CD INCLUDED

Microsoft® Windows NT™ Resource Kit

Essential, new information exclusively for owners of the Microsoft Windows NT Resource Kit, Version 3.51

VERSION 3.51 UPDATE 2

For Windows NT Workstation and Windows NT Server Version 3.51

Microsoft® Press

Microsoft® Windows NT™ Resource Kit

Version 3.51 Update 2

For Windows NT Workstation and Windows NT Server Version 3.51

Microsoft® Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data pending.

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QMQM 1 0 9 8 7 6

Distributed to the book trade in Canada by Macmillan of Canada, a division of Canada Publishing Corporation.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office. Or contact Microsoft Press International directly at fax (206) 936-7329.

Acquisitions Editor: Casey Doyle
Project Editor: Wallis Bolz

This book is dedicated to the many users of the Windows NT operating system.
May they work in peace!

Contributors to this book include the following:

Technical Writers:
Janice Breyer, Peter Costantini, Peggy Etchevers, Gert Gustedt, Sharon Kay, Annie Pearson, Cary Reinstein, Laura Sheppard, Maureen Sullivan

Technical Consultants:
Brian Andrew, Eugene Baucom, Brian Bosserman, Phillip Carver, Tom Donnelly, James Gilroy, Gert Gustedt, Bryna Hebert, Eric Hough, John Jacobs, Gary Kimura, Azfar Moazzam, Toby Nixon, Jonathan Perera, Tom Prisk, Zephan Schroeder, Mark Sestak,Tom Vukovic, Sean Wheeler, and numerous other hardworking Windows NT Developers, Program Managers, and Product Support Specialists

Technical Editors:
Pamela Miller, Sharon Tighe

Managing Editor:
Sonia Marie Moore

Software Program Managers:
Louis Kahn, Ryan Marshall

Documentation Project Manager:
Peggy Etchevers

Indexer:
Barbara Sherman

Production Team:
Cheryl Capriola, Karye Cattrell, Cathy Pfarr, Keri Segna, Jeff Weaver

Graphic Designers:
Kathleen Evanoff, Sue Wyble

Contents

Figures and Tables

Figures

Tables

Tables *(continued)*

Introduction

Welcome to the *Microsoft® Windows NT™ Resource Kit Volume 6: Windows NT Update 2* book.

The Microsoft *Windows NT Resource Kit for Windows NT Workstation and Windows NT Server version 3.51* consists of the four volumes that were shipped with the version 3.5 release, a new Volume 5, this additional Volume 6, and a single compact disc (CD) containing new utilities and updated versions of the existing ones. Floppy disks are no longer available.

The *Windows NT Update 2* book presents detailed information on topics that are either new for version 3.51 or reflect issues that our Product Support people consider timely and important. The appendixes also include information on changes that have been made to the four-volume set of books for version 3.5 and Volume 5 for version 3.51. When looking for the latest information on a topic, start with the *Windows NT Update 2* book, and then work your way back through the other books in the 6-volume set.

The information provided in this volume is a technical supplement to the documentation included as part of the Windows NT Workstation and Windows NT Server version 3.51 product. It does not replace that information as the source for learning how to use the product features and utilities.

This introduction includes the following types of information you can use to get started:

- The first section outlines the contents of this book, so that you can quickly find pertinent technical details.
- The second section introduces the *Windows NT Resource Kit* CD.
- The third section describes the support policy for the *Windows NT Resource Kit.*
- The fourth section describes the conventions used to present information in this book.

About the Update 2 Book

This book includes the following chapters. Additional tables of contents are included in each part to help you quickly find the information you want.

Part I, Windows NT Workstation Deployment Planning Basics - Beta Draft

Chapter 1, "Overview of the Process," provides an overview of the major steps involved in the deployment process.

Chapter 2, "Deployment Strategy and Details," contains the details about how to make decisions and perform the actions listed in the overview.

Chapter 3, "Lab Tests," describes the steps you would take to set up a lab and then test the deployment process in that lab. This includes most of the steps in the deployment process.

Chapter 4, "Pilot Rollout," discusses the additional steps, such as user training, involved in planning and performing a pilot rollout.

Chapter 5, "Final Rollout," discusses the final rollout of Windows NT Workstation.

Chapter 6, "Unattended Installations and Upgrades," discusses ways to automate the deployment process. Using an automated deployment process can produce significant time and money savings, as well as provide better control of the process, if you are installing on more than a few computers.

Part II, Using Windows NT

Chapter 7, "Windows NT File System," provides a comparison of the Windows NT File System (NTFS) and the file allocation table (FAT) file system to help you determine whether you want to use one or both of these file systems on your computer. It also includes a description of NTFS compression, a comparison to other compression techniques, and information on how to move and copy compressed files.

Chapter 8, "Fault Tolerance for Disks," describes fault tolerance and the steps for creating a fault-tolerant disk configuration, preparing for recovery, recovering disk information after a hardware failure, and rebuilding fault-tolerant disk sets after a hardware failure.

Chapter 9, "Printing," describes the printing process when using either Windows NT Workstation or Windows NT Server. The information on printing that was previously published in the *Windows NT Resource Guide* has been reorganized and updated. Additional information about common problems that users might encounter and how to troubleshoot printing problems is also provided.

Part III, Appendixes

Appendix A, "Major Revisions to Windows NT Update 1," includes two major sections: first, a complete replacement of Appendix B, "Major Revision to Windows NT Messages," and of the "Windows NT Debugger" section of Chapter 2, "Windows NT Executive Messages," in *Windows NT Messages*; second, a replacement of the section titled "LAN Manager MIB II for Windows NT Objects" in Appendix A, "Major Revisions to the Windows NT Networking Guide," in *Windows NT Update 1*. These major changes will not be incorporated into any reprinted edition of Volume 5, *Windows NT Update 1*.

Appendix B, "Minor Revisions to Existing Resource Kit Books," includes a list (organized by book, chapter, and page number) of all the changes that have been incorporated into the revised editions of the five-volume set of resource kit books for version 3.51.

Appendix C, "RAS Reference," provides an overview of the most important modem compatibility standards and how they work within the Remote Access Service (RAS), a series of quick-reference charts to give you a high-level perspective on how RAS works during a call to a Windows NT RAS server, and reference tables for RAS server and client computers that detail the different versions of RAS and the features they support.

Appendix D, "RFCs and Port Reference for MS TCP/IP," provides lists of Request for Comments (RFC) and port assignments in the Microsoft implementation of the UDP and TCP/IP protocol suite, as currently known. This information is intended for field representatives, system engineers, and system integrators.

Index to this *Windows NT Update 2* book.

Resource Kit Compact Disc

The CD that accompanies the *Windows NT Resource Kit* contains utilities that apply to information in the *Windows NT Resource Guide,* the *Windows NT Networking Guide*, and the *Windows NT Update 1* and *Windows NT Update 2* books. This CD includes a collection of information resources, tools, and utilities that can make networking and working with the Windows NT platform even easier. The Windows NT Messages database and the utilities that apply to information in the *Optimizing Windows NT* book are also included on the *Windows NT Resource Kit* CD. This new CD replaces the previous ones.

After installing the *Windows NT Resource Kit*, please refer first to the following two files:

- The README.WRI file, which contains a complete list of all the tools and utilities on the *Windows NT Resource Kit* CD and additonal setup instructions for some of them.
- The RKTOOLS.HLP file, which provides basic instructions on how to use all of the tools and utilities, along with links to additional documentation and, in some cases, to the actual program files.

Patches with the most current corrections to those tools and utilities and their documentation, as well as the POSIX and Perl public domain source code files, are available on the Internet at the following Microsoft FTP site:

ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/nt351/.

The *Windows NT Resource Kit* CD includes a wide variety of tools and utilities to help you work more efficiently with Windows NT Workstation and Windows NT Server. Many of the items that were included in version 3.5 are described in the "Introduction" section of the *Windows NT Resource Guide.* A few of the new or significantly updated items for version 3.51 were described in *Windows NT Update 1.* The following notes describe some of the enhancements made to the existing tools and utilities and introduce new ones that have been added for this second version 3.51 release.

Computer Diagnostic Tools

- Crystal Reports for the *Windows NT Resource Kit* provides an easy way to extract and publish information found in the System, Application, and Security event logs. Specific enhancements have been made for this release to enable more flexibility in filtering and analyzing data contained in the Security event log. Crystal Reports includes 12 useable reports that quickly provide information, in a single report, from either one or multiple computers,.

Desktop Tools

- TextViewer, TEXTVIEW.EXE, provides a graphical interface for quickly viewing similar text files within multiple subdirectories on local or shared drives. For example, you can use it to skim through all the .TXT files on the *Windows NT Resource Kit* CD or all the source files (*.C, *.CPP, *.CXX) on the CD for the Win32 SDK. It also provides basic editing and searching capabilities.

File Tools

- The FTEdit tool, FTEDIT.EXE, can help you in the recovery of fault-tolerant volumes. FTEdit can rebuild any kind of fault-tolerant set, including stripe sets, volume sets, stripe sets with parity, and mirror sets.

Internet and TCP/IP Services/Tools

- Mail Server is a Simple Mail Transport Protocol (SMTP) and Post Office Protocol (POP) server for the Windows NT platform. The intermediate files and mailboxes are all spooled securely (when using the NTFS file system) on the computer running Windows NT Server and are accessed through any POP-compliant public domain (PD) or commercial client. This version contains several bug fixes and usability enhancements.

Network Diagnostic Tools

- SNMPMon is a standalone executable utility that accepts a configuration file as input. This utility monitors multiple SNMP-enabled nodes, logs query results to ODBC-enabled data sources, and runs command-line programs when specified thresholds are crossed. You can enable logging for all queries or limit it to particular thresholds, which can be either edge or level triggered.

Registry Tools

- The Registry Entries Help file, REGENTRY.HLP, has been updated again for this newest version. The corresponding chapter in the *Windows NT Resource Guide* has not been updated.

Server/Network Administration Tools

- Two remote-computing utilities are provided. REMOTE enables a user to log on to a remote computer within the security context of the user who started the service. All commands are then executed on the remote computer as if the user who started the service were logged on locally. RCMD provides a method for users to execute commands on a remote computer within their own security context.

Resource Kit Support Policy

The SOFTWARE supplied in the *Windows NT Resource Kit* is not officially supported. Microsoft does not guarantee the performance of the *Window NT Resource Kit* tools, response times for answering questions, or bug fixes to the tools. However, we do provide a way for customers who purchase the *Windows NT Resource Kit* to report bugs and receive possible fixes for their issues. You can do this by either sending Internet e-mail to RKINPUT@MICROSOFT.COM or by referring to one of the options listed in the "Your Guide to Service and Support" pamphlet, which is included with your Windows NT product. This e-mail address is only for *Windows NT Resource Kit*-related issues.

The SOFTWARE (including instructions for its use and all printed and online documentation) is provided "AS IS" without warranty of any kind. Microsoft further disclaims all implied warranties, including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance of the SOFTWARE and documentation remains with you.

In no event shall Microsoft, its authors, or anyone else involved in the creation, production, or delivery of the SOFTWARE 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 of or inability to use the SOFTWARE or documentation, even if Microsoft has been advised of the possibility of such damages.

Conventions in This Manual

This document assumes that you have read the Windows NT Workstation and/or Windows NT Server version 3.51 documentation sets and that you are familiar with using menus, dialog boxes, and other features of the Windows operating system family of products. It also assumes that you have installed Windows NT Workstation or Windows NT Server version 3.51 on your system and that you are using a mouse. For keyboard equivalents to menu and mouse actions, see Microsoft Windows NT Help.

This document uses several conventions to help you identify information. The following table describes the typographical conventions used in the *Windows NT Update 2* book.

Convention	Used for
bold	MS-DOS–style command and utility names such as **copy** or **ping,** and switches such as **/?** or **-h.** Also used for Registry value names, such as **IniFileMapping,** and OS/2 application programming interfaces (API).
italic	Parameters for which you can supply specific values. For example, the Windows NT root directory appears in a path name as *systemroot*\SYSTEM32, where *systemroot* can be C:\WINNT35 or some other value.
ALL CAPITALS	Directory names, filenames, and acronyms. For example, DLC stands for Data Link Control; C:\PAGEFILE.SYS is a file in the boot sector.
Monospace	Sample text from batch and .INI files, Registry paths, and screen text in non-Windows–based applications.

Other conventions in this document include the following:

- "MS-DOS" refers to Microsoft MS-DOS version 3.3 or later.
- "Windows-based application" is used as a shorthand term to refer to an application that is designed to run with 16-bit Windows and does not run without Windows. All 16-bit and 32-bit Windows applications follow similar conventions for the arrangement of menus, dialog box styles, and keyboard and mouse use.

- "MS-DOS–based application" is used as a shorthand term to refer to an application that is designed to run with MS-DOS, but not specifically with Windows or Windows NT, and is not able to take full advantage of their graphical or memory management features.
- "Command prompt" refers to the command line where you type MS-DOS–style commands. Typically, you see characters such as C:\> to show the location of the command prompt on your screen. In Windows NT Workstation and Windows NT Server, 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, and 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 NT System Guide* and the Windows online tutorial.

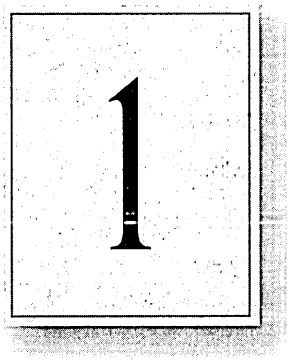
PART I

Windows NT Workstation Deployment Planning Basics

Part One provides an overview of a recommended process that administrators can use to plan and implement a major deployment in their organization of computers running Windows NT Workstation version 3.51. These chapters are adapted from the Beta version of the Windows NT Workstation version 4.0 *Deployment Planning Guide*, which will be available in 1996 following the release of Windows NT Workstation version 4.0.

CHAPTER 1

Overview of the Process



This chapter is for administrators who are responsible for corporate implementation of Windows NT Workstation. It provides an overview of the major steps in the deployment process. Chapter 2, "Deployment Strategy and Details," contains the details about how to make decisions and perform actions listed in the overview. Some of the tasks described in these chapters might not be necessary for your organization.

For step-by-step instructions on installation, see the *Windows NT Workstation Installation Guide.*

The deployment process for Windows NT Workstation consists of several distinct phases, including the following:

- Reviewing Windows NT Workstation
- Preparing the Planning and Support teams
- Identifying the preferred network-client configuration
- Performing lab tests of the client configuration
- Planning the pilot rollout
- Conducting the pilot rollout
- Finalizing the rollout plan
- Rolling out Windows NT Workstation

This chapter contains a section that outlines, in checklist form, the required tasks for each deployment phase.

The following sample shows how to read a deployment checklist for any phase.

Table 1.1 Description of the Deployment Phase

Task	Team	Start week	Duration
1. Summary of the task.	Who will perform this task?	When does the team begin this task?	How long will it take to complete?

The following teams, made up of employees from your organization, are responsible for performing the tasks described in deployment checklists:

- The Executive team includes the deployment project manager (usually the head of the Information Systems department) and members of the executive committee of the corporation. This team must include one or more individuals with decision-making authority over company policies and procedures.
- The Planning team includes the deployment project manager, key Installation team members, and a representative from the Support and Training teams.
- The Installation team includes technicians and individuals who will be conducting the installation. This team must include a specialist in 32-bit applications who can evaluate the proposed Windows NT Workstation 4.0 configuration for compatibility.
- The Support team includes staff of the help desk or Support department, and select individuals from the Planning team. This team develops a plan for supporting Windows NT Workstation during and after deployment, integrating new methods and processes into the existing support scheme as needed.
- The Training team includes individuals responsible for user training.

At certain phases, you might choose to vary the makeup of the teams by adding or omitting individuals.

Reviewing Windows NT Workstation Features

When implemented, Windows NT Workstation can yield significant benefits to your organization in terms of reduced costs and increased system control. Because many decisions—starting with the decision to acquire Windows NT Workstation—depend on these and other anticipated benefits, becoming familiar with the features and benefits of Windows NT Workstation is the first step in deployment planning.

The following checklist provides sources of information on Windows NT Workstation features and benefits.

Table 1.2 Reviewing Windows NT Workstation Features

Task	Team	Start week	Duration
1. Acquire additional copies of the *Windows NT 3.51 Resource Kit* for review during the deployment process.	Planning	Week 2	1 day

Preparing the Teams

After review of Windows NT Workstation features and benefits, the next step is to prepare the Planning, Installation, and Support teams. If you did not fully staff the Planning team for the review phase, assemble the people you need for the Planning and Installation teams at this time. Then gather the equipment and tools to be used in planning the Windows NT Workstation implementation, and arrange for Support team training. The following checklist outlines the processes of assembling the Planning and Installation teams and their resources and coordinating Support team training.

Table 1.3 Preparing the Teams

Task	Team	Start week	Duration
1. Assign the project manager, if appropriate (usually this is the head of the Information Systems department).	Planning	Week 2	—
2. Select key Planning and Installation team members, if appropriate. Make sure to include an applications specialist, for evaluating 32-bit applications.	Planning, Installation	Week 2	5 days
3. Acquire Windows NT Workstation.	Planning	Week 2	1 day
4. Plan your client and server hardware and software configurations on the network.	Planning	Week 3	5 days
5. Set up a testing lab.	Planning	Week 2	1 day
6. Acquire test computers for use as the network server and clients. Choose computer models that are typical of those used in your organization.	Planning	Week 2	5 days

Table 1.3 Preparing the Teams *(continued)*

Task	Team	Start week	Duration
7. Install the applications software and line-of-business tools in the lab to simulate the network environment. Also identify the mission-critical and noncritical business and other applications typically used in your organization. Create a checklist for evaluating the compatibility and performance of these applications during testing.	Planning	Week 3	3 days
8. Review detailed discussions of product features in the *Windows NT 3.51 Resource Kit* to prepare for configuration planning.	Planning, Installation	Week 3	3 days
9. Study the entire *Windows NT 3.51 Resource Kit*. As an option, arrange for the team and other individuals, as appropriate, to attend training at a Microsoft Authorized Technical Education Center and participate in the Microsoft Certified Professional program to prepare for supporting Windows NT Workstation.	Support	Week 3	10 days

Deciding on the Preferred Client Configuration

With the Planning and Installation teams assembled and educated about Windows NT Workstation capabilities, the next task for these teams is to determine the preferred configuration for client computers on the network. (For the purposes of this discussion, "client computer" refers to any computer running Windows NT Workstation, including computers that act as peer servers by running File and Printer Sharing services.) The teams will use this configuration for evaluation and testing, before full implementation of Windows NT Workstation in your organization.

The tables in this section summarize options to consider in planning your preferred configuration. Using the information in these tables, evaluate the available features and the related alternatives before making a decision. Microsoft recommends that you begin your evaluation with the ideal configuration, that is, a configuration that uses all of the most powerful features of Windows NT Workstation. Then gradually modify this configuration, adding or removing features, until you achieve a configuration that more closely fits your company's needs. When you have identified the preferred configuration, document the configuration layout and the selected features to make sure you install and test the correct configuration.

The following table presents an overview of configuration layout decisions and feature options for the ideal network client. An additional table lists features that Microsoft recommends for implementation by all organizations; these features define how Windows NT Workstation will be installed and administered in your organization. The final table shows optional features that might be useful in some organizations.

Table 1.4 Configuration Layout Decisions

Configuration option	Decisions and issues
Location of application files *To maximize performance or hard disk space on the client computer*	Depends on your need to maximize central administration versus performance on the client computer. Also depends on the hardware platform of the client computer. Keep in mind that most applications are not designed as true client-server applications and thus might not perform well over the network. However, to save disk space it might make sense to have the least used applications run from the server. Use of SMS would considerably reduce the need for central administration of upgrades. Options: ▪ Run applications on the client computer for best performance and reduced network traffic. ▪ Run applications from the server to save hard disk space on client computers and make it easier to upgrade components or drivers later, especially for multiple computers.

Table 1.5 Key Features of the Ideal Network Client

Preferred feature	Decisions and issues
Use TCP/IP (Transmission Control Protocol / Internet Protocol) *To provide the best network interoperability over WANs and network routers.*	A suite of protocols for communicating in a heterogeneous interconnected network. With TCP/IP used as the enterprise networking protocol, an IP addressing scheme is needed for your company. Another consideration is the use of WINS Servers or LMHosts files for name resolution and the need for connectivity with UNIX-based networks. ▪ When TCP/IP is used as a transport protocol with Windows NT, Windows NT computers can communicate with other kinds of systems such as UNIX workstations and servers or an IP configured printer without additional networking software. ▪ When Microsoft TCP/IP is used as the enterprise networking protocol, Windows networking solutions can be used on an existing internetwork. ▪ Microsoft TCP/IP in combination with other parts of Windows NT provides a scalable solution for enterprise networks that include a mix of system types and that require simple, easy software solutions that work on many platforms.
Use NWLink IPX/SPX (Internetwork Package Exchange / Sequenced Package Exchange) *The Microsoft NWLink IPX/SPX Compatible Transport protocol is an NDIS transport that provides communication between a Windows NT computer and another Windows NT computer or a NetWare server if Client Service for NetWare is installed.*	Depends on compatibility with your choice of client. Options depend on your choice of protocol. Novell also supplies an IPX ODI protocol for Windows NT Workstation. ▪ Microsoft IPX/SPX-compatible protocol is preferred (with or without IPX over NetBIOS). ▪ NWLink supports NetBIOS network applications, such as Lotus Notes. ▪ IPX ODI protocol might be needed to run some network applications.

Table 1.5 Key Features of the Ideal Network Client *(continued)*

Preferred feature	Decisions and issues
Use NetBeui (NetBIOS extended user interface) *A small, efficient, and fast protocol tuned for small LANs.*	NetBEUI is designed to support department-sized LANs that consist of 20 to 200 workstations. NetBEUI does not support traffic across routers. ▪ NetBEUI provides for both connectionless and connection-oriented traffic on a single network segment. ▪ NetBEUI is self-configuring and self-tuning. ▪ NetBEUI can be installed and bound to a network adapter card automatically when Windows NT is installed.
Use DLC (Data Link Control) *A protocol for accessing IBM® mainframe computers or printers attached directly to the network.*	The Data Link Control (DLC) protocol isn't used for general networking on Windows NT. The DLC protocol provided with Windows NT is used primarily to access IBM mainframe computers. For example, Microsoft SNA Server for Windows NT uses the DLC protocol device driver when communicating with mainframes on the token ring interface. ▪ Windows NT DLC allows Windows NT computers to connect to IBM mainframes by using 3270 emulators. You can also connect to IBM AS/400® computers by using 5250 emulators. ▪ Windows NT DLC works with either token ring or ethernet media access control (MAC) drivers. ▪ The DLC protocol works with Windows NT-based programs and with MS-DOS–based and 16-bit Windows-based programs.
Use AppleTalk® protocol *The protocols used to route information and configure zones so that a Windows NT Workstation computer can share resources with Apple® Macintosh® computers, if a Windows NT Server computer running Services for Macintosh is available on the network.*	The AppleTalk protocol is used to deliver data to a network destination when a Windows NT Server computer configured with Windows NT Services for Macintosh is available on the network, making it possible for IBM-compatible computers and Apple Macintosh workstations to share files and printers. ▪ The AppleTalk protocol is also used by software application developers who are creating cross-platform applications for Windows NT and the Macintosh. When used for transferring files across ethernet or for remote debugging in this way, Windows NT Server Services for Macintosh is not required on the network.

Table 1.5 Key Features of the Ideal Network Client *(continued)*

Preferred feature	Decisions and issues
Use Remote Access Service *Allows remote computers to dial into a LAN.*	Windows NT Remote Access Service allows a user at a remote site to dial in to a Remote Access server and use the network as if the computer were directly connected to the network. Remote Access Service contains two main components: ▪ Remote Access client, which is a computer that dials in to a remote access server to use the resource on a LAN. This component can be installed on a Windows NT Workstation computer. ▪ Remote Access server, which is a computer that allows a remote computer to dial in to and use the resource on a LAN. This component is available only with Windows NT Server.
Use Network Monitor Agent *Provides performance counters for the network adapter card and provides an agent that can be used by other monitoring software to analyze traffic on a LAN.*	The Network Monitor Agent on a Windows NT computer collects and displays statistics about the network adapter card in the computer. You can view the statistics yourself, and an administrator can arrange for the statistics to be collected by a central computer to help diagnose problems with the network. ▪ When your Windows NT computer runs the Network Monitor Agent, you see no indication that a central server is collecting the statistics; it has no impact on your work. ▪ A server running SMS and Network Monitor collects statistics from computers running the Network Monitor Agent. This data helps administrators perform routine troubleshooting tasks, such as locating a server that is down or that is receiving a disproportionate number of work requests.

Table 1.6 Recommended Windows NT Workstation Features for Client Configurations

Windows NT Workstation feature	Decisions and issues
Use User Manager *To enable centralized administration capabilities of Windows NT Workstation or add control of the user's desktop*	Choose this feature to enable centralized administration and control of user account and policies. Options: ▪ With User Rights Policy Editor, you can define policies at any time.

Table 1.6 Recommended Windows NT Workstation Features for Client Configurations *(continued)*

Windows NT Workstation feature	Decisions and issues
User Profile Editor *To allow multiple users to use a single computer with their own settings or, conversely, to allow personalized settings per user on multiple computers*	A User Profile is created by default when a user account is created. With Windows NT Server, it is possible to have server-based profiles. Server-based profiles can be user-based or mandatory. Options: ▪ Users can control changes to their user profiles and update them as they want. ▪ Administrators can predefine a mandatory profile for specific users, that can only be changed by the administrator.
Enable remote administration *To allow an administrator to remotely manage the file system, network sharing, or Registry of the individual computers*	Join a domain to enable remote administration privilege.
Use unattended answer files for installation *To allow automated installation on client computers*	Choose this feature if you must install Windows NT Workstation on more than five computers. Setup Manager offers an easy to use, graphical tool for creating unattended answer files.
Use peer resource sharing services *To allow a client computer to share files and resources such as printers and CD-ROM drives with other computers*	Choose this feature based on your site's security needs. If users are allowed to share local resources on their computers, then peer resource sharing can save network traffic and hard disk space on the server. For central control or to prevent users from turning on this feature, use User Rights Policy Editor.

Table 1.6 Recommended Windows NT Workstation Features for Client Configurations *(continued)*

Windows NT Workstation feature	Decisions and issues
Use user-level security *To implement control for a variety of services beyond network resource access, including File and Printer Sharing, Remote Registry, backup agents, and other network and system management functions*	Users and groups have access to local shared resources (including the Registry). Validation by a Windows NT Server or a NetWare server can also be required before access to any resources is possible under Windows NT Workstation. Options: ▪ Users can specify access rights for individuals and groups to shared resources. ▪ User access is validated based on user accounts on a Windows NT domain or a Novell NetWare bindery. ▪ User-level security is required for remote administration of the Registry and for network access to full user profiles. ▪ Optionally, share-level security can be used to protect files on Windows NT networks or Windows NT Workstation peer networks.

Table 1.7 Other Optional Windows NT Workstation Features

Windows NT Workstation feature	Decisions and issues
Use Windows NT Workstation mobile computing features *To enable Windows NT Workstation features that support mobile computing or switching between portable and docking-station configurations*	Depends on the particular hardware and the working needs of mobile-computing users. Some of these features are not installed by default, but can be specified in the installation process during Network Setup. Or they can be configured later through control panel network settings. Options: ▪ Remote Access Service client software for dial-up connection to the company network. ▪ User profiles to provide a custom desktop for each user, no matter where users log on to the network.

Performing the Lab Test

Using the preferred client configuration specified in the previous phase, proceed with installing the configuration in the lab for testing and evaluation. Because only the client-computer configuration is being installed (server installation is described in the following section), this test only determines whether the preferred configuration performs as expected, and whether it is compatible with your current applications and processes.

Depending on how the test installation proceeds, it might be necessary to modify the configuration, by either adding or removing selected features. If more than one configuration is being considered, side-by-side evaluations of different configurations can be performed to help determine which one works best.

The following checklist outlines the tasks in performing the lab test of the client configuration. These tasks apply for each computer used to install a client configuration.

Table 1.8 Performing the Lab Test

Task	Team	Start week	Duration
1. Make sure that the computer meets your company's standards and the Windows NT Workstation minimum standards for operation—at least a 16-MB 486DX/33 or better. If not, perform the hardware upgrades now.	Installation	Week 4	0.1 day
2. Defragment the hard disk, and scan it for viruses.	Installation	Week 4	0.1 day
3. Back up and verify key data and configuration files, such as INI, AUTOEXEC.BAT, and CONFIG.SYS files. Also back up the Windows and MS-DOS directories, and all files in the root directory. Make a system startup disk containing COMMAND.COM, SYS.COM, and FDISK.EXE.	Installation	Week 4	0.1 day
4. Make sure that the current network client software is functioning properly and, referring to the checklist of inventoried applications, make sure that all important applications operate correctly.	Installation	Week 4	1 day
5. Install Windows NT Workstation on the test computer in the lab, using the preferred client configuration identified in the previous phase. Use the installation method you will use in the final rollout. For example, if you will use Systems Management Server to roll out Windows NT Workstation, use it in the lab at this point.	Planning, Installation	Week 4	1 day

Table 1.8 Performing the Lab Test *(continued)*

Task	Team	Start week	Duration
6. Test the installation: ▪ Can you connect to and browse the network? ▪ Can you print both locally and across the network? ▪ Can you perform the core operations of each application locally and on the network (including opening, closing, and printing)? ▪ Can you shut down successfully?	Planning, Installation	Week 4	2 days
7. Optionally, if you have several test computers, compare your old client configuration under Windows 3.x and your new preferred configuration. How do the two compare in terms of the following: ▪ Functionality for administering the computer? ▪ Performance for local disk and network actions? ▪ Ease of use for performing common tasks? ▪ Stability of the computer under stress? ▪ Compatibility with applications and hardware?	Planning, Installation	Week 5	2 days
8. If the specified client configuration did not work as expected, modify and document the differences until a working preferred client configuration is installed.	Planning, Installation	Week 5	As required
9. Perform a complete restoration of operating system files and system capabilities for your old client configuration on the computer running Windows NT Workstation.	Installation	Week 5	1 day
10. Evaluate the restoration process for problems. Document the process and the modifications made.	Planning, Installation	Week 5	0.5 day
11. Have all team members participate in installing the preferred configuration on a variety of hardware.	Planning, Installation	Week 5	3 days

Planning the Pilot Rollout

In this phase, appointed teams determine the best methods for automatically installing the specified configuration for a pilot or trial rollout. Planning for this pilot program involves creating the automated installation process, determining the logistics of testing, and preparing a training plan for users. The following checklist outlines the tasks in planning the pilot rollout.

