entry **CachedFileHandles=** in the **[boot]** section of SYSTEM.INI. For your network, you may also be able to increase the number of file handles available on the server.

Running Non-Windows Applications

If you have problems running a network-specific application that is a non-Windows application, you may need to run the application in exclusive mode. You can use PIF Editor to do this. If the application creates or deletes files, it may not be running exclusively even if it has been set to run in exclusive mode. This is because Windows is alerted every time a file is manipulated. You can prevent this alert by setting **FileSysChange=no** in the **[386enh]** section of SYSTEM.INI.

If you have problems running a non-Windows application or running the MS-DOS Prompt, you may need to adjust the **TimerCriticalSection=** entry in the **[386enh]** section of SYSTEM.INI. The **TimerCriticalSection=** value prevents switching or multitasking for a specified time period (in milliseconds), ensuring that only one non-Windows application at a time receives timer interrupts. For example, to specify a 10,000 millisecond time-out, set **TimerCriticalSection=10000**.

Note Using the **TimerCriticalSection=** entry may slow down Windows.

Other Memory Considerations

If a user has problems running Windows or other applications with network software loaded in the high memory area (the HMA, or first 64K of extended memory), try disabling the network's HMA option. Consult your network documentation for more information.

Some memory managers, such as EMM386, 386MAX, and QEMM, can load networks and other device drivers in the upper memory area. If the network is loaded in the upper memory area and a user has problems running Windows, try loading the network in conventional memory.

Creating Custom User Environments



There are other options available for setting up a user's environment:

- If you have many workstations to set up, you may want to create a master PROGMAN.INI file with the settings you want and then have the file copied to the user's WINDOWS directory during Setup. For more information about copying files during Setup, see "Modifying .INF Files for Custom Installations" in Chapter 2, "The Windows Setup Information Files," and see "Other Windows Initialization Files" in Chapter 4, "The Windows Initialization Files." The PROGMAN.INI file that gets copied during Setup will overwrite any existing PROGMAN.INI file in the user's WINDOWS directory.
- You can specify Program Manager restrictions by modifying the PROGMAN.INI file. You can disable some of the commands in Program Manager, remove the File menu altogether, or specify other restrictions, depending on the functionality you want users to be able to access. For information, see "Configuring Program Manager for a Network Workgroup" in Chapter 6, "Tips for Configuring Windows 3.1."
- You can also set up shared Program Manager groups and change the name of the Startup group. For information, see "Setting Up Shared Program Manager Groups" in Chapter 6, "Tips for Configuring Windows 3.1."
- You can use environment variables in PIFs so that you can create a single PIF to be used on various workstation configurations. For more information about PIFs, see Chapter 7, "Non-Windows Applications," in the *Windows User's Guide.*
- You can set up password protection for a workstation by using a screen saver with a password. For more information, see "Using a Screen Saver" in Chapter 5, "Control Panel," in the *Windows User's Guide.*

3Com Networks

Windows Setup recognizes the following 3Com networks:

- The 3Com 3+Share network is based on Microsoft Network (MS-Net) and will be handled as such by Windows. See the "Microsoft Network and Compatibles" section later in this chapter.
- The 3Com 3+Open LAN Manager network is based on Microsoft LAN Manager and will be handled as such by Windows. See the "Microsoft LAN Manager Networks" section later in this chapter.

When you install Windows for 3Com 3+Share or 3Com 3+Open LAN Manager, Setup adds these entries to the **[386enh]** section of SYSTEM.INI:

```
TimerCriticalSection=10000
UniqueDOSPSP=true
PSPIncrement=5
```

The following items should also be noted in relation to 3Com networks:

- Some copies of 3Com 3+Open LAN Manager version 1.1 will require a maintenance upgrade to work with Windows. If you have problems, contact your network customer support organization.
- When you install Windows, Setup modifies your AUTOEXEC.BAT file to make sure the WINDOWS directory appears on your path ahead of any conflicting directories. However, because of the way that 3+Share implements its startup procedures, the network might change this path when you log on. If so, the system administrator must modify the network logon procedure so that the paths are set correctly to run Windows.
- On a 3Com 3+Open network, the user's name displayed by Print Manager might not match the user's actual logon name.
- Some 3Com 3Station diskless workstations come with an All ChargeCard. Make sure your machine is not configured to load the network into the upper memory area, because you will not be able to install Windows or work reliably with Windows standard mode if your system is configured in this manner.
- If you use the 3C505 Network Adapter Card, you must disable the 3Com Link Plus to run Windows in 386 enhanced mode. You can disable it by removing the following lines from your CONFIG.SYS file:

  ```
  device=psh.sys <arguments>
  device=pth.sys
  ```

  Also, remove the following line from your AUTOEXEC.BAT (or other batch file you use to start your network):

  ```
  LDR
  ```

- Windows does not support the 3Com network aliasing scheme (three-part names).

Artisoft LANtastic

LANtastic versions below 3.0 are not compatible with Microsoft Windows.

LANtastic Versions 3.x

If you are running LANtastic versions 3.x you cannot run Windows 386 enhanced mode on a nondedicated server.

If you install Windows for LANtastic 3.x, Setup adds the following entries in the **[386enh]** section of SYSTEM.INI:

```
EMMExclude=D800-DFFF
INDOSPolling=TRUE
NetHeapSize=76
```

To print from Windows across LANtastic 3.x, you must either turn off the Fast Printing Direct To Port option in Control Panel or use the LPT*x*.DOS printing method.

To turn off the Fast Printing Direct To Port option in Control Panel:

1. Choose Printers option from Control Panel, and then choose the Connect button.
2. In the Connect dialog box, clear the Fast Printing Direct To Port check box.

When you turn off the Fast Printing Direct To Port option, it affects all ports.

To use the LPTx.DOS printing method:

1. If you are using LPT1 or LPT2 to print, you can use the **LPT1.DOS=** or **LPT2.DOS=** entries already in your WIN.INI file. If you upgraded from Windows 3.0, the WIN.INI entry appears as "LPT*x*.OS2."

   If you are using LPT3 to print, add **LPT3.DOS=** to the **[ports]** section of the WIN.INI file.
2. Quit Windows and redirect the LPT*x*.DOS port by using the LANtastic NET USE command.
3. Restart Windows.
4. Choose the Printers icon in Control Panel, and change your printer port to LPT*x*.DOS.

If you have a network connection made on LPT1, you will not be able to see it in Print Manager or Control Panel.

LANtastic Versions 4.x

If you install Windows for LANtastic 4.x, you must have the LANTASTI.386 file somewhere on your path. This file should be in your LANtastic directory.

If you install Windows for LANtastic 4.x, Setup adds the following entries in the **[386enh]** section of SYSTEM.INI:

```
EMMExclude=D800-DFFF
INDOSPolling=TRUE
NetAsynchFallback=TRUE
NetAsynchTimeout=50
NetHeapSize=64
PerVMFiles=0
```

The **EMMExclude=** entry is required for the default configuration of the Artisoft LANtastic Enhanced 2mbps network adapter. If you have changed the memory address setting on your adapter, you will need to modify this entry in SYSTEM.INI to reflect the appropriate address range. If you are not using this network adapter, you can remove the **EMMExclude=** entry.

Some commands, such as XCOPY, require that the **NetHeapSize=** entry in the **[standard]** and **[386enh]** sections of SYSTEM.INI be set higher than the default value. Increasing the **NetHeapSize=** value leaves less memory available for non-Windows applications, so only increase this value if necessary.

Banyan VINES

Setup will correctly handle the following Banyan VINES versions:

- Banyan VINES 4.0 is compatible with Microsoft Network (MS-Net), and will be handled as such by Windows.
- Banyan VINES 4.1 and above come with additional Windows components, which will be installed when you set up Windows for Banyan VINES 4.1.

Banyan VINES 4.0x

If you are running VINES 4.0 and want to run non-Windows applications that use NETBIOS, you must install VINES 4.00(5) or later. Also, you will need to start NETBIOS before starting Windows.

To use Windows printing functions over VINES 4.0, you must have VINES 4.00(5) or higher and must have the NETBIOS support loaded. If you cannot print to a network printer, the network administrator should verify that you are loading NETBIOS and that you are running the proper version of VINES.

If you are running VINES version 4.0x, and you have a problem with intermittent time-outs while printing, you can either update to VINES 4.1x, or use the VINES SETPRINT utility to remove the time-out on your printer port by specifying the following:

setprint lpt*n servicename* **/d:infinite**

Lpt*n* is the printer port and *servicename* is the name of the network print queue.

Banyan VINES 4.1x

When you install for Banyan VINES 4.1, Setup adds the following entries to the **[386enh]** section of SYSTEM.INI:

```
Network=*dosnet,*vnetbios, vvinesd.386
TimerCriticalSection=5000
UniqueDOSPSP=TRUE
PSPIncrement=5
```

If you are running Banyan VINES 4.10 or 4.11, you do not need to load NETBIOS unless you are running NETBIOS-specific applications. If you are running version 4.11, you can load NETBIOS either before starting Windows or after starting Windows. For more information on running memory-resident applications after starting Windows, see Chapter 7, "Non-Windows Applications," in the *Windows User's Guide.*

If you are running version 4.10 and you want to run NETBIOS-specific applications, you must start NETBIOS before starting Windows. However, currently you can only run one NETBIOS application at a time.

DEC Pathworks

Windows 3.1 supports DEC Pathworks versions 4.0 and above.

When you set up Windows for a DEC Pathworks network, the following entries are added to the **[386enh]** section of SYSTEM.INI:

```
Network=*dosnet, decnet.386, decnb.386
TimerCriticalSection=10000
```

If you configured your remote-boot workstation to run Windows 3.0, you will be loading a program called RIPLMEM.EXE in your AUTOEXEC.BAT. This program is no longer required for running Windows 3.1, so Setup will remove it from your AUTOEXEC.BAT file when you set up Windows. If this program is not being loaded from your AUTOEXEC.BAT file, you or the system administrator should remove it manually.

DEC Pathworks supports additional APIs that are not supported by the files installed by Windows 3.1.

DEC Pathworks 4.1

The following guidelines will help you when you use Windows 3.1 with DEC Pathworks 4.1:

- To access DEC Pathworks network services from Microsoft Windows, make sure that you have run the WIN3SETU utility and responded Yes to the prompt: "Do you wish to access PATHWORKS network services under Windows 3.0?"

  If your PC has access to a DEC Pathworks network driver, a Network icon is displayed in the Windows Control Panel. If you click this icon, a message box appears, "Windows 3.0 Network Driver for PATHWORKS," followed by a version number. If your PC does not have access to a DEC Pathworks network driver, no Network icon is displayed in the Control Panel, and you must run the WIN3SETU utility.

- While you run Windows, make sure no interrupt (IRQ) conflicts occur. Interrupt conflicts that affect your PC's mouse or serial port may not be evident until you use Windows.

- When you run Netsetup in Intermediate or Advanced mode and specify the Basic Redirector, do not load the Basic Redirector into XMS memory.

- Do the following before using the VT320 terminal emulator application:

  Type **set netwin=1** at the command prompt to set the MS-DOS environment variable NETWIN to 1.

  Run Netsetup to load the LAT, CTERM, or TELNET network component into your PC's memory.

  If you load LAT, use the LATCP DEFINE SCB command to define enough session control blocks (SCBs) to accommodate the number of terminal emulator sessions you want to run simultaneously. For example, if you want to run four terminal emulator sessions simultaneously, type **latcp define scb 4 4** at the command prompt.

- Your PC must use the latest SCSI adapter card (Adaptec 1540B) if your PC has a SCSI disk drive, your PC is a DECstation, and you load DEC Pathworks network components into the UMBs.

  If all these conditions exist, make sure that your SCSI adapter card is an Adaptec 1540B (not 1540 or 1540A) and that the SCSIHA.SYS driver file is installed on your PC from the floppy diskettes that come with your DECstation. If your SCSI adapter card is an Adaptec 1540 or 1540A, replace it with the latest card.

- If your PC uses a third-party super VGA driver and the display does not work properly, install the standard VGA driver that Microsoft Windows provides. If the display works properly now, there is a problem with the third-party driver.

- Change the **files=** setting in your PC's CONFIG.SYS file to **40**. This setting allows Windows applications to have up to 40 MS-DOS files open simultaneously. You may need to change this setting to a higher value, depending on the applications you run.

- A PC that uses the IBM XGA or 8514 Windows drivers and that loads the DEC Pathworks broadcast receiver (RCV) will hang when it receives a broadcast message. In this case, replace the XGA or 8514 drivers with the IBM supplemental drivers kit that includes the standard VGA.DRV file.

Questions and Answers about DEC Pathworks

The following provides some answers for common questions about the DEC Pathworks device drivers included with Windows 3.1.

Q: Do the network device drivers that come with DEC Pathworks 4.1 work with Windows 3.1, or should I use the drivers included in Windows 3.1?

A: Although the network drivers in DEC Pathworks 4.1 work with Windows 3.1, it is best to use those included in Windows 3.1. The DEC Pathworks network driver PCSA.DRV that is included with Windows 3.1 provides specific support for Windows 3.1.

Q: I have already installed DEC Pathworks 4.1, and I am using the DEC Pathworks Windows network drivers that are included in that kit. What must I do to use the DEC Pathworks network drivers that are included in Windows 3.1?

A: Run Windows Setup. In the Network field of the Change System Settings dialog box, choose the DEC Pathworks (version 4.1 or higher) option. Windows 3.1 will install the DEC Pathworks network drivers included with Windows 3.1 and modify the appropriate entries in SYSTEM.INI.

Q: What is the difference between the DEC Pathworks network device drivers included with DEC Pathworks 4.1 and those included with Windows 3.1?

A: Windows 3.1 includes a new version of the DEC Pathworks network driver, called PCSA.DRV, that allows users to disable persistent connections, which is a new feature of Windows 3.1.

The virtual device drivers VDNET.386 and VNETBIOS.386 in with DEC Pathworks 4.1 are called DECNET.386 and DECNB.386, respectively, in Windows 3.1. VDNET.386 was changed to DECNET.386 to make it clear that the driver is a DECnet virtual device driver; VNETBIOS.386 was changed to DECNB.386 because Microsoft includes a driver called VNETBIOS.386 (which does not support Digital's enhanced NETBIOS functions) in Windows 3.1.

DECNET.386 and DECNB.386 improve network performance by changing synchronous socket calls to asynchronous socket calls. VDNET.386 and VNETBIOS.386 can also do this. But for VDNET.386 and VNETBIOS.386, you must add these entries to the **[386enh]** section of SYSTEM.INI:

- **DNETNoBlock=true** (for DECnet socket calls)
- **NetNoBlock=true** (for NETBIOS calls)

Q: If my client still runs DEC Pathworks 4.0, can I install the DEC Pathworks network drivers included with Microsoft Windows 3.1?

A: Yes. Run Windows Setup. In the Network field of the Change System Settings dialog box, choose the DEC Pathworks version 4.0 option. Windows 3.1 will install the DEC Pathworks network drivers included with Windows 3.1 and modify the appropriate entries in SYSTEM.INI.

Note Windows Setup may display a message that says that a newer version of DEC Pathworks is available. Because DEC Pathworks 4.0 works with Windows 3.1, you can ignore this message.

Q: I see that Windows printer drivers for the following Digital printers are included in with DEC Pathworks 4.1:

- DEClaser 2100 (LN05)
- DEClaser 2200 (LN06)
- LA324
- LA70
- LJ250 and LJ252

I see also (previously in this section) that eight Windows printer drivers for Digital PostScript printers are included with Windows 3.1. However, I do not see Windows printer drivers available for the following Digital printers:

- LA75
- LN03
- LN03 PLUS
- DEClaser 1100 (LN07)
- DEClaser 3200 (LN08)

Do Windows printer drivers for these printers exist and, if so, how can I get them?

A: Yes, they do exist. You can get them through DEC directly or through Digital's DECPCI CompuServe Forum (enter **GO DECPCI** at the prompt).

Note The DEC Pathworks CompuServe Forum is called DECPCI (as in DEC PC Integration).

IBM OS/2 LAN Server

IBM OS/2 LAN Server, which is also known as the IBM DOS LAN Requester, is based on Microsoft LAN Manager. Setup supports the following versions:

- LAN Server versions 1.x are equivalent to Microsoft LAN Manager 1.x. See the "Microsoft LAN Manager Networks" section later in this chapter.

  However, IBM distributed CSD 5015 for LAN Server 1.3, which brings it up to the equivalent of LAN Manager 2.0.

- LAN Server version 2.0 is also treated as Microsoft LAN Manager 2.0 but also includes some LAN Server–specific components.

Running Windows with LAN Server version 2.x requires the files LANSRV.DRV, WINDLR.DLL, NETAPI.DLL, and PMSPL.DLL. All four files should be in your LAN Server directory.

Windows includes updated versions of the NETAPI.DLL and PMSPL.DLL files shipped with LAN Server. These files are installed automatically during Setup.

LAN Server versions below 2.0 do not allow you to use the IBM OS/2 LAN Server aliasing scheme from inside Windows applications. Also, LAN Server versions below 2.0 do not allow you to log on through Windows Control Panel. Doing so might cause your machine to hang. Always log on at the command prompt, either before starting Windows or while Windows is running.

IBM PC LAN Program

When installing for IBM PC LAN Program, Setup adds the following entry to the **[386enh]** section of SYSTEM.INI:

```
INDOSpolling=TRUE
```

The IBM PC LAN Program is based on Microsoft Network (MS-Net) and will be handled as such by Windows.

The IBM PCLP Extended Services refers to network drives and printers by two different names. Usually, when connecting to a network, you select connections from a list containing descriptive names, but internally connections are made using hidden names such as \\SERVER\IBMXS001. With PCLP Extended Services, you must type the hidden names in Control Panel or File Manager to make new network connections. The network administrator can supply the proper names to use.

Some versions of PCLP lock up your system if you receive a broadcast message while running Windows. You can disable these messages with the NET PAUSE MESSENGER command. Or you can start up your network with NET START RDR instead of NET START RCV or NET START MSG commands.

When you connect to a network printer in Control Panel, that connection will not be visible to non-Windows applications. It will, however, still be usable by Windows and Windows applications.

Invisible Software—NET/30 For Windows

To use NET/30 For Windows, choose the Custom Setup option when you run Setup. For your network configuration, choose Other (which requires a disk provided by manufacturer). You are prompted for a path to the directory containing your NET/30 files. Your SYSTEM.INI and WIN.INI files will automatically be updated.

If you are installing Windows on a nondedicated file server, and you are sharing a serial printer, you must add an entry to SYSTEM.INI. If you are sharing COM1, add **COM1AutoAssign=0** to the **[386enh]** section. If you are sharing COM2, add **COM2AutoAssign=0** to the **[386enh]** section.

Microsoft LAN Manager Networks

Many networks, such as IBM LAN Server, DEC Pathworks, and 3Com 3+Open are based upon Microsoft LAN Manager. Consult your network documentation, network distributor, or system administrator to learn whether your network falls into this category and, if so, on which version of LAN Manager it is based.

Network software based on MS-DOS LAN Manager Basic must not be loaded in the high memory area (HMA). The network administrator can determine if these conditions apply to a specific workstation.

If you are using the Ungermann-Bass XNS transport stack and Windows in 386 enhanced mode, you must make sure the transport stack is not using expanded memory. If your system is configured to provide expanded memory with a memory manager such as EMM386.EXE, you can disable the expanded memory option. Or, if you want to continue using expanded memory, you can disable its use by the Ungermann-Bass transport stack with the following steps.

To disable expanded memory for the Ungermann-Bass transport stack:

1. Open your PROTOCOL.INI file by using a text editor such as Notepad. Your PROTOCOL.INI file is in your LAN Manager directory.
2. In the section **[XNS_XIF]**, add a **noemm** entry, or, if the entry already exists, make sure that it is not commented out with a semicolon at the beginning of the line.
3. Save your PROTOCOL.INI file, then quit and restart Windows.

Your system might slow down considerably when you run Windows in 386 enhanced mode if your LAN Manager components are configured to use expanded memory. If this is a problem, turn off the use of expanded memory by modifying your LANMAN.INI and PROTOCOL.INI files. The exact procedure will vary depending on what components you are running. See the documentation for LAN Manager for further details.

To disable expanded memory for LAN Manager components:

1. Open your LANMAN.INI file by using a text editor such as Notepad. Your LANMAN.INI file is in your LAN Manager directory.
2. Add **lim=no** in the **[workstation]** section.
3. Save your LANMAN.INI file, and restart your computer.

Or you can stop your workstation and restart it again using the NET STOP WORKSTATION and NET START WORKSTATION commands. Also, you may want to configure other network components (such as protocol stacks) to avoid using expanded memory. Consult your network's documentation for the specific component for instructions on how to do this.

Some early copies of LAN Manager 1.x will require a maintenance upgrade to run with Windows. If you experience problems, contact your network vendor for an upgrade.

If you upgrade from an earlier version of LAN Manager Basic to version 2.1 Basic using the Over-the-Network Upgrade process, your SYSTEM.INI and WIN.INI files might be left with incorrect information. This will only happen if your Upgrade procedure has been configured to update Windows installations. This will cause you to receive a warning that "WinPopup" cannot be found, and all network functions in File Manager, Print Manager, and Control Panel will be unavailable.

To correct this problem, change the value of the **load=** entry in your WIN.INI file by removing the value **WinPopup** or **WinPopup.exe**. Then run Windows Setup from the command prompt while in your WINDOWS directory. Choose the Microsoft LAN Manager (version 2.1 Basic) option to specify the network.

LAN Manager Enhanced lets you use pop-up services to see incoming broadcast messages. These pop-up services can disrupt the Windows display, so you must either disable them or run the LAN Manager WinPopup utility so they can coexist with Windows.

The WinPopup utility should be in your LAN Manager NETPROG directory, included on your path. To automatically start the utility when you start Windows, you must include the entry **load=winpopup.exe** in the **[Windows]** section of WIN.INI. If you install Windows for LAN Manager 2.0 or 2.1 Enhanced, this procedure is taken care of automatically. If you are running LAN Manager 1.x Enhanced, you must insert this entry yourself.

Note Windows 3.1 includes updated versions of several LAN Manager files, including the WinPopup utility.

Alternatively, if you want to disable the pop-up services, use the following procedure.

To disable pop-up services:

1. In the LANMAN.INI file in your LAN Manager root directory, change the **wrkservices=** entry by removing the **netpopup** or **minipop** value.

   If you remove the **netpopup** and **minipop** values but leave the **messenger** value, messages will still be received but will be recorded in the MESSAGES.LOG file in your LAN Manager directory rather than being displayed on the screen.

   If you do not wish to log your messages, you can also remove the **messenger** value from this same line.

2. Save the file, quit Windows, and restart your computer.

For Windows to support LAN Manager 2.0 Enhanced, you must have the files NETAPI.DLL and PMSPL.DLL. These files are in your LAN Manager NETPROG directory, which should be on your path. Windows also includes updated versions of these files that will be installed automatically on some older versions of LAN Manager.

For Windows to support LAN Manager 2.1 Enhanced, you must have the files NETAPI.DLL, PMSPL.DLL, LANMAN21.DRV, WINPOPUP.EXE, and LM21_W31.HLP. These files are usually in your LAN Manager NETPROG directory, which is on your path. If you are missing any of these files, you may receive warning messages when starting Windows or be unable to use Windows network functions.

Microsoft Network and Compatibles

If a network is present and detected, but cannot be identified during Setup, Microsoft Network (MS-Net) is the default network that will be set up.

The Microsoft Network driver does not support any vendor's name aliasing scheme.

The following notes apply to the general category of networks that support the MS-DOS network interface and NETBIOS protocols:

- Because of the way some of these networks handle print queues, Print Manager cannot distinguish multiple print queues on the same server. The lists of print jobs in these queues might be displayed improperly. Also, Print Manager might not be able to display the job title when displaying queue information.
- When you set up Windows for Microsoft Network, the following entries are added to the **[386enh]** section of SYSTEM.INI:

```
TimerCriticalSection=10000
UniqueDOSPSP=TRUE
PSPIncrement=5
```

For some networks, you may be able to remove one or more of these entries to improve performance.

Novell NetWare

When you are running Novell NetWare, Windows Setup will copy the following files to your WINDOWS directory:

NETX.COM	IPXODI.COM	TBMI2.COM
IPX.OBJ	LSL.COM	

If you are running a shared copy of Windows from a network, these files are in the shared network directory.

Before running Windows, you should take the following steps.

To get ready to run Windows with Novell NetWare:

1. Replace your current NetWare shell with the NETX.COM provided.

   Windows 3.1 requires that the NetWare shell components on your workstation (such as NET5.COM) be version 3.01e or higher. You can use these shell components even if your servers are running older versions of NetWare (as far back as NetWare 2.10).

   Your current shell may be called by any of the following names:

NET3.COM	XMSNET3.COM	EMSNET3.COM
NET4.COM	XMSNET4.COM	EMSNET4.COM
NET5.COM	XMSNET5.COM	EMSNET5.COM
NETX.COM	XMSNETX.COM	EMSNETX.COM

   NETX.COM will run on MS-DOS version 3.0 or higher.

2. If you are running Novell's IPXODI.COM and LSL.COM, you will need to upgrade these to the versions provided.

3. If you will be running Windows in standard mode, you should load the TBMI2.COM memory-resident utility when running Windows. Novell recommends that you create a batch file that will load this utility, then start Windows, then unload the utility when Windows exits. The batch file could be named RUNWIN.BAT and look like this:

   ```
   tbmi2
   win
   tbmi2 /u
   ```

   TBMI2 allows you to safely run applications that use the NetWare IPX and SPX functions under standard mode Windows or the MS-DOS 5.0 task switcher.

4. If you are running Novell's IPX.COM, the system administrator must build a new version using the IPX.OBJ provided.

The following notes also apply to Novell NetWare:

- If you have a Western Digital Ethercard Plus and have problems accessing drives using the MAP ROOT command, contact Western Digital for an updated driver.
- Do not try to log on, log out, attach, or detach a Novell server from the MS-DOS Prompt within Windows. You must perform these operations before starting Windows.

- By default, NetWare gives you access to only 40 files at a time. When you are running applications with Windows, you can exceed this limit rather quickly. If so, you might see unexpected error messages. To increase the file access limit, add the following line to your SHELL.CFG file:

  ```
  file handles = 60
  ```

  You should also make sure the following line is in your CONFIG.SYS file:

  ```
  files = 60
  ```

Flowchart 4.4 Garbled Output on Novell Network

- If you get unusable output when printing to a NetWare server (such as incorrect fonts, garbled text, invalid page breaks, or blank lines), you might need to edit your print job configuration. Use the NetWare PRINTCON utility to set the Auto Endcap and Enable Time-out options to No.
- If you are running Windows in 386 enhanced mode, paging to a drive on a NetWare 286 server might cause Windows to take up to a minute to start. For details about how to control the location of your paging file, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide*.
- A NetWare file server does not include the directory entries dot (.) and double dot (..) as MS-DOS does. However, the 3.01 or higher NetWare shells can emulate these entries when applications attempt to list the files in a directory. If you have problems listing files or deleting directories, turning on the Show Dots feature will help. Add the following line to your SHELL.CFG file:

  ```
  show dots = on
  ```

  Turning on Show Dots will cause problems with earlier versions of some 286-based NetWare utilities, such as BINDFIX.EXE and MAKEUSER.EXE. Make sure you upgrade these utilities if you upgrade your NetWare shell.
- Redirected drives might look different in Windows. For example, a device mapped as *FileServerName/Volume:Directory* will appear in Windows simply as *FileServerName/Volume:*\, showing the root of the file server's shared volume, rather than the subdirectory you expected. If so, you can correct this discrepancy with the MAP ROOT entry, as explained in the following paragraph.
- In some cases Windows applications may change the default directory on a drive. This might upset your configuration if, for example, your path includes the current directory on a network drive (that is, your path includes a drive letter without a specific directory, such as Z: instead of Z:SYSTEM).

  You can prevent this problem by using MAP.EXE and LOGIN.EXE versions 3.0 and later to set up fake roots, or choose the MAP ROOT function when mapping a connection in File Manager. This feature simulates the MS-DOS **subst** command, which sets the root of a given drive to a directory designated by the user instead of setting it to the true root of the volume.

For example, you might usually map drive F to the HOME\TERRI directory on the SERVER\SYS volume, and include drive F in your path. To do this, include the following command in your AUTOEXEC.BAT file or type the command before starting Windows:

```
map f:=count\sys:home\terri
```

The default directory on drive F will then be HOME\TERRI. To prevent Windows from changing this, replace the command above with this one:

```
map root f:=count\sys:home\terri
```

This command would make the directory HOME\TERRI appear to be the root of drive F.

- If you are running Windows in 386 enhanced mode, you can adjust the way Windows handles your network drive mappings by using options in the Network section of Control Panel. Usually, when you exit Windows, all of your drive mappings are restored to the way they were before you started Windows, and all changes you made inside Windows are lost. If you clear the Restore Drives option, the mappings you made inside Windows will remain when you exit Windows.
- Usually, each virtual machine you start from Windows will have its own set of drive mappings. Changes you make in one virtual machine will not affect another. If you set the NW Share Handles option, drive mappings will instead be global, and changes made in one virtual machine will affect all other applications.

  See the help available in the Network dialog box in Control Panel for more information on these options.
- When installing for Novell NetWare, Windows Setup should add the entry **load=nwpopup.exe** to the **[Windows]** section of your WIN.INI file. This entry automatically starts the NetWare pop-up utility, which displays your incoming broadcast messages. If this entry has not already been added, set **load=NWPopup** in your WIN.INI file.
- If you are using NetWare Lite, during Setup choose either "No Network Installed" or "Microsoft 100% LAN Compatible" in the System Settings dialog box. If you choose "Microsoft 100% LAN Compatible Network," all NetWare Lite connections will appear with a full path in the File Manager and Control Panel. You cannot, however, assign new network drives through File Manager Control Panel.

  For either network choice, you should make all drive and port assignments before starting Windows. These connections will then be accessible through File Manager, Print Manager, Control Panel, and applications.

If you install Windows on a NetWare Lite Network server, then you should set these parameters:

- Increase the number of client tasks in the NetWare Lite server configuration by 20 for each client and increase the **files=** setting in CONFIG.SYS by 20 for each client that will be connected to the server.
- Increase the **files=** setting in the CONFIG.SYS on the client machine.
- You may need to increase the MS-DOS SHARE.EXE parameters if you are trying to open many files (for example, when using database applications).
- Set the read-only attribute for all .EXE, .COM, and .OVL files in the WINDOWS directory and SYSTEM subdirectory.
- Set **UniqueDOSPSP= true** in the **[386enh]** section of SYSTEM.INI.
- Do not use INT 2, INT 9, or above and do not use a base I/O port address below 300 for your LAN card (such as ARCnet or Token Ring) unless you use VPICDA.386, which you can obtain from either Microsoft or Novell.

TCS 10Net

Tiara Computer Systems 10Net (formerly known as DCA 10Net) versions 4.1 and above are compatible with Windows 3.1. Windows Setup treats these as Microsoft Network–compatible.

If you purchase 10Net for Windows, it will install additional Windows components that will enhance the functionality available in Windows File Manager, Control Panel, and Print Manager.

The following notes apply for TCS 10Net:

- You cannot use File Manager to format floppy disks when running versions of 10Net earlier than 5.0.
- When installing for TCS 10Net, Windows Setup adds the entry **TimerCriticalSection= 10000** to the **[386enh]** section of SYSTEM.INI. The **TimerCriticalSection=** value may be set lower on some versions of 10Net.
- When installing for TCS 10Net version 4.1x for use with the DCA 1 MB card, Windows Setup adds the following entries to the **[386enh]** section of SYSTEM.INI:

```
INDOSPolling=TRUE
PSPIncrement=5
TimerCriticalSection=10000
UniqueDOSPSP=TRUE
```

Ungermann-Bass Net/One

Make sure you choose the proper network option:

- Ungermann-Bass Net/One is based on Microsoft Network (MS-Net). Choose the Microsoft Network or 100% Compatible option when running Windows Setup.
- Ungermann-Bass Net/One LAN Manager is based on Microsoft LAN Manager. Choose the Microsoft LAN Manager option when running Windows Setup. Setup should correctly identify your network along with the proper version number.

If you are using an Ungermann-Bass network card with any network, including networks not produced by Ungermann-Bass, your system may crash randomly when running Windows in 386 enhanced mode unless you set the value **TimerCriticalSection=10000** in the **[386enh]** section of SYSTEM.INI.

Troubleshooting



Part

6 Troubleshooting Windows 3.1

Chapter 13 **Troubleshooting Windows 3.1** 395

About Troubleshooting 396
Troubleshooting Setup 399
Troubleshooting Windows Configuration 403
Troubleshooting Windows Running in Standard Mode 418
Troubleshooting Windows Running in 386 Enhanced Mode 420
Troubleshooting for Non-Windows Applications 423
Troubleshooting Fonts 425
Troubleshooting Printing 426
Troubleshooting Networks 430
Troubleshooting GP Faults 431
Troubleshooting Multimedia 434

Chapter 14 **The Windows Resource Kit Disk** 439

The Windows Resource Kit Utilities 440
Other Windows Resource Kit Files 443

Chapter

13 Troubleshooting Windows 3.1

This chapter provides information about troubleshooting Microsoft Windows for both general users and experts. If you have trouble installing Windows, or if Windows doesn't run as well as you expected, this chapter will help you find out why and show you how to isolate and solve common problems.

Related Information

- *Windows User's Guide:* Chapter 15, "Maintaining Windows with Setup"

  See also Chapter 4, "Troubleshooting," in the *Getting Started* booklet
- *Windows Resource Kit:* "The Troubleshooting Flowcharts for Windows 3.1" in "Welcome"

Contents of this chapter

About Troubleshooting 396
Getting Started with Troubleshooting 396
Creating a "Clean Boot" for Troubleshooting 398
Troubleshooting Setup 399
Troubleshooting TSRs During Setup 400
Troubleshooting MS-DOS Mode Setup 401
Troubleshooting Windows Mode Setup 402
Troubleshooting Windows Configuration 403
Troubleshooting the Desktop Configuration 403
Troubleshooting TSR Compatibility Problems 404
Troubleshooting Display Problems 408
Troubleshooting MS-DOS Configuration Problems 413
Troubleshooting EMS Memory Problems 415
Troubleshooting Windows Running in Standard Mode 418
Troubleshooting Windows Running in 386 Enhanced Mode 420
Troubleshooting for Non-Windows Applications 423
Troubleshooting Fonts 425
Troubleshooting Printing 426
A Strategy for Troubleshooting Printing 426
Tips for Specific Printing Problems 428
Troubleshooting Networks 430
Troubleshooting GP Faults 431
Tips for Troubleshooting GP Faults 432
Dr. Watson and GP Faults 434
Troubleshooting Multimedia 434

About Troubleshooting

As a general orientation to troubleshooting, begin with these basic problem-solving actions:

- Try to reproduce the problem using the minimum system configuration described in "Creating a 'Clean Boot' for Troubleshooting." later in this chapter.
- Try to reproduce the problem in both standard mode and 386 enhanced mode, to see if the problem is mode-specific.
- Try to reproduce the problem while running another application and, if possible, while running the applications provided with Windows.
- Check the README.WRI, PRINTERS.WRI, and NETWORKS.WRI files in the WINDOWS directory, looking especially for notes related to your specific hardware or for TSRs, device drivers, and other third-party software that you are running with Windows.

Windows is a stable environment that was tested on thousands of computers before it was shipped. However, your system might have a display adapter or other hardware that was not in existence during the latest Windows development cycle. This means you might be installing or running Windows on an untested configuration. This chapter, and the README files in your WINDOWS directory, can help you find solutions.

Getting Started with Troubleshooting

You can use this decision tree to begin troubleshooting for Windows.

Decision Tree for Troubleshooting Windows 3.1

- During Setup, you get an error message or your system crashes.
  → See **"Troubleshooting Setup"** and **Flowchart 1.2**
- Your display is garbled or wrong or you have other display problems.
  → See **"Troubleshooting Display Problems"** and **Flowchart 2.3**

Decision Tree for Troubleshooting Windows 3.1 *(continued)*

- You have memory-related configuration problems.
  → See **"Troubleshooting MS-DOS Configuration Problems"** and **Flowchart 1.2**
- You have problems using the upper memory area.
  → See **"Troubleshooting EMS Memory Problems"** and **Flowchart Series 2**
- You cannot run Windows in standard mode or you get not-enough-memory messages.
  → See **"Troubleshooting Windows in Standard Mode"** and **Flowcharts 1.1 and 5.9**
- You cannot run Windows in 386 enhanced mode, or the system crashes, or you have hard disk problems.
  → See **"Troubleshooting Windows in 386 Enhanced Mode"** and **Flowchart Series 2** and **Series 5**
- You get out-of-memory messages or the application won't start when you try to run a non-Windows application under Windows.
  → See **"Troubleshooting Non-Windows Applications"** and **Flowcharts 5.4, 5.5., and 5.8**
- You experience problems when using TrueType fonts.
  → See **"Troubleshooting Fonts"** and **Flowchart Series 3**
- You cannot print from Windows or have problems with printing.
  → See **"Troubleshooting Printing"** and **Flowchart Series 4**
- You have problems installing or running Windows on a network, or you have problems printing over a network.
  → See **"Troubleshooting Networks"** and **Flowcharts 1.3, 4.2,** and **4.4**
- You receive GP-fault messages while running Windows applications.
  → See **"Troubleshooting GP Faults"** and **Flowcharts 5.3, 5.4,** and **5.5**
- You have problems with multimedia accessories.
  → See **"Troubleshooting Multimedia"** and **Flowchart Series 6**

Creating a "Clean Boot" for Troubleshooting

Flowchart 1.5 Clean Boot Disk

Use a "clean boot" floppy disk to boot your system for troubleshooting problems with incompatible TSRs and drivers and problems with conflicting uses of the upper memory area. Remember that you must reboot your system after any change in AUTOEXEC.BAT or CONFIG.SYS for the change to take effect.

To create a "clean boot" disk for troubleshooting:

1. Back up your hard disk before beginning.
2. Minimize the configuration in your CONFIG.SYS and AUTOEXEC.BAT, including only the drivers and programs necessary for the computer to boot and run Windows, and removing all other TSRs or drivers.

   For testing purposes, reduce the CONFIG.SYS file to:

```
files=40
buffers=20
device=c:\windows\himem.sys
device=c:\windows\ega.sys                    ; if you have an EGA monitor
stacks=9,256                                 ; if MS-DOS 3.3 or later
shell=c:\dos\command.com c:\dos /p /e:256    ; make sure command.com
[plus any network command lines]             ; is in the C:\DOS directory
```

   Reduce the AUTOEXEC.BAT file to:

```
path=c:\;c:\dos;c:\windows
set temp=c:\windows\temp                     ; TEMP must point to a valid directory
prompt $p$g
[plus any network command lines]
```

3. Make a boot disk. To do this, place a disk in drive A and type the following command at the command prompt:

```
format a:
```

4. Copy the configuration files to this disk by typing the following commands at the command prompt:

```
copy \config.sys a:
copy \autoexec.bat a:
```

5. Save your current initialization files on the floppy disk by typing the following commands at the command prompt:

```
copy \windows\win.ini a:
copy \windows\system.ini a:
copy \windows\progman.ini a:
```

If you want to establish a system configuration that ensures optimal performance, or if you want to customize other features in Windows 3.1, you can also begin with a standard configuration by using the "clean boot" disk. Make one change at a time to make troubleshooting easier.

Note Use the standard drivers included in the Windows 3.1 or MS-DOS 5.0 packages. Avoid third-party drivers unless you can confirm that such drivers have been tested and are known to be compatible with Windows 3.1.

Troubleshooting Setup

Flowchart Series 1
Troubleshooting Setup

Windows Setup is two programs bound into one executable file. The first program is MS-DOS Mode Setup, which installs the critical system files used to start Windows. The second program is Windows Mode Setup, which installs the Windows applications and the supporting files, sets up the Program Manager groups, and installs printer drivers.

Many installation problems are caused by your system configuration—for example, by TSRs or third-party memory managers that conflict with the Windows memory managers. Failure can occur at three key points in Setup:

- During initial detection of TSRs in your system.
- During hardware auto-detection in MS-DOS Mode Setup.
- During the initial loading of Windows.

If Setup fails at any of these stages, you might see an error message describing the problem, or Setup might crash and return to the command prompt, or your system might require rebooting. The following sections discuss possible solutions to problems you might experience during Setup.

Tip To see a list of switches that you can use to start Setup, type **setup /?** at the command prompt.

Troubleshooting TSRs During Setup



Windows Setup can check your system for memory-resident programs, looking in AUTOEXEC.BAT and CONFIG.SYS and also checking the programs currently running in memory. If Setup finds a program with which Microsoft testing has uncovered problems running with Windows 3.1, a message appears on your screen, giving you the opportunity to continue with Setup or exit.

To check for conflicting memory-resident software with Windows Setup:

- Type **setup /t** at the command prompt and press ENTER.

If you receive a message that incompatible TSRs or drivers are present when you try to run Setup, we recommend that you take the following steps.

To solve Setup problems related to incompatible TSRs:

1. Read SETUP.TXT on Windows Setup disk #1 for information about TSRs you are running. This information will tell you if you should install Windows 3.1 while this program is loaded.

   If you choose to install Windows 3.1 without changing your system configuration, follow the instructions on the screen to continue with Setup.
2. If you decide that you should not install Windows 3.1 with the programs loaded, follow the instructions on the screen to exit Setup.
3. Uninstall the TSR by following instructions in the program's documentation or by removing the filename from your AUTOEXEC.BAT or CONFIG.SYS file.
4. Reboot your computer and start Windows Setup again.
5. After Windows is successfully installed, you might want to restore the memory-resident programs.

Some TSRs run with Windows if you start them after starting Windows. Read the SETUP.TXT file on Windows installation disk #1 for information about the best way to run a specific TSR with Windows.

Check the entries in the **[incompTSR1]** and **[incompTSR2]** sections in the SETUP.INF file in your Windows SYSTEM subdirectory for up-to-date lists of software that conflicts with Windows Setup. The TSRs listed in **[incompTSR2]** can cause problems with Setup. We recommend that you uninstall any of this software before running Windows Setup. The TSRs and device drivers listed in **[incompTSR2]** can cause problems if they are running when you run either Setup or Windows 3.1.

Troubleshooting MS-DOS Mode Setup

Flowchart 1.2 Windows Hangs During Setup

In the first stage of MS-DOS Mode Setup, Setup uses an auto-detection scheme to determine your system's hardware configuration, specifically identifying the type of processor, machine, display adapter, mouse, and network.

If Setup hangs your system or pops out to the command prompt at this point and your system doesn't have any TSRs running, then the Setup auto-detection scheme is probably causing the problem. You can start Setup with the **/i** switch to tell Setup to ignore auto-detection. To do this, type **setup /i** at the command prompt and press ENTER to start Setup.

If you turn off auto-detection, Setup will select a standard configuration. When the hardware configuration screen appears, the list will be similar to this:

Computer:	MS-DOS System
Display:	VGA
Mouse:	Microsoft, or IBM PS/2
Keyboard:	Enhanced 101 or 102 key US and Non US keyboards
Keyboard Layout:	US
Language:	English (American)
Network:	Network not installed

If your system does not match the default settings, you might need to select the appropriate settings manually, following the instructions on screen.

These machines can be specifically selected during Setup:

- MS-DOS System
- AST: All 80386- and 80486-based machines
- AT&T PC
- AT&T NSX 20: Safari Notebook
- Everex Step 386/25
- Hewlett-Packard (all machines)
- IBM PS/2 model L40SX
- IBM PS/2 model P70
- Intel 386 SL based system with APM
- MS-DOS System with APM
- NCR: All 80386- and 80486-based machines

- NEC PowerMate SX Plus
- NEC ProSpeed 386
- Toshiba 1200XE
- Toshiba 1600
- Toshiba 5200
- Zenith: All 80386-based machines

These networks (and specific versions of most networks) can be specifically selected during Setup:

- No Network Installed
- 3Com 3+Open
- 3Com 3+Share
- Artisoft LANtastic
- Banyan Vines
- DEC Pathworks
- IBM OS/2 LAN Server
- IBM PC LAN Program
- Invisible Software Invisible Network (version 2.21)
- Microsoft LAN Manager
- Microsoft Network (or 100% compatible)
- Novell Netware
- TCS 10Net

Troubleshooting Windows Mode Setup

The second point where Setup might fail occurs when MS-DOS Mode Setup is finished and Windows Mode Setup is starting. During MS-DOS Mode Setup, modifications are made to WIN.INI, identifying Setup as the shell. The last thing that MS-DOS Mode Setup does is to start Windows. Then Windows begins loading the Windows files, in this order:

1: WIN.COM	4: MOUSE.DRV	8: GDI.EXE
2: KRNL286.EXE	5: DISPLAY.DRV	9: USER.EXE
or KRNL386.EXE	6: SOUND.DRV	10: NETWORK.DRV
3: SYSTEM.DRV	7: COMM.DRV	11: SETUP.EXE

Flowchart 1.2 Windows Hangs During Setup

If you are not running any TSRs and your machine hangs when Windows Mode Setup is starting to run for the first time, a file was probably unable to load. Most problems with Setup failing at this point are due to an incorrect display driver selection. Either the auto-detection scheme incorrectly detected the system configuration or your custom choices were incorrect.

Certain display adapter chip sets lead Setup to detect the display adapter incorrectly. Typically, this error is due to an incomplete implementation of BIOS on older display adapters. For example, if the card is detected as a Video 7 VGA with 512K and it is an older machine, the display adapter might not be identifying itself correctly.

An easy test is to rerun Setup and select a basic display adapter (usually EGA or VGA) to see if that works.

For more troubleshooting tips about display adapters, see "Troubleshooting Display Problems" later in this chapter.

Note If you are installing Windows from a network using **setup /n**, and Setup fails just as Windows is loading for the first time, check to see that all the files in the list above are present on the network and that they are not damaged.

Troubleshooting Windows Configuration

Five main topics are described in this section:

- Desktop configuration problems
- TSR compatibility problems
- Display problems
- MS-DOS configuration problems
- Expanded memory management problems

Troubleshooting the Desktop Configuration

You might get an error message when you start Windows that indicates a .GRP file is invalid or damaged, and that instructs you to reconstruct the group. This error can occur if:

- A group (.GRP) file has been deleted or damaged.
- The files were moved into another directory, but the PROGMAN.INI file was not updated to reflect this change.

You might also need to rebuild groups if no group windows or icons appear when you enter Program Manager after starting Windows. To rebuild groups, follow the procedure outlined in Chapter 15, "Maintaining Windows with Setup," in the *Windows User's Guide.*

Troubleshooting TSR Compatibility Problems

Setup does not detect all memory-resident programs that might cause problems running with Windows. Also, if you run a TSR after installing Windows, Windows itself cannot detect a program to warn you if there is a problem. Two kinds of problems might occur:

- The interaction of the program with Windows might result in an unexpected system response such as a system crash or an incorrect display.
- Either the program or Windows might require that you take special steps to get these programs to run together.

Flowcharts 5.6 and 5.7 Disabling Norton Desktop and PC-Tools

If you have problems running Windows or a TSR program, check the list in SETUP.TXT in the Windows SYSTEM subdirectory to see if there is a known conflict and a solution. Otherwise, begin troubleshooting by removing the program from memory to see how well Windows runs when the program is not present.

TSRs That Cause Problems with Setup

These programs might cause problems with Setup and should be removed before installing Windows, but should work with Windows after installation:

- ASP Integrity Toolkit version 3.7 causes various problems with Setup.
- Data Physician Plus version 2.0 (VirAlert) causes problems during Setup. Version 3.0 appears to correct this problem.
- Norton Anti-Virus version 1.0 can cause Setup to crash. Version 1.5 appears to correct this problem.
- PC-Kwik version 1.59 might cause Setup to crash. Version 2.0 appears to address this problem.
- SoftIce Debugger causes Windows Mode Setup to crash.
- Vaccine causes Setup to hang.
- VDefend, a PC-Tools Deluxe TSR, should not be in use during installation, but you can run it with Windows after Setup is complete.
- Virex-PC version 1.11 causes Setup to crash.
- ViruSafe version 4.0 causes Setup to crash. Version 4.5 appears to correct the problem, but its authors recommend that you run ViruSafe with the **/C-** switch with Windows.

TSRs That Cause Problems Running with Windows 3.1

These programs cause problems running with Windows 3.1:

- ANARKEY version 4.00 might prevent Windows 3.1 from loading under certain circumstances. Version 4.01 appears to correct these problems.
- APPEND, an MS-DOS utility, interferes with the ability of Windows and Windows applications to build valid paths for files they want to access. Do not use APPEND with Windows 3.1.
- DOSCUE, a command-line editor, produces unreliable results with Windows 3.1. We recommend that you do not use this program with Windows 3.1.
- GRAPHICS, an MS-DOS utility, is loaded into all VMs for non-Windows applications running under Windows 3.1. Printing from those VMs can be unpredictable, because the VM that has the focus sends its output to the printer, and printing becomes intermixed if you switch between non-Windows applications.
- JOIN, an MS-DOS utility, works fine with Windows 3.1 if you do not change the state of the joined drives during a Windows 3.1 session. This includes adding or removing joined drives from within Windows. We recommend that you do not use JOIN when running Windows Setup or Windows 3.1.
- LanSight version 2.0 is a utility for controlling and monitoring workstations attached to a Novell network, and it requires TSRs to be loaded on the workstations before Windows is started. If a message asking permission to view the screen is sent from the supervisor workstation to a workstation running Windows 3.1, the workstation running Windows might require rebooting.
- Lockit version 3.3 (PC Vault, Secure It) strip spaces out of the **load=** entry in WIN.INI. Version 3.3+ appears to correct this problem.
- MIRROR, an MS-DOS utility, cannot be removed from memory with the **mirror /u** command if it is loaded from Windows standard mode using the File Run command.
- Newspace version 1.07, a disk compression utility, is not compatible with Windows 3.1.
- Norton Utilities version 5.0 programs Diskreet and Ncache are not compatible with Windows 3.1 and can cause your system to hang. Version 6.01 solves this problem, with some limitations.
- Norton Utilities version 6.01 DiskMon, the disk monitoring program, may cause inconsistent results under Windows 3.1 if you are trying to access files in a directory that DiskMon is monitoring. We do not recommend that you use this TSR with Windows 3.1.