Table 1.9 Planning the Pilot Rollout

Task	Team	Start week	Duration
1. Install Windows NT Workstation source files on a server. Make setup choices based on your preferred client configuration tested in the lab. Perform the following steps: ▪ Set up the distribution server. ▪ Set up the client from the network.	Planning, Installation	Week 6	1 day
2. Create and test an automated installation by creating an unattended answer file to predefine answers for Setup. Document the key parts of the setup file that vary by installation.	Planning, Installation	Week 6	2 days
3. Determine and test how you will push the installation from the server without having to touch the client computers. Options: ▪ Modify login scripts on the server. ▪ Use management software, such as Microsoft Systems Management Server. ▪ Send a setup script (batch file) that runs Windows NT Workstation Setup as an embedded link in an electronic mail message. Document the process for the rest of the Installation team.	Planning, Installation	Week 6	3 days

Table 1.9 Planning the Pilot Rollout *(continued)*

Task	Team	Start week	Duration
4. Evaluate the Windows NT Workstation installation process for opportunities to upgrade or improve your organization's existing technology infrastructure. For example, a system management software tool can help you administer computers on the network more easily, and it can help with the push installation process.	Planning, Executive	Week 7	2 days
5. Document in checklist form the logistics of the pilot installation, such as the total time for installation, the new software or tools to be purchased, the group selected as the pilot users, and the scheduling of specific installations. Use this prior to the rollout to make sure you are completely prepared. Also, document goals for the pilot rollout to be used as evaluation criteria for rating the success of the rollout.	Planning, Installation	Week 7	3 days
6. Send a memo to your users to clearly explain how the installation process will affect their daily work schedule and describe the differences they will see after the installation is completed.	Planning	Week 7	1 day
7. Develop a user training course (or hire a training vendor to prepare one). Use the Windows NT Workstation Help to jump start your training efforts.	Planning, Support, Training	Week 6	5 days
8. Establish a support plan for the pilot user group. This includes the names and phone numbers of persons to contact for assistance, a short list of the top questions and answers, and troubleshooting tips.	Planning, Support	Week 7	5 days
9. Set up the lab or classroom with computers for training.	Training	Week 7	2 days
10. Edit the Windows NT Workstation Help file (if appropriate) to include any company-specific information. Repeat this after the pilot rollout is completed.	Planning, Support	Week 8	4 days

Conducting the Pilot Rollout

The goal of the pilot program is to test your automated installation in everyday use among a limited group of users (for example, between 15 and 50). This process helps to identify problems that could impede or delay the deployment process, and helps to determine what resources you'll require for the final, company-wide rollout. It's important to make the pilot rollout as successful as possible, because it sets the tone for the rest of the deployment process. If pilot users are satisfied, their enthusiasm can influence others to cooperate, which in turn helps the rest of the process to move smoothly.

The following checklist outlines the tasks in conducting the pilot rollout. Use the same pilot user group and follow the same tasks when rolling out 32-bit applications.

Table 1.10 Conducting the Pilot Rollout

Task	Team	Start week	Duration
1. Select a pilot user group that is willing and able (particularly in terms of their workload) to handle the installation process.	Planning	Week 8	2 days
2. Train the users.	Training	Week 8	5 days
3. Back up the Windows and MS-DOS directories and the files on the root directory of the test computers.	Installation	Week 9	5 days
4. Following the logistics checklist prepared in the previous phase, perform the installation in the same manner that you expect to install Windows NT Workstation throughout the company. Compare your results against goals and evaluation criteria (developed in the previous task) for this process.	Installation	Week 9	10 days
5. Have your technicians on site for the initial installations to document the process and problems and to support the users. Have other technicians monitor time and all measurable factors in the installation process. Record these measurements for later evaluation.	Support	Week 9	15 days

Table 1.10 Conducting the Pilot Rollout *(continued)*

Task	Team	Start week	Duration
6. Make sure that all computers are up and running as expected. Make note of possible improvements to the installation, training, or support, where appropriate.	Planning, Support, Installation	Week 11	3 days
7. Survey members of the pilot user group about their satisfaction with the installation process and take feedback on what could have been done better.	Planning	Week 12	3 days
8. Continue to monitor the pilot installation for a week to make sure that everything continues to run smoothly.	Support, Planning	Week 11	5 days
9. Prepare a checklist of issues to resolve for the final rollout. Include in this checklist the areas identified in step 6 as needing improvement, comments from the user survey, and the results of comparing your rollout goals and evaluation criteria against actual performance.	Support, Planning	Week 11	5 days
10. If the pilot program did not run smoothly or user feedback was poor, conduct additional pilot installations until the process works well.	Planning, Installation	Week 12	See "Planning the Pilot Rollout"

Finalizing the Rollout Plan

The results of the pilot installation provide the basis for developing a plan for final rollout. Using the actual time and resource requirements from the smaller-scale pilot rollout, teams make projections for time and resources, corresponding to the company-wide scope of the final rollout. If additional resources are required, identify these and acquire them at this time. In addition, update company policies and standards regarding computer and network use to accommodate the Windows NT Workstation implementation.

Table 1.11 Finalize the Rollout Plan

Task	Team	Start week	Duration
1. Determine your rollout goals—specifically the number of computers on which you will install Windows NT Workstation and the time expected for completion. During preparation for final rollout, check off items on this list as they are resolved.	Planning, Executive	Week 12	5 days
2. Budget the resources, in terms of personnel and tools, required to meet your goals.	Planning	Plannin g	3 days
3. If necessary, present the budget and obtain approval for the resources and the rollout process.	Planning, Executive	Week 13	2 days
4. Hire and train the extended Installation team and purchase the additional software or tools needed.	Training, Installation	Week 13	10 days
5. Update the company's hardware and software standards lists.	Planning	Week 13	2 days
6. Update the company's policies and practices manuals or guidelines for use of computers and the network.	Planning	Week 13	2 days
7. Notify your users that company standards and policies for computer use will be enforced before the installation, and that they must bring their computers into compliance.	Planning	Week 13	1 day
8. If appropriate, edit the Windows NT Workstation Help file to add company-specific Help for line-of-business applications.	Planning, Support	Week 14	3 days
9. For each computer, create a template as a database for documenting and tracking any system problems or deficiencies that require further attention.	Installation	Week 13	2 days
10. Post the updated template to a central network location.	Installation	Week 13	2 days

Rolling Out Windows NT Workstation

After the extensive research, planning, testing, and analysis performed in the previous phases, the deployment teams arrive at the final phase—rolling out the Windows NT Workstation installation to the entire company. Although each prior phase was critical to the overall success of the deployment process, only this phase can fulfill the purpose of the entire planning process, by delivering the substantial new benefits of Windows NT Workstation to your broadest base of users. At this phase, weeks of preparation pay off in a smooth migration of all your users to an operating system that is more powerful, more robust, and easier to use.

The following checklist outlines the tasks required for the final rollout of Windows NT Workstation.

Table 1.12 Rolling Out Windows NT Workstation

Task	Team	Start week	Duration
1. Set up the distribution servers.	Installation	Week 15	1 day
2. Customize the server installation.	Installation	Week 15	2 days
3. Notify the users of the upcoming installation.	Planning	Week 15	1 day
4. Train the users on Windows NT Workstation.	Training	Week 16	As required
5. If needed, upgrade the hardware on the client computers and remove any software not complying with company policy.	Installation	Week 16	As required
6. If needed, back up critical data and configuration files on the client computers.	Installation	Week 16	As required
7. If needed, defragment the client hard disks.	Installation	Week 16	As required
8. Optionally, you can temporarily reset the user password and ID for each computer, to allow your technicians easy access to the client computer and make sure that the login scripts and environment operate correctly.	Planning	Week 17	As required
9. Make sure that the client computers are fully operational and the real-mode network, if present, is running.	Installation	Week 17	As required
10. Prepare the client computers for the push installation process: edit the login scripts; run the management software; or send the setup script, by electronic mail, to the users.	Installation	Week 18	As required
11. Initiate the installation by having the users log on, double-click the setup script file, and so on.	Installation	Week 18	As required

For step-by-step instructions on how to set up, maintain, and use Windows NT Workstation in a corporate environment, see the appropriate chapters of the product documentation and the *Windows NT 3.51 Resource Kit.*

CHAPTER 2

Deployment Strategy and Details

This chapter describes a plan for implementing a large-scale installation of computers running Windows NT Workstation.

Reviewing Windows NT Workstation Features

During this initial phase, the Executive and Planning teams learn about Windows NT Workstation features and benefits. For example, they learn how Windows NT Workstation helps reduce support costs and increase business profitability. Publications are available from Microsoft Press and from independent industry analysts to provide the information you need.

During this phase, the Executive and Planning teams need to review the Windows NT Workstation product documentation and version 3.51 of the *Microsoft Windows NT Resource Kit*. Written to assist administrators in installing, supporting, and managing Windows NT Workstation 4.0 on corporate networks, the *Windows NT Resource Kit* is a technical supplement to the Windows NT Workstation product documentation.

Preparing the Teams

This phase involves gathering the resources, including equipment, software, and staff, to properly plan for testing and evaluating Windows NT Workstation. Members of the Support team should receive training during this phase.

Acquiring Staff and Software

The deployment project manager participates in the Executive team and leads the Planning team. This individual is usually the head of the Information Systems department. You might find that a Microsoft Solution Provider is the best choice for a project manager, since they can provide expertise without diverting resources from other work in your organization.

When setting up the Planning team, try to include individuals from the various groups involved in the deployment process. This includes people from the Corporate Support and Employee Training departments, the Corporate Standards Committee, and key Installation team members. Individuals from the Finance and Accounting group will need to take part in planning and evaluation later on, but need not be assigned to the team for the full duration of the deployment process.

Your Installation team should include an applications expert who can evaluate 32-bit applications that run with Windows NT Workstation.

Obtain Windows NT Workstation during this phase. It is recommended that you purchase the compact-disc version, so that you can use Server-based Setup and administrative software tools not provided on the floppy disks.

Conducting a Sample Inventory

You'll need to survey a representative sample of your network to identify the hardware and software typically used on client and server computers. By doing this sample inventory of your company's active equipment, you can accurately simulate the organizational environment in the lab. Such a simulation helps you make broad decisions about your company's computing infrastructure, such as the choice of protocol or the default desktop configuration as it pertains to applications.

Software management tools are available to query computers on the network for hardware and software configurations. For detailed information about a large number of computers on a network, use a system management program, such as the Microsoft Systems Management Server to conduct the inventory.

Testing Lab Setup and Equipment

To effectively evaluate and test the Windows NT Workstation installation process, you need to set aside enough physical space and assemble a sufficient number of computers to test everything from Server-based Setup to hand-tuning options for the local computer. In addition, if your network environment includes the use of portable computers that dial in to the company, or if you use additional servers or mainframe computers for business data, you need to make sure that the lab computers have full access to the network and an analog phone line.

It is important that you test and implement all of the Windows NT Workstation features comprehensively in the lab with all of your mission-critical and noncritical business applications before moving to the pilot installation.

Installation of Windows NT Workstation on a server requires 90 megabyte (MB) of free disk space.

Training the Teams

By reviewing specific portions of the product documentation and the *Windows NT Resource Kit* version 3.51, the Installation and Planning teams can gain an extensive understanding of Windows NT Workstation features and functionality.

Support team members must become familiar with all information in the *Windows NT Resource Kit* to prepare for their role in the deployment process. Team members can receive instruction at a Microsoft Authorized Technical Education Center and participate in the Certified Professional program. Call (800) SOLPROV (or (800) 765-7768) for information about authorized training offered for Windows NT Workstation and the Certified Professional program, and for referral to a local Microsoft Solution Provider Authorized Technical Education Center (ATEC).

Deciding on the Preferred Client

Detailed analysis is required to determine your preferred client-computer configuration. Starting with the ideal configuration, which uses the most functional and best-performing client software, evaluate each feature against your organization's needs and environment to determine whether the feature is appropriate and compatible. If you are considering different configuration alternatives, repeat this evaluation for each configuration.

The following sections describe features and decisions to evaluate in specifying the network client configuration.

Configuration Layout

When deciding where to place Windows NT Workstation files, consider how the computer will be used, and evaluate the benefits of each placement option. Data files shared by the users in a group are generally kept on a server. Windows NT Workstation executable files, and the executable files for applications, can be stored either on network servers or on the local hard drives of individual computers. Swap files and TEMP files can reside either on a network drive or on the local drive. Here are some guidelines to help you decide which option to choose.

Place executable files, swap files, and TEMP files on the workstation's local hard disk if the following is true:

- The computers are personal workstations.
- The computers are portable computers that occasionally connect to the network.
- The computers are used in workgroups that only share data and applications, such as word processors (not operating system software).

Install Windows NT Workstation files so that all executable files and applications run from the network if the following is true:

- You want to run a shared copy of Windows NT Workstation on computers that do not have hard disks.
- You want to provide a central location for managing users' system configurations.

In this case, swap files and TEMP directories are placed on network drives.

Support for diskless workstations is available for NetWare® networks with the initial release of Windows NT Workstation. For information about support under Windows NT, contact your Microsoft sales support representative.

Key Features of the Ideal Configuration

This section describes the features that might be included in an ideal network client configuration.

Using a 32-bit, Protected-Mode Network Client

For best performance, Windows NT Workstation 3.51 includes the 32-bit Microsoft Client for NetWare Networks and the Client for Microsoft Networks. Each of these has a 32-bit redirector. The benefits of using a 32-bit, protected-mode client include the following:

- Provides for easy installation and configuration using built-in Windows NT Workstation 3.51 tools
- Provides faster data I/O across the network
- Offers greater stability than real-mode redirectors
- Allows more than one redirector to be run at one time, and thereby enables access to servers for multiple networks without having to reload the operating system for a new network client
- Makes networking seamless in the Windows NT Workstation user interface; users can browse the server for multiple networks in Network Neighborhood, all within the same name space—users don't need to know which type of network they are browsing

If you are using another type of network, contact your network vendor regarding the availability of a 32-bit, protected-mode network client. If a protected-mode client is unavailable, you can run a protected-mode Windows NT Workstation 4.0 client such as Client for Microsoft Networks in conjunction with a real-mode network client.

Using a 32-bit, Protected-Mode Protocol

If you select a 32-bit, protected-mode network client, then by default Windows NT Workstation 4.0 also sets up a 32-bit, protected-mode protocol. Even if you are running a real-mode client, such as the Novell® 3.x workstation shell (NETX) with a real-mode implementation of IPX/SPX to access NetWare® servers, you can still load the 32-bit version of the Microsoft IPX/SPX-compatible protocol. The benefits of adding the protected-mode protocol are better performance and better stability for network communications to servers that are not running NetWare (for example, computers running Windows NT Workstation or Windows NT Server).

In addition, for protocols such as TCP/IP, the Microsoft 32-bit version enables additional functionality such as the ability to use DHCP and WINS servers that dynamically set the IP addresses and resolve computer names for client computers on the network.

Recommended Features for Network Clients

The following optional features are recommended for your preferred configuration. These features define how Windows NT Workstation will be installed and administered in your organization.

Using User Manager

For centralized administration of client computers, you must enable system policies. User Manager allows you to centrally edit and control individual user and computer configurations. For example, if you want to place a custom Start menu on user desktops or limit access to Control Panel options, User Manager makes it easy to do this from a central location for a large number of users.

Enabling policies creates a single file that resides on the server, and thus does not involve physically touching the client computer. In general, the policy file can be modified on the server after Windows NT Workstation is installed; however, some types of changes, such as adding group support or a nonstandard server path for product updates, require configuration on the client computer.

Using User Profiles

With user profiles, users can use personalized desktop settings each time they log on to a computer. This is especially useful for multiple users sharing a single computer who want to customize their desktops and have those custom settings loaded at logon. Conversely, a single user can move between computers using the same profile if the administrator stores that profile on the server. An administrator can also take advantage of profiles to require that a mandatory desktop configuration be loaded each time a user logs on. The ability to change profile settings can be controlled by the administrator.

User profiles are not needed when only one person uses the computer or when a custom desktop adds no value. By not enabling user profiles, the logon process is shortened slightly, because the system does not need to locate and load the profile.

Enabling Remote Administration

To remotely administer a computer's Registry, you must first install the network service called Microsoft Remote Registry service, enable user-level security, and enable the Remote Administration feature. Remote administration capabilities allow you to conduct a variety of tasks remotely over the network, such as administering the file system, sharing or restricting directories, or querying and making changes to the Registry. If you plan to do any of these tasks, be sure to enable this feature during Windows NT Workstation installation.

You should not enable remote administration if you don't need these services, because doing so causes unnecessary, extra processes to run on the client computer and on the network. These extra remote services could then *theoretically* be used by individuals on the network—provided they knew the appropriate password—to access information on client computers. However, Windows NT Workstation comes with security capabilities to protect against unauthorized use of the Remote Registry service.

Using Unattended Answer Files

Unattended answer files allow you to predefine responses to prompts that appear during Windows NT Workstation Setup. The choice to use an unattended answer file is straightforward. If you need to conduct a similar installation more than five times, you should use one. Begin planning for unattended installations during this phase, as you are specifying the preferred client configuration. Make sure that you document each feature needed, so that you can automate the selection of these features.

For more information, see "Deployment Utilities" in Chapter 3, "Lab Tests."

Using Push Installations

You need to understand and plan in advance how the push installation process will work for a given computer. There are several alternatives for remotely initiating the installation, ranging from editing the client's login script, to sending by electronic mail a link that contains a setup script. You will want to consider how to push the installation for each computer and make sure that the client computers are configured to support this process.

For organizations with 50 or more computers, being physically present to install each client computer is not a viable option, because of the cost. In that case, you might need to turn to an administrative software solution, such as Microsoft Systems Management Server. When using administrative software tools, additional client-side software might be needed. Be sure to include this software in the installation plan.

For more information, see Chapter 6, "Unattended Installations and Upgrades."

Using Peer Resource Sharing Services

The peer resource sharing capability in Windows NT Workstation allows your client computers to share files and printers directly from a local personal computer, instead of on a central server. Peer resource sharing may reduce the traffic and disk space required on central servers, because you are leveraging the power of individual computers.

Security for peer resource sharing services takes the form of user-level security, based on the user accounts on a Windows NT or NetWare network. Notice that a Microsoft Windows NT Client Access License is required if the computer will be connecting to servers running Windows NT Server. For information, contact your Microsoft reseller.

If you don't have servers to provide security validation or don't want to use user-level security, you can use share-level security, with each individual implementing security and a password scheme on the local computer. Share-level security is set on a directory-by-directory basis.

If you do not want to use peer resource sharing services and want to disable the capability on each client computer, you can do so by selecting the appropriate option in system policies.

Using User-Level Security

User-level security is based on user account lists that are stored on servers runing Windows NT Server or Novell NetWare. The user accounts specify which users have access rights on the network. Windows NT Workstation passes on a user's request for access to the servers for validation. Pass-through user-level security protects shared network resources by requiring that a security provider authenticate a user's request to access resources.

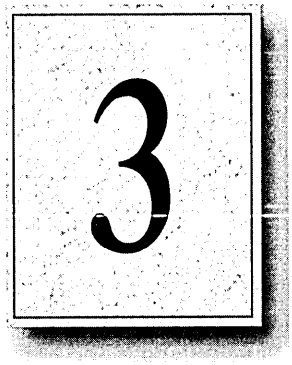
User-level security is required for remote administration of the Registry and for network access to full user profiles.

Using Remote Access Service

The Remote Access Service (RAS) is client software that allows the computer to use popular, server-based dial-in packages, such as Novell NetWare Connect and Shiva NetModem. RAS provides additional security for remote dial-up connections and requires some additional configuration of protocols and software.

CHAPTER 3

Lab Tests



This phase in the deployment process includes preparing the site, conducting the installation, testing the installation, and restoring the system.

Preparing for the Lab Test

Preparing the test site and equipment involves the following tasks:

- Ensuring that the installation site suits your needs. For example, make sure that you have the appropriate jacks for connecting to the network.
- Ensuring that the computers meet the minimum requirements for running Windows NT Workstation. For example, make sure the computers have the appropriate amount of free hard disk space, at least 12 megabyte (MB) of random access memory (16 MB is recommended), and a processor (486DX/33 or better is recommended).
- Running virus detection, disk scanning, and defragmentation programs on the computers to correct any problems there might be before installation. Although the computers might appear to be operating properly, software upgrades often uncover hardware or software problems, because of the way they read and write data to the hard disk.
- Backing up critical data and configuration files for the system, in case the installation fails or you need to revert to the previous operating system for some reason. This task includes backing up .INI files (such as WIN.INI and SYSTEM.INI), GRP files, AUTOEXEC.BAT, CONFIG.SYS, and all key data files. As an added precaution, create a system startup disk and back up the Windows and MS-DOS directories and all the files in the root directory.

Tip If you need to automate the restoration, consider using a commercial backup program, instead of copying the files by hand.

Installing Windows NT Workstation

Before setting up Windows NT Workstation for the first time, verify that the existing network is working properly. Then use the product documentation and Chapter 3, "Customizing Windows NT Setup," in the *Windows NT Resource Kit* version 3.51 to help you correctly install and configure Windows NT Workstation. Take note of which options you want to predefine as entries for the TXTSETUP.SIF file used for the setup script, which can be used to automate the installation process.

Testing the Installed Configuration

After you've set up a computer with Windows NT Workstation, you'll need to run a variety of tests to make sure that it runs correctly on your network and that you can still perform all of your usual tasks. Use your own testing methodology or test the following to verify correct system operation:

- Connect to and browse the network.
- Set up a printer and test printing to local and network printers.
- Open, run, and close applications on both the client computer and on the server.
- Shut down the computer.

Make sure to test all mission-critical applications for proper function. If you encounter problems, try removing related features from the proposed configuration as a solution. Document any changes made to the original configuration.

Testing Optional Features and Components

If the preferred client configuration works as expected, you might also want to conduct additional testing of the optional software features and components in Windows NT Workstation. This task can help you determine whether you are running Windows NT Workstation optimally. For this kind of testing, conduct side-by-side evaluations on two computers, changing individual features on each one, to determine the following:

- Performance in terms of responsiveness and throughput
- Ease of use
- Stability
- Compatibility
- Functionality

Testing in a NetWare Environment

To evaluate network client software for Novell NetWare, run your network performance tests in the following configurations:

- Windows NT Workstation installed with an existing 16-bit, Novell-supplied workstation client (NETX), using ODI drivers
- Windows NT Workstation added to an existing installation of Windows 3.*x* and NetWare, using Client for NetWare Networks and protected-mode networking support components (NDIS adapter drivers)
- Windows NT Workstation as a new installation using all protected-mode components, including both Client for NetWare Networks and Client for Microsoft Networks, plus peer resource sharing support

Perform several common tasks, such as connecting to the network, administering a remote NetWare server, and so on, to test for ease of use. Similarly, you'll want to run any business-specific NetWare applications under Microsoft Client for NetWare Networks to make sure that they run compatibly. Any stability issues should become apparent during this testing.

When you have identified a configuration that performs well during testing, test the same configuration using other hardware from your company.

Testing the Restoration Process

After thorough testing of the preferred client configuration, completely restore one of the test computers to the previous client configuration and document the process.

Testing Deployment Procedures

The deployment procedures you use will depend in part on which of the available utilities you feel best suit your organization and the number of computers scheduled to receive the installation. If you are installing on large numbers of computers that are already using a network, you will probably want to automate the installation.

Using Deployment Utilities

An existing client-server based network is required to take advantage of Unattended Setup options. An installation of Windows NT Workstation can be set up on an existing client computer with access to the distribution server where setup files are located. This type of installation is referred to as an upgrade, in the sense that the client operating system is being upgraded to a more powerful and useful way to access and use the network. Most often an upgrade is thought of as replacing an older version of the same operating system, which is also a viable option. Installation of new operating systems is included in the upgrade category, because they replace older ways of connecting to the network.

There are several methods for upgrading your existing network clients to Windows NT Workstation:

- Unattended upgrades can be performed on small local area networks (LANs), enabling system administrators to upgrade Windows NT Workstation on several existing computers without having to monitor the installations.
- If you have a large network, and a number of servers, Microsoft Systems Management Server (SMS) provides the most control, speed, security and flexibility for smooth, easy migration. For example, SMS can deliver a job to prepare a target computer by running a virus detection program or a job to run an Unattended Upgrade.
- Other options include modifying logon scripts so setup runs automatically when a user logs on to the network, or sending a batch file as an embedded link in an electronic mail message to run Setup from individual desktops at the convenience of the user.

The deployment utilities available to you include the following:

- Unattended Setup is designed to assist in relatively few upgrades, or to upgrade computers that are not connected to a network.
- Computer Profile Setup (CPS) is designed to do all the preliminary installation work on large numbers of identical systems. The computername, username, and any other unique information is performed in a separate step. CPS would be used if your organization buys or sells large numbers of identical computers.
- Systems Management Server (SMS) uses an existing network to maintain a database of the hardware and software in use on a network, and to deliver jobs to computers on the network. The database can be queried to produce the detailed inventory that is a necessary first step to any deployment, and jobs can be sent to run the Setup program for any software, including Unattended Setup or the **winntp** portion of CPS. You can send jobs tailored to the needs of different groups of users or different types of computers.

Unattended Setup or Upgrade

This section contains information about how to automate the installation or upgrade of Windows NT Workstation and Windows NT Server so that the Setup program runs without requiring user input. Be sure to read the entire contents of this section to ensure that you are aware of the capabilities of the Unattended Setup mode. Also, read the commented UNATTEND.TXT file included with this resource kit for additional information.

Make sure the Unattended Setup is run in the test lab before implementing the functionality in a production environment. If the Unattended Setup does not complete, pauses for user input, or configures hardware incorrectly, computer usage could be interrupted.

To use Unattended Setup, you must do the following:

- Place the Windows NT Workstation or Windows NT Server distribution files on a server. If you have both products, the distribution files of each can be placed on the same server, but the files must be in different directories.
- Create one or several answer files.
- Connect to the distribution server from the computer where Windows NT Workstation or Windows NT Server is to be installed or upgraded.
- Start the Setup program.

Performing an Unattended New Installation

To run Setup in Unattended mode for new installations, use either the **winnt** or **winnt32** command with the **/u** option, and specify the location of an answer file. Use the **winnt** command to install Windows NT Workstation or Windows NT Server on a computer running MS-DOS. Use the **winnt32** command to upgrade Windows NT Workstation or Windows NT Server on a computer already running Windows NT Workstation or Windows NT Server. This command is discussed in more detail later in this chapter.

The Unattended Setup command syntax is as follows:

winnt /u:*answer_filename* **/s:***source*

or

winnt32 /u:*answer_filename* **/s:***source*

where:

/u
Indicates Unattended Setup mode.

answer_filename
Includes the location of the answer file.

/s
Indicates that the distribution files are located on a different drive.

source
Is the location of the distribution files.

For example, to connect to the \\CORPNET\WNTW server where distribution files are located, type the following command at the command prompt on the computer where you are about to install Windows NT Workstation or Windows NT Server:

```
net use x: \\corpnet\WNTW
```

The X: drive on the computer is connected to the server.

To begin an Unattended new installation on a computer running MS-DOS, type the following at the command prompt:

```
x:\winnt /u:c:\unattend.txt /s:x:\
```

or

```
x:\winnt32 /u:c:\unattend.txt /s:x:\
```

Either command indicates that an unattended installation should be performed. The command also indicates that an answer file is located on the computer's C drive and that the source of the distribution files is the X drive.

When using the **winnt/u** or **winnt32/u** command, by default, new installations are installed on the same drive as the temporary directory that is created on your computer. However, the default location can be overridden if a **/t** option is specified with the Unattended Setup mode command.

To override the default location of a new installation on a computer running Windows NT Workstation or Windows NT Server, type the following command at the command prompt:

```
x:\winnt /u:c:\unattend.txt /t:e /s:x:\
```

or

```
x:\winnt32 /u:c:\unattend.txt /t:e /s:x:\
```

where:

/t:e
Indicates that Windows NT Workstation or Windows NT Server files should be installed on the E drive.

Performing an Unattended Upgrade

The **winnt32 /u** command is also used to perform Unattended Upgrades from Windows NT Workstation 3.*x* or Windows NT Server 3.*x*. Specifying an answer file with the command is optional, because the Setup program can use the existing information on the system. However, specifying an answer file enables you to have more control over the upgrade. To change or override the existing information on the system, you must specify an answer file with the Unattended Upgrade command.

▸ To perform an Unattended Upgrade

1. Connect to the server where the distribution files are located.
2. Type the following command at the command prompt:

```
winnt32 /u
```

Note To perform an Unattended Upgrade and change existing information on the system, type the following command at the command prompt:

```
winnt32 /u <answer_filename>
```

---

3. When prompted, enter the location of the distribution files.

If you are upgrading a computer running Windows NT 3.1 and TCP/IP, you must specify an answer filename with the Unattended Setup command if you want to enable automatic Dynamic Host Configuration Protocol (DHCP) configuration. Windows NT 3.1 did not include DHCP, so the system would not contain existing information about DHCP for the Setup program to use. For more information about the specific parameter that must be used in the answer file to enable automatic DHCP configuration, see the discussion of the !UpgradeEnableDhcp parameter in the UNATTEND.TXT file.

Note If your computer contains an SCSI drive, check the *Hardware Compatibility List* to ensure that it is supported in this release of Windows NT Workstation or Windows NT Server. In rare instances, Windows NT Workstation or Windows NT Server does not include a driver for certain SCSI drives. If the SCSI drive is the drive on which you want to install the operating system startup files and the drive is not supported, Unattended Setup will not work, because the target drive is not visible to the Setup program.

For more information about the **winnt** or **winnt32** command, see Chapter 1, "Installing Windows NT Workstation" in the *Windows NT Workstation Installation Guide,* or see Chapter 1, "Installing Windows NT Server" in the *Windows NT Server Installation Guide.*

Setting Up and Configuring a Windows NT Workstation

This section is for administrators not using an unattended answer file for setup of Windows NT Workstation 4.0.

Before you set up Network Client, you need to determine the following:

- The username to be used

  The username identifies a member of the workgroup or domain. Choose a unique name in the workgroup or domain.

- The name to be assigned to the computer

  This unique name identifies the computer within the network. Often this name is a variation of the username.

- The name of the user's workgroup and/or domain

  This name determines how your computer fits in with other computers that are already organized into groups on the network. These are not names that you make up; they already exist on the network. Ask the network administrator if you don't know what names to use.

- The manufacturer and model of the network adapter

  The network adapter is the card inside the computer where you plug in the network cable. The Setup program attempts to determine the model of network adapter in the computer, but it might be incorrect. Some network adapters have further configuration options. See the network adapter documentation if you need to change configuration options.

- The network protocol used on this network

  The protocol is like a language used by computers to talk to each other. Your computer must use the same protocol as the computers to which it connects. If you don't know what protocol the computers on this network use, ask the network administrator.

There are other options that you can change with the Setup program, but these are the most important options.

Setup Manager

Windows NT Setup Manager is an administrative tool that enables system administrators to install or upgrade Windows NT on several existing client computers without having to monitor the installations or upgrades. Setup Manager is used to create answer files that are used by Windows NT Setup to perform unattended installations or upgrades of Windows NT. Answer files contain the information that Setup would normally prompt users for while it is installing or upgrading Windows NT.

The parameters set in answer files do not automatically eliminate the necessity for user input when Setup runs. The parameters in the answer file must be set so as not to require user interaction. For more information, see Setup Manager Help. Also, if you are upgrading an existing Windows NT installation, Setup will use the parameters of the existing installation and ignore parameters specified in the answer file.

▸ **To use Setup Manager**

- Double-click the Windows NT Setup Manager icon in the Administrative Tools program group, or, at the command prompt, type **setupmgr** with the appropriate options:

  setupmgr [*answerfile*]

  Where

  answerfile

  Is a text file supplying the answers to the prompts encountered during setup.

  If the file specified on the command line does not exist, you will be prompted for the name of a file to edit or you will be given the option to start a new file.

If Setup Manager is started without the name of an answer file, the default values will be loaded into the application. You can then select the Open button to open an existing answer file or edit the default values and then save it as a new file. For more information, press F1 while using Setup Manager.

The following files are required for Setup Manager:

- SETUPMGR.EXE
- SETUPMGR.INF
- SETUPMGR.HLP

Computer Profile Setup

Computer Profile Setup (CPS) enables you to easily install either Windows NT Workstation or Windows NT Server on multiple identical x86-based computers.

Two utility programs are included in CPS: UPLODPRF.EXE (Upload Profile), and WINNTP.EXE (Windows NT Profile Setup). The **uplodprf** command is used to make a copy (a profile) of an installed Windows NT system on a *source* computer. This profile can then be loaded and installed on any number of identical *target* computers, by using the **winntp** command.

If you have a few minor variations in configuration, you might choose to merge master profiles and difference profiles with CPS. In this case, you create one master profile, with smaller profiles that only specify variations from the master profile.

If there is a lot of variety in the configurations used in your organization, you might want to create unattended answer files to answer all of the questions that Setup asks, rather than using **winntp** to install on target computers. The Windows NT Setup Manager utility, included with this resource kit, makes it easy to create the unattended answer file. You can use **uplodprf** to create a baseline answer file, then modify it with Setup Manager.

See the "Setup Manager" section, earlier in this chapter, for more information.

Getting Started with Computer Profile Setup

The following sections describe the prerequisites for Computer Profile Setup.

Source (Model) Computer Requirements

The source computer is the prototype where the system that will serve as the Computer Profile is installed and configured. Before running any CPS utilities on this computer, you must first use Windows NT Setup to install Windows NT Workstation or Windows NT Server. Then load any additional software you want to propagate with this Computer Profile.

Only common program groups are propagated with the Computer Profile, so be sure that any installed software uses common program groups. Otherwise, these program groups will not appear on the target computers.

Local user accounts are not part of the profile; however, local group accounts and domain and local group permissions are part of the profile.

After the source computer has been configured, you can load the CPS utilities onto the source computer, upload the profile to the distribution server with UPLODPRF.EXE, and distribute the configured system to target computers by using the **WINNTP** command.

Note All software to be propagated should reside on the same volume (logical disk) as the Windows NT system directory, which should also be the boot drive (typically C:\systemroot\SYSTEM32).

Target Computer(s) Requirements

The target computer must have access to the Computer Profile directory on the distribution server (either by way of the network or a removable drive) and must have MS-DOS 5.0 or later installed.

The target computers should have hardware configuration identical to that on the source computer, but the following exceptions might not cause problems:

- The disk on the target computer can be larger, but cannot require a different driver, because the required driver might not be installed or configured properly.
- Memory (RAM) can be larger on the target computer.

Problems are sure to occur in the following situations:

- Different network card on the target computer. Hardware Autodetect is not run during Profile Setup, and the necessary driver file will not be available.
- Smaller disk or RAM on the target computer.
- Any hardware (such as a video monitor) that requires a different driver or configuration from the source computer.

If you have systems that are slightly different, you can merge profiles rather than dedicate space for a full profile for each of the similar systems. This task involves making a master profile and then as many difference profiles as needed.

For more information, see the RKTOOLS.HLP Help file in the *Windows NT Resource Kit* version 3.51.

Using Computer Profile Setup

Setting up multiple computers with Computer Profile Setup requires these activities:

- Setting up the source computer
- Uploading a copy of the configuration to the distribution server by using **uplodprf**
- Copying the configuration to the target computers by using **winntp**

▸ To set up the source computer

1. Configure the source computer, including all the directories and files on the Windows NT system drive that are to be part of the Computer Profile.

   Windows NT Workstation should be installed on the boot drive. Other applications and files to be propagated should be on the same volume as Windows NT Workstation.

2. Create a new directory, or clean an existing directory where the Computer Profile is to be stored, either on a network share point or a removable disk.

3. Decide which file system to install on the source computer, either keeping the existing FAT file system or converting the volume to NTFS after files are installed.

   It's easiest to configure the source computer with an NTFS volume (for example, using Autoconvert during Setup). Then you can create either FAT or NTFS on the target computer by using Computer Profile Setup. Going from FAT on the source computer to NTFS on the target computer might not provide the correct file protection (because the defaults will be used). You need to set permissions on the source computer for these to be propagated; therefore, use NTFS on the source computer.

4. Copy the CPS utilities to the source computer (or run them from a floppy disk).
5. Edit the input (.INI) file to customize the action of **uplodprf**.

   At a minimum, the entries in the [DefaultInfData] section (company name, time zone, and others) should be either set to describe your organization or set to null. You are now ready to run **uplodprf**.

Running UPLODPRF (Upload Profile)

The **uplodprf** (Upload Profile) command reads the Windows NT Registry, user account, and security information of the source computer and generates the necessary information and Registry files for reinstallation, and then copies all the files to the Computer Profile directory on the distribution server.

▸ **To run the Upload Profile program**

At the command prompt on the source computer, type **uplodprf** with the appropriate options:

uplodprf [*switches*] **/s:***driveletter* **/i:***filename* [**/a** | *\dirs*]

Where:

/a

Saves the entire volume containing the Windows NT Workstation operating system. This switch only copies files in subdirectories of the root directory and not files in the root directory itself. Because root directory files might be special and not always suitable to copy to other computers, the files found in the root directory that are to be part of the profile must be manually added to the [SystemFilesToSubstitute] section of PROFILE.INI. An example of such a file is NTLDR. If you are not planning to use the **/a** option and want to specify the directories to profile on the command line, then the default version of PROFILE.INI should work fine.

/b

Copies only the boot sector to the specified file.

/b-

Does not copy the boot sector. (All other processing is performed.)

/f:{f | n}

Overrides the default file system to be installed on the target computer. The **/f:f** option will install a FAT file system on the target computer, and the **/f:n** option will install the NTFS file system. The default is to install the same file system as is found on the source computer.

/h

Used for profiling only the Registry. Causes only the files listed in the [HivesOnlyFilesToSubstitute] section of the PROFILE.INI file to be copied to the share directory. For example, to copy certain files (such as those used for a specific adapter driver) when uploading the profile with the hives-only option, you would list those files in the [HivesOnlyFilesToSubstitute] section of PROFILE.INI, and then use the **/h** option with the **uplodprf** command.

/p:*filename*

Specifies the name of the setup script file (unattended answer file) to create for use in Unattended Setup with the **winnt** or **winnt32** command (as opposed to using **winntp**). Script files are used to answer all of the questions that Setup asks. The Setup Manager utility, included with the *Windows NT Resource Kit* version 3.51, makes it easy to modify the script file.

/m

Rescans the share directory and update the WINNT.INF file.

/n

Dumps the access control lists of the files found in the directory list (for NTFS volumes only).

/n-

Does not dump access control lists in the profile.

/q

Suppresses information messages. If you want to check progress messages, you can save them in a log file by using MS-DOS redirection, then search the log file for "Error" to see if there were any problems in copying files. For example:

```
uplodprf /s\\mysys\profile /i:profile.INI /a > profile.log.
```

/r

Dumps the access control lists of the profiled Registry keys.

/r-

Does not dump the Registry key access control lists.

/u

Generates the user account definition file specified in the .INF file, and nothing else.

/u-

Does not generate a user account definition file.

/s:*driveletter*

Specifies the location of the Computer Profile directory, which will contain the information derived from the source computer to be used by the target computers for downloading and installation. This directory must be emptied before you run **uplodprf**.

/i:*filename*

Specifies the computer setup information file, for example PROFILE.INI.

\dirs

Additional directories from the source computer's *\systemroot* directory to be included in the profile.

Setting Up the Target Computer

Once the profile has been created, the **winntp** command can be run on the target computer.

▸ **To install the Computer Profile files on a target computer**

At the command prompt on the target computer, type **winntp** with the correct options.

winntp [**/D:***sysroot*] [**/S:***sourcepath*] [**/T:***tempdrive*] [**/I:***inffile*] [**/B**] [**/B-**] [**/E:**{*YES*|*NO*}] [**/X** | [**/F**] [**/C**]] [**/M:***COMPUTERNAME*] [**/N:***domainname*] [**/O:***orgname*] [**/U:***username*] [**/Z:***timezone*]

Where:

/D[:]*sysroot*
Removes the Windows NT Workstation operating system files from the installation in SysRoot (the root directory for the Windows NT system files, which is usually C:*systemroot*\SYSTEM32).

/S[:]*sourcepath*
Specifies the source location of the Windows NT Workstation files for Computer Profile Setup, which contains the information derived from the source computer. This must be a full path name of the form **x:**[*path*] or *server**share*[*path*]. To load several profiles in sequence, list them, separated by commas, in the order you want them to load. For example, list the master profile first, then any difference profiles.

/T[:]*tempdrive*
Specifies a drive to contain temporary setup files. If this switch is not specified, Setup attempts to locate a drive for you.

/I[:]*inffile*
Specifies the filename (with no path) of the Setup information file. The default is DOSNET.INF, which is created during the upload process by **uplodprf**. If you specify a different filename in the WinntInfFileName entry in the [SetupFiles] section of the PROFILE.INI file, you must use this switch for that filename to be used with the **winntp** command.

/B

Specifies that the computer reboot and continue with graphical mode setup as soon as all files are downloaded, without waiting for input from the user.

Note If the **/B** option is used, the user must ensure that there is no disk in the floppy drive. The user will not be prompted to remove the disk. If the user is running **winntp** from a floppy disk, and using the **/B** option, the disk can be removed as soon as **winntp** starts copying files.

/B-

Specifies that the computer does not reboot after all files are downloaded, but exits to the MS-DOS prompt.

/C

Specifies to skip the free-space check on the Setup boot floppy you provide.

/F

Specifies not to verify files as they are copied to the Setup boot floppy.

/X

Specifies not to create the Setup boot floppy.

/E:{*YES\NO*}

Indicates whether or not to prompt for the creation of the Emergency Setup Disk. If this switch is present, it must be followed by either YES or NO. If it is not present, then the default value defined in the profile will be used. If no value is defined, then the user will be prompted for the creation of the Emergency Setup Disk.

/M:*COMPUTERNAME*

Overrides the default computername (if any) defined in the profile. This is the unique computername used to identify the computer in the network. This name should be in all capital letters.

/N:*domainname*

Specifies the domain to join during setup. The computer must already have an account in the domain. Otherwise, the Join Domain dialog box will appear during setup.

/O:*orgname*

Overrides the default registered organization name (if any) defined in the profile. The organization name must be enclosed in double quotes only if there is a space in the name.

/U:*username*

Overrides the default registered owner name (if any) defined in the profile. The user name must be enclosed in double quotes only if there is a space in the user name.

/Z:*timezone*

Sets the time zone used by the computer. This does not set the system time. The system time is read from the CMOS clock during setup. The time zone name must be enclosed in double quotes. For a list of valid entries see the README.TXT file that accompanies the CPS utility files.

For example, if drive letter Z has been assigned to the distribution share, the following command could be used:

```
winntp /s:z:\
```

With this command, **winntp** copies all files from the Computer Profile directory and installs them into the appropriate directories on the target computer, then asks the user to restart the computer, after which the graphical portion of Windows NT Setup runs to complete the installation.

Creating a Network Installation Startup Disk

▸ **To create a network installation startup disk**

1. Make sure that the disk has been formatted using the MS-DOS operating system. Use a high-density system disk that fits the target computer's drive A. Insert the network installation startup disk in drive A of a computer running the MS-DOS operating system version X.XX or later.
2. At the MS-DOS command prompt, type **sys a:**, and then press ENTER to copy the hidden MS-DOS system files (IO.SYS and MSDOS.SYS) and the MS-DOS command interpreter (COMMAND.COM) to the network installation startup disk.
3. Using the Network Client Administrator Utility on a Windows NT Server computer, follow the directions displayed on the screen to create a network installation startup disk.
4. When prompted, insert the formatted, high-density, system disk.
5. Continue following the directions displayed on the screen.

Note At this point, you may want to add an answer file to the startup disk or to the shared directory on the distribution server.

▸ To use the network installation startup disk

1. Insert the network installation startup disk in drive A: of the target computer.
2. Reboot the target computer, and then follow the directions displayed on the screen. When the computer restarts, it will automatically run the Network Client installation program.
3. A prompt appears asking you for a username and password. Supply a username and password for an account with permission to connect to the directory on the Windows NT Server computer where Windows NT Workstation Setup files are stored. When the computer displays a message about creating a password-list file, type **n** and then press ENTER.
4. The client must make a connection to the shared directory on the Windows NT Server computer. If the computer displays an error message saying that "the specified shared directory cannot be found," check that the Windows NT Server computer is indeed sharing the directory.
5. If the computer displays an error message about lack of memory, modify the CONFIG.SYS file on the network installation startup disk to use extended memory. For example, EMM386.EXE and HIMEM.SYS provide extended memory for MS-DOS 5.0 and later. If you do not have extended memory, use the NetBEUI or IPX protocols, because they use less memory.
6. The work installation startup disk was configured with the default settings for the network adapter. Please verify that the default settings are correct for your network adapter and modify them, if necessary. (The settings are in the A:\NET\PROTOCOL.INI file.)
7. After connecting to the shared directory that contains the setup files, run the **winnt** setup program, using the **\u** option for Unattended Setup and the **\b** option for a setup not using floppy disks. If you are using an unattended answer file, follow the instructions for command syntax as described earlier in this chapter.
8. Follow the instructions of the Setup program.

Systems Management Server

Systems Management Server (SMS) can be used to deliver any job that can run on the target computer—including an over-the-network setup or upgrade.

If you plan to use SMS in your actual deployment, you should also use it in the lab test of the deployment. See Chapter 6, "Unattended Installations and Upgrades," for details.

Reviewing the Results

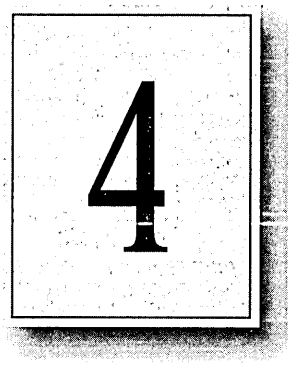
When the lab tests have been completed, the deployment team needs to review the results. What adjustments had to be made? How could the process be streamlined? Are your plans for preparing the users adequate, or should additional information be provided to them before deployment?

Using the results of the lab tests, you might choose to alter your high-level plan. Test the changes in the lab to make sure they don't need further adjustment. In some cases, this can mean reconfiguring the lab so that it again simulates your organization's predeployment environment, and repeating the test deployment. It is far more time effective to repeat the tests in the lab than to disrupt part of your production environment with a pilot test that you are not quite ready to implement.

When all is well, continue to the next step, planning and implementing the pilot rollout.

CHAPTER 4

Pilot Rollout



Once you have tested and refined your rollout procedures in the lab, you are ready to test them in the production environment. The pilot rollout introduces the variables of user reactions and the activity in the production network, but on a limited scale—usually 15 to 50 users. Based on this new information, you might make adjustments to your plan for the final rollout. If you make extensive changes to the final rollout plan, additional pilot rollouts might be in order.

It's important to make the pilot rollout as successful as possible, because it sets the tone for the rest of the deployment process. If pilot users are satisfied, their enthusiasm can influence others to cooperate, which in turn helps the rest of the process to move smoothly.

Planning the Pilot Rollout

This phase involves three major efforts: automating the installation, documenting the logistics of the pilot installation, and preparing the user training plan. These efforts are a combination of planning and lab-testing work.

Installing the Source Files for Setup

You need to designate a network server that will be used as the source file directory for installing Windows NT Workstation over the network.

Automating the Installation

Automating the installation is a key step in reducing the cost of migration. By creating an unattended answer file with predetermined answers for installation questions, the installation process can run from start to finish without user intervention. It is also possible to "push" the installation from the server, so that you can install Windows NT Workstation on an individual personal computer without ever touching the computer. For multiple installations on identical computers, the Computer Profile Setup utility makes it easy to migrate to the Windows NT Workstation operating system. This automation work is done in the lab before conducting the pilot rollout.

For more information, see "Using Deployment Utilities" in Chapter 3, "Lab Tests."

Depending on the common network configuration at your site, you might determine that you need to remove a line from one or more configuration files as a global procedure before starting Windows NT Workstation Setup. For example, you might want to use a protected-mode protocol, such as Microsoft TCP/IP, during Setup instead of the real-mode version of TCP/IP currently used on the target computers. In addition, users might be running certain terminate-and-stay-resident (TSR) programs or applications that should be closed before running Windows NT Workstation Setup. In these cases, you can modify NETDET.INI on NetWare networks. On other networks, including Microsoft networks, modify the [Install] section of MSBATCH.INF to automate these changes.

In addition, you might want to manually add other files to the shared directory on the server, such as custom bitmaps for screens or a predefined WKGRP.INI file for workgroup organization, so that client computers are fully configured when Windows NT Workstation is installed.

Creating a push installation process involves doing some final work on the server, such as editing the login script for the user, or sending a link in electronic mail to a batch file that runs Windows NT Workstation Setup, so that the user only needs to log on or double-click an icon to start the installation. System management software, such as Microsoft Systems Management Server, can also be used to start the installation centrally. If you plan to use system management software in automating the installation, make sure this has been acquired and tested.

For more information, see Chapter 6, "Unattended Installations and Upgrades."

Documenting Rollout Logistics

This task involves determining the timing and the process for pilot installation and choosing the pilot user group.

Although it is a test, the first pilot rollout sets the tone for and presents an example of the final rollout, so it is important to be completely prepared with all aspects of the rollout. This requires that you determine the time it will take for installation, the personnel and tools needed to facilitate the process, and the overall schedule.

Start by identifying the target computers and their location. Then use the following list as the basis of your checklist for rollout logistics:

- Has a verified backup been performed for each of the target computers?
- Have passwords been reset for CMOS, the network, and applications?
- Have virus checking and disk defragmentation been performed?
- How many systems will be installed per day?

  Start with a conservative estimate and then increase or decrease the number, based on your experiences with the initial installations.
- At what time of day should the installations occur?

  You might want to schedule installations to occur on weekdays after normal business hours or on weekends.
- Who are the pilot users?

  Choose a pilot user group or department that is willing and able to accommodate the rollout. This group, ranging from 15 to 50 persons, should be representative of your overall user base. Try not to select a department that is attempting to meet a schedule deadline during the rollout, or a group that is traditionally slow in adopting new technology.
- What is the schedule for pilot installations?

  When determining the installation time for the pilot rollout, base the projections on how long it takes for installation of an individual computer; remember to schedule the downtime for each user.
- Who will participate in the installations?

  In addition to the Installation team members, be sure to assign a system administrator with full rights on the server, including the right to administer mail or database server passwords.
- Is the deployment methodology as automated as possible?

As you develop the checklist of logistics, consider your goals for the pilot rollout and the factors that define its success. For example, you might set a percentage for successful upgrades or for automated installations that, if achieved, would indicate that the rollout had been successful. Document these goals and criteria, so that teams can monitor performance against them during the rollout.

Developing User Training

The first steps in developing a training plan are to acquire a training lab, set up computers in the lab, and appoint a team member as instructor. (If in-house resources are not available, use a vendor to develop and conduct the training.) The instructor will be responsible for creating and testing the training program.

There are a number of training approaches and a variety of tools you can use. A recommended approach is to divide the training into sessions corresponding to three distinct topics: The Basics, Corporate-Specific Applications, and Customization.

The session entitled "The Basics" includes the top 10 functions any user needs to know to accomplish daily work.

Schedule training sessions of no more than 30 minutes each; in each session, users should receive information that is *just enough* to be productive using Windows NT Workstation.

The Corporate-Specific Applications session varies by the environment and the types of applications run on the network. This session should focus on the top 5 to 10 functions that will change because of the upgrade to Windows NT Workstation.

The Customization session is intended for more experienced users. The purpose of this session is to provide information and guidance that will help these users learn on their own after the training, and teach them how to work more productively with Windows NT Workstation. After creating and testing the program, schedule training sessions to occur immediately before the rollout so that the instruction is *just in time,* ensuring that users retain most of what they learn by putting it to use right away.

Developing the Support Plan

Similar to the training plan, the support plan must be ready to go online the first day you begin performing Windows NT Workstation installations. Because the quality of support that's available during the pilot rollout will be seen as an indicator of the quality of the rollout as a whole, it is important that you plan carefully to make sure effective support is available.

Staff the Support team for your pilot rollout with some of your best technicians dedicated solely to the pilot group for the first few weeks. The assigned technicians should carry pagers or be available by phone at all times to give immediate assistance to users.

Notifying Users of the Rollout

Another step at this stage is informing users about the pilot rollout plan. You can use a videotape presentation, an interoffice memo, or a company meeting as the means for communicating with users about the rollout. Regardless of the form used, the message must explain to users the benefits of moving to Windows NT Workstation and describe the overall plan and process by which each group or department will make the move. This makes it easier for your users to plan for and accept the migration to Windows NT Workstation as part of their schedules.

Conducting the Pilot Rollout

This phase consists of simulating the final installation process, testing the capabilities and performance of the system, surveying user feedback, and making adjustments as needed.

Repeat this pilot rollout process for 32-bit applications.

Simulating the Installation Process

The schedule for the pilot rollout should simulate—on a smaller scale—the schedule for the final rollout. As you conduct the pilot rollout, you might find that certain tasks take more or less time than expected, that some tasks need to be added, or that some tasks can be left out. Modify the pilot rollout schedule to account for such changes, and use the pilot schedule for projecting the final rollout timetable.

Testing Windows NT Workstation Performance and Capabilities

In addition to the technicians responsible for conducting the pilot installation, extra technicians should be assigned to measure, observe, and test the installation. By tracking the time per installation, handling problems that arise, and identifying areas for improvement or automation, these individuals help ensure the success of both the pilot and final rollouts by making the installation more efficient.

In addition, after Windows NT Workstation is installed, these technicians test system capabilities, such as remote administration, for proper operation and monitor the client computers for performance, stability, and functionality, highlighting any inconsistencies with the lab configuration.

Surveying Users for Feedback

The final part of the pilot rollout involves surveying the users to gauge their satisfaction and proficiency with the new installation and to evaluate the level of training and support provided. Test users' proficiency by having them perform a few common tasks or use several of the new features in Windows NT Workstation—for example, have these users register their survey results on the server.

When collected, combine the survey results with the ideas for improvements identified during the pilot rollout. Use this information to prepare a checklist of open issues that must be resolved before the final rollout. Then assign team members to take the actions necessary for solving problems or making improvements. Indicate on the checklist how and when each item was resolved, adjusting the deployment plan if appropriate.

CHAPTER 5

Final Rollout

After analyzing the results of the pilot rollout(s), you are ready to finalize the plans for the full-scale rollout. Your team and your users can then enjoy a well-planned and well-organized full deployment of Windows NT Workstation.

Finalizing the Rollout Plan

The final rollout plan is an extension of the pilot planning process, with the added steps of documenting, budgeting for, and carrying out the final logistics. As you perform these steps, you should also update the guidelines governing network and computer use in your company, and create a template for a central database that tracks specific configurations and uses of each network computer.

Completing the Rollout Logistics and Budget

As you prepare for final rollout, estimate the length and scope of the overall installation process. Also plan for all tools needed to complete the process within the stated timeframe. If necessary, propose a formal budget for the company-wide implementation, and present it to management for approval. Your budget should include the costs for personnel and resources, such as system management software.

After obtaining any necessary approval, purchase the resources required to facilitate the installation. If you need additional staff, be sure to hire experienced and qualified individuals for the team, and train them extensively before getting started.

Complete your training, communication, and staffing plans for the final rollout at this time.

Updating the Policies and Guidelines

Before final rollout, update all company policies regarding the use of the network and computers by employees. Make sure to cover items such as password length and expiration requirements, and the level of approval needed to obtain remote dial-up privileges.

In addition, update the corporate standards lists for hardware and software usage; use this is a reference for bringing all computers into compliance during the rollout process. Use the *Hardware Compatibility List* to update your company's hardware requirements.

Creating a Template for the Rollout Database

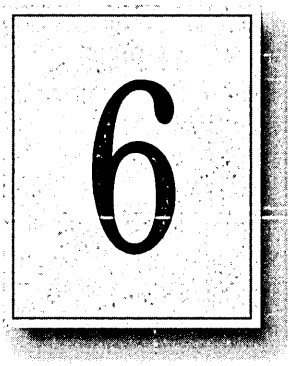
A template is used to create a central database for monitoring the progress of the rollout and to document any areas requiring further action. During preparations for the final rollout, create the template, using appropriate database management software. Complete the template with configuration information for every computer and user in the company, and place the template on the server. Then, during company-wide installation, the Installation team fills in the template for each computer and user, indicating whether any additional upgrading is needed. The team can then use the template to track open items following the rollout and to measure actual progress against original objectives.

Rolling Out Windows NT Workstation

Following weeks of planning, organization, testing, communication, and training, the deployment teams and your organization as a whole should be ready for full-scale rollout of Windows NT Workstation. The extensive preparation for this event might make deployment seem almost routine for the teams involved; however, that's exactly the kind of uncomplicated rollout a systems administrator dreams of. And, soon after the installations, users might not know how they got their work done without Windows NT Workstation. If this happens in your company, then you know your rollout has been a success!

CHAPTER 6

Unattended Installations and Upgrades



Microsoft Systems Management Server (SMS) is used by administrators to centrally manage hardware and software on the network. SMS automates the following key tasks:

- Maintains an inventory of the hardware, software, and configuration of computers on the network.
- Distributes, installs, and updates software and files.
- Lets administrators view diagnostic information for remote client computers and, with the user's permission, take direct control of those computers.
- Lets administrators monitor network data flow.

Once installed, SMS can simplify the detailed inventory that is a prerequisite of a successful rollout. For example, SMS *packages*, which contain all the files needed for an installation or upgrade, can be created, and then sent to distribution servers. The information in the packages can be used by SMS *jobs* to run specific commands, including the command to install Windows NT Workstation on target computers. Queries to the SMS database can be used to determine which computers on the network are targeted for a specific Setup command (with specific parameters and tailored installation files). Setup is then run over the network from the client computer.

Before using SMS, you should be familiar with the deployment utilities described in Chapter 3, "Lab Tests."

For more information on SMS, see the *Microsoft Systems Management Server Administrator's Guide*.

Using Systems Management Server for Deployment

Systems Management Server (SMS) is best utilized on an existing client-server network.

SMS gives administrators the ability to query the SMS database to produce an inventory of the computers that are capable of running Windows NT Workstation. SMS also enables remote administration of utilities to prepare each computer for the installation or upgrade. Use SMS to create a package containing the files and command lines (includingoptions) to install specific configurations of Windows NT Workstation. Finally, create an SMS job to execute commands in the package on the workstations that meet the criteria of a specific query. Once the job is sent, you can monitor and evaluate the job status and results.

If you are upgrading Windows NT Workstation on even as few as 100 computers, SMS will give you significant time savings over manual installation. The larger your organization, the greater the time savings. Once SMS is in place, these savings are repeated every time you need to inventory the hardware and software in use, and every time you need to deploy an upgrade or new software.

At this point, you have defined your preferred client configurations and set up a local area network (LAN) in the lab that simulates your production LAN. If you have not already installed SMS, do so at this time. SMS requires a Windows NT file system (NTFS) partition and at least 100 megabytes (MB) of free disk space. It is recommended that you allow at least 1 gigabyte (GB) of disk space for the deployment of Windows NT Workstation. If you did not choose to have Windows NT Setup convert a partition to NTFS, convert it before beginning the SMS Setup program. For more information on setting up SMS, refer to your SMS documentation.

Create SMS queries that specify the minimum requirements you want a computer to meet before you install Windows NT Workstation. These requirements can include free disk space, installed memory, and CPU (for example, Pentium, 486). You might also choose to specify the department the computer belongs to, or currently installed software, depending on your organization.

Next, use SMS and the queries to perform a test deployment in the lab. This task involves the following steps, which are discussed in greater detail in the remainder of this chapter:

1. Copy the Windows NT Workstation Setup files to the distribution server in the test lab, making sure the directory they reside in is a shared directory.
2. Inventory the lab using SMS by running queries to determine capable computers.
3. Create Machine Groups for these computers.
4. Create an SMS package to install Windows NT Workstation on target computers by using Setup scripts and .PDF files with SMS.
5. Create an SMS Job to run the package.
6. Monitor the SMS job status.
7. Evaluate distribution results.