- Printer Assistant is a TSR that allows workstations on Netware 286 and 386 to share a printer. Some Windows applications print garbage mixed with correct output when this TSR is in use.
- XGAAIDOS.SYS causes Windows 3.1 to freeze at the opening logo in standard mode.

TSRs That Need Special Consideration

These programs will run with Windows if you take special action to make the two programs work together:

- BOOT.SYS, a power user's tool for booting under multiple configurations from a menu, creates several sections in CONFIG.SYS and AUTOEXEC.BAT. Windows Setup only modifies the first section in CONFIG.SYS and AUTOEXEC.BAT. You need to manually modify the other sections to use Windows 3.1 with the alternate configurations.
- Doubledisk version 2.5 creates "phantom" disk drives that Windows 3.1 might try to access. Vertisoft (makers of Doubledisk) has a DRVOFF utility to make these phantom drives invisible to Windows 3.1. Contact Vertisoft to get this utility if you want to run Doubledisk with Windows.
- FASTOPEN, a RAM resident MS-DOS utility, might need to be removed in low-memory situations for Windows 3.1 to load and function properly. Also, FASTOPEN causes File Allocation Table (FAT) problems when used with disk defragmenting utilities.
- LaserTools Control Panel version 2.2 might cause your system to hang in both standard and 386 enhanced modes if the TSR is loaded from within Windows 3.1. If you use this product, we recommend that you load it before starting Windows 3.1.
- The LANtastic KBFLOW TSR should not be loaded before you start Windows 3.1. To use KBFLOW, start Windows 3.1 first and run the TSR from within Windows.
- Le Menu version 1.0 can cause environment information such as PATH, PROMPT, and WINDIR to be lost when you run a non-Windows application under Windows. If you are going to start Windows 3.1 from a Le Menu menu option, we recommend that you set up Windows 3.1 as a batch file menu option so that COMMAND.COM can be loaded before Windows 3.1. (See the Le Menu documentation.)

- Logitech Mouse Software versions 5.0 and 6.0 must be loaded from within a non-Windows application after it is launched in Windows if you want to use Click and Logimenu, even if the TSRs were loaded before starting Windows 3.1. After exiting Windows 3.1, reload these TSRs for them to function from the command prompt.
- Norton Utilities version 6.01 Ncache program prevents you from creating a permanent swap file if this caching program is on a disk drive. Windows 3.1 will work with this program, however.
- PC-Tools Deluxe version 6.0 (Desktop) might cause your machine to hang if you launch it from a non-Windows application running under standard mode Windows. In 386 enhanced mode, it might cause your machine to reboot. Version 7.0 of PC-Tools appears to correct these problems. Even with version 7.0, however, you should not run Desktop from within a non-Windows application if the TSR was loaded prior to Windows 3.1. We recommend that you run Desktop from a PIF.
- Pyro! version 1.0, a screen saver, will blank the screen after the delay period has expired if it is loaded before Windows is started. At that point, it is impossible to restore the Windows screen.
- SideKick versions 1.0 and 2.0 and SideKick Plus cause several problems with Windows 3.1. We recommend that you run SideKick from within Windows with a PIF instead of loading it as a memory-resident program.
- SPEEDFXR is not compatible with Windows 3.1 as a TSR. Running SPEEDFXR -X appears to work better with Windows 3.1, because it does not remain in memory.
- SUBST, an MS-DOS utility, works fine with Windows 3.1 if you do not change the state of the substituted drives while running Windows. This includes adding or removing substituted drives from within Windows.
- Trantor T100 Host Adapter Driver (TSCSI.SYS) identifies the SCSI hard disk drive as a removable drive, so your system might crash if you try to access this drive through File Manager. Other unexpected results can occur while deleting files through File Manager.

For information about running network drivers with Windows, see "Troubleshooting Networks" later in this chapter.

Troubleshooting Display Problems

Flowchart 2.3 Display Adapter Problems

Many problems with garbled displays, inadequate screen refreshing, and other odd behavior are related to the display adapter or display driver. If you cannot find another explanation in this chapter for faulty behavior in your system, check the details in this section closely for a description of symptoms you may be experiencing.

Display drivers actually have three parts:

- **The Windows display driver** (*.DRV; for example, VGA.DRV), which contains the code that knows how to talk both to the display adapter board (hardware) and Windows (software).
- **The grabber**, which contains the code that supports data exchange between Windows and non-Windows applications. The standard mode grabber only supports copying and pasting text; the 386 enhanced mode grabber supports copying text and graphics up to the standard VGA resolutions (video mode 13). The grabber is built into the display driver file, but requires supporting font files. These files use the filename extension .2GR for standard mode and .3GR for 386 enhanced mode (replacing the .GR2 and .GR3 extensions used in Windows 3.0).
- **The Virtual Display Driver** (VDD*x*.386), which provides virtual display support for running non-Windows applications in virtual machines in 386 enhanced mode.

The naming conventions listed parenthetically are not required. However, most drivers use these naming conventions.

Display-related problems typically fall into two categories:

- Problems with third-party display drivers (those provided by the display adapter manufacturer).
- Problems with display adapters that use nonmaskable interrupts (NMI).

This section will help you determine the source of and solutions for display problems.

1. Windows or Windows applications are not displayed correctly.

In this case, the display driver may be incorrect or corrupted, and a correct driver must be installed. Or the problem may be related to NMIs, described in item #4 of this section.

2. You can't copy and paste data between non-Windows applications and Windows.

Usually this is because the grabber files are incompatible. You might encounter grabber problems in an incomplete installation of a third-party display driver. Windows might run fine, but non-Windows applications are not displayed correctly. To correctly install third-party display drivers, you must copy the driver and the appropriate grabber files, and you must also update the entries for **286grabber=** or **386grabber=** in the **[boot]** section of SYSTEM.INI.

3. You can't start non-Windows applications, or an error message identifies an incompatible display adapter.

The problem is probably that your system does not have a complete set of drivers for that display adapter. Or else that display adapter is not supported by Windows.

If the display adapter can be configured as standard VGA, set the card for that mode, and make sure Setup detects the appropriate display adapter, so that Windows recognizes the display adapter as a standard VGA board.

Also, check with the manufacturer to see if updated Windows 3.1 display driver files (.DRV, .2GR, .3GR, and VDD*x*.386 files) are available.

4. The Windows logo appears in 386 enhanced mode, then disappears and leaves the screen blank.

A few display adapters use nonmaskable interrupts (NMI) to switch video modes to support higher resolutions and to switch modes on the fly. Display drivers shipped with Windows do not support this type of dynamic mode switching, and Windows will not update the display.

A typical scenario with NMI display adapters is that Windows runs correctly in standard mode, but when you try to run Windows in 386 enhanced mode, the logo screen appears briefly, then scrolls off the bottom of the screen, leaving the screen blank. Windows is actually running, but the screen is not being updated. (At this point, you can usually press ALT+F4 and ENTER to exit Windows and return to the command prompt.)

To solve the problem, check the manual for the display adapter to find out how to turn off the NMI or extended modes for the display adapter. For example, some ATI Technologies display adapters use NMI support, but ATI Technologies also provides Windows display drivers that handle NMI correctly.

5. An incorrect-system-version error message appears when you try to run a non-Windows application in 386 enhanced mode.

The VDD*x*.386 and .3GR file versions are probably mismatched. If the Virtual Display Device and 386 grabber have mismatched versions, edit SYSTEM.INI to make sure that one of the following configurations is installed:

- These are the SYSTEM.INI entries for Windows 3.1 VGA and Super VGA:

```
[boot]                          [386enh]
386grabber=vga.3gr              display=*vddvga
```

  The "*" in the **display=** entry indicates that the information is built into the WIN386.EXE file, rather than existing in a separate file.

- These are the SYSTEM.INI entries for some third-party and OEM display adapters that require Windows 3.0 versions of VDD and 386 grabber files:

```
[boot]                          [386enh]
386grabber=vga30.3gr            display=vddvga30.386
```

Some OEM display drivers for Windows 3.1 that are shipped with display adapters also include VDD files, 386 grabber files, or both. Be sure that the correct files are installed for the display adapter and that SYSTEM.INI is updated. If the OEM driver disks include a 386 grabber file, but SYSTEM.INI contains the entry **display=*vddvga** in the **[386enh]** section, you must change this entry to **display=vddvga30.386**. Also, the VDDVGA30.386 file must be copied and, if necessary, expanded into the Windows SYSTEM subdirectory from the Windows Setup disks.

6. Your system has a VGA display adapter, and you experience problems running Windows in 386 enhanced mode.

Some VGA-compatible cards such as Video 7 and Paradise use additional memory to enhance the performance of their cards. Display adapter boards with the main chips manufactured by Tseng Labs, Chips and Technologies, Paradise, and Headland Technologies also work this way.

Windows detects most of these cards and automatically excludes the additional memory. However, you must add an **emmexclude=C400-C7FF** entry in the **[386enh]** section of SYSTEM.INI if:

- Your system has both an 8514 and a VGA display adapter but you configured Windows for the 8514.
- Your system has an enhanced VGA adapter that Windows does not recognize.
- Your system has both a VGA display adapter (primary) and an 8514 (secondary), and you have problems switching between full-screen non-Windows applications when running in 386 enhanced mode.

For more information about other problems that can be solved with entries for **emmexclude=**, see "Troubleshooting Windows Running in 386 Enhanced Mode" later in this chapter.

7. Your system has a VGA adapter, and an application does not display properly.

The digital-to-analog converter (DAC) on the VGA card might need to be updated. Contact the display adapter manufacturer for an upgrade. In the meantime, if your application has a command to refresh the screen, you might be able to temporarily correct the display so that you can use the application. For example, if the display is a color display, you can type **mode co80** at the command prompt and press ENTER.

See the application's documentation for information about commands that refresh the screen.

8. Your system has a SuperVGA adapter, and you experience display problems.

If you do not update your display driver to a Windows 3.1 driver, Windows might not work properly when running in 386 enhanced mode. Update your Windows system configuration to use the Windows 3.1 SuperVGA driver if you have a video card and multifrequency monitor with 800x600 16-color resolution but are currently using a third-party Super driver or one of these Windows 3.0 Supplemental Drivers:

- ATI Mode 54h (800x600 16-color) version 2.00
- CHIPS Super VGA 82C451 (800x600 16-color)
- CHIPS Super VGA 82C452 (800x600 16-color)
- Paradise VGA (800x600 16-color)

The Windows 3.1 SuperVGA driver also contains new features and performance enhancements, including the ability to obtain over 50 percent more screen space when running Windows.

9. Your system has a TIGA or DGIS display adapter, and you want improved performance.

If you are using a Texas Instruments Graphics Architecture (TIGA) or Direct Graphics Interface Standard (DGIS) display adapter, use the Windows 3.1 drivers for this display. Windows provides two versions of the TIGA display driver: "TIGA (Small Fonts)" and "TIGA (Large Fonts)."

We recommend that you use the Large Fonts version if you use the display at resolutions of 1024x768 and above. If you are using a 16-inch or larger monitor and want to fit more information on screen, use the Small Fonts version. To change the resolution, you must quit Windows and use the TIGAMODE utility or another resolution utility supplied with your display adapter. Unless you want to use a different version of the TIGA driver (Small Font or Large Font) to change font sizes, you don't have to run Windows Setup after changing your TIGA screen resolution.

If you use a TIGA-based display adapter and run Windows in 386 enhanced mode, you can increase the amount of conventional memory available to non-Windows applications by loading the TIGACD driver from WINSTART.BAT instead of from the AUTOEXEC.BAT file. This ensures that the TIGACD driver will load only when Windows is running and will not occupy conventional memory space in 386 enhanced mode.

We do not recommend loading the TIGACD driver from WINSTART.BAT if you run TIGA applications outside of the Windows environment. Also, to run Windows in standard mode on a TIGA display adapter, you must load the TIGACD driver into memory before starting Windows.

10. You experience specific display problems with the PS/2, Toshiba plasma display, Video Seven 256-color adapter, or other hardware.

See the README.WRI file in the WINDOWS directory. The README file contains up-to-date information about problems with display adapters for specific hardware.

Troubleshooting MS-DOS Configuration Problems

This section describes some problems related to the upper memory area and other MS-DOS issues. For information about problems with expanded memory, see "Troubleshooting EMS Memory" later in this section.

1. Your device drivers or programs won't load in the high memory area.

You should be able to run programs in the upper memory area if all the following conditions are met:

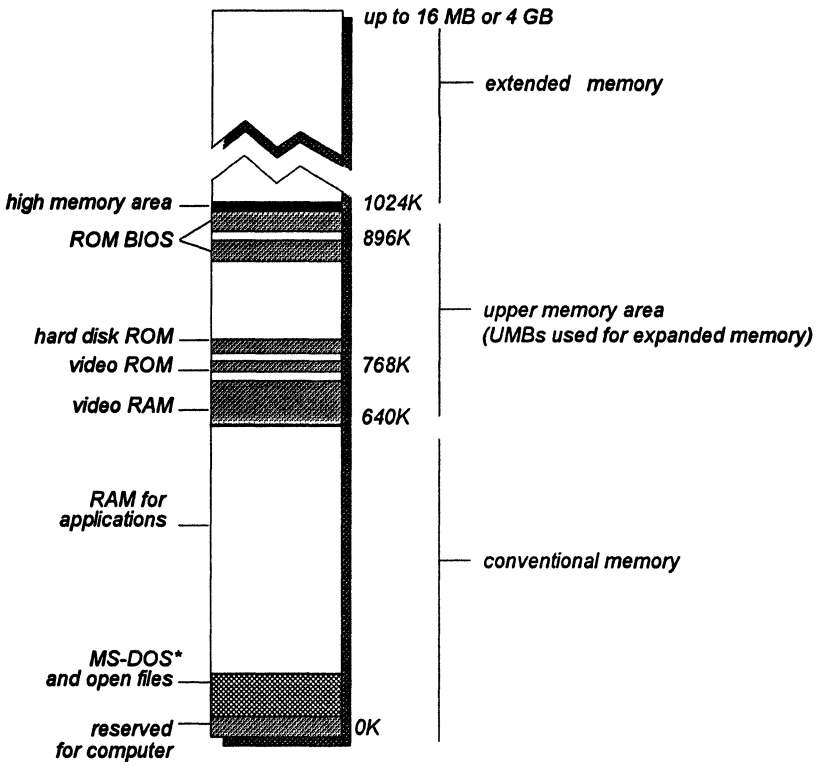
- The computer has an 80386 or higher processor.
- The computer has at least 350K of extended memory available.
- CONFIG.SYS contains a **device** command for HIMEM.SYS before any other **device** or **devicehigh** commands.
- CONFIG.SYS contains either a **dos=umb** or **dos=high,umb** command.
- CONFIG.SYS contains a **device** command for EMM386.EXE, which must include either the **noems** switch or the **ram** switch. This command must appear after the **device** command for HIMEM.SYS but before any **devicehigh** commands.
- CONFIG.SYS contains a **devicehigh** command for each device driver you want to load into the UMBs.
- AUTOEXEC.BAT contains a **loadhigh** command for each memory-resident program (TSR) you want to run in the UMBs.

If you have set up the system to run drivers and programs in the UMBs, but when you use the **mem /c** command, nothing appears in the upper memory section of the **mem** command output, you might be loading EMM386 with the **ram** switch in CONFIG.SYS, but your system might not have enough space in the UMBs to load programs or drivers.

To adjust EMM386 to provide additional space in the UMBs, follow the instructions in "Optimizing Use of the UMBs" in Chapter 6, "Tips for Configuring Windows 3.1." See also the *Microsoft MS-DOS Getting Started* booklet, p. 37.

Figure 13.1

PC memory map



2. Your computer stops only when you load a specific driver or memory-resident program (TSR) with the devicehigh or loadhigh command.

You might not be able to load that driver or TSR into a UMB. MS-DOS 5.0 device drivers and memory-resident programs should load into the UMBs without any problems (except HIMEM, EMM386, and SMARTDrive must be loaded in conventional memory).

Make sure the device drivers work by replacing the **devicehigh** command with the **device** command, so the device is loaded into conventional memory. You might also have to remove a size switch. If loading the device driver with the **devicehigh** command was the problem, your computer will now work properly.

3. You get a cannot-find-file error message, and you are running the MS-DOS APPEND utility.

When Windows opens a file such as WINWORD.EXE, it records the complete drive and path of this file, so that the file can be reopened later without regard to the current drive and directory. For example, Windows might record a path as C:\WINWORD\WINWORD.EXE.

The MS-DOS APPEND utility prevents Windows from reliably determining the proper path of files. For example, if C:\EXCEL is the current drive and directory when a file is opened, and D:\WINWORD is on the APPEND search path, APPEND might cause Windows to mistakenly build the filename as C:\EXCEL\WINWORD.EXE.

Later, when Windows tries to open this file again, an error occurs because the file is not in the drive and directory indicated. This error results in the message:

```
Change Disk
Cannot find WINWORD.EXE, Please insert in drive A:
```

Windows Setup will warn you if APPEND is found in your configuration and ask you to remove it before Setup proceeds. But Windows will not warn you about its presence if you install APPEND after Setup. APPEND is usually found in AUTOEXEC.BAT, and it should be removed or commented out.

Troubleshooting EMS Memory Problems

This section discusses solutions for several problems that can occur if you use EMM386.EXE with Windows 3.1.

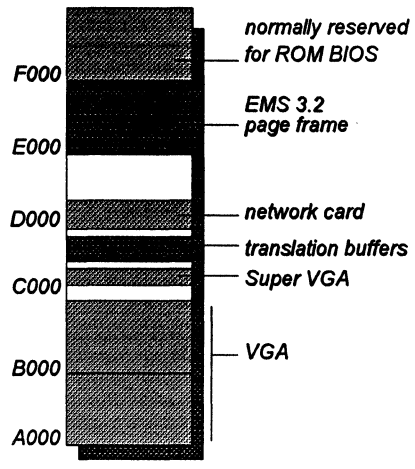
For more information about EMM386 and its startup parameters, see pages 605–609 of the *Microsoft MS-DOS User's Guide and Reference*. For more information about how to use program information files (PIFs) to provide expanded memory to non-Windows applications, see Chapter 7, "Non-Windows Applications," in the *Windows User's Guide*.

Tip To see a list of the switches for controlling EMM386.EXE after it has been loaded, type **emm386 /?** at the command prompt.

Figure 13.2

Example of UMBs in the upper memory area

For a VGA or V7VGA display adapter, a network card at D000-D7FF, and an EMS 3.2 page frame at E000-EFFF



1. You get an invalid-path-for-EMM386 error message when you start Windows in 386 enhanced mode.

If this happens, add the **/y=path** switch to the command line that loads EMM386 in CONFIG.SYS. This switch specifies the location of EMM386 when you start Windows. For example, if EMM386.EXE is in the C:\DOS directory, the command in CONFIG.SYS would be:

```
device=emm386.exe /y=c:\dos\emm386.exe
```

Make sure you specified the correct directory in CONFIG.SYS for the HIMEM.SYS and EMM386.EXE files. Also, make sure HIMEM and EMM386 are functioning properly. To do this, make a test boot disk with a CONFIG.SYS file that contains only the **device** commands for loading HIMEM and EMM386. The command that loads EMM386 should be:

```
device=c:\dos\emm386.exe noems x=a000-efff
```

And include a test AUTOEXEC.BAT file that contains the following:

```
path=c:\dos
prompt $p$g
set TEMP=C:\
```

If the computer stopped when you used the test disk to boot the system, you might have a problem with HIMEM or the hardware.

If the computer functions properly when you boot with this test disk, HIMEM works with EMM386.

2. Some other programs conflict with EMM386.

If you use a third-party disk-caching program other than SMARTDrive, it might not be working properly with EMM386. Remove that program temporarily and put a command in your AUTOEXEC.BAT file to load SMARTDRV.EXE. Contact the manufacturer of the third-party disk-caching program for more information about using it with EMM386.

3. Another program uses the same UMB space as EMM386.

If the computer stops, and none of the preceding solutions have worked, EMM386 may be using the same memory addresses as another program or card. For example, this could happen when you install a network card or a card for a device.

If you suspect a memory-resident program (TSR) is causing the problem, edit the AUTOEXEC.BAT file to remove the **loadhigh** command so the program loads into conventional memory. If loading the program with the **loadhigh** command was the problem, the computer should now work properly.

If your system has a card or drive that might conflict with EMM386, check its documentation to see what address space the card or driver uses. If the conflict occurs over a network, check with the network administrator to find which memory addresses are being used. Then use the **x=** switch to prevent EMM386 from using those addresses. For example, if your system's network card uses the D800h-DFFFh memory addresses, the command line that loads EMM386 in CONFIG.SYS should look like this:

```
device=c:\dos\emm386.exe noems x=d800-dfff
```

You can experiment with the command that loads EMM386 to determine which memory addresses are being used by the other program or device, as described in "Optimizing Use of the UMBs" in Chapter 5, "Windows 3.1 and Memory Management." For details about the command-line switches that can be used with EMM386, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

4. Microsoft Windows 3.1 does not provide expanded memory after EMM386 is installed.

The symptoms of this problem vary. Common symptoms include:

- When you start a non-Windows application, the application displays an out-of-memory message and stops running, or it behaves erratically and displays out-of-memory messages when you try to perform certain tasks.
- Windows 3.1 stops when you start some non-Windows applications, but you have no problem starting other non-Windows applications.

To solve the problem, change the **device** command line for EMM386 so that it includes the **ram** switch instead of the **noems** switch. If the application runs normally after you restart the computer, the **noems** switch caused the problem.

After you add the **ram** switch, other programs or drivers might not load into UMBs. (Use the **mem /c** command to find out.) If so, follow the instructions in "Adjusting EMM386 to Provide More UMBs" in Chapter 6, "Tips for Configuring Windows 3.1."

If your programs and device drivers still do not load into UMBs, the system might not have enough UMBs available to run those programs. See the computer's hardware documentation to find out which addresses are used.

Troubleshooting Windows Running in Standard Mode

Flowchart 5.9 Cannot Run Windows in Standard Mode

If you have successfully installed Windows, then using the **win /s** switch should force Windows to start in standard mode, provided that:

- The machine has an 80286 or higher processor.
- There is at least 256K conventional memory free.
- HIMEM.SYS or another XMS driver is installed.
- There is at least 192K extended memory free.

If you cannot start Windows in standard mode, the problem is usually related to the XMS driver or to not having enough extended memory.

1. RAMDrive or other resident software may be using extended memory and preventing Windows from accessing that memory.

Decrease the size of the RAM drive or do not load the memory-resident software.

2. The XMS driver is out of date or incompatible with Windows 3.1.

Replace the driver with the version of HIMEM.SYS in the Windows package.

3. HIMEM.SYS may be incorrectly identifying the machine type.

HIMEM uses an algorithm to detect the machine type, then uses the appropriate routine to access extended memory. When HIMEM loads at boot time, it reports which A20 handler is installed. A20 is a technical reference to the line on the processor that allows access to extended memory. During boot time, the display should read:

```
Installed A20 handler number x.
```

The value for *x* identifies the routine used to access the A20 line.

Certain machine types cause HIMEM to mistakenly identify which A20 handler to use. To force HIMEM to use the correct driver, you must use the **/m** switch in the **device=himem.sys** line in CONFIG.SYS, where **m** identifies the type of computer as either the number or name from the following table.

Number	Name	Computer type
1	at	IBM AT or 100% compatible
2	ps2	IBM PS/2
3	ptlcascade	Phoenix Cascade BIOS
4	hpvectra	HP Vectra (A and A+)
5	att6300plus	AT&T 6300 Plus
6	acer1100	Acer 1100
7	toshiba	Toshiba 1600 and 1200XE
8	wyse	Wyse 12.5 Mhz 286

For example, either of the following lines can be used to tell HIMEM to use the IBM AT method of accessing extended memory:

```
device=c:\dos\himem.sys /m:at    or    device=c:\dos\himem.sys /m:1
```

The table above lists values for some of the most popular hardware. Contact your hardware manufacturer for specific values if your system does not match values in the table.

For more details about using HIMEM.SYS with Windows 3.1, see Chapter 14, "Optimizing Windows," in the *Windows User's Guide.*

4. Your system doesn't have enough extended memory.

To make more extended memory available:

Flowchart 5.8 Out-of-Memory Errors

- Make sure your computer has as much physical extended memory as the program needs. If the computer does not have enough physical extended memory, you might want to contact your hardware vendor to determine how to add more to the system.
- If CONFIG.SYS includes a **device** command for RAMDrive or EMM386, these programs could be using too much extended memory. If so, reduce the amount of memory specified by editing the **device** command for each driver. For information about editing **device** commands in CONFIG.SYS, see pages 283–287 of the *Microsoft MS-DOS User's Guide and Reference.*
- Make sure there are no commands in CONFIG.SYS and AUTOEXEC.BAT to start unnecessary programs that use extended memory.

Troubleshooting Windows Running in 386 Enhanced Mode

If you have successfully installed Windows, Windows will start automatically in 386 enhanced mode provided that:

- The machine has an 80386 or higher processor.
- There is 256K of free conventional memory.
- An XMS driver such as HIMEM.SYS is already loaded.
- There is 1024K of free extended memory.

This section describes solutions to problems you might encounter when trying to run Windows in 386 enhanced mode.

1. You can run Windows in standard mode, but not in 386 enhanced mode.

Flowcharts 5.1 and 5.2 Cannot Run in 386 Enhanced Mode

If Windows runs only in standard mode, this indicates that there may be a conflict in the Upper Memory Blocks (UMBs). This conflict occurs when Windows does not detect a network card, an extended video board, a scanner board, or some other device.



To determine if there is a UMB conflict, try running Windows with the **/d:x** switch, which excludes all of the upper memory area (similar to setting **emmexclude=A000-FFFF** in the **[386enh]** section of SYSTEM.INI).

If Windows runs when you use this switch, there was a conflict with some device in the UMBs. To solve the problem, you need to identify the boards or devices in the machine. Also, use MSD.EXE to see if either shadow ROM or shadow RAM is enabled in the E000 segment. After you determine which memory addresses the devices use, add an entry for the **emmexclude=** in SYSTEM.INI to avoid accessing that address range. For example, many high-resolution video boards use the address range above C000 to C7FF. In that case, the correct entry in SYSTEM.INI would be **emmexclude=C000-C7FF**.

Limiting the excluded range to the specific address range is important because Windows uses that memory address range to place the API translation buffers and page frame. If Windows cannot locate enough free memory in the UMBs, it will allocate conventional memory for the translation buffers, taking memory away from non-Windows applications that you want to run. For more information, see "Placing Translation Buffers in the Upper Memory Area" in Chapter 5, "Windows 3.1 and Memory Management."

Note You can run MSD.EXE or use the MS-DOS **mem /p** command to see what drivers might be using the UMBs and their respective addresses. Also, you can type **win /?** at the command prompt to see a list of switches that you can use to run Windows 3.1.

---

2. The system crashes when you run Windows in 386 enhanced mode.

When Windows starts in 386 enhanced mode, it installs the API translation buffers in any free UMB area it finds. Windows might also compete for memory with adapter cards that don't sign in at the memory area they occupy. The standard is for adapters to leave a ROM signature 0xAA55 followed by the size of the area that is being used, but not all cards do this. If Windows tries to use an area that appears to be free but that is also being used by an adapter card, the system might crash.

With 386 loadhigh drivers such as QEMM and 386MAX, there can be additional contention problems within the UMBs, because the loadhigh drivers might already be using that area. With conventional memory at a premium on most machines, you must consider how much memory to allocate to the loadhigh drivers and which areas to leave open for the translation buffers.

To keep the translation buffers from competing for memory, add the entry **emmexclude=A000-EFFF** to the **[386enh]** section of SYSTEM.INI. This entry tells Windows not to position the translation buffers in the UMBs. Windows will allocate the buffers in conventional memory.

If Windows 386 enhanced mode crashed before implementing this switch, and now it doesn't crash, the next step is to isolate the specific area causing Windows to crash:

- If your system has a nonstandard video board, you can begin by excluding the range C600-C800, which some display adapters use without warning. If your system uses shadow RAM or has copied either system BIOS or video BIOS into the UMBs, then you should turn the feature off, or exclude the area that is being used (usually **emmexclude=E000-EFFF**).
- If the display adapter supports extended modes (like SuperVGA), also try turning off the extended mode support, which is typically a nonmaskable interrupt (NMI). For details, see item #4 in "Troubleshooting Display Problems" earlier in this chapter.

Flowchart 2.6 Cannot Access Hard Disk

3. You experience hard disk problems when running Windows in 386 enhanced mode.

Windows 386 enhanced mode, without SMARTDrive loaded, supports only standard ST506 and ESDI controllers. To use 386 enhanced mode with SCSI or other nonstandard devices, you must load SMARTDrive. When Windows detects SMARTDrive, it directs all disk access requests through SMARTDrive, so that SMARTDrive can function effectively as a double-buffering hard disk device driver, which eliminates most compatibility problems.

But even if SMARTDrive is loaded, in some configurations you might experience hard disk access errors, such as the following:

```
Cannot Read from Drive C:
Cannot Write to Drive C:
Drive C: Not Ready Error
```

If this happens, add the entry **VirtualHDirq=off** in the **[386enh]** section of SYSTEM.INI. Also, you must have SMARTDrive installed without the **/B-** switch. For example, add the **VirtualHDirq=off** entry to SYSTEM.INI if you use a Plus Hardcard or a SCSI hard disk drive using DMA.

This is the highest level of hard disk compatibility that can be achieved under Windows 3.1. If you still experience hard disk problems, contact the disk manufacturer for further assistance.

4. You get an internal-stack-overflow message while running Windows in 386 enhanced mode.

Stacks are temporary data structures that MS-DOS and applications use for processing hardware events. If you get a stack-overflow error:

- And if your system uses MS-DOS 3.2, include this line in CONFIG.SYS:

  ```
  stacks=9,192
  ```

- And if your system uses MS-DOS 3.3 or above, include this line in CONFIG.SYS:

  ```
  stacks=9,256
  ```

5. Typing slows down when you run more than one application, but your system seems to have plenty of memory.

The timeslice for multitasking is taking processing time away from the foreground application. You can increase the foreground priority in the 386 Enhanced dialog box in Control Panel or in the application's PIF. Better still, you can increase the priority of the foreground application only when it receives a keystroke by adding the entry **KeyBoostTime=.005** (or some larger increment) in the **[386enh]** section of SYSTEM.INI.

Troubleshooting for Non-Windows Applications

This section describes some solutions to problems that can occur while running non-Windows applications. See also the section "Troubleshooting Windows Configuration" in this chapter, because it discusses many problems that also relate to non-Windows applications.

1. You receive an "Unexpected MS-DOS Error #11" message.

If you receive this error message when you try to run a non-Windows application from Windows 3.1, it is a Kernel internal error that indicates Windows has tried to execute a file that has an invalid format.

If this happens in standard mode, then either the WINOLDAP.MOD file or the 286 grabber is corrupted; if in 386 enhanced mode, then either the 386 grabber or WINOA386.MOD file is corrupted.

Flowchart 1.7 Expanding Files from the Windows Disks

To solve this problem, copy the grabber file plus WINOLDAP.MOD or WINOA386.MOD file from the source disks, and expand these files, using EXPAND.EXE (which can be found on your Windows 3.1 installation disks or in your MS-DOS directory).

2. You experience problems running non-Windows applications in standard mode.

- If the application doesn't run in the startup directory specified in the application's PIF, check the Working Directory setting for the program item in Program Manager. This setting overrides PIF settings.
- If data is garbled or lost, check the Directly Modifies options in PIF Editor to ensure that the application is controlling the keyboard and any COM ports it uses. Also, make sure No Save Screen is not checked.
- If you can't paste screen captures, make sure No Screen Exchange is not checked in PIF Editor.
- If you can't switch back to Windows, make sure Prevent Program Switch isn't checked, and that Keyboard is not checked in the Directly Modifies options in PIF Editor.

3. You experience problems running non-Windows applications in 386 enhanced mode.

- If your system runs out of memory when you are running a high-resolution graphics application, check High Graphics in the PIF's Video Memory option and check Retain Video Memory.

  Conversely, to free as much memory as possible if your system has a Hercules or CGA adapter, check Low Graphics and make sure Retain Video Memory is not checked.

- If the mouse doesn't work when you run the application in full-screen mode, make sure that an MS-DOS mouse driver is installed. See also Flowchart 2.8 for specific troubleshooting procedures.

  Also, try choosing Full Screen option if your system is running out of memory.

- If the application doesn't display properly, make sure the Monitor Ports options are checked. If text is garbled or the cursor is in the wrong place, make sure Emulate Text Mode is not checked.

- If you can't paste into a non-Windows application, make sure Allow Fast Paste is not checked in the PIF Advanced Options dialog box.

4. You experience performance problems when running some applications at the same time, but each application runs okay alone.

You might have a device driver or memory-resident application that requires expanded memory or extended memory, but EMS or XMS memory is locked in the PIF for another non-Windows application. Examine the PIFs of all applications that run simultaneously, and clear the EMS Memory Locked and XMS Memory Locked options if you are trying to run applications simultaneously.

5. Lotus 1-2-3 version 3.1 won't start.

If Lotus 1-2-3 no longer runs properly, or it prevents you from opening large files, there could be several causes:

- Your system might not have enough XMS memory available. You need at least 384K. For information about how to make more extended memory available, see "Troubleshooting Windows Running in Standard Mode" earlier in this chapter.

- The PIF for Lotus 1-2-3 has been modified so that it no longer specifies enough extended memory. Specify that 384K of XMS memory is needed in the PIF. For more information, see Chapter 8, "PIFs and PIF Editor."

- EMM386 isn't set up to work with Lotus 1-2-3. For example, you might have loaded EMM386 with the **noems** switch.

Because Lotus 1-2-3 version 3.1 and EMM386 both run in protected mode, they need to communicate to avoid conflicts. Lotus and EMM386 can use the DPMI protocol to communicate. DPMI assumes that a page of expanded memory exists. A page is not available if EMM386 uses the **noems** switch. You must change the CONFIG.SYS setting. For example:

```
device=c:\dos\emm386.exe ram p0=d000
```

The **ram** switch gives programs access to both expanded memory and the UMBs. The **p0=d000** setting specifies a page for EMS.

When you restart the computer after making this change, you'll be warned that a LIM 3.2-compatible page frame does not exist. The standard size for a page frame is 64K, but since Lotus 1-2-3 is using XMS memory (not EMS), it only needs a 16K page. This provides more space for programs in UMBs.

6. **You experience problems running a specific application or using non-Windows 3270 applications.**

Check the README.WRI file for up-to-date notes on specific applications.

7. **You receive error messages such as general protection faults, application execution errors, or system integrity violations.**

See Flowcharts 5.3, 5.4, and 5.5 for specific troubleshooting procedures.

Troubleshooting Fonts

1. **You experience problems with your system when you choose TrueType fonts.**

Flowchart Series 3 Font Problems

Some display drivers that cache fonts in unused video memory have problems displaying TrueType fonts. If you experience problems, contact your display adapter manufacturer for updated display drivers.

2. **You use a third-party printer driver, and you cannot print TrueType fonts.**

Most printer drivers created for Windows 3.0 do not support TrueType fonts. Use one of the printer drivers in your Windows 3.1 package, or contact your vendor for an updated driver.

3. **You are using a font converter, and you experience system problems when you use TrueType fonts.**

Some font converters are not completely compatible with TrueType, causing problems in code segments while running. If you experience system crashes or other problems, do not use that font converter. Contact the manufacturer for an updated version.

4. **You have a minimum configuration for running Windows, and you experience problems when you use TrueType fonts.**

You might have problems if you attempt to use TrueType on a system with less than 2MB of memory. If you have such a system, to avoid low-memory situations, choose vector fonts or bitmap fonts, and avoid TrueType.

5. **You use Bitstream FaceLift and have a high-resolution display, and you experience problems when you use TrueType fonts.**

FaceLift is incompatible with some Windows 3.1 display drivers. Contact Bitstream for software updates, or disable FaceLift when you want to use TrueType. Or use your old Windows 3.0 display drivers with FaceLift.

6. **You have a type manager installed in your system, but you want to remove it.**

See Flowcharts 3.2, 3.3, and 3.4 for instructions on disabling Adobe Type Manager, Bitstream FaceLift, and HP Intellifont.

Troubleshooting Printing

Flowchart Series 4 Printing Problems

Many of the printing problems reported to Microsoft Product Support Services are actually hardware or configuration problems not related to Windows. This section provides tips for general troubleshooting for printing, then provides solutions to some specific problems related to printing in Windows.

A Strategy for Troubleshooting Printing

If you have problems printing, begin with these steps to isolate a minimal number of variables. By eliminating the variables, you can solve the problem.

- **Read the PRINTERS.WRI file in the WINDOWS directory.**

The PRINTERS file contains up-to-date information about printing from Windows with specific hardware.

- **Make sure you can print from MS-DOS.**

Copy a file to the printer port to make sure the communication path is working correctly. To do this, at the command prompt, type **dir > lpt***x* or **dir > com***x* (where *x* is the number for the port that the printer is connected to).

This method will not work for PostScript printers. To check a PostScript printer, copy TESTPS.TXT to the printer from your Windows SYSTEM subdirectory.

If you cannot print from MS-DOS outside of Windows, it's not a Windows printing problem. Contact your printer manufacturer or distributor.

- **Check the printer's cable connections.**

Make sure the printer is turned on and that it is online. Check that it has paper and toner.

- **Use a SET TEMP statement in AUTOEXEC.BAT.**

Make sure you include a **SET TEMP** statement in AUTOEXEC.BAT and that it points to an existing directory. For testing purposes, we recommend that you use the statement **SET TEMP=C:**. Verify that enough space is free on the hard disk for spooling temporary files while printing. If the TEMP environment variable points to a network drive, you must have create and write access to the target directory.

- **Make sure you are using the correct printer driver for the printer.**

The current printer driver can be seen by choosing the Printers icon in Control Panel. If the highlighted printer name doesn't match the printer that you are using, then you need to install a printer driver that does match the printer, as described in Chapter 5, "Control Panel," in the *Windows User's Guide.*

- **If you print to a COM port, you might need to configure the COM port with the MS-DOS Mode command or with Control Panel settings.**

See the MS-DOS *User's Guide* for information about the **mode** command. In Windows, choose the Ports icon in Control Panel to configure the COM port. Make sure that the printer is selected, and then try to print. If there is a program that redirects or reconfigures the printer at the MS-DOS level, you should print to LPT*x*.DOS (where *x* is the port number), which tells Windows to let the operating system handle the printing. For details, see "WIN.INI [Ports] Settings and the LPT*x*.DOS Method" in Chapter 10, "Printing."

- **Check the communication settings for serial printers.**

Flowchart 2.2
Serial Port Problems

If you have a serial printer, make sure the communications settings are correct. The settings on the printer and the Windows 3.1 settings should be the same. In Windows, the communications settings are set in Control Panel. Almost all serial printers, including Hewlett-Packard plotters and most PostScript printers, use a setting of **9600 baud, no parity, 8 bits, 1 stop bit**, and **hardware handshaking**.

To make sure that the printer has the same settings as Windows, use the printer's interactive front panel or DIP switches.

Handshaking for PostScript printers can be set in the Options section of the Windows Printer Setup dialog box. The PostScript driver will send a short program to the printer telling it to switch to the desired handshaking mode.

- **Override direct printing for testing purposes.**

Check the decision tree shown in "Determining Print Method" in Chapter 10, "Printing." If the decision tree indicates that Windows is trying to print directly to a parallel port on the system, try overriding this with the LPT*x*.DOS method discussed in that chapter.

- **If you are printing on a network without a network driver, use the LPT*x*.DOS method.**

You must use LPT*x*.DOS if you are printing to a network-redirected port that is also present in hardware. For example, your system might have a physical LPT1 port in the machine and also have a virtual LPT1 on the network. Again, this is only necessary if your system is on a network and you are not using a Windows network driver.

Also, make sure you are using an up-to-date Windows network driver. For more information, see "Troubleshooting Networks" later in this chapter.

Tips for Specific Printing Problems

The following items suggest solutions to specific printing problems.

1. **You get a cannot-print error message.**

This error message can occur on a network or on a physical parallel port. If your system is on a network, make sure you are using the right Windows network driver. If you are not using a Windows network driver, and the network is not supported, make sure you use the LPT*x*.DOS method.

Windows 3.1 requires that the Centronics Paper Out signal be Off and that the Select In signal be On. If one of these conditions is not met, Windows gives a cannot-print error message when you try to print. The possible causes of such a problem include:

- The printer is off line, out of paper, or jammed.
- The printer has faulty cabling.
- The printer uses an incompatible print-sharing device, switchbox, or buffer.

For testing purposes, remove all such devices and connect the printer directly to the computer's parallel port with a single cable. If the problem persists, try a different cable. If you use a switchbox to switch printers electronically, test printing with the switchbox removed. If you are printing over a network, print locally (directly connected to the machine) to see if the network is part of the problem.

Check the TEMP variable in your AUTOEXEC.BAT file. Make sure you haven't run out of disk space.

2. You get a printer error message in the middle of printing a document.

You can set the Windows printer time-outs for each individual printer in the Printers section of the Control Panel. The time-out options are Device Not Selected and Transmission Retry. The Device Not Selected option indicates how long Windows will wait for the printer to be online before it gives an error. The Transmission Retry option sets the time that the printer will stay busy before Windows gives an error.

Try increasing the Transmission Retry setting. The Transmission Retry time-out starts counting as each character is sent to the printer. If the time-out lapses before the printer is ready to accept another character, you'll get a printer error message.

3. Your PostScript print jobs are incomplete or missing.

- Check the communication settings and the cable.
- Check that the printer has enough memory to handle the print job.

If the problem is intermittent, or if it only happens with a particular application, download the PostScript Error Handler to help debug the problem.

To get a PostScript error report:

1. Choose the Printer icon in Control Panel, then choose the Options button in the Printer Setup dialog box.
2. Choose the Advanced button and, in the Advanced Options dialog box, check the Print PostScript Error Information option. Specify any other options you want in the dialog box, then choose the OK button.
3. Repeat the procedure that resulted in the problem.

If you see a PostScript error message, the problem is being caused by your application or by the degree of compatibility of your PostScript printer.

If you don't see a PostScript error message, recheck the other solutions discussed in the previous section.

Troubleshooting Networks

Most network-related problems that occur in 386 enhanced mode are related to the system configuration. Conflicts with hardware occur in three areas:

- Interrupt conflicts with devices or drivers
- Memory port addresses (I/O addresses)
- Shared-memory addresses

You might also experience problems with specific network drivers.

Make sure you read the NETWORKS.WRI file in the WINDOWS directory for up-to-date information about running Windows on a network. Also, check the Network selection in the Windows Setup dialog box to be sure that the correct network option was selected during Windows installation.

1. You suspect interrupt conflicts with either device drivers or other hardware.

If you suspect interrupt conflicts, refer to the list of IRQs in Troubleshooting Flowchart 2.7. You can run MSD.EXE from the command prompt to see what devices or boards are using specific IRQs.

2. You suspect I/O port address conflicts.

Flowchart 2.2 Serial Port Problems

If you suspect that the memory port addresses are conflicting, determine the I/O port addresses for the add-on boards in the system. Then select an address that does not conflict.

Typical I/O port addresses allowed on network adapters do not conflict with the machine. However, the add-on boards often conflict. On some machines, the CMOS Setup program identifies the I/O ports that the communications devices and other standard devices are using.

3. You suspect shared-memory address conflicts.

The shared-memory addresses are located in the UMBs and can cause subtle problems when video cards are in high-graphics mode or when the network card is being accessed.

Also, shadow RAM and shadow ROM can cause memory address conflicts, so you might want to investigate the CMOS Setup to check for shadowing options.

4. You suspect network driver problems.

First, review the NETWORKS.WRI file for any information about your specific network driver.

Some network shells and drivers cause problems. For example, Novell shells earlier than version 3.01e can cause problems, especially if you use the EMSNET3.COM, EMSNET4.COM, XMSNET3.COM, or XMSNET4.COM shell. These shells have exhibited random problems, but the most significant and reproducible problems concern printing.

- Replace your current NetWare shell with the NETX.COM provided with Windows 3.1. NETX.COM will run on any version of MS-DOS 3.0 or higher.
- If you send a job to the printer and get garbage, determine which shell is loaded. Closer inspection of the garbage printout might reveal that the print job started at address 0:0 and printed out the interrupt vector table, the MS-DOS environment, or other factors. These clues can help you find the source of the problem.

Troubleshooting GP Faults

Flowcharts 5.3-5.5 GP Faults and Other Errors

A General Protection fault (GP fault) occurs when a Windows application writes to a memory space where it doesn't have access. This corrupts any other code that already occupies that memory space. In Windows 3.0, after a GP fault happened, sometimes the memory was so corrupted that even applications that didn't cause the problem could trigger a GP fault message; the system remained unstable so that you had to exit Windows and reboot the system.



Windows 3.1 and Windows applications designed especially for Windows 3.1 are significantly more reliable. A new internal feature called Parameter Validation checks for invalid parameters.

If you do see a GP fault message while running Windows 3.1, the message will provide specific information about which application caused the error and where the error occurred. You can probably continue running the application after a GP fault, so that you can save your work before exiting the application.

You can use Dr. Watson, a utility provided with Windows 3.1, to learn more about what is causing a GP fault. For details, see "Dr. Watson and GP Faults" later in this section.



You can also use the Windows 3.1 local reboot by pressing CTRL+ALT+DEL if a Windows application or a non-Windows application is hung or is not responding to the Windows command to close. When you use local reboot, you will see an error message with instructions for closing the application.

Tips for Troubleshooting GP Faults

This section describes a few reasons why a GP fault might occur.

1. **You are running an application that was designed for an earlier Windows version (2.x) under Windows 3.1.**

If an application was designed for Windows versions earlier than 3.0, it displays a dialog box when you try to start it, describing this error. If the application has not been specifically designed to run under Windows 3.0 or later versions, it will trigger a GP fault.

2. **You manually selected an incorrect machine or network during Windows installation, or Setup's auto-detection misidentified elements of your system.**

When you install Windows 3.1 on some machines and networks, you can't use Setup's auto-detection as described in "Troubleshooting Setup" earlier in this chapter. Instead, you have to override the default and specifically make the correct selection, or Windows will not operate properly. If you don't specifically select these machines and networks, you might get a GP fault.

For details about system selections in Setup, see "Troubleshooting MS-DOS Mode Setup" earlier in this chapter.

3. **Your CONFIG.SYS or AUTOEXEC.BAT files contain either incompatible terminate-and-stay-resident programs (TSRs) or unsupported network drivers.**

You can test your system for incompatible TSRs or unsupported network drivers by temporarily removing them so that the system is at a minimum operable configuration. If doing this solves the GP fault problem, then you've begun to isolate the cause.

To find the specific cause, replace the lines for the TSRs and network drivers in CONFIG.SYS or AUTOEXEC.BAT one by one until the problem reappears. Then you'll know which one is causing the GP fault.

4. Your system has a page-mapping conflict when you run Windows in 386 enhanced mode.

Page-mapping conflicts only occur in Windows 386 enhanced mode. And this usually occurs because Windows doesn't see the page frame and so maps over it when you are using a third-party EMS manager instead of EMM386.EXE.

Test for a page-mapping conflict by running Windows with the **win /s** switch (standard mode) or the **win /3 /d:x** switch (excluding the upper memory area). If the problem doesn't happen, you might be experiencing a page-mapping conflict in the UMBs.

To isolate the problem, start by excluding the entire UMB from mapping with an **emmexclude=A000-EFFF** entry in the **[386enh]** section of SYSTEM.INI, then begin narrowing the range as described in "Troubleshooting Windows Running in 386 Enhanced Mode" earlier in this chapter.

If you have a Micro Channel (MCA) bus machine, such as an IBM PS/2, you might be able to determine the location of hardware adapters by booting with the machine's Reference Disk. If you have a standard ISA machine, consult the documentation for the adapters, or contact the manufacturer's technical support services to find out which memory locations they use.

5. Your system has an incorrect version of MS-DOS.

If your machine is not running the right version of MS-DOS, you might get GP faults. You should only use OEM versions of MS-DOS (such as Compaq MS-DOS or IBM PC-DOS) on OEM hardware platforms. For example, only use Compaq MS-DOS on Compaq machines.

To determine which version of MS-DOS to use, follow this rule:

- If your machine's manufacturer has an OEM version of MS-DOS, use it.
- If your machine's manufacturer doesn't have an OEM version of MS-DOS, use Microsoft MS-DOS and upgrade to version 5.0.

Dr. Watson and GP Faults


Dr. Watson

Dr. Watson is a Microsoft diagnostic tool that provides detailed information on the internal state of Windows when a system error occurs. If an application error occurs, Dr. Watson automatically creates a special file in your WINDOWS directory named DRWATSON.LOG. Dr. Watson also prompts you for details about the circumstances under which the application error occurred. Sometimes Dr. Watson detects a fault that might not be fatal, and you are given the opportunity to ignore the fault or close the application.

When you choose the Ignore option after an application error is detected, Windows continues without performing the faulting instruction. You might be able to save your work in a new file at this point, but do not continue using the application for other actions. We recommend you exit Windows after saving your file, then restart Windows.

If you ignore a fault, the same dialog box might appear again. If choosing Ignore a few times doesn't allow you to continue operation, choose Close to terminate that application, because that is the only option available.

When you choose Close, the General Protection Fault dialog box appears, with an explanation and some data that may be useful in tracking the source of the error. Save this information. Contact your application vendor to discuss the source of the application error.

Troubleshooting Multimedia

This section provides some tips for troubleshooting the multimedia accessories.

1. Your multimedia features don't work.

Flowchart Series 6 Multimedia Problems

Make sure your computer has installed the multimedia extensions to Windows by opening the Control Panel in Program Manager. The multimedia extensions add the following icons: Display, Drivers, Joystick, MIDI Mapper, and Screen Saver. If these icons aren't there, you must run Windows Setup to install the multimedia extensions.

Flowchart 6.5
Sound

2. Your application won't play audio.

- Make sure you have the speakers plugged into the correct port. With some configurations, it's easy to confuse the microphone port with the speaker port.
- Make sure that speakers are getting the power they need to operate. Some speakers require an external power source or batteries.
- Check the volume setting for your PC audio hardware. You might have it turned all the way down.
- Choose the Drivers setting in the Control Panel to make sure you have the correct audio driver installed and that the settings are configured according to the hardware manufacturer's documentation.

3. Audio is too loud or too soft.

If you can hear audio, but it's too loud or too soft, check the volume control on your PC hardware and make any necessary adjustments. Also check the batteries in your speakers to see if they need to be replaced.