Inventorying the Test Lab

Whatever installation method you use, an up-to-date inventory of the target computers is essential to a smooth deployment. Once SMS is installed, it can be used to perform this task quickly and accurately. By using SMS to inventory the lab, you can test your SMS queries and become proficient at performing and interpreting inventories with SMS. You can then use the same procedures to perform inventories for the pilot and final rollouts.

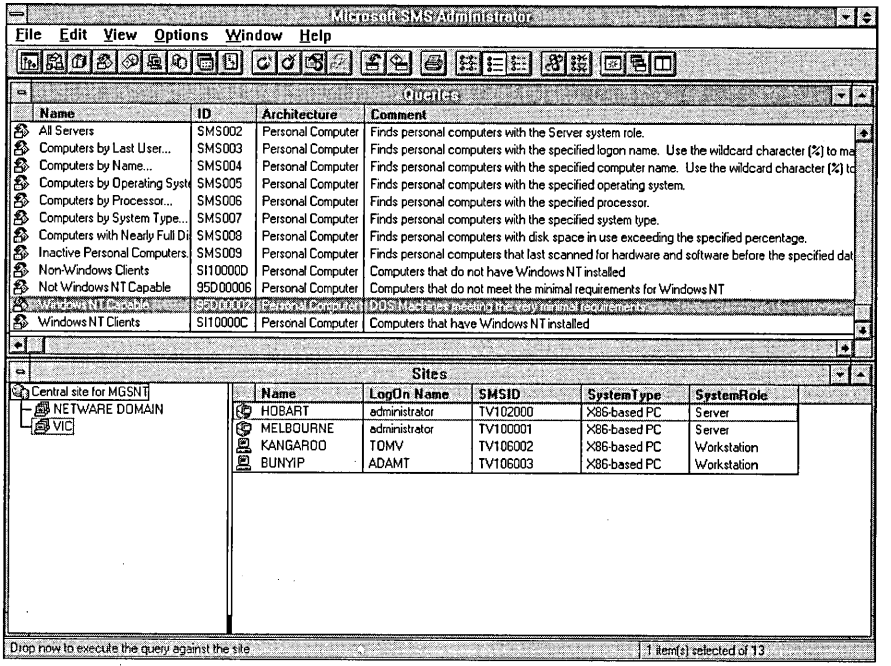
Creating and Running Queries

You will need to create and run queries to determine which of the computers listed in your SMS database are to have Windows NT Workstation installed. You might also want to determine which computers are not suitable for installation of Windows NT Workstation, so that you can schedule upgrades or help the users free the necessary disk space.

After you run the queries, you can create a Machine Group with the results. This Machine Group will be your target group for the jobs that run a virus checking program and install Windows NT Workstation.

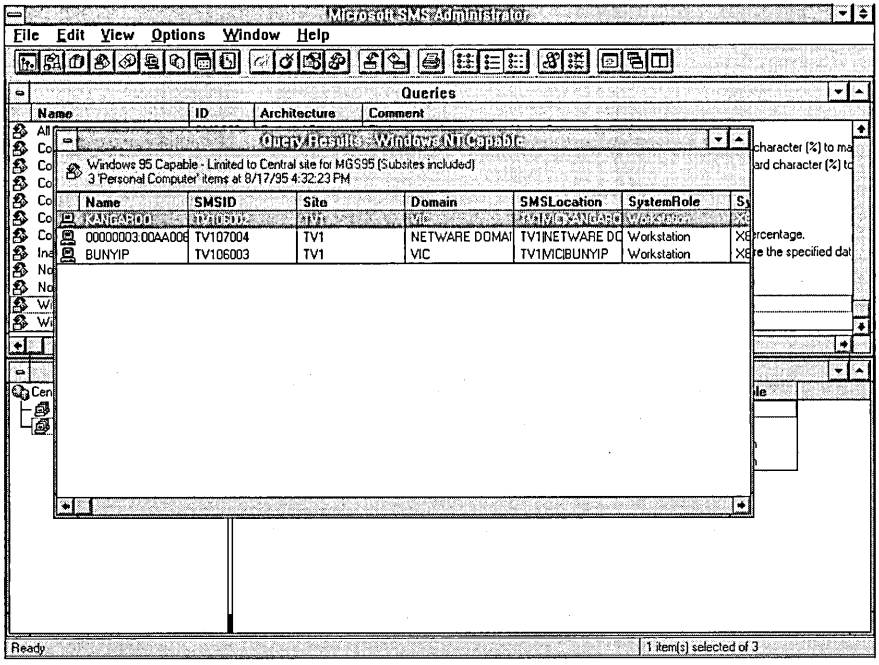
▸ To run the query

1. From the File menu of the SMS Administrator window, choose Execute Query. The Queries dialog box appears.



2. Drag the query to the site (in the Sites window) on which you want it to run.

When you run the query, you should get a result something like that shown on the following screen:



Using Setup Scripts

Windows NT Workstation allows the use of special information files, (.INF files), that can specify all possible custom settings. The format and options for these files are described in Chapter 3, "Customizing Windows NT Setup," of the *Windows NT Resource Guide*.

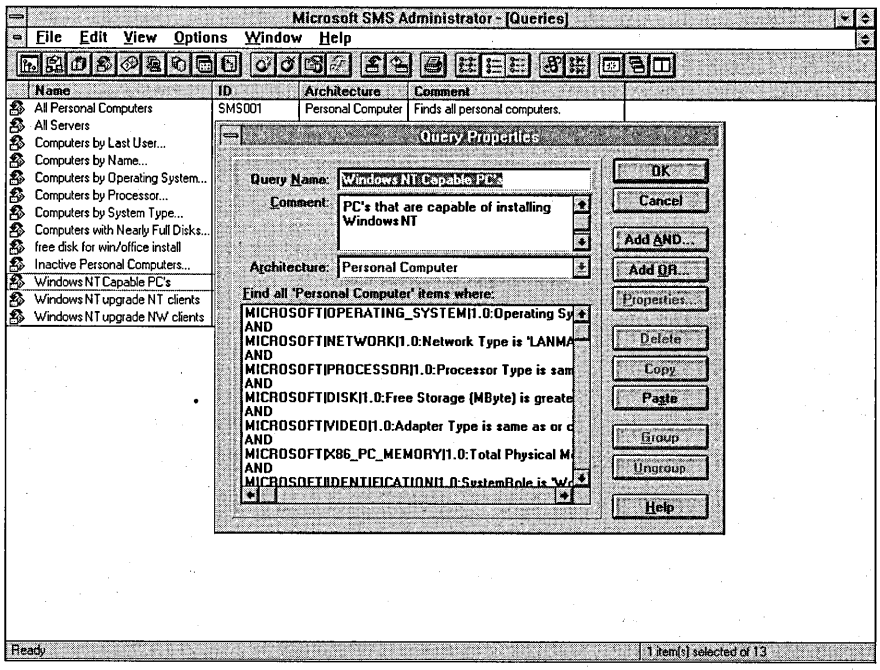
SMS is able to use these information files for automated installations of the software. The vehicle for this is the Package Definition File, or .PDF file. A PDF file is an ASCII file that specifies setup programs, installation options, and execution command lines for the software you will install. If the .PDF file contains a reference to an .INF file, then the .INF file is made available to the target computers when a the package is delivered through an SMS job. Several .INF files can be included in a single package. If the .PDF file has multiple options for installing Windows NT Workstation on computers currently running MS-DOS, Windows 3.*x*, or Windows 95, the queries need to target each platform individually.

You can create modified versions of master queries, and save them with unique names. For example, if you have both 486- and Pentium-based computers that meet the criteria of your "Windows NT Capable" query, but only want to target the 486-based computers, you can edit the query to specify this restriction. You can find detailed information on queries in Chapter 7, "Queries," of the *Systems Management Server Administrator's Guide*. The following section is a brief description of the steps in editing a master query.

▸ To edit a query

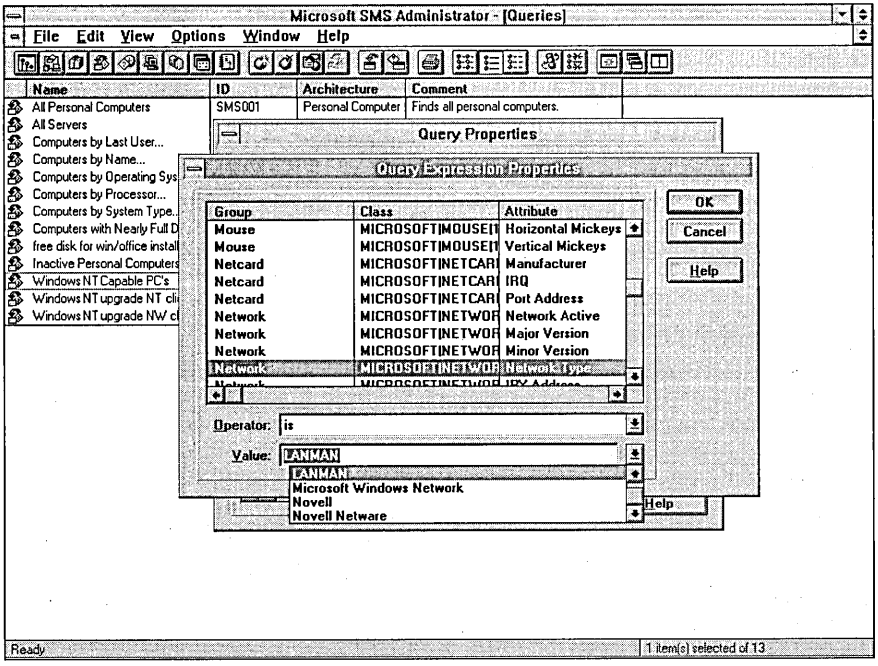
1. Open the Query window in SMS Administrator.
2. Select the query (in this case the "Windows NT Capable" query), and then select Edit.

   The Query Properties dialog appears.

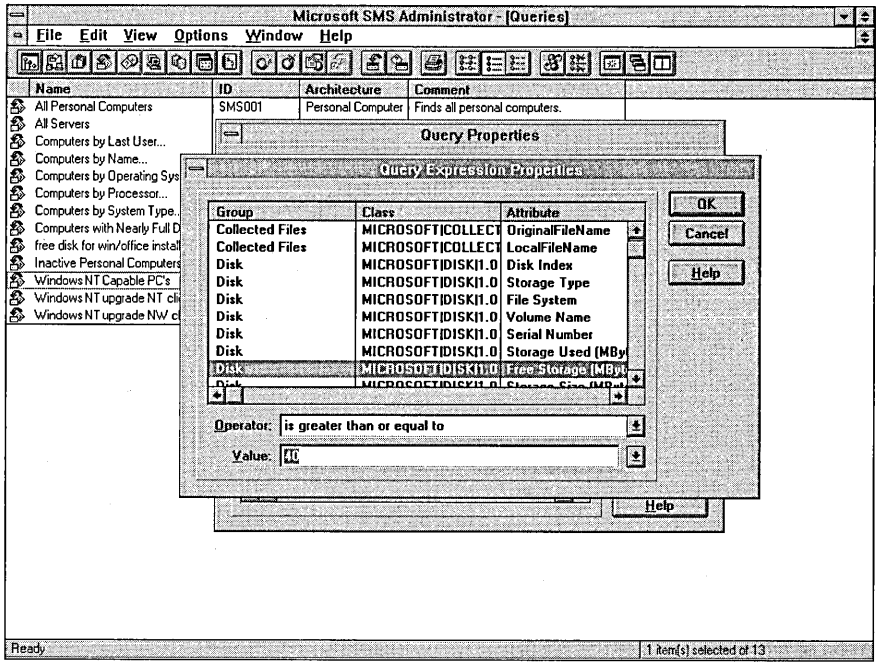


3. Highlight a line you want to edit, and then select Properties.

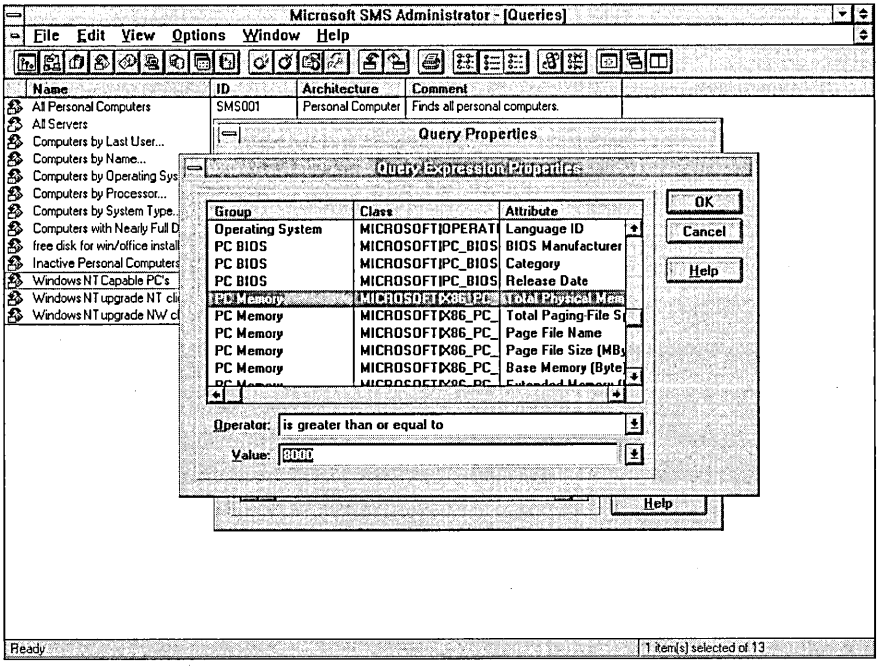
   The Query Expression Properties dialog box appears. The choice of operators and values you have in this dialog box depends on the line you chose to edit. For example, if you choose the line describing the network type, the following dialog box appears:



If you choose the line describing the disk storage, the following dialog box appears:



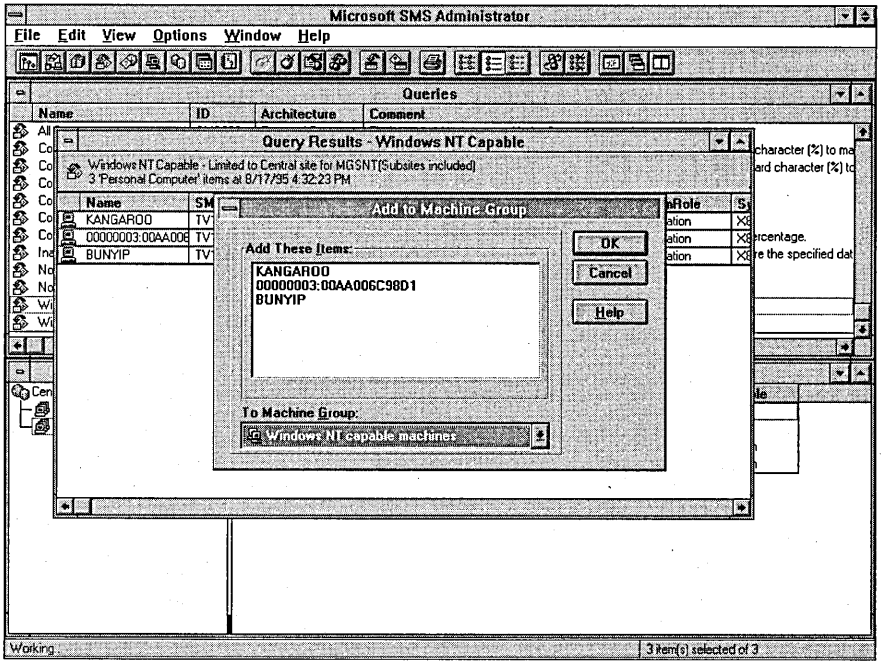
If you choose the line describing the system role, the following dialog box appears:



4. Select the operator and value you want to select from the drop-down boxes at the bottom of the dialog box, and then choose OK.
5. Repeat steps 3 and 4 until the query is as you want it.
6. Choose OK to close the Query Properties dialog box.

Once you have altered the query to suit your needs, you are ready to run it. The query is run against the contents of the SMS database, which has been filled by SMS as individual computers logged on.

By running the modified version of the query and dragging or pasting the query results into a Machine Group window, you can create a group of all the computers that you want to upgrade to Windows NT Workstation. This group can then be used in subsequent steps. The following screen shows a group created from the results of the query:



You might want to create several different machine groups to deploy different configurations. Use variations of the query to create different groups.

All the queries should be tested before proceeding further, to make sure you are selecting for the computers you really want to target.

See Chapter 7, "Queries," of the *Systems Management Server Administrator's Guide* for more information on creating and executing queries. See also Chapter 14, "Machine Groups and Site Groups," of the *Systems Management Server Administrator's Guide* for more information on machine groups.

Creating a Package to Install Windows NT Workstation

To install Windows NT Workstation on the target computers, the **setup** command that installs the operating system must be run on each workstation. An SMS Distribution Package can be used to run this command locally on each targeted workstation.

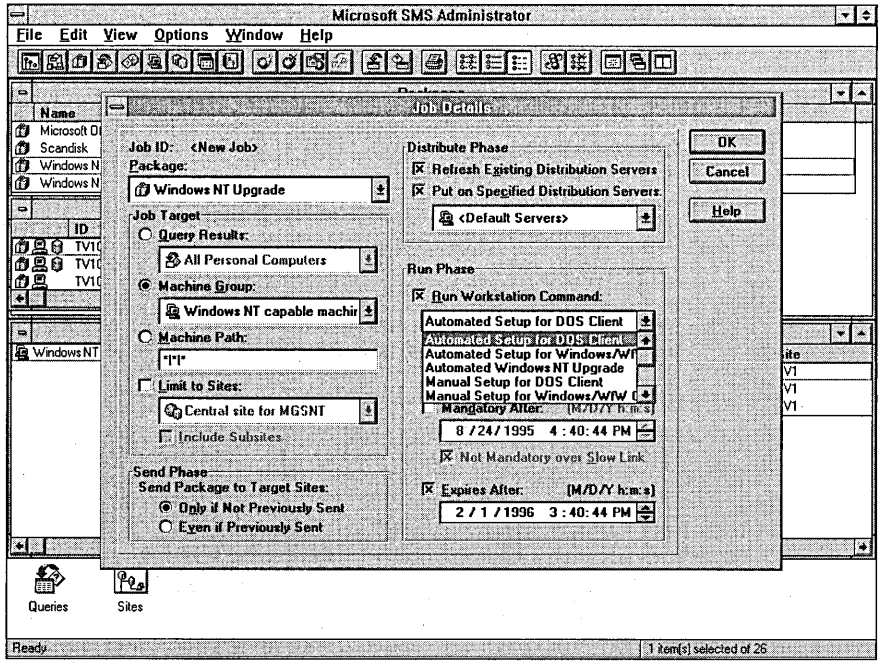
This package, called the Windows NT Workstation Setup package, is created in the same way as the package used to run ScanDisk. Make sure this package points to the proper disk location for the Windows NT Workstation source code.

Each package is completely self-contained: it has all the files needed for the task or tasks it is designed to do. Also, a single package can contain several different sets of helper files, .EXEs, and .INFs. You specify which command to use when you use the package to create a job.

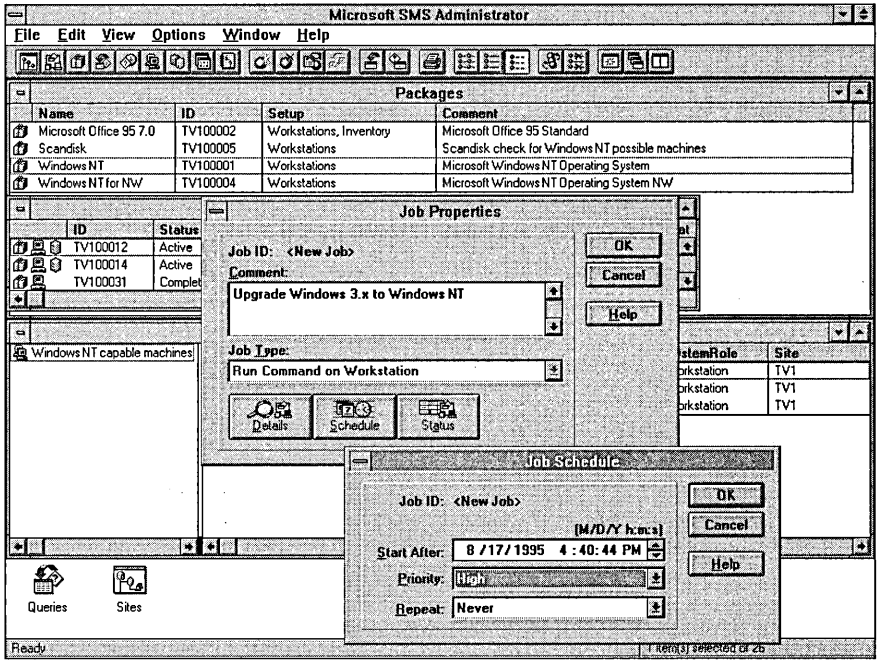
For information on creating packages, see Chapter 10, "Packages," of the *Systems Management Server Administrator's Guide*.

Creating a Job to Execute the Package

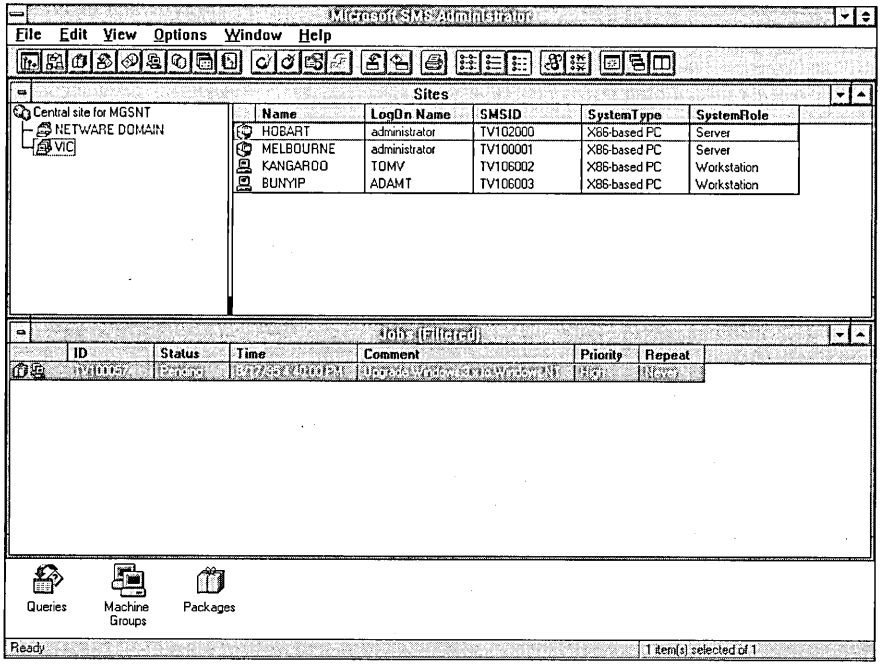
For the command in the Windows NT Workstation Setup package to be run on the target computers, it must be made available in a Run Command on Workstation job. The job is created by dragging the package to the Machine Group on the site and filling in the Job Details dialog box that appears as a result. Use the following screen as a model for the job details.



You might also choose to take advantage of the job scheduling and configuration options, as shown in the following screen.



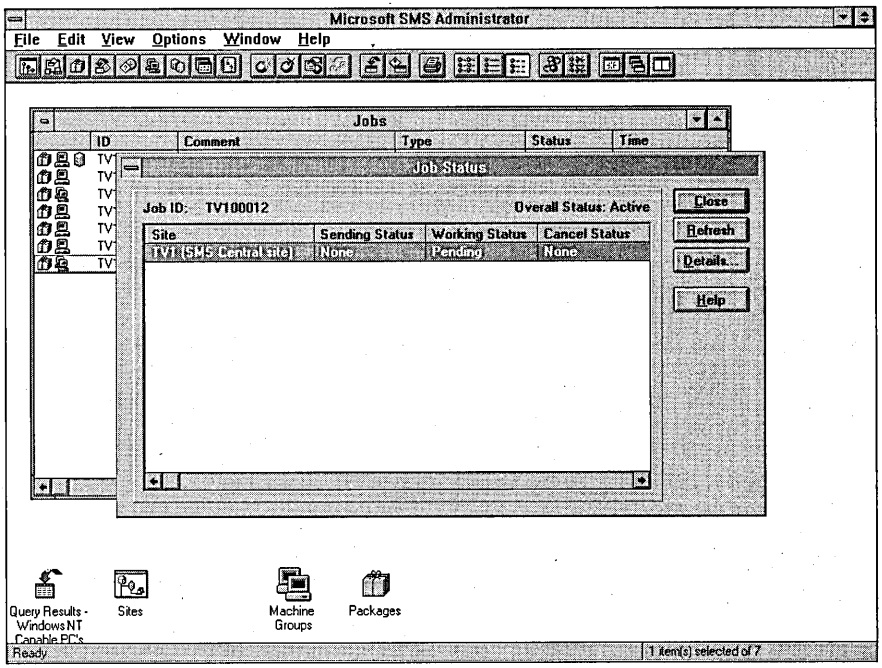
After you have set up the job, it should appear in the Jobs window as a pending job, as shown in the following screen.



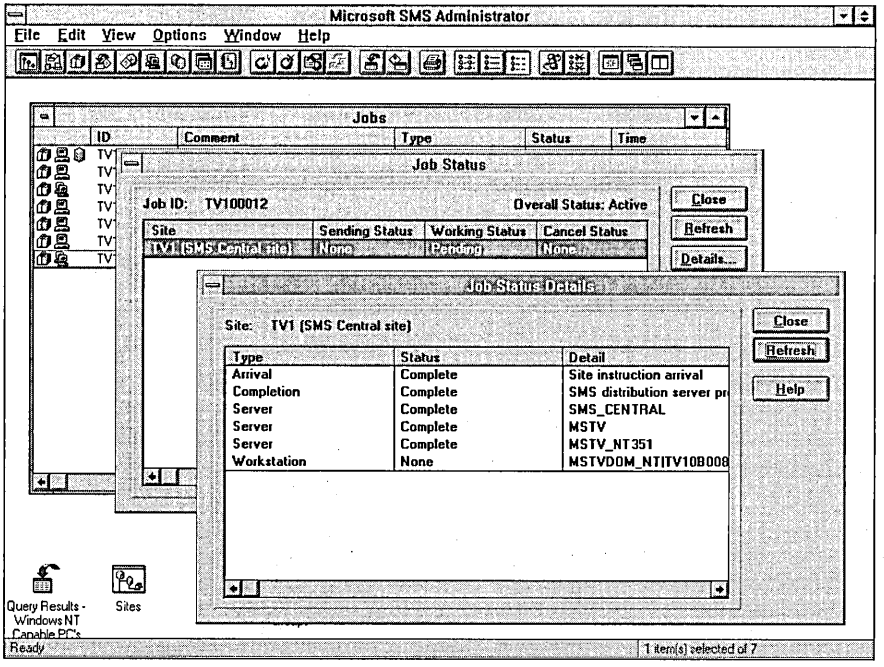
While the job is pending, you can make modifications to it. After the job status changes to Active, you can no longer modify the job.

Monitoring the SMS Job Status

Monitor the status of the job by selecting that job in the Job Properties dialog box, choosing the Status button, and viewing the Sending and Working columns in the Job Status dialog box. The following screen shows the Job Status dialog box.



After the job has had a chance to propagate, you can check on details of the job. In the small network you have set up in the lab, this could take as little as 30 minutes. In your production network, you'll need to allow more time. To view the details of the job, select Details from the Job Status dialog box. The Job Status Details dialog box appears.



When the Status column changes to Complete, the job has been distributed. The total duration of the job depends on a number of factors, including parameters set for the job and the behavior of your users (who must log on to accept the SMS Package).

For more information on jobs, see Chapter 11, "Jobs," in the *Systems Management Server Administrator's Guide.*

Evaluating Distribution Results

When the job status is marked Complete, it does not necessarily indicate successful deployment of the operating system. You must perform additional analysis to determine results.

First, make sure the SMS Workstation Inventory update has been performed for all computers in the target group of workstations. Then create SMS queries to determine which installations have been successful, and run them against the same targeted group of workstations to determine results. (The Machine Groups you made earlier from query results can be used to define your targets now.)

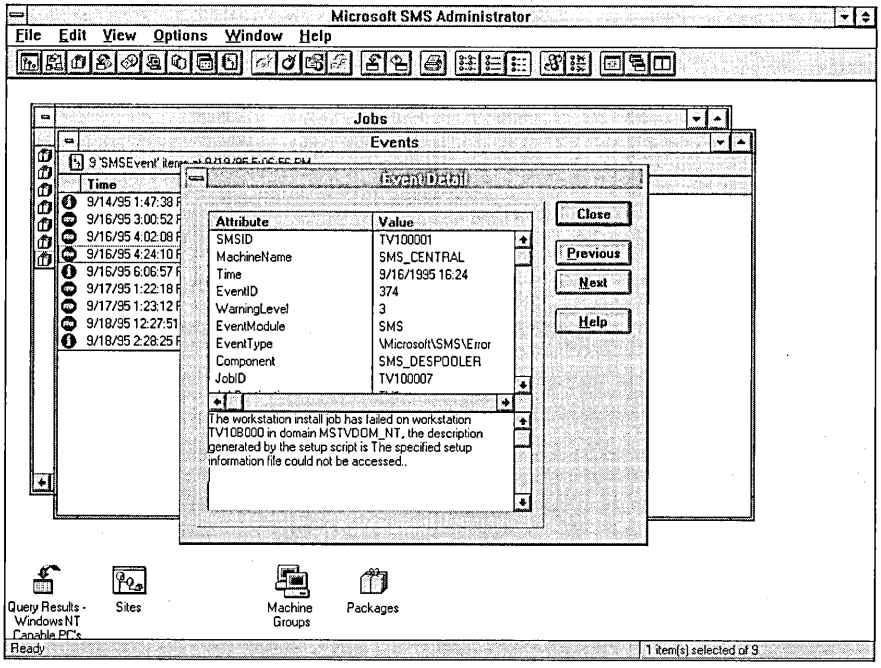
You will probably want to examine the computers that were not successfully upgraded on a case-by-case basis. Check, for example, whether there were changes in available disk space or in the availability of some other resource between the time the computer was added to the target group and the time the upgrade was performed.

The computers that were successfully upgraded should be tested to make sure the configuration you specified works as expected on these computers.

Job Events

You can check details of job events to troubleshoot problems. SMS system events are reported to both the Windows NT event log and the SMS Event database. For more information, see Chapter 11, "Jobs," and Chapter 18, "Events," of the *Systems Management Server Administrator's Guide.*

For example, if a job has failed, perhaps because the .INF file was not available where it was expected, this failure would be reported in the SMS Events database and could be viewed in the Events window, as shown in the following screen.



To view the job events

1. From the SMS Administrator window, choose Jobs.

   The Jobs window appears.
2. Select the job.
3. From the File menu, select Properties.

   The Job Properties dialog box appears.
4. Select Status.

   The Job Status dialog box appears.
5. Choose Details.

Post-Deployment Queries

You can now create SMS queries to evaluate the results after SMS has been used to deploy the operating system or systems. These might simply query for the current operating system, or might include other criteria, such as remaining free disk space. Execute these queries to determine which workstations have the operating system or systems successfully installed, and which do not.

Additional Help

If you need to contact Microsoft Product Support for help with the deployment, be sure you have the copies of the following items for the time the problem occurred:

- A Registry dump
- The trace logs:
  - ROOT\SMS\LOGS*.LOG
  - ROOT\SCMAN.LOG
  - *.LO_ files
- The package definition (.PDF) file or files used in your deployment of Windows NT Workstation
- The .INF file or files used in your deployment of Windows NT Workstation
- The SETUPLOG.TXT files for the computers on which installation failed

Testing the Installation

When you have assured yourself that all has gone well with the job, test the computers in your lab network to make sure that the configuration you have installed will serve your organization as expected.

- Connect to and browse the network.
- Set up a printer and test printing to local and network printers.
- Open, run, and close applications on both the client computer and on the server.
- Shut down the computers.

Make sure to test all mission-critical applications for proper function. If you encounter problems, try removing related features from the proposed configuration as a solution. Document any changes made to the original configuration.

If the preferred client configuration works as expected, you might also want to conduct additional testing of the optional software features and components in Windows NT Workstation. Additional testing can help you determine whether you are optimally running the operating system. For this kind of testing, conduct side-

by-side evaluations on two computers, changing individual features on each one, to determine the following:

- Performance in terms of responsiveness and throughput
- Ease of use
- Stability
- Compatibility
- Functionality

When you have identified a configuration that performs well during testing, test the same configuration using other hardware from your company.

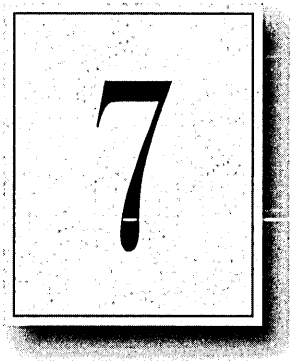
After thorough testing of the preferred client configuration, completely restore one of the test computers to the previous client configuration and document the process.

PART II

Part Two goes into the details for using the Windows NT File System (NTFS) and the changes introduced with version 3.51 of the Windows NT operating system. Building on that foundation of file-system knowledge, it then goes into how to create and maintain a fault-tolerant system. A revised chapter on the printing process and troubleshooting printing problems rounds out this part on using the Windows NT operating system.

CHAPTER 7

Windows NT File System



This chapter contains additional information beyond that provided in Chapter 5, "Windows NT File Systems and Advanced Disk Management," of the *Windows NT Resource Guide*. Chapter 5 contains more details about the organization of Windows NT File System (NTFS) and File Allocation Table (FAT) partitions and directories; that information is not repeated here.

The *Update Information for Version 3.51* manuals of the Windows NT Workstation and Windows NT Server documentation contain identical information about file compression on NTFS partitions and moving and copying files within NTFS partitions. The information from those books has been included in this chapter, along with new material about these topics.

Note The Microsoft Windows NT operating system is the only operating system that implements NTFS. You can't access NTFS partitions when you dual boot any other operating system. However, a program designed to run under other operating systems, such as MS-DOS, can access NTFS files when running under Windows NT Workstation or Windows NT Server.

Be wary of using disk utilities that were not designed for computers running Windows NT Workstation or Windows NT Server. These utilities will not recognize NTFS partitions and, on FAT partitions, most of them do not understand the method that the Windows NT platform uses to store long filenames.

You can find additional NTFS information in the following sources:

- The *Comprehensive Index* manual of the Windows NT Server documentation set lists all of the NTFS topics in the documentation set. Each manual in the documentation set also has its own index.
- The *Installation Guide* and the *System Guide* manuals of the Windows NT Workstation documentation set list all of the NTFS topics in the respective manuals.
- *Inside the Windows NT File System*, by Helen Custer (Microsoft Press 1994, ISBN 1-55615-660-X) documents the NTFS file system design.

Comparing NTFS and FAT

This section contains overview information about the two file systems supported by the Windows NT platform, FAT and NTFS, and summarizes the advantages of each. Chapter 5, "Windows NT File Systems and Advanced Disk Management," of the *Windows NT Resource Guide* provides more details.

Note Windows NT Server 3.51 and Windows NT Workstation 3.51 no longer support the High Performance File System (HPFS), other than booting from an HPFS partition.

Overview of FAT

The FAT file system is by far the more simplistic of the two file systems supported by the Windows NT platform. It is characterized by the file allocation table that resides at the beginning of the partition. To protect the partition, two copies of the table are kept, in case one becomes damaged. In addition, the tables and the root directory must be stored in a fixed location so that the system's boot files can be correctly located.

A partition formatted as FAT is allocated in clusters, whose sizes are determined by the size of the partition. The cluster number must fit in 16 bits and must be a power of two. Therefore, the cluster size for a 640 megabyte (MB) partition is 16 kilobytes (K).

When a file is created, an entry is created in the directory, and the first cluster for data is established. The entry either indicates that this is the last cluster of the file, or points to the next cluster.

Updating the file allocation table is very important, as well as time consuming. If the table is not regularly updated, it can lead to data loss. It is time consuming because the disk heads must be repositioned to the partition's logical track zero each time the table is updated.

There is no organization to the FAT directory structure, and files are given the first available location on the partition. In addition, FAT supports only read-only, hidden, system, and archive file attributes.

Overview of NTFS

Like the FAT file system, the NTFS file system uses clusters as the fundamental unit of disk allocation. The default cluster size depends on the partition size, and ranges from 512 bytes to 4K.

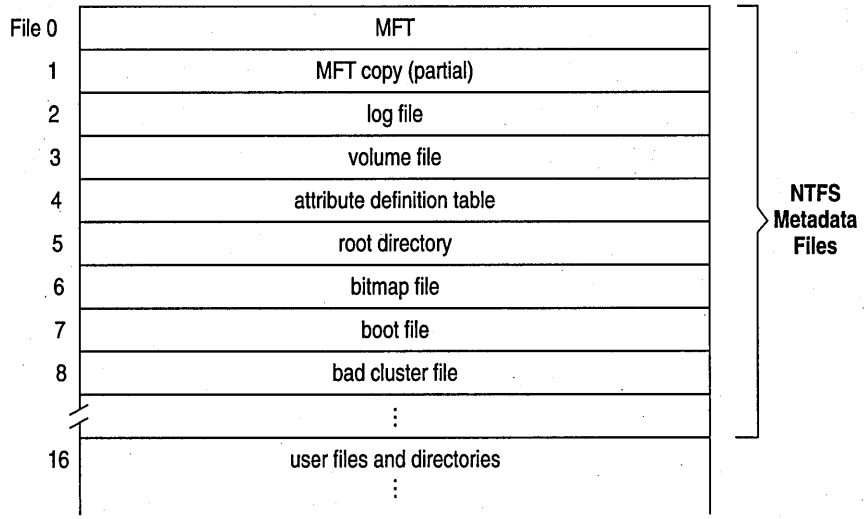
Everything on the NTFS partition is a file—a directory is just another file, but with a different set of attributes.

The NTFS file system maintains information about files and directories in the master file table (MFT), where each file and directory is a record. The MFT record size also depends upon the partition size, and can be 1K, 2K, or 4K.

When you format an NTFS partition, the format utility creates a set of files that contain the metadata used to implement the file system structure. The NTFS file system reserves the first 16 records in the MFT for the information about these metadata files. The only file on an NTFS partition that must be at a specific disk address is the boot file, and the information about this file is in one of these metadata records. The NTFS file system uses approximately 1 MB for the metadata files and the first 16 records in the MFT.

Formatting an NTFS partition also results in the creation of the log file, which the operating system uses to restore consistency to the NTFS partition in the event of a system crash. The log file size depends upon the partition size, and can be as large as 4 MB.

The following figure shows the structure of the master file table (MFT).



See *Inside the Windows NT File System* for more details about the NTFS file system and the metadata files.

Controlling Access to NTFS Files and Directories

The NTFS file system provides access controls to individual files and directories. However, users can perform certain actions to files or directories even if permissions are set on a file or directory to prevent access to users.

For example, if you have a directory (DIR1) containing a file (FILE1), and you grant Full Control to a user for the directory DIR1, but specify that the user have No Access to FILE1, the user will be able to delete FILE1. The user's Full Control rights in the directory allow the user to delete contents (or children) of the directory.

To prevent files from being deleted, you must set permissions on the file itself, and you must set permissions for the directory containing the file. Anyone who has Full Control in a directory will be able to delete files from the directory.

Similarly, anyone who has List, Read, or greater permissions in a directory will be able to view file properties on any file in the directory, even if they are prevented by file permissions from seeing the contents of the file.

Using Removable Drives and Floppy Disks

It is not possible to format a floppy disk with the NTFS file system; the Windows NT operating system formats all floppy disks with the FAT file system, because the additional files required for the NTFS file system use too much space on a floppy disk.

If you format a removable drive with the FAT file system, you can remove the drive while Windows NT Workstation and Windows NT Server is running.

When you format any removable drive using the NTFS file system, however, you must force the Windows NT operating system to unmount the drive before you can remove it. This task is necessary because the removable drive could be used, and modified, on another system. If the NTFS drive was allowed to be remounted without being closed, it is possible that the NTFS information stored in memory would no longer be accurate.

The steps used to unmount a removable drive include:

- Closing all open file handles, while ensuring that no new files are opened.
- Flushing all data, including user data and file system metadata, to the drive.
- Unregistering the drive, so that it is no longer possible to access it.

The tricky part has to do with ensuring that no program accesses the drive after NTFS has tried to flush the data, but before the partition is taken off line. Flushing works at shutdown, because shutdown terminates all processes that might want to write to the disk so there's nothing left running after the final flush.

File Manager uses the floppy drive icon for removable drives.

Using the Chkdsk Command

If you have a FAT or NTFS partition, keep the following points in mind when using the **chkdsk /f** command.

On an NTFS partition, *never* run the **chkdsk /f command** without first doing a backup. If there is a problem with any of the metadata files, the index entries for them will be deleted and you will not be able to reboot the system.

If you run the **chkdsk /f** command on a current partition, the following error message will be displayed.

```
Cannot lock the current drive
```

For example, if you are trying to run the **chkdsk /f** command on the D partition, make the C partition the current partition by typing **C:** and pressing ENTER before running the **chkdsk /f** command.

Before attempting to run the **chkdsk /f** command, make sure all files are closed. If the files are open, the following error message will be displayed.

```
Cannot lock the drive for single user.
```

If the Windows NT operating system is not installed on the partition on which you're trying to run the **chkdsk** command, close all applications that might have files open on the partition. If you have a page file on the partition, you need to move the page file by using the Virtual Memory option in Control Panel. You should then be able to run **chkdsk** with the **/f** option.

If the Windows NT operating system is installed on the partition on which you are trying to run **chkdsk /f**, then it will not be possible to fix errors without restarting Windows NT.

When **chkdsk /f** is unable to run, it prompts the user with a message similar to the following:

```
Chkdsk cannot run because the volume is in use by another process.
Would you like to schedule this volume to be checked the next time
the system reboots? (Y/N)
```

If you choose Y, **chkdsk /f** is executed the next time the operating system is started.

Lost Delayed-Write Data Error Message

This error message comes from the cache manager when it detects any kind of an error writing the cache, and it can occur with both NTFS and FAT partitions.

Some errors can be the result of uncontrollable circumstances, such as a server going down or a network connection being lost.

Some of the errors can be the result of user action. For instance, if the user uses File Manager to copy a file to a remote computer, the cache will probably not have been emptied before File Manager returns control to the user. If the user disconnects from the server before the write actually finishes, this error will occur.

Another user-created cause of this error is an invalid disk address. For example, if the user resizes an external drive array from 100 gigabytes to 80 gigabytes, Windows NT does not receive information about the change, and its internal information on the partition (stored in the registry) still indicates that there are 100 gigabytes available. This situation can cause data corruption, and is very time consuming to repair on these types of large, externally controlled partitions.

Note In Windows NT, there is no way to resize a partition. You have to back up the data, repartition the disk(s), reformat the partition, and then restore the data.

Creating and Formatting Partitions

To use the NTFS or FAT file systems, you must first create a disk partition and format it for use by the file system.

You can create and format a partition by using the following methods:

- If you have Windows NT installed on your computer, you can use Disk Administrator (in the Administrative Tools program group) to create the partition. You can also format the partition by using Disk Administrator. For information about using Disk Administrator, see Disk Administrator Help, or the chapter titled "Disk Administrator" in the *System Guide* volume of the Windows NT Server and Windows NT Workstation documentation set and Online Books.
- In MS-DOS, you can use the **fdisk** command to create a partition and format the partition by using the **format** command. See MS-DOS Help for information about these commands.

During Windows NT Setup, you can:

- Format an existing, unformatted partition.
- Convert an existing FAT partition to an NTFS partition, which preserves the data on the partition.
- Reformat an existing partition to an NTFS or FAT partition, which erases all files.

See the *Installation Guide* volume of the Windows NT Server and Windows NT Workstation documentation set or Online Books for information about these features.

You can also use the **convert** utility to convert a partition from FAT to NTFS. This utility preserves the data on the partition. For information about using the **convert** utility, see Chapter 5, "Windows NT File Systems and Advanced Disk Management," in the *Windows NT Resource Guide*.

The "Capacity Planning" section later in this chapter discusses limits on the sizes of partitions.

Note Converting a partition from FAT to NTFS can be a lengthy process, depending on the current state and size of the FAT partition. For example, converting a 1-gigabyte partition can take up to several hours. When using the **convert** utility or choosing to convert the partition in Windows NT Setup, no status information is displayed to indicate that the conversion is proceeding. Although it might appear that the convert process is hung, it might simply be taking an extended period of time. If this situation occurs, please allow the conversion to run overnight.

Choosing a File System

The FAT and NTFS file systems support long filenames (up to 255 characters), so naming conventions do not matter when it comes to choosing the file system.

FAT File System Advantages

The FAT file system can be used with operating systems other than Windows NT, such as Windows 95, Windows for Workgroups, MS-DOS, and OS/2.

If you delete a file by mistake, you can use the **undelete** command to restore the file if it was on a FAT partition and you restart your computer under MS-DOS. Windows NT Server and Windows NT Workstation do not support the undelete of a file on either a FAT or NTFS partition, because this task requires direct access to the hardware. This is something that Windows NT does not enable you to do.

The FAT file system is best for drives or partitions under approximately 200 MB, because the FAT file system starts out with very little overhead. The FAT file system is a better choice for partitions that are smaller than approximately 100 MB, because of the amount of disk space overhead involved in NTFS. This overhead is in the form of NTFS system files and the log file, which can use several percent of the total disk space on a small partition.

NTFS File System Advantages

The NTFS file system is best for use on partitions of about 400 MB or more because performance does not degrade with larger partition sizes under NTFS, as it does under the FAT file system.

The NTFS file system enables you to assign permissions to individual files, so you can specify who is allowed various kinds of access to a file or directory. The NTFS file system offers more permissions than the FAT file system, and you can set permissions for individual users or groups of users. The FAT file system only provides permissions at the directory level, and FAT permissions either allow or deny access to all users. However, there is no file encryption built into the NTFS file system. Therefore, someone can start the system under MS-DOS, or another operating system, and then use a low-level disk editing utility to view data stored on an NTFS partition.

The recoverability designed into the NTFS file system is such that a user should seldom have to run any disk repair utility on an NTFS partition. In the event of a system crash, the NTFS file system uses its log file and checkpoint information to automatically restore the consistency of the file system.

Which is Faster, FAT or NTFS?

There are no simple answers to this question.

For small directories, the FAT file system may be faster to get to the file, because:

- The FAT directory structure is simpler.
- The FAT directory size is smaller for an equal number of files.
- FAT provides security only at the directory level, and the permissions either allow or deny everybody access to the directory. Therefore, the system doesn't have to check permissions for an individual file or whether a specific user has access to the file or directory.

The NTFS file system uses a binary tree structure for all directories. This structure minimizes the number of disk accesses required to find a file, which means that the NTFS file system should be faster for larger directories.

In comparing performance on large directories having both long and short filenames, the speed of a FAT operation depends on the operation itself, as well as the history of the directory. Creating files on a FAT directory might be faster. Opening a file might be faster on a FAT directory if the file is at the front of the directory. Not finding the file would be slower. Directory enumeration might be faster on a FAT directory.

Several factors unique to the two file systems affect the speed with which Windows NT reads or writes a file.

- Fragmentation of the file — with the FAT file system, the pointer to the next fragment of a file is at the end of the current fragment. With the NTFS file system, the master file table (MFT) record for the file contains all of the allocation information for the file, unless the file is very large or badly fragmented, in which case the NTFS file system uses disk space outside the MFT for the additional information.
- Cluster size — for the FAT and NTFS file systems, cluster size is a function of the partition size. The FAT file system addresses are 16 bits, so the cluster size for a 640 MB partition is 16K. Cluster sizes for the NTFS file system range from 512 bytes to 4K. The default size depends upon the disk size, and the user can specify a different cluster size when formatting the partition.
- Location of small files — with the NTFS file system, the entire file is contained within the MFT record. The maximum file size that will fit in the MFT record depends upon the cluster size and the number of attributes for the file.

Capacity Planning

The maximum file size and partition size for the FAT file system is 2^{32} bytes (4 gigabytes). FAT can support a maximum of 65,536 clusters per partition.

By using 64 bits for disk addressing and cluster numbering, the NTFS file system has greatly increased the possible file and partition size, so that they can now be up to 2^{64} bytes (16 exabytes or 18,446,744,073,709,551,616 bytes). However, with today's hardware, you will find it hard to get beyond 2 terrabytes (2^{41} bytes).

Note Partitions formatted using the **fdisk** command are limited to 4 gigabytes, and all Windows NT boot partitions are limited to 4 gigabytes. The largest partition that MS-DOS can use is 2 gigabytes.

With the NTFS file system, you can have a maximum of 2^{32} clusters per file.

There is no specific limit to the number of files on an NTFS partition. However, you won't be able to create any new files when your partition is so full that you can't allocate another entry in the MFT.

Hardware and Operating System Considerations

Here are guidelines to help you choose your file system(s):

- If the computer does not need to start any other operating system in addition to Windows NT Workstation or Windows NT Server, use only the NTFS file system.
- If the computer needs to start another operating system, such as Windows 95, Windows for Workgroups, MS-DOS, or OS/2, use the FAT file system for your system partition. You can use the NTFS file system on additional partitions on the computer, as long as those partitions don't need to be accessed by an operating system other than Windows NT Workstation or Windows NT Server.

These additional guidelines might affect your decision as to which file system(s) to use on your computers:

- For an x86-based computer, Windows NT looks for certain files in the root directory of the C partition at startup. This partition can be formatted with either the NTFS or FAT file system. This partition should be large enough to accommodate all the files you need to access under that file system.
- For a RISC-based computer, the system partition must be formatted with the FAT file system. You can use the NTFS file system on your boot partition, which needs to be large enough for all Windows NT system components. If you configure your partitions in this way, your system partition should be 5–10 MB, and your boot partition should be about 100 MB.

  Because you must format your system partition as a FAT file system on RISC-based computers, you can use Disk Administrator to secure the system partition. This prevents anyone who does not have administrative privileges from accessing the system partition in spite of the fact that it is formatted as FAT. See the section "Securing System Partitions" in the chapter titled "Disk Administrator" chapter of your Windows NT Server or Windows NT Workstation *System Guide* for more information.

Note The system partition contains the Master Boot Record, the partition table, the boot sector, and other files needed to load the operating system, such as NTLDR (for x86-based computers) and OSLOADER (for RISC-based computers). The boot partition needs to include the directory with the operating system. The boot partition and the system partition can be a single partition with other directories in it.

NTFS Compression

Windows NT platform version 3.51 supports compression on an individual file basis for NTFS partitions.

Files that are compressed on an NTFS partition can be read and written by any Windows-based application without first being decompressed by another utility. Decompression happens automatically during the read of the file. The file is compressed again when it is closed or explicitly saved.

Only the NTFS file system can read the compressed form of the data. If another program, such as an application like Microsoft Word for Windows or an operating system command like Copy, requests access to the file, the NTFS file system uncompresses the file before making it available. For example, if you copy a compressed file from a server to a compressed directory on your hard drive, the file will be uncompressed, copied, and recompressed.

This compression functionality is similar to that provided by the MS-DOS 6.0 DoubleSpace® and MS-DOS 6.22 DriveSpace™ compression, with one important difference — the MS-DOS functionality compresses the entire partition while the NTFS file system allows the user to compress individual files and directories in the NTFS partition.

Note Windows NT Workstation and Windows NT Server do not support DoubleSpace or DriveSpace compression.

Compressing and Decompressing Directories and Files

Each file and directory on an NTFS partition has a compression state.

You can set the compression state of directories and compress or uncompress files by using File Manager or a command-line utility called **compact**. Using File Manager, you can set the compression state of an NTFS directory without changing the compression state of existing files in that directory.

Using File Manager

With File Manager, you can:

- Set the compression state of an NTFS directory.
- Specify whether to compress or uncompress all files in the directory when you change the state of the directory to compressed or uncompressed.
- Compress or uncompress individual files in the directory.

To set the compression state of the directory:

- Select the directory, and then select Compress or Uncompress from the File menu.
- Select the directory you want to compress or decompress. From the File menu, choose Properties to display the Properties dialog box, and then select or clear the Compressed check box.

File Manager pops up a dialog box asking whether all the files and subdirectories in the directory should be compressed or decompressed. Existing files or subdirectories in the NTFS directories retain their compression state unless you select Yes in this dialog box.

To work with individual files, select one of the following methods.

- Select the files, and then select Compress or Uncompress from the File menu.
- Select the files you want to compress or decompress. From the File menu in File Manager, choose Properties to display the Properties dialog box, and then select or clear the Compressed check box to compress or decompress the files.

  The Properties dialog box can also be used to view the compressed size and compression ratio of a selected file.
- Select the files you want to compress or decompress, and then choose the Compress or Uncompress buttons on the File Manager toolbar.

  The compress and uncompress buttons are displayed on the toolbar only if you customized the File Manager toolbar.

  For more information about customizing the File Manager toolbar, see "Customizing the Toolbar," in the File Manager chapter of the Windows NT Workstation *System Guide* or Windows NT Server *System Guide*.

Note Compressed files, directories, and subdirectories are displayed in blue in File Manager. Therefore, do not configure your desktop to use a blue background or blue text.

Using the Compact Utility

The **compact** utility is the command line version of the compression functionality in File Manager. The **compact** command displays and alters the compression of directories and files on NTFS partitions. It also displays the compression state of directories.

The format of the command is:

compact [**/c**] [**/u**] [**/s**[**:***dir*]] [**/a**] [**/i**] [**/f**] [**/q**] [*filename* [...]]

The following table describes the options. For more information about **compact** and its options, type **compact /?** at the command prompt.

Parameter	Description
none	Displays the compression state of the current directory.
/u	Compresses the specified directory or file.
/u	Uncompresses the specified directory or file.
/s[:dir]	Specifies that the requested action (compress or uncompress) be applied to all subdirectories of the specified directory, or to the current directory if none is specified.
/f	Ignores errors.
/f	Forces compression or uncompression of the specified directory or file.
/a	Displays files with the hidden or system attribute.
/q	Reports only the most essential information.
filename	Specifies a pattern, file, or directory. You can use multiple filenames and wild cards.

There are reasons why you would want to use this utility instead of File Manager:

- You can use compact in a batch script. Using the **/i** option enables you to skip files that cannot be opened when you are running in batch mode, such as when a file is already in use by another application.
- If the system crashed when compression or decompression was occurring, the file or directory is marked as Compressed or Uncompressed, even if the operation did not complete. You can force the operation to complete by using the **compact** utility with the **/f** option (with either the **/c** or **/u**).

Note Unlike File Manager, the **compact** utility does not prompt you on whether you want to compress or uncompress files and subdirectories when you set the compression state of a directory. It automatically does the compression or decompression of any files that are not already in the compression state you just set for the directory.

Effects of Moving and Copying Files

When you replace an existing file in an NTFS directory, the file might retain its compression state regardless of the compression state of the directory and the compression state of the source file. Thus, if you copy a compressed NTFS file to a compressed NTFS directory, but it replaces an uncompressed file, the resulting file will probably be uncompressed. This situation is true whether the source directory was on an NTFS or FAT partition.

Note The effects described in this section are generally true when using Microsoft applications. Third-party utilities might function differently.

Moving and Copying Files Within NTFS Partitions

On an NTFS partition, if you move a file from one directory to another, the compression attribute, like any other attribute of the file, is retained regardless of the compression of either the target or source directory. For example, if you move an uncompressed file to a compressed directory, the file remains uncompressed after the move.

However, if you copy a file from one directory to another, the compression attribute of the file is changed to that of the target directory. For example, if you copy a compressed file to an uncompressed directory, the file is automatically decompressed when it is copied to the directory.

Moving and Copying Files Between FAT and NTFS Partitions

Unlike files moved between NTFS directories, files moved or copied from a FAT directory to an NTFS directory always inherit the compression attribute of the target directory. Moving a file from a FAT directory to an NTFS directory causes a copy of the file, followed by a delete.

Since Windows NT supports compression only for NTFS files, any compressed NTFS files moved or copied to an FAT partition are automatically decompressed. Similarly, compressed NTFS files copied or moved to a floppy disk are automatically decompressed.

Adding Files to an Almost Full NTFS Partition

When adding files to an NTFS partition that is almost full, you can get unexpected error messages. The philosophy behind these errors is that NTFS wants to make sure it has enough disk space to write the entire file if it can not be compressed, regardless of the degree of compression in the file when it is opened. For instance, it is possible to get a read error when you are trying to open a compressed file.

If you copy files to a compressed NTFS directory that doesn't have enough room for all of the files in their uncompressed state, and they will all fit when compressed, you may get an error that says "...there is not enough space on the disk." Since NTFS allocates space based upon the uncompressed size of the file, you can get this error even if the files are already compressed.

This situation occurs because compression is handled asynchronously, and Windows NT uses lazy write (see *Inside the Windows NT File System* for information about lazy writes). NTFS does not wait for the compression and write of one file to complete before it begins work on subsequent files, and the system doesn't get the unused space back from compression until after the buffer is compressed.

When you save files from an application to a compressed directory on a partition that is almost full, the save might or might not be successful — it depends on how much the file compresses, whether the beginning of the file compresses well, and numerous other factors.

If you can't delete any files or don't have any files that you can compress, you can usually copy all of the files if you copy the largest and/or the ones that compress best first. You can also try copying them in smaller groups rather than all at one time.

Determining Directory Usage

You can use the **DirUse** command in the *Windows NT Resource Kit* CD to get the actual usage of space for compressed files and directories. The **DirUse** command provides general functionality for FAT partitions as well as NTFS partitions.

The format of the command is:

diruse [/s | /v] [/q:#] [/m | /k | /b] [/a] [/l] [/d] [/o] [/c] [/,] [/*] [*dirs*]

For more information about the **DirUse** command, type **diruse /?** at the command line, or see the Windows NT Resource Kit Tools Help for information about this utility.

The option of interest for compressed directories and files is **/c**, which causes the display of compressed file or directory size instead of apparent size. For example, if your D drive is an NTFS partition, type the following command at the command prompt:

diruse /s /m /c d:

to get the disk space actually used (in MB) and number of files in each of the directories. To see compression information for an individual file, you need to use File Manager, select the file, and select Properties from the File menu.

Compression Algorithm

The book *Inside the Windows NT File System* provides a high-level description of NTFS compression.

Basically, the NTFS file system provides real-time access to a compressed file, decompressing the file when it is opened and only compressing when it is closed.

When writing a compressed file, the system reserves disk space for the uncompressed size. The system gets back unused space as each individual compression buffer gets compressed.

Note Some applications do not allocate space before beginning the save, and only pop up an error message when they run out of disk space.

NTFS file system compression uses a 3-byte minimum search rather that the two-byte minimum used by DoubleSpace. This type of search enables a much faster compression and decompression (roughly two times faster) while only sacrificing two percent compression for the average text file. That is, when Windows NT finds a 3-byte match, it stores it into the hash table, provides an ID number, and uses that number as a hash mark.

Each NTFS data stream contains information that indicates if any part of the stream is compressed. Individual compressed buffers are identified by "holes" following them in the information stored for that stream. If there is a hole, then Windows NT knows to decompress the preceding buffer to fill the hole.

NTFS Compression Compared to Other Methods

Other compression utilities are available to compress files on computers running Windows NT. These utilities differ from NTFS compression in the following ways:

- They usually can be run only from the command line.
- Files cannot be opened when they are in a compressed state — the file must first be uncompressed by using the companion utility to the one used to compress the file. When you close the file, it is saved in an uncompressed state, and you have to use a utility to compress it.

The *Windows NT Resource Kit* includes a compress utility and two expand utilities. The compress utility can only be run from the command line. There are two versions of the expand utility: one runs from the command line, and the other is a Windows-based application.

As described earlier, the DoubleSpace and DriveSpace compression features in MS-DOS cannot be used with Windows NT Workstation or Windows NT Server.

Using Compress Utility

This command line utility can be used to compress one or more files.

To use this utility, type **compress** with the appropriate options at the command line:

compress [**-r**] [**-d**] *source* [*destination*]

Parameter	Description
-r	Renames compressed files.
-d	Updates compressed files only if out of date.
source	Specifies the source file. The "*" and "?" wildcards can be used.
destination	Specifies the destination file or path. The destination can be a directory. If source specifies multiple files and the **-r** option is not specified, then *destination* must be a directory.