4. You can't hear audio when running a non-Windows application that supports audio.

Several non-Windows applications support audio. If you try to run one of these applications in 386 enhanced mode, the following message might appear: "This application will not be able to use audio." This is because Windows 386 enhanced mode takes over the audio hardware and won't relinquish control to a non-Windows application. You can run the application with audio in two ways:

- Exit Windows and run the application from MS-DOS.
- Exit Windows, and restart Windows in standard mode by typing **win /s** at the command prompt. This will let you run the non-Windows application from Windows with audio intact.

5. Your PC doesn't recognize the CD-ROM drive.

Flowchart 2.4
Cannot Access
CD-ROM

- If you have an external drive, make sure the cable is securely attached and connected to the correct port. Also, some drives require an external terminator. Make sure this is installed if needed. See your CD-ROM drive documentation for details.
- Make sure the MSCDEX CD-ROM driver is installed. You should have installed this at the same time as you set up your CD-ROM drive. Contact your CD-ROM drive manufacturer to obtain this software.
- If you have installed MSCDEX correctly, check the PATH statement in your AUTOEXEC.BAT file to make sure the directory that holds the MSCDEX software is in the path.

6. Your CD drive can't read the CD disc correctly.

- Make sure you place the disc in its drive caddy before inserting it into the drive.
- Make sure you insert the disc with its label side up.
- If you upgraded your PC with an MPC upgrade kit, make sure you installed and configured the hardware according to the manufacturer's directions.
- Make sure to use the correct kind of disc. CD-ROM discs contain information and multimedia applications; CD audio discs contain music. You can't use File Manager or any other MS-DOS or Windows file commands on an audio disc.
- Even though discs are quite durable, they are plastic and can become damaged. Excessive heat can warp or even melt a disc. Scratches and fingerprints on the surface can cause problems. Clean a disc by wiping with a soft cloth from the center of the disc outward in a straight line. Do not use alcohol-based cleaning solutions.

7. Music Box doesn't play correctly.

If Music Box won't play a disc or plays very slowly, check the following:

Flowchart 6.4 Music Box

- Make sure you have a disc in the drive before starting Music Box. Make sure to place the disc in its drive caddy before inserting it into the drive.
- Make sure you're using the right kind of disc. Music Box can't play a disc that doesn't have audio.
- Make sure you're using the most current version of the CD-ROM driver software (MSCDEX). If you're using an older version, this can cause Music Box to play too slowly. Contact your CD-ROM drive manufacturer to obtain the latest version of MSCDEX.

8. Sound Recorder doesn't record.

Flowchart 6.6 Sound Recorder

- Make sure the microphone is turned on. And make sure the microphone is plugged into the correct port.
- Make sure you speak clearly and distinctly into the microphone.
- Make sure your PC has enough disk space to hold the recorded file.
- Make sure the output volume level of your audio hardware is set correctly. If you set it too low, you can see the audio recording, but you won't be able to hear it when you play it back.

Flowchart 6.2
Media Player

9. The Media Player won't play MIDI files.

- Make sure the correct MIDI driver is installed. Choose the Drivers setting in the Control Panel to see whether you have a MIDI driver set up and ready for use.
- Make sure you set Media Player to play MIDI files through the Device menu.
- Make sure you have the right MIDI mapping setup. If you added MIDI hardware, check the MIDI Mapper settings in Control Panel.

10. The Media Player won't play animation.

- Make sure you set Media Player to play animation files through the Device menu.
- Make sure the animation driver is installed. Choose the Drivers setting in Control Panel to see whether you have the Microsoft MCI Driver for MMP set up and ready for use.

11. The Media Player won't play audio files.

- Make sure the correct audio driver is installed. Choose the Drivers setting in Control Panel to see whether you have the correct audio driver set up and ready for use.
- Make sure you set Media Player to play audio files through the Device menu.
- Make sure the file you want to play is a .WAV file.

Disk Utilities



Chapter

14 The Windows Resource Kit Disk

This chapter describes the utilities, programs, drivers, and documentation provided on disk with the *Windows Resource Kit*. These utilities can also be downloaded from the Microsoft Product Support Download Service. For instructions on using this service, see "Online Services" in Appendix A, "Windows Resource Directory."

To install the Windows Resource Kit utilities:

1. Insert the Windows Resource Kit disk into a floppy drive.
2. Choose Run from the File menu in File Manager or Program Manager.
3. In the command-line box, type **a:setup** (if A is the drive that contains the disk) and press ENTER.

This installs the utilities in the directory of your choice, and also creates a Program Manager group called "Windows Resource Kit."

If you decide to remove any of these utilities, see the description of the individual utilities later in this chapter for the associated filenames.

Contents of this chapter

The Windows Resource Kit Utilities 440
- Graphics Viewer 440
- Network Assistant 440
- File Size Information 441
- SMARTDrive Monitor 441
- System Resource Monitor 442
- TopDesk 442

Other Windows Resource Kit Files 443
- README Files 443
- Windows Driver Library Files 443
- Network Support Files 444

The Windows Resource Kit Utilities

The *Windows Resource Kit* utilities range from applications for manipulating desktop wallpaper to File Manager extensions. The utilities include:

Graphics Viewer	SMARTDrive Monitor
Network Assistant	System Resource Monitor
File Size Information	TopDesk

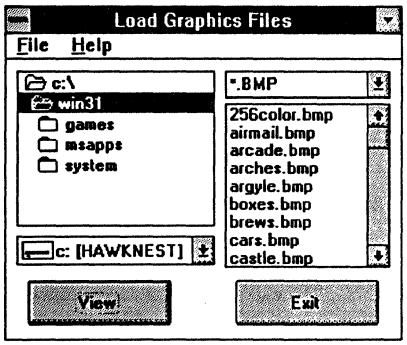


Graphics Viewer

Graphics Viewer (BMPVIEW.EXE) allows you to view graphics files that are in Bitmap, MetaFile, and Icon formats. Graphics Viewer also allows you to load any files in Bitmap format as the desktop wallpaper. To run Graphics Viewer, double-click its icon in the Windows Resource Kit group. When the Graphics Viewer is running, press F1 for information about its features.

Figure 14.1

Graphics Viewer





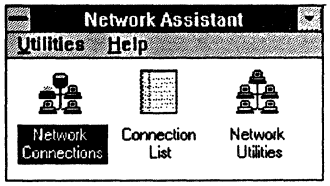
Network Assistant

Network Assistant (NETAPP.EXE) is an application for managing the network connections for both disk and printer devices. This utility has three parts that you can see by double-clicking the icon in the Network Assistant main window:

- Network Connections, for managing your disk and printer connections.
- Connection List, for maintaining a list of frequently used connections.
- Network Utilities, for viewing network specific information.

Figure 14.2

Network Assistant

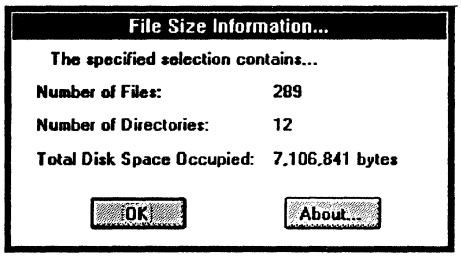


File Size Information

File Size Information (FILESIZE.DLL) is an extension to File Manager. After you install it, this extension adds a menu to File Manager named Info. Choose this menu for information about a selected item such as Number of Files, Number of Directories, and Total Disk Space. This is especially useful for viewing the disk space occupied by a selected parent directory and all its subdirectories.

Figure 14.3

File Size Information



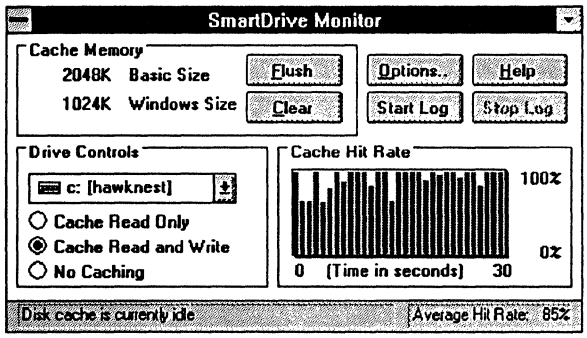


SMARTDrive Monitor

The SMARTDrive Monitor (SMARTMON.EXE and SMARTMON.HLP) provides an interface to the SMARTDrive 4.0 disk cache included with Windows 3.1. SMARTDrive Monitor displays a graph showing Cache Hit Rate. Each column in this graph represents the percentage of "hits" on the cache during the specified sampling frequency (the default is 500 msec). A hit occurs each time Windows looks to SMARTDrive for information and finds it in the cache. The SMARTDrive Monitor also displays the Hit Rate when minimized. To run this application, double-click its icon in the Windows Resource Kit group. When SMARTDrive Monitor is running, press F1 for information about its features.

Figure 14.4

SMARTDrive Monitor



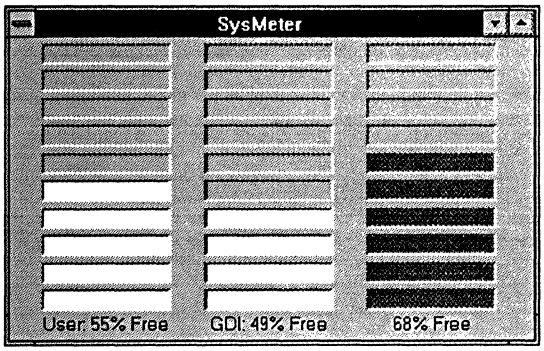


System Resource Monitor

The System Resource Monitor (SYSMETER.EXE and SYSMETER.HLP) displays the current use of the USER heap, the GDI heap, and system memory. The application displays three bars that represent the percent-free of each of the three monitored areas. To run this application, double-click its icon in the Windows Resource Kit group. This utility also provides information while minimized. When System Resource Monitor is running, press F1 for information about its features.

Figure 14.5

System Resource Monitor



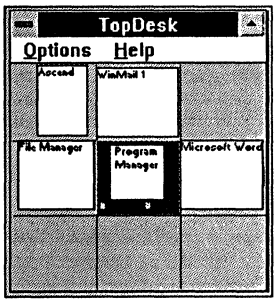


TopDesk

TopDesk (TOPDESK.EXE and TOPDESK.HLP) presents a small representation of the virtual desktop, showing your current desktop, the home desktop, all visible windows, and, optionally, all ghost windows. TopDesk lets you manipulate all of these objects with various keyboard and mouse actions. The model it presents gives you a clean, intuitive way to switch between several applications while conserving precious screen space. To run this application, double-click its icon in the Windows Resource Kit group. When TopDesk is running, press F1 for information about its features.

Figure 14.6

TopDesk



Other Windows Resource Kit Files

The other files on the Windows Resource Kit disk include README files, selections from the Windows Driver Library, and network support files.

README Files

Several document files are copied during the installation of the utilities. These README files include additional information about Windows 3.1. You can read (and print) the files with Windows Write. These files include:

README file	Description
SYSINI.WRI	Procedures for modifying settings in the SYSTEM.INI file.
WININI.WRI	Procedures for modifying settings in the WIN.INI file.
NETWORKS.WRI	Information about running Windows with specific network configurations.
PRINTERS.WRI	Information about configuring particular printers and fonts for the Windows environment.
IPX.WRI	Procedures for running the Microsoft IPX upgrade utility.
README.WRI	Information that became available after the *Windows Resource Kit* was printed.

Windows Driver Library Files

The Windows Resource Kit disk includes a selection of files from the Windows Driver Library (WDL). These drivers were released with the *Windows Resource Kit* on April 6, 1992.

The complete Microsoft Windows Driver Library contains device drivers made available after Windows version 3.1 was released. Device drivers are added to the WDL as they become available. If you have a modem, you can download the WDL device drivers from network services including CompuServe, GEnie, Microsoft OnLine, various user-group bulletin-board services (BBSs on the APCUG network), and the electronic downloading service maintained by Microsoft (206 637-9009). If you do not have a modem, you can obtain disks by calling Microsoft Customer Support Services at 800 426-9400.

If your copy of Windows was acquired outside the United States, the package contains a subsidiary information card listing bulletin-board services (BBSs) from which you can obtain updated device drivers. The card also lists local subsidiary telephone numbers you can call to obtain drivers if no BBS is available or you cannot access a BBS.

The following sound drivers are included on the Windows Resource Kit disk.

Source	Device driver from the Windows Driver Library
Creative Labs	SoundBlaster Pro
Media Vision	Pro Audio Spectrum
Artisoft	Sounding Board
IBM	Multimedia Audio Board
CompuAdd	Multimedia Board

To install a sound driver file:

1. Insert the Windows Resource Kit disk in your floppy drive.
2. Run Control Panel, and double-click the Drivers icon.
3. Click Add and choose Unlisted or Updated Driver.
4. Select the driver you want from the list and press ENTER.

Network Support Files

The Microsoft IPX Upgrade utility assists network administrators who need to update the Novell NetWare version on their client workstations in order to run Windows 3.1. The Windows 3.1 components from Novell require the workstation to be running recent versions of the NetWare client software, including the NetWare shell, IPX, and (if you are running ODI-model) LSL. Updated versions of these components are included with Windows 3.1 and are copied to the user's Windows SYSTEM subdirectory when Setup detects NetWare. The files are also copied to a shared WINDOWS directory during Administrative Setup (**setup /a**). For details about upgrading workstations, see NETWORKS.WRI on the Windows Resource Kit disk.

The NetWare shell and the ODI-model IPXODI.COM and LSL.COM files are easy to install. However, to upgrade a dedicated-model IPX.COM, the administrator must use a set of NetWare utilities to link a new version of IPX.COM and then manually configure the hardware options. The Microsoft IPX Upgrade utility automates this time-consuming process, allowing new versions of IPX.COM to be built and properly configured without the administrator's intervention. The utility can be run by the user, by an administrator, or from the user's logon script or AUTOEXEC.BAT file. The utility can also be run from a shared network drive or from a floppy disk.

For details about using the Microsoft IPX Upgrade utility, see the IPX.WRI file on the Windows Resource Kit disk.

References, Resources, and Appendixes



Part 7

References, Resources, and Appendixes

Glossary 447

Appendix A Windows Resource Directory 457

Appendix B Hardware Compatibility List 487

Appendix C Windows 3.1 Disks and Files 499

Appendix D Articles 505

Windows for Pens 506
QEMM-386 and Windows 3.1 515
PC-NFS and Windows 3.1 519
FastDisk: An Introduction to 32-Bit Disk Access 520

Index 525

Glossary

Other glossary resources

- The Glossary in the *Windows User's Guide* for terms related to Microsoft Windows.
- The Glossary in Windows online help for definitions of terms related to the screen elements in Microsoft Windows.
- The *Computer Dictionary*, Microsoft Press, 1991, for definitions of general computing terms.

16-bit An adjective that describes systems and software that handle information in words that are 2 bytes (16 bits) wide.

32-bit An adjective that describes hardware or software that manages data, program code, and program address information in 32-bit-wide words.

386MAX.SYS A 386 expanded memory manager from Qualitas Corporation.

A

adapter segment A name sometimes used for the upper memory area, at hexadecimal addresses A000 through EFFF (640K to 1024K).

API An abbreviation for application programming interface, which is a set of routines that an application program uses to request and carry out lower-level services performed by the computer's operating system. For Windows, the API also helps applications manage windows, menus, icons, and other GUI elements. For a local area network, an API provides applications with routines for requesting services from lower levels of the network.

application Software such as a word processor or spreadsheet that you use to perform a specific type of work.

attribute A characteristic or property; for example, the color of a line, the length of a data field, or a typeface characteristic such as size.

B

backfilling To designate memory on an expanded memory card and make it available for use as conventional memory.

bank switching A technique for sequentially reading page frames in expanded memory. The 64K page frames are brought into the upper memory block from an expanded memory card, then read and swapped back to the card so that the next 64K page frame can be brought in.

Bezier curve A mathematical construct for describing a curve, used in CAD and illustration software to draw complex graphic shapes.

BIOS An acronym for basic input/output system, which is software (usually contained in ROM) that supports the transfer of information between system elements such as memory, disks, and the monitor.

bitmap A sequential collection of bits that represents, in memory, an image to be displayed on the screen.

bridge For a LAN, a device that links a network to another of the same type.

buffering The process of using buffers to hold data being moved to or from I/O devices such as serial ports and disk drives.

buffers A reserved part of memory where data is held temporarily until the data is transferred from storage to another location in memory. Some printers have their own buffers.

C

CEMM.EXE A 386 expanded memory manager provided with all Compaq 386 computers.

character-based interface The traditional user interface common to non-Windows applications, where all screen images are made up of text characters.

client A stand-alone workstation that uses the services of any type of server on a network.

client application Any Windows application that can accept, display, and store Object Linking and Embedding (OLE) objects.

client/server computing A mode of distributed network computing in which an application is executed cooperatively by two computers. The client component of the application (the front end) executes on one node, while the server component (the back end) runs on a second node. For PCs, communications between the nodes is implemented most frequently using the named pipes interprocess communication (IPC) protocol. Client/server computing allows more effective use of computing resources, higher performance, greater flexibility, simpler upgrades, and (for some applications) greater reliability and data integrity.

clock speed The rate in which the computer clock oscillates, usually given in MHz (one million cycles per second). The computer's clock rate is a prime determinant of overall processing speed. Typical clock speeds for AT-compatibles are 6, 8, 10, 12, 16, and 20 MHz; for 80386 machines, typical speeds are 16, 20, 25, and 33 MHz.

CMOS RAM A RAM chip that uses the complementary metal-oxide semiconductor process, which provides high speed and extremely low power consumption, making the chip useful for hardware such as the microcomputer clock.

code page For MS-DOS version 3.3 and later, a table that relates the keys on the keyboard to the characters displayed on the screen, thereby providing support for the character sets and keyboard layouts used in different countries. Code page 437 is the code page for the United States.

command interface A user interface that requires the user to enter commands at the command prompt. Also called command-line interface.

command prompt A screen symbol such as *C:>* that requests user input. Also called the MS-DOS prompt.

compound document A document file that contains embedded and linked data that was created in other kinds of applications.

context switching The kind of multitasking that is done in standard mode Windows, where the CPU switches from one task to another, rather than allocating time to each task in turn, as in *timeslicing.*

conventional memory The physical memory in a PC from 0K to 640K.

CPU An abbreviation for the computer's central processing unit.

cut-and-paste A method of moving data into another document or file, which could be for a different but compatible application.

D

database server Software such as Microsoft SQL Server that provides high-performance database access by splitting the DBMS function into a front-end component (where data is manipulated by users or applications) and a database-intelligent, back-end component (where data is stored, retrieved, and managed). In PC networks, the front-end component often resides on a personal computer controlled by a single user, while the back-end component resides on a high-performance PC that services requests for data submitted over the network by users.

DBMS An abbreviation for database management system.

DDE See *dynamic data exchange.*

demand paging The common implementation of virtual memory, where pages of data are read into memory from storage in response to page faults. See also *paging* and *virtual memory.*

device driver The software that the system uses to communicate with a device such as a display, printer, mouse, or communications adapter.

discardable code segment A Windows application code segment in memory, which can be discarded, overwritten, or reloaded from disk when necessary.

disk cache A part of RAM that is set aside to temporarily hold data read from disk. A disk cache doesn't have to hold an entire file, as a RAM disk does, but can hold parts of running application software or parts of a data file. Disk-caching software such as SMARTDrive manages the process of swapping data to and from the disk cache.

DLL See *dynamic-link library.*

DMA An abbreviation for Direct Memory Access, which is memory access that doesn't use the microprocessor, often used to transfer data directly between memory and a peripheral device such as a hard disk drive.

DOS An acronym for *disk operating system,* which is the operating system software that is loaded from a disk when you boot your computer. See also *MS-DOS.*

DOS Extender See *extended memory manager.*

double buffering The use of two buffers rather than one to temporarily hold data being moved to and from an I/O device. Double buffering increases data transfer speed because one buffer can be filled while the other is being emptied.

downloadable fonts Fonts that reside on your hard disk and that are sent to the printer as needed.

DPMI An abbreviation for DOS Protected Mode Interface, an industry standard that allows MS-DOS applications to execute code in the protected operating mode of the 80286 or 80386 processor. The DPMI specification is available from Intel Corporation.

dynamic data exchange (DDE) The form of interprocess communications (IPC) used by Microsoft Windows to support exchange of commands and data between two applications running simultaneously. In Windows 3.1, this capability is enhanced with *Object Linking and Embedding.*

dynamic-link library (DLL) An executable code module for Microsoft Windows that can be loaded on demand and linked at run time, and then unloaded when the code is no longer needed.

E

EISA Acronym for Extended Industry Standard Architecture, a bus standard introduced in 1988 by a consortium made up of AST Research, Compaq, Epson, Hewlett-Packard, NEC, Olivetti, Tandy, Wyse, and Zenith.

EMM See *expanded memory manager.*

EMM386.EXE The Microsoft 386 expanded memory manager.

EMS See *Expanded Memory Specification.*

encapsulation In object-oriented programming, the grouping of data and the code that manipulates it into a single object. If a change is made to an object class, all instances of that class (that is, all objects) are changed. Encapsulation is one of the benefits of object-oriented programming.

expanded memory Memory provided by a physical add-in memory card or through a 386 expanded memory manager. Expanded memory is made available to application software as 16K pages, mapped into a 64K page frame.

expanded memory manager (EMM) A driver that provides a software interface to expanded memory. An expanded memory manager is specific to a particular physical expanded memory board or uses the 80386 processor to emulate expanded memory in extended memory.

Expanded Memory Specification (EMS) A description of a technique for adding memory to PCs to bypass the 640K limit on addressable memory. The EMS supports additional memory through a hardware expansion board and an expanded memory manager. To use expanded memory, an application must be specifically designed to interact with an expanded memory

manager. There are two expanded memory specifications: LIM 3.2 provides for a maximum of four contiguous 16K bankable pages; LIM 4.0 provides for a maximum of sixty-four 16K bankable pages, which do not have to be contiguous.

extended memory The memory at physical addresses from 1024K that can be accessed by a 80x86 CPU running in protected mode. Windows version 3.0 and later take full advantage of extended memory, including all memory above 640K, when running in standard mode or 386 enhanced mode.

extended memory manager (XMM) A driver such as HIMEM.SYS that is necessary to access extended memory. Applications do not have to be customized to interact with an extended memory manager to use extended memory under Windows 3.x. In some applications, the extended memory manager is referred to as an MS-DOS Extender.

Extended Memory Specification (XMS) A Microsoft standard that provides a way for real mode applications to access extended memory. Non-Windows applications running under Windows 3.x can use the XMS to access extended memory. The XMS document and sample source code is available from the Microsoft Information Center.

F

fault tolerance The ability of a system to recover from an error, a failure, or a change in environmental conditions (such as loss of power). True fault tolerance provides for fully automatic recovery without disruption of user tasks or files, in contrast to manual means of recovery such as restoring data loss with tape backup files.

file server A network node that usually consists of fixed-disk storage and a CPU, for storing programs and data shared by users on a network. File servers offer operating system-type file and print capabilities, while database servers provide database intelligence such as transaction processing, indexing, logging, security, and so on.

font In Windows, a complete set of characters for one style of a specific typeface, including all the letters, numbers, and punctuation marks. For example, Courier New is a font.

font family In Windows, a group of typefaces with similar characteristics. The families that Windows recognizes for font installation are Roman, Swiss, Modern, Script, and Decorative. For example, Arial, Arial Bold, Arial Bold Italic, Arial Italic, Small Fonts, and MS Sans Serif are all part of the sans serif Swiss font family.

G

grabber Software that supports exchange of data in video memory between non-Windows applications and Windows 3.x.

graphical user interface (GUI) An interface that allows users to choose commands and other options by pointing to a graphical icon and then activating the choice with either the keyboard or a mouse.

group In Windows, a collection of applications in the Program Manager window.

GUI See *graphical user interface*

H

high memory area (HMA) The first 64K of extended memory, into which MS-DOS 5.0, drivers and utilities can be loaded to save conventional memory.

I

independent software vendor (ISV) A participant in the computing industry marketplace whose primary focus is development of software.

interface 1) The user interface, where people communicate with the software through commands and other devices. 2) The connections in software that allow an application to work with the operating system, or that allow the operating system to work with the hardware. 3) The hardware cards, plugs, and other devices used to move data from place to place.

interrupts A request-for-attention signal sent by either hardware or software to the CPU that causes the CPU to suspend some operations and transfer control to an interrupt handler (designated, for example, as Int 13).

I/O An abbreviation for input/output.

IRQ Abbreviation for "interrupt request lines," which are hardware lines over which I/O devices, keyboards, and disk drives can send interrupts to the CPU. The IRQs are built into the hardware, with preassigned priority levels.

ISA Abbreviation for Industry Standard Architecture, an unofficial designation for the bus design of the IBM PC/XT.

ISO Abbreviation for International Standards Organization, an international association of standard-setting organizations.

ISO/OSI model The International Standards Organization Open Systems Interconnection model, which standardizes levels of services and types of interaction for exchanging data through a communications network. The model separates computer-to-computer communications into seven layers: the application layer, data-link layer, network layer, physical layer, presentation layer, session layer, and transport layer.

K

kernel 1) The part of an operating system that performs basic functions such as allocating hardware resources. 2) The central part of other system software. 3) One of the core dynamic-link libraries that make up Windows.

keyword A predefined word in a computer or command language.

L

LAN See *local area network.*

LAN Manager The Microsoft networking environment.

LIM (Lotus/Intel/Microsoft) 3.2 and 4.0 See *Expanded Memory Specification.*

local area network (LAN) Two or more computing units connected for local resource sharing, or a network in which communications are limited to an area such as a single office building and do not extend across public rights-of-way. A LAN allows PCs to have access to common data and peripherals, and it typically consists of PCs with adapter cards, file servers, printers, gateways to departmental or corporate computers, and network software to integrate these components. More sophisticated LANs permit sharing of computing resources, along with sharing printers and file storage facilities. For example, networks equipped with MS LAN Manager support *client/server computing.*

local heap A memory storage area limited to 64K in size.

logging on The procedure that links a user to a network.

M

MCI An abbreviation for Media Control Interface, a standard control interface for multimedia devices and files.

memory-mapped files An operating system feature that maps a file into a program's address space, allowing an application to read and modify a file's content as if the file were an in-memory data structure. Memory-mapped files use demand paging to allow applications to access and manipulate files regardless of how big the files are. Without memory-mapped files, every application that works with large files must include code to handle the movement of data between the disk and the main memory.

memory protection Strict management of system memory to prevent the corruption of any application's program code or data by interference from another application. Memory protection ensures that concurrent applications are compatible and prevents them from interfering with system processes.

Microsoft Consulting Services (MCS) The Microsoft consulting service, founded in 1990 to help clients develop mission-critical business systems and to work with major systems integrators to provide specialized expertise in PC technologies in cooperative processing environments.

MIPS An acronym for millions of instructions per second, which is a measure of processing speed that refers to the average number of machine language instructions performed by the CPU in one second. A typical Intel 80386-based PC is a 3 to 5 MIPS machine, whereas an IBM System 370 mainframe typically delivers between 5 and 40 MIPS. MIPS measures raw CPU performance, but not overall system performance.

movable code segment A Windows application code segment that can be moved around in memory.

MS-DOS Microsoft Disk Operating System, the standard operating system for computers equipped with the Intel 8086 and 8088 microprocessors, and for 80286 and 80386 microprocessors running in 8086 mode.

MS-Net Microsoft Network, the MS-DOS networking systems software product shipped in 1985. Largely superseded by Microsoft LAN Manager.

multitasking The concurrent management of two or more distinct tasks by a computer. Although a computer with a single processing unit can only execute one application's code at a given moment, a multitasking operating system can load and manage the execution of multiple applications, allocating processing cycles to each in sequence. Because of the processing speed of computers, the apparent result is the simultaneous processing of multiple tasks. Standard mode Windows performs multitasking only in the form of context switching; 386 enhanced mode allows multitasking in the form of timeslicing.

N

NetBIOS An API used by application programs on a PC LAN that uses MS-DOS or some version of UNIX, providing application developers with a uniform set of commands for requesting lower-level network services.

NetWare A local area network operating system produced by Novell, Inc. Several families of NetWare products exist, including NetWare 386.

network A configuration of data processing devices and software connected for information interchange.

network application A network program, or combination of a program and data, that performs a task (typically involving two or more computers).

non-Windows application A term used in this document as a shorthand term to refer to an application that is designed to run with MS-DOS but not specifically with Windows and that may not be able to take full advantage of all Windows features (such as memory management).

O

object Encapsulated data in a compound document. For example, a single cell, a range of cells, or an entire spreadsheet could be embedded as an object in a word processing text file. Any data can be an object if it was created in a Windows application that uses OLE.

object handler A dynamic-link library for an OLE server application. This DLL acts as an intermediary between the client and server applications; for example, to improve performance when it's necessary to redraw an object in the window of the client application.

Object Linking and Embedding (OLE) An enhancement to DDE protocol that allows you to embed or link data created in one application in a document created in another application, and subsequently edit that data in the original application without leaving the compound document.

OEM An abbreviation for original equipment manufacturer.

Open Software Foundation (OSF) A consortium of computing industry vendors organized to develop an open, UNIX-based operating system standard. OSF's first operating system release,

called OSF/1, is built on the IBM UNIX-based AIX kernel and the Motif graphical user interface.

Open Systems Interconnect (OSI) An ISO publication that defines seven independent layers of communication protocols. Each layer enhances the communication services of the layer just below it and shields the layer above it from the implementation details of the lower layer. In theory, this allows communication systems to be built from independently developed layers.

operating system The software that governs the operation of a computer, providing services such as resource allocation, scheduling, input/output control, and data management. Popular operating systems include MS-DOS and UNIX.

OSI See *Open Systems Interconnect.*

P

package In a compound document, an embedded icon that contains an object, a file or part of a file, or a command.

page fault An interrupt that results when software requests a memory location that is being saved for auxiliary storage.

page frame A physical address in the upper memory area where a page of expanded memory can be mapped. The page frame itself can be a maximum of four 16K pages.

paging A technique for implementing virtual memory, where the virtual address space is divided into fixed-sized blocks called pages, each of which can be mapped onto any physical addresses available on the system.

palette management A mechanism to choose which colors to display when the display device supports a limited number of colors for simultaneous display. For example, a VGA monitor can display 16 colors out of an overall palette of 256, so a DIB bitmap with more than 16 colors would have to be palettized.

paste A function that copies the contents of the Clipboard and inserts it at the current cursor position.

path 1) In MS-DOS, the route the operating system follows to locate files on a disk or disks. 2) On a network, the route between any two nodes.

PC-NET The IBM PC LAN program, an MS-DOS-based networking systems software product developed by Microsoft and IBM, which has been shipping since 1985.

PIF An abbreviation for program information file, which contains information about a non-Windows application, such as how much memory it needs. Windows uses this information to run the application in the most efficient way.

pipe 1) A portion of memory used to pass data from one process to another, symbolized in MS-DOS by the | character. For example **dir | sort** sends the output of the directory listing to the **sort** command. 2) In a sample statement for a Windows initialization file or in an MS-DOS command, the symbol used to show that only one of the possible values can be used in the statement.

pitch Denotes the horizontal size of a fixed-width font in characters per inch.

point size The vertical size of a font. A point is about 1/72 inch.

POSIX A set of software standards being developed by the IEEE POSIX Working Group, to allow applications to be written to a single target environment in which they can run unchanged on a variety of systems. The POSIX specifications define characteristics for operating systems, DBMS, data interchange, network services, user interface, and programming interfaces. Several vendors plan to make their operating systems POSIX-compliant.

print server A network node, usually consisting of fixed-disk storage and a CPU, that controls one or more printers that can be shared by users.

printable screen fonts Windows screen fonts that can be translated for output on the printer.

program group A set of several programs whose icons can be acted on as a single entity in Windows Program Manager. For example, if icons are associated with a single group icon, clicking the program group icon causes the icons for each of the programs in the group to be displayed.

protected mode An operating mode for the Intel 80286 and higher processors that supports multitasking, data security, and virtual memory.

provider interface An API for the provision of centralized administrative support of the client workstation over the network. Providers of network services will use this Microsoft Windows API, which will be fully open to the development community.

Q

QEMM386 A 386 expanded memory manager from Quarterdeck Corporation.

query 1) A specific request for data or instructions. 2) A request for information from a database using specific conditions; for example, a request for a list of all customers with balances greater than $1,000.

Query-By-Example (QBE) A program product developed by IBM to work in conjunction with DB2, a mainframe relational DBMS. QBE is used to write queries graphically.

R

raster fonts A set of characters for screen display or printing that are stored as bitmaps in specific character sizes.

registration database The REG.DAT file, which provides the information used by File Manager and Print Manager to support drag-and-drop and other actions, and which is used by OLE applications to find file types and OLE capabilities for embedded objects.

relational database 1) A database organized and accessed according to relationships between data items. 2) A data structure perceived by its users as a collection of tables. A relational database consists of tables, rows, and columns. Typical examples are dBase IV from Borland and R:Base 5000 from Microrim. Relational databases differ from nonrelational databases in that there are no system dependencies stored within the data. Hierarchical databases are not relational, because they contain pointers to other data.

resource 1) Any facility of a computing system or operating system required by a job or task, including memory, input/output devices, processing unit, data files, and control or processing programs. 2) A network component such as a file, printer, or serial device that is shared by other components of the network. 3) In Microsoft Windows, the definition of elements such as fonts, templates, accelerators, and error messages are all stored in resource files.

ring A network configuration where a series of attached devices are connected by unidirectional transmission links to form a closed path.

routine A program, or a sequence of instructions called by a program, that has some general or frequent use.

S

scalability The ability of a computing element such as a process, processor, or structure to grow seamlessly.

SDK The Windows Software Development Kit.

server On a local area network, the computer that runs the administrative software to control access to the network. The server makes network resources available to the workstations. See also *file server* and *print server.*

server application 1) On a network, a process or program that runs on a server, typically as a network resource; for example, a communications program that organizes and allocates network communications resources. More sophisticated server programs, known as client/server applications, run interactively with an application resident on network nodes. See also *client/server computing.* 2) In a compound document, any Windows application that allows you to edit an

object when Windows informs it that you selected the object in an OLE client application.

shadow ROM A process used in many 386 machines to map ROM BIOS activities into faster 32-bit RAM memory. Shadow memory must be loaded with BIOS routines each time the computer boots.

shared resource A printer, file, or serial computer communications device made available through the LAN to multiple computers not physically attached to the resource.

source code A set of programming language instructions that must be translated to machine instructions before the program can run.

spooler A program that intercepts the data going to a device driver and writes it to the disk. The data is later printed or plotted when the required device is available. A spooler prevents the intermixing of output from different sources.

Structured Query Language (SQL) A database query and programming sublanguage. SQL (commonly pronounced "sequel" and originally developed for IBM mainframe computers) is an established set of statements used to add, delete, or update information in a table, or request information from one or more tables in the form of a report. There is now an ANSI-standard SQL definition for all computer systems.

system administrator The person responsible for planning computer or network installation and for ensuring the successful installation and use of the system by other users.

T

TCP/IP See *Transport Control Protocol/Internet Protocol.*

timeslicing A method of multitasking that allocates CPU attention to tasks in fractions of a second, making it appear to users that the multitasking operations are happening simultaneously. Tasks are either assigned priority levels or processed in sequence to maintain order.

Transport Control Protocol/Internet Protocol (TCP/IP) A set of network protocols that has become an industry standard in engineering, government, and educational local area network environments. The TCP/IP protocol family includes transport, file transfer, terminal emulator, messaging, and network management definitions.

transparent application integration The ability to manipulate information with a variety of applications within a single document, without moving from one application to another to produce the desired results. This is implemented as *Object Linking and Embedding* in Windows 3.1.

TSR Terminate-and-stay-resident software program.

U

upper memory area The part of physical memory that starts at 640K and ends at 1024K (hexadecimal addresses A000 through FFFF). Also called the *adapter segment,* because that portion of memory is often used by hardware adapters such as display adapters.

upper memory blocks (UMB) The unused parts of the upper memory area. For an 80386 or higher computer, information can be mapped (or copied) from another type of memory to upper memory blocks, freeing more conventional memory.

user context A user session created by an operating system in response to a logon request, and typically characterized by privilege sets that strictly define the user's authority to access system resources and information. Contexts restrict unauthorized access to facilities and data and protect the system itself from user and applications interference, accidental or otherwise. Contexts are a feature of most multiuser operating systems, usually integrated with the security system.

V

VAR Value-added reseller. This term often refers to a reseller of computer products who also provides both integration and software customization services to business clients.

VCPI An abbreviation for Virtual Control Program Interface. An 80386 memory management standard created by Phar Lap Software in conjunction with other software developers to provide an interface between applications using MS-DOS Extenders on an 80386 machine and applications using 386 expanded memory managers. For example, the 386 expanded memory managers QEMM, 386MAX, and CEMM support the VCPI specification.

vector fonts A set of lines that connect points to form characters.

virtual device A device that software can refer to but that doesn't physically exist.

virtual hard drive memory factor The available space on a hard drive partition that Windows can address as physical memory.

virtual machine (VM) Software that mimics the performance of a hardware device. For Intel 80386 and higher processors, a virtual machine is protected memory space that is created through the processor's hardware capabilities.

virtual memory A technique of simulating additional memory for an application to use. Applications access the memory through virtual addresses, which are mapped onto physical addresses on the disk. Two common ways to implement virtual memory are *paging* and segmentation (which was used by 80286 and earlier Intel processors to address memory).

Virtual Memory Manager (VMM) The facility in Windows 386 enhanced mode that manages the task of swapping data in and out of virtual memory for multiple non-Windows applications running virtual machines. The Windows VMM can manage paging in up to 48 MB of virtual memory.

visual programming The use of graphical development tools and visual metaphors to create software.

VMM An abbreviation for Virtual Memory Manager.

W

wide area network (WAN) A network that provides data communication capability in areas larger than those serviced by a local area network. A WAN typically relies on serial line protocols to interconnect subnetworks.

Windows application A term used in this document as a shorthand term to refer to an application that is designed to run with Windows and does not run without Windows. All Windows applications follow similar conventions for arrangement of menus, style of dialog boxes, and keyboard and mouse use.

workstation A terminal or personal computer, usually connected to a mainframe or to a network, on which a user can run applications. A workstation generally cannot share its resources with other network nodes or a host computer. However, on a LAN Manager network, an OS/2 workstation can share resources by invoking the peer server option. See also *server*.

X

X.25 A standard network communications protocol for implementing the internetwork communications function specified by the third network layer of the ISO/OSI model. X.25 is a packet-switched protocol often used in wide-area network configurations. MS LAN Manager is supported by X.25 communications packages provided by various third-party vendors.

XMM See *extended memory manager.*

XMS See *Extended Memory Specification.*

Appendix

A Windows Resource Directory

This appendix contains a list of resources to support learning and using Microsoft Windows. With Windows becoming accepted as a standard for PC computing, many companies are making products and services available to enhance the Windows experience. Microsoft also offers a wide array of support services.

Because there so many companies providing Windows-related products and services, it is inevitable that we have missed a resource that you know to be of value. If this is the case, please contact us so we can include it in future updates of the Directory. You can fax them to us at 206/883-8101, or send suggestions to:

Windows Resource Directory Updates, 3/2
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-0399

Contents of the Resource Directory

Microsoft Support Services 458
Windows for Foreign Languages 458
Microsoft Consulting Services 458
Microsoft Consultant Relations Program 458
Microsoft Certified Professional Program 459
Microsoft Authorized Training Centers 460
Independent Training Services 466
Support Coordinator Training 467
Disks, Tapes, and Videos Training 467
Microsoft University 468
Online Services 472
User Groups 479
Hardware Support 482
Reference Materials 483

Microsoft Support Services

The following numbers can be called for assistance with using Microsoft Windows.

General Support

Microsoft
206/637-7098
Windows 3.0 support

Microsoft
206/637-7098
Windows Environment

Applications Support

Microsoft
206/635-7070
Excel for Windows support

Microsoft
206/635-7145
PowerPoint for Windows support

Microsoft
206/646-5100
WinStart: for corporate named accounts

Microsoft
206/462-WORD
Word for Windows support

Microsoft
206/635-7130
Works for Windows support

Windows for Foreign Languages

Windows is currently available in Chinese, Danish, Dutch, Finnish, French, German, Hanguel, Italian, Japanese, Norwegian, Portuguese, Spanish, and Swedish. Microsoft has also scheduled with the release of Windows 3.1 Arabic, Czech, Hebrew, Hungarian, Russian, Thai, Turkish, Bahasa Malaysia, Bahasa Indonesia, and Polish.

Microsoft Consulting Services

Microsoft Consulting Services (MCS) is aimed at helping you develop and manage a technology plan that gets your company off to a fast start. Our experienced staff helps with planning, management education, training—everything you need to successfully implement new technology. MCS helps to design and develop custom systems and to train thousands of employees.

MCS Australia
PO Box 91
North Ryde, New South Wales 2113
Australia
61/2/870-2200
61/2/805-0979 (fax)

MCS Boston
9 Hillside Ave
Waltham, MA 02154
617/487-6400
617/487-7925 (fax)

MCS England
Microsoft Place
Winnersh,
Wokingham, Berkshire, RG11 5TP
United Kingdom
44/734 270001
44/734 270002 (fax)

MCS Germany
Edisonstr. 1
D-8044 Unterschleissheim
Munich, Germany
49/89/317-050
49/89/317-05100 (fax)

MCS San Francisco
One Market Plaza, Suite 1550
Spear Street Tower
San Francisco, CA 94105
415/905-0235
415/905-0236 (fax)

Microsoft Consultant Relations Program

The Microsoft Consultant Relations Program (CRP) is designed for independent microcomputer consultants to help maximize the success of their business. In the program there are Associate consultants and Partner consultants. To qualify as an Associate CRP consultant you must first pass the Microsoft Certified Professional exams for MS-DOS/MicroHardware and Microsoft Windows, and submit an application. To qualify as a "Partner," you must first be an Associate and then take an application specific exam (for example, MS Excel and MS Word for Windows), submit sample code, references, and an application.

As an "Associate," you will receive substantial discounts on selected Microsoft products, a quarterly newsletter, a private CompuServe forum, and priority access to Microsoft technical support. As a "Partner," you also receive business referrals, beta testing opportunities, a regular monthly CD-ROM disk containing selected products and the contents of the Microsoft technical support database, and a license to use the "Microsoft Consultant Relations Partner" logo. For more information, call 800/227-4679.

If you are a customer in need of assistance from a Microsoft Consultant Relations Partner, please call 800/227-4679 and ask for the CRP Referral Specialist.

Microsoft Certified Professional Program

If you're a support coordinator, support engineer, systems engineer, reseller sales support engineer, consultant, network administrator, or anyone else who must gain and display technical expertise in Microsoft Windows, you should become a Microsoft Windows Certified Professional. Current certification — in this case for Microsoft Windows version 3.1 — demonstrates that you are the source for the latest information about Microsoft Windows. When a new version of Windows is released, you will be notified and you will be given the allotted time to complete any new requirements to maintain current certification status.

Upon successful completion of the Microsoft Windows Certified Professional requirements, you will receive a Welcome Kit that contains a Membership Card, Product Certification Certificate, CompuServe Intro Kit, product specific newsletter, a listing in the Microsoft Certified Professional Directory, and more.

To obtain the latest Microsoft Certified Professional brochure or if you have any questions about the Microsoft Certified Professional Program, please call Microsoft Insides Sales — System Software at 800/227-4679.

Microsoft Windows Operating System Version 3.1 Administration Exam Topics

Installation and Setup
MS-DOS Mode and Windows Mode installation events
Customizing Setup
Express and Custom Setup
SYSTEM.INI, WIN.INI, and PROGMAN.INI

The Windows Files
Windows core components
Dynamic-link libraries
Files required for Windows to run in standard mode
Files required for Windows to run in 386 enhanced mode

Memory Management
Conventional, expanded, extended, and virtual memory
Loading MS-DOS into the HMA and advantages of MS-DOS 5
UMB: characteristics, loading device drivers
Standard mode: characteristics, minimum memory and hardware requirements
386 enhanced mode: characteristics, minimum memory and hardware requirements
Using conventional, expanded, extended, and virtual memory
Expanded memory LIM 3.2 and 4.0 specifications
Swap files
Temporary and permanent swap files
HIMEM.SYS in Windows
EMM386.EXE in Windows
SMARTDRV.EXE in Windows
Windows system resources

Fonts and Printing
Typefaces and fonts
Windows screen font and a Windows printer font
Raster and vector screen fonts
TrueType fonts and standard fonts
Types of Windows printer fonts
LPT*x*.DOS port
Supporting COM3 and COM4 with Windows

Non-Windows Applications and PIFs
PIFs and PIF editor
Levels of non-Windows application support provided in standard mode and 386 enhanced modes
_DEFAULT.PIF

Networks
Installing and sharing Windows on a network
Setting up a Windows workstation with Windows installed locally
Setting up a diskless Windows workstation
Locations of each of the key files if running Windows from an attached file server
Advantages and disadvantages of setting up a network with Windows installed locally or from a file server
Network persistence feature of Windows
Winnet API
Network drivers and virtual network driver

Microsoft Authorized Training Centers

Microsoft Authorized Training Centers offer the most up-to-date Microsoft product information, courseware, and instructional techniques. Accreditation and support from Microsoft is your assurance of training that meets the highest professional standards. In addition to user training, most Authorized Training Centers offer a broad range of services, including customized training, train-the-trainer courses, courseware development, consulting services, and other out-of-classroom support.

To maximize the powerful productivity of Microsoft software, contact your nearby Microsoft Authorized Training Center by calling 800/426-9400.

ATC Training Services and Products

The following is a list of Authorized Training Centers (ATCs). This list is current as of February 10, 1992. For an up-to-date listing of the Microsoft third-party training organizations, please call Microsoft Sales and Services at 800/227-4679.