Expanding Compressed Files

You can use the file expansion utilities to expand one or more compressed files from the Windows NT CD or a file that you compressed by using the **compress** command.

You use the Windows-based File Expansion utility as follows:

- If you have installed the *Windows NT Resource Kit*, you can double-click the File Expansion utility icon in the Resource Kit program group
- From File Manager, open the EXPNDW32.EXE file. If you have installed the *Windows NT Resource Kit*, this utility is in the directory that you used for the install. On the *Windows NT Resource Kit* CD, use the version that corresponds to the hardware platform that you are using.
- For help while you are using the utility, choose the Help button or press F1.

The MS-DOS-based version of the utility runs from the command line. Type the **expand** command with the appropriate options:

expand [**-r**] *source* [*destination*]

Parameter	Description
-r	Renames expanded files.
source	Specifies the source file. The "*" and "?" wildcards can be used.
destination	Specifies the destination file or path. The destination can be a directory. If source specifies multiple files and the **-r** option is not specified, then *destination* must be a directory.

NTFS Compression Issues

The only check for whether a user is allowed to change the compression state on a file is whether they have read or write permission. If so, then they can change the compression state locally or across the network.

The two ways to measure the performance of NTFS data compression are size and speed.

You can tell how well compression works by comparing the uncompressed and compressed file and directory sizes. The earlier section of this chapter titled "Using File Manager" describes using the Properties dialog box to view the compressed size and compression ratio of a selected file. The earlier section of this chapter titled "Determining Directory Usage" describes using a utility to see the compressed size of directories.

Using NTFS compression might cause performance degradation. One of the reasons this might happen is that, even when copied inside the same computer, a compressed NTFS file is decompressed, copied, and then recompressed as a new file. Similarly, on network transfers, the file is decompressed, which affects bandwidth as well as speed.

With data compression, the question is "How will it affect performance on a computer running Windows NT Workstation or Windows NT Server?"

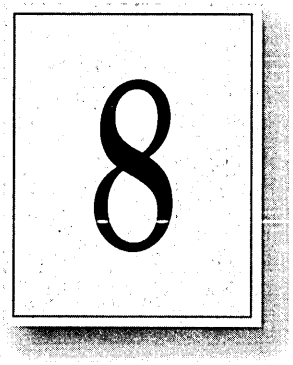
The current implementation of NTFS compression is definitely oriented toward Windows NT Workstation. Compression on a Windows NT Workstation does not seem to produce a substantial performance degradation. No one who has started using NTFS data compression at Microsoft has lodged any complaints about performance degradation.

A Windows NT Server can be another story. Some normal production servers at Microsoft (source servers and binary release servers) have been converted to use NTFS data compression, without any complaints about performance. However, using a tough server benchmark, like NetBench, shows a performance degradation in excess of 50%. Read-only or read-mostly servers, or any lightly loaded servers, may not see a severe performance penalty. Heavily loaded servers with lots of write traffic are poor candidates for data compression, as shown by NetBench.

You really need to measure the effects of data compression in your own environment.

CHAPTER 8

Fault Tolerance for Disks



Hardware and software technology exist today on Windows NT Server to improve the reliability and recoverability of data. This chapter discusses improving reliability by storing redundant information on the hard drives, using a family of techniques and standards called Redundant Array of Independent Disks (RAID). You can use RAID hardware with both Windows NT Server and Windows NT Workstation.

For a complete discussion of RAID, as well as background information about disk subsystems and disk arrays, see *The RAIDbook — A Source Book for Disk Array Technology*, ISBN 1-879936-90-9. It is published by the RAID Advisory Board, St. Peter, MN. The fourth edition is dated June 1995.

In this chapter, it is assumed that the reader is familiar with disk terminology, such as controller, bus, duplexing, volumes, and partitions. The purpose of this chapter is to describe the steps for creating a fault-tolerant disk configuration, recovering disk information following a hardware failure, and rebuilding fault-tolerant disk sets after a hardware failure. There are currently many sources for this information:

- The section titled "Windows NT Fault-Tolerant Mechanisms" in Chapter 5, "Windows NT File Systems and Advanced Disk Management" of the *Windows NT Resource Guide.* The disk information from that section has been incorporated into this chapter.
- Chapter 7, "Managing Fault Tolerance and UPS," of the *Concepts and Planning Guide* of the Windows NT Server documentation set. The disk information from that chapter has been incorporated into this chapter.
- Chapter 2, "Troubleshooting," of the *Installation Guide* of the Windows NT Server and Windows NT Workstation documentation sets.
- Chapter 18, "Disk Administrator," of the *System Guide* of the Windows NT Server documentation set.

- Chapter 16, "Disk Administrator," of the *System Guide* of the Windows NT Workstation documentation set.
- Knowledge Base articles, which are available on the *Windows NT Resource Kit* CD, Microsoft Development Library (MSDN), and Microsoft TechNet.

This chapter brings together all of the pertinent information on the following subjects:

- Planning a fault-tolerant disk configuration
- Creating a fault-tolerant volume set
- Preparing for recovery
- Data recovery for Windows NT Workstation and Windows NT Server
- Restoring disk configuration information
- Recovering a fault-tolerant volume set
- Using FTEdit to update the Registry

Planning a Fault-Tolerant Disk Configuration

Over the last several years, the unit cost of mass storage has dropped dramatically, while computer speed and capacity have increased. Several changes in today's computing environment are changing the way users access data, and the volume of data that they use.

- Networks are now common, and most universities and mid-size and large-size businesses rely on large, fast networks to conduct everyday activities and communication.
- A small number of powerful servers provide storage, backup, printing, and other services for other computers on the network.
- New graphical operating systems and applications with graphical interfaces have increased the volume of data and programs used on both workstations and servers. The trend to use graphics in databases has accelerated the increase in the volume of data used by typical users.

The challenge for mass storage devices is to provide cost effective access to data in the modern computing environment. This challenge is difficult, because the most cost-effective mass storage technologies still rely on electromechanical components. These devices require more power, and generate more noise, vibration, and heat than purely electronic devices. This makes disks slower and more failure-prone than the more expensive electronic solutions.

RAID is a disk array in which part of the physical storage capacity contains redundant information about data stored on the remainder of the storage capacity. The redundant information enables regeneration of user data in the event that one of the array's member disks or the access path to it fails.

RAID technology is a way to provide a solution for the two main concerns in data storage:

- To improve I/O performance. Performance improvements need to keep pace with other technologies, so that access to data does not become the limiting factor for applications.
- To improve the reliability of data storage. It is essential that the level of data reliability be equal to or greater than the expected lifetime of the systems that process the data.

Overview of RAID Technology

You can implement RAID fault tolerance in either hardware or software. In a hardware solution, the controller interface handles the creation and regeneration of redundant information. In Windows NT Server, this activity can also be performed in the software, using the Windows NT file system (NTFS) and the File Allocation Table (FAT) file system.

Windows NT Server offers three of the RAID strategies (levels 0, 1, and 5) in a software solution. Both Windows NT Server and Windows NT Workstation support RAID 0, which is not fault tolerant. It is included here because:

- It is the basis for RAID 5.
- It improves I/O performance.
- Windows NT Workstation and Windows NT Server need the same kind of information, both on disk and in the Registry, to identify and access these three kinds of volumes.

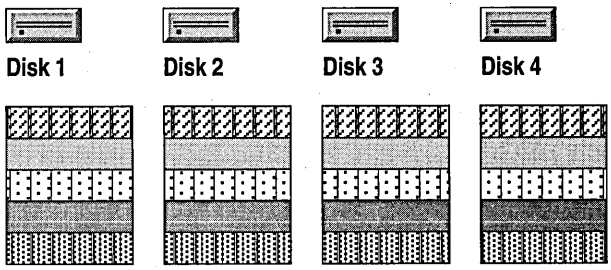
RAID 0 — Stripe Set

RAID 0 is a performance-oriented data mapping technique. Since it is simple, the array management software imposes very little processor overhead. With modern SCSI bus mastering technology, multiple I/O operations can be done in parallel, further enhancing performance.

RAID 0 arrays use from two to 32 physical disks to form a larger virtual disk. The data are written in strips of equal size on each disk. The size of the strips is based on an allocation unit size, which can be any size, from a cluster to a logical track. Smaller allocation unit sizes with parallel access improve the single stream data transfer rate. Other implementations are based on a larger allocation unit size. When the allocation unit size is large compared to the average transfer size, the I/O request rate improves, because the average data request can be made in a single block read.

For Windows NT Server and Windows NT Workstation, the allocation unit size is 64K.

The next figure shows a stripe set using four disks. A strip is one of the blocks on one of the disks. In this figure, stripe 1 consists of the four strips that are the first block on each of the four disks. Stripe 5 is made up of the strips that are the last block on each disk.



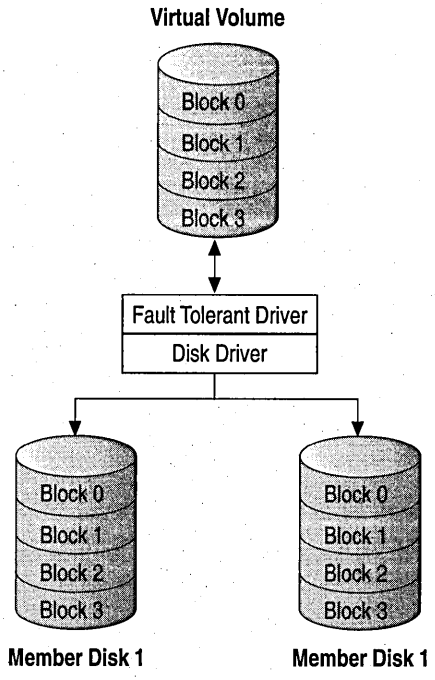
Although the data are spread across multiple disks, there is no data redundancy and therefore no fault tolerance. When any disk fails, the whole array fails, and no data can be recovered. In addition, the reliability for the array is worse than the least reliable disk in the set.

RAID 1 — Mirror Set

RAID 1 is disk mirroring. Disk mirroring provides an identical twin for a selected disk; all data written to the primary disk is written to the twin or mirror disk, which results in disk space utilization of only 50 percent. In this chapter, the term primary disk or primary partition refers to the original partition, and shadow disk or shadow partition refers to the mirror disk or partition.

Because dual-write operations can degrade system performance, many mirror set implementations use duplexing, where each mirror drive has its own disk controller. While the mirror approach provides good fault tolerance, it is relatively expensive to implement, because only half of the available disk space can be used for storage, while the other half is used for mirroring.

Mirroring is not restricted to a partition identical to the primary partition in size or number of tracks and cylinders, nor to disk drives made by the same manufacturer. This means that you do not have to replace a failed drive with an identical model. For practical purposes, though, the shadow partition should be the same size as the primary partition. The shadow partition cannot be smaller than the primary. If the shadow partition is larger than the primary, the extra space on the shadow drive is left as free space.



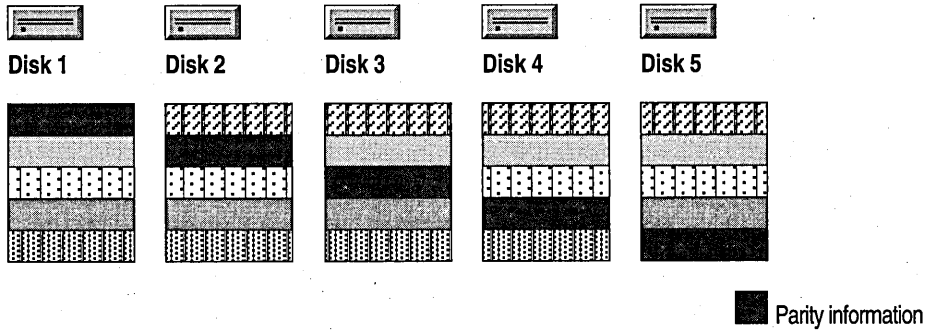
If there is a read failure on one of the drives, the system reads the data from the other drive in the mirror set. If there is a write failure on one of the drives in the mirror set, the system uses the remaining drive for all accesses.

When compared to stripe sets with parity (RAID 5), a mirror-set implementation has a lower entry cost (because it requires only two disks, whereas a stripe set with parity requires three or more disks), requires less system memory, provides the best overall performance, and does not show performance degradation during a failure. However, its cost-per-megabyte is higher than that for RAID 5. You can mirror the boot partition, which significantly reduces the amount of time needed to get your Windows NT Server back up if there is a problem with the hard disk containing your operating system.

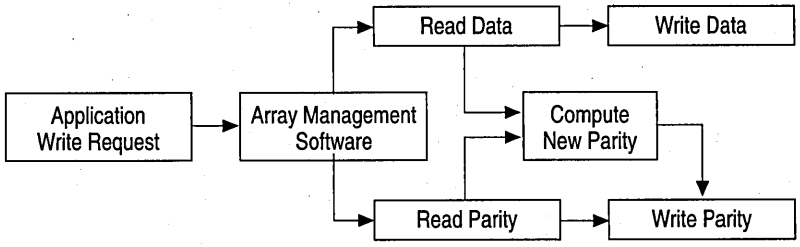
Note With Windows NT Workstation and Windows NT Server, the name system partition refers to the disk partition containing hardware-specific files needed to start Windows NT Server or Windows NT Workstation (such as the x86-based files NTLDR and BOOT.INI or the RISC-based files OSLOADER.EXE and HAL.DLL). The boot partition contains the operating system files and support files. The boot partition and the system partition can be the same partition.

RAID 5 — Stripe Set With Parity

RAID 5 adds fault tolerance to the RAID 0 technology by including parity information with the data. RAID 5 dedicates the equivalent of one disk for storing the parity strips, but distributes the parity strips across all the drives in the group. The data and parity information are arranged on the disk array so that they are always on different disks. In the next figure, the first block on disk 1 is the parity strip for the four data strips in stripe 1. In stripe 2, the parity strip is on disk 2, and so on.



Because the parity strip is simply the exclusive OR (XOR) of all the data values for the data strips in the stripe, as long as the old data and the old parity values are known, the new parity for a write can be calculated without having to read the corresponding strips from the other data disks. Thus, only two disks are involved in a write operation: the target data disk and the disk that contains the parity strip.



There must be at least three disks and no more than 32 disks in a stripe set with parity. A partition of approximately the same size should be selected from each disk. The disks can be on the same or different controllers.

If one of the disks in a stripe set with parity fails, none of the data are lost. When a read operation requires data from the failed disk, the system reads all of the remaining good data strips in the stripe and the parity strip. Each data strip is subtracted (with XOR) from the parity strip; the order isn't important. The result is the missing data strip.

When the system needs to write a data strip to a drive that has failed, it reads the other data strips and the parity strip and backs them out of the parity strip, leaving the missing data strip. The modifications needed to the parity strip can now be calculated and made. Because the data strip is bad, it is not written; only the parity strip is written.

There is no effect on a read operation when the disk that failed contains a parity strip. (The parity strip isn't needed for a read, unless there is a failure in a data strip.) When the failed drive contains a parity strip, the system does not compute or write the parity strip when there is a change in a data strip.

A stripe set with parity implementation has better read performance and a lower cost-per-megabyte than a mirror set, but it requires more system memory (recommended minimum RAM is 12 MB; 16 MB or greater is preferred), and loses its performance advantage when a member is missing. RAID 5 is recommended over RAID 1 for applications that require redundancy and are primarily read-oriented.

Neither the system partition nor the boot partition can be configured as a stripe set with parity.

Hardware Versus Software Solutions

On heavily loaded Windows NT Server computers, there might be advantages to using a hardware solution for RAID levels 1 or 5. There are many factors involved, which depend on the specific site, hardware being used, and the load on the computer running Windows NT Server.

Here are some points to consider when deciding whether to implement fault tolerance in hardware or software.

- RAID 1 and RAID 5 software apply only to Windows NT Server.
- RAID 1 and RAID 5 hardware can be used on both Windows NT Server and Windows NT Workstation.
- Hardware fault tolerance is faster.
- Software fault tolerance is less expensive.
- A hardware fault tolerance solution might lock you into a single vendor.
- In a hardware fault tolerance implementation, some vendors support hot-swappable drives when there is a drive failure.

Regardless of whether you implement fault tolerance by using hardware or software, implementing fault tolerance does not reduce the need for backups.

Creating a Fault-Tolerant Volume Set

Once you've decided what data you want to put on a fault-tolerant volume set, you might have to procure and install additional disk drives and/or disk controllers. If you are using fault-tolerant hardware, follow the manufacturer's installation instructions. When using software fault tolerance, using SCSI controllers is more efficient and provides an additional recovery mechanism of sector sparing. See the section titled "Summary of Windows NT Data Recovery," presented later in this chapter, for information about sector sparing.

If you plan to use software fault tolerance, you need to use Disk Administrator to configure your disks.

Creating a Mirror Set

Although there are times when it might not be possible (such as when rebuilding a member of a mirror set following a drive failure), you should use identical drives, with the same formatting, for your primary and shadow partitions. This practice is important when the mirror set contains your boot or system partition. Otherwise, you might never be able to boot from a shadow partition if the primary disk fails, and will have to boot from your fault tolerance boot floppy disk.

See the sections "Preparing for Recovery" and "Recovering a Mirror Set," presented later in this chapter, for more information.

Use Disk Administrator to establish a mirror set as follows:

1. Select the partition that you want to duplicate and an area of free space the same size or larger on another hard disk by selecting the first partition, and then pressing CTRL and clicking the area of free space.
2. From the Fault Tolerance menu, choose Establish Mirror.

   Disk Administrator creates an equal-sized partition in the free space for the mirror and assigns the drive letter to the mirror set.

Note In some cases, where a mirror set is being established, a slightly larger amount of space is required for the shadow partition than for the primary partition. This happens when geometric differences exist between the drives.

   If you are mirroring the system partition, Disk Administrator displays a message box reminding the user to create a fault-tolerance boot floppy disk.
3. From the Partition menu, select Commit Changes Now. Select Yes in the Commit Changes dialog box.

   A Confirm change/restart dialog box is displayed; again select Yes. A message box prompts you to update your Emergency Repair Disk, which you do by using the **rdisk** command. Finally, you are prompted to shutdown and restart the system. The only option is OK. The system shuts down and restarts.
4. After the system restarts, open Disk Administrator.

   The drive letter, volume label, and partition size of the mirror set appear in red text. This visual cue indicates that the system is generating the partition in the background. While the text is highlighted in red, fault tolerance is not functional.
5. If your mirror set contains the system partition, set the active partition flag for the shadow partition by choosing Mark Active in the Partition menu.

   You need to set this flag to be able to boot from the shadow partition without using the fault tolerance boot floppy disk if the primary partition fails.

Note In the Disk Administrator display, information about the partition is not updated automatically. If you click on the partition, the information for that volume updates. You can also check the Event Log to know when the generation of the shadow partition is finished.

Creating a Stripe Set With Parity

When using a software stripe set with parity, keep the operating system and page file on a controller other than the RAID array containing data. This improves performance. You can use different types of controllers when you have the operating system on a different controller from the stripe set with parity.

▸ **To create a stripe set with parity**

1. Select areas of free space on three to 32 hard disks by selecting the first area of free space on the first disk, pressing CTRL, and then choosing each additional area of free space on each of the other hard disks.

   If the areas selected vary in size, then the smallest area is used as the base partition size. For example, if the available space consists of a 700 MB partition on disk 1, and 900 MB partitions on disks 2 and 3, only 700 MB on each disk is used as part of the stripe set with parity. The space left over remains free space and can be partitioned and formatted as separate logical drives.

2. From the Fault Tolerance menu, choose Create Stripe Set With Parity.

   Disk Administrator displays the minimum and maximum sizes for the stripe set with parity. The default size is the maximum allowed for the disk areas selected.

   The size you choose is the total disk space that is used, not the size available for data on the stripe set with parity. For example, if four 200 MB partitions are selected:

   - Disk Administrator displays 800 MB for the maximum size.
   - You have 600 MB available for data, if you select the maximum size. The other 200 MB is used to store the parity information.
   - If you want to use less than the maximum size, the formula is

     ```
     data size = total size - (total size/4)
     ```

3. In the Create Stripe Set With Parity dialog box, type the size of the set that you want to create, and then choose OK.

   Disk Administrator divides the total size that you enter by the number of disks in the set to create equal-sized unformatted partitions on each of the selected disks. It then assigns a single drive letter to the collection of partitions that make up the stripe set with parity. Disk Administrator displays the stripe set with parity as New Unformatted space.

4. From the Partition menu, select Commit Changes Now.

   You are prompted to save your disk configuration changes. Select Yes.

5. You are notified that, for the changes to the configuration to take effect, you need to restart the computer. Select Yes from the dialog box.

6. You will then be notified that the disks were updated successfully.

   This means that the new configuration information has been saved in the Registry. This dialog box also prompts the user to update the Emergency Repair Disk.

7. Finally, you are prompted to shut down the system.

   The only option is OK.

The system automatically closes all open applications and reboots the system. Your stripe set with parity begins initializing immediately upon reboot. The initialization process runs as a background task. Disk Administrator displays a message on its status bar to indicate that the stripe set with parity is initializing. The drive letter, volume label, and partition size of the stripe set with parity appear in red text during initialization.

Note A stripe set with parity is distinguished from a stripe set only in the status bar. The color shown in the legend is the same for both types.

Preparing for Recovery

There is no way to make a computer running Windows NT Server or Windows NT Workstation failure proof. You can only make the computer more failure resistant. A memory module, cabling, or controller failure can corrupt the data in a fault-tolerant array. In this event, the only option is to restore from a tape or a backup server that contains a copy of the data. You also need to develop plans and procedures for recovering from failures before you have one.

If your system or boot partition is on a mirror set, you should create and test fault tolerance boot floppy disks, as described in this section. You should also maintain configuration information for your computers running Windows NT Server and Windows NT Workstation. At a minimum, you should keep track of the version of the operating system installed on each computer, including service packs and hotfixes. You should also know the hardware configuration of each computer running Windows NT Workstation and Windows NT Server, especially the disk configurations.

The section titled "Recovering a Server" in Chapter 8, "Backing Up Network Files," of the Windows NT Server *Concepts and Planning Guide* contains more information about things to consider to make recovery easier.

Note Be wary of using disk utilities that were not designed for computers running Windows NT Workstation and Windows NT Server. These utilities do not recognize NTFS partitions and, on FAT partitions, most of them are not compatible with the method that the Windows NT platform uses to store long filenames.

Creating a Fault Tolerance Boot Floppy Disk

You need to use a fault tolerance boot floppy disk during the following circumstances:

- The system and boot partitions are the same partition, and the primary partition has failed.
- The system and boot partition are different, and the primary system partition has failed. Even though the boot partition is still good, the only way to start the system is to use the fault tolerance boot floppy disk or install Windows NT Server on the replacement drive. Using the floppy disk is faster, and you do not have to install Windows NT Server, because the system regenerates the data on the replacement drive.
- Your Windows NT Server computer is an x86-based computer, the system partition is different from the boot partition, and the primary boot partition has failed. Even though you still have the system partition, the ARC path in the BOOT.INI file on the system partition now points to the failed drive. If you have a FAT system partition, you can boot from an MS-DOS floppy disk, edit the BOOT.INI file to include the path to the shadow drive, and then start the operating system. When you create a fault tolerance boot floppy disk that has an entry in BOOT.INI for the shadow partition, you can use that floppy disk to load the operating system from the shadow partition for both FAT and NTFS partitions.

You create a fault tolerance boot floppy disk by using the operating system or the command prompt to format a floppy disk, and then copying the Windows NT startup files to the floppy disk. You might need to use this floppy disk if your mirror set system or boot partition fails.

There are two procedures involved in creating the floppy disk: formatting it and copying files to it.

Formatting the Disk

You can format your floppy disk by using File Manager, as follows:

1. Insert a disk into a floppy drive.
2. From the Disk menu, choose Format Disk.
3. In the Disk In box, make sure that the drive shown is the one containing the floppy disk you want to format. If necessary, select the correct drive.
4. In the Capacity box, select the size for the disk you want to format.
5. To give the disk a volume label, type a name in the Label box under Options.
6. To quickly reformat a disk that has been previously formatted, select the Quick Format check box. A quick format deletes directory information in the file allocation table and root directory, but does not check for bad sectors.
7. Choose OK.
8. A message prompts you to confirm that you want to format the disk. Formatting deletes all information on a disk, so make sure the disk is the one you want to format. Then choose Yes.

As formatting proceeds, File Manager displays its progress in a dialog box. You can still use File Manager while formatting continues in the background. To cancel formatting, choose the Cancel button. To hide the message box while formatting continues, choose the Hide button.

After your disk is formatted, a message prompts you to indicate whether you want to format another disk.

To format the floppy disk from the command prompt, use the **format** command.

Copying the Windows NT Boot Files for an x86-based Computer

Copy the following files from the C partition root directory to the root directory of the floppy disk you just formatted:

- NTLDR — Windows NT multiboot loader program.
- BOOT.INI — describes the location of the boot partitions, specified by using Advanced RISC Computing (ARC) naming conventions.
- NTDETECT.COM — used for hardware detection.
- NTBOOTDD.SYS — required only if you are using the *scsi* syntax in place of *multi* syntax in BOOT.INI.

The NTBOOTDD.SYS file is a renamed copy of the SCSI miniport driver used on your Windows NT Server computer. For example, if you are using the Adaptec 1542B SCSI host adapter, copy AHA154X.SYS to the floppy disk, and then rename it to the NTBOOTDD.SYS file. See the section titled "Creating Alternate Boot Selections for an x86-Based Computer," later in this chapter, for information about the BOOT.INI file.

These files normally have the Read Only, System, and Hidden attributes set. If the files have either the System or Hidden attribute set, they are not visible. You need to make the files visible before you can copy them to a floppy disk.

▸ **To copy the files:**

1. From the View menu in File Manager, select By File Type.
2. Select the Show Hidden/System files check box, and then choose OK.
3. Select the file for which you want to change attributes.
4. Choose Properties from the File menu.
5. From the Attributes box, uncheck the Hidden and System check boxes and select OK.
6. Select the files, and then copy them to the floppy disk.

Copying the Windows NT Boot Files for a RISC-based Computer

RISC-based computers boot from the system firmware. You can define a boot selection in the firmware that points to a fault tolerance boot floppy disk, as described in the section titled "Creating Alternate Boot Selections for a RISC-Based Computer," later in this chapter. You only need to use the fault tolerance boot floppy disk if your primary partition failed, and it contained the system partition (the partition with OSLOADER).

When you install Windows NT Server or Windows NT Workstation on a RISC-based computer, it creates a directory, \os*systemroot*, that contains the HAL.DLL and OSLOADER.EXE files. On Alpha AXP-based computers, this directory also contains several files with the .PAL extension. Some or all of these files might have the system, hidden, or read only attributes set.

A fault tolerance boot floppy disk for a RISC-based computer should have a directory tree identical to the RISC-based system partition. For Window NT Workstation and Windows NT Server version 3.51, you should create the \os*systemroot* directory on the floppy disk.

Copy the following files from the \os*systemroot* directory on your hard drive to the same directory on the floppy disk:

- OSLOADER.EXE
- HAL.DLL

On AXP-based computers, copy all the files with the .PAL extension to the \os*systemroot* directory on the floppy disk.

Understanding ARC Names

The path to each Windows NT Workstation and Windows NT Server installation file is described in a startup file that uses conventions that are part of the ARC specifications. These conventions are used for compatibility with ARC computers that run on the Windows NT platform.

There might be multiple entries in your startup file for loading different operating systems, or, in the case of mirrored or duplexed drives, force loading of the operating system from a different physical drive.

Note The BOOT.INI file is the startup file for x86-based computers. On RISC-based computers, the system firmware provides similar functionality.

The syntax of the ARC path name is:

<controller>(W)disk(X)rdisk(Y)partition(Z)\<path>

Where:

- <controller>(W) — The only valid values for this parameter are *multi* and *scsi*.

  For Windows NT Server and Windows NT Workstation version 3.5 and above, the default is to use *multi*, except where:

  - The bootable drive is on a SCSI controller, with BIOS disabled.
  - You have a dual channel SCSI controller.
  - You are using multiple SCSI controllers.

  Note The SCSI BIOS is disabled when starting from other than the primary SCSI card, such as a failed duplex controller or a failed primary partition. Also, some older SCSI cards, such as the AHA 1510, do not even have a BIOS.

  The *scsi* syntax is often required to boot from the fault tolerance boot floppy disk even though the *multi* syntax works to boot from the hard drive. If your system fails to boot from the floppy disk, try *scsi* instead of *multi*.

In Windows NT Workstation and Windows NT Advanced Server version 3.1, the default was to use *scsi* with a SCSI controller and *multi* for IDE and ESDI drive controllers.

W is the controller ordinal, using zero-based numbering. For systems that have a single disk controller, the number is zero. With multiple disk controllers, 0 is the first disk controller to initialize. Numbers increase consecutively for all controllers in the system.

When using multiple disk controllers, the *scsi* addressing scheme should always be used, and the NTBOOTDD.SYS file must be present in the root directory of the system partition or fault tolerance boot floppy disk.

Note The primary difference between the *scsi* and *multi* syntax is that *scsi* relies on a device driver that is loaded during the boot sequence (NTBOOTDD.SYS), whereas *multi* goes through the BIOS to access the controller and drive.

- disk(X) — For the *scsi* syntax, using single SCSI controllers, X is the target ID of the disk. The section titled "Dual Channel Controllers," later in this chapter, contains information about computing X in this situation.

  For *multi*, X is always 0.

- rdisk(Y) — For the *scsi* syntax, Y is almost always 0 because the disk(x) field provides the necessary information. For *multi*, Y is the ordinal for the disk on the adapter and is generally either 0 or 1.

  For example, to boot from the second physical drive of an x86-based computer when the SCSI ID of the second drive is 3, the *multi* form of the ARC path appears as follows:

  ```
  multi(0)disk(0)rdisk(1)partition(1)\WINNT35
  ```

  To access the same device by using the *scsi* notation, use the syntax shown in the following example:

  ```
  scsi(0)disk(3)rdisk(0)partition(1)\WINNT35
  ```

- partition(Z) — Z is the ordinal value for the partition on the disk. Primary partitions are numbered first, followed by logical drives. Partition numbering starts with 1, which is the first partition visible to the operating system. Some partition types are not recognized by the boot loader. These include MS-DOS extended partitions (type 5), and unused partitions (type 0). EISA configuration partitions, such as those used by Compaq servers, are type 0 and cannot be used by Windows NT Workstation or Windows NT Server.

- <path> is a directory path.

Creating Alternate Boot Selections for an x86-based Computer

To start Windows NT Server after the failure of a mirror set system or boot partition, you might need to change the hardware identifier on the ARC path name in the BOOT.INI file. Therefore, you should edit the BOOT.INI file to create boot options for every contingency for your configuration.

If you can configure your Windows NT Server computer with the *multi* syntax in the BOOT.INI file, use consecutive IDs for both drives of the mirror set, and have set the active partition flag for both partitions, you should be able to reboot your system from either drive. Be sure to test rebooting for this configuration.

Note The system BIOS controls whether or not your system starts up from a floppy or hard drive. The BOOT.INI file controls from which hard drive and partition Windows NT Workstation and Windows NT Server loads.

The BOOT.INI file has the Read Only attribute set by default. It is necessary to remove this attribute before editing the file. Restoring the attribute is optional. Windows NT Setup sets the attribute to prevent accidental deletion.

Single SCSI Controller

The following sample shows a BOOT.INI file from a system with a single SCSI controller. On computers with a single controller, you should always use the *multi* notation and must enable the BIOS, if the controller has a BIOS.

This BOOT.INI file loads the system from the shadow drive, if the primary drive is not available. The only requirements are that the shadow drive be the next available SCSI ID on the bus, and the primary drive must be powered down or removed from the bus. The following example uses the *multi* syntax.

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\FT_TEST

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\FT_TEST="Windows NT Server"
multi(0)disk(0)rdisk(0)partition(1)\FT_TEST="Windows NT Server [VGA]"
/basevideo
```

If the primary drive cannot be powered down, you need to have an entry in the BOOT.INI file to force NTLDR to load from the shadow drive. This BOOT.INI file loads the system from the shadow drive of a mirror set while the primary drive is still functional. Remember, rdisk(1) refers to the ordinal number and not the SCSI ID of the drive. This example has an entry for the shadow drive.

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\FT_TEST

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\FT_TEST="Primary Drive"
multi(0)disk(0)rdisk(1)partition(1)\FT_TEST="Shadow Drive"
```

Multiple SCSI Controllers

If there are multiple SCSI controllers on the computer, startup after a failure to a mirror set is much easier if you use the following guidelines:

- Use identical SCSI controllers. This minimizes problems with driver, performance, and feature set differences.
- When using duplex sets, load balance the controllers. Include the same number of drives on each controller, if possible. This step helps with performance.

When building a fault tolerance boot floppy disk for a computer with multiple SCSI controllers, use the *scsi* syntax in the BOOT.INI file. Here is a typical BOOT.INI file for starting a computer with a duplexed mirror. In this case, there are two SCSI buses, and both the primary and shadow drive are assigned SCSI ID 0 on their respective controllers. This example has two *scsi* controllers.

```
[boot loader]
timeout=30
default=scsi(0)disk(0)rdisk(0)partition(1)\FT_TEST

[operating systems]
scsi(0)disk(0)rdisk(0)partition(1)\FT_TEST="Primary Drive"
scsi(1)disk(0)rdisk(0)partition(1)\FT_TEST="Shadow Drive"
```

If the shadow drive was assigned SCSI ID 3, then the second entry should be:

```
scsi(1)disk(3)rdisk(0)partition(1)\FT_TEST="Shadow Drive"
```

If you use two different SCSI controllers in a duplexed system, you have to create and maintain two separate fault tolerance boot floppy disks, because each requires an NTBOOTDD.SYS file specific to the controller. Having two different SCSI controllers makes a trouble-free system restart in the event of a hardware failure far more difficult.

Dual Channel Controllers

Many SCSI controllers have more than one channel on the card. Even though there is a single card, it appears to the computer as two separate controllers. There is quite a bit of variation in the way different manufacturers handle multiple channels, so it is critical to test any recovery tools before bringing a Windows NT Server computer into production.

On a computer running Windows NT Server with dual channel controllers, the disk(X) parameter of the ARC path should be calculated as follows:

X= [(scsi bus/channel number) x (32)] + (SCSI ID of the disk)

For example, the Adaptec AHA-2740T/2742T has two channels: SCSI Channel A (bus 0) and SCSI Channel B (bus 1). Each channel can accommodate a SCSI bus with up to seven devices. If the shadow drive on a computer running Windows NT Server is located on the first partition on a SCSI disk with the target ID of 4, and is connected to SCSI Channel B (bus number 1) of an Adaptec AHA-2742T controller, the ARC name in the BOOT.INI file would be as shown in this example:

```
scsi(0)disk(36)rdisk(0)partition(1)\FT_TEST="Shadow on second channel"
```

If a valid ARC path cannot be found to start the computer from the shadow drive, it might be necessary to make changes in the system configuration. Many multi-channel controllers have an option in their Setup utility, or the EISA configuration utility, that enables the user to specify the channel to use for the boot device. It might be necessary to consult with the controller vendor to determine the correct configuration to be able to start Windows NT Server from the shadow drive if the primary drive or SCSI channel fails.

Creating Alternate Boot Selections for a RISC-based Computer

Since RISC-based computers boot from the system firmware, the functionality of the x86-based computer's BOOT.INI file is located in the configuration menus in the firmware.

If the primary drive in a mirror set fails on a RISC-based computer, you need to have alternate boot paths created in the system firmware. As is the case with x86-based computers, you can create a path to boot from the shadow partition, or create a path to the floppy disk drive and use a fault tolerance boot floppy disk.

RISC-based computers require that the system partition be formatted with the File Allocation Table (FAT) file system. The operating system can also be installed on this partition, but it is not a requirement. The examples in this section use the most common configuration, which is a small FAT system partition of 5–10 MB.

When the system partition and boot partition are separate partitions, but on the same physical drive, it is best to mirror both partitions. When both the system and boot partitions are mirrored, or they are the same partition, use the following method to create an alternate boot selection for the shadow partition.

Note The examples in the remainder of this section are for an Alpha AXP computer. The firmware menus that you use to get to the configuration options can be different for MIPS and PPC computers.

Creating a Path to the Shadow Partition

Start the computer. Select the options listed in the following table to get to the Boot selections menu.

Menu	Select this option:
System Boot	Supplementary menu
Supplementary	Setup the system
Setup	Manage boot selections

You should now see the Boot selections menu.

```
ARC Multiboot DEC Version 3.5-4 Thursday, 10-19-1995 10:08:43 AM

Copyright (c) 1993 Microsoft Corporation
Copyright (c) 1993 Digital Equipment Corporation

Boot selections menu:

     Add a boot selection <-------------------------------------
     Change a boot selection
     Check boot selections
     Delete a boot selection
     Dump boot selections
     Rearrange boot selections
     Setup menu...

Use the arrow keys to select, then press Enter.
```

Select Add a boot selection. The following example shows creating a path to the shadow partition where the RISC-based system partition and the boot partition are on separate partitions on the same physical disk. Both partitions were mirrored.

Note The right justified arrows (<----) in these examples indicate which selection was made, or the information that was entered.

```
                    Thursday, 10-19-1995 10:32:43 AM

Select a system partition for this boot selection:

     SCSI Bus 0 Hard Disk 0 Partition 1
     New system partition <-------------------------------------

Enter location of system partition for this boot selection:

  Select Media:
     SCSI Hard Disk <-----------------------------------------------
     Floppy Disk
     CD-ROM

     Enter SCSI bus number: 0 <-----------------------------------
     Enter SCSI ID: 2     <-----------------------------------
     Enter Partition: 1    <-----------------------------------

Enter the osloader directory and name: \os\winnt351
\osloader.exe <-----

Is the operating system in the same partition as the osloader:

     Yes
     No <-----------------------------------------------------------

Enter the location of os partition:

  Select Media:
     SCSI Hard Disk <-----------------------------------------------
     Floppy Disk
     CD-ROM

     Enter SCSI bus number: 0 <-----------------------------------
     Enter SCSI ID: 2     <-----------------------------------
     Enter Partition: 2    <-----------------------------------

Enter the operating system root directory: \winnt35   <-----------
Enter a name for this boot selection: Boot Shadow Drive <---------

Do want to initialize the debugger at boot time:

     Yes
     No <-----------------------------------------------------------
```

After entering the data, you will be back at the Boot selections menu. Select Setup menu. On the Setup menu, select Supplementary menu, and then save the changes.

When you define a new boot selection in this manner, it becomes the default boot selection. To change the default boot selection back to the primary drive, you need to return to the Manage boot selections menu and select Rearrange boot selections. You can then select the boot selection that you want to be the default.

Creating a Path to the Fault Tolerance Boot Floppy Disk

Once you have created the fault tolerance boot floppy disk, you need to create the path to it. Listed below is a sample of the configuration options used to build an alternate boot selection when the Windows NT boot files are located on the fault tolerance boot floppy disk. As in the earlier examples, the arrows indicate which selections were made or which data was entered.

```
                    Thursday, 10-19-1995 11:18:43 AM

Select a system partition for this boot selection:

     SCSI Bus 0 Hard Disk 0 Partition 1
     New system partition <-------------------------------------

Enter location of system partition for this boot selection:

  Select Media:
     SCSI Hard Disk
     Floppy Disk <-----------------------------------------------
     CD-ROM

  Enter floppy drive number: 0 <-----------------

Enter the osloader directory and name: \os\winnt351
\osloader.exe <------

Is the operating system in the same partition as the osloader:

     Yes
     No <--------------------------------------------------------

Enter the location of os partition:

  Select Media:
     SCSI Hard Disk <--------------------------------------------
     Floppy Disk
     CD-ROM
     Enter SCSI bus number: 0 <----------------------------------
     Enter SCSI ID: 2     <-----------------------------------
     Enter Partition: 2    <-----------------------------------
```

```
Enter the operating system root directory: \winnt35     <---------
Enter a name for this boot selection: Boot from floppy disk <-----

Do want to initialize the debugger at boot time:

     Yes
     No <-------------------------------------------------------
```

After entering the data, you will be back at the Boot selections menu. Select Setup menu. On the Setup menu, select Supplementary menu, and then save the changes.

Be sure to use the Rearrange boot selections menu to change the default boot selection back to the primary partition when you no longer need to use the fault tolerance boot floppy disk.

Saving Critical Information

Hardware failures or power failures can corrupt information that your computer needs to start Windows NT Server or Windows NT Workstation. To make recovery easier, there are several system directories or files that you should back up every time you make certain changes to your Windows NT configuration.

Creating an Emergency Repair Disk

A current Emergency Repair Disk is your most valuable tool in recovering information that you need to start your system. The Emergency Repair Disk is intended to provide just enough recovery to restore a system to a bootable state and is not a replacement for regular backups.

Windows NT Server and Windows NT Workstation version 3.51 come with the **rdisk** command for building and maintaining repair information. You can use this utility to update the repair information stored in the *\systemroot*\repair directory and to copy the information to a floppy disk. You can use the Emergency Repair Disk to replace corrupt system files, restore damaged or incorrect registry information, and rebuild the startup environment.

You should update the repair information and create a new Emergency Repair Disk any time you change the system configuration in any significant way. For instance, if you add or remove hardware from the system or change the disk drive configuration, you should update the repair information.

There are several points to consider concerning maintaining and using the Emergency Repair Disk:

- When using the Repair procedure in Windows NT Setup, and replacing the system registry hives, all passwords in the system return to the passwords in effect at the time the disk was created.

  Note When you restore these hives from the Emergency Repair Disk, the versions that were saved when you installed the operating system are restored. To avoid losing information, select the option to back up the Registry when you run Windows NT Backup. You can then restore these hives from your backup after the Repair process finishes.

- If you did not update the Emergency Repair Disk after configuring fault tolerance in Disk Administrator, it might be difficult or impossible to recover data on the fault-tolerant volumes.
- The Emergency Repair Disk is not a replacement for backups.

For more information, see Help for the Repair Disk utility, RDISK.HLP.

Saving Disk Partition Information

You should save configuration information about currently defined drive letters, volume sets, stripe sets, stripe sets with parity, and mirror sets each time you change any of this information.

▸ **To use Disk Administrator to save the disk configuration**

1. From the Partition menu, select Configuration.
2. From the Configuration menu, choose Save.

   A message is displayed describing what will be saved and where you should save it.
3. Insert any floppy disk with enough free space to hold the configuration information (about 512K).

   Using the fault tolerance boot floppy disk is highly recommended.
4. Choose OK to write the data to the floppy disk.

Saving Registry Information

The DISK key in the Registry contains information for fault-tolerant disk configurations, as well as CD-ROM mappings and hard disk mappings. Although the Disk Administrator saves this key when you save the disk configuration, there are times that you might want restore just the DISK key.

▸ **To use the Registry (REGEDT32.EXE) to save the DISK key**

1. Choose the DISK key from HKEY_LOCAL_MACHINE\SYSTEM.
2. Select the Save Key option in the Registry menu.

   The Save Key dialog box appears.
3. From the Drives scroll list, select the drive to which you want to save your hive.

   Saving the key to the fault tolerance boot floppy disk is highly recommended.
4. In the Directories list, select the directory in which you want to save your hive.
5. In the File Name list, assign a name to the hive (for example, disk.reg).
6. Choose OK.

Summary of Windows NT Data Recovery

When using any of the fault-tolerant disk schemes, Windows NT Server uses a device driver called FTDISK.SYS (known as FtDisk) to receive commands and respond appropriately, based on the type of fault-tolerant system that is being used. Thus, when the file system generates a request to read data from a file, the normal disk system receives the request from the file system and passes it to the FtDisk.

Windows NT Server and Windows NT Workstation provide two kinds of data recovery:

- Dynamic data recovery by using sector sparing, which is only available on SCSI drives.
- NTFS bad cluster recovery.

Windows NT Server provides additional recovery mechanisms using mirror sets and stripe sets with parity.

The following table summarizes what happens if a sector goes bad on a system running Windows NT Server. The second part of the table is also valid for Windows NT Workstation. For more information, see Chapter 5, "Volume Management and Fault Tolerance" in the book *Inside the Windows NT File System.*

Description	FtDisk installed with a SCSI disk that has spare sectors	FtDisk installed with a non-SCSI disk or disk with no spare sectors	FtDisk not installed with any kind of disk
Fault-tolerant volume	1. FtDisk recovers the data. 2. FtDisk replaces the bad sector. 3. File system doesn't know about the error.	1. FtDisk recovers the data. 2. FtDisk sends the data and bad-sector error to the file system. 3. NTFS performs cluster remapping. 4. FAT doesn't do anything about the error.	N/A
Non-fault-tolerant volume	1. FtDisk can't recover the data. 2. FtDisk sends a bad-sector error to the file system. 3. NTFS performs cluster remapping. On a read operation, data is lost. 4. FAT loses the data on both read and write.	1. FtDisk can't recover the data. 2. FtDisk sends a bad-sector error to the file system. 3. NTFS performs cluster remapping. On a read operation, data is lost. 4. FAT loses the data on both read and write.	1. Disk driver returns a bad-sector error to the file system. 2. NTFS performs cluster remapping. On a read operation, data is lost. 3. FAT loses the data on both read and write.

FtDisk recovers the data by reading it from the mirror disk (RAID 1) or recalculating the data from a stripe set with parity (RAID 5).

When the file system can't recover from an error on a non-fault-tolerant volume, the user sees the message "Abort, Retry, or Fail?"

Restoring Disk Configuration Information

If your system files, Registry information, or boot sector are corrupt, and you are unable to recover the previous startup configuration by using the Last Known Good method, you can use the Repair process in Windows NT Setup to restore your system to its initial setup state.

To repair a Windows NT Server or Windows NT Workstation installation, Windows NT Setup needs either the configuration information that is saved on *\systemroot*\repair or the Emergency Repair disk created when you installed the operating system (or you created later by using the **rdisk** command).

If your system becomes corrupt and you cannot repair it by using the Emergency Repair disk or the information in the *\systemroot*\repair directory, you must reinstall Windows NT Server or Windows NT Workstation from the original installation source.

Note Windows NT Setup and the **rdisk** command store registry information on the Emergency Repair Disk in compressed format. However, Disk Administrator does not compress the registry hives. In the event that disk information must be restored manually by using either Disk Administrator or the Registry editor, you can uncompress the SYSTEM._ file by using the **expand** command.

Using the Emergency Repair Disk

The section titled "Using the Repair Process" in Chapter 2, "Troubleshooting," of the *Installation Guide* of the Windows NT Server and Windows NT Workstation documentation sets describes the procedures for using the Emergency Repair Disk.

You cannot repair all disk problems by using the Emergency Repair Disk. For the best results, your Emergency Repair Disk needs to match the system you have installed on your Windows NT Server or Windows NT Workstation. It also needs to have current configuration information.

These are some of the problems that Windows NT Setup does or does not fix:

- It repairs bad registry data, using information on the Emergency Repair Disk.
- It restores corrupt or missing files on the system partition.
- It replaces a corrupt Kernel. The Kernel is at the core of Windows NT architecture and manages its most basic operations. The Kernel is responsible for thread dispatching, multiprocessor synchronization, and hardware exception handling.
- It replaces a bad boot sector for a FAT partition.

- The only unmountable partition that it can repair is the system partition, which is always the C drive on x86-based computers. You have to repair partitions other than the system partition by using a low-level disk utility.
- It does not replace a damaged NTFS boot sector.

Note It is not obvious how to repair unmountable volumes. If you have an unmountable volume, or need help diagnosing and fixing other disk problems, contact Microsoft Product Support Services.

The Repair process in Windows NT Setup can perform the following steps, depending upon the selections you make:

1. Runs the **chkdsk** command on the boot partition, which contains the operating system files. On x86-based computers, it also runs **chkdsk** on the system partition. (This task is optional, and must be enabled manually.)
2. Verifies that each file in the installation is good, through a checksum algorithm. If files are missing or corrupt, they are restored from the Windows NT Workstation or Windows NT Server installation media.
3. Replaces the default system and security (SAM) Registry hives by using the Emergency Repair Disk, subject to user confirmation.

Note When you restore these hives from the Emergency Repair Disk, the versions that were saved when you installed Windows NT Workstation or Windows NT Server are restored. To avoid losing information, select the option to backup the Registry when you run Windows NT Backup. You can then restore these hives from your backup after the Repair process finishes.

---

4. Reinstalls information needed for loading Windows NT Workstation or Windows NT Server (such as the boot sector, BOOT.INI and NTLDR for x86-based computers, or OSLOADER for RISC-based computers).

Note The Emergency Repair Disk is computer specific, and should only be used on the system on which it was created.

Restoring Disk Partition Information

You can use this procedure to update the Registry with current disk configuration information. Because the Emergency Repair Disk also contains the disk configuration information, you would generally use the procedure described in this section only if:

- You do not have an Emergency Repair Disk.
- You can not read the Emergency Repair Disk, or the information isn't current.
- You had to reinstall Windows NT Server or Windows NT Workstation.

To use this procedure, you need to have saved the information, following the procedures in the section titled "Saving Disk Partition Information," earlier in this chapter. Use Disk Administrator to restore the disk configuration information as follows:

1. From the Partition menu, select Configuration.
2. From the Configuration menu, choose Restore.

   A message warns you that this operation will overwrite your current disk configuration information with what was previously saved on the floppy disk. Also, any changes made during this session will be lost.
3. Insert the floppy disk containing the saved configuration information.
4. Choose OK.
5. Disk Administrator initiates a restart of your computer.

If you do not have a backup of the disk configuration information, you can search for disk configuration information for other installations and restore that information. In this case, the information might not match the disk configuration you had been using.

Use Disk Administrator to search for and restore the disk configuration information as follows:

1. From the Partition menu, choose Configuration.
2. From the Configuration menu, choose Search.

   A message warns you that this operation will overwrite your current disk configuration information with the information from a different installation of Windows NT Workstation or Windows NT Server. Also, any changes made during this session will be lost.
3. Choose OK.

   Disk Administrator scans your disk for other installations, and then displays a list of the installations.
4. Select an installation.
5. Choose OK.
6. Disk Administrator initiates a restart of your computer.

Restoring Registry Information

Both **rdisk** and Disk Administrator save the DISK key with the other information that they save. You would use this procedure only if you do not have current information on either of these media or you cannot read them.

If you have previously saved the DISK key by using the Registry Editor (see the section titled "Saving Registry Information," earlier in this chapter), then you can restore the DISK key with the following steps:

1. Select the DISK key from HKEY_LOCAL_MACHINE\SYSTEM.
2. Select the Restore option in the Registry menu.
3. From the Drives scroll list, choose the drive on which the DISK hive is located.

   If you are restoring the hive to a remote computer's Registry, the C drive designation that appears in the Drives scroll list refers to the C drive on the remote computer.
4. In the Directories list, select the directory in which the hive is located.
5. In the File Name list, select the correct filename.
6. Choose OK.

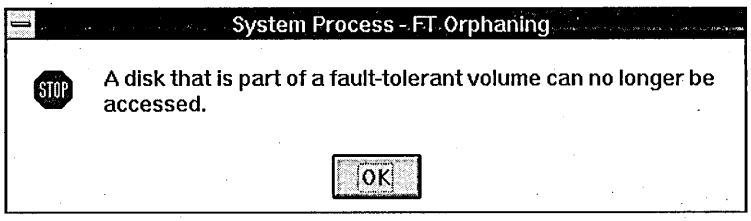
Recovering a Fault-Tolerant Volume Set

The process of error detection and recovery for software fault-tolerant sets is very similar for both mirror sets and stripe sets with parity. The exact system response to the problem depends on when the problem occurred. For recovery of a hardware fault-tolerant set, see the documentation for the controller that you are using.

When a drive that is part of a mirror set or a stripe set with parity fails during normal operation, it becomes an orphan. When FtDisk determines that a drive has been orphaned, it directs all reads and writes to the other drive in the set.

It is important to note that the process of orphaning a partition does not occur during a read, only during a write, because the read cannot possibly affect the data on the disks, so performing orphan processing would be superfluous.

The following error message is displayed:



The operating system should continue to work normally. Users accessing resources over the network should not be affected.

You should back up important data immediately, since the volume set is no longer fault tolerant. Use a new tape for backup, not an existing tape. You should replace the failed drive and rebuild the mirror set or stripe set with parity as soon as possible.

During system initialization, if the system cannot locate a partition in a mirror set or a stripe set with parity, it logs a severe error in the event log, marks the partition as an orphan, and uses the remaining partition(s) of the mirror set or stripe set with parity. The system continues to function by using the fault-tolerant capabilities inherent in such sets.

Recovering a Mirror Set

If the primary drive of the mirror set fails, and it contains the system or boot partition, always power down the failed drive or remove it from the SCSI bus before attempting to boot from the shadow partition. If the operating system tries to access the failed drive, the boot might fail with a STOP blue screen.

When the failed drive contained your system partition, you need to pay special attention to what drive you use to rebuild the mirror. If your primary and shadow drives were not identical when you first created your mirror set, you should replace a failed primary drive with one identical to the original primary disk when you rebuild the primary partition. If you plan to temporarily use a disk that is not identical to the original primary disk when rebuilding your mirror set, you might need to boot from the fault tolerance boot floppy disk until you obtain and rebuild your permanent replacement disk.

Use Disk Administrator when you are ready to replace the failed member of the mirror set, as follows:

1. Open Disk Administrator and break the mirror.

   You must first break the mirror-set relationship to expose the remaining secondary partition as a separate volume. This step prevents problems when restarting the system.
2. The remaining, working member of the mirror set receives the drive letter that was previously assigned to the complete mirror set. The orphaned partition receives the next available drive letter, or whatever letter you want to assign.

You can shut down the system and replace the failed drive. The failed drive can be replaced with any drive that is the same size or larger. It is a good idea to use a drive as similar to the remaining drive as possible. This is most important if the mirror set contains the system partition.

Note When you move or replace a drive that was at the end of a SCSI bus, be sure that you terminate only the drive that is now at the end of the bus.

Follow these steps to ensure a trouble-free system restart:

1. Perform a low-level format of the new drive on the same model controller that will be used with the new drive.

   This step eliminates any possibility of drive translation problems.

   If the failed drive was the shadow drive, use the same SCSI ID as the failed drive.

   If the failed drive was the primary drive, you might want to swap SCSI IDs, so that the remaining drive becomes SCSI ID 0. When the failed drive contained the system or boot partition, only switch SCSI IDs if both the drives in the mirror set were identical.

2. Reboot the system.

   The computer should start directly from the hard drive if:

   - You have set up your computer as described in the earlier section titled "Creating a Fault Tolerance Boot Floppy Disk for Recovering Boot Partitions."
   - You have tested your recovery procedures to be able to reboot following any failure scenario.

   If this is not the case, start the computer from the fault tolerance boot floppy disk.

3. Once you have restarted the computer, follow the procedure in the earlier section titled "Creating a Mirror Set" to rebuild the mirror.

   This step requires a second reboot of the system to initialize the mirror.

4. After the mirror initialization is complete, make sure all recovery tools are updated, as described in the section "Saving Critical Information," earlier in this chapter.

   You should test your fault tolerance boot floppy disk to be sure you can start the system from either drive. You should set the active partition on the shadow drive, if this has not already been done.

   It would be a good idea to have a backup drive of the same type available.

Recovering a Stripe Set With Parity

When a member of a stripe set with parity is orphaned, you can regenerate the data for the orphaned member from the remaining members. Select a new area of free space that is the same size as, or larger than, the other members of the stripe set with parity. Then choose the Regenerate command from the Fault Tolerance menu. When you restart the computer, the fault-tolerance driver reads the information from the strips on the other member disks, and then recreates the data of the missing member and writes it to the new member.

To regenerate a recoverable stripe set with parity

1. Select the recoverable stripe set.
2. Select an area of free space of the same size or larger.

   However, if the stripe set failure is due to a power failure or cabling failure on a single device, you can regenerate within the orphaned member of the original stripe set once the hardware state is restored.
3. From the Fault Tolerance menu, choose the Regenerate command.
4. Quit Disk Administrator and restart your computer.

The regeneration process occurs in the background. In Disk Administrator, text associated with stripe set with parity is red until regeneration is complete.

Note In the Disk Administrator display, information about the partition is not updated automatically. If you click on the partition, the information for that volume updates. You can also check the Event Log to know when the regeneration is finished.

You can receive the following error message when attempting to regenerate an orphaned drive:

```
The drive cannot be locked for exclusive use...
```

This error occurs if Disk Administrator is not allowed exclusive access to the stripe set with parity. This could be because the page file, or some other system service, like Microsoft SQL Server or Microsoft Systems Management Server, is accessing the drive. You must temporarily shutdown these services and relocate the page file to regenerate the stripe set with parity.

Using FTEdit to Update the Registry

The FTEdit utility in the *Windows NT Resource Kit* aids in the recovery of fault-tolerant volumes. FTEdit can rebuild any kind of fault-tolerant set, including stripe sets, volume sets, stripe sets with parity, and mirror sets.

Note Although stripe sets and volume sets are not fault tolerant, the operating system considers stripe sets, volume sets, stripe sets with parity, and mirror sets all to be virtual volumes. You can recover and rebuild information for all of these sets in the same way. However, rebuilding a mirror set with FTEdit is a waste of time, since the mirror will have to be regenerated anyway, so you are better off simply breaking the mirror using Disk Administrator, and then recreating it.

Because stripe sets and volume sets are not fault tolerant, Windows NT Server and Windows NT Workstation perform orphan processing differently than for stripe sets with parity and mirror sets. If one of the partitions fails, the entire volume is no longer usable. If one of the partitions in a volume set or stripe set cannot be located, all the partitions are marked as orphans. In Disk Administrator, the partitions are displayed as Unknown.

All data about the configuration of fault-tolerant volumes is contained in the Registry key HKEY_LOCAL_MACHINE\SYSTEM\DISK. This disk configuration information is stored in binary format, so it is not practical to edit it manually. When the Registry information is corrupt or missing, and no backups are available, you can use FTEdit to build the Registry information and allow the operating system to read the volumes.

For example, if the disk containing the operating system becomes unusable, or the machine fails, you might want to move the drives containing fault-tolerant volumes to another computer. Alternatively, you might decide to reinstall the operating system on a new drive. In either case, the operating system does not have any information about the fault-tolerant volumes.

The operating system has information about which drives are members of fault-tolerant sets, but it cannot distinguish between a stripe set, a stripe set with parity, a volume set, or a mirror set without the Registry information. In this situation, when the volumes are displayed in Disk Administrator, the file system might be displayed as Unknown, and there will be no drive letter assigned to the volumes. This is to prevent writing to the volume, which could corrupt it.

For more information about using FTEdit, see FTEDIT.HLP in the *Windows NT Resource Kit.*

The remainder of this section describes using FTEdit to build Registry information for a stripe set with parity that has been moved to another computer. (Disks 1, 2, and 3 are the disks that have been moved. Each disk is 519 MB and has one partition.) Both computers are running Windows NT Server build 1057.

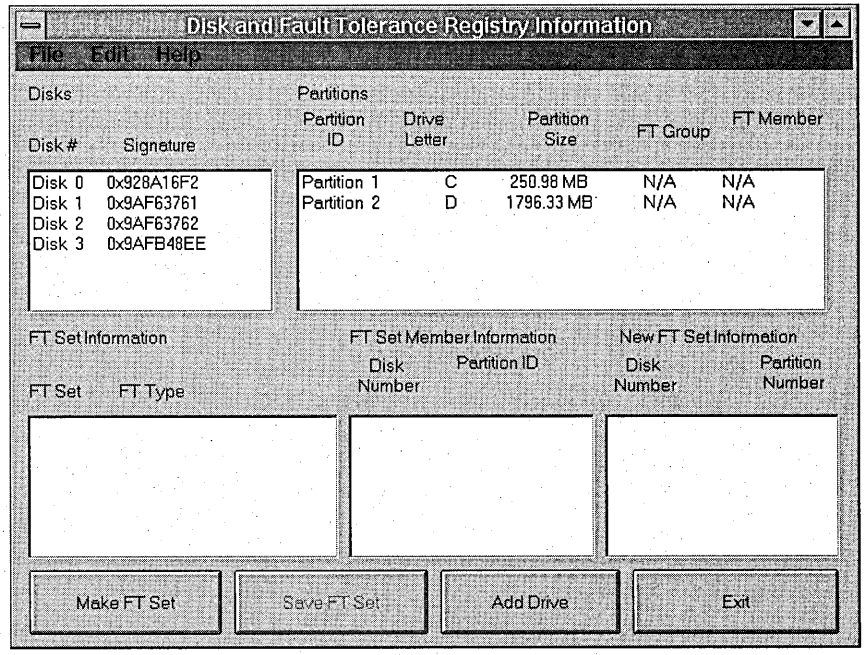
1. Open Disk Administrator, so that you can update the disk configuration information.