Alabama

ExecuTrain of Birmingham
2 Chase Corporate Dr., Suite 240
ExecuTrain, AL 35244
205/987-9877

Arizona

Computer Advantage, Inc.
1855 W. Baseline
Mesa, AZ 85014
602/897-1919

Computer Support Professionals
4545 N. 12th St., Suite B
Phoenix, AZ 85014
602/285-0655

ExecuTrain of Phoenix
410 N. 44th St., Suite 150
Phoenix, AZ 85008
602/231-0406

Mesa Community College
145 N. Centennial Way
Mesa, AZ 85201
602/461-4965

Typetronics
5717 N. 7th St.
Phoenix, AZ 85014
602/274-7253

WJM Computer Sales
3001 N. Stone Ave.
Tucson, AZ 85705
602/7924220

California

Abacus, Inc.
514 Bryant St.
San Francisco, CA 94117
415/759-2481

ABC Schools/ACT Training
4560 Alavarado Canyon Rd.
San Diego, CA 92120
619/280-9933

Accelerated Computer Training
3255 Wilshire Blvd., Suite 903
Los Angeles, CA 90010
213/388-0551

Agenda Business Center
111 N. First St.
Burbank, CA 91502
818/843-7954

Catapult
1 Market Plaza Stewart Tower
San Francisco, CA 94105
415/543-5400

Catapult
3333 Michelson Dr./Office Tower
Irvine, CA 92715
714/261-0477

Computer City SuperCenter
Village Shopping Center, 7901
Garden Grove, CA 92641
800/544-2936

ContractEd Computer Training
13315 Washington Blvd., Suite 203
Los Angeles, CA 90066
213/827-0303

Dublin Group, Inc.
4633 Old Ironside Dr., Suite 420
Santa Clara, CA 95054
800/434-8344

Dublin Group, Inc.
100 1st Street, Suite 350
San Francisco, CA 94105
415/227-4777

ExecuTrain of Los Angeles
1055 Wilshire Blvd., Suite 800
Los Angeles, CA 90017
213/482-3030

ExecuTrain of San Diego
5375 Mira Sorrento Place
San Diego, CA 92121
619/552-8800

ExecuTrain of San Jose
25 Metro Dr., Suite 620
San Jose, CA 95110
408/436-7990

ExecuTrain of Santa Ana
4 Hutton Centre Drive #670
Santa Ana, CA 92715
714/540-6440

Informative Edge
100 Bush Suite 845
San Francisco, CA 94104
415/392-5544

InfoTec Training Institute
5201 Great America Pkwy, Ste. 254
Santa Clara, CA 95954
408/562-6272

Innovative Solutions, Inc.
1320 National Drive
Sacramento, CA 95834
916/928-1700

Institute of Computer Technology
589 W. Fremont Ave/PO Box F
Sunnyvale, CA 94087
408/786-4291

ATC Training Services, continued

Learnsoft, Inc.
5880 Oberlin Drive, Suite 600
San Diego, CA 92121
619/546-1400

MacMentor
2880 Zanker Rd., Suite 107
San Jose, CA 95134
408/433-0990

New Horizons Computer Learning Center
1231 E. Dyer Rd., Suite 140
Santa Ana, CA 92705
714/556-1220 x118

PC Etcetera, Inc.
690 Market Street
San Francisco, CA 94104
415/291-8585

Productivity Point Intl. Training World
562 Missions St., Suite 400
San Francisco, CA 94105
415/243-0100

Quickstart Computer Training
5862 Bolsa Ave #103
Huntington Beach, CA 92649
714/894-1448

Synrgy Computer Training
225 S. Lake Ave., Suite 401
Pasadena, CA 91101
818/356-0380

The Training Alternative
50 First St., Suite 505
San Francisco, CA 94105
415/442-0101

Colorado

Productivity Point International
4500 Cherry Creek Dr. S., Ste. 22
Denver, CO 80222
303/756-5756

Random Access
2370 S. Trenton Way
Denver, CO 80231
303/745-9600

SofTeach
3801 E. Florida Ave., Suite 504
Denver, CO 80210
303/759-1127

Connecticut

CPCE, Inc.
One Corporate Drive, Suite 310
Shelton, CT 06484
203/925-0400

CPCE, Inc.
148 Eastern Blvd., Suite 210
Glastonbury, CT 06033
203/633-0400

ExecuTrain of Fairfield
Five High Ridge Park
Stamford, CT 06905
203/322-8444

Information Products, Inc.
17 Connecticut S. Dr.
East Granby, CT 06026
203/653-7822

Sys-Ed Microcomputer Learning Centers
3001 Summer Street
Stamford, CT 06905
203/356-1230

District of Columbia

Applied Technology Assoc., Inc.
1612 K Street NW, Suite 802
Washington, DC 20006
202/293-0909

Future Enterprises, Inc.
1331 Pennsylvania Ave NW
Washington, DC 20004
202/662-7676

Personal Computer Learning Centers
1101 17th Street NW #411
Washington, DC 20036
202/452-0888

Delaware

Online Consulting, Inc.
913 Market Street Mall
Wilmington, DE 19801
302/658-3018

Ziff Technologies
1001 Jefferson St., Suite 100
Wilmington, DE 19801
302/428-1330

Florida

Computer Savvy, Inc.
2382 W. Oakland Park Blvd.
Ft. Lauderdale, FL 33311
305/486-0644

Computer Tutors USA, Inc.
1242 Tiimberlane Rd.
Tallahassee, FL 32312
804/668-4090

ExecuTrain of Florida-Miami
8240 NW 52nd Terrace, Suite 305
Miami, FL 33166
305/470-2001

ExecuTrain of Florida-Orlando
604 Courtland Street, Suite 150
Orlando, FL 32804
407/740-7762

ExecuTrain of Florida-Tampa
4300 W. Cypress St., Suite 455
Tampa, FL 33607
813/878-2000

MicroAge
1335 Gateway Dr., Suite 2008
Melbourne, FL 32901
407/724-4691

Georgia

Caber Systems, Inc.
3159 Campus Drive
Nordcross, GA 30071
404/840-0067

Computer Essentials
6 Executive Park Dr., Suite 110
Georgia, GA 30329
404/633-3046

Executive Development Group, Inc.
1950 North Park Place
Atlanta, GA 30339
404/952-3030

ExecuTrain of Atlanta
1000 Abernathy Road
Northpark 40
Atlanta, GA 30328
404/396-9200

Micro Center Education
1221 Powers Ferry Road
Marietta, GA 30067
404/859-1545

ATC Training Services, continued

Software IQ
7000 Central Parkway, Suite 1000
Atlanta, GA 30328
404/698-0800

Hawaii

Computer Training Academy
810 Richards Street, Suite 123
Honolulu, HI 96813
808/531-0575

Idaho

ComputerLand
The Learning Center
4795 Emerald, Suite A
Boise, ID 83706
208/345-8024

Illinois

Catapult
1501 E. Woodfield Rd. Suite 105
Schaumburh, IL 60173
708/619-4216

Computer Support Centers/Satellite
108 Wilmot Rd., Suite 390
Deerfield, IL 60015
708/405-0545

Computer Support Centers
890 E. Higgins Rd., #156
Schaumburh, IL 60173

Crawford & Associates, Inc.
6250 River Road, Suite 12-150
Rosemont, IL 60018
708/698-6670

ExecuTrain of Rolling Meadows
1600 Golf Rd., Suite 500
Rolling Meadows, IL 60008
708/806-6515

Productivity Point International
208 S. LaSalle St., Suite 526
Chicago, IL 60604
312/332-3865

Productivity Point International
1419 Lake Cook Rd., Suite 390
Deerfield, IL 60015
708/940-8495

Ziff Technologies
111 W. Washington St.
Chicago, IL 60602
312/807-4949

Indiana

ExecuTrain of Indianapolis
9100 Keystone Crossing, Ste. 750
Indianapolis, IN 46240
317/574-7057

LMB Microcomputers
6330 E. 75 St.
Indianapolis, IN 46250
317/849-4999

Iowa

ComputerLand of Iowa City
5 Sturgis Drive
Iowa City, IA 52246
319/351-4849

Kansas

ExecuTrain of Kansas City
6900 College Blvd., Suite 670
Overland Park, KS 66211
913/451-2898

Haddock Computer Center
PO Box 68440
Wichita, KS 67208
316/683-5211

Productivity Point International
505 N. Murlen Road
Olathe, KS 66062
913/829-7300

Louisiana

ExecuTrain of New Orleans
1450 Poyoras Street, Suite 610
New Orleans, LA 70112
504/593-2200

Maine

Professional Learning Center
2385 Congress St.
Portland, ME 04102
207/772-2335

Maryland

ExecuTrain of Baltimore
1829 Reisterstown Rd., Suite 360
Baltimore, MD 21208
410/602-0200

Image Tech., Inc.
7257 Parkway Dr., Suite 100
Hanover, MD 21076
410/712-4146

Washington Information Systems
2316 Price Ave.
Wheaton, MD 20902
301/949-8500

Massachusetts

Boston Center for
Computer Applications
336 Newbury St.
Boston, MA 02115
617/247-0538

Catapult
888 Washington St., Suite 120
Dedham, MA 02026
617/329-3290

Computer Tutor Corporation/PPI
3 Newton Executive Park
Newton Tower, MA 02162
617/964-5858

Professional Development
Group, Inc.
99 Bedford Street
Boston, MA 02111
508/872-1499

Professional Development
Group, Inc.
160 Cochituate Road
Framingham, MA 01701
508/872-1499

Ziff Technologies
1601 Trapelo Road
Waltham, MA 02154
617/890-0009

Ziff Technologies
101 Federal St. 9th Floor
Boston, MA 02110
617/345-6731

Michigan

Computer City SuperCenter
Oakland Mall, 637 John R. Road
Troy, MI 48083
800/544-2936

Eglehart Training Centers
2882 Thornhills SE
Grand Rapids, MI
616/285-6898

ATC Training Services, continued

ExecuTrain
25330 Telegraph Rd., Suite 210
Southfield, MI 48034
313/352-9210

Midland Computer Shoppe
244 E. Main Street
Midland, MI 48640
517/631-5440

New Horizons/CTSC
33027 Schoolcraft Road
Luionia, MI 48150
313/525-1501

Remex Corporation
6095 28th St. SE
Grand Rapids, MI 49546
616/942-9800

Minnesota

Computer City SuperCenter
Centennial Lakes Plaza, 7491
Edina, MN 55435
800/544-2936

ExecuTrain of Bloomington
8400 Normandale Lake Blvd.
Bloomington, MN 55437
612/921-8844

Inacomp Computer Centers - Education
10371 W. 70th Street
Eden Praire, MN 55344
612/828-6784

Productivity Point International
1700 First Bank West
Minneapolis, MN 55402
612/341-0750

Missouri

Empower Trainers & Consultants
4714 Grand Ave.
Kansas City, MO 64112
816/333-8646

ExecuTrain of St. Louis
540 Maryville Centre Dr., Suite 1
St. Louis, MO 63141
314/469-9979

Forsythe Computers
8514 Eager, Suite G
St. Louis, MO 63144
314/962-9100

Personal Computer Institute
8961 Page Blvd., Suite A
St. Louis, MO 63114
314/426-2200

Systems Service Enterprises, Inc.
795 Office Parkway, Suite 101
St. Louis, MO 63141
314/997-4700

Nebraska

Computer Systems, Inc.
2819 S. 125 Ave., Suite 276
Omaha, NE 68144
402/330-3600

Nevada

Helpware/Microage
4084 S. Kietzke Lane
Reno, NV 89502
702/828-7000

Professional Computer Training
3973 S. McCarran Blvd.
Reno, NV 89502
702/829-8300

New Hampshire

Seacoast Computer, Inc.
12 Portland Ave.
Dover, NH 03820
603/749-6952

New Jersey

B & N Technology
32 Prospect Plains Road
Cranbury, NJ 08512
908/329-2349

Computer Applications Learning Center
100 Hanover Ave.
PO Box 1477
Morristown, NJ 07962
201/539-6050

ExecuTrain of Northern New Jersey
8 Campus Drive
Parsippany, NJ 07054
201/539-4444

Personal Computer Learning Centers
301 Route 17 North
Rutherford, NJ 07070
201/939-0003

New York

Buller, Owens & Associates
989 Ave. of the Americans
New York, NY 10018
212/695-0755

Compu-Tron
1620 Burnet Ave.
Syracuse, NY 13206
315/471-1953

Computer Applications Learning Center
1285 Ave. of the Americas
New York, NY 10019
212/765-9670

Computer City SuperCenter
SouthTown Shopping Ctr., 3333
Henrietta, NY 14623
800/544-2936

Computer City SuperCenter
Bells/Park Edge Rd.
2309 Eggbert
Tonawanda, NY 14150
800/544-2936

CPCE, Inc.
500 Fifth Ave., Suite 2400
New York, NY 10110
212/840-2470

Custom Computer
1775 Express Dr. North
Hauppauge, NY 11788
516/582-6699

Electronic Directions
220 East 23rd St., Suite 503
New York, NY 10010
212/213-6500

ExecuTrain of Western New York
403 Main St., Suite 627
Buffalo, NY 14203
716/852-6716

Logical Operations, Inc.
595 Blossom Road
Rochester, NY 14610
716/482-7700

Margolin, Winer & Evans
600 Old Country Road, Suite 210
Garden City, NY 11530
516/228-9077

PC Etcetera, Inc.
462 Seventh Avenue, 4th Floor
New York, NY 10018
212/736-5870

ATC Training Services, continued

PC Learn
240 West 35th St., Suite 400
New York, NY 10001
212/268-7278

PC Skills, Inc.
4476 Main St., Suite 202
Amherst, NY 14226
716/839-1304

Personal Computer Learning Centers
777 Old Sawmill River Road
Tarrytown, NY 10591
914/347-3200

Personal Computer Learning Centers
230 Park Avenue
New York, NY 10169
212/953-9800

Productivity Point International
1225 Jefferson Rd.
Rochester, NY
716/232-6650

Productivity Point International
19 West 44th Street
New York, NY 10036
212/398-6410

Sage Colleges Institute for Computer Education
140 N. Scotland Ave.
Albany, NY 12208
518/445-1707

Tiger Information Systems
84 William Street, 2nd Floor
New York, NY 10038
212/425-2222

North Carolina

Computer Co-Op
300 S. Elliot Rd.
Chapel Hill, NC 27514
919/942-7704

Computer Decision, Inc.
3100 Gateway Centre Blvd.
Morrisville, NC 27560
919/460-0071

Ohio

Auburn Career Center
8140 Auburn Road
Concord Twp., OH 4077-97
800/544-9750

Blue Chip Computers
3085 Woodman Drive
Kettering, OH 45420
513/299-4594

Cap Gemini America
3 Commerce Park Square, 23200
Beachwood, OH 44122
216/464-8616

Computer Business Skills, Inc.
3055 Kettering Blvd., Suite 210
Dayton, OH 45439
513/294-4636

Computrain, Inc.
6155 Rockside Road #300
Independence, OH 44131
216/447-4400

ExecuTrain of Cincinnati
4555 Lake Forest Dr., Westlake C
Cincinnati, OH 45242
513/769-6913

ExecuTrain of Cleveland
2 Summit Park Dr., Suite 430
Independence, OH 44313
216/524-4190

Inacomp Computer Centers
6 E. Fourth St., Suite 200
Cincinnati, OH 45202
513/421-0332

Jakab & Webster, Inc.
7333 Paragon Road
Dayton, OH 45459
513/438-8070

Logtec
2900 Presidential Dr., Suite 130
Fairborn, OH 45324
513/429-2928

Micro Center Education
11755 Masteller Road
Sharonville, OH 45241
513/782-8509

Micro Center Education
1349 Som Center Road
Mayfield Hei, OH 44124
216/449-7016

Micro Center Education
1555 W. Lane Avenue
Columbus, OH 43221
614/481-4407

Micro Computer Center/ Connecting Point
1353 Lyons Road
Dayton, OH 45458
513/435-9355

Smith/Boehm Associates, Inc.
150 E. Campus View Blvd, Ste. 26
Columbus, OH 43235
614/847-0292

Smith/Boehm Associates, Inc.
8044 Montgomery Road, Ste. 405
Cincinnati, OH 45236
513/891-5212

Oregon

Alpha Computers, Inc.
15110 SW Bangy Road
Lake Oswego, OR 97035
503/620-7000

ExecuTrain of Portland
10220 SW Greenburg Rd., Ste. 535
Portland, OR 97223
503/452-6000

Group 3 Consultants
1201 SW 12th Ave., Suite 310
Portland, OR 97205
503/224-4961

Kinetic Computer Solutions
9225 SW Numbus
Beaverton, OR 97005
503/520-0555

Northwest Computer Learning Center
2659 Commercial SE #210
Salem, OR 97302
503/362-4818

University of Oregon, Continuation Center
720 SW 2nd St.
Portland, OR 97204
503/725-3055

Pennsylvania

Computer Resources & Training, Inc.
1617 J.F.K. Blvd., Suite 1145
Philadelphia, PA 19103
215/569-9100

ATC Training Services, continued

Gannon University
Microcomputer Training
Ste. 101, Rothrock Bldg., 121 W.
Erie, PA 16501
814/456-3838

MicroAge
575 Springdale Drive
Exton, PA 19341
215/524-6665

MicroPower Systems
7942 Saltsburg Road, Route 380
Pittsburgh, PA 15239
412/337-0600

Professional Training
Services, Inc.
1150 First Ave., Parkview Tower
King of Prussia, PA 19406
215/337-8878

Rockey & Associates
The Farmhouse, 12 Great Valley
Malvern, PA 19355
215/640-4880

UBM Training Center
45 Long Lane
Upper Darby, PA 19082
215/352-1444

Ziff Technologies
Liberty Center-1001 Liberty Ave.
Pittsburgh, PA 15222
412/355-0498

Ziff Technologies
7 Penn Center, Suite 405
Philadelphia PA 19103
215/557-7895

Ziff Technologies -- Corporate
1265 Drummers Lane, Suite 102
Wayne, PA 19087
215/293-4625

South Carolina

Entre Computer Center
1850 Sam Rittenberg Blvd.
Charleston, NC 29407
803/769-5600

PPI/Computer Consulting Group
1110 W. Butler Road, Suite C
Greenville, SC 29606
803/277-2870

Tennessee

Computer Learning Center
134 North Peters Road
Knoxville, TN 37923
615/691-1515

ExecuTrain of Memphis, Inc.
65 Germantown Ct., Suite 201
Cordova, TN 38018
901/753-2200

TCBC
5959 Shallowford, #203
Chattanooga, TN 37421
615/899-2464

Texas

AMRIS Training Systems
& Solutions
4255 Amon Carter Blvd.
Ft. Worth, TX 76155
817/963-3520

AMRIS Training Systems
& Solutions
1950 Stemmons Freeway, Ste. 303
Dallas, TX 75207
214/746-5660

BusinessWare Learning Centers
6565 West Loop South, Suite 140
Bellaire, TX 77401
713/665-1985

BYTE Management, Inc.
4907 Blue Mound Road
Ft. Worth, TX 76106
800/759-9300

CompuServe Systems
Integration Group
1950 Stemmons Freeway, Ste. 206
Dallas, TX 75207
214/750-3388

Computer City SuperCenter
Highland Commons
5775 Airport Blvd.
Austin, TX 78752
800/544-2936

Computer City SuperCenter
Greenspoint Commons
12061 I-45 N.
Houston, TX 77060
800/544-2936

Computer City SuperCenter
Weslayan Plaza
3908 Bissonnett St.
Houston, TX 77005
800/544-2936

Edutech
11211 Katy Freeway, Suite 604
Houston, TX 77079
713/827-1314

ExecuTrain of Dallas
12201 Merit Drive, Suite 350
Dallas, TX 75251
214/387-1212

ExecuTrain of Texas, Inc.
2000 Bering Dr., Suite 700
Houston, TX 77057
713/784-1818

Learn Easy, Inc.
5615 Richmond #232
Houston, TX 77060
713/783-3279

MBA Seminars, Inc.
1950 Stemmons Fwy, Suite 5001
Dallas, TX 75207
214/746-5350

Napier & Judd
820 Gessner, Suite 1135
Houston, TX 77024
713/467-8300

Napier & Judd, Inc.
1950 Stemmons Frwy., Ste. 5037M
Dallas, TX 75207
214/746-5028

Productivity Point International
7800 Shoal Creek, #160-E
Austin, TX 78757
512/452-1200

Productivity Point International
8301 Broadway, Suite 120
San Antonio, TX 78209
512/828-9000

Software Academy
14785 Preston Road. #880, LB 58
Dallas, TX 214/788-1694
214/788-1694

Utah

USA Computer Training Center
1739 S. Redwood Road
Salt Lake City, UT
801/972-3422

ATC Training Services, continued

Virginia

American Computer Institute of Training
105 W. Broad St., Suite 501
Falls Church, VA 22046
703/538-2248

Applied Technology Assc., Inc.
131 Park Street NE
Vienna, VA 22180
703/281-7200

Catapult
2011 Crystal Dr. #608
Arlington, VA 22202
703/271-9800

Chaney Computer Associates
1215 U George Washington Memorial
Tabb, VA 23693
804/873-1580

ExecuTrain of Washington DC
1600 Spring Hill Road, Suite 220
Vienna, VA 22180
703/506-9800

First Step Computers/MicroAge
4925 W. Broad St., #406
Richmond, VA 23230
804/358-5881

Gestalt Systems, Inc.
610 Herndon Pkwy., Suite 900
Herndon, VA 22070
703/471-6842

Software Control Intern'l, Inc.
1735 N. Lynn St., Suite 975
Arlington, VA 22209
703/528-4008

Washington

Catapult
13000 Northrup Way, Suite 4
Bellevue, WA 98005
206/882-2036

Computer City SuperCenter
Totem Lake Mall, 12526-A
Kirkland, WA 98034
206/800/544-2936

Dotson Institute
2310 130th Ave NE
Bellevue, WA 98005
206/882-6030

ExecuTrain of Seattle
19515 North Creek Parkway
Bothell, WA 98011
206/485-2508

Green River Community College
841 N. Central Ave., Suite 106
Kent, WA 98031
206/872-6361

Interface Computer School
N. 4601 Monroe, 3rd Floor
Spokane, WA 99205
509/327-7717

Olympia Computer Central, Inc.
3700 Pacific Ave South, Suite A
Olympia, WA 98501
206/491-8768

Wisconsin

ExecuTrain
300 N. Corporate Drive, Suite 140
Brookfield, WI 53045

Starting Point, Inc.
333 Bishops Way, Suite 124
Brookfield, WI 53005
414/784-0557

Independent Training Services

Attar Computer Consulting
90 Park Avenue 30th Floor
NYC, NY 10016
Contact: Alan Attar
212/543-6623
Onsite Windows 3.0 and Word for Windows training for novice and intermediate users.

Cybernetic Research Assc., Inc.
4 Wheatley Avenue
Searingtown, NY 11507-1515
Contact: Steve Lefton
516/873-9668
516/248-5990 (fax)
Onsite Windows 3.0 support for novice users.

Finnegan O'Malley & Co., Inc.
1945 Washington St., 7th Floor
San Francisco, CA 94109-2967
Contact: Fran Finnegan
415/929-7386
CompuServe: 76244,145
Provides consulting and contract programming services.

Introduction to Windows
Oxko Corporation
PO Box 6674
Annapolis, MD 21401
410/266-1671
410/226-6572 (fax)
Contact: Steven Oxman
Windows for novice users.

Michael Price Associates
3517 Edison Way, Suite B
Menlo Park, CA 94025
Contact: Michael Price
415/369-4800
415/369-1379 (fax)
Onsite Windows installation and consulting services for all users.

Personalize Software, Inc.
6636 11th Ave N.
St. Petersburg, FL 33710
Contact: Joe Poston
813/343-3488
Microsoft C for Windows.
Basic Windows Tutorial.
Microsoft C for advanced users.

Support Coordinator Training

Inside Windows 3.0
Nationwide travelling workshops
for intermediate to advanced users.
Now through May 13, 1992
Sponsor: Mastering Computers, Inc.
Continental Plaza
11000 N. Scottsdale Rd., Ste. 175
Scottsdale, AZ 85254
800/800-3436

Inside Windows 3.1
Nationwide travelling workshops
for intermediate to advanced users.
June 30, 1992 - February 10, 1993
Sponsor: Mastering Computers, Inc.
Continental Plaza
11000 N. Scottsdale Rd., Ste. 175
Scottsdale, AZ 85254
800/800-3436

International Windows Developers Conference
Boston, November 11-15, 1992
Sponsors:
Microsoft and Boston University
Contact: Andree Fontaine
508/649-9731

London Windows Developers Conference
London, February 18-20, 1992
Sponsors:
Microsoft and Boston University
Contact: Andree Fontaine
508/649-9731

Windows Discovery Days
Nationwide travelling show
introducing users to Windows.
June 1991 to May 1992
Sponsor: Microsoft
800/426-9400

Windows World '92
A comprehensive Windows
conference held in conjunction
with Comdex.
Chicago, April 6-9, 1992
Sponsor: The Interface Group
300 First Avenue
Needham, MA 02194
617/449-6600

Disks, Tapes, and Videos Training

Disks

Microsoft Productivity Pack 3.1
This is a program that provides
a complete tutorial for Windows 3.1.
It features lesson modules on all of
the major Windows components from
launching programs to installing new
TrueType fonts. For more information
or to get the Microsoft Productivity
Pack call 800/426-9400 and ask for
the reseller location nearest you.

Windows Basics
Comsell, Inc.
500 Tech Parkway
Atlanta, GA 30313
(404) 872-2500
(404) 874-0451 (fax)
Videodisk training for novices.

Tapes

OneOnOne Computer Training
A Division of Mosaic Media, Inc.
2055 Army Trail Rd., Dept. W1091
Addison, IL 60101
800/424-8668
708/628-0550 (fax)
Cassette tutorial for novice users
of Windows 3.0 and 3.1, Word for
Windows, and Excel for Windows.

Personal Training Systems
828 South Bascom Ave., Ste. 100
San Jose, CA 95128
408/286-3843
408/977-1166 (fax)
Self-paced tutorial.

Videos

Fundamentals of Microsoft Windows Programming
Microsoft, Order No. 157-050V200
Hands-on video training on
beginning programming basics
for Windows 3.0.

Getting to Know Windows
PBU (Power Back-Ups)
22924 Hartland Street
West Hills, CA 91307
818/888-9348
Introduction to Windows for novices.

Information at Your Fingertips
Microsoft, Order No. 098-20101
Bill Gates' keynote speech at
Comdex '90

Personal Computing:
The Second Decade Begins
Microsoft, Order No. 098-25752
Bill Gates' strategy for the next
10 years of computing.

The Dealer Training Video
Microsoft, Order No. 098-10771
A reseller's introduction to Windows.

Windows Video
Microsoft, Order No. 050-030-570
Self running Windows introduction.

Technical Training When and Where You Need It:

Microsoft University

Microsoft University provides a proven curriculum on Microsoft systems and network products and a highly skilled staff of experts dedicated to launching courses that meet our customers' needs. Since 1987, over 25,000 software developers, programmers, support and systems engineers, network administrators, data processing professionals, and MIS managers have attended Microsoft University for hands-on technical training.

Microsoft University offers an integrated curriculum on Microsoft Windows, Microsoft Visual Basic, Microsoft C and C++, LAN Manager, Microsoft SQL Server, Microsoft OS/2, and Microsoft OS/2 Presentation Manager.

Microsoft University offers training at ten regional training facilities: Atlanta, Boston, Chicago, Dallas, Los Angeles, New York, San Francisco, Seattle, Toronto, and Washington, D.C., as well as technical training through the Microsoft international subsidiaries.

A Hands-On Approach to Learning

At Microsoft University, we believe that the proof of excellent training is in a student's ability to apply it. All Microsoft University skill-based courses are lab-based; the labs are designed as an integral part of the course. We provide a unique learning environment and exceptionally qualified instructors who combine real-world experience with academic excellence. Our classrooms are furnished with state-of-the-art hardware and classroom equipment.

Schedule an On-Site Course at Your Facility or Ours

Besides our formal classroom program, hundreds of companies have found that hosting a Microsoft University course at their location is a cost-effective way to leverage their training dollars. We provide the instructor, student materials, and expertise. On-site training also provides the opportunity to discuss product strategy and design issues more openly than might be appropriate in a more public setting. Companies can rent a classroom in any of our ten regional facilities to host a Microsoft University course for their own employees.

Video Training: Fast-Forward Your Skill Development

Microsoft University recently introduced a powerful series of technical video courses to give you additional flexibility to meet your ongoing training needs. Video courses are comprehensive, timely, and cost-effective. In addition, they offer an added benefit—a permanent resource for training new employees at a moment's notice. Microsoft University video courses boldly illustrate technical concepts through 3-D animation, lecture, and demonstrations. Hands-on lab exercises and a student guide reinforce the learning experience.

Custom Courses Expand Your Training Options

Unique training needs call for a custom course -- one developed with your specific goals in mind. Because we work with you to develop cost-effective solutions, you can take full advantage of your initial training investment now -- and in the future. Custom courses may be delivered on-site by Microsoft University or licensed for delivery by your internal training organization.

Licensing Programs Present a Global Opportunity

Corporations that need to train more than 20 employees on Microsoft systems have found many advantages in the Microsoft University licensing program. You receive timely, hands-on technical courses from the vendor of the product while bypassing the high cost of developing your own course. And you have the ultimate flexibility to have your own internal training organization deliver the course at any time, to any number of employees, in any of your offices.

Management Education Seminars Enlighten Decision Makers

As new platforms mature and economic pressures increase, leveraging your investment in information technology becomes increasingly important. Microsoft University has pioneered a series of high-level executive seminars and video briefings that address the organizational issues, methodology, and life cycles of new information technologies. These seminars are offered as public seminars and are also ideally suited and cost-effective for presentation at your location.

Training for Developers Using Microsoft Windows 3.1

Microsoft University recognizes the tremendous need for timely, relevant training on the Microsoft Windows operating system version 3.1. We have provided hands-on technical training for Microsoft Windows since its first internal release, and continue to expand our curriculum with courses that provide solutions for your most pressing development needs.

Fundamentals of Microsoft Windows Programming
$1595* per video course; $249 each additional set of student materials

This course is for experienced software developers who want to learn the fundamentals of creating applications for the Microsoft Windows operating system. This course covers the material similar to the Microsoft Windows Programming Environment course and prepares students for the Microsoft Windows Programming Advanced Topics course. Instruction is a mix of lecture and hands-on lab work designed to provide a broad-based foundation of programming skills in Windows. In addition to creating basic applications in Windows, students will learn how to use the Windows messaging system; implement basic user interface objects, such as windows, menus, and dialog boxes; and understand basic memory management techniques.

Microsoft Windows Programming Environment
Five days, $1500 per person

In this course, students create a basic application that displays a window. As the course progresses, increasing functionality is added to the application, such as menus, dialog boxes, and text. Students will learn to recognize and apply the basic elements of the Microsoft Windows application programming interface (API). At course completion, students will know how to use the Windows messaging system and basic memory management techniques, implement basic user interface objects, and create basic Windows applications.

Microsoft Windows Programming—Advanced Topics
Five days, $1500 per person

This intensive course is for experienced programmers who want in-depth information on various aspects of development for Windows. Students study user interface considerations, advanced menus, the help system, windowing, user data entry, advanced dialog boxes, multitasking, and the graphics device interface (GDI). At course completion, students will know how to create multiple document interface (MDI) applications, implement dynamic data exchange (DDE), and make full use of memory and the processor.

Exploring Controls: Video Training for Developers Using the Microsoft Windows Environment
$495* per video course; $99 each additional set of student materials

This video course is for experienced software developers who want to understand how to implement controls for Microsoft Windows applications. It explores various controls in the Microsoft Windows graphical environment, including buttons, static controls, edit controls, list boxes, combo boxes, scroll bars, and custom controls. Students will learn techniques for creating, managing, and using controls, including advanced techniques to use when a standard Windows control doesn't meet an application's requirements. Technical concepts are visually illustrated through 3-D animation and reinforced with hands-on lab exercises and a student guide.

Microsoft Visual Basic—Discovery
Two days, $600 per person

Coming Soon! This course helps students become familiar with the Microsoft Visual Basic programming system. The course covers structured programming and applications development for Microsoft Windows and prepares students to develop prototypes or simple multiple form applications. The Discovery course is the stepping stone to more advanced Visual Basic programming courses.

Microsoft Visual Basic Programming
Three days, $900 per person

This course is for programmers who have experience with structured programming languages and want to develop Windows applications using the Microsoft Visual Basic programming system for Microsoft Windows. The course teaches students the features and capabilities of the Visual Basic programming system, as well as the concepts needed to write sophisticated, event-driven graphical programs.

Advanced Programming for Microsoft Visual Basic
Three days, $900 per person

Coming Soon! This course introduces the advanced features of Microsoft Visual Basic. Students learn how to implement object linking and embedding (OLE), mulitiple document interface (MDI), advanced memory management, and optimization techniques. At course completion, students will know how to use these features, along with other Visual Basic tools, to add advanced capabilities and robustness to their applications for the Microsoft Windows environment.

Application Development Using Microsoft Excel 4.0
Three days, $900 per person

Coming Soon! An exciting new course for systems analysts, support engineers, and MIS professionals. The course assumes that students know how to build a simple spreadsheet, format cells, and create charts. Students will learn usage of macro code, dialog boxes, menus, and an introduction to dynamic data exchange (DDE). Students also explore advanced topics such as object oriented dialog boxes. At course completion, students will be able to create robust and flexible applications based on the Microsoft Excel 4.0 macro language.

* Offer good only in the 50 United States. A charge of $7.50 shipping and handling, plus applicable state sales taxes, will be added to each video course order.

Training for Support Professionals Using Microsoft Windows 3.1

Microsoft University offers a comprehensive curriculum of hands-on courses to help support professionals optimize and customize computing platforms and improve their troubleshooting skills. These courses are based on Microsoft Windows 3.1, providing a timely opportunity to get a head start on this important new software release.

MS-DOS 5 Installation and Optimization: Video Training for Support Engineers and Advanced Users
$149* per video course

This short video course consolidates all the information support engineers and advanced users need to successfully install, optimize, customize, and support MS-DOS 5 on a group of PCs. The course features live demonstrations, 3-D animation, and screen displays to communicate technical information. Students learn how to optimize MS-DOS 5, automate the installation for multiple machines on a network, and customize the MS-DOS 5 shell to meet an organization's needs. In addition, students will be able to show end-users how they can take advantage of the new features of MS-DOS 5.

Microsoft Windows for Support Professionals
Four days, $1200 per person

This in-depth course introduces support professionals to the basic knowledge and skills needed to support users of Windows 3.1. These skills revolve around installation, optimization, customization, and support of the Windows operating system rather than the support of specific applications running under Windows. Students will be learn how Windows operates and will be able to optimize and customize Windows, automate the installation process over a network, and troubleshoot common problems encountered by end users. Lectures are supplemented with hands-on lab exercises to help you apply what you've learned immediately.

Upgrading to Microsoft Windows 3.1: Video Training for Support Professionals
$395* per video course; $99 each additional set of student materials

This fast-paced video course helps experienced Windows 3.0 support professionals learn the new features of Windows 3.1. Students will gain the technical expertise needed to support this upgrade throughout an organization. The course assumes extensive knowledge of Windows 3.0 and focuses on the new features and enhancements of Windows 3.1, including object linking and embedding (OLE), TrueType fonts, and new .INI and .INF files. Students will learn to optimize and customize Windows 3.1, automate the installation process over a network, and troubleshoot common problems encountered by end users. Technical concepts and skills are visually illustrated through demonstrations and 3-D animation, and reinforced through hands-on lab exercises and a student guide.

* Offer good only in the 50 United States. A charge of $7.50 shipping and handling, plus applicable state sales taxes, will be added to each video course order.

Microsoft University Authorized Training Centers

Take the Microsoft Windows for Support Professionals course closer to home. Because of student demand, now you can take the Microsoft Windows for Support Professionals course at Microsoft University authorized training centers across the country. The Microsoft University Training Alliance members listed here are licensed to teach this comprehensive course on installing, optimizing, customizing, and supporting the Microsoft Windows 3.1 graphical environment.

Contact these organizations directly for schedule information or to register for a course. Our program is rapidly expanding with new locations. For information on Microsoft University Training Alliance members opening soon in your area, call Microsoft University at 206/828-1507. When your representative answers, please mention Department 812.

East

CompuServe
New York, New York
800/783-7894

CompuServe
Washington, DC
800/783-7894

MicroAge of Exton
Exton, Pennsylvania
215/524-8585

MicroAge of Exton
Philadelphia, Pennsylvania
215/524-6665

MicroAge of Hartford
West Hartford, Connecticut
203/233-0011

Professional Computer Development Corp.
Exton, Pennsylvania
800/766-7232

Professional Computer Development Corp.
Mars, Pennsylvania
800/766-7232

Microsoft University Authorized Training Centers, continued

South

Computer Savvy, Inc.
Ft. Lauderdale, Florida
305/486-0644

DCA
Alpharetta, Georgia
404/442-4565

Professional Computer Development Corp.
Marietta, Georgia
800/766-7232

Computer Learning Center
Knoxville, Tennessee
615/691-1515

Southwest

CompuServe
Dallas, Texas
800/783-7894

CompuServe
Houston, Texas
800/783-7894

Midwest

Applied Computer Services, Inc.
Orland Park, Illinois
708/614-4900

CompuServe
Chicago, Illinois
800/783-7894

CompuServe
Columbus, Ohio
800/783-7894

Connect Computer Company
Eden Prairie, Minnesota
612/944-0181

Fourth Wave Technologies, Inc.
Troy, Michigan
313/362-2288

LANmind, Inc.
Chicago, Illinois
800/325-5267

Professional Computer Development Corp.
Toledo, Ohio
800/766-7232

Professional Computer Development Corp.
Cincinnati, Ohio
800/766-7232

Professional Computer Development Corp.
Livonia, Michigan
800/766-7232

Professional Computer Development Corp.
Mount Prospect, Illinois
800/766-7232

Professional Computer Development Corp.
Plymouth, Minnesota
800/766-7232

Professional Computer Development Corp.
Schaumburg, Illinois
800/766-7232

Systems Evaluation and Analysis Group, Inc. (SEAG)
Springfield, Illinois
217/523-1717

WORDLINK, INC.
Champaign, Illinois
800/359-5620

WORDLINK, INC.
Schaumburg, Illinois
800/359-5620

WORDLINK, INC.
St. Louis, Missouri
800/359-5620

West

American Digital Technologies
Costa Mesa, California
714/433-1300

CompuServe
Los Angeles, California
800/783-7894

CompuServe
San Francisco, California
800/783-7894

Infotec Development, Inc.
Santa Clara, California
800/282-7990

Infotec Development, Inc.
Santa Ana, California
800/282-7999

Infotec Development, Inc.
San Diego, California
800/282-7999

Learnsoft Corporate Training
San Diego, California
619/546-1400

Learnsoft Corporate Training
Newport Beach, California
714/253-7700

SQLSoft
Kirkland, Washington
206/822-1287

WORDLINK, INC.
San Francisco, California
800/359-5620

Online Services

The following are numbers for popular BBSs. These numbers will help you log on to the service from any city in the country. Listed below each BBS are the appropriate forums from which information or support for Windows is available.

Microsoft
Product Support Services BBS
206/637-9009

CompuServe
800/848-8990

WUGNET, the Windows Users Group Network, is an independent technical organization of Windows professionals. The mission of the organization is to foster the independent exchange of technical information support in the Microsoft Windows environment and related software/hardware. WUGNET has been a critical component of the Windows support community since 1988. If you have any questions, do contact us by typing **GO WUGNET** at any ! prompt.

This is a quick guide to online technical support in various aspects of Windows computing and specific Windows application support. WUGNET is an international technical organization of Windows professionals covering all aspects of Windows technology and computing. Just type **GO XXXX** where *XXXX* is the forum name you want to access. And enjoy!

Forum ID	Section vendor name	Products	Phone
ADOBE	5 Adobe	Illustrator	(415) 962-6076
	5 Adobe	Streamline	(415) 962-6076
	7 Adobe	Type Library/PC	(415) 962-6076
	17 Adobe	ATM for Windows	(415) 962-6076
	17 Adobe	Type Align/Windows	(415) 962-6076
ACAD	11 Auto Desk	AutoCAD for Windows	(415) 332-2344
BANFORUM	15 Banyan	Windows Support	(000) 000-0000
BLYTH	2 Blyth	Omnis 5/Windows	(415) 571-0222
	10 Blyth	Omnis 7/Windows	(415) 571-0222
BORAPP	14 Borland	Screenery	(408) 438-8400
	15 Borland	Object Vision	(408) 438-8400
BPROGA	8 Borland	Turbo Pascal/Windows	(408) 438-8400
BPROGB	6 Borland	C++ for DOS & Windows	(408) 438-8400
	8 Borland	Windows Programming	(408) 438-8400
	10 Borland	Paradox Engine	(408) 438-8400
	13 Borland	Windows Frameworks	(408) 438-8400
CD-ROM	4 CD-ROM Forum	Windows	
	12 Nautilus	Nautilus for Windows	(614) 761-2000
CENTRAL	9 Central Point	Windows Apps	(503) 690-8090
CLMFORUM	9 Computer Lang Mag	Windows Technician	(000) 000-0000
DBADVISOR	4 DB Advisor	Windows & Graphics	(619) 483-6400
DIGITALK	5 Digitalk	Smalltalk/V Windows	(213) 645-1082
DTPFORUM	3 Desktop Publishing	Desktop Pub/Windows	
	9 Desktop Publishing	Shareware Fonts	
DTPVEN	6 Custom Applications	Freedom of Press/W	(508) 667-8585
GRAPHVEN	7 Rix Softworks	Win RIX	(714) 476-8266
GUPTA	Gupta Support BBS	Quest	(415) 321-9500
	Gupta Support BBS	SQL Windows	(415) 321-9500
HPPERIPHER	4 Hewlett-Packard	LaserJet Windows Driver	(800) 752-0900
	5 Hewlett-Packard	DeskJet Windows Driver	(800) 752-0900
	6 Hewlett-Packard	PaintJet Windows Driver	(800) 752-0900
	7 Hewlett-Packard	ScanJet Windows Driver	(800) 752-0900

Forum ID	Section vendor name	Products	Phone (continued)
HPSYSTEMS	10 Hewlett-Packard	New Wave	(800) 752-0900
IBMDESK	4 IBM Desk Software	Current	(203) 783-7000
	7 IBM Desk/Claris	Hollywood	(203) 783-7000
LOTUSA	5 Lotus	123 for Windows	(617) 577-8500
LOTUSB	2 Lotus	Freelance for Windows	(617) 577-8500
	10 Lotus	Notes	(617) 577-8500
	11 Lotus/CC:Mail	CC:Mail for Windows	(617) 577-8500
LOTUSWP	2 Lotus WP	Ami & Ami Pro	(404) 256-2272
	3 Lotus WP	Ami Pro Macros	(404) 256-2272
	6 Lotus WP	SmarText	(404) 256-2272
	7 Windows Shareware		
	8 Clip Art & Graphics		
	11 Problem Uploads		
LTL	Lotus	Lotus Tech Library	(Tech Files)
MACAVEN	4 CE Software	Calendar Maker PC	(515) 224-1995
	5 Delta Point	Delta Graph Pro/W	(408) 648-4000
	14 Claris	Hollywood for Windows	(408) 987-7000
MACBVEN	6 Oldauvi	Read-It OCR	(305) 665-4665
	7 Software Ventures	Microphone II	(510) 644-3232
	9 T/Maker	I Hate Algebra	(415) 962-0195
	9 T/Maker	Windows Clip Art	(415) 962-0195
	14 Radius	Pivot for Windows	(408) 434-1010
MACCVEN	6 CoStar	Address Writer/Windows	(203) 661-9700
MIDIVEN	2 Turtle Beach	Wave for Windows	(717) 843-6916
	8 Big Noise Software	Cadenza for Windows	(904) 730-0754
WINAPA	1 Asymetrix	ToolBook	(206) 637-1500
	2 Attitash	AttiTools	(510) 654-0116
	2 Attitash	Dragnet	(510) 654-0116
	2 Attitash	Wide Angle	(510) 654-0116
	3 DaVinci	DaVinci E-Mail	(919) 881-4320
	4 Future Soft	DynaComm Async	(713) 496-9400
	4 Future Soft	DynaComm Elite 3270	(713) 496-9400
	4 Future Soft	DynaComm New Wave	(713) 496-9400
	4 Future Soft	Terminal Plus	(713) 496-9400
	5 Geographix	SeisMap	(303) 595-0596
	6 hDC Computer	hDC File Apps	(206) 885-5550
	6 hDC Computer	hDC First Apps	(206) 885-5550
	6 hDC Computer	hDC Icon Designer	(206) 885-5550
	6 hDC Computer	hDC Power Launcher	(206) 885-5550
	6 hDC Computer	hDC Windows Express	(206) 885-5550
	7 Hi-Q	Mission Control	(904) 756-8988
	8 MCAE	Inertia	(317) 497-1550
	9 Meta Software	Design/IDEF	(617) 576-6920
	9 Meta Software	Design/OA	(617) 576-6920
	9 Meta Software	MetaDesign	(617) 576-6920
	10 Micrografx	Charisma	(214) 234-1769
	10 Micrografx	Designer	(214) 234-1769
	10 Micrografx/Astral	Picture Publisher	(214) 234-1769
	10 Micrografx	Windows Draw	(214) 234-1769
	10 Micrografx	XPort	(214) 234-1769
	11 Polaris	PackRat	(619) 743-7800
	13 Publishing Tech	Batchworks	(512) 346-2835
	13 Publishing Tech	MultiTack	(512) 346-2835
	13 Publishing Tech	PubTech File Org	(512) 346-2835
	13 Publishing Tech	PubTech Text Edit	(512) 346-2835

Forum ID	Section vendor name	Products	Phone (continued)
WINAPA, continued	14 Roykore	OPUS I	(415) 563-9175
	14 Roykore	Instant Orgchart	(415) 563-9175
	15 Wilson Ware	Command Post	(206) 937-9335
	15 Wilson Ware	Reminder	(206) 937-9335
	15 Wilson Ware	Win Batch	(206) 937-9335
	15 Wilson Ware	Win Check	(206) 937-9335
	15 Wilson Ware	Win Edit	(206) 937-9335
	16 ICOM Simulations	Intermission	(708) 520-4440
	17 Playroom	Button Maker	(704) 536-3093
	17 Playroom	Makeover	(704) 536-3093
	17 Playroom	OSFrame	(704) 536-3093
WINNEW	1 New Uploads		
	2 Microsoft	Set Up/Installation	
	3 Microsoft	Running DOS Apps	
	5 Icons/Bitmaps/RLEs		
	6 Microsoft	Printing	
	7 Microsoft	Cardfile	
	8 Microsoft	File Manager	
	9 Microsoft	Paintbrush	
	10 Microsoft	Terminal	
	11 Microsoft	Hardware	
	12 Microsoft	Windows 2.x	
	13 Microsoft	The Soap Box	
	14 Microsoft	Customer Support	
	15 Microsoft	Write/Notepad	
WINSDK	1 Microsoft	Tools	
	2 Microsoft	Menus/Accelerators	
	3 Microsoft	Controls	
	4 Microsoft	Dialog Boxes	
	6 Microsoft	Fonts	
	7 Microsoft	Palettes & Color	
	8 Microsoft	Bitmap Prog/API	
	9 Microsoft	MDI	
	10 Microsoft	Windows Help System	
	11 Microsoft	DPMI/TSR	
	12 Microsoft	Setup & Install	
	13 Microsoft	Hooks	
	17 Microsoft	General	
WOLFRAM	4 Wolfram Research	Math Reader/Windows	(217) 398-0700
	4 Wolfram Research	Mathematica	(217) 398-0700
WORDSTAR	8 Word Star	Word Star Legacy	(415) 382-8000
	14 Word Star	Correct Grammar/Win	(415) 382-8000
	14 Word Star	Correct Letters/Win	(415) 382-8000
	14 Word Star	Correct Quotes/Win	(415) 382-8000
	14 Word Star	Correct Writing/Win	(415) 382-8000
WPMA	1 Windows/PM Assoc	Public	(408) 562-6065
	2 Windows/PM Assoc	Resources/Events	(408) 562-6065
	3 Windows/PM Assoc	Feedback/Comments	(408) 562-6065
	4 Windows/PM Assoc	Demos/Source	(408) 562-6065
WPSGA	10 WP Support Group	Word Perfect Windows	(301) 387-7322
XTALK	10 DCA/Crosstalk	Crosstalk for Windows	(404) 442-3210
ZENITH	15 Windows Shareware		

PC Magazine Areas

These areas on CompuServe can be accessed by typing **PCM:XXXX** where *XXXX* is the name of the desired forum.

When you want to leave the *PC Magazine* area and return to CompuServe itself, just type **CIS:XXXX** where *XXX* is the next forum you want to visit.

PC Magazine Forum ID	*Section vendor name*	*Products*	*Phone*
PCM:UTILFORUM	10 PCMAG Windows Utils		
PCM:PROGRAMMING	2 User Environments		
MSAPP	2 Microsoft	PowerPoint for Windows	
	3 Microsoft	MS Mail for Windows	
	7 Microsoft	Project for Windows	
	10 Microsoft	Works for Windows	
	12 Microsoft	Word for Windows	
	13 Microsoft	Publisher for Windows	
	14 Microsoft	Customer Support	
	17 Microsoft	Money for Windows	
MSBASIC	5 Microsoft	Visual Basic	
	9 Microsoft	Projects (Lurker)	
	14 Microsoft	Customer Support	
MSEXCEL	3 Microsoft	Excel for Windows	
	14 Microsoft	Customer Support	
MSKB	Microsoft	Knowledge Base	(Tech Files)
MSL	Microsoft	Software Library	(Tech Files)
MSLANG	7 Microsoft	Quick C for Windows	
	14 Microsoft	Customer Support	
MSNETWORKS	1 Microsoft	General	
	11 Microsoft	3rd Party Drivers	
MSOPSYS	10 Non-MS Win Drivers		
	11 Shareware/FreeWare		
	12 Press Releases		
	13 Software Demos		
	14 Microsoft	Customer Support	
	16 Microsoft	ISV Communications	
MULTIMEDIA	6 Multimedia Corp.	Windows Multimedia	(408) 737-7575
	12 Multimedia Corp.	MPC - Multimedia PC	(408) 737-7575
MULTIVEN	2 Meridian Data	Windows CD ROM Kit	(408) 000-0000
	4 Video Logic	DVA-4000 Card	(000) 000-0000
	8 Authorware	Authorware Pro Win	(415) 595-3101
	9 Ntergaid	Hyper Writer 3/Win	(203) 368-0632
	10 HSC Software	HSC Quick Show	(310) 392-8441
NORUTL	9 Peter Norton Group	Norton Anti-Virus/W	(408) 253-9600
	12 Peter Norton Group	Norton Desktop	(408) 253-9600
	13 Peter Norton Group	Norton Backup/Windows	(408) 253-9600
	14 Peter Norton Group	NDW Batch Tips	(408) 253-9600
NOVB	15 Novell	Windows	(801) 379-5900

PC Magazine Forum ID	Section vendor name	Products	Phone (continued)
NOVLIB	1 Novell File Libs	Novell New Uploads	(Tech Files)
	3 Novell File Libs	Netware 2.x Specific	(Tech Files)
	4 Novell File Libs	Netware 3.x Specific	(Tech Files)
	5 Novell File Libs	Client/Shell Drivers	(Tech Files)
	7 Novell File Libs	Btrieve/XQL	(Tech Files)
	11 Novell File Libs	Tech Info/TMSPs	(Tech Files)
	13 Novell File Libs	Other Patches/Driver	(Tech Files)
	14 Novell File Libs	Independent Develop	(Tech Files)
	15 Novell File Libs	Shareware/Demo	(Tech Files)
	16 Novell File Libs	Public Domain	(Tech Files)
	17 Novell File Libs	Other New Uploads	(Tech Files)
ORACLE	4 Oracle	Oracle Card/Windows	(415) 506-7000
	4 Oracle	ToolBook DLL	(415) 506-7000
PCEO	5 Intel	FaxIt for Windows	(800) 438-4769
	11 Intel	Net Satisfaxtion	(800) 438-4769
PCVENA	1 Button Ware	Take Note	(206) 454-0479
	13 Foresight Resources	Windows Drafix CAD	(816) 891-1040
PCVENB	8 DAC Easy	Main LAN for Windows	(214) 248-0305
PCVENC	15 Beyond	Beyond Mail for Windows	(617) 621-0095
PCVEND	2 Attisoft	LANtastic for Windows	(602) 293-6363
	8 Timeslips	Time Sheet Pro/Windows	(508) 768-6100
	15 Bloc Publishing	3D Charts to Go	(305) 445-0903
	15 Bloc Publishing	Wind-Ease	(305) 445-0903
PCVENE	4 Farallon	Apple Talk PC	(510) 000-0000
SBSALDFOR	2 Aldus	Pagemaker for Windows	(206) 628-2320
	3 Aldus	Persuasion for Windows	(206) 628-2320
	4 Aldus	Photo Styler for Windows	(206) 628-2320
	5 Aldus	Free Hand for Windows	(206) 628-2320
SPCFORUM	5 Software Publishing	Pro Write Plus	(415) 962-8910
	8 Software Publishing	Harvard Graphics/Win	(415) 962-8910
	9 Software Publishing	Harvard Draw/Windows	(415) 962-8910
	14 Software Publishing	Info Alliance/Windows	(415) 962-8910
	15 Software Publishing	Superbase Support	(415) 962-8910
	16 Software Publishing	Superbase 4 Adv	(415) 962-8910
	17 Software Publishing	Superbase 4 Devel	(415) 962-8910
SPINNAKER	2 Spinnaker	Plus for Windows	(617) 494-1200
	11 Spinnaker	PFS: Windows Works	(617) 494-1200
SYMFORUM	4 Symantec	On Target	(408) 253-9600
	11 Symantec	Just Write	(408) 253-9600
	13 Symantec/Zortech	C++ for Windows	(408) 253-9600
VENTURA	4 Ventura Software	Ventura Pub/Windows	(800) 822-8221
	14 Ventura Software	Form Base	(800) 822-8221
	14 Ventura Software	Ventura DB Publish	(800) 822-8221
VIRUS FORU	2 McAfee Associates	Windows Scan Virus Scan	(000) 000-0000
WINADV	1 New Uploads		
	2 Microsoft	Accessories	
	3 Microsoft	Batch Files	
	4 Microsoft	Comm Ports	
	5 Microsoft	Enhanced Mode	
	6 Games		
	7 Microsoft	Graphics/Video	
	8 Microsoft	Memory Issues	
	9 Microsoft	Networks	
	10 Microsoft	Optimization	
	11 Windows Shareware		
	12 Windows Utilities		

PC Magazine Forum ID	Section vendor name	Products	Phone (continued)
WINADV, continued	13 The Soap Box		
	14 Microsoft	Customer Support	
	15 Miscellaneous		
	17 Microsoft	Dr. Watson Logs	(Upload Area)
WINAPB	1 Caseworks	CASE:W	(404) 399-6236
	2 Corel Systems	Corel Draw!	(613) 728-8200
	3 NBI	Legacy	(303) 444-5710
	4 Owl International	Guide	(206) 747-3202
	5 Chip Soft	Turbo Tax for Windows	(619) 453-8722
	6 Whitewater Group	Actor Professional	(708) 328-3800
	6 Whitewater Group	Object Graphics	(708) 328-3800
	6 Whitewater Group	Whitewater Res Tool	(708) 328-3800
	6 Whitewater Group	WinTrieve	(708) 328-3800
	7 ZSoft	Photo Finish	(404) 428-0008
	7 ZSoft	Pub Paintbrush/Windows	(404) 428-0008
	7 ZSoft	Pub Type Foundry	(404) 428-0008
	7 ZSoft	Soft Type	(404) 428-0008
	8 WUGNET	Windows Journal	(215) 565-1681
	8 WUGNET	Windows Users Group Support	
	9 Kidasa	Milestones	(512) 282-1544
	10 SoftCraft	Soft Fonts	(608) 257-3300
	11 Softbridge	Auto Test Facility	(617) 576-2257
	11 Softbridge	Bridge Batch	(617) 576-2257
	11 Softbridge	Bridge Tool Kit	(617) 576-2257
	12 Zenographics	Import for Windows	(714) 851-6352
	12 Zenographics	Super Print	(714) 851-6352
	13 Computer Present	Color Lab	(513) 281-3222
	13 Computer Present	Image Prep	(513) 281-3222
	14 WindowCraft Corp	WindowCraft	(617) 272-0999
	15 Knowledge Garden	Knowledge Pro/Windows	(518) 766-3000
	16 Delrina Technology	PerForm Pro	(416) 441-3676
	16 Delrina Technology	WinFax	(416) 441-3676
	17 Windows Corporate Forum	(Members Only Area)	(215) 565-1861
WINAPC	1 Matesys	Object Script	(415) 925-2900
	1 Matesys	Object Tel	(415) 925-2900
	1 Matesys	Object View	(415) 925-2900
	1 Matesys	Simple Windows	(415) 925-2900
	2 Bell Atlantic	Thinx	(708) 974-1210
	3 The Stirling Group	Dbx SHIELD	(708) 307-9197
	3 The Stirling Group	Demo SHIELD	(708) 307-9197
	3 The Stirling Group	Install SHIELD	(708) 307-9197
	3 The Stirling Group	Log SHIELD	(708) 307-9197
	3 The Stirling Group	Mem SHIELD	(708) 307-9197
	3 The Stirling Group	Tbx SHIELD	(708) 307-9197
	4 Berkeley Systems	After Dark/Windows	(510) 540-5535
	5 Glockenspiel	C++ for Windows	
	6 Campbell Services	On Time For Windows	(313) 559-5955
	7 Metz Software	File Manager F/X	(206) 869-6292
	7 Metz Software	Metz Shareware	(206) 869-6292
	8 Within Technologies	Realizer	(609) 273-9890
	9 Abacus	Becker Tools	(800) 451-4319
	9 Abacus	No Mouse	(800) 451-4319
	9 Abacus	Virus Secure	(800) 451-4319
	11 Saros	Mezzanine	(206) 646-1066
	11 Saros	Saros File Share	(206) 646-1066

GEnie 800/638-9636

Microsoft Roundtables M_505.

The following are online Windows library services. These services are available by calling the numbers listed or by contacting the commercial networks suggested.

Windows Driver Library

The Windows Driver Library is for all users of Windows who own peripheral devices not supported in the Windows retail package.

The WDL will include the Windows 3.1 compatible device drivers that are not included in the Windows 3.1 retail box, including Printer, Display, Network, Fastdisk and VDD and Grabbers for the display drivers. All drivers in the WDL are compatibility tested by Microsoft. The WDL will be updated regularly as new and improved drivers become available.

The WDL will be distributed on network services including CompuServe, GEnie, ONLine, APCUG, and Microsoft PSS Download Service. The WDL is free of charge when downloaded from these services. The WDL is also available for a $20 fee from Microsoft Customer Service at 800/426-9400.

The Windows Technical Support Library

This service will help users who are either PC support professionals or who support Windows users in their organization. This Library provides better and faster access to the information needed to solve Windows technical support problems. It will also help you update and customize the Windows Resource Kits with up-to-date technical information related to specific applications, hardware devices, and networks.

The Windows Technical Support Library will be available by referencing the appropriate forums on network services such as GEnie and CompuServe or free of charge by calling Microsoft PSS Download Service. To have information faxed or mailed call Microsoft Product Support Services.

User Groups

The following is a list of some top user groups. Although phone numbers may be listed, these groups are staffed by volunteers, and we suggest when possible that correspondence be directed to the address given. For the most updated information about user groups near you, call 800/426-9400 to speak to a Microsoft operator who will give you the most recent address information.

United States

Arizona

Phoenix PC Users Group
PO Box 35637
Phoenix, AZ 85069-5637

Windows SIG
Contact: Ray Moore
602/222-8511

Tuscon Computer Society
PO Box 1489
Tuscon, AZ 85702
Newsletter: monthly

Windows SIG
Contact: Kevin Breslin
602/571-7128

Arkansas

Arkansas/Oklahoma PC User Group
PO Box 853
Fort Smith, AR 72901

Windows SIG
Contact: Greg Disch
501/783-0293

Arkansas Windows Developers Users Group
Foxfire, Inc.
Box 198
Heindsville, AR 72738

California

Corporate Windows Council
1105 Colman, Ave.
San Jose, CA 95108
Contact: FMC Corporation
415/601-5000

Diablo Valley PC User Group
1635 School Street, Suite 101
Morago, CA 94556-1125
Newsletter:monthly

Windows SIG
Contact: Ronald Ogg
510/376-7174

Golden Gate PC SIG in Windows
PO Box 151696
San Rafael, CA 94915-1696

Los Angeles/UCLA Computer Society
2629 Manhatten Ave. Suite 273
Hermosa Beach, CA 90254

Windows SIG
Contact: Tom Cox

Los Angeles Windows Developers User Group
Microsoft Corporation
6701 Center Dr. #1300
Los Angeles, CA 90045

Windows SIG
Contact: Margaret Johnson

North Orange County Computer Club/NOCC
5592 Danny Ave.
Cypress, CA 90630
Newsletter: monthly/Orange Bytes
Contact: Jaunita Lockwood
714/761-2583

Windows SIG

Orange Coast IBM User Group
3021-B Harbor Blvd.
Costa Mesa, CA 92626
or PO Box 6100-211
Costa Mesa, CA 92628
Newsletter: monthly

Windows SIG
Contact: Richard Villa
714/841-6991

Pasadena User Group
711 E. Walnut, Suite 306
Pasadena, CA 91101
Newsletter:monthly/Prompt>
Contact: Steve Bass
818/795-2300
BBS #: 818/332-6580

Windows SIG

Sacramento PC Users Group
1661 Pebblewood Drive
Sacramento, CA 95833
Newsletter: monthly/Sacra Blue

Windows SIG
Contact: Steve Friedman
916/920-5573

San Francisco PC Users Group
455 Eddy St. #1109
San Francisco, CA 94109

Windows SIG
Contact: Carol Kramer

San Francisco Users Group
3145 Geary Blvd., Suite 155
San Francisco, CA 94118
or 240 Chattanooga, #30
San Francisco, CA 94114
Newsletter: monthly

Silicon Valley Computer Society
780 North 4th Street
San Jose, CA 95112-5014
Newsletter: monthly

Windows SIG

The Software Entrepreneurs' Forum
PO Box 61031
Palo Alto, CA 94306

Windows SIG
Contact: Fran Finnigan
415/929-7386

Windows Presentation Manager Association
5201 Great American Parkway
Santa Clara, CA 95054
Contact: Techmart Executive
408/562-6065

Colorado

Windows on the Rockies
PO Box 9929
Denver, CO 80209-0929
Contact: Jack McKinley
(303) 733-1277

User Groups, continued

District of Columbia

Capital PC Users Group
Washington, DC

Windows/Presentation Manager SIG
Contact: Charles Kelly, Rod Wright

Pentagon PC User Group
PO Box 47047
The Pentagon
Washington, DC 20050-7046
Newsletter: The Review

Florida

Alachua County Computer Users Group
Rt 1 Box 200
Newberry, FL 32669
Newsletter:monthly/Gator Bytes

Windows SIG
Pinellas IBM-PC User Group
5100 Dover Street NE
St. Petersburg, FL 33703

Windows SIG
Contact: Dick Kellerman
813/525-0989

Georgia

Atlanta Windows User Group
PO Box 28788
Atlanta, GA 30358

Illinois

Chicago Area Microcomputer Professionals/CAMP
950 Skokie Blvd., Suite 205
Northbrook, IL 60062

Chicago Computer Society
PO Box 8681
Chicago, IL 60680
or 2849 Arlington Ave.
Highland Park, IL 60035
Newsletter: monthly

Guidance for Users of Integrated Data Processing Equipment/GUIDE
111 E. Wacker Drive, Suite 600
Chicago, IL 60606

The Widows SIG of the Chicago Computer Society
638 N. Stephen Dr.
Palatine, IL 60067
Contact: Ray Elseth
708/705-9721

Indiana

Indianapolis Computer Society
PO Box 2532
Indianapolis, IN 46205
or 6135 N. Meridan Street
Indianapolis, IN 46208
317/251-2003

Indiana/Kentucky PC Users Group
PO Box 3564
Louisville, KY 40201
Newsletter: monthly

Windows SIG
Contact: Jim Travelstead
502/636-3361

Kentucky

Indiana/Kentucky PC Users Group
PO Box 3564
Louisville, KY 40201
Newsletter: monthly

Windows SIG
Contact: Jim Travelstead
502/636-3361

Maryland

Capital PC Users Group
51 Monroe Street, Plaza East 2
Rockville, MD 20850
Newsletter: monthly

Windows SIG
Contact: Dan Schnal

Massachusetts

Boston Computer Society (BCS)
188 Needham Street
Newton, MA 02164
617/235-9505

Windows SIG

Michigan

Central Michigan Windows User Group
2356 Science Parkway, Suite 100
Okemos, MI 48864
Contact: Stewart Nutter
517/3470-800

Windows Users Group Network
6934 W. Fort St.
Detroit, MI 48209

Windows SIG
Contact: Ann Stanzel

Minnesota

Twin City PC/TCPC
5701 Normandale Blvd.
Edina, MN 55424

Windows SIG
Contact: Tim Breen
612/377-7317

Missouri

St. Louis User Group
18 Covington Meadows
St. Louis, MO 63132
Newsletter: every other month
314/458-9604

New Jersey

Microcomputer Management Association/MMA
PO Box 4615
Warren, NJ 07060
Newsletter: TBA

PC User Group/Amateur Computer Group of New Jersey
Country Lights Villa #121
Bensalem, PA 19020
Contact: Bob Todd
215/752-4604

Windows SIG

User Groups, continued

New York

Buffalo IBM-PC User Group
(BIBMUG)
PO Box 609
Amhurst, NY 14226-0609
or 3380 Sheridan Dr., Suite 422
Columbus, OH 43209
Newsletter: monthly

Windows SIG

New York PC User Group/NYPC
40 Wall St., Suite 2124
New York, NY 10005-1301
Newsletter: monthly/NYPC

Windows SIG
Contact: Nancy Thalblum
212/548-5868

Word for Windows
Business Users Group
420 Lexington Ave., Suite 640
New York, NY 10170

Ohio

Cincinnati PC Users Group
PO Box 3097
Cincinnati, OH 45201
or 7822 Eustic Court
Cincinnati, OH 45236
Newsletter: monthly

Windows SIG
Contact: Darrel Booth
513/745-9356

Columbus Computer Society
PO Box 1556
Columbus, OH 43216
Newsletter: monthly

Windows SIG
Contact: Bruce Hotte

Oklahoma

Arkansas/Oklahoma PC User Group
PO Box 853
Fort Smith, AR 72901

Windows SIG
Contact: Greg Disch
501/783-0293

Oregon

Portland PC Club
921 SW Morrison, Suite 529
Portland, OR 97205
Newsletter: monthly/Silicone Forest
Gazette

Windows SIG
Contact: Mark Walsh
503/284-6718

Pennsylvania

National Windows User Group
Network
PO Box 1967
Media, PA 19063
Contact: Howard Sobel
215/565-1861

Philadelphia Area PC Users Group
c/o La Salle University
Box 312
Philadelphia, PA 1914
Newsletter: monthly/DATA BUS

Windows SIG
Contact: Jack Kramer

Texas

HAL-PC (Houston Area League of
PC Users) Headquarters
1200 Post Oak, Suite 106
Houston, TX 77056
713/623-4425
Newsletter: WinApps Solution

Windows Excel/DBMS SIG
Contact: Ken Penrod
713/651-1276

Windows SIG
Contact: Larry Sands
713/432-1000

Windows Developers SIG
Contact: Richard Bermudez
713/681-2102

Windows Publishing SIG
Contact: Larry Sands
713/432-1000

Word for Windows SIG
Contact: Charles Barnhart
713/792-4802

North Texas PC User Group
PO Box 780066
Dallas, TX 75378-0066
Newsletter: monthly/
N.Texas PC News
214/746-4699

Windows SIG
Contact: James Dunn

Utah

Utah Blue Chips
PO Box 510811
Salt Lake City, UT 84151
or 9296 S. Greenwood
Sandy, UT 84070
Newsletter: monthly

Washington

Pacific NW PC Users Group
1925 7th Avenue
Seattle, WA 98101
Newsletter: monthly

Windows SIG
Contact: Patrick Long
206/789-0087

Wisconsin

Madison PC Users Group
PO Box 2598
Madison, WI 53701-2598
608/255-1205

Canada

Vancouver Windows User Group
210-1090 W. Pendor Street
Vancouver, BC V6E 2N7
Contact: Ron Usher

WUGNET Toronto
PO Box 95 Streetsville, Mississauga
Ontario, Canada L5 M2B7
Contact: Don Roy

United Kingdom

The Windows Form (Softtech
Consultants Ltd.)
Charles Street
Windsor SL4 5DS, United Kingdom

Windows Users Group
Enterprise House Carlton Road
Workshop, Notts S81 7QF
United Kingdom

Hardware Support

Hardware manufacturers also offer support for Windows running on their hardware. Call your hardware manufacturers for assistance at the appropriate numbers listed below.

ALR
714/581-6770

Acer Technologies
800/637-7000

Actech
703/641-1223

Alphacom International
510/490-1669

Ares Microdevelopment
800/322-3200

Argent Technologies
802/288-1238

Ariel
800/552-7435

AST Research, Inc.
714/727-4141

AT&T
800/922-0354

Austin
800/752-4141

Automated Computer Tech
800/521-9237

Bell Computer
818/909-3501

Caliber Computer
408/942-1220

Comark (dba USA Flex)
800/441-5416

Compaq Computer Corp.
800/231-0900

Compton Multimedia Publishing Group
415/597-5555

CompuAdd (PC Source)
800/999-9901

Computer Marketing Corp.
612/894-7150

Computer Sales Professionals, LTD
201/865-8870

Corporate Software
800/677-4003

Cube Computer
914/592-8282

Cumulus
216/464-2211

Danjen
518/377-3377

Data General Corporation Personal Computers
800/DGHELPS

Dell
800/624-9896

Digital Equipment Corporation
800/DIGITAL

DTK
818/810-88880

Entertainment Marketing
713/995-4433

Epson
800/992-8911

Ergo
800/633-1925

Everex Systems
510/498-4411

Gateway 2000
800/232-2000

Goldstar Technology
408/432-1331

Hauppauge
516/434-3197

Hertz Computer
212/684-4141

Hewlett-Packard
800/858-8867

Hyundai
800/234-3553

IBM
800/426-3333

Laser
708/540-5022

Leading Edge
508/836-4800

Level IV
800/589-5848

Logitech, Inc.
510/795-8501 or
510/795-8754

MTD Data Services (dba Bear Computer)
206/869-5900

Management Advisory Software
800/446-1778

Master Computers
206/365-1156

Maxum Systems
718/643-3385

Memorex-Telex
800/522-0822

Mitsubishi
800/344-6352

NCR
800/262-7782

NEC
800/388-8888

The Network Connection
800/327-4853

Northern Computer Systems
612/934-8856

Northgate
800/446-5037

PC Craft
714/256-5023

Packard Bell
800/733-4411

Hardware Support, continued

Phoenix
617/551-4100

Positive Support
800/252-6345

Samsung
408/434-5479

Sysorex Information Systems
703/273-9200

Tandon Support
800/487-8324

Tandy
817/878-6875

Texas Instruments
512/250-7407

Toshiba
800/999-4273

US Micro Express
206/462-1011

Ultra-Comp
314/298-1998

Unisys
800/328-0440 x03852

Wang
800/253-3932

Wyse
408/435-2770

Xycom
313/429-4971

Zenith
For Windows/MSDOS on Zenith hardware, contact your local Zenith dealer

Zeos
800/228-5390

Numbers for manufacturers largely provided courtesy of WUGNET and may be found in Volume 1, Number 5 of the *Windows Journal.*

Reference Materials

Books

Introductory Books

10 Minute Guide to Windows 3
SAMS
Doug Sabotin and Katherine Murray, 1991
Tutorial for novice users.

10 Minute Guide to Windows 3.1
SAMS
Katherine Murray and Doug Sabotin
Tutorial for novice users.
Pending software release

ABCs of Windows 3.0
Sybex
Kris Jamsa, 1990
Tutorial for novice users.

The Best Book of Microsoft Windows 3
SAMS
Carl Townsend, 1991
Tutorial/reference for all users.

The First Book of Microsoft Windows 3
SAMS
Jack Nimersheim, 1991
Tutorial/reference for novice users.

The First Book of Windows 3.1, 2nd Ed.
SAMS
Jack Nimmersheim
Tutorial/reference for novice users.
Pending software release

Getting Started With Windows 3.1
Microsoft Press
Russell Borland
$19.95 ($27.95 Canada)
ISBN 1-55615-472-0
Tutorial for novice users.

Learning and Running Windows 3.1
Microsoft Press
Craig Stinson
$39.95 ($54.95 Canada)
ISBN 1-55615-431-3
Training guide for all users; includes software.

Mastering Windows 3.0
Sybex
Robert Cowart, 1990
Tutorial for intermediate users.

Running Windows 3.1
Microsoft Press
Craig Stinson
$27.95 ($37.95 Canada)
ISBN 1-55615-373-2
Tutorial for all users.

Simply Windows
Osborne McGraw-Hill
Mary Campbell
Introduction for novice users.

Using Windows: New Ed.
Que
Ron Person and Karen Rose, 1991
Introduction to Windows for novices.

Windows 3 QuickStart
Que
Ron Person and Karen Rose
Introduction to Windows for novices.

Windows 3.0 Made Easy
Osborne McGraw-Hill
Tom Sheldon, 1990
Teaches novices to use Windows.

Windows 3.0 Quick Reference
Que
Tutorial for all users.

Using Windows 3.1
Que
Ron Person and Karen Rose
Introduction to Windows 3.1 for novices. Pending software release

Windows 3.1 Companion
Microsoft Press
The Cobb Group
$27.95 ($37.95 Canada)
ISBN 1-55615-372-4
Step-by-step tutorial for novice users.

Windows 3.1 Made Easy
Osborne McGraw-Hill
Tom Sheldon, 1992
Tips and tricks for all users.

Windows 3.1 QuickStart
Que
Ron Person and Karen Rose
Introduction to Windows 3.1 for novices. Pending software release

Tips and Tricks Reference Books

1-800-H-E-L-P With Windows 3.1
Carl Townsend
SAMS
Reference guide for all users.
Pending software release

Concise Guide to Microsoft Windows 3.1
Microsoft Press
Kris Jamsa
$12.95 ($17.95 Canada)
ISBN 1-55615-470-4
Reference for intermediate users.

Easy Windows
Que
Shelley O'Hara
Reference guide for novice users.

Easy Windows, 3.1 Edition
Que
Shelley O'Hara
Reference guide for novice and intermediate users.
Pending software release

Inside Windows 3.1
New Riders Publishing
1992; pending software release.
Tutorial and references for all users.

Instant Reference 3.0
Sybex
Marshall Moseley, 1990
Reference guide for all users.

Maximizing Windows 3
New Riders Publishing
Jim Boyce, 1991
Tips and tricks; customization techniques for more advanced users.

Maximizing Windows 3.1
New Riders Publishing
Jim Boyce, 1992; pending software release. Tips and tricks for more advanced users.

Tricks of the Windows 3.1 Masters
SAMS
Ed Tiley
For intermediate to advanced users.
Pending software release

Upgrading to Windows 3.1
Que
Que Development Group
Reference guide for all users.
Pending software release

Using Microsoft Windows 3
Que
Ron Person and Karen Rose, 1991
Tips and tricks for all users.

Windows 3.0 On Command
New Riders Publishing
1991
Reference guide for all users.

Windows 3.0 Secrets
IDG Books
Brian Livingston
Tips and tricks for all users.

Windows 3.0: The Complete Reference
Osborne McGraw-Hill
Tom Sheldon, 1991
Reference guide for all users.

Windows 3.1 On Command
New Riders Publishing
1992; pending software release
Reference guide for all users.

Windows 3.1 Quick Reference
Que
Que Development Group
Reference guide for all users.
Pending software release

Windows 3.1 Revealed
SAMS
Barry Sosinksy and Robert Mullin
For intermediate and advanced users.
Pending software release

Windows 3.1: The Complete Reference
Osborne McGraw-Hill
Tom Sheldon, April 1992
Reference guide for all users.

Windows: The Pocket Reference
Osborne McGraw-Hill
Allen Wyatt, April 1992
Quick reference for all users.

Windows Tips and Techniques
Mastering Computers, Inc.
Thomas Graunke, 1990
Tips for intermediate to advanced users. Optimizing to troubleshooting.

Programming Books

Developing Windows Applications with Microsoft SDK
SAMS
Brent E. Rector, 1991
For novice to intermediate programmers.

The Microsoft Guide to Designing the User Interface for Windows 3.1-Based Applications
Microsoft Press
Microsoft Corporation
$27.95 ($37.95 Canada)
ISBN 1-55615-384-8
Reference guide for all Windows programmers.

Microsoft Visual Basic Workshop
Microsoft Press
John Clark Craig
$34.95 ($44.95 Canada)
ISBN 1-55615-386-4
Book-and-software package for programmers.

Microsoft Win32 Applications Programming Interface: The Programmer's Reference, Vol. 1 & 2
Microsoft Press
Microsoft Corporation
$45.00 ($60.00 Canada), each volume
Special Offer: $70.00 for both
ISBN 1-55615-433-X
Complete reference for developers.

Microsoft Windows 3 Developer's Workshop
Microsoft Press
Richard Wilton
$24.95 ($32.95 Canada)
ISBN 1-55615-244-2
Reference for experienced programmers.

Microsoft Windows 3.1 Programmer's Reference Library
These references are a complete reference for serious Windows programmers. This six-book series is the official documentation of the Microsoft Windows 3.1 Software Development Kit (SDK) and is included with that software.

Programming Books, continued

Microsoft Windows 3.1 Guide to Programming
Microsoft Press
Microsoft Corporation
ISBN 1-55615-452-6
Available Summer 1992

Microsoft Windows 3.1 Programmer's Reference Vol. 1: Overview
Microsoft Press
Microsoft Corporation
ISBN 1-55615-453-4

Microsoft Windows 3.1 Programmer's Reference Vol. 2: Functions
Microsoft Press
Microsoft Corporation
ISBN 1-55615-463-1

Microsoft Windows 3.1 Programmer's Reference Vol. 3: Messages, Structures, Macros
Microsoft Press
Microsoft Corporation
ISBN 1-55615-464-X

Microsoft Windows 3.1 Programmer's Reference Vol. 4: Resources
Microsoft Press
Microsoft Corporation
ISBN 1-55615-494-1

Microsoft Windows 3.1 Programming Tools
Microsoft Press
Microsoft Corporation
ISBN 1-55615-454-2

The Microsoft Windows Multimedia Programmer's Reference Library
For developers who want to add a high level of sound, music, audio, and other multimedia elements to their programs. These three volumes are the official Microsoft documentation for the Microsoft Multimedia Development Kit, and are included with that software product.

Microsoft Windows Multimedia Programmer's Workbook
Microsoft Press
Microsoft Corporation
$22.95 ($29.95 Canada)
ISBN 1-55615-390-2

Microsoft Windows Multimedia Programmer's Reference
Microsoft Press
Microsoft Corporation
$27.95 ($36.95 Canada)
ISBN 1-55615-389-9

Microsoft Windows Multimedia Authoring and Tools Guide
Microsoft Press
Microsoft Corporation
$24.95 ($32.95 Canada)
ISBN 1-55615-391-0

Microsoft Windows for Pen Computing Programmer's Reference
Microsoft Press
Microsoft Corporation
$24.95 ($34.95 Canada)
ISBN 1-55615-469-0
Mini tutorial and complete reference for programmers.

The Programmer's PC Sourcebook, 2nd Ed.
Microsoft Press
Thom Hogan
$39.95 ($54.95 Canada)
ISBN 1-55615-391-X
Reference tables for programmers.

Programming Windows, 2nd Ed.
Microsoft Press
Charles Petzold
$29.95 ($39.95 Canada)
ISBN 1-55615-264-7
Reference for programmers.
Available Summer, 1992

Programming in Windows: New Ed.
Que
Tim Farrell and Runnoe Connally, 1991
For all Windows programmers.

Programming in Windows 3.1
Que
Tim Farrell and Runnoe Connally
Tips and techniques for programmers.
Pending software release

Windows 3: A Developer's Guide
M & T Books
Jeffrey M. Richter, 1991
For experienced Windows users.

Windows 3.1 Power Tools: For Windows 3.0 and 3.1, 2nd Ed.
Bantam Computer Books
The LeBlond Group, April 1992
For intermediate to advanced Windows programmers.

Windows Programmer's Reference
Que
James McCord, 1991
For experienced Windows programmers.

Windows 3.1 Programmer's Reference
Que
James McCord
Reference guide for experienced Windows programmers.
Pending software release

Windows Programming: An Introduction
Osborne McGraw-Hill
William Murray, 1990
For novice Windows programmers.

Miscellaneous

Introduction to Microsoft Windows 3.0
Computer Consultants International, 1990
Training manual for instructors and beginning students. Call 604/463-7510 for the nearest reseller.

Moving to Windows
HDC (available through Mastering Computers), 1991
Examples in reducing standardization costs, and case studies from corporations standardizing in Windows.

Windows 3.1 Networking
New Riders Publishing
1992; pending software release.
"How to" guide for networking in Windows.

Windows Shopper's Guide
WhiteFox Communications, Inc.
PO Box 7125
Beaverton, OR 97007-9938
800/669-5612; 503/645-8642 (fax)
A catalog for Windows applications and services.

Publishers' Numbers

Bantam Books
800/347-7828

Howard W. Sams & Co. (SAMS)
317/573-2500
800/448-3804 (fax)

IDG Books
800/282-6657

Mastering Computers, Inc.
602/998-7500

Microsoft Press
800/MSPRESS

New Riders Publishing (NRP)
317/573-2500
317/573-2500

Osborne McGraw-Hill
510/548-2805

Que Corporation
317/573-2500
800/448-3804 (fax)

Sybex
800/227-2346

WhiteFox Communications
503/629-5612

Newsletters and Magazines

ACKnowledge, The Windows Letter
ACKnowledge Press and the Windows Technology Group
144 Talmadge Road
Mendham, NJ 07945
201/543-2273; 201/543-6033
Contact: Carole Patton
Interviews with key Windows people, Microsoft strategy, and a watch on Windows applications.

For Windows Only
PJM Communications
PO Box 6172
Burbank, CA 91510
818/563-4285 or 800/248-4396
818/955-9832 (fax)
Tips and tricks for Windows.

ICON
International GUI Group
PO Box 20150
Houston, TX 77225-0150
713/432-1000
Contact: Larry Sands
BBS # to be established
CompuServe ID# 71630,407
Tips and tricks, events, new product information.

Inside Microsoft Windows
The Cobb Group
Ziff Communications
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220
800/223-8720
Tips and tricks for Windows.

Microsoft Systems Journal
Microsoft
PO Box 1903
Marion, OH 43305
800/669-1002 or 614/382-3322
Tips and tricks, questions and answers for Microsoft systems products.

The Graunke Report: Windows Tips and Tricks
Mastering Computers
11000 North Scottsdale Rd., Ste. 175
Scottsdale, AZ 85254
602/998-7500
Contact: Lesa Harpole

View, The WPMA Newsletter
Windows and Presentation Manager Association
5201 Great American Pkwy.
Ste. 320
Santa Clara, CA 95054
408/562-6065; 408/562-6068
Contact: Holden Aust
Product release information, relevant current events and a focus on communication between developers and corporate users.

Windows Flash
Corporate Software
275 Dan Road
Canton, MA 02021
617/821-4500; 800/677-4000
A bulletin covering fast breaking news, issues, and developments with Windows.

Windows Journal (formerly WUGNET Journal)
WUGNET Publications, Inc.
PO Box 1967
Media, PA 19063
215/565-1861
Contact: Joel Diamond
CompuServe ID: 76702,1023
Technical resource for Windows users and developers. Tips and tricks, new product information.

Windows Magazine
CMP Publications, Inc.
600 Community Dr.
Manhasset, NY 11030
516/562-5370
Circulation: Kathy Shay
New product news, Windows strategy, tips and tricks, events.

Windows Watcher
CompuThink
15127 NE 24th, Suite 344
Redmond, WA 98052-5530
206/881-7354
Contact: Lori White, Circ. Man.
CompuServe ID: 71337,2052
News watch, industry watch, strategy information, new products, Windows marketing, events.

Appendix

B Hardware Compatibility List

This appendix contains a list of machines, video displays, keyboards, pointing devices, networks, and printers that are compatible with Windows 3.1.

For a list of hardware manufacturers who offer support for Windows running on their systems, see "Hardware Support" in Appendix A, "Windows Resource Directory."

Contents of this appendix

Windows Compatible Machines 488
Windows Compatible Displays 495
Windows Compatible Keyboards 495
Windows Compatible Pointing Devices 495
Windows Compatible Networks 495
Windows Compatible Printers 496

Windows Compatible Machines

:3C! PRO 3
:3C! PRO4
:3C! Slim 386 C
:3C! Slim 386 X
:3C! Slim 486
:3C! Slim 486 X
Acer 1100/33
Acer 1120SX
Acer 1125E
AcerAnyware 386S
Acerframe 3000SP (Acer 3235)
AcerMate 386/33
AcerPower 386SX
AcerPower 486/33
AcerPower 486SX
AcerPower 500/20
AcerPower 500/33
Acros 340
Acros 400
Acros 525
Acros 615
Acros 620
Acros 723
Add-X AS 316sx
Add-X AS 320sx
Add-X AS 450
Add-X AS 450
ADLER ROYAL P45
AEG Olystar 70 H20V
Alcatel 701/33
Alcatel 7100/33
Alcatel 801/20
Alcatel 8100/20
Alcatel 8551
Alcatel 901/33
Alcatel 9300/33
Alcatel APC 701/33
Alcatel APC 7100/33
Alcatel APC 801/20
Alcatel APC 8100/20
Alcatel APC 8551
Alcatel APC 901/33
Alcatel APC 9300/33
ALIF 386DX25 ti
ALIF 386DX33/64c ti
ALIF 386SX16 ti
ALIF 433DX128 ti
ALR Business STATION 386/33
ALR Business STATION 486/33
ALR Business STATION 486/50
ALR Business STATION 486ASX/20
ALR Business STATION 486SX/20
ALR BusinessVEISA 386/33
ALR BusinessVEISA 486/33
ALR BusinessVEISA 486/50
ALR MPS 386/33
ALR MPS 486/33
ALR MPS 486/50
ALR MPS 486ASX/20
ALR PowerFlex 386SX/16
ALR PowerFlex 386SX/20
ALR PowerFlex 486ASX/20
ALR PowerFlex 486DX/25
ALR PowerFlex Flyer 386SX/25
ALR PowerFlex Flyer 486SX/20
ALR PowerPRO VM 386/33
ALR PowerPRO VM 486/33
ALR PowerPRO VM 486/50
ALR PowerPRO VM 486ASX/20
ALR VIP 386SL/25 Notebook
ALR VIP M 386SX/20 Notebook
ALR VIP M 486SX/20 Notebook
Amkly 386/25E
Amkly 386/33E
Amkly 486/25E
Amkly 486/33E
Amkly Station 386SX/20
Amkly Station 486/33
Amkly Station 486SX/20
Amkly Station 486SX/25
Amstrad PC-5286HD40
Apricot FTs 386-25
Apricot LS 386SX-20
Apricot XEN-LS 386SX 16M
Apricot XEN-LS 386SX-20
Apricot XEN-LS 486SX 20 MHz
ARES 486-33
ASEM DS 286 16 MHz
ASEM DS 386E-20 MHz
ASEM SP 486E
ASI 386DC-25
ASI 386DX-33
ASI 386SX-16
ASI 386SX-20
ASI 486DX-33 ISA
ASI 486SX-20
AST Advantage! 386SX/20
AST Advantage! 486SX/20
AST Advantage! 486SX/25
AST Bravo 386/25S
AST Bravo 386SX/20
AST Bravo 486/25
AST Bravo 486/33
AST Premium 386/33
AST Premium 386SX/20
AST Premium 486/33
AST Premium Exec 386SX/20
AST Premium Exec 386SX/25
AST Premium Exec 386SX/25 Color
AST Premium II 386/25
AST Premium II 386/33
AST Premium II 386SX/20
AST Premium II 486/33
AST Premium II 486SX/20
Astrix 486/33 Complete
AT&T 6386/25 WGS
AT&T 6386E WGS
Austin 386/25VL Winstation
Austin 386/33 Multi-Media
Austin 386/33 Multi-Media Winstation
Austin 386/33 Winstation
Austin 386/33-40VL Winstation
Austin 386/40 Winstation
Austin 386/SX16-20VL Winstation
Austin 386SX (16/20/25) Winstation
Austin 486/33E Wintower
Austin 486/33I Winstation
Austin 486/33VLI Winstation
Austin Bell PAT48PX-433c/256 KB Cache
Beaver Avanti 025
Brett 386DX-25
Brett 386DX-33/64K Cache
Brett 386SX-16
Brett 386SX-20
Brett 386SX-25
Brett 486DX-33:256K Cache
Brett 486SX-20 MHz
Brother BCN3386DX/33C
Brother BCN5486SX
Canon HS-20H
Canon US-20Y
Chicony NB5500SXL
Chicony NB6600SL
Chips MK38600DX
Chips MK38605DX
Commodore C286-12

Windows Compatible Machines, continued

Commodore C286-16
Commodore C286-LT
Commodore C386-25
Commodore C386-25C
Commodore C386SX-16
Commodore C386SX-20
Commodore C386SX-20C
Commodore C386SX-LT
Commodore C486-25C
Commodore C486-33C
Commodore C486SX-20
Commodore PC 60-III
Commodore PC 70-III
Commodore T386-25C
Commodore T486-25C
Compaq 286N
Compaq Deskpro 286N
Compaq Deskpro 386/16
Compaq Deskpro 386/20e
Compaq Deskpro 386/25
Compaq Deskpro 386/25e
Compaq Deskpro 386/25M
Compaq Deskpro 386/33
Compaq Deskpro 386/33L
Compaq Deskpro 386N
Compaq Deskpro 386s
Compaq Deskpro 386s/20
Compaq Deskpro 386s/20N
Compaq Deskpro 486/25
Compaq Deskpro 486/33L
Compaq Deskpro 486/33M
Compaq Deskpro 486/50L
Compaq Deskpro 486s/16M
Compaq Deskpro 486s/25M
Compaq LTE 386/20
Compaq LTE 386SL/20
Compaq LTE LITE 20
Compaq Portable 386/20
Compaq Portable 486c
Compaq Portable III
Compaq SLT 286
Compaq SLT 386/20
Compaq Systempro 486/33
Compaq Systempro/LT 386/25
Compaq Systempro/LT 486S/25
CompuAdd 316s
CompuAdd 320SC
CompuAdd 325LP
CompuAdd 325SC
CompuAdd 333
CompuAdd 333LP
CompuAdd 420S
CompuAdd 425
CompuAdd 433E
CompuAdd 433I
Compudyne 286/12
Compudyne 3DN/25
Compudyne 3DX/25
Compudyne 3DX/33
Compudyne 3SX/16
Compudyne 3SX/16HM
Compudyne 3SX/20
Compudyne 3SX/20HM
Compudyne 4DX/33
Compudyne 4DX/33 Y
Compudyne 4DX/33LB
Compudyne 4DX/33LBTMC
Compudyne 4SX/20
Cube 386 ATX Series
Cube 486 ATX Series
Cumulus GLC 386DX-25 w/ cache
Cumulus GLC 386DX-25 without cache
Cumulus GLC 386DX-33 w/ cache
Cumulus GLC 386SX w/cache
Cumulus GLC 386SX-16 w/ cache
Cumulus GLC 386SX-16 without cache
Cumulus GLC 386SX-20 w/ cache
Cumulus GLC 386SX-20 without cache
Cumulus GLC 386SX25 w/cache
Cumulus GLC 486-20 w/ cache
Cumulus GLC 486-20 without cache
Cumulus GLC 486DX-25 w/ cache
Cumulus GLC 486DX-25 without cache
Cumulus GLC 486DX-33
Cumulus GLC 486DX-33 w/ cache
Cumulus GLC 486DX-33 without cache
Cumulus GLC 486SX-20 w/ cache
Cumulus GLC 486SX-20 without cache
Cumulus GLC 487SX-20 w/ cache
Cumulus WorkBox 386/DX/25
Cumulus WorkBox 386/SX/16
Cumulus WorkBox 386/SX/20
Cumulus WorkBox 486/SX/20
Daewoo CPC-2000
DAK Industries BSR 386SX/16
Data General Dasher II-386Sx/20A
Data General Dasher II-486/eeTE
Data General Dasher II-486Sx/20A
Data General Walkabout /320
Data I/O Club 286
Datamedia NETmate/sx20i
Datamedia NETmate/sx420i
Datamedia NETmate/sx425i
Datapoint Svenska DPC386SX
Datatech Grafika 3D
Datatech Grafika 3E
DECpc 316sx
DECpc 320 SX Notebook
DECpc 320p
DECpc 333 Portable
DECpc 420sx
DECpc 433T
DECStation 210
DECStation 212
DECStation 212lp
DECStation 316
DECStation 316+
DECStation 316sx
DECStation 320
DECStation 320sx
DECStation 325c
DECStation 425
DECStation 425c
Dell PowerLine 420DE
Dell PowerLine 420SE
Dell PowerLine 425DE
Dell PowerLine 425SE
Dell PowerLine 433DE
Dell PowerLine 433SE
Dell PowerLine 450DE
Dell PowerLine 450SE
Dell System 210
Dell System 310
Dell System 316
Dell System 316SX
Dell System 320LT
Dell System 320LX
Dell System 320N
Dell System 320N+
Dell System 320SX
Dell System 325
Dell System 325D
Dell System 325P
Dell System 333D
Dell System 333P
Dell System 433P
Dell System 486D/20
Dell System 486D/25
Dell System 486D/33
Dell System 486P/16
Dell System 486P/20
Dell System 486P/25

Windows Compatible Machines, continued

Dell System 486P/33
Donatec 486DX33 Cache
DTK PEM-4001 80386-40 w/cache
DTK PKM-3330Y 80486-33 w/cache
Elonex PC 325XB
Elonex PC-325XM
Elonex PC-333B
Elonex PC-333M
Elonex PC-420SXB
Elonex PC-420SXM
Elonex PC-450B
Elonex PC-450M
Eltech Research PCdel 4200 486SX/20
Eltech Research PCdel 4300 486/33
Eltech Research PCdel 9970 386/33
Epson EISA Series 486/33 Tower
Epson EISA Series 486/50
Epson EISA Series 486SX/25
Epson EL Plus
Epson EL2 Plus
Epson EL3/33
Epson EL3s Plus
Epson EL4s Plus
Epson Equity 320SX Plus
Epson Equity 386/20
Epson Equity 386/25
Epson Equity 386/25 Plus
Epson Equity 386/33 Plus
Epson Equity 386SX
Epson Equity 386SX Plus
Epson Equity 386SX/16 (diskless)
Epson Equity 386SX/20 Plus
Epson Equity 486SX/25 Plus
Epson Equity 486SX/25 Plus
Epson Express Station
Epson NB-SL/20
Epson NB-SL/25
Epson NB-SL/25C
Epson NB3s (UK)
Epson NB3s (USA)
Ergo Electronics Macro-3/40
Ergo Electronics Macro-4/33
Ergo Electronics Macro-4X/20
Ergo Electronics Microtower-3/25
Ergo Electronics Microtower-3/40
Ergo Electronics Minipro-3/25
Ergo Electronics System-3/40
Everex AGI 386/33 EV-1851
Everex AGI 486/33 EV-18134
Everex Step 386/20 EV-1816
Everex Step 386/25 EV-1816
Everex Step 386/33 EV-18108
Everex Step 486/33 EV-18113 12-slot EISA
Everex Step 486/33 EV-1878
Everex Step 486/33 EV-8038
Everex Step 486SX/20 EV-18108+EV-18116
Everex Step Server 386/33 EV-18108
Everex Tempo 286/16 EV-1824
Everex Tempo 386/25 EV-1869
Everex Tempo 386SX/20 EV-8048
Everex Tempo Modular 386/25
Everex Tempo Modular 386/33
Everex Tempo Modular 486sx/25
Everex Tempo-C 286/16 EV 18122
Everex Tempo-C 386/33 EV-18103
Everex Tempo-C 386sx/20 EV18121
Everex Tempo-C 486sx/20 EV 18103
Flytech Carry-1 80386SX
Flytech Carry-1 80386SX
Flytech Carry-1 FT-5320
Fujitsu MY320
Fujitsu MY333
Fujitsu MY420
FUJITSU S-420
FUJITSU S-4863
Future AT 386DX-40 MHz
Future AT 386SX-20 MHz
Future AT 486 EISA 33 MHz
Future AT 486DX-33
Future AT 486SX-20
G2 HL 386/33C
G2 SL 286/16
G2 SL 386/20
G2 SL 386/33
G2 SL 486/25
G2 SL 486/33C
Gateway 2000 386-25 ISA
Gateway 2000 386-33C ISA
Gateway 2000 386SX-20
Gateway 2000 386SX/16
Gateway 2000 486-33 EISA
Gateway 2000 486-33 ISA
GRiD 1720
GRiD 1750
GRiD 1755
GRiD 286 is
GRiD 286-MFP
GRiD 286N
GRiD 325sc
GRiD 386 is-16
GRiD 386 is-20
GRiD 386 is-25
GRiD 386mc
GRiD 386NX
GRiD 386sx-MFP
GRiD 386sx-MFP20
GRiD 486ei-25/SRV
GRiD 486ei-33
GRiD ADT/425se
GRiD MFP-320s
GRiD MFP/420s
GS 316M
GS 31IE
GS 31IM
GS 325S
GS 340D
GS 386 DXH/40
GS 386 SL/20
GS 386S/16
GS 386SX-M 125
GS 386SXM/16
GS 425S
GS 486SX/25
GS 610
GS 725
GS Mastersport286
Hantarex Vegas 3625N
Hantarex Vegas 3633
Hantarex Vegas 386
Hantarex Vegas 4620S
Hantarex Vegas 4633
Hauppauge D3/33-01
Hauppauge D3/33-02
Headstart 486/20XP
Headstart 486SX
HERMES H 260S
HERMES H 320
HERMES H 400
HERMES H 400 E
HERMES H 410
HERMES WALKSTATION 386 33
HERMES WALKSTATION 386 SX
HERMES WALKSTATION SX20
Hertz 386/33
Hertz 386/SX
Hertz 486/33

Windows Compatible Machines, continued

Hewlett-Packard Vectra 286/12
Hewlett-Packard Vectra 386/16N
Hewlett-Packard Vectra 386/20N
Hewlett-Packard Vectra 386/25
Hewlett-Packard Vectra 386s/20
Hewlett-Packard Vectra 486/20
Hewlett-Packard Vectra 486/25
Hewlett-Packard Vectra 486/33T
Hewlett-Packard Vectra 486s/20
Hewlett-Packard Vectra ES/12
Hewlett-Packard Vectra ES/12
Hewlett-Packard Vectra QS/16
Hewlett-Packard Vectra QS/16N
Hewlett-Packard Vectra QS/16s
Hewlett-Packard Vectra QS/20
Hewlett-Packard Vectra RS/16
Hewlett-Packard Vectra RS/20c
Hewlett-Packard Vectra RS/20c
Hewlett-Packard Vectra RS/25c
IBM 30/286 w/Snapin 386
IBM 50 w/Snapin 386
IBM 60 w/Snapin 386
IBM 8540
IBM 8550
IBM 8550Z
IBM 8557-SX
IBM 8570-486
IBM 8580
IBM AT /Inboard 386 AT
IBM AT 6MHz w/Snapin 386
IBM PS/1 286
IBM PS/2 55 Note
ICL CS386s
ICL CX386s/25
ICL DRS M75
ICL FX486s
ICL MX486
ICL NB386s
Intel 300SX20
Intel 301Z
Intel 302-25
Intel 386/33H
Intel 401
Intel 403E
Intel 403E 33MHz 486CPU EISA
Intel LP386SX-20
Intel LP486/33E
Intel X486E
ITOS 286/12
ITOS 386SX/16
ITOS 386SX/20
ITOS 486/33
KLH 386DX-40
KLH 486DX-33
KLH 486SX
Kontron IPLite Color 486/33
Kontron PIC 386DX/20
L40SX
Leading Edge D3/33
Leading Edge D3/33MT
Leading Edge D3/SX20c
Leading Edge D4/33
Leading Edge D4/SX20
Leading Edge MT33
Leading Edge N3/SX25
Leading Technology PC Partner LX
Leading Technology PC Partner QI
Leading Technology PC Partner VX
Leading Technology PC Partner XI
Leading Technology PC Partner ZI
Leading Technology PC Partner ZX
Linear Dynamics 386DX-40
Linear Dynamics 486DX-33
Linear Dynamics 486SX
Lodestar Computers 386-33
Lueski G-M340V-130-4M
Lueski G-MT 433-210-4M
Maxstation 486/20XP
Maxstation 486SX
Maxum 3020
Maxum 3133
Maxum 3140
Maxum 4133
Memorex Telex 8036
Memorex Telex 8057
Memorex Telex 8067
Memorex Telex 8080
Memorex Telex 8090
Memorex Telex 8257
Memorex Telex 8267
Memorex Telex 8280
Mind 386-25
Mind 386-33
Mind 386-40
Mind 386SX-16
Mind 386SX-25
Mind 486-25
Mind 486-33
Mind 486-33 EISA
Mind 486-50
Mind 486SX-20
Mind 486SX-25
Mitac 3026E
Mitac 3027F
Mitac 3060D
Mitac 3060F
Mitac 3060G
Mitac 3062E
Mitac 3062F
Mitac 3072F
Mitac 4060F
Mitac 4060G
Mitac D3033
Mitac D4033
Mitac D4125
Mitsubishi MP-386
Model 25SX
NCR System 3000 Model 3210
NCR System 3000 Model 3320 (3420)
NCR System 3000 Model 3320 (3431)
NCR System 3000 Model 3335
NCR System 3000 Model 3345
NEC PowerMate 286/12
NEC PowerMate 286C
NEC PowerMate 386/20
NEC PowerMate 386/33E
NEC PowerMate 386/33i
NEC PowerMate SX/16
NEC PowerMate SX/20
NEC UltraLite 286V
NEC UltraLite SL/20
Nevis SVC 486/33MHz
Normerel Comete
Normerel Hexagone
Normerel Jupiter 386
Normerel Jupiter 486
Normerel Saturn 386
Normerel Saturn 486
Normerel Saturn 486DX
Northgate 386SX/25
Northgate 486DX/33
Northgate EISA 486
Northgate Elegance SP
Northgate Elegance ZXP
Northgate Slimline SP 486DX/33
Olivetti CP 486
Olivetti CP 486/33
Olivetti Laptop D33
Olivetti Laptop S20
Olivetti LSX5010
Olivetti LSX5020
Olivetti M300-25
Olivetti M380
Olivetti M480-10
Olivetti M480-30
Olivetti M486
Olivetti Notebook S20

Windows Compatible Machines, continued

Olivetti Notebook V16
Olivetti P500
Olivetti P500-10
Olivetti P560
Olivetti P750
Olivetti P750-S
Olivetti P800
Olivetti PC Pro 290 SP
Olivetti PC Pro 486SX
Olivetti PCS 386SX
Opus Panther SX
Opus PC 4-286/16
Opus PC 5-386
Opus PC 7/40
Opus PCSX
Opus Powerstation SX
Osborne 386-33
Osborne 386SX-20
Osborne 386SX-25
Osborne 486-33 EISA
Pac 286 8Mhz
Packard Bell 486-33
Packard Bell Multimedia III
PC 340
PC Brand 286-12 system
PC Brand 386/33 Cache System
PC Brand 386/SX-16 System
PC Brand 386/SX-20 System
PC Brand 386/SX-25 Notebook
PC Brand 386sx-20 Cache System
PC Brand 386SX-20 System
PC Brand 386SX-20 System
PC Brand 386sx-25 Cache System
PC Brand 386sx-25 System
PC Brand 486-33 System
PC Brand 486-50 System
PC Brand 486/33 EISA Cache System
PC Brand 486/50 Cache System
PC Brand 486/50 EISA Cache System
PC Brand 486sx-20 Cache System
PC Brand 486sx-20 System
Philips MPC
Philips PCD 215
Philips PCD 315
Philips PCL 304
Pionier P386 C
Pionier P386 X
Pionier P486 CT
Pionier P486 X
Poly 12/486-33/sf
Poly 486-33E
Poly 486-50V
Poly 486/25sx
Poly 486/33E (EISA)
Positive PC-1000 286-12
Positive PC-1000 386sx-20
Positive PC-1000 386sx-20c
Positive PC-1000 386sx-25
Positive PC-1000 386sx-25c
Positive PC-1000 486-33
Positive PC-1000 486-50
Positive PC-1000 486sx-20c
Positive PC-2000 286-12
Positive PC-2000 386sx-20
Positive PC-2000 386sx-20c
Positive PC-2000 386sx-25
Positive PC-2000 386sx-25c
Positive PC-2000 486-33
Positive PC-2000 486/50
Positive PC-2000 486sx-20c
Positive PC-3000 386/sx-20
Positive PC-3000 386sx-25
Positive PC-3000 486/50
Positive PC-325
Positive PC-340
Precision 386sx/20 ISA
Precision 486/33 ISA
Premier 386DX-40
Premier 486DX-33
Premier 486SX
Professional 486/20XP
Professional 486SX
Proj 559R
QNIX Omni 386sx/20c
RadiSys EPC-2
RadiSys EPC-3
RadiSys EPC-4
RadiSys EPC-5
RadiSys EPC-6
RadiSys EPC-7
RadiSys RIC 386
Reply Model 32 386/33c
Reseach Machines M-Series PC 286
Reseach Machines M-Series PC 386
Research Machines Nimbus AX/2
Samsung Desk Master 386S/25N (SD740)
Samsung Desk Master 486S/25 (SD923N)
Sanyo MBC-17 PLUS
Sanyo MBC-17LX
Sanyo MBC-17NB
Sanyo MBC-18NB
Sanyo MBC-18PLUS
Sanyo MBC-18SXMT
Sanyo MBC-27MT
Seanix ASI 9000 (386DX)
Seanix ASI 9000 (386SX)
Seanix ASI 9000 (486DX)
Siemens Nixdorf 8810 /30
Siemens Nixdorf 8810 /50
Siemens Nixdorf PCD -3M
Siemens Nixdorf PCD-2B
Siemens Nixdorf PCD-2M
Siemens Nixdorf PCD-2N
Siemens Nixdorf PCD-3Bsx
Siemens Nixdorf PCD-3Msx
Siemens Nixdorf PCD-3Nsl
Siemens Nixdorf PCD-3Nsx/16
Siemens Nixdorf PCD-3Nsx/20
Siemens Nixdorf PCD-3Rsx
Siemens Nixdorf PCD-3T
Siemens Nixdorf PCD-3Tsx
Siemens Nixdorf PCD-4Gsx
Siemens Nixdorf PCD-4M
Siemens Nixdorf PCD-4Msx
Siemens Nixdorf PCD-4Rsx
Siemens Nixdorf PCD-4T
Siemens Nixdorf PCD-4Tsx
Siemens Nixdorf PCE-4C
Siemens Nixdorf PCM-3Dsx/20
Siemens Nixdorf PCM-4T
Siemens Nixdorf PWS M30
Siemens Nixdorf PWS M50
Siemens Nixdorf PWS M60
Siemens Nixdorf PWS M70
Steiner POWER STC i80386DX-33 MHz
Steiner POWER STC i80386SX-16/20 MHz
Steiner POWER STC i80386SX/25 MHz
Steiner POWER STC i80486-33 ISA
Steiner POWER STC i80486-33 MHz
Steiner POWER STC i80486SX-20/25 MHz
SWAN 286/16 Desktop
SWAN 386/33M Desktop
SWAN 386/33M Tower
SWAN 386SX/16 Desktop
SWAN 386SX/20
SWAN 386SX/20 Notebook
SWAN 386SX/25 Desktop
SWAN 486/33 Desktop
SWAN 486/33 Tower

Windows Compatible Machines, continued

SWAN 486/33E Desktop
SWAN 486/33M Desktop
SWAN 486/33M Tower
SWAN 486SX/20 Desktop
SWAN 486SX/20 Tower
SWAN 486SX/20M Desktop
SWAN 486SX/20M Tower
SWAN 486SX/25 Desktop
SWAN 486SX/25 Tower
SWAN 486SX/25M Desktop
SWAN 486SX/25M Tower
SWAN 487SX/20 Desktop
SWAN 487SX/20 Tower
SWAN 487SX/25 Desktop
SWAN 487SX/25 Tower
SYNCOMP MEGA+386i-33 PC
SYNCOMP MEGA+386i-40 PC
SYNCOMP MEGA+486i-33 PC
SYNCOMP MICRO 386i-33 PC
SYNCOMP MICRO 386i-40 PC
SYNCOMP MICRO SXi-20 PC
SYNCOMP MICRO386SXi-16 PC
SYNCOMP MICRO 386SXi-25 PC
SYNCOMP MINI 386i-33 PC
SYNCOMP MINI 386i-40 PC
SYNCOMP MINI 486i-33 PC
SYNCOMP MINI 486i-33 PC
SYNCOMP MINI 486SXi-20 PC
SYNCOMP MINI 486SXi-20 PC
SYNCOMP MINI SXI-16 PC
SYNCOMP NETerm 386SXi-16 PC
SYNCOMP NETerm 386SXi-20 PC
SYNCOMP ProSport Portable
Tandem CP 386SX/20
Tandem EP 486/25
Tandon 286/N
Tandon 386/20
Tandon 386/25
Tandon 386/33
Tandon 386/33 Tower
Tandon 386/N
Tandon 386/N-16
Tandon 386/N-20
Tandon 386/N-25
Tandon 486/33
Tandon 486/50 Desktop
Tandon 486/50 Tower
Tandon Corp. MCS 286/12
Tandon Corp. MCS 386sx/20
Tandon Corp. MCS 386sx/20c
Tandon Corp. MCS 386sx/25
Tandon Corp. MCS 386sx/25c
Tandon Corp. MCS 486/33
Tandon Corp. MCS 486/50
Tandon Corp. MCS 486sx/20c
Tandon Corp. MCS Pro 386/33
Tandon Corp. MCS Pro 486/33
Tandon Corp. MCS Pro 486/50
Tandon Corp. MCS Pro 486sx/20
Tandon Corp. Option 286/12
Tandon Corp. Option 386sx/20
Tandon Corp. Option 386sx/20c
Tandon Corp. Option 386sx/25
Tandon Corp. Option 386sx/25c
Tandon Corp. Option 486/33
Tandon Corp. Option 486/50
Tandon Corp. Option 486sx/20c
Tandon Corp. Option Pro 386/33
Tandon Corp. Option Pro 486/33
Tandon Corp. Option Pro 486/50
Tandon Corp. Option Pro 486sx/20
Tandon Notebook 386sx/16
Tandon Notebook 386sx/25
Tandon PacII 386sx-25c
Tandon PacII 386sx/20
Tandon PacII 486/33
Tandon PacII 486/50
Tandon Tower 386/33
Tandon Tower 486/33
Tandy 2500SX
Tandy 2500SX/20
Tandy 2500SX/25
Tandy 2500XL
Tandy 2500XL2
Tandy 2810
Tandy 2820
Tandy 3000NL
Tandy 3810
Tandy 3820
Tandy 4000
Tandy 4000LX
Tandy 4000SX
Tandy 4016DX
Tandy 4016SX
Tandy 4020LX
Tandy 4020SX
Tandy 4025LX
Tandy 4820SX/T
Tandy 4825SX
Tandy 4833LX/T
Tandy 4850
Tandy 5000MC
Texas Instruments TravelMate 3000
Texas Instruments TravelMate 3000 WinSX
Tiki-386 DX-33
Tiki-386 SX
Tiki-386 SX Light
Tiki-486 SX-20
Tiki-486-33e
Tiki-486-33i
Tiko PS 325
Tiko PS 325C
Tiko PS 340C
Tiko PS 420C
Tiko PS 433C
Toshiba DynaBook V386
Toshiba J-3100ZX
Toshiba T1200XE
Toshiba T2000
Toshiba T2000SX
Toshiba T2200SX
Toshiba T3100e
Toshiba T3100SX
Toshiba T3200
Toshiba T3200SX
Toshiba T3200SXC
Toshiba T3300SL
Toshiba T4400SX
Toshiba T5200
Toshiba T5200C
Toshiba T6400
Toshiba T6400SXC
Toshiba T8500
Tri Gem SX386E

* From Toshiba: There is a known incompatibility with the Trackball mouse plugged into the PS/2 port on Toshiba T2200SX machines with BIOS versions earlier than 1.20. Users should either plug their mouse into the Com port or obtain a BIOS upgrade from Toshiba, 800/999-4273.

Windows Compatible Machines, continued

TriGem 286E+
TriGem 486XE
TriGem SX386/20NP
TriGem SX386M+
TriGem SX486E
TRIUMPH ADLER P45
TRIUMPH ADLER TA DARIO 386 SX
TRIUMPH ADLER TA DARIO 486
TRIUMPH ADLER TA DARIO 486 SX
TRIUMPH ADLER TA DARIO 486 SXE
TRIUMPH ADLER TA DARIO S/20
TRIUMPH ADLER TA WALKSTATION 386 33
TRIUMPH ADLER TA WALKSTATION 386 SX
TRIUMPH ADLER TA WALKSTATION SX20
Tulip AT 386/25
Tulip NB 386sx
Tulip TR 486e
Tulip Vision 1
Tulip Vision Line DC Compact
Tulip Vision Line DC/DT 286
Tulip Vision Line DC/DT 386SX
Tulip Vision Line DE/TR 486dx/e-33
Tulip Vision Line DE/TR 486sx/e-25
Tulip Vision Line DT 486dx/e-33
Tulip Vision Line DT 486sx/e-25
Tulip Vision Line TR 486sx/e
Tulip Vision Line WS 386sx
Ultra-Comp 386-25 Non-Cache Power
Ultra-Comp 386-33 Power
Ultra-Comp 386SX-20 Power
Ultra-Comp 486-33 EISA Power
Ultra-Comp 486-33 ISA Power
Ultra-Comp 486SX-20 Power
Unisys Option LWS 286
Unisys Option LWS 386
Unisys Option PP 386DX
Unisys Option PP 386SX
Unisys Option PW2 3162
Unisys Option PW2 3203
Unisys Option PW2 3206
Unisys Option PW2 3256
Unisys Option PW2 3336
Unisys Option PW2 3336 80386/DX
Unisys Option PW2 4336
Unisys Option PW2 500/16A
Unisys Option PW2 500/20A
Unisys Option PW2 600/20SX
Unisys Option PW2 800/20
Unisys Option PW2 800/20C
Unisys Option PW2 800/25A
Unisys Option PW2 800/33A
Unisys Option PW2 800/486-25A
Unisys Option PW2 Advantage Model 3163
Unisys Option PW2 Advantage Model 3256
Victor V286/16M
Victor V286/16MW
Victor V286D
Victor V386-25M
Victor V386-33M
Victor V386/20MWX
Victor V386/20MX
Victor V386/25MT
Victor V386/25MW
Victor V386/33MT
Victor V386DSX
Victor V486/20MTX
Victor V486/20MWX
Victor V486/20MX
Victor V486/33M
Victor V486/33MT
Victor V486/33MW
Victor V486/33SFE
Victor Vicki 286
Victor Vicki 386sx
Video Technology Laser 286/X
Video Technology Laser 386
Video Technology Laser 386SXE
Video Technology Laser 486
Video Technology Laser 486SX
Video Technology Laser LT321
Video Technology Laser PAL 386SX
Viglen Genie 3DX33
Viglen Genie 3SX25
Viglen Genie 4DX33
Viglen Genie 4SX20
Wand PC 362/25SC
Wang EC 480/25C
Wang EC 480/33C
Wang Exec 3000
Wang Exec 3051
Wang EXEC 3100
Wang EXEC 3110
Wang Exec 3120
Wang EXEC 3120
Wang Exec 3131
Wang EXEC 3131
Wang Exec 3450
Wang EXEC 3450
Wang Exec 3500
Wang Exec 3501
Wang MC 350/16S
Wang PC 240
Wang PC 280
Wang PC 280/20
Wang PC 321/16S
Wang PC 321/20S
Wang PC 350/33C
Wang PC 350/40C
Wang PC 361/20S
Wang PC 361/25SC
Wang PC 380/33C
Wang PC 381
Western 386SX25VC
Western 486H33
Western386V33
WYSE Decision 386/25
WYSE Decision 386/40
WYSE Decision 386SX/16 (slimline)
WYSE Decision 386sx/16s
WYSE Decision 386sx/16s (slimline)
WYSE Decision 386sx/20
WYSE Decision 486/25
WYSE Decisionmate SX Laptop (slimline)
WYSE pc Model 3016sx
XYCOM 8450-5164
Zenith Data Systems Cat 20
Zenith Data Systems Mastersport 286
Zenith Data Systems Mastersport 386SX
Zenith Data Systems Orion 486DX 25
Zenith Data Systems Otter II
Zenith Data Systems Supersport SX
Zenith Data Systems Z-386/25
Zenith Data Systems Z-386SX
Zeos 386-25CA
Zeos 386-25NC
Zeos 386-35MWS
Zeos 486-33E
Zeos 486-33U
Zeos 486SX-20
Zeos MXA386-33
Zeos Notebook 386SX16

Windows Compatible Displays

Any display 100% compatible with those listed below. Other displays may be compatible using a driver provided in the Windows Driver Library or from the vendor.

Compaq Portable Plasma
EGA (enhanced graphics adapter) with high-resolution color display
EGA (enhanced graphics adapter) with high-resolution color display (black & white)
EGA (enhanced graphics adapter) with high-resolution monochrome display
Hercules with-high resolution monochrome display
IBM 8514/a
IBM MCGA (multicolor graphics array)
IBM XGA
Olivetti/AT&T monochrome or PVC Display
Super VGA
TIGA
VGA
VGA with Monochrome display
Video 7 (FastWrite, 1024i, VRAM, VRAM II)

Keyboards

Any keyboard 100% compatible with those listed below. Other keyboards may be compatible using a driver provided in the Windows Driver Library or from the vendor.

All AT type keyboards (84 - 86 keys)
AT&T "301" keyboard
AT&T "302" keyboard
Enhanced 101 or 102 key US and Non US keyboards
Hewlett-Packard Vectra keyboard (DIN)
Olivetti 101/102 A keyboard
Olivetti 83 key keyboard
Olivetti 86 Key keyboard
Olivetti M24 102 key keyboard
PC-XT 83 key keyboard
PC/XT - Type keyboard (84 keys)

Pointing Devices

Any pointing device 100% compatible with those listed below. Other pointing devices may be compatible using a driver provided in the Windows Driver Library or from the vendor.

Genius serial mouse
HP Mouse (HP-HIL)
IBM PS/2 mouse
Logitech mouse
Microsoft BallPoint mouse
Microsoft mouse
Mouse Systems serial or bus mouse
Olivetti/AT&T Keyboard Mouse

Networks

Any network 100% compatible with those listed below. Other networks may be compatible using a driver provided in the Windows Driver Library or from the vendor.

3Com 3+Open LAN Manager
3Com 3+Share
Artisoft LANtastic version 3.0 or higher
Banyan Vines version 4.0 or higher
DEC PATHWORKS
IBM OS/2 LAN Server version 1.2 or higher
IBM PC LAN Program
Microsoft LAN Manager
Microsoft Network (or 100% compatible)
Novell NetWare version 3.01 or higher
TCS 10Net version 4.1 or higher

Windows Compatible Printers

Any printer 100% compatible with those listed below. Other printers may be compatible using a driver provided in the Windows Driver Library or from the vendor.

Agfa 9000 Series PS
Agfa Compugraphic 400PS
Agfa Compugraphic Genics
Agfa-Compugraphic 9400P
Apple LaserWriter
Apple LaserWriter II NT
Apple LaserWriter II NTX
Apple LaserWriter Plus
Apricot Laser
AST TurboLaser/PS
AT&T 435
AT&T 470/475
AT&T 473/478
C-Itoh 8510
Cannon Bubble-Jet BJ-300
Cannon Bubble-Jet BJ-330
Canon Bubble-Jet BJ-10e
Canon Bubble-Jet BJ-130e
Canon LBP-4
Canon LBP-8 II
Canon LBP-8 III
Citizen 120D
Citizen 180D
Citizen 200GX
Citizen 200GX/15
Citizen GSX-130
Citizen GSX-140
Citizen GSX-140+
Citizen GSX-145
Citizen HSP-500
Citizen HSP-550
Citizen PN 48
Dataproducts LZR-2665
Diconix 150 Plus
Digital Colormate PS
Digital DEClaser 1150
Digital DEClaser 2150
Digital DEClaser 2250
Digital DEClaser 3250
Digital LN03R ScriptPrinter
Digital PrintServer 20/turbo
Digital PrintServer 40
Epson DFX-5000
Epson EPL-6000
Epson EPL-7000
Epson EPL-7500
Epson EX-1000
Epson EX-800
Epson FX-100
Epson FX-100+
Epson FX-1050
Epson FX-185
Epson FX-286
Epson FX-286e
Epson FX-80
Epson FX-80+
Epson FX-85
Epson FX-850
Epson FX-86e
Epson GQ-3500
Epson JX-80
Epson L-1000
Epson L-750
Epson LQ-1000
Epson LQ-1050
Epson LQ-1070 ESC/P 2
Epson LQ-1170 ESC/P 2
Epson LQ-1500
Epson LQ-2500
Epson LQ-2550
Epson LQ-500
Epson LQ-510
Epson LQ-570 ESC/P 2
Epson LQ-800
Epson LQ-850
Epson LQ-870 ESC/P 2
Epson LQ-950
Epson LX-80
Epson LX-800
Epson LX-810
Epson LX-86
Epson MX-100
Epson MX-80
Epson MX-80 F/T
Epson RX-100
Epson RX-100+
Epson RX-80
Epson RX-80 F/T
Epson RX-80 F/T+
Epson SQ-2000
Epson SQ-2500
Epson T-1000
Epson T-750
Fujitsu DL 2400
Fujitsu DL 2600
Fujitsu DL 3300
Fujitsu DL 3400
Fujitsu DL 5600
Fujitsu DX 2100
Fujitsu DX 2200
Fujitsu DX 2300
Fujitsu DX 2400
Hermes H 606
Hermes H 606 PS (13 fonts)
Hermes H 606 PS (35 fonts)
HP 7470A (HP Plotter)
HP 7475A (HP Plotter)
HP 7550A (HP Plotter)
HP 7580A (HP Plotter)
HP 7580B (HP Plotter)
HP 7585A (HP Plotter)
HP 7585B (HP Plotter)
HP 7586B (HP Plotter)
HP ColorPro (HP Plotter)
HP ColorPro with GEC (HP Plotter)
HP DeskJet
HP Deskjet 500
HP Deskjet 500C
HP DeskJet Plus
HP DraftMaster I (HP Plotter)
HP DraftMaster II (HP Plotter)
HP DraftPro (HP Plotter)
HP DraftPro DXL (HP Plotter)
HP DraftPro EXL (HP Plotter)
HP LaserJet
HP LaserJet 2000
HP LaserJet 500+
HP LaserJet IID
HP LaserJet IID PostScript
HP LaserJet III
HP LaserJet III PostScript
HP LaserJet IIID
HP LaserJet IIID PostScript
HP LaserJet IIIP
HP LaserJet IIISi
HP LaserJet IIISi PostScript
HP LaserJet IIP
HP LaserJet IIP Plus
HP LaserJet IIP PostScript
HP LaserJet Plus
HP LaserJet Series II
HP PaintJet
HP PaintJet XL
HP ThinkJet (2225 C-D)

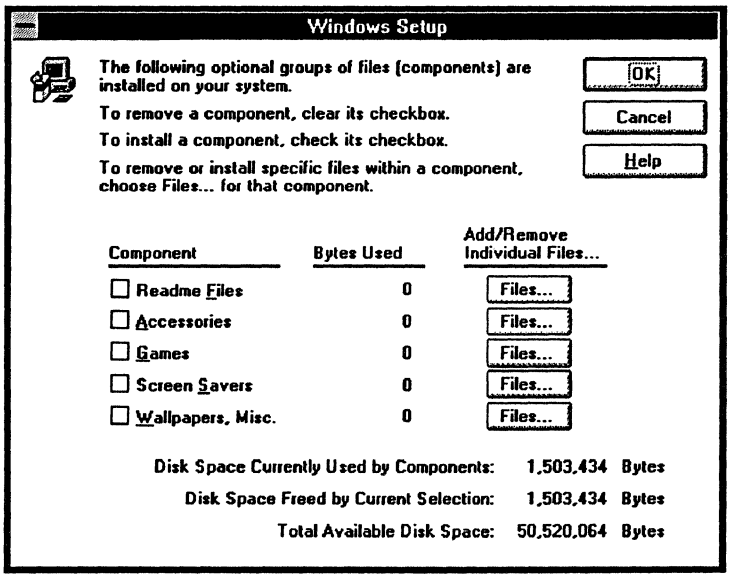
Windows Compatible Printers, continued

IBM Color Printer
IBM ExecJet
IBM Graphics
IBM Laser Printer 4019
IBM LaserPrinter 4019 PS17
IBM LaserPrinter 4019 PS39
IBM LaserPrinter 4029 PS17
IBM LaserPrinter 4029 PS39
IBM Personal Page Printer II-30
IBM Personal Page Printer II-31
IBM Personal Pageprinter
IBM Proprinter
IBM Proprinter II
IBM Proprinter III
IBM Proprinter X24
IBM Proprinter X24e
IBM Proprinter XL
IBM Proprinter XL II
IBM Proprinter XL III
IBM Proprinter XL24
IBM Proprinter XL24e
IBM PS/1
IBM QuickWriter 5204
IBM QuietWriter III
Kyocera F-5000 (USA)
Kyocera F-Series (USA)
Linotronic 200/230
Linotronic 330
Linotronic 530
Linotronic 630
Microtek TrueLaser
NEC Colormate PS/40
NEC Colormate PS/80
NEC Pinwriter CP6
NEC Pinwriter CP7
NEC Pinwriter P2200
NEC Pinwriter P5200
NEC Pinwriter P5300
NEC Pinwriter P5XL
NEC Pinwriter P6
NEC Pinwriter P7
NEC Pinwriter P9XL
NEC Silentwriter LC 860 Plus
NEC Silentwriter LC890
NEC Silentwriter LC890XL
NEC Silentwriter2 290
NEC Silentwriter2 90
NEC Silentwriter2 990
OceColor G5241 PS
OceColor G5242 PS
Oki OL840/PS
Okidata LaserLine 6
Okidata ML 192
Okidata ML 192 Plus
Okidata ML 192-IBM
Okidata ML 193
Okidata ML 193 Plus
Okidata ML 193-IBM
Okidata ML 320
Okidata ML 320-IBM
Okidata ML 321
Okidata ML 321-IBM
Okidata ML 380
Okidata ML 390
Okidata ML 390 Plus
Okidata ML 391
Okidata ML 391 Plus
Okidata ML 393
Okidata ML 393 Plus
Okidata ML 393C
Okidata ML 393C Plus
Okidata ML 92-IBM
Okidata ML 93-IBM
Okidata OL-400
Okidata OL-800
Olivetti DM 109
Olivetti DM 309
Olivetti ETV 5000
Olivetti PG 108
Olivetti PG 208 M2
Olivetti PG 303
Olivetti PG 306
Olivetti PG 306 PS (13 fonts)
Olivetti PG 306 PS (35 fonts)
Olivetti PG 308 HS
Olivetti PG 308 HS PostScript
Panasonic KX-P1123
Panasonic KX-P1124
Panasonic KX-P1180
Panasonic KX-P1624
Panasonic KX-P1695
Panasonic KX-P4420
Panasonic KX-P4455 V51.4
QMS ColorScript 100
QMS-PS 2200
QMS-PS 800
QMS-PS 800 Plus
QMS-PS 810
QMS-PS 820
QuadLaser I
Seiko ColorPoint PS Model 04
Seiko ColorPoint PS Model 14
Tandy LP-1000
Tegra Genesis
Tektronix Phaser II PX
Tektronix Phaser II PXi
Tektronix Phaser III PXi
TI 850/855
TI microLaser PS17
TI microLaser PS35
TI OmniLaser 2108
TI Omnilaser 2115
Toshiba P1351
Toshiba P351
Toshiba PageLaser12
Triumph Adler SDR 7706
Triumph Adler SDR 7706 PS13
Triumph Adler SDR 7706 PS35
Unisys AP9210
Unisys AP9415
Varityper VT-600
Wang LCS15
Wang LCS15 FontPlus
Wang LDP8

Appendix

C The Windows 3.1 Disks and Files

This appendix contains information about the Windows 3.1 installation disks and tips for minimizing the amount of disk space that Windows takes on your system, beginning with removal of components through Windows Setup.



Contents of this appendix

The Windows 3.1 Installation Disk Set 500
Minimizing the Windows "Footprint" 500
Removing Windows Accessories, Drivers, and Fonts 500
Removing Other Files 501
Creating the Minimum Windows Footprint 504

The Windows 3.1 Installation Disk Set

The Windows 3.1 installation disks are available in 360K, 720K, 1.2 MB, and 1.44 MB sizes. See README.WRI on the Windows Resource Kit disk for a complete list of files that includes:

- The disk location for each Windows file, based on the disk format size.
- The expanded file size for each file.
- A description of the purpose of each file.

Minimizing the Windows "Footprint"

This section describes how to minimize the Windows "footprint," descreasing the amount of space that Windows takes on your hard disk. If you follow these steps, you can minimize the space that Windows takes on your hard disk to as low as 2.6 MB. For details about the files described here, see Chapter 3, "The Windows 3.1 Files."

Removing Windows Accessories, Drivers, and Fonts

If you want to remove files from Windows, begin with the Windows utilities, then remove unused drivers and fonts. The following steps will reduce the Windows footprint to about 5.6 MB.

To remove Windows components:



1. Run Windows Setup from Program Manager.
2. Choose Add/Remove Windows Components from the Options Menu.
3. Use this utility to remove components of Windows you don't use.

To remove unnecessary drivers:



1. Run Control Panel from Program Manager.
2. Double-click the Drivers icon in Control Panel, select any drivers that you do not use, then choose Remove.
3. Double-click the Printers icon in Control Panel, select any printers you don't use, then choose Remove.

These steps remove the references to the device driver or the printer driver, but the driver file may still exist on your disk. To delete the files, run File Manager and delete the associated .DRV file.

Also, some printers have related supporting files and help files. For example, the HP Laserjet IIISi driver uses the font installer files FINSTALL.DLL and FINSTALL.HLP. You can delete these files if you remove that printer. For details about which .DRV files and support files are associated with drivers and printers, see Chapter 3, "The Windows 3.1 Files."

To remove unused fonts:



1. Run Control Panel from Program Manager and double-click the Fonts icon.
2. Select the fonts you don't use.
3. Make sure the Delete Font File from Disk option is checked, and choose Remove.
4. When the dialog box prompt appears, choose Yes to remove the file.

Removing Other Files

After you complete the steps in the previous section, there are still some files you can manually delete from Windows. These files are grouped below by function. Be careful when deleting these files. Pay attention to the side effects described in the notes.

Removing Other Accessories

After you use the Windows Setup program to remove the accessories, other accessory files remain that can be deleted. Deleting the following files will decrease the Windows footprint by 565K.

Files for other accessories that can be removed

CLIPBRD.EXE	The Clipboard Viewer. Without this file, you cannot view the information stored on the Clipboard.
MSD.EXE MSD.INI	The files for the Microsoft Diagnostics tool.
PIFEDIT.EXE	The PIF Editor. If you delete this file, you cannot edit or create program information files to support non-Windows applications.
PRINTMAN.EXE	The Windows Print Spooler. If you delete this file, Windows cannot spool your print jobs.
SYSEDIT.EXE	The System Files Editor.
TASKMAN.EXE	The utility that presents a list of running tasks when you press CTRL+ESC.
WRITE.EXE WRITE.HLP	The files for Windows Write.

Removing Support for Standard Mode

If you run Windows in 386 enhanced mode, and never run Windows in standard mode, there are additional files you can delete. Deleting the following files will decrease the Windows footprint by 183K.

Files for standard mode support that can be removed

DOSX.EXE	The MS-DOS Extender.
DSWAP.EXE	Standard mode support for swapping non-Windows applications.
KRNL286.EXE	The standard mode kernel for 80286 PCs and for 80386 and higher PCs with less than 2 MB of RAM.
WINOLDAP.MOD	The standard mode support for executing non-Windows applications.
WSWAP.EXE	Standard mode support for swapping Windows applications.
*.2GR	The standard mode grabber file. The actual filename depends on the display driver installed. For example, for VGA.DRV, this file is VGACOLOR.2GR.

Note Deleting any of the above files disables standard mode Windows.

Removing Support for 386 Enhanced Mode

If you run Windows only in standard mode, and never in 386 enhanced mode, you can delete additional files. Deleting the following files will decrease the Windows footprint by 781K.

Files for 386 enhanced mode support that can be removed

CGA40WOA.FON CGA80WOA.FON EGA40WOA.FON EGA80WOA.FON	The fonts for non-Windows applications.
CPWIN386.CPL	The support file for the 386 Enhanced Mode icon in Control Panel.
DOSAPP.FON	The fonts that support sizable text in a window in 386 enhanced mode.
*.3GR	The 386 enhanced mode grabber file. The actual filename depends on the display driver installed. For example, for VGA.DRV, the grabber file is VGA.3GR.
*.386	Various 386 enhanced mode support drivers.
WIN386.EXE	The Windows Virtual Machine Manager.
WIN386.PS2	Support for PS2 architecture.
WINOA386.MOD	The 386 enhanced mode support for executing non-Windows applications.

Note The above files will only be present on 80386 and higher PCs. Deleting any of the above files disables 386 enhanced mode Windows.

Removing Support for Non-Windows Applications

If you are running only Windows-based applications, and will never run non-Windows applications, you can delete the files that support these applications. Deleting the following files will decrease the Windows footprint by 250K.

Files for non-Windows application support that can be removed

*.PIF	Any PIFs can be deleted.
APPS.INF	The information file for building PIFs.
CGA40WOA.FON CGA80WOA.FON EGA40WOA.FON EGA80WOA.FON	Fonts for non-Windows applications.
*.2GR, *.3GR	The standard mode and 386 enhanced mode grabber files.The actual filenames depend on the display driver installed. For example, for VGA.DRV, the grabber files are VGACOLOR.2GR and VGA.3GR.
DOSAPP.FON	The fonts that support sizable text in a window in 386 enhanced mode.
DSWAP.EXE	Standard mode support for swapping non-Windows applications.
WINOLDAP.MOD	The standard mode support for executing non-Windows applications.
WINOA386.MOD	The 386 enhanced mode support for executing non-Windows applications.

Note Deleting the above files will remove all support for non-Windows–based applications, including the MS-DOS Prompt.

Removing Support for Help Files

You can delete the online help files. Deleting the following files decreases the Windows footprint by 421K.

Files for the Windows Help system that can be removed

CONTROL.HLP	The help file for Control Panel.
SETUP.HLP	The help file for Windows Setup.
WINHELP.EXE	The Windows Help engine.

Note If you delete WINHELP.EXE, you cannot access help from any Windows applications.

Removing Miscellaneous Files

Several other files that you can remove are shown in the following table. Deleting these files will decrease the Windows footprint by 340K.

Miscellaneous files that can be removed

AUTOEXEC.WIN CONFIG.WIN	Files created by Setup that contain suggested changes for CONFIG.SYS and AUTOEXEC.BAT.
BOOTLOG.TXT	The file that holds a log of files loaded during the Windows boot procedure.
EMM386.EXE	If you are not loading EMM386.EXE in CONFIG.SYS, and if you do not need EMS memory for non-Windows applications, you can delete this file.
MORICONS.DLL	A dynamic-link library that contains icons for many non-Windows applications.
MOUSE.INI MOUSE.SYS MOUSE.COM LMOUSE.COM	Mouse drivers. You can delete any of these files that you are not loading in CONFIG.SYS or AUTOEXEC.BAT.
RAMDRIVE.SYS	Support for a RAM drive in extended memory. You can delete this file if it is not loaded in your CONFIG.SYS file.
SETUP.TXT	The README file for Windows Setup.

Creating the Minimum Windows Footprint

By removing all of the following kinds of files, you can create the minimum Windows footprint. With this minimum footprint, Microsoft Windows will run only in standard mode, requiring 2.6 MB of disk space:

- Remove Windows components.
- Remove unnecessary drivers.
- Remove unused fonts.
- Remove other accessories.
- Remove 386 enhanced mode support.
- Remove non-Windows application support.
- Remove the Windows Help system.
- Remove miscellaneous files.

Appendix

D Articles

This appendix contains technical articles on these topics:

- Microsoft Windows for Pens
- Quarterdeck's QEMM–386 and Windows 3.1
- PC-NFS and Windows 3.1
- FastDisk: An Introduction to 32–Bit Access

Contents of this appendix

Windows for Pens 506
Why Pens? 506
Technical Highlights 508
The Internal Architecture of Pen for Windows 509
RC Manager 510
Pen for Windows Support Resources 511
Quarterdeck's QEMM–386 and Windows 3.1 515
QEMM–386 Features for Windows 515
Troubleshooting for QEMM-386 516
Getting Additional Help 518
PC-NFS and Windows 3.1 519
Installation Tips 519
Using PC-NFS With Windows 519
FastDisk: An Introduction to 32-Bit Disk Access 520
What Is 32-Bit Disk Access? 520
The User Interface for 32-Bit Disk Access 522
The Components of the 32-Bit Disk Access System 523
The Risks of 32-Bit Disk Access 523

Windows for Pens



Why Pens?

With pen computing, you can use a pen-like stylus on a computer with a special display to print, draw, and annotate information much as you would with pen and paper. Beneath the computer's display is a digitizer, an electronic sensing device that captures data and displays it as "ink" or interprets it as text or as a command. Pen computing supports:

- **Creation of highly portable devices.** With the pen, hardware developers are no longer constrained by the keyboard, and can deliver smaller, lighter machines that can be used while standing or walking.
- **Development of new markets.** With smaller, more portable pen PCs, new categories of applications such as notetakers and calendars become possible.
- **More intuitive features for users,** combining the familiar feel of pen and paper with the power of the computer. Pen computers may resemble clipboards, notebooks, and pocket calendars.
- **Improved efficiency.** The pen is a superb pointing device for accurate data selection and object manipulation. Gestures (described later in this paper) make data manipulation economical because they can simultaneously communicate position (or object selection) and initiate an action.
- **Flexibility.** The pen is a great drawing tool for creating detailed graphics and, with the Windows graphical user interface and Object Linking and Embedding features, can support creating compound documents. The pen is ideal for taking notes, sketching, and annotating documents.

Who Will Use Pens?

Consider the following scenarios:

- Nurses and doctors can track and record patient histories as they make their rounds, using clipboard-like computers and pens to look up and enter patient information over a wireless network from servers throughout the hospital.
- Insurance claims adjusters can assess automobile damages on site, looking up relevant cost information with the tap of the pen, then printing the estimate and writing a check to the repair shop at the end of the visit.
- Sales representatives can track inventory and the effect of promotional campaigns in retail stores, using a pen computer. At the end of the day, the information is transmitted through a phone line back to headquarters.
- Financial analysts can perform "what if" analyses in meetings with customers or managers, using a pen to change a number or drag a line on a chart.
- Business people can maintain their calendars on computers that feel more like notepads and calendars than current computers. They jot down notes during meetings, conduct interactive presentations, annotate documents, or edit using simple handwriting.

Pen users will include:

- Service providers such as field sales and service representatives, field engineers, route managers, and insurance adjusters who need to look up information on site, or capture it in a format that is machine-readable to be used in the field or to be communicated to other users through cellular communications or other network solutions.
- Users of laptop and notebook PCs, who will find that the next generation of notebook PCs will most likely be "convertible," allowing users to detach the display for pen-exclusive uses. These pen computers will be used as data entry devices, and may support smaller, less expensive compact disk storage and applications for data retrieval.
- Professionals who will add pen devices to existing hardware, controlling the pace and cost of the migration to pen computing.

What Is Windows for Pens?

The modular design of the Windows operating system allows the integration of pen extensions into the basic Windows 3.1 product. Windows for Pens is an OEM product to be packaged and sold with a variety of new pen computers.

Windows for Pens offers a base of pen-specific APIs to allow Windows developers to build pen-centric applications. Windows for Pens supports existing Windows applications without special modification through a special compatibility layer so users can:

- Point and tap as with the mouse.
- Issue commands using a standard set of 13 system-wide gestures.
- Issue commands using 26 user-assigned gestures.
- Print characters in any text field. Hand-printed entries in Windows applications are translated and appear as typed text.

Windows for Pens provides an open, modular approach to handwriting recognition, so you can replace or supplement the Windows for Pens recognizer with third-party recognizers.

Windows for Pens uses the Windows-specific functions such as scalable fonts, Advanced Power Management, support for Flash EPROM, IC card storage, and the ability to run in ROM. Windows for Pens also includes pen-specific DLLs and device drivers so that applications can run on new pen hardware platforms. Applications can run in portrait or landscape mode and can support scaling for different sized displays.

Why Choose Windows for Pens?

Windows for Pens, a complete operating environment for pen computers, offers compelling benefits for all groups involved in developing and using pen applications:

- **For new computer users,** Windows for Pens provides the building blocks for creating easy-to-use pen-centric applications. Unlike other environments, Windows for Pens does not impose a specific metaphor on end-users, leaving the choice of user interface design and functionality to the application developer.
- **For existing computer users,** Windows for Pens draws on the Microsoft Windows environment, so they can use the same applications in the office or on the road. Users can use the pen with Windows applications to point and tap as they would with a mouse, issue commands using gestures or hand-printed characters to be translated as text.

  Windows for Pens, because it is based on the Windows environment, supports a multitude of communication options, printing devices, database engines, and file formats.
- **For Independent Software Vendors** (ISVs), Windows for Pens provides: comprehensive API support for controlling ink, recognition, and gestures; extensive driver support for various pen platforms and OEM hardware; support for multiple recognition engines to facilitate access to domestic and international markets, and all the capabilities provided by the Windows 3.1 environment.

  Windows for Pens offers developers maximum flexibility to design and implement applications using the Windows interface or innovative metaphors. With Windows for Pens, software developers can use their programming experience and the code base created for Microsoft Windows, and can choose from over 120 different development tools— third generation languages, object oriented development environments, quick UI and application generators, and database front ends.

  Because of the large and growing installed base of Windows users, ISVs have access to established markets and to new market segments that require connectivity solutions.
- **For corporate MIS,** Windows for Pens provides an environment where developers can draw on a market of skilled developers and consultants, or use their in-house Windows experience with development tools and application generators that shorten the development cycle. New pen-centric applications can be integrated with existing desktop solutions to create enterprise-wide solutions, avoiding the inefficiencies associated with supporting specialized niche environments, and eliminating the need to maintain different operating systems, code bases, and skill sets.
- **For Original Equipment Manufacturers** (OEMs), Windows for Pens offers a systems solution that can be scaled across platforms and form factors. Its modularity ensures that it can run on small silicon-memory–based pen computers as well as larger clipboard-type machines and convertible notebooks. Its modular design means that developers can scale down Windows for Pens to under 2 MB — most of which can be run in ROM. Because Windows for Pens can run on a variety of hardware platforms, OEMs can capitalize on their investment in hardware design and implementation.

Technical Highlights

Windows 3.1 includes several special pen features and capabilities. These features, described in detail later in this paper, include:

- **Gestures,** implemented as 13 standard system-wide gestures available across all applications plus 26 application-mappable gestures.
- **The Pen Palette** is a floating toolbar that features buttons for invoking the Writing window, the Trainer, the Gesture Manager, and the On-Screen Synthetic Keyboard.
- **Control Panel settings for pen and ink,** so that users can specify their own custom database of handwriting samples.

The core components are described in "The Internal Architecture of Pen for Windows" later in this paper. These components include:

- **The RC Manager.** This includes the pen, display drivers, and the recognition engine. These components handle the data that flows between the pen and the application.
- **The Windows for Pens Compatibility Layer.** Also referred to as the Pen Message Interpreter, this function interacts with standard Windows applications, so they can use gestures and handwriting recognition.

Gestures

A gesture is a special handwriting symbol that issues a command, such as Copy, or produces a non-printing character, such as a carriage return or backspace.

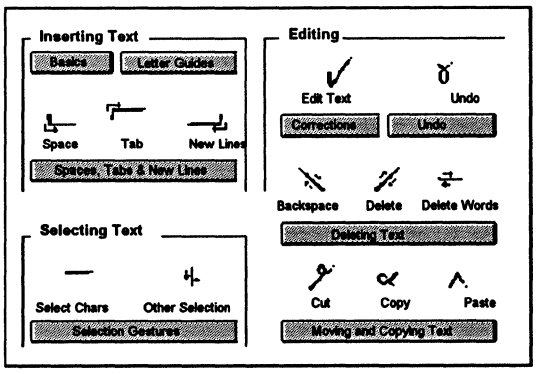


Figure D.1 Gestures

Gestures can save time, especially for editing or correcting text. With gestures, the user can simultaneously specify an object or function and initiate an action. Windows for Pens gestures are similar in form and purpose to common handwritten editing symbols. Three types of gestures are supported by Windows for Pens:

- Thirteen system-wide gestures that are uniform across applications.
- Application-mappable gestures that allow application-specific commands to be mapped to an unused set of symbols.
- User-defined gestures, so that users can define their own symbols.

The Pen Palette

The Pen Palette is displayed whenever Windows for Pens is started, providing a pen interface (menus and dialog boxes to execute the most common pen commands), an expandable writing area, and an on-screen keyboard. The Pen Palette is not required for applications that call the pen API directly. To support the feel of pen and paper, Windows for Pens was designed so that users can write anywhere in the text area of an application rather than in a special writing window.

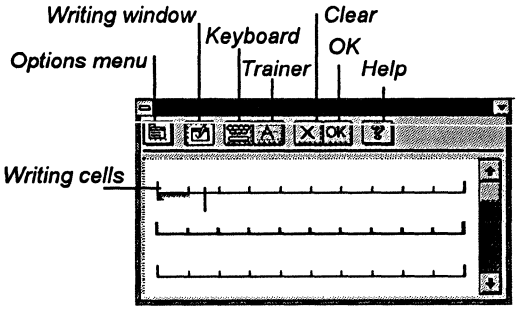


Figure D.2 The Pen Palette

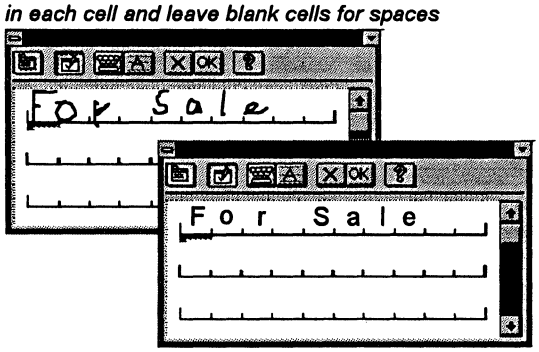
The other elements that support the Pen Palette are:

- **The Writing window and Edit box,** invoked with an edit gesture that resembles a check mark.

  The Writing window and Edit box provide a large area for entering, editing, and reviewing hand-printed text. The Writing window appears below the Pen Palette.

- **The Trainer** supports sophisticated on-the-fly handwriting training. Users are not required to train the system before using it and can call the Trainer as needed, without closing work in progress.
- **The Gesture Manager** controls the definition of 26 circled character gestures based on the 26 characters in the alphabet.

To enter text in the Writing window, print one letter in each cell and leave blank cells for spaces



The printed text is recognized after you lift the pen and pause briefly

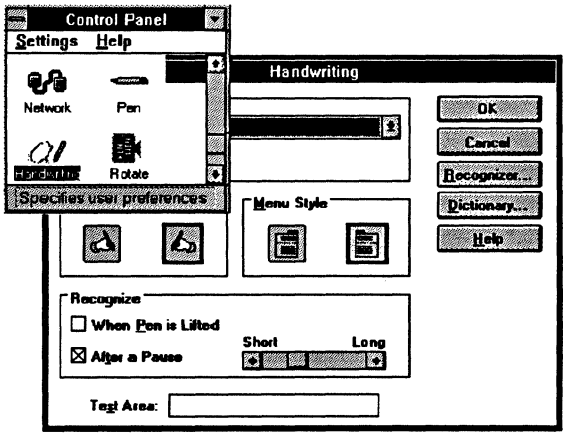
Figure D.3 The Wiring window

- **The On-Screen Synthetic Keyboard** provides a tool for entering characters, CTRL and ALT key combinations, and keyboard shortcuts that can be used in conjunction with the Gesture Manager to replace keyboard-based menu accelerators with circled character-based gestures.

The Control Panel

By defining values in the Control Panel, users can control the width of the ink generated by the pen, the orientation of the menus, and the type of user (left- or right-handed). The Control Panel supports multiple users and different display orientations and also defines when recognition occurs—for example, when the pen is lifted or after a time-out.

Figure D.4 Control Panel items for pens



The Internal Architecture of Pen for Windows

This discussion describes how Windows for Pens is for applications programming, and is intended for readers who understand some basics of Windows programming.

Adding the pen to Windows involves enhancing both input and output. For input, the pen device driver provides pen data to the system. For output, pen data is echoed on the display as "ink" while drawing takes place. This ink layer is handled through a new device driver entry point and involves no changes to the GDI interface. The presence of ink has no impact on Windows applications. At this lowest level in the system, connection between pen input and ink on the display is immediate. This is identical to the mouse cursor location being updated at interrupt time.

Above the device-driver level, pen input is interpreted and given meaning. The device driver-to-device driver communication in the hardware guarantees that users see the pen as a writing tool that can freely leave ink on the display. The levels above interpret the ink as text, shapes, and gestures.

Windows applications interact with the pen through the RC Manager, which handles the pen device driver, recognizers, and dictionaries on behalf of all programs. The RC Manager sends pen input to the appropriate window and coordinates pen input processing for that window.

The diagram in Figure D.5 shows the various components of the pen enhancements to Windows. The RC Manager is in the Systems components layer, which also includes MS-DOS functionality, input handling (labeled "Windows" to designate the functionality in the Windows Kernel and USER.EXE libraries), and graphics output (handled by GDI.EXE). The "RC" in RC Manager stands for Recognition Context.

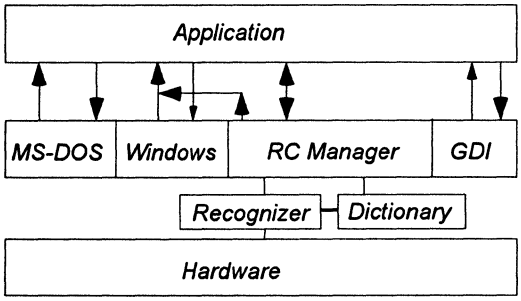


Figure D.5 The pen enhancements to Windows

RC Manager

The RC Manager is the heart of the extensions to Windows provided in Windows for Pens, providing the architectural component that orchestrates pen input handling. Code for the RC Manager is contained in PENWIN.DLL, so the terms "RC Manager" and "Pen DLL" are synonymous.

Whenever a windows program wants to work with the pen extensions, that program obtains an RC, fills in the fields of the RC appropriately, and hands the RC to the RC Manager. Within the RC data structure are fields that specify exactly how the pen data is to be handled. In this sense, the RC is patterned after the Device Context used with GDI.

In general, one RC is used with each inking "transaction" between the user and the program. Often, programs will find it convenient to allocate an RC permanently and fine-tune it as needed (just like an "own" DC).

Fields within the RC identify which recognizer should be applied to the ink and which dictionaries used to select the "best guess" string. Other fields specify what conditions will end the inking transaction (such as the pen going up, or the pen going down in a specific location, or a time-out for pauses). The RC also provides parameters used by the recognizer and the dictionaries themselves.

At the end of each inking transaction, the RC Manager returns the results of ink processing to the window that initiated the transaction. If the window does not fill in an RC and initiate an inking transaction with the RC Manager, the pen will function exactly like a mouse.

The Pen and Pen Driver

Pens come both tethered and untethered, to serve different markets, but both styles of pen act the same under Windows for Pens. The pen works together with electronics in the PC to encode (*x*, *y*) locations and other information. The pen must sense the difference between touching and not touching the drawing surface, typically accomplished with a tip-switch that acts just like the left mouse button. Many pens actually report pressure levels at the surface, but this additional information is not required. Some pen systems also provide location even when the pen is not quite touching the surface (called proximity sensing).

Many pens will also provide a button on the barrel. An OEM might also implement a pen that records the slant of the pen or rotation of the pen barrel. This information is not required by Windows for Pens, but a standard method is provided for applications to determine whether the information is present and, if so, to access the information.

All pen devices must be capable of recording at least 120 events each second, and must have a resolution of at least 200 spots per inch. Many pens will be able to record data at much higher resolution or rates. These requirements guarantee sufficient data for character recognition by the vector-based character recognizers used with Windows for Pens.

All pen data is reported by the Pen Device Driver to the system at 0.001" resolution. This standard resolution allows improvements in pen hardware. It also is the same as the HIENGLISH Mapping mode in the GDI, and is therefore convenient for programs.

The Pen SDK includes sample device drivers for WACOM and Scriptel tablets. These drivers can be easily modified to support any pen whose hardware uses an interrupt to return data through I/O addresses.

Display Driver

The display driver under Windows for Pens is an enhancement of the standard display driver for Windows 3.1. The extensions support echoing ink on the display as it is drawn by the user, and moving the origin of the screen to support landscape and portrait orientations. Ink support should be provided in such a way that removing the ink does not generate redraw messages to Windows programs. The display driver must also provide a new cursor—the Pen cursor.

Inking is supported by two function calls in the display driver, **InkReady()** and **GetLPDevice()**, and is perhaps the easiest of the functions to implement. Supporting portrait and landscape modes is somewhat more difficult. The Pen SDK includes a sample driver for VGA devices.

Recognizer

A recognizer is a dynamic-link library (DLL) that collects ink from the device driver and interprets that ink as characters in one or more alphabets. When inking begins, the RC Manager calls the recognizer specified in the RC for the transaction. Windows programs do not directly interact with recognizers, though programs may use application-specific recognizers.

Recognizers are required to handle an inking transaction according to the fields set in the RC for that transaction. The recognizer will end the transaction when the conditions specified in RC fields are met. The

recognizer will examine the ink for characters in the alphabets and character sets named in the RC. The recognizer will only return characters from the character sets named, and, within those sets, only characters recognized at the level of confidence specified.

During recognition, the recognizer collects and analyzes the ink. The original ink is returned to the calling program along with the results of analysis, which is returned in a symbol graph that is a sequence of 32-bit values. Each 32-bit symbol is either a recognized character or a meta-character used to group alternate possibilities where more than one character meets the confidence level set for the transaction in the RC. For example, the string of symbols "clea{r|n}" would be a symbol graph representing two possible strings ("clear" and "clean"). In this example, "c," "l," "e," "a," "r," and "n" represent recognized characters, each of which is associated with a confidence level. The "{," "|," and "}" symbols are meta-characters used to define the graph of alternatives.

All recognized characters in a symbol graph are associated with both the confidence the recognizer assigns to that character and the specific strokes in the ink that make up the character. If the recognizer cannot assign a character to some part of the ink with the required confidence, it specifies "don't know" at that position in the symbol graph.

A recognizer returns the results of its analysis to the RC Manager, which arranges to return the results to the calling program. The recognizer returns ink, a symbol graph, symbol elements (containing the symbol value and confidence), and symbol correlations (which list the strokes for a recognized character). Recognizers may return partial results during an inking transaction.

Beyond the requirements for collecting ink and returning results, there are no constraints placed on how recognizers work. Windows for Pens was built from the ground up to accommodate recognizers for languages using non-Roman alphabets. The recognizer interface is open and published, and third-party recognizers will be widely available. At least some Windows programs will want to provide custom recognizers for special symbols.

Compatibility with Windows Applications

The Pen Message Interpreter allows the pen to be used with existing Windows (and non-Windows) applications by acting as a "middleman" between these applications and Windows for Pens. Most pen input is treated as mouse input (for example, a downward motion that depresses the pen point is treated as a mouse click) and is passed to the application's mouse buffer for processing. Existing Windows applications already buffer mouse input separately from keyboard input, simplifying the pen message interpreter's tasks.

However, when the user moves the pen into a text entry box, Windows for Pens recognizes that the user is now over a writing area. This causes several changes: the cursor shape changes to that of a pen; the display device driver temporarily throws a transparent overlay over the existing screen to show the "ink" left by the pen; and pen input is sent to the handwriting recognition engine for processing, then sent to the application's keyboard buffer. Gestures are handled similarly and mapped to command events, such as "copy."

Pen for Windows Support Resources

CompuServe

CompuServe features a special forum for Windows for Pens beta developers. The forum is used daily by engineers and support specialists who answer technical questions and place technical articles that are of wide interest to Windows for Pens developers. To access this forum, developers need to be on the Windows for Pens beta program. Section 12 under "winbtdev" is dedicated to Windows for Pens issues.

Contact: 800/ 848–8199

We recommend using TAPCIS, The Access Program for the CompuServe Information Service. Put together by Support Group, Inc.

Contact:
Phone: 800/ 872–4768 or 301/ 387–4500
Fax: 301/ 387–7322

Pen for Windows Consultants

Andersen Consulting

Andersen Consulting is a world wide information systems integrator, with offices in 146 cities. In the U.S., there is an Andersen Consulting office in all major metropolitan areas. Since installing the first business computer in 1954, Andersen Consulting has helped leaders in virtually every industry to apply information technology for competitive advantage and to manage the profound change associated with it. We offer clients our expertise in industry, business, and advanced technology, plus the services necessary to integrate complex business and technology systems. This unmatched blend of capabilities has made Andersen Consulting one of the world's largest commercial systems integrators. A single source for the total solution, from strategic planning and systems design and development to change management, facilities management, hardware, and software.

As one of the largest systems integration firms in the world, our clients list is extensive. We provide professional services to a significant proportion of the Fortune 500 and to the leading companies in most local markets. In industries like Insurance, Aerospace, and Telecommunications, we are presently working with nearly all the major entrants.

Product Specific Experience: Andersen Consulting is a leader in the development of custom Windows-based applications for major corporations. As part of our ongoing application platform research, we have followed the development of pen-based technology with interest since its inception. We have been a participant in the Windows for Pens beta program since its inception, evaluating it as a basis for client solutions and demonstrating its capabilities to existing clients.

Pen Consulting Services: For the organization considering the application of pen technology, Andersen Consulting can provide a number of services, including:

- Information technology strategy
- Information planning
- Analysis of competitive and management issues
- Systems development planning
- Systems design and installation
- Organizational change management
- Systems support and management

Applying any new technology simply for its own merits is a waste of scarce corporate resources. Andersen Consulting assists its clients to identify and to exploit the opportunities a new technology can provide. We work with our clients to develop solutions that can provide significant, sustained competitive advantage.

Contact for Andersen Consulting Information:
Timothy M. Boudreau
69 West Washington Street
Chicago, IL 60602
Phone: 312/ 507–6526
Fax: 312/ 507–2548

K.C. Watkins
801 Second Avenue
Seattle, WA 98104
Phone: 206/ 233–8241
Fax: 206/ 386–8615

Joseph McCall & Company

Joseph McCall & Company provides professional consulting services in North America in the following functional areas:

- Strategic services: information planning and technology audits
- System design and implementation
- Software evaluation and selection
- Project management and planning
- System testing and production systems support
- Training design and execution

Major accounts include: Department of Business Regulation, State of Florida; Allis–Gleaner (AG Co.); Continental Telecom (Contel); Alabama Gas Company; Blue Cross-Blue Shield of Florida.

Pen-Based Experience: In early 1990, Joseph McCall & Company made a strategic decision to pursue pen-based technology. Since that time we have established ourselves as strategic marketing partners with Microsoft and NCR Corporation and have successfully implemented one of the largest applications of pen-based technology. Our goal is to provide innovative pen-based solutions that meet the information processing needs of our clients. With the experience and knowledge gained from our involvement in pen-based projects, we have successfully applied our expertise across various industries including: Insurance and Healthcare; Consumer Products; Federal, State and Local Government; and Manufacturing.

Pen Consulting Services: Joseph McCall and Company is an information systems consulting firm that has been providing professional services for over six years. Our focus is on helping our clients achieve the promise of today's advanced technologies. Our goal is to provide innovative pen-based solutions that address the specific challenges facing our clients. We have a dedicated team of professionals, with expertise in all phases of the systems development life cycle. Our challenge as systems integrators is to help companies to build on their investment in existing technologies, while also allowing them to effectively assimilate pen-based technologies into their current systems architecture. Through the strategic application of pen-based technology we can assist organizations in realizing the significant benefits offered by today's advanced technologies.

Contact for Joseph McCall & Co. Information:
Scott J. Brady, Manager, Pen-Based Development
1819 Peachtree Street
Suite 520
Atlanta, GA 30309
Phone: 404/ 352–4241
Fax: 404/ 351–3728

Knowledge-Based Solutions (KBS)

The areas of expertise for Knowledge-Based Solutions include our QUICK START rapid prototyping program for early adopters of pen-based systems; domain experience includes field maintenance, sales force automation, financial, health care, and insurance, with projects throughout U.S., though primarily New York and New England. Representative clients include:

- Startup software company (health care)
- Windows for Pens hardware OEMs
- Windows for Pens ISVs
- Government agency

Product Experience: Experience with all major Windows–based development tools, including but not limited to ObjectVision, Visual Basic, Windows for Pens SDK, as well as connectivity with databases, wireless networks, and other Windows applications.

Pen Consulting Services: We are experts in the deployment of field automation systems, specifically involving pen-based systems. Our early work on pen computing with Microsoft and other pioneers, combined with extensive contacts, provides KBS clients with access to the most complete base of experience with this technology.

KBS works closely with vendors as an approved NCR Software Alliance Partner and a Microsoft Consultant Relations Program member.

Our low-cost, low-risk QUICK START program provides clients with a chance to quickly produce a "proof of concept" system to get started with pen-based computing. Other KBS services include full system development, executive overviews, technology transfer, and strategic analysis. Please contact us for a copy of our QUICK START brochure.

Contact for KBS Information:
Randy Parker
65 Westbourne Terrace, Suite 300
Brookline, MA 02146
Phone: 617/ 566–2451
Fax: 617/ 734–8442

Microsoft Consulting Services

Microsoft Consulting Services are available worldwide. MCS Consultants have assisted organizations throughout the world. MCS has already served hundreds of clients with tasks ranging from developing feasibility studies to providing onsite, long-term consultants. We've worked with dozens of industries, including aerospace, public utilities, law, finance, government, hotels and resorts, telecommunications, health care, automobile, manufacturing, and others.

Microsoft Consulting Services is staffed by consultants with a wide range of expertise in the field of information technology management and world-class technical consultants in areas such as the Microsoft Windows graphical environment, OS/2 operating systems, Microsoft Visual Basic modern programming system for Windows, Microsoft Mail for PC Networks, LANs, multimedia, enterprise networking, cooperative processing, development tools, and office automation. And because MCS is an integral part of Microsoft, clients have excellent resources. MCS is directly linked to Microsoft product development and support groups and is able to obtain their tools, services, and expertise.

Major clients include: Baxter Healthcare, British Petroleum, Coca-Cola Company, Dun and Bradstreet, Eli Lilly and Company, JP Morgan Company, Nike International, and Texaco are a partial list of organizations served by MCS worldwide.

Pen Consulting Services: MCS offers four primary services, tailored to meet the needs of individual clients and devised to get the job done for the mobile professional:

- Planning
- Design
- Development
- Implementation and support

We are committed to helping you leverage your investment in information technology to your best advantage. We can provide assistance with all phases of a project in developing systems that take advantage of the portability and the familiarity of pen computing. Our mission is to help you apply the leading information technologies such as pen computing to the information needs of your organization.

Contact for MCS Information:

Western Regional Office:
Microsoft Consulting Services
Microsoft Corporation
One Market Plaza, Suite 1550
Spear Street Tower
San Francisco, CA 94105
Phone: 415/ 905–0235

Eastern Regional Office:
Microsoft Consulting Services
Microsoft Corporation
200 State Street
Boston, MA 02109
Phone: 617/ 951–2780

Sales Technologies

Sales Technologies (ST) develops, installs, and supports tailored, networked systems that improve sales force productivity for corporations with mobile field sales forces. ST's software product consists of five core application modules: Territory Management, Electronic Mail, Electronic Forms, Electronic Reports Distribution, and Spreadsheets. Pen-based systems are already being used with the Territory Management module, and further development of pen-based software is underway. ST also offers a broad range of services to keep customers' systems running smoothly, including design consulting, database development, 24-hour help line, data center, hardware repair and replacement services, and comprehensive user training.

ST has developed sales force productivity solutions for companies in many different industries, including consumer package goods, industrial, oil and gas, and health care.

Major clients include: 3M, General Mills, Gillette, Mobile Oil, SmithKline Beecham, The Upjohn Company.

Pen Consulting Services: Sales Technologies offers custom solutions based on its core products. Our primary services include:

- Planning
- Design
- Development
- Implementation and support.

Contact for Sales Technologies Information:
Gitesh Ramamurthy
3399 Peachtree Rd. NE
The Lenox Building, Suite 700
Atlanta, GA 30326
Phone: 404/ 841–4000
Fax: 404/ 841–4115

Quarterdeck's QEMM–386 and Windows 3.1

Quarterdeck's QEMM–386 versions 6 and above support Microsoft Windows 3.1 in both standard and 386 enhanced mode. Windows and MS-DOS 5.0 provide memory management features, but QEMM–386 remains popular as a 386/486 memory manager because it combines performance and peak optimization in an easy-to-use package. In complex Windows installations, where insufficient memory for non-Windows applications or memory conflicts are a problem, QEMM–386 can often provide solutions.

QEMM–386 Features for Windows

- QEMM–386 combines the functions of HIMEM.SYS and EMM386.EXE with a smaller "footprint" in memory below 640K (3K to 4K).
- New Stealth technology provides up to 211K of upper memory blocks in which to load resident programs by hiding machine ROMs and allowing UMBs to be mapped over them.
- Squeeze technology allows larger resident programs to be loaded by squeezing them temporarily into the EMS page frame while loading. With the increased amount of UMBs that Stealth provides, Squeeze generally eliminates the need to disable EMS support to load resident programs into UMBs.
- Manifest, Quarterdeck's memory analysis program, is currently included free. Manifest provides extensive information about the computer where it is running and is helpful for optimizing a system or diagnosing a memory problem or conflict.
- The Optimize program automatically loads TSRs and device drivers into the optimum regions of upper memory.
- The Analysis program identifies memory conflicts that might exist on your machine.
- QEMM–386 provides the option to map ROMs into fast RAM—a feature that often results in substantially better performance, especially where screen update speed is important.
- QEMM–386 can manage shadow RAM or Top Memory. Many 386 systems turn 384K of the first megabyte of memory into shadow RAM or Top Memory, which is invisible to MS-DOS. QEMM automatically makes this memory available for Windows or non-Windows programs.
- QEMM–386 supports Windows in both standard and 386 enhanced mode, whether or not QEMM–386 is active. Currently, all other 386 memory managers only support Windows 386 enhanced mode. No special setup is required; just start Windows in the mode you prefer. Real mode is also supported for Windows 3.0.
- QEMM–386 automatically frees unused UMBs to support Windows translation buffers. This further lowers the overhead of Windows in MS-DOS and allows you to run larger non-Windows programs.
- The Loadhi programs allow TSRs and drivers to be loaded into specific upper memory locations, for efficient use of upper memory. With the Optimize program, optimum configuration is achieved automatically.
- EMS and XMS "share" extended memory, so there's no need to divide extended memory between extended and expanded. Applications can use up to the maximum amount of memory available as EMS, XMS, or a combination of the two, without reconfiguring or rebooting the system.
- QRAM and QEMM, companion products, allow users of 80286 machines to use Loadhi technology, while maintaining an interface consistent with 386 and 486 machines running QEMM–386.

Installing QEMM

To install QEMM–386 on a system where Windows has already been installed:

- Run the QEMM–386 Install program from the command prompt.

The Install program copies the necessary files to your disk and makes the necessary changes to CONFIG.SYS and AUTOEXEC.BAT. Install detects whether Windows is on the machine and adds the entry **SystemROMBreakPoint=false** to the **[386enh]** section of SYSTEM.INI, so that Windows can support the QEMM–386 Stealth feature. Then Install runs the Optimize program to set the system up in an optimum configuration to support Windows and other non-Windows applications you might run.

After installation, Windows 3.1 should operate correctly in both modes with QEMM–386.

To install Windows if QEMM–386 is already installed:

1. Leave the QEMM–386 drivers installed and active on your machine. The Windows Setup program detects QEMM–386 and installs in a way that is completely compatible with it.
2. Run the Qwinfix program in the QEMM directory, which adds the entry **SystemROMBreakPoint=false** to the **[386enh]** section of SYSTEM.INI, so that Windows can support the QEMM–386 Stealth feature.
3. Run the Optimize program in the QEMM directory to ensure that the system is still configured properly after Windows adds or updates drivers in CONFIG.SYS or AUTOEXEC.BAT.

Tips for the Optimize Program

The Optimize program runs automatically when you install QEMM–386 for the first time. This sets up your system to load the resident programs and drivers you are currently using into the most efficient locations in upper memory.

However, if you add or upgrade drivers later, memory requirements may change. Therefore, whenever you change CONFIG.SYS or AUTOEXEC.BAT or install programs or upgrades (such as Windows) that may change these files, always run Optimize from your QEMM–386 directory. It takes only a couple of minutes and keeps your system configured correctly.

Troubleshooting for QEMM-386

Quarterdeck rarely sees problems related to QEMM–386 when running Windows in standard mode. However, because of the range of hardware and software combined in some systems, we have seen problems in some configurations when trying to run Windows in 386 enhanced mode. These problems include failure of Windows to run in 386 enhanced mode, system crashes when exiting Windows, and, in some cases, problems running Windows in 386 enhanced mode.

Windows Fails to Run in 386 Enhanced Mode

If you have trouble starting Windows in 386 enhanced mode, check your system to be sure that it has been set up as suggested in the QEMM documentation. The following items are important to 386 enhanced mode operations.

- **Be sure you have enough free memory.** You should have at least 1 MB of XMS memory free after QEMM–386 has been loaded to run Windows in either standard or 386 enhanced mode. The Manifest utility reports this memory under "Extended/XMS" memory.

  There may be conditions under which you can run Windows in either mode using HIMEM.SYS and EMM386.EXE, but cannot with QEMM–386. This is because QEMM–386 creates more upper memory space than the Microsoft utilities and, as a result, uses more extended memory. If your machine has a smaller amount of extended memory, you may have to reduce the amount of upper memory created by forgoing the Stealth option, or decrease the amount of extended memory being used by SMARTDrive.

- **Be sure you have enough free disk space.** Unless you created a permanent swap file, you should plan to keep 2 to 3 MB of free disk space on the drive that Windows uses for its swap files. Otherwise Windows may not run in 386 enhanced mode.

- **Be sure you have enough file handles available.** Microsoft recommends setting **files=30** in CONFIG.SYS. QEMM's FILES.COM program can be used to load some of these files into upper memory. However, we recommend having at least **files=15** in your CONFIG.SYS file and loading the rest of your files with FILES.COM. Windows may not run in 386 enhanced mode if you do not have a **files=** statement in your CONFIG.SYS, or if it is set below 15.

- **Remove the SORT parameter if your machine has different speeds of memory.** And if your machine usually has less than 640K conventional memory, where QEMM–386 is filling the memory to 640K, you must use the NOFILL (NO) parameter. Although QEMM–386 supports memory sorting and conventional memory filling, Windows cannot support these features and cannot run in 386 enhanced mode when they are active. It is possible to use these features with Windows standard mode, since in this mode QEMM remains active.

Reinstall QEMM–386 from the original disk and allow the Optimize program to run. The Install and Optimize programs will automatically reset many of the parameters necessary for Windows to work correctly, in case any of these have been deleted.

If the Preceding Suggestions Fail . . .

If it is difficult to get Windows to run in 386 enhanced mode, here are some other things to check:

- If the system has shadow RAM, try QEMM–386's NOSHADOWRAM (NOSH) parameter. If you are not sure, try it anyway. The parameter will not have harmful effects if your system does not have shadow RAM.
- If random characters appear on the display, accompanied by beeping, and Windows drops back to the command prompt when you try to run 386 enhanced mode, check for the file WINHIRAM.VXD in the QEMM directory on your hard disk. If you are running QEMM–386 version 6.0 or greater, WINSTLTH.VXD should also be present. Also, make sure these files are identical to the ones that came with the version of QEMM–386 you are running.
- Run the Windows Setup program from the command prompt (not from Program Manager) and make sure you have properly identified your machine. The default Setup selection specifies "MS-DOS System," and many users accept this option. However, Windows provides specific setups for many common PC compatibles such as AST, NCR, NEC, Toshiba, Everex, and Hewlett-Packard. Check this list if you are having problems. If your machine is on the list, use that selection.
- Run Manifest and, in the "Expanded" memory section, verify that the EMS Page Frame has not been set lower than C000 or higher than E000. Although QEMM–386 can establish the frame outside these limits, Windows will not support a frame outside this range and will not run in 386 enhanced mode. If the frame is outside these limits, consult the QEMM–386 documentation regarding moving it.

In some cases the following modifications to SYSTEM.INI have solved problems:

- Add the entries **DualDisplay=True** and **EMMExclude=A000-FFFF** to the **[386enh]** section of SYSTEM.INI.
- If you have an XGA card, replace the line **Display=XGAVDD.386** with the line **Display=*VDDVGA** in the **[386enh]** section of SYSTEM.INI.

  Or use the QEMM "EXCLUDE" parameter to exclude the first 4K of the 32K region where the XGA ROM is located in the 640K-1024K area. For details, see the Quarterdeck Technical Note #208, "XGA Video and Quarterdeck Products."

Problems Running Windows in 386 Enhanced Mode

When Windows runs in 386 enhanced mode, control of memory (including all of the upper memory area) passes to Windows, using drivers that were co-developed by Quarterdeck and Microsoft. When Windows runs in 386 enhanced mode, QEMM–386 is inactive. When you quit Windows, QEMM–386 again takes control. As a result, under 386 enhanced mode, QEMM–386 can neither directly cause problems nor prevent them if they occur. However, QEMM may have loaded some resident programs or may have made changes to the system before being disabled. These changes may later present problems in Windows 386 enhanced mode.

One way to determine whether QEMM–386 causes the problem is to replace the QEMM driver with HIMEM.SYS and EMM386.EXE. Then see if the problem persists. If it does, the problem is not related to QEMM. Follow the troubleshooting steps in Chapter 13, "Troubleshooting Windows 3.1," of the *Windows Resource Kit* to find a solution.

Programs Fail When Loaded High

Some resident programs fail when loaded into the upper memory area because they contain logic that fails when the program is loaded above 640K. With the growing popularity of memory managers such as QEMM–386 that load programs above that barrier, most of these programs have been rewritten, so the likelihood that you will run into one of these programs becomes less and less. Still, if you are having a problem and find the problem goes away when you remove one of your resident programs, you should try loading that program into conventional memory.

Expanded Memory TSRs

QEMM–386 can load resident programs (TSRs) into upper memory and provide expanded memory for their use. However, not all resident programs are compatible with Windows, particularly when they are using expanded memory. If you are running such programs before running Windows and are experiencing problems, try removing them. If the problems go away and the programs can be configured to use XMS, try configuring them that way. If problems persist, contact the manufacturer of the program to ask about compatibility with Windows.

Bus-Mastering Devices

Bus-mastering devices do their own direct memory addressing (DMA). DMA is done without going through the machine's processor. Currently, the most common bus-mastering devices are SCSI hard disk controllers, but technically, other types of devices can also be bus-mastering.

The problem with these devices is that they put data into absolute memory addresses and assume that the content of those memory addresses always remains constant. However, when running Windows 3.1, this can be an incorrect assumption. When Windows is managing memory, the same physical memory addresses can at any given moment hold different data, depending on which virtual machine is current.

To run Windows 3.1 properly with bus-mastering devices, you should obtain a driver from the device manufacturer that meets the VDS (Virtual DMA Services) specification. With such a driver, Windows should be able to operate properly with your bus-mastering device.

When you are not running Windows in 386 enhanced mode, QEMM–386's DISKBUF (DB) parameter can be used to work around these problems. But again, because QEMM–386 must be disabled when Windows runs in 386 enhanced mode, the DISKBUF parameter will have no effect. SMARTDrive can perform this sort of buffering as well. So if you are having trouble running Windows in 386 enhanced mode, loading SMARTDrive may help.

Windows Runs Slowly in 386 Enhanced Mode

A full hard disk impairs the Windows paging abilities and reduces the size and number of programs you can run in Windows. If Windows seems to be slower or your system runs out of memory, check that you have at least 2 MB of free disk space on the drive to which the TEMP environment variable points.

Speed can also be affected by fragmentation of your hard drive. In these cases, setting up a permanent Windows swap file, as outlined in the *Windows User's Guide*, might be useful.

Finally, if your machine does not provide caching of video ROM into fast RAM, you should try using QEMM's ROM parameter, which provides this feature through software. Often, this will speed up the Windows screen display.

Getting Additional Help

If you have problems you have not been able to resolve while running Windows with Quarterdeck products, help is available through the following sources:

- **Q/FAX: 213/314-3214**

FAX-on-demand service. Contains up-to-date technical notes, product and upgrade information. Available 24 hours a day, 7 days a week. Call from your fax machine.

- **Electronic Bulletin Board Service: 310/314-3227**

Available 24 hours a day, 7 days a week.

- **Telephone Support: 310/392-9701**

Available business days, from 9:00 a.m. to 5:00 p.m. Pacific Standard Time

- **CompuServe and BIX**

CompuServe subscribers can reach the Quarterdeck Support Forum by typing **GO PCVENDB** at the prompt. For BIX subscribers, type **GO DESQVIEW.**

PC-NFS and Windows 3.1

The tips that follow will let you more effectively use PC-NFS software. See the PC-NFS documentation for more information about these topics.

Installation Tips

- Install Windows before you install PC-NFS.
- Make sure you have 1 MB of free disk space available to perform a Basic PC-NFS installation, or 3 MB of disk space for a Full PC-NFS installation.
- Sun recommends that you do not use the Windows Setup "Network" option when installing PC-NFS.
- The PC-NFS installation program performs all of the Windows-specific configuration tasks necessary to use PC-NFS with Windows, including modifying files such as PROGMAN.INI and SYSTEM.INI.

  If you choose not to let the installation program modify system files, you must edit SYSTEM.INI and PROGMAN.INI manually. See Chapter 3 of the PC-NFS documentation for information about modifying these files.
- PC-NFS installation copies several .EXE, .DLL, .HLP, .GRP, .ICO, .SYS, .DRV, and .PIF files to the installation directory, and creates new Windows files, including:

WINHOSTS.FIL	Names of NFS file servers to browse
WINHOSTS.PRN	Names of print servers to browse

  Also, PC-NFS installation modifies these existing Windows files:

PROGMAN.INI	Adds PCNFS.GRP
SYSTEM.INI	Adds lines that define PC-NFS drivers

  During its operation, PC-NFS modifies these files:

WINFILE.INI	Adds parameters used by File Manager
WIN.INI	Adds connected drives
PCNFS.INI	Lists default values for mounting drives and printers
- During PC-NFS installation, you will be prompted to assign an interrupt number to any 3COM Ethernet adapter you will use. The installation program uses IRQ 3 as its default value. You must change this number for the adapter to operate properly with Windows. Change the interrupt value to any unused interrupt. IRQ 5 is a commonly available value.
- If you upgrade from PC-NFS version 3.5 to version 4.0, remove any /Y options from the statement **device=pcnfs.sys** to your CONFIG.SYS file.

Using PC-NFS With Windows

- The PC-NFS terminate-and-stay-resident (TSR) component called the Resident Transport Module (RTM) must be loaded before Windows to use PC-NFS and its utilities. This file cannot be started with a WINSTART.BAT file. Sun recommends adding the command-line **rtm.exe** to the AUTOEXEC.BAT file of computers that will run Windows.

  Another way to use the RTM is to create a batch file that loads the RTM, runs Windows, and then unloads the RTM when you exit Windows.
- If you plan to receive messages using the PC-NFS Console Messaging utility, you must load the Listener TSR program before starting Windows.
- Sun recommends a minimum of 4 MB of memory for running PC-NFS with Windows.
- All swap files and temporary files used by Windows must reside on a local (not a network) disk drive. Therefore, be sure that each user's TEMP environment variable refers only to a local disk drive. Also, the SYSTEM.INI settings shown below should not refer to a network drive:

  ```
  [386enh]                    [NonWindowsApp]
  PagingDrive=                SwapDisk=
  ```
- If you manually edit the SYSTEM.INI file, make sure the following entry appears in order to avoid "General Failure - Abort, Retry, Ignore" errors:

  ```
  [386enh]
  UniqueDOSPSP=TRUE
  ```
- To run PC-NFS, Windows, and Norton Desktop 1.0, add the following entry to the PCNFS.INI file in the WINDOWS directory:

  ```
  [Compatibility]
  Norton1.0=1
  ```

FastDisk: An Introduction to 32-Bit Disk Access

The new 32-bit disk access feature in Windows 3.1, also known as "FastDisk," is a system of Windows components that allows Windows to talk directly to the hard drive, bypassing the BIOS.

Normally, in MS-DOS, when an application wants to read from the disk, it follows this sequence:

1. The application makes an INT 21H call to MS-DOS.
2. MS-DOS decides where the requested part of the file exists on the disk.
3. MS-DOS sends an INT 13H call to the disk BIOS, which contains a device driver for each hard disk in the system.
4. The BIOS driver talks directly to the hard drive controller. Each hard drive controller requires its own disk BIOS driver, usually specific to the controller board or adapter that connects the hard drive to the rest of the system.

This process is illustrated in Figure D.6.

If your system has an off-the-shelf disk drive, your computer's ROM BIOS may not know how to talk to it without a specific device driver. Such a driver might be installed in your CONFIG.SYS or, for example, if you use a CD-ROM drive, the MSCDEX file is the device driver that provides the BIOS interface. These drivers trap INT 13H calls and attempt to access the drive without passing it to ROM BIOS.

What Is 32-Bit Disk Access?

32-bit disk access replaces the disk BIOS. Like MSCDEX, it serves as a device driver that talks to the hard drive controller, watching for INT 13H calls and handling them directly or passing them on to the BIOS.

32-bit disk access must exactly match the hard drive controller, not the specific hard drive. Windows 3.1 supports the standard introduced by Western Digital for 1003 controllers. This standard supports all types of hard drive controllers except ESDI and SCSI. And 32-bit disk access is an open standard that Microsoft promotes to all manufacturers of hard disk controllers, so they can write a FastDisk device to support this functionality for their hardware.

However, 32-bit disk access is not required. If you don't have a FastDisk device installed, your system will run the same way it did in Windows 3.0.

For Windows 3.1 in 386 enhanced mode:

- 32-bit disk access speeds up disk access.
- 32-bit disk access allows you to run more non-Windows applications, and it speeds up the system when you are running non-Windows applications in the background.
- 32-bit disk access provides a new mechanism for ISVs to write powerful disk utilities.

FastDisk Speeds Up Disk Access

On many machines, the disk BIOS is slow and unwieldy. When a FastDisk replaces the BIOS functions, many systems show impressive performance improvements. Also, 32-bit disk access allows Windows to handle BIOS requests in protected mode, rather than in real mode. Under 386 enhanced mode, the disk-reading sequence described at the beginning of this article becomes more complicated, because Windows and Windows applications run in protected mode, yet the system must switch to real mode (or virtual 8086 mode) in the background to run code such as MS-DOS and the BIOS. Switching modes is time consuming.

Figure D.6: Disk access without FastDisk

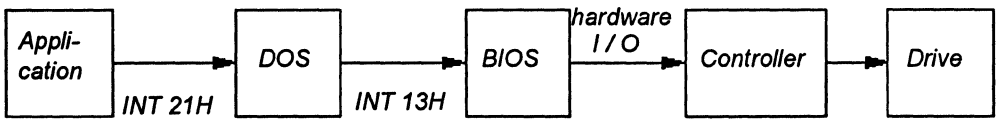
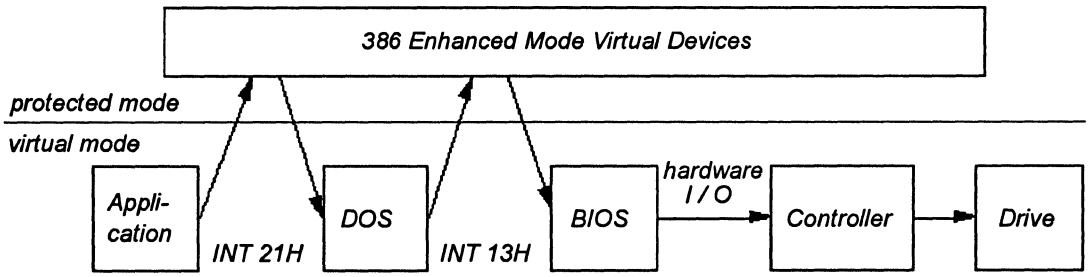


Figure D.7: Disk access in 386 enhanced mode without FastDisk



For this process in 386 enhanced mode, the system begins in virtual mode while running the application. When the application makes a call to read from a file, Windows traps this interrupt and switches to protected mode, where several virtual devices check the call for actions they need to perform. If none of the devices intervenes, Windows hands the call to MS-DOS and switches back to virtual mode. MS-DOS finds the requested location on the disk, then generates an INT 13H call to the BIOS. Again, Windows traps that call and switches to protected mode for additional processing before handing it back to the BIOS, then switches back to virtual mode. The BIOS passes the call to the hard drive controller to set the physical read in motion, waits until that physical read is finished and returns from the INT 13H call. Windows switches back to protected mode to perform more processing, then returns to virtual mode to let MS-DOS see the return and perform more processing before returning from INT 21H back to protected mode. Windows again performs additional processing before finally switching back to virtual mode for the application that started this process.

With 32-bit access, a FastDisk device is inserted to trap INT 13H calls and handle them in protected mode. This saves two mode transitions and perhaps more, because MS-DOS might have performed many INT 13H calls for the task. This is illustrated in Figure D.8.

FastDrisk Allows You to Run More Non-Windows Applications

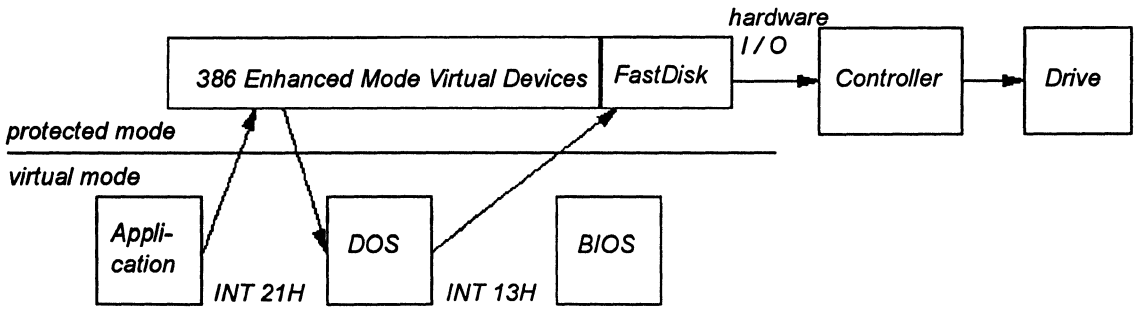
Users often complain that when the Help About dialog box shows, for example, 14 MB of free memory they can run large Windows applications, but they get out-of-memory messages when trying to start another non-Windows application. And even though all that memory is available, the Windows applications run slowly and hit the disk constantly.

These discrepancies occur because of the difference between virtual memory (of which you often have a great deal) and physical RAM (which is usually still limited). Virtual memory might be four times the amount of physical RAM, or even more under Windows 3.1. Although virtual memory is great for Windows applications, it doesn't always fill the need.

Each non-Windows application runs in its own virtual machine, taking up 640K or more of virtual memory. But when that virtual machine is running, all of it must occupy physical RAM, even if the application is idle in the background.

Windows applications can be paged to disk so that only the parts being used take up physical memory. But paging for non-Windows applications presents problems. If part of a non-Windows application were

Figure D.8: Disk access in 386 enhanced mode with FastDisk



allowed to be paged to the disk, it might happen to be the application's data buffer. Windows would think that it's okay, because it can read the data back when the application tries to touch it. But suppose the application makes a call to MS-DOS to read data from the disk into that buffer. MS-DOS begins handling the call, then sends an INT 13H call to the BIOS, which in turn talks to the hard disk controller until data begins to be transferred into the buffer. Windows see that it needs to page the buffer that's currently on disk back into memory, but it can't call MS-DOS, because MS-DOS is already busy and cannot be reentered—if Windows calls again, the call fails. Windows can't read the data by calling the BIOS, because the BIOS is also in the middle of a call and also can't be reentered. So Windows can't let the current call finish until it pages in the buffer, and it can't page the buffer back in until the current call finishes.

The only way to ensure that this situation does not occur has been to avoid paging in anything while the non-Windows application is in the middle of an MS-DOS or BIOS call. Windows has done this by never paging an active non-Windows application out to disk. In other words, all 640K or more of that virtual machine had to be in physical memory.

This rapidly reduces free RAM. The Help About box might show that your system has plenty of memory, but it is virtual memory. If you try to start another non-Windows application, which requires yet another 640K of physical RAM, it can't be done.

In Windows 3.0, if you had a swap file, Windows could page using INT 13H BIOS calls. Windows 3.1 can do this with calls to the 32-bit disk access system, which can queue multiple requests and is completely reentrant. It doesn't rely on MS-DOS or the BIOS, so it can always page data in from the hard drive. The application in the virtual machine can be completely paged, no longer taking up physical RAM.

With 32-bit access, if you start three non-Windows applications, and then start another, it will work. When the Help About dialog box shows 14 MB, you can actually use all 14 MB.

FastDisk Makes the System More Responsive for Non-Windows Applications

By paging non-Windows applications with 32-bit access, Windows can free more RAM for running multiple non-Windows applications. You can run more applications, and the system runs more quickly.

You may also notice a difference when switching between non-Windows applications.

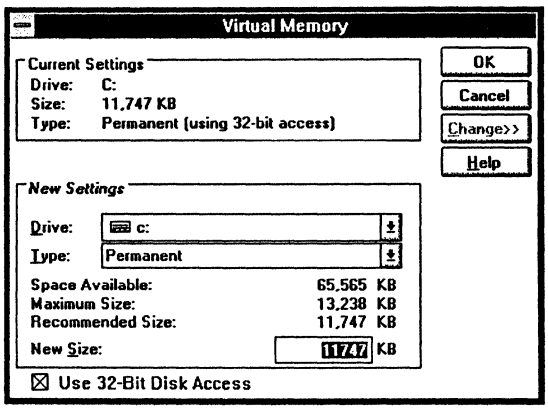
Without 32-bit-access, if a non-Windows application were configured so that it could not run in the background, then it could be swapped out to disk. But when you press ALT+TAB, it could take a long time for the applica-tion to be read into RAM. By paging with 32-bit disk access, Windows can load it into just the amount of memory that the application is actually using, so task switching becomes almost instantaneous.

FastDisk Enables New Classes of Applications

32-bit disk access devices provide services to other virtual devices. Those services can also be provided to applications. New applications and utilities will soon appear that rely completely on 32-bit disk access. These utilities might, for example, allow applications to perform asynchronous I/O, reading and writing unobtrusively in the background, or might use a FastDisk for a tape drive to back up your system from inside Windows. These applications could not be written without 32-bit disk access, but this means that 32-bit access will be required for some applications.

The User Interface for 32-Bit Disk Access

32-bit disk access is turned on and off in the Control Panel through a checkbox in the 386 Enhanced dialog box that appears if FastDisk drivers are available for one or more of the controllers on the machine. Click the checkbox in Control Panel to toggle the value for **32BitDiskAccess=** in the **[386enh]** from **Off** to **On**. You must restart Windows for the change to take effect.



You can also run Windows at the command prompt with the **win /d:f** switch to disable FastDisk temporarily (for that Windows session).

The Components of the 32-Bit Disk Access System

The 32-bit disk access system consists of these 386 enhanced mode virtual devices:

- **WDCtrl,** which is the FastDisk device that talks to standard Western Digital 1003 or ST506 hard drive controllers (about 90 percent of existing disk controllers). This device is only installed if Windows Setup detects a compatible hard drive controller.
- **BlockDev,** which coordinates devices calling block I/O services and the FastDisk devices that provide those services for specific hard drive controllers. This device is always installed.
- **PageFile,** which handles the virtual memory paging file. It calls through BlockDev if any FastDisk devices are available. This device is always installed.
- **Int13,** which traps and emulates INT 13H BIOS calls by calling BlockDev. This device is only installed when at least one FastDisk device is present.

Because WDCtrl is the only FastDisk device included with Windows 3.1, the standard components only support 32-bit disk access on Western Digital-compatible controllers. When you run Setup, it automatically detects these controllers and, if these controllers are present, adds the following lines to the **[386enh]** section of SYSTEM.INI:

```
32BitDiskAccess=off
device=*int13
device=*wdctrl
```

The 32-bit disk access is turned off by default.
If you turn on 32-bit disk access in the Virtual Memory dialog box, then **32BitDiskAccess=** is reset to **On**.
If you turn off 32-bit disk access, you do not need to delete any files from your hard disk, because the devices are built into WIN386.EXE.

The Risks of 32-Bit Disk Access

Microsoft has identified machines where WDCtrl can crash the system, usually in cases where hard disk controllers appear to be 1003 compatible, but are not. WDCtrl has been made as safe as possible. Each time it starts, it tests to ensure that it communicates properly with the hard drive controller. After much redundant checking, WDCtrl tries to read data from the disk, and, if that passes, it tries writing to the disk and reading it back. Only if all this passes does FastDisk continue and Windows is started.

However, there are drives that appear to be 1003 compatible but are not, and that are not detected. Also, there can be problems even with compatible controllers on some portable computers, specifically those that power down the hard disk to conserve power without telling the running software. If the disk is powered down and back up while its state is being tracked by WDCtrl, the state information will be incorrect, and serious damage can result.

This is why 32-bit disk access is not turned on by default. You must decide to turn it on in the Control Panel. For OEMs who preinstall Windows on a known system, however, 32-bit disk access can be turned on by default in complete safety.

If the initial WDCtrl tests fail, you will not know that anything is wrong. You just won't receive the benefit of 32-bit disk access. If a later test fails, an error message appears, warning you that something has gone wrong and advising you to reboot.

Index

; as .INI file comment character, 159
! as field delimiter, 128
* denoting built-in devices in SYSTEM.INI entries, 100, 193
^ in WIN.INI, 171

.2GR grabber files, 94, 290, 184, 409, 410
[.3gr] font sections in SETUP.INF, 96
32-bit disk access, 193, 245, 520-523
386 enhanced mode Windows. *See also* Standard mode Windows; Windows 3.1
  described, 63, 238-239
  memory
    expanded memory, 238, 240-243, 249
    extended memory, 239
    system requirements, 9, 63, 217, 420
    virtual memory, 243-247
  networks, 36, 370-373
  optimizing, 260, 271
  removing support for, 502
  troubleshooting, 34-36, 420-422
[386enh] section in SYSTEM.INI, 117, 193-218
.3GR grabber files, 94, 184, 290, 409, 410
386MAX memory manager, 34, 93, 154, 243, 247, 373, 421
[386 max] section in SETUP.INF, 93
3C505 Network Adapter Card, 375
3Com networks, 102, 143, 374-375
80286 and 80386 machines
  files installed during Setup, 105-106,
  memory, 228-229, 234, 247
  system requirements, 9
8086 and 8088 machines, 229, 231
8514 display, 104-105, 139, 150, 312-314, 411

A

A20 handler, 35, 44, 118, 193, 418
Acer computer, 35, 418
Adapter card. *See* Display adapter
Adapter segment, 231, 240, 241, 447, 455
Ad Lib, *See* MIDI
Administrative Setup (setup /a), 72-74
Adobe Type Manager (ATM), 28, 32, 44, 315, 336
Advanced Power Management driver, 140, 150
All ChargeCard, 375
ALR computers, troubleshooting, 12
AMI BIOS, troubleshooting, 12
Anarkey, 405
ANSI character set, 76, 188, 310, 315, 317
API (application programming interface), defined, 447
Append utility (MS-DOS), 405, 414
Apple LaserWriter Plus, 333
Application. *See also* Non-Windows applications; Windows applications
  defined, 284, 447
  files copied during Windows installation, 107-108
  running at Windows startup, 168
  Windows *vs.* non-Windows, described, 284
Application execution error, 39
Application swap files, 243-244, 290
APPS.INF
  content summary, 121-125
  customizing, 133
  editing procedure, 87
  format, 88-89
  non-Windows applications with PIF data in, 296-297
Apricot Laser printer, 325
ARCNet, 36, 11
Arial font, 113, 310, 311, 317-320
Artisoft LANtastic, 31, 113, 102-103, 375-377, 406
ASCII character set, 310
ASP Integrity Toolkit, 404
Aspect ratio, and fonts, 314
AST computer, troubleshooting, BIOS, 12
AT computer. *See* IBM AT
ATI display adapter, 411
AT&T computer, 138, 139, 149, 187-188, 418
Attrib command (MS-DOS), 369
Audio CD, troubleshooting, 49, 435
Austin computers, troubleshooting, 12
Auto-detection for Setup, troubleshooting, 401, 432
AUTOEXEC.BAT
  editing, with System Editor, 161
  modifying during Setup, 369
  optimizing MS-DOS 5.0 configuration, 248
  running a shared copy of Windows, 368
  troubleshooting, 13-14, 398, 413, 416, 432
AutoFont Support files, 315
Automated Setup (setup /h), 75-81
Award BIOS, troubleshooting, 12

B

Backfilling and bank switching, 211, 231, 234, 447
Banyan VINES, 31, 102-103, 143, 150, 377-378
Batch Mode Setup. *See* Automated Setup
Baud rate, 174
BIOS (basic input/output system)
  EBIOS identified in Setup, 120
  SYSTEM.INI entries, 199
  troubleshooting, 12, 23, 37, 421
Bitmap files, for wallpaper, 153-154
Bitstream fonts, 29, 34, 315, 336, 426
Block devices, 34, 113
[Blowaway] section in SETUP.INF, 116
[Blue Max] section in SETUP.INF, 93
Bold
  document and syntax conventions, 4, 6, 159
  font characteristic, 310, 313
Boolean, as .INI entry value, 159
[Boot] section in SYSTEM.INI, 184-186
[Boot.Description] section in SYSTEM.INI, 187
Boot disk, creating, 13, 14, 398
BOOT.SYS, 406
Break points, 206, 213
Buffers
  COM port, 196
  data transfer, 209, 372
  defined, 447
  DMA, 198, 201, 208, 372

Buffers, *continued*
double buffering, defined, 449
FIFO, 196
hard disk, 201
keyboard, 202
network, 208-209, 218, 372
non-Windows applications requirements, 190
permanent swap file, 210
printing from serial ports, 344
settings in CONFIG.SYS, 259
Building-font-database message, 328-329

C

Cannot-find-file error message, 414-415
Canon Laser Beam printer fonts, 336-337
Caret (^), and filename extensions, 171
CD (audio) troubleshooting, 49
CD-ROM, 21, 119, 435-436
CEMM.EXE, 243, 247, 448
CGA display, 145, 147, 149, 150
Character-based interface, defined, 448
Character sets, 314
ANSI, 310, 315, 321
ASCII, 310
extended character codes, 321
matching screen fonts to printer fonts, 316
OEM, 310, 315, 321
Windows 3.1, 316, 317
Characters-per-inch (CPI), 311, 322
Chips and Technologies, troubleshooting, 12, 410-411
Chkdsk command (MS-DOS), 260
Clean-boot floppy, 13, 14, 398
Client application, defined, 353, 448
Clipboard
data formats and OLE, 355-356
non-Windows applications, 291
Clock speed, 448
CMOS, 12, 25, 213, 430, 448
Code pages
defined, 448
files in Setup, 97, 99
translation tables, 147, 188
U.S., 146, 448
[Codepages] section in SETUP.INF, 99
Code segments, kinds of, 236
Color, 182, 207, 222
[Colors] section in WIN.INI, 182
COMMAND.COM, for troubleshooting, 13-14
Command environment, 189
Command prompt, 448
Comments, in initialization files, 159
Communications driver, 140, 184
Compaq, MS-DOS version, 35, 433
COM port
in 386 enhanced mode, 206
buffering, 196
configuring, 174, 195-197
driver, 140, 184
hardware interrupts, 25, 197, 343
printing from, 343-345
ROM BIOS address conflicts, 346-348
troubleshooting, 19, 31, 427-428
Compound documents
Clipboard support, 354-356
described, 352-358, 448
Compound documents, *continued*
Edit menu commands supporting, 354, 357-358
File Manager support, 354-355
Compressed files, and installing Windows 3.1, 59
Computers
compatible with Windows 3.1, 488-494
identified in Setup, 117-118
specifying for Automated Setup, 78
CONFIG.SYS
editing, with System Editor, 161
network adapter cards in, 371
optimizing performance, 248, 259
troubleshooting, 13-14, 398, 413, 432
Context-sensitive help, 273, 277, 300
CONTROL.INF
content summary, 126-128
editing procedure, 87
format, 88-89
Windows installation, 77
CONTROL.INI, 161, 222
Control Panel
changing values in .INI files, 159
initialization (.INI) file, 222
installation information (.INF) file, 126-128
maintaining Windows with, 82
multimedia extensions, 46
supporting files, 151
CONTROL.SRC, 161
Conventional memory
described, 228-229
freeing, 263
operating problems, 44
specifying for PIFs, 301
Windows operating modes, 217, 235, 238
Copy command (MS-DOS), 59, 73
Core files (Windows), 136-137, 250-251
[Country] section in CONTROL.INF, 128
Courier font, 112, 313, 314, 317, 320, 336
Courier New font, 318, 320
CPU (central processing unit), 448
CTRL+ALT+DEL
defining behavior, in SYSTEM.INI, 204
when to use, 256, 294-295
Cursor, WIN.INI entry, 164
Customizing files
changing .INF files for
custom applications, 130
custom groups to Program Manager, 132
custom PIF information, 133
custom Setup, 128
creating custom installation script, 77-78

D

Data Physician Plus, 404
[Data] section in SETUP.INF, 90
Date, defining formats, 172
DDE (dynamic data exchange), defined, 439
Debugging COM ports, 19, 346-347
DEC Pathworks, 102, 143 150, 381, 379-382
Default
in .INI files, 159
printer, WIN.INI entries, 165-166
_DEFAULT.PIF, 285, 289, 294, 300, 305-306
Deleting files, 155, 500-504
[DelFiles] section in SETUP.INF, 107-108

Demand paging, 448, 451
Desktop, 169-170, 182, 403, 442
[Desktop] section in WIN.INI, 169, 170
Destination document (OLE), 353
Device
  defined in WIN.INI, 181
  global, 201
  synonyms in SYSTEM.INI, 183
Device command (MS-DOS), 264
Device contention, and non-Windows applications, 291
Device drivers. *See also* specific device driver types
  defined, 449
  memory management, 230
  specifying during installation, 77-78
  SYSTEM.INI entries, 187, 204
  troubleshooting, 10, 34, 46, 430
Devicehigh command (MS-DOS), 248, 264-265, 413
[Devices] section in WIN.INI, 181
[Dialog] section in SETUP.INF, 90
Direct Graphics Interface Standard (DGIS) display adapter, 412
Direct memory access. *See* DMA
Discardable code segment (in Windows applications), 236, 449
Disk cache, 416, 441, 449. *See also* SMARTDrive
Disk compacting utility, 245, 260
Diskless workstation, 74, 259, 262, 369
Disk partition, troubleshooting, 13-14
Disk space. *See also* Swap files
  freeing by deleting files, 155, 500-504
  TrueType requirements, 320
[Disks] section in SETUP.INF, 91
Disk swapping. *See* Swap files
Display adapters
  compatible with Windows 3.1, 139, 495
  resolutions, supported by Windows 3.1, 144-145
  SYSTEM.INI entries, 194, 198
  troubleshooting, 20, 26, 408-420
Display driver
  files, 139
  identified in Setup, 94-96
  optimizing performance, 260
  SYSTEM.INI entry, 185
  troubleshooting, 20, 27, 37, 44, 403, 408-412
[Display] section in SETUP.INF, 95, 129
DLL (dynamic-link library), 449
.DLL filename extension, 138
DMA (direct memory access)
  buffers, 198, 201
  defined, 449
  EISA machines, 199
  Micro Channel machines, 207
  network and paging, 207
  page addresses, 206
[Dontfind] section in APPS.INF, 121
DOS. *See* MS-DOS
Doscue, 405
DOS extender. *See* Extended memory manager
[Dos mouse.driver] section in SETUP.INF, 101
DOSX.EXE, 90, 136, 150, 290
Dot matrix printers, 317, 322, 340
Double buffering, 70, 201, 255-256, 439
Double-click, defining, 165
Doubledisk, 406
Downloadable soft fonts
  changing typeface names for, 337-338
  described, 312, 320-321, 330, 449
Downloadable soft fonts, *continued*
  font installers, 140-141, 326-328
  HPPCL, 324-330
  page limit message, 330
  PostScript, 336
  TrueType installation, 320
DPMI (DOS Protected Mode Interface), 237, 239, 295, 449
Drivers. *See also* Display drivers; Keyboard drivers; Memory drivers; Mouse drivers; Multimedia drivers; Printer drivers, Sound drivers; System drivers
  described, 136-137
  identified in Setup, 94-97, 100-101, 113-114, 126
  incompatible drivers, in Setup, 58, 66
  MS-DOS drivers for use with Windows 3.1, 148
  removing files, 500
  requirements in UMBs, 264-266
  switches for MS-DOS system drivers, 258
  SYSTEM.INI entries, 185
  third-party, troubleshooting, 10, 27, 31, 399
  virtual support files, 150
  Windows Driver Library, 443
[Drivers] section in SYSTEM.INI, 187
.DRV (printer driver) file, 340
Dr. Watson, and general protection faults, 37, 434
DSWAP.EXE file, 150
DTK BIOS, troubleshooting, 12
Dynamic data exchange (DDE), defined, 449
Dynamic-link library (DLL), defined, 449, 452

E

[Ebios] section in SETUP.INF, 120
Editing
  .INI and .SRC files, 159-161
  using System Editor, 160-161
Edit menu commands and OLE, 357-358
Edit utility (MS-DOS), 160
EGA display device
  driver and grabber files, 139, 149
  fonts, 104-105, 312-314
  SYSTEM.INI entries, 199
  troubleshooting, 27
  virtual support files, 150
EGA.SYS, 148
EISA (Extended Industry Standard Architecture) bus machines, 197, 199, 240, 449
[Embedding] section in WIN.INI, 178
EMM. *See* Expanded memory manager
EMM386.EXE
  backfilling with, 234
  defined, 230, 449
  deleting file, 155
  driver, 148
  emulating expanded memory, 228, 231, 258, 296
  incompatibility with memory expansion board, 259
  loading in CONFIG.SYS, 259, 264, 371
  network issues, 273, 284
  non-Windows applications, 238, 240, 242, 243, 296
  optimizing performance, 266-268, 268-269
  troubleshooting, 413-417
EMMExclude=
  display driver conflicts, 411
  excluding UMB ranges, 199-200, 204, 212, 215
  troubleshooting UMB conflicts, 416-417, 420-421, 433
EMS. *See* Expanded Memory Specification
EMS memory. *See* Expanded memory

Environment variables, 189, 301
EPT ports, 174, 345-346
Error 20 message, 330
ESDI controllers, 421
Etherlink, identified in Setup, 119
Ethernet, troubleshooting, 11, 36
Exec command (MS-DOS), 136
Expanded memory. *See also* Expanded memory managers (EMM); Expanded Memory Specification (EMS)
  board or card, 228, 234, 237-238, 242, 296
  described, 228-229, 235, 449
  driver, 209
  LIM 3.2 and 4.0 EMS, 231, *illus.* 233
  problems with, 235
  upper memory area, *illus.* 232
  in Windows
    386 enhanced mode, 238, 240-243
    non-Windows applications, 230, 241, 301, 303
    standard mode, 237-238
    SYSTEM.INI entries, 194, 199-200, 209, 213, 214
    troubleshooting, 415-417
Expanded memory managers (EMM). *See also* EMM386.EXE
  386 managers, 242-243, 456
  backfilling and bank switching, 234
  emulating expanded memory, 228, 230, 231, 258, 296, 449
  ignoring third-party managers, 201
  QEMM–386, 454, 515-518
  troubleshooting, 23-34
Expanded Memory Specification (EMS), 230-235, 449, 450
Expanding files, 61, 152
Extended memory. *See also* Extended memory managers
  described, 228-229, 450
  requirements for
    386 enhanced mode, 9, 238-239, 250, 420
    non-Windows applications, 237, 301, 303, 424
    standard mode, 9, 235-237, 418
    Windows, 9, 418, 420
  SYSTEM.INI entries, 194, 213-214
  troubleshooting, 34, 424
Extended memory managers (XMM). *See also* HIMEM.SYS
  defined, 450
  drivers, 236, 418-420
  troubleshooting, for standard mode, 418-419
  Windows 3.1, 150, 237, 247-248
  Windows Setup, 90
Extended Memory Specification (XMS), 230, 450
[Extensions] section in WIN.INI, 171

F

FaceLift. *See* Bitstream FaceLift
Fastopen utility (MS-DOS), 406
File. *See also* specific types
  copying
    additional files to a user's directory, 130
    from installation disks, 63
    files for Windows installation, 105-108
  corruption, 43
  deleted in Setup, 107-108
  deleting to free disk space, 155, 500-504
  expanding, 17
  filenames translated in Setup, 115
  ignored in Setup, 121
  miscellaneous hardware and other support files, 154
  printing to a file, 31, 33
  removing, 500-504
file= settings in CONFIG.SYS, 259
FILE:, WIN.INI entry, 174
File allocation table (FAT), 262
File handles, 259, 373
File Manager
  changing values in .INI files, 159, 223
  customizing, 278-279
  initialization (.INI) files, 223
  new features in Windows 3.1, 277
  OLE support, 279, 354-355
Filename extensions
  associated with filenames, 171
  defined in WIN.INI, 171
  .PIF, 287
Filenames
  translated in Setup, 115
  working with long filenames, 370
File Size Information, Windows Resource Kit utility, 441
FINSTALL.DIR file, 327-328
Fixed font, 144, 311, 316
Floppy drive, troubleshooting, 22-23, 25, 202
Font. *See also* Downloadable soft fonts; HPPCL printer fonts; PostScript fonts; Printer fonts
  cartridges
    dot-matrix printers, 322
    HP LaserJet-compatible, 323-324
    installing in printers, 316
    Microsoft 1Z font cartridge, 314
    PostScript, 336
  converters, 27, 425
  dot matrix printer fonts, 317, 322, 340
  duplicate font names, 316-317
  files, 104-105, 112, 144-147, 501
  font families in Windows, 310
  font matching in Windows, 316-317
  installing, 315-316
  kinds of
    printer fonts, 312, 320-337
    raster fonts, 311, 313-314
    screen fonts, 175, 312
    TrueType fonts, 311, 318-320
    vector fonts, 311, 315. 317, 456
  substituting font names, 175
Font cache, 318-319
FontEdit utility, 315
Font Installer, for HP LaserJet, 315-316, 321, 324-337
Font mapping table, 316
Font Metrics dialog box, 338
[Fonts] section in WIN.INI, 175
[fontSubstitutes] section in WIN.INI, 117, 175
.FOT file, 318

G

GDI
  .EXE file, 137, 250
  heap, 251, 442
General protection (GP) faults, 27, 37-38, 431-434
Grabber
  application execution error, 39
  defined, 94, 408, 450
  files, 148-149, 290, 292
  identified in Setup, 96
  SYSTEM.INI entries, 184
  troubleshooting, 26, 408-410, 423
Graphical user interface (GUI), defined, 450

Graphics adapter. *See* Display adapter
Graphics utility (MS-DOS), 405
Graphics Viewer, Windows Resource Kit utility, 440
Groups. *See* Program Manager group
[Group#] sections in SETUP.INF, 110-113, 133
[Groups] section in PROGMAN.INI, 220
.GRP file, 369
GUI (graphical user interface), 450

H

Handshaking, 342, 344
Hard disk
  correcting interleave without reformatting, 258
  ensuring maximum space, 263
  interrupts, 25, 217
  optimizing performance, 258-260
  printing, space requirements, 30, 32
  Stacker, 17
  troubleshooting, 24, 34
Hardware
  adapters, and problems with expanded memory, 235
  compatible with Windows 3.1, 487-498
  creating custom Setup, 129
  general protection faults, 37
  manufacturer support, 482-483
  multimedia, 47-51
  optimizing performance, 258-259, 271
  support files, 154
  troubleshooting, 21-26, 35, 44, 430, 442
Headland Technologies display adapter, 410
Help window, WIN.INI entries, 178-179
Helvetica font, 175, 320
Helv font, 107, 314
Hewlett-Packard
  EBIOS for 80386 computer, 120, 150
  Intellifont, disabling, 28
  keyboard driver, 135
  mouse driver, 139, 148
  MS-DOS version, 35
  plotters, 332, 427
  Printer Control Language (HPPCL), 323
  printer drivers, 140
  printer fonts, 323-333
  soft font installer files, 141
  system driver (for HP Vectra), 138
  Type Director, 315
  Vectra, 138, 419
  virtual support files, 150
High area. *See* High memory area (HMA); Upper memory area
High memory area (HMA)
  defined, 228, 450
  disabling a network's HMA option, 373
  loading MS-DOS 5.0 into, 248, 263-264
  troubleshooting, 413-415
HIMEM.SYS
  accessing extended memory, 229, 230, 236, 239
  file, 148
  loading in CONFIG.SYS, 259
  optimizing, 248
  troubleshooting, 412, 416, 418
"Hints," for TrueType, 318
HP DeskJet printer fonts, 331-332
HP PaintJet printer fonts, 332
HPPCL-compatible printers
  driver files, 141
  fonts, 317, 321, 323-331
  setting up unsupported printers in Windows, 340
  types of, 330, 331
HPPCL printer fonts
  described, 323
  downloadable soft fonts
    font summary file, 328-329
    installing, 324
    permanent *vs.* temporary downloading, 325-326
    recovering fonts after reinstalling Windows, 326-328
    soft font limits, 315, 330
  font cartridges, 323-324
  memory settings for, 323
HP plotter printer fonts, 332

I

IBM 4019 Laser Printer fonts, 337
IBM computer
  AT, A20 handler, 419
  DOS LAN requester. *See* IBM computer, OS/2 LAN Server
  MS-DOS version, 35, 433
  OS/2 LAN Server, 102, 383
  PC LAN Program, 102, 143, 383-384
  PS/2 computer, 24, 25, 150, 204, 412
IDE drive, troubleshooting, 24, 217
[incompTSR#] sections in SETUP.INF, 113, 401
Independent software vendor (ISV), defined, 450
.INI file, 369
Initialization (.INI) files. *See also* CONTROL.INI; PROGMAN.INI; SYSTEM.INI; WIN.INI; WINFILE.INI
  content summary, 158-159
  editing procedure, 159-161
  format, 158, 159
  preserving, after reinstalling Windows, 15
  source files, 161
  updated in Setup, 116
[Ini.upd.*] sections in SETUP.INF, 116
[Installable drivers] section in SETUP.INF, 114
Installation disks, 59
Installation information (.INF) files. *See also* APPS.INF; CONTROL.INF; SETUP.INF
  content summary, 86-87
  customizing, 128-133
  editing procedure, 87
  format, 158-159
Installing Windows 3.1
  Administrative Setup option (setup /a), 72-74
  AUTOEXEC.BAT modifications, 368-369
  Automated Setup (setup /h), 58, 74-81
  changing values in .INI files, 159
  creating system settings scripts for multiple systems, 76-81
  customizing, 60-61, 74, 76-81, 128-134
  Custom Setup, 58, 62, 64-65
  Express Setup, 58, 62, 64
  from floppy disks, 59
  hardware identification, 65-66
  installation information (.INF) file, 88-120
  maintaining Windows with Setup, 82-83
  MS-DOS Mode Setup, 58, 63-68, 401-402
  Network Setup (setup /n), 73-75
  new Setup features, 58
  overview, 57-62
  setting up multiple systems

Installing Windows 3.1, *continued*
Automated Setup option (setup /h), 72, 75-76
creating a system settings script, 76-81
setting up on a network
Administrative Setup option (setup /a), 72-74
disabling pop-up service to prevent Setup failure, 370
Network Setup option (setup /n), 72, 74-75
on network server *vs.* workstations, 74
overview, 71-72
shared Windows files, 72-73
setting up on a single workstation, 71
specifying the WINDOWS directory, 64
switches for Windows Setup command, 60
SYSTEM.INI settings, 370-371
technical discussion, 62-71
troubleshooting, 9-17, 399-403
upgrading from an earlier version, 64-65
Windows Mode Setup, 58, 68-71, 402-403
Insufficient memory, 43
INT 13, 252-253, 520-522
INT 21, 349, 350, 520-522
Intel, snapin, 24, 386
Intellifont. *See* Hewlett-Packard Intellifont
International support in Windows
formats defined for Control Panel, 128, 172-173
support files, 138, 147
WIN.INI entries, 172-173
Interrupts
32-bit disk access, 520-522
COM ports, 19, 196, 197
defined, 451
floppy drives, 22
hard disks, 24, 217
keyboards, 193, 194, 203
list of, 25
memory-resident software, 202, 211
mouse, 18, 208
network, 36, 430
printers, 165, 213
system timer, 214
virtual machines, 201
[Intl] section in WIN.INI, 172-173
Invisible Software—NET/30 For Windows, 384
I/O port, troubleshooting, 430
[IO.*] sections in CONTROL.INF, 126-129
IPX Upgrade utility from Microsoft, 444
IRQ. *See* Interrupts
ISA computers (adapter locations), 240-241, 372, 433, 441
ISV (independent software vendor), 450
Italic
document and syntax conventions, 4-6
font characteristic, 310, 313

J

Join utility (MS-DOS), 405

K

Kernel, 90, 107, 137, 250, 451
Keyboard
buffer, 202
compatible with Windows 3.1, 495
defining behavior, 166
drivers, 97, 138, 185, 187, 202
identified in Setup, 97
interrupts, 25, 203
Keyboard, *continued*
language tables, 138
password, 204
reboot, 204, 294-295
tables, 98, 128, 138
timeslicing, 202
troubleshooting, 12, 422
[Keyboard.drivers] section in SETUP.INF, 97
[Keyboard] section in SYSTEM.INI, 187-188
[Keyboard.tables] section in SETUP.INF, 98
[Keyboard.types] section in SETUP.INF, 97, 129
Keyname, defined, 87, 158

L

Language
code page, 188
defining international usage, 173
drivers and libraries, 138, 147, 185
identified in Setup, 120
supported by Windows 3.1, 147, 458
used for Windows Setup, 91
[Language] section in SETUP.INF, 120
LAN Manager 348, 451, 456. *See also* specific networks
LanSight, 405
LANtastic. *See* Artisoft LANtastic
Laser printers, and unsupported printer compatibility, 340
LaserTools, 406
LATCP DEFINE SCB command, 380
Le Menu, 406
LIM. *See* Expanded Memory Specification (EMS)
[Lmouse] section in SETUP.INF, 101
Loadhigh command (MS-DOS)
optimizing performance with, 248, 264-265
SYSTEM.INI entries, 204
troubleshooting, 413, 416
Local heap, 251, 451
Lockit utility, 405
Logitech mouse
driver, 139, 148, 158
identified in Setup, 101
troubleshooting, 18, 407
virtual support file, 150
Lotus 1-2-3, troubleshooting, 424-425
Lotus/Intel/Microsoft (LIM). *See* Expanded Memory Specification (EMS)
LPT port
configuring, in WIN.INI, 174
default I/O starting locations for, 343
hardware interrupts, 25, 343
printing from, 342-343
troubleshooting, 31, 426-427
LPTx.DOS
printing method, 31, 174, 350, 376
troubleshooting, 427-428
WIN.INI entry, 174
LPTx.OS2. *See* LPTx.DOS
LRU (Least Recently Used) paging, 205, 246-247

M

[Machine] section in SETUP.INF, 117-118, 129
MAP ROOT command, 388
Math coprocessor, interrupts, 25
MCA. *See* Micro Channel (MCA) computers
MCI (Media Control Interface), 143, 451
[MCI Extensions] section in WIN.INI, 177

[MCI] section in SYSTEM.INI, 189
Measures, defining formats, 172
Media files, 177
Media Player, troubleshooting, 47, 437
Mem command (MS-DOS)
  troubleshooting with, 413, 417, 420
  viewing system memory, 229-230, 264
Memory. *See also* 386 enhanced mode memory; Expanded memory; Extended memory; High memory area (HMA); Memory drivers; Standard mode memory; Upper memory area; Virtual memory
  conventional memory, 228, 260, 263, 448
  DPMI *vs.* VCPI specification, 247-248
  excluding upper memory settings, 372
  memory conflicts, 238-239
  memory protection, 451
  Microsoft Diagnostics utility (MSD.EXE), 229
  network memory management with Windows 3.1, 370-372
  non-Windows applications and PIF settings, 229, 301-305
  optimizing performance, 258-259, 262
  out-of-memory errors, 43
  overview, 228-229
  relative addresses of different kinds of, 229
  requirements, 9, 249-250
  types of, 228-229
  viewing, in your system, 229
  Windows system resources, 250-251
Memory drivers, 230
Memory expansion board, 258-259
Memory managers, 229, 230, 373. *See also* Expanded memory managers; Extended memory managers; Virtual Memory Manager (VMM)
Memory-resident software. *See* TSRs
Micro Channel (MCA) computers
  changing memory settings for network, 372
  COM port settings, 197
  DMA extensions, 207-208
  interrupts, 197, 433
  troubleshooting, 12
Microsoft 1Z font cartridge, 314
Microsoft Authorized Training Centers (ATCs), 460-466
Microsoft Certified Professional Program, 459
Microsoft Consultant Relations Program (CRP), 458
Microsoft Consulting Services (MCS), 452, 458
Microsoft Diagnostics utility (MDE.EXE), 229
Microsoft LAN Manager networks
  described, 384-386
  disabling expanded memory for, 385
  disabling pop-up services, 386
  driver, 143
  identified in Setup, 102
  MS-DOS LAN Manager Basic, 384
Microsoft Network
  described, 387, 452
  driver, 143
  equivalent for other networks, 143, 368
  identified in Setup, 102
  MS-Net Redirector, 247, 348
Microsoft Product Support Download Service, 439
Microsoft University
  authorized training centers, 470-471
  training programs, 468-470
MIDI, 48, 114, 143, 150, 437
Mirror utility (MS-DOS), 405
Mode command (MS-DOS), 348
Modern font, 315, 317
Mouse. *See also* specific types
  compatible with Windows 3.1, 139, 495
  defining behavior, 167, 169
  drivers, 139, 148, 186, 260
  identified in Setup, 100-101
  interrupts, 25, 208
  in non-Windows applications, 26, 190, 208, 296, 424
  serial mouse and COM port problems, 348
  in standard mode, 192
  troubleshooting, 18, 26
  virtual device, 150, 208
  in Windows Setup, 91
MOUSE.COM and MOUSE.SYS, 148
Mouse Systems mouse, 139, 150
Movable code segment (in Windows applications), 236, 452
MS-DOS
  OEM versions and general protection faults, 433
  optimizing performance, 259-260
  printing to LPT ports through MS-DOS, 345-346, 426
  troubleshooting, 35, 413-414
  unexpected MS-DOS error #11, 423
  version 5.0
    compatibility with Windows 3.1, 248, 249, 269-270
    device drivers for managing memory, 148, 230
    loading in the high memory area, 248-249, 263-264
    Windows 3.0 conflicts, 238
MS-DOS Extender. *See* Extended memory managers (XMM)
MS-DOS Mode Setup. *See also* Installing Windows 3.1
  configuration identification routines, 65-66
  described, 58, 63-68
  failure, 68
  MS-DOS version check, during Setup, 64
  modifying SYSTEM.INI for system configuration, 66-67
  setting the WINDOWS directory, 64
  troubleshooting, 401-402
  updating the Windows version, 64-65
MS-DOS Prompt, 198, 287, 373
MS-DOS system, hardware configuration, 401
MS-NET. *See* Microsoft Network
MS Sans Serif and MS Serif fonts, 112, 311, 313-314, 317
Multimedia
  capabilities, supported by Windows 3.1, 143
  CD-ROM, 21, 436
  driver files, 114, 143
  SYSTEM.INI entries, 189
  troubleshooting, 46-51, 434, 435-437
  WIN.INI entries, 177, 179
Multitasking. *See also* Timeslicing
  defined, 448, 452, 455
  demand paging and virtual memory management, 245-247
  PIF Editor options, 304
  problems running non-Windows applications, 373
  vs. task switching, in 386 enhanced mode, 290-291
  Windows *vs.* MS-DOS 5.0 task swappers, 249
Music Box, troubleshooting, 49, 436

N

Named pipes, 218, 448
NEC computer, MS-DOS version, 35
NetBIOS, 190, 208, 209, 381, 452
NetWare. *See* Novell NetWare
Network; *See also* Networks and Windows 3.1; specific network types
  adapter card, 371
  buffers, 208, 209, 218

Network, *continued*
compatible with Windows 3.1, 143, 401-402, 495
drivers, 143, 186, 209
interrupts, 202
printer redirection, 350
redirector software, and ROM BIOS data segment, 348-349
Token Ring, 214, 454
transport stacks, 371
virtual support files, 150
Network Assistant, Windows Resource Kit utility, 440
Networks and Windows 3.1. *See also* Network; specific network types
customizing Windows installation
modifying logon procedure to set paths, 369
restricting user actions, 221, 276
user environments, 374
installation information, 101-103
installing a shared copy of Windows
Administrative Setup (setup /a), 73-74
Automated Setup (setup /h), 76-81
copying Windows files to a user's directory, 130
forcing Network Setup option, 75, 130
Network Setup (setup /n), 74-75
procedures, 71-74
restoring previous connections, 177
IPX upgrade utility in Windows Resource Kit, 440
optimizing performance, 262, 370-373
printing, 31, 33
running a shared copy of Windows, 368-370, 443
shells, 431
supported networks
3COM, 102, 374-375
Artisoft LANtastic, 102-103, 376-377
Banyan VINES, 102-103, 377-378
DEC Pathworks, 102-103, 379-382
IBM OS/2 LAN Server, 102-103, 383
IBM PC LAN Program, 102-103, 383-384
Invisible Software—NET/30 For Windows, 384
Microsoft LAN Manager, 102-103, 384-386
Microsoft Network, 102-103, 215, 387
Novell NetWare, 102-104, 387-391
TCS 10Net, 391
Ungermann-Bass Net/One, 392
swapping pages to a network drive, 247
troubleshooting
386 enhanced mode, 36
operation, 430-431
printing, 31, 33, 428
Setup, 11, 402
virtual support for, 150
[Network] section in SETUP.INF, 101, 129
[Network] section in WIN.INI, 177
[Network_Specific] section in SETUP.INF, 103
[Network_Version] section in SETUP.INF, 102
[New.groups] section in SETUP.INF, 109
NewSpace, 405
Nonmaskable interrupts (NMI), 209, 408-409, 421
Nonpreemptive scheduling, 291
Non-Windows applications. *See also* APPS.INF; PIF Editor option; PIFs (program information files)
defined, 284, 452
display updates, 217
error messages
execution error, 39
file-change messages, 200
incorrect system version, 410
Non-Windows Applications, *continued*
out-of-memory errors, 43
system integrity violation, 40
files providing support in Windows, 146, 148-149, 290, 292
memory for
expanded memory, 231, 237-238, 296
extended memory, 236-237, 238, 239-240, 295
requirements, 229
mouse, 18, 26, 208, 296
removing support for, 503
running in Windows 3.1, 288-293, 373
scrolling, 213
setting up, 283-288
SYSTEM.INI entries for, 293
optimizing performance, 258-262
tips, 294-295
troubleshooting, 30, 409, 423-424
[NonWindowsApp] section in SYSTEM.INI, 189-191
Norton utilities
Stacker and troubleshooting, 17
troubleshooting, 41, 44, 404, 405, 407
Not-enough-memory error message, 294
Novell NetWare
described, 387-390, 452
drivers and files, 143, 387, 444
increasing the file access limit, 389
NetWare Lite, setting parameters, 391
preparing to run with Windows, 388
Printer Assistant, 406
SETUP.INF information, 102-103,
troubleshooting, 31, 33, 349, 431
updating with Microsoft IPX Upgrade utility, 444

O

Object Linking and Embedding (OLE)
creating compound documents, 352-356
support in windows, 354-356, 360
troubleshooting, 45
WIN.INI entries, 178, 360
Object Packager, 355
OEM (original equipment manufacturer)
character set, 310, 315, 317
defined, 452
font, 144, 145
[OEMdisks] section in SETUP.INF, 91
OEMSETUP.INF file, 335
Olivetti computers, 138, 139, 149, 187-188
Online support services
bulletin board forums, 472-474
library services, 478
PC magazine forums, 475-477
OnTrack Disk Manager (DMDRVR.BIN), 13-14
Operating modes. *See* 386 enhanced mode; Standard mode
Operating system, defined, 453. *See also* MS-DOS
Orientation, printing, 322
Out-of-memory errors, 43, 251, 289, 323, 417

P

Package, in compound documents, defined, 353, 443
Page fault, 243, 453
Page frames. *See also* Expanded memory
defined, 447, 453
memory conflicts in 386 enhanced mode, 235, 239, 240, 242
specifying range in SYSTEM.INI, 200, 212
troubleshooting, 417, 425

Pageswap device, 245
Page swapping. *See* Paging; Virtual memory
Page table, 206, 433
Paging. *See also* Virtual memory
  buffers, 210
  described 245-246, 453, 456
  drive, file, and disk space, 207, 210
  maximum size, 206
Paradise VGA cards, troubleshooting, 410-411
Parallel port. *See* LPT port
Parity, 174
Password, setting up protection for a workstation, 374
Path, 13-14, 181, 453
Pathworks. *See* DEC Pathworks
PC-Kwik, 404
.PCM file, 324
PC-NET, 453
PC-NFS and Windows 3.1, 519
PC Tools,
  Stacker and troubleshooting, 17
  troubleshooting, 17, 22, 42, 44, 404, 407
Peak/DM BIOS, troubleshooting, 12
Pen Windows, 506-514
Permanent swap file, 210, 211, 260, 262
.PFM file, 325, 326
Phoenix Cascade BIOS, 12, 419
PIF Editor options, 301-305
.PIF filename extension, 287
PIFs (program information files). *See also* APPS.INF; Non-Windows applications; PIF Editor options
  changing icons in Program Manager, 287-288
  modifying, 133, 305-306
  overview, 299-300
  setting up PIF
    after Windows installation, 286, 373
    during Windows installation, 121-125, 285-286
    from application manufacturer, 287
[PIF] section in APPS.INF, 122-125, 133
Pipe (|), in .INI parameters, 159
Plotters
  fonts, 315, 317, 321, 332
  setting up unsupported printers in Windows, 340
Plus Hardcard, troubleshooting, 24, 422
Pointing devices, 129, 139, 485. *See also* Mouse
[Pointing device] section in SETUP.INF, 100, 129
Point size, font characteristic, 310, 315, 318, 453
Pop-up messages (network), 73, 370
Ports, 174, 180. *See also* COM port; LPT port
PostScript printers
  cartridges, 336
  described, 335-336
  drivers, 141-142
  fonts, 317, 333, 335-336
  printing error information, 336
  specifying virtual memory, 334
  troubleshooting, 30, 426-427, 429
  TrueType, 333-334, 335
  unsupported printers in Windows, 340
Preemptive scheduling, 291
Printer. *See also* Printer fonts; specific printers
  compatible with Windows 3.1, 140-142, 496-498
  drivers, 27, 126-128, 140-142, 500-501
  interrupts, 213
  specifying during Setup, 80
  troubleshooting, 30-33, 427-429
  TrueType compatibility, 27, 318, 425
Printer cartridge metrics (.PCM) file, 324
Printer fonts. *See also* specific printers and printer types
  character sets for, 321
  defined, 312, 320
  dot matrix printers, 322
  HP-compatible printers, 323-333
  PostScript printers, 333-336
  printable screen fonts, 312, 320, 322, 453
Printer ports. *See also* COM port; LPT port
  port entries in the ROM BIOS data segment, 346-348
  troubleshooting, LPTx.DOS method, 345-346
  .WIN.INI entries, 180, 181
[PrinterPorts] section in WIN.INI, 180
Printer Setup, 159, 315, 321
Printing
  determining Windows printing method, 349
  network, 177, 350
  setting up unsupported printers, 340-341
  technical issues related to, 339-349
  troubleshooting, 30-33, 426-429
Print Manager, 168
[Progman.groups] section in SETUP.INF, 110, 132
PROGMAN.INI
  contents, 219-221
  modifying for custom user environments on networks, 374
  preserving, after reinstalling Windows, 15
Program information files (PIFs). *See* PIFs
Program Manager. *See also* Program Manager groups
  changing values in .INI files, 159
  customizing, 132, 272-274, 276
  initialization (.INI) file, 219-221
  maintaining Windows with, 83
  restricting user action, 221
Program Manager groups
  created during Setup, 109-111
  customized, 132, 220, 273-274
  setting up shared groups, on networks, 274-275, 374
[Programs] section in WIN.INI, 181
Protected mode, 150, 239-240, 454
Pyro!, 407

Q

QEMM–386 expanded memory manager
  386 enhanced mode support, 243, 247
  described, 454, 515-518
  networks, 373
  troubleshooting, 34, 421, 517-518
Quadtel BIOS, troubleshooting, 12

R

RAM drive,
  creating a RAM disk, 230
  during Setup, 70
  optimizing performance, 259
  swap files with, 245
  TEMP variable on diskless workstations, 262, 369
  troubleshooting, 418-419
RAMDRIVE.SYS, 148, 259, 265
Random access memory (RAM), described, 228
Raster fonts, 145, 311, 313-315, 317, 454
README files
  choosing not to install, 79
  compound documents, 353
  files for, 154, 443
Read-only files on network, 73, 369

Real mode, and Windows 3.1, 9
Rebooting
  CTRL+ALT+DEL use, 204, 256, 294-295, 432
  defining behavior, in SYSTEM.INI, 204
  from a batch file, 256,
  nonmaskable interrupts, 209
  non-Windows applications, 294-295
  Windows, after installation, 71, 77, 80-81
Reference materials, 483-486
REG.DAT file, 359, 454
REGEDIT.EXE, 360-365
Registration database, 454
  described, 359
  installing or modifying file types, 361-363
  restoring a deleted or corrupted registration database, 363
Registration Info Editor
  described, 360
  filename extensions defined, 171
  installing a . REG file, 361
  modifying .REG file types, 363
  restoring the registration database, 363
  supporting files, 151
  WIN.INI entries, 178
Reinstalling Windows, 15
Resolution
  display, 144-145
  printer, 314, 318, 322
Restricting user actions, in Program Manager, 221
[Restrictions] section in PROGMAN.INI, 221
RIPLMEM.EXE program, 379
Roman font, 315, 317
ROM BIOS
  debugging a COM port, 347-349
  INT 17, 349
  shadowing, 267
  use of upper memory area, 212-213, 216, 231, *illus.* 232
[Run] section in SETUP.INF, 90

S

Screen fonts, 144-146, 175, 312
Screen savers, 154, 168, 374
Script font, 315, 317
SCSI
  adapter card (ADAptec 1540B), 380
  drive, 24, 421, 520
SDK (Windows Software Development Kit), 454
Segmentation, for Windows code, 256, 456
Semicolon, as comment character, 159
Serial communications port. *See* COM port
Server
  database, 448
  file, 450
  network, 177, 454
  print, 453
Server applications (OLE), 353, 454-455
[Settings] section in PROGMAN.INI, 219
Setup. *See* Installing Windows 3.1; Windows Resource Kit disk setup; Windows Setup
SETUP.EXE file, and installing Windows on a local drive, 66
SETUP.INF
  checking during Setup, 64
  content summary, 88-89
  customizing, 128-133
  editing procedure, 87
  format, 88-89
SETUP.INF, *continued*
  sections
    copy-files sections for applications, 107-108
    copy-files sections for Windows, 105-106
    display driver, 94
    fonts, 112
    general installation, 89
    incompatible driver, 113
    keyboard and code pages, 97
    mouse driver, 100
    network installation, 101-103
    Program Manager groups, 109-111
    system configuration, 117
    system fonts, 104
  Windows installation, 64, 77
[Setup] section in SETUP.INF, 89
SETUP.SHH file, 76-77
Setup switches
  setup /a, 73-74
  setup /h, 75-76
  setup /i, 65, 401
  setup /n, 11, 74-75
  setup /t, 400
  summary, 60
Shadow RAM, 250
Shadow ROM, 420, 430, 455
Share command (MS-DOS), 368
Shared memory, troubleshooting, 430
Shared resources, 304, 455
Shell command (MS-DOS), 260
[Shell] section in SETUP.INF, 94
Shortcut keys, defined in PIFs, 302-303
SideKick, 407
Small Font, 112, 310, 314, 317
SMARTDrive 3.x, *illus.* 252
SMARTDrive 4.0
  described, 252-256, 439
  double buffering, *illus.* 253, 255-256
  installation of, 70
  optimizing Windows performance, 230, 259
  Stacker, 17, 256
  troubleshooting, 416, 421-422
  viewing switches, 258
SMARTDrive Monitor, Windows Resource Kit utility, 441
SMARTDRV.EXE, 148
SoftCraft WYSIFonts, 315
Soft fonts. *See* Downloadable soft fonts
SoftIce Debugger, 404
Sound
  drivers on Windows Resource Kit, 443
  files, 154
  troubleshooting, 48-51
  WIN.INI entries, 179
SoundBlaster, files identified in Setup, 114
Sound Recorder, troubleshooting, 51, 436
[Sound] section in WIN.INI, 179
Source document, for OLE, 355
[Special_adapter] section in SETUP.INF, 119
Speedfxr, 407
Speedstor, troubleshooting, 13-14
Spooler, 168, 455
SQL (Structured Query Language), 455
Stacker, 13-14, 17, 66, 244, 256
Stack overflow error, 422
Stacks, in CONFIG.SYS, 422

Standard mode Windows. *See also* 386 enhanced mode Windows; Windows 3.1
described, 63, 235
memory
conventional memory, 235
expanded memory, 236-238
extended memory, 235-237
system requirements, 9, 63, 410
optimizing performance, 260
removing support for, 502
troubleshooting, 44, 418-419, 423
[Standard] section in SYSTEM.INI, 191-192
Startup directory, specifying in PIF Editor, 301
Startup group, 58, 220
Startup requirements, for Windows, 249-250
Stop-bits, 174
Subst command (MS-DOS), 390, 407
SuperPrint, troubleshooting, 32
SuperVGA, troubleshooting, 20, 411, 421
Swap disk, for non-Windows applications, 191
Swapdisk memory manager, 230
SWAPFILE.EXE, file deleted in Setup, 107
Swap files
changing size and location of, 244
creating, for virtual memory, 238, 244
non-Windows applications using, 290, 521-522
permanent, 210, 211, 244-245, 260, 262
temporary, 207, 210, 244
Virtual Memory Manager, 243-244
Swappable code segment (in Windows applications), 236
Swapsize, defined, 236
Switches, command-line
Mem command (MS-DOS), 229-230
MS-DOS system drivers, 258
PIF Editor options, 301
Setup, 60
Windows, 249
Symbol font, 112, 314, 317, 318, 333
System
configuration specified in Setup, 117-120, 401-402
modifying settings during Setup, 65-66
requirements, 9, 418, 420
System Editor, 160-161
System events, WIN.INI entries, 179
System font, 104, 144-145, 312-315
SYSTEM.INI
contents, 183-218
content summary, 183
entries for
386 enhanced mode performance, 271
conventional memory, 238, 250
disabling expanded memory, 241
hardware configurations, 66-67, 271
network performance, 272, 370-371
standard mode performance, 238
virtual memory, 246
preserving entries, after reinstalling Windows, 15
source file (SYSTEM.SRC), 66, 161
updates defined in Setup, 116
Windows 3.1 changes, 162-163
Windows installation changes, 66, 77
System integrity violation, 40
System Information screen, *illus.* 66, 77
System memory, displaying current use, 442
System Resource Monitor, Windows Resource Kit utility, 442
System resources, 250-251, 442
System screen fonts. *See* System font
[System] section in SETUP.INF, 117-120, 129
System settings script, 75-81
SYSTEM.SRC, 77, 161
SYSTEM subdirectory
defined, 58
files defined for Setup, 93, 127

T

Tandon, troubleshooting, 12
Task Manager
defined, 290, 373
specifying, 186
SYSTEM.INI entry, 186
vs. Windows Task List, 248, 249
WIN.INI entry, 164
TCS 10Net, 102, 391
TEMP directory, 369
TEMP environmental variable
"clean boot" disk, 398
optimizing performance, 262-263
printing requirements, 427
setting RAM drive as, 259
troubleshooting, 13-14, 427
Windows on a network, 369, 427
Temporary swap file
described, 243-244
optimizing performance on a network, 260, 262
SYSTEM.INI entries, 207, 210
Terminal font, 144, 313, 317
Texas Instruments Graphics Architecture (TIGA) display adapter, 412
Third-party
memory managers, 269-270
soft fonts, 316
task manager, 272
Thunder Board (Media Vision), files identified in Setup, 114
TIGA display driver, 139, 150, 412
Time
defining formats, 172
interrupts, 25, 214, 215, 373
synchronizing with CMOS clock, 213
Time-out settings, for printer ports, 180
Timer (MCI driver), 114, 143
Timer interrupts, 214, 215, 373
Timeslicing
between Windows and non-Windows applications, 291-292
defined, 455
keyboard, 422
PIF Editor options, 304
virtual machines, 207
Windows applications, 217-218
Times New Roman font, 113, 310-311, 317-318, 320
Tms Rmm font, 107, 314
Token Ring networks, 214
TopDesk, Windows Resource Kit utility, 442
Toshiba computer, 12, 412, 419
Training, 466-467, 468-471. *See also* Microsoft Authorized Training Centers (ATCs); Microsoft Certified Professional Program; Microsoft University
[Translate] section in SETUP.INF, 115
Translation buffers, 239
controlling mapping in upper memory, 239, 241, 242
freeing conventional memory, 268-269
Transmission retries, 169

Transport stack, 371
Trantor (TSCSI.SYS), 407
Troubleshooting
  386 enhanced mode, 420-422
  creating a "clean boot" disk, 398
  desktop configuration, 403
  display problems, 408
  EMS memory problems, 235, 237, 415-417
  fonts, 425
  general protection faults, 431-433
  MS-DOS configuration problems, 413-414
  multimedia, 434
  networks, 430-431
  non-Windows applications, 423-425
  printing, 426-429
  Setup, 399-403
  standard mode, 418-419
  strategy, 396-398
  TSRs, 400, 404-407
  UMB conflicts, 237
TrueType
  benefits, 318
  Canon printers, 337
  changing typeface names, 337-338
  described, 311, 318-320
  disk space, memory, and speed, 319-320
  files, 113, 146, 318
  HPPCL printers, 323
  IBM 4019 laser printers, 337
  installing more fonts, 315
  number of fonts on a system, 320
  PostScript printers, 333-335
  printer support, 316, 317, 319
  troubleshooting, 27, 425-426
  using in Windows applications, 319, 320
  WIN.INI entries, 176
[TrueType] section in WIN.INI, 176
Tseng Labs display adapter, 410
TSRs (terminate-and-stay resident software)
  described, 239, 294, 455
  optimizing performance, 259, 261
  identified in Setup, 113, 401, 404
  SYSTEM.INI entries, 202, 211, 214
  troubleshooting
    general protection faults, 37, 432
    in Setup, 10, 13-14, 66, 400-401, 404
    in Windows, 404-407, 414
.TTF file, 318
Typeface, defined, 310

U

Underscore (_), as indicator of compressed files, 59
Ungermann-Bass
  network card, 392
  XNS transport stack, 384
Universal printer driver, 318, 321-322, 330
Unrecoverable application error (UAE), local reboot, 204
[Update.files] section in SETUP.INF, 115
Upper memory area. *See also* Upper memory blocks (UMBs)
  conflicting uses of, in 386 enhanced mode, 242
  described, 228-233, 237-238
  illus., 229, 232, 233
  placing translation buffers in, 241
  troubleshooting, 414, 415-417
Upper memory blocks (UMBs)
  controlling, in 386 enhanced mode, 242
  defined, 237, 455
  dual display adapters, 198
  excluding ranges, 199-200, 204, 212, 215, 416-417
  illus., 415
  mapping, 242
  optimizing use of, 263-266, 266-268
  troubleshooting for
    386 enhanced mode, 420-421
    general protection faults, 433
    networks, 430
    non-Windows applications, 425
USER
  .EXE file, 137, 250
  heap, 250-251, 442
User groups, listed by location, 479-481
User interface, defined, 450
User name, specifying in Windows Setup, 77, 79
[User] section in SETUP.INF, 92
.USP file, 325

V

Vaccine, 404
VAR (value-added reseller), defined, 456
VCPI (Virtual Control Program Interface), 237, 247, 295, 456
VDefend, 404
VDNET.386 virtual device driver, 381
Vector fonts, 145, 311, 315, 317, 456
VGA
  driver and grabber files, 139, 149
  font files, 104-106, 312-314
  memory addresses, 216
  troubleshooting, 20, 26, 409-412, 424
  virtual support file, 150
Video memory
  SYSTEM.INI entries, 212, 216
  troubleshooting, 424, 425
Video Seven VGA
  driver and grabber files, 139, 149
  troubleshooting, 403, 410, 412
  virtual support file, 150
Viralert, 404
Virex-PC, 404
Virtual COM Driver, 196
Virtual device, 150, 456
Virtual Display Driver (VDD), 94, 102, 408, 410
Virtual DMA Services, 255-256
Virtual Hard Disk (VHD), 34, 422
Virtual Keyboard Device (VKD), 204
Virtual machine (VM)
  addresses for applications, 215, 453
  COM port interrupt processing, 196
  described, 239
  device drivers in, 204
  idle interrupts, 201
  memory
    allocation, 211, 213-214
    expanded memory, 213-214
    extended memory, 213-214, 239
  printer interrupts, 213
  read-write requests, 209
  SYSTEM.INI entries, 193, 196, 205, 207, 213-214
  timeslicing, 207, 291

Virtual memory
  32-bit disk access, 520-523
  386 enhanced mode, 243-247
  creating swap files for, 244-245
  defined, 453, 456
  demand paging, 245-247, 448
  management of, 243, 245-247
  optimizing, 245, 520-523
  pages, page swapping, and page table, 230, 245, 245-246
  PostScript printer requirements, 334
  swapping pages to network drive not recommended, 247
Virtual Memory Manager (VMM), 206, 230, 243, 245-246
Virtual Mouse Device (VMD), 100
ViruSafe, 404
VMM. *See* Virtual Memory Manager (VMM)
VT320 terminal emulator application, 380

W

Wallpaper, 153, 154, 170
Wave audio, 114, 154, 437
Western Digital
  32-bit disk access and 1003 controller standard, 520, 523
  Ethercard Plus, 388
  troubleshooting, 11, 34, 36
WIN386.EXE file, 136, 150, 245
WIN.COM, 11, 68, 136-137, 152
[Win.*] copy-file sections in SETUP.INF, 108-109, 130-131
Win command switches, 249
Window (on screen)
  running non-Windows applications in a, 292-293
  WIN.INI entries, 164-169, 178-179
Windows 2.x, troubleshooting, 432
Windows 3.1. *See also* 386 enhanced mode Windows; Standard mode Windows; Windows applications; Windows Setup
  compatibility with MS-DOS 5.0, 269-270
  configuration, troubleshooting, 403-417
  foreign language versions, 458
  maintaining Windows
    on a network, 369-370
    using Setup, 82
  minimum installation, 504
  MS-DOS 5.0 configuration for, 248-249, 261
  new features in
    32-bit disk access, 520-523
    Character Map, 321
    File Manager features, 277
    icons, 288
    multimedia, 135
    network features, 368
    Object Linking and Embedding, 352
    PIFs, 285
    printing, 278
    Program Manager features, 273
    Setup, 58
    SMARTDrive 4.0, 252-256
    swap file features, 244
    system registration database, 359-363
    TrueType fonts, 311, 318-320
    universal printer driver, 318, 321-322
    upper memory management, 243
Windwos 3.1, *continued*
  optimizing performance
    adjusting UMBs, 263-266, 266-269
    hardware, 258-260
    MS-DOS, 259-260
    network, 262
    Windows configuration, 260
Windows applications
  custom installation, 130
  defined, 284, 456
  files, 151-152
  removing, 500
  troubleshooting, 43
WINDOWS directory
  defined, 58
  files copied to by Setup, 92, 130
  installing Windows in, during Setup, 77-78
  personal *vs.* shared, 368
  setting during installation, 64
Windows Driver Library, 478
Windows files
  core files, 136, 137
  deleting, 155, 500-504
  driver files, 136-143
  font files, 144-146
  MS-DOS support files, 148-150
  overview, 135-136
  Windows applications and related files, 151-155
Windows for Pens, 506-514
Windows help files, 152, 503
[Windows Help] section in WIN.INI, 178-179
Windows Mode Setup
  checking for SMARTDrive, 70
  completing Setup, 71
  custom application installation, 69-70
  first stages of, 58, 68
  system files update, 70
  troubleshooting, 402-403
  updating from earlier Windows version, 71
Windows Resource Directory
  hardware support, 482-483
  online services, 472-478
  reference materials, 483-486
  training services, 458-471
  user groups, 479-481
Windows Resource Kit disk
  downloading, 439
  installing a sound driver file, 444
  Microsoft IPX upgrade utility, 444
  README files, 443
  setup, 444
  utilities described, 440-442
  Windows Driver Library, 443, 444
[Windows] section in SETUP.INF, 93
[Windows] section in WIN.INI, 164-169

Windows Setup. *See also* Installing Windows 3.1
  Administrative Setup option (setup /a), 72-74
  Automated Setup (setup /h), 58, 75-81
  Custom Setup, 58, 64-65
  Express Setup, 58, 64
  MS-DOS Mode Setup, 58, 63-68, 401-402
  Network Setup (setup /n), 73-75
  overview, 58-63
  setup-related files, 152-153
  shell, 94
  switches for Windows Setup command, 60
  technical discussion, 62-71
  troubleshooting, 9-17, 399-403
  Windows Mode Setup, 58, 68-71, 402-403
Windows shell, 272-276, 277-279
[Windows.system *] sections in SETUP.INF, 93
Windows Technical Support Library, 478
[Winexec] section in SETUP.INF, 90
WINFILE.INI, 223
Wingding (TrueType font), 146
WIN.INI
  changes in the system registration database, 354
  content, 163-182
  content summary, 163
  preserving, after reinstalling Windows, 15
  size, 163
  source file (WIN.SRC), 16, 77, 161
  substituting screen fonts in, 335
  Windows 3.1 changes, 116, 161-162
  Windows installation changes, 77
WinOldApp file, 148, 290, 292, 423
WIN.SRC, 16, 77, 161
WINSTART.BAT file, 261
Workstations, network, 71,75, 456
.WPD (Windows PostScript Definition) file, 335
Wyse computer, 12, 419

X

X.25 network communications protocol, defined, 456
XGA display driver, 139, 150, 380
XGAAIDOS.SYS, 406
XMM. *See* Extended memory managers (XMM)
XMS. *See* Extended Memory Specification
XMS memory. *See* Extended memory

Z

Zenith, troubleshooting, 12, 35

Windows Resource Kit Feedback Sheet

We update the *Windows Resource Kit* as we release new versions of Windows, and we'd like you to tell us the type of additional information you'd like to see in the Kit. If there are topics you want to be covered in an update, please fill out this sheet and send it to the address at the bottom of the page. Your input will be valuable when we prepare future information for the Kit.

Your Feedback

My priorities for more information are (number these in order of importance):

__ Installation and Setup

__ Memory Management

__ Non-Windows Applications and PIFs

__ Fonts, Printing, and Compound Documents

__ Networks

__ Troubleshooting

__ Other:_____

I found the following information in this Kit the most valuable: _____

_____

_____

I would like access to more utilities, such as:_____

_____

_____

I would find the information in this Kit more helpful if:_____

_____

_____

Please send your completed feedback sheet to:

Microsoft
Attn: *Windows 3.1 Resource Kit*
One Microsoft Way
Redmond, Washington 98052-6399

Windows Resource Directory Updates

If your organization is listed in the Windows Resource Directory and its address has changed, or if you want to be added to the Directory, please fill out this form and return it to us.

Change of address

Organization: _____

New Address: _____

Contact: _____

Phone / Fax: _____

Please add my organization to the Windows Resource Directory

Organization: _____

New Address: _____

Contact: _____

Phone / Fax: _____

I am offering the following new information:

(List courses, books, seminars, services, etc.)

_____

_____

_____

_____

_____

Please send this form to:

Windows Resource Directory Updates, 3/2
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Or fax it to us as 206/883-8101

The Windows Resource Kit

For Microsoft® Windows™ Operating System Version 3.1

This exclusive Microsoft collection provides complete technical information on Windows version 3.1 for support professionals. It contains both a technical reference manual and a disk full of helpful utilities, system resource viewers, drivers, and accessories. Its purpose is to provide you with the information you'll need to support installations of Windows in your company.

Quick Access

The Windows Resource Kit reference manual gives you many ways to access valuable technical information:

- An in-depth table of contents to steer you to the information you need
- Cross-referenced charts of structured troubleshooting methods for more than 40 of the most common issues you may encounter while running Windows version 3.1
- Sections of the manual cross-referenced in the Troubleshooting Charts are flagged for easy identification
- Icons in the manual call out new aspects of Windows version 3.1
- In-depth, fully cross-referenced index

Comprehensive

The Windows Resource Kit technical reference manual is the most comprehensive technical information source for Windows version 3.1. It contains sections on installation and setup, configuring Windows version 3.1, working with non-Windows–based applications and PIFs, and on using Windows version 3.1 (including chapters on fonts and printing). You'll also find sections on networks, more information on troubleshooting, a complete description of the utilities and drivers included in the Kit, a glossary, an updated Hardware Compatibility List, and articles on pen computing and multimedia.

The Microsoft Commitment to Worldwide Technical Support

With nearly 200,000 copies in circulation, the Windows Resource Kit is only one of the ways that Microsoft offers technical support to customers worldwide. The Kit is part of our Premier worldwide technical support program, which includes:

- The Windows Resource Kit—distributed worldwide with planned translations for French, German, Italian, Spanish, and Swedish
- Worldwide Windows Technical Workshop tours in Argentina, Australia, Brazil, Canada, Denmark, England, France, Germany, Mexico, Norway, Spain, Sweden, and Venezuela. These workshops have been attended by tens of thousands of Microsoft customers.

We also offer training through Microsoft University, both at company sites and at 10 locations across the United States. In addition, our Reseller Training Alliance (RTA) program (with its coalition of resellers across the U.S.) gives you access to support for Windows at a location near you.

0292 Part No. 31645

Microsoft®