   Disk Administrator needs to store the disk signature and other basic drive information before the drives can be recognized.

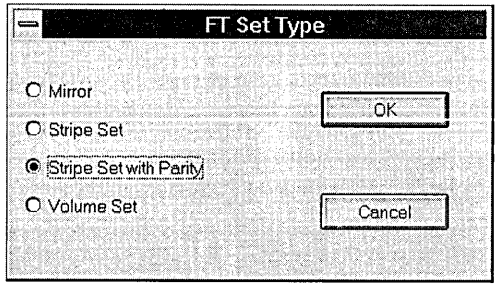
   Because the DISK key only exists on a computer if you have run Disk Administrator, a newly installed system will not have a DISK key. The information in the DISK key must be generated by Disk Administrator. When you run Disk Administrator for the first time on the computer, it creates the DISK key, with information about the drives and partitions on the computer. Close Disk Administrator after it has created the key.

2. Open FTEdit.

   The following dialog box is displayed. It contains the information about disk 0 in the Partitions list box. Disk 0 is two gigabytes, and contains two partitions.

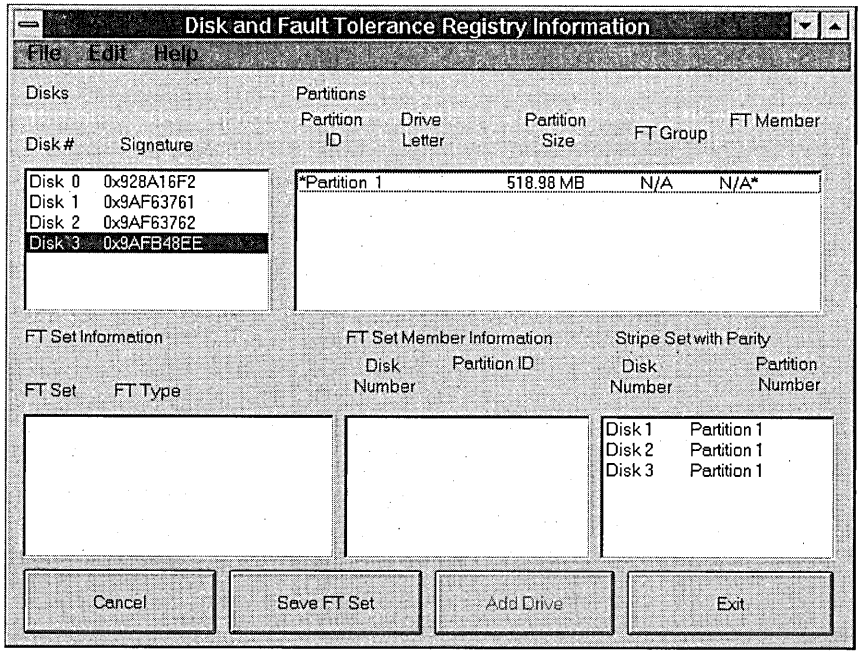


3. From the Edit menu, select Create FT Set. In the FT Set Type dialog box, select the radio button for the type of fault-tolerant volume you are building. In this example, select Stripe Set with Parity.

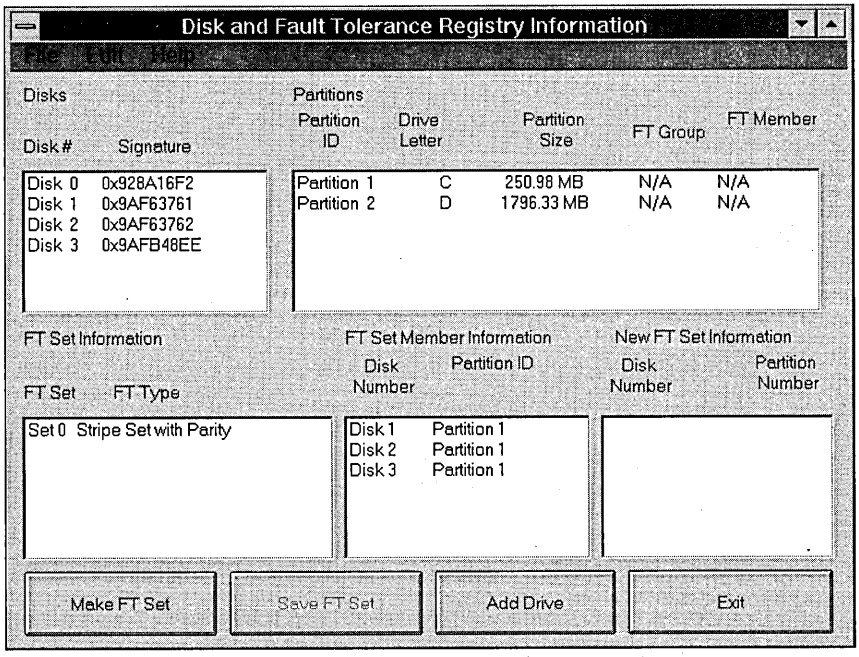


4. In the Disks list box, select the first drive that will be part of the stripe set with parity. Available partitions are displayed in the Partitions list box on the right.

   Select the partition from each drive that will make up the stripe set with parity, and double-click on it. The disk and partition information will be displayed in the list box on the lower right. The title of this list box reflects the type of fault-tolerant volume being built. The next screen shot shows the dialog box after you have selected and double-clicked Disk 1, Disk 2, and Disk 3.



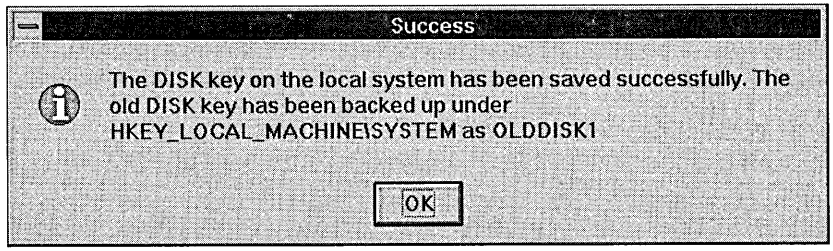
5. Select the Save FT Set button. The information is transferred to the FT Set Information list box. Check to make sure the information is correct. FTEdit should look like the following illustration:



The Partitions list box shows the information for Disk 0.. The disks that are used in creating the new fault-tolerant set are those listed in the FT Set Information and FT Set Member Information list boxes.

6. From the Edit menu, choose Save Changes to System.

   The following message box is displayed:



7. Exit FTEdit, close any open applications, and shutdown and restart the system.

   You need to do this to get the new information loaded into the Registry.

8. After restarting the system, open Disk Administrator.

   The reconstructed fault-tolerant set is displayed. If the volume was shut down dirty, the fault-tolerant set might be displayed as Initializing in the Disk Administrator status bar. If the volume was functional before the disks were moved, and the correct fault-tolerant information was entered in FTEdit, the data on the volume should be intact.

9. From the Disk Administrator Tools menu, select Drive Letter. In the Assign Drive Letter dialog box, select Assign drive letter, and then choose a drive letter for the new volume.

   The volume is now accessible from File Manager and the rest of the operating system.

10. After confirming that the volume is correctly configured and is accessible, update your recovery tools with the new configuration information, as described in the section "Saving Critical Information," earlier in this chapter.

11. If there is any data on the fault-tolerant volume that has not been recently backed up, back it up immediately.

These same steps can be used to recover volume sets, and stripe sets. Since FTEdit is really editing the binary information in the registry, it cannot recover damaged or corrupt data. To successfully use FTEdit to build Registry information, you must know the following information:

- Which disks contain fault-tolerant volumes
- Which partitions belong to the fault-tolerant volumes

If your configuration involves multiple fault-tolerant volumes on several different drives, you might face a very difficult job remembering which partitions belong to which fault-tolerant volumes. For this reason, it is a good idea to keep fault-tolerant volumes simple, and to keep recovery information as up to date as possible.

CHAPTER 9

Printing

This chapter replaces information originally presented in Chapter 6, "Printing," of the *Windows NT Resource Guide*. Much of that information is included here in an updated and reorganized format.

New and additional information about troubleshooting is presented along with a Question and Answer section that focuses on questions about using either Windows NT Workstation or Windows NT Server as a print server for cross-platform network printing.

The chapter begins by defining key printing terminology and concepts. The first section provides general information about print jobs and print devices that provide a foundation for understanding printing. Next are sections that describe the print server services and the various print clients that can be supported by a Windows NT Server or Windows NT Workstation print server. Following are sections that provide information about the software modules contained in the print spooler. The spooler sections are followed by sections on using Print Manager, establishing printers, managing print forms, implementing print security and print auditing, and print troubleshooting. The final section of this chapter covers questions and answers.

For additional information about printing with Windows NT Server or Windows NT Workstation, you may also refer to the following documentation.

Documentation Source	Descriptions
Microsoft Knowledge Base	An effective and consistent resource for troubleshooting printing problems. Specific printing related articles used as input in this document include Q100346 and Q132460.
Microsoft Technet	Microsoft White Papers, TechEd, and other articles are available on the Microsoft Technet CD.
Support Fundamentals for Windows NT, Chapter 13, "Printing with Windows NT"	A Microsoft Press self-paced training manual.

Printing Terms

In Window NT, a *print device* refers to the actual hardware device that produces printed output. A *printer* is the data structure to which applications send print jobs to and is the basic object managed by Print Manager.

Print device resolution is measured in *dots per inch* (DPI). The greater the DPI, the better the resolution.

In Windows NT terminology, a *queue* is a group of documents waiting to be printed. In the NetWare and OS/2 environments, queues are the primary software interface between the application and print device: users submit print jobs to a queue. However, with Windows NT, the printer is that interface—the job is sent to a printer, not a queue.

Print jobs are classified into *data types* based on what modifications the spooler should make to the job (if any). For instance, using one data type implies that the spooler should not modify the job at all; using another data type implies that the spooler should add a form feed to the end of the job.

The print *spooler* is a collection of DLLs that receives, processes, schedules, and distributes print jobs.

Spooling is the process of writing the contents of a print job to a file on disk. This file is called a *spool file*. *Despooling* means reading the contents from a spool file and sending those contents to a print device.

Rendering means converting a print job from whatever commands the application uses to describe output into commands that can be interpreted by a print device.

A *print server* is the computer that receives print jobs from clients. A print server can be a special hardware device that connects a print device to the network with a net tap on one side and a parallel or serial port on the other side.

Network-interface printers are print devices with their own network cards; they need not be adjacent to a print server, because they are directly connected to the network.

The terms *workstation* and *print server* refer to two different roles in over-the-network printing. The workstation is the computer that sends print jobs over the network; the server is the computer that receives print jobs. Do not confuse these terms with Windows NT Workstation and Windows NT Server. Both Windows NT Workstation and Windows NT Server can operate in either workstation or print server roles. However, because Windows NT Workstation is limited to 10 connections from other computers, it does not make a practical print server, except in small-network situations. Unless otherwise specified, all topics in this chapter apply equally to both Windows NT Workstation and Windows NT Server.

A print *client* is any application that creates a print job. The client can be an application on the local computer or on a computer on the network.

Print Server services run on the print server and receive print jobs from clients. The print server services alter the job if necessary, and pass the print jobs to the spooler. Print server services are designed to support print jobs from various clients. For example, Services for Macintosh (SFMSRV) receives print jobs from Macintosh clients.

About Print Jobs and Print Devices

This section provides a brief overview about print jobs, print-job data types, and print devices. These sections provide a foundation for discussions later in this chapter, particularly the sections concerned with the print server services, the spooler, and the print processor modules contained within the spooler.

Print Jobs

It may help you understand printing if you realize that print jobs are not data; they are source code that contains both data and directions for print processing of the data. An application, (the print client), creates print jobs. For example, Microsoft Word 6.0, combines data objects such as text, fonts, and graphics, with information from a print device driver to create source code, the print job. The print job is sent to a targeted print device and interpreted by the print device to produce the desired hardcopy output.

Each print job is assigned a data type by the application. When the print job is sent to a print device by using a print server, the data type indicates how the print server should alter the job.

The data types that can process are summarized in the following table.

Table 9.1 Print Job Data Types

Data type	Print server services receives as...	Spooler and print processor actions
RAW	Job that is already fully rendered.	Does not alter the job at all.
RAW [FF Auto]	Job that is simple text sent by an application that does not add a form feed to the end of its jobs.	Adds a PCL (HP printer control language) command to produce a form feed at the end of the job.

Table 9.1 Print Job Data Types *(continued)*

Data type	Print server services receives as...	Spooler and print processor actions
RAW [FF Appended]	Job that is simple text sent by an application that does not add a form feed to the end of its jobs.	Adds a PCL command to produce a form feed at the end of the job, unless check indicates a form feed exists.
TEXT	Job that is simple text. This data type is most useful with print devices that do not accept simple text as a valid print job, such as PostScript print devices or plotters.	Uses GDI and the printer driver to create a print job that prints the original job's text on the target print device.
Journal (NTJNL1.000)	Job sent from a Windows NT-based application running locally on the print server and that is halfway rendered into print device commands.	Uses the graphics engine and the printer driver to finish rendering the job into printer commands.
PSCRIPT1	Job is PostScript code from a Macintosh client, targeted for a non-PostScript print device.	Interprets the PostScript code, creating a bitmap that the GDI32 and the printer driver can convert into the language of the target print device.

Print Devices

Print devices are electronic computing devices that produce hardcopy output by rendering the source code in the print job.

Print devices have their own input and output channels; jobs can be directed to input channels such as parallel or serial cables, or network adapters. Output can be produced on different forms or media, including paper, film, or fabric.

Print devices have their own internal processors which can be proprietary or general-purpose, such as the Motorola 680xx-series chips in Apple LaserWriter devices. Incoming data is stored in the print device RAM, this can be a few bytes or a hundred megabytes. Each print device has firmware that implements a programming language interpreter, such as PostScript, PCL, or HP-GL/2 interpreters.

Printer Drivers

Printer drivers are the software that enables an application or print server service to interface with the variety of available print devices, regardless of the device type, model, or programming language interpreter. In general, print drivers are composed of three separate files that work together as a printer driver unit. These printer driver component files are:

- A *printer graphics driver* responsible for rendering device driver interface (DDI) commands from the graphics engine to commands that a print device can understand. Each graphics driver handles different printer languages. For example, a PSCRIPT.DLL deals with the PostScript printer language.
- A *printer interface driver* that includes the user interface you see when you configure a printer in Print Manager.
- A characterization data file that is used by the other two components of the print driver as needed. It provides information about the configuration capabilities of a specific make and model of print device, including what resolutions the print device is capable of, whether it can print on both sides of the page, and what paper sizes it can accept.

These three files work as a unit. For example, when you create a new printer in Print Manager, the interface driver enables you to pick the default resolution. It displays the proper choices because it queries the characterization data file for this information. When you print, the graphics driver queries the interface driver to find what resolution you chose, so that it can create the right printer commands to generate the resolution you specified.

The variety of available print devices can be classified as one of three types; raster, PostScript, or plotter. To support these three classes of print devices, Windows NT provides the following generic printer drivers:

- Universal printer driver
- PostScript printer driver
- HPGL/2 plotter driver

Each type of driver is discussed in the following sections.

Universal

The Universal printer driver, is sometimes referred to as the *raster* driver. It is an improved version of the Windows 3.1 driver that supports raster-graphics printing.

It includes support for scaleable TrueType fonts, device fonts, compression-run length encoding (RLE), and Tag Image File Format (TIFF) version 4.0. It also includes mechanisms that provide for smaller, more efficient bitmaps. These mechanisms include ignoring whitespace and supporting *rules*, which are printable rectangles extracted from the bitmap and sent to the printer as a separate command, as supported by Hewlett-Packard LaserJet and compatible print devices.

- The component files of the Universal printer driver are:
- RASDD.DLL, the printer graphics driver for printer languages based on raster (bitmap) images, including PCL, and most dot matrix printer languages.
- RASDDUI.DLL, the printer interface driver.
- Raster minidriver, the characterization data file.

An important, new facet of the Windows NT 3.51 Universal driver is a generic, text only (TTY) driver. This driver only prints the subset of a document that has been formatted as a TTY-specific font, such as Courier 12. This driver ignores all other text and all graphics. This is different from the 16-bit Windows TTY driver, which accepts all text, regardless of font.

This difference is significant for users whose applications do not process text based on fonts. For example, a user might use a proprietary print device and associated software that generates as output the commands that the print device interprets. If the proprietary device and software is designed for a Windows 16-bit application, it probably does not format the output in any particular font, because the designers of the proprietary device knew that the 16-bit Windows TTY driver passes any text to the print device. When the proprietary print device and associated software runs under Windows NT and the Windows NT TTY driver, the print device is unlikely to receive any output, because the TTY driver ignores anything that is not explicitly formatted in the Courier 12 font.

PostScript

The Windows NT PostScript driver supports Adobe version 4.0-compatible PostScript printer description (.PPD) files. (Windows NT does not use the .WPD or .MPD files used by Windows 3.1). This driver supports key features, including binary transfer compression from Level II, resolution, and paper source.

The component files of the Windows NT PostScript printer driver:

- PSCRIPT.DLL, the printer graphics driver.
- PSCRIPTUI.DLL, the printer interface driver.
- *x*.PPD, the PostScript printer description (PPD) characterization data file.

Note The Windows NT PostScript driver supports Adobe version 4.0-compatible .PPD files. *x*.PPD files are the only printer driver files that are generally binary-compatible across processors and platforms.

HPGL/2 Plotter

The Windows NT plotter driver supports a variety of plotters that use the HPGL/2 language. HPGL is not supported. There is a significant difference between HPGL and HPGL/2. The output from the Windows NT plotter driver requires a plotting device that can process all of the enhancements built into the HPGL/2 language.

Each of the printer drivers discussed in this and the preceding sections Universal, Postscript, and Plotter, is really a set of driver files that work together. Each printer driver includes a graphics driver, an interface driver, and a characterization data file.

The component files of the Windows NT plotter driver are:

- PLOTTER.DLL, the printer graphics driver.
- PLOTUI.DLL, the printer interface driver.
- *x*.PCD, the characterization data file.

Cross Platform Printer Drivers

Printer drivers are generally not binary-compatible across hardware-processor platforms.

Printer drivers must be installed on the Windows NT Workstation or Windows NT Server print server for each client hardware platform. For example, when *x*86-based clients running Windows NT Workstation are served by a Digital Alpha AXP-based Windows NT Server print server, you must install *x*86 printer drivers on the Alpha AXP-based print server. This is because the print client first attempts to use a local printer driver, if no local printer driver is available, the x86 printer driver from the Windows NT print server is downloaded. The driver image is copied to local memory and used to create the client print job.

Platform-specific printer drivers are available for the following:

- Intel x86-basedcomputers.
- MIPS RISC-based computers.
- Digital Alpha AXP-based computers.

About Network Printing

Windows NT is the first operating system that truly supports remote printing. There is no need to install a printer driver manually on the local computer before printing with Windows NT. For Windows NT users, print resources seem to be provided automatically from each application, and the Windows NT printing architecture enables users to simply "point-and-print."

How Windows NT prints a document is somewhat more complicated than the user's "point-and-print" perspective. The real power of the Windows NT printing architecture lies in the modules that are transparent to the user.

Managing networks in which a Windows NT-based computer is the print server requires an understanding of both the print server services supplied with Windows NT and the various types of clients that can be supported by a Windows NT print server. The following two sections discuss the Windows NT print server services and Windows NT print clients.

Print Server Services

Print server services are the software on the Windows NT print server that receive print jobs from a print client and send the job to the spooler by calling the spooler application program interfaces (API). The client can be located on a computer located somewhere on the network or a locally run application on the Windows NT print server computer itself.

Both Windows NT Server and Windows NT Workstation provide the following:

- Windows NT Server Service
- TCP/IP Print (Line Printer Daemon (LPD)) Service

Only Windows NT Server provides a print server service for Macintosh clients, named Services For Macintosh (SFM).

Locally run print client applications on the Windows NT print server use these additional software modules:

- Windows NT Virtual DOS Machine (NTVDM)
- Graphical Device Interface 32-Bit (GDI32)

Note The NTVDM and GDI32 modules are only used for locally run print client applications.

The following table lists each print server service and the type of print client processed by the service.

Table 9.2 Windows NT Print Server Services

Service	Registry Name	Supported Clients
Windows NT Server service	SRV.SYS	MS Network clients using NetBIOS redirectors on MS-DOS, Windows, Windows for Workgroups, or Windows NT computers.
Services for Macintosh	SFMSRV.SYS	AppleTalk clients on Macintosh computers.
TCP/IP Print service	LPDSVC.SYS	LPR (line printer) clients on UNIX or Windows NT computers where the TCP/IP protocol is used.

The Windows NT print server services determine whether the Windows NT print server spooler should process and alter in any way the print jobs received from client applications. This concept is often misunderstood, because it is not generally known that the print server can alter the print job. Even less understood is the programmatic logic used to alter print jobs.

Each print server service uses different programmatic logic and makes different assumptions about print jobs from their supported print clients. For detailed explanations of how and when print job alteration occurs and how to troubleshoot these problems see the Microsoft Knowledge Base article "Troubleshooting Windows NT Print Server Alteration of Print Jobs" (Q132460).

Windows NT Server Service

The Windows NT Server service receives jobs from applications and computers that are using the MS Network client and NetBIOS redirectors. This includes print clients using any of the following:

- LAN Manager
- MS Network Client (often used on clients running MS-DOS alone or MS-DOS with Windows Versions 3.0, 3.1 or 3.11)
- Windows For Workgroups
- Windows NT

The Windows NT Server service does not set the default data-type value when it submits the print job to the spooler. Instead, the spooler uses the data type specified in the Default Datatype in Print Manager. This Default Datatype can be changed using Print Manager. (See the section titled "Print Jobs" earlier in this chapter for details on print-job data types.)

To change the way the spooler alters print jobs from MS Network clients:

1. Open Print Manager.
2. Select the printer you want to configure on the Printer menu.
3. Select Properties from the Printer menu.
4. Select Details to display the dialog box.
5. Select the Default Datatype list box and choose the appropriate data-type.

Note If you select NTJNL1.00 or PSCRIPT1, the print spooler ignores this data and, instead, uses the RAW data type to process print jobs. Detailed information on print spooler processing of data types is provided later in this chapter in the "Print Processors" section.

TCP/IP Print Service

The TCP/IP Print service is generally referred to as LPD, which stands for line printer daemon. The TCP/IP Print service receives print jobs from line printer (LPR) utilities running on client systems. LPR clients are often UNIX systems, but LPR software exists for most operating systems, including Windows NT.

The Windows NT TCP/IP Print service receives jobs from LPR clients and submits them to the spooler. LPR clients always send a control file containing administrative information with each print job. The Windows NT TCP/IP Print service assigns a data type based on the control commands in that control file.

If the client sends the **f**, **o**, or **p** control command, the Windows NT TCP/IP Print service assigns the TEXT data type to the print job. If the client sends the **l** control command, the Windows NT TCP/IP Print service assigns the RAW data type to the print job. These control commands are documented in the LPR specification, Request For Comment (RFC)1179, sections 7.17 through 7.29. For detailed information about RFC 1179, refer to the Windows NT Knowledge Base article, "Text of RFC1179 Standard for Windows NT TCP/IP Printing", reference number Q124734.

Because the Windows NT TCP/IP Print service assigns a data type explicitly, the Default Datatype value has no effect on print jobs received by the Windows NT TCP/IP Print service. To change the behavior of the TCP/IP Print service, you must reconfigure the LPR client to send a different control command with the print job. Again notice that changing the Default Datatype value using Print Manager does not effect the behavior of print jobs received by the Windows NT TCP/IP Print service.

Services For Macintosh Service

The Services For Macintosh service, also referred to as SFMSRV, receives print jobs from Macintosh clients. From the client's perspective, a printer shared through this service looks just like a share on an AppleShare server, or a standalone AppleTalk print device.

The Services for Macintosh service assigns the RAW data type to all jobs targeted to PostScript printers, and also assigns the PSCRIPT1 data type to all jobs targeted for non-PostScript print devices. There is no way to override this logic. For additional information about processing of Macintosh print jobs refer to the section "Services for Macintosh Print Processor (SFMPSPRT)" later in this chapter.

Windows NT Virtual DOS Machine (NTVDM)

MS-DOS-based applications run in a special Win32-based component referred to as a virtual DOS computer (VDM). The NTVDM component translates MS-DOS operating system calls into calls used by the WIN32 subsystem.

If you print from a locally run MS-DOS-based application, or from a Windows NT command-line utility that sends data to a printer port, NTVDM also receives the job. What happens next depends on whether the Windows NT redirector or spooler manage the port to which the application printed. First, NTVDM queries the redirector to find out whether the redirector is managing the target port. This would be true if a NET USE LPT1:\\SERVER\SHARE command had been previously issued. If so, the redirector takes control of the print job, and none of the spooler options affect it. If the redirector is not managing the port, and a printer defined in Print Manager is printing to that port, the job goes to that printer, and that printer's Default Datatype value and its spooling options are in effect. If neither the redirector nor any printers are managing the port, the job goes to the port device driver, unaltered.

For example, suppose one printer in Print Manager prints to COM1, and none of the printers in Print Manager print to LPT1. Also, suppose that you issue a NET USE command to redirect output from LPT2 to a network print share. If an MS-DOS-based application prints to LPT2, the job is sent to the network print share. If the application prints to COM1, NTVDM submits the job to the printer that prints to COM1, and that printer's Default Datatype is used. If the application prints to LPT1, NTVDM submits the job directly to the parallel port device driver, and no print job alteration occurs.

Windows NT Graphical Device Interface (GDI32) for 16-Bit Applications

On a Windows NT computer, Windows 3.x-based (16 bit) applications are supported by a Graphical Device Interface 32 (GDI32) engine. The GDI32 engine translates print and display application programming interface (API) calls to 32-bit WIN32 services.

The GDI32 graphics engine (GDI32.DLL) is the printing component that provides What You See Is What You Get (WYSIWYG) support across devices. The graphics engine interfaces with Windows-based applications through the Windows Graphics Device Interface (GDI) and with printer drivers through the Device Driver Interface (DDI).

When a Windows-based application creates a print job, it describes the output it wants in a series of GDI commands. The GDI32 graphics engine is the component that translates these GDI commands into the DDI commands understood by components like printer drivers and print processors.

The GDI32 graphics engine sends commands to the printer driver about the characters, fonts, locations, and point sizes to print and when. The GDI32 graphics engine queries the printer driver to identify the capabilities of the print device, including supported fonts. Using this information, the GDI32 graphics engine uses other DDI commands to specify the positioning of each character in the document by the print device. The GDI32 graphics engine also uses DDI commands to define how the printer should draw and fill graphics, and how to manipulate and print bitmaps.

The GDI32 graphics engine provides services to the printer driver, including compatibility with the environment subsystem (for example, MS-DOS or OS/2) and performance optimization, caching, client-server communications, and ANSI-to-Unicode conversion.

The GDI32 graphics engine calls the printer driver and provides information about the type of printer needed and the data type used. In response, the printer driver provides the graphics engine with the printer's fully qualified path name for the printer and printer-setting information. This information is passed to the spooler.

The GDI32 graphics engine sends messages to the Windows NT print spooler to determine which data type to use when the print job is sent to the print spooler. If the specified data type is RAW, the graphics engine calls the printer driver to render the DDI calls. If the data type is journal, the graphics engine writes a journal file and does not call the printer driver to render the DDI calls. When the graphics engine passes the journal file to the spooler, spooling happens quickly, because journal files are small and there is no wait for printer-specific rendering. Rendering is done later as a background process. Although journal files contain DDI calls rather than printer commands, they are device-dependent.

Journal files differ from metafiles. Windows NT does not spool metafiles, because they are device-independent and thus do not translate reliably to an individual printer's page layout. Metafiles are pictures, not pages. In addition, metafiles often contain a list of acceptable font and color substitutions for a document. For WYSIWYG accuracy, such color and font substitutions are unacceptable. In contrast, use of journal files guarantees that Windows NT provides true reproduction of spooled documents.

Journal files are concise and precise. They only contain calls that make a difference. For example, some applications add hundreds of unnecessary or redundant instructions for creating a graphic. The journal file includes only those necessary to draw that picture. Journal files are tuned for a particular device; they are not device-independent. For example, a journal file created for a 150-DPI LaserJet® printer cannot print on a 300-DPI LaserJet printer. A journal file is created to play back on a specific device and therefore is tuned for the device's specific coordinate space, color space, bits-per-pixel, and fonts.

Print Clients

Print clients are applications, including applications running on the local computer; workstations on the network that send jobs to a Windows NT print server; and Print Manager application, that use a Windows NT print server.

When a client application creates a print job on a network client computer, the print job is sent to a Windows NT print server computer. The Windows NT print server services receive the print job as an input/output (IO) request that must be directed to the appropriate, lower layer, system process.

When the print job is sent from a MS Network client, the appropriate process is the Windows NT print spooler. For example, a user on an MS Network client, such as Windows for Workgroups or the Microsoft Network Client 3.0, creates a print job. The Windows NT Server service receives the job, processes the IO request, and submits the job to the Windows NT Spooler for print processing.

When the print job is sent from an LPR client, the Windows TCP/IP Print service (also referred to as the LPD Service) receives the job, processes it, and submits it to the Windows NT spooler.

The appropriate print server service determines whether or not the spooler should alter the job (and if so, how) and assigns the job the corresponding data-type value. The appropriate print server service can also leave the data-type value blank and let the spooler apply a default value.

Each of the print services, SRV.SYS, SFMSRV.SYS, and LPDSVC.SYS, uses different logic to determine how the job should be altered. The following sections describe protocols, general architecture, and process logic for each of the following types of print server client:

- 16-bit Windows network clients
- MS-DOS network clients
- Windows NT network clients
- Windows NT local clients
- UNIX clients
- Macintosh clients

The following illustration shows both local and network client printing.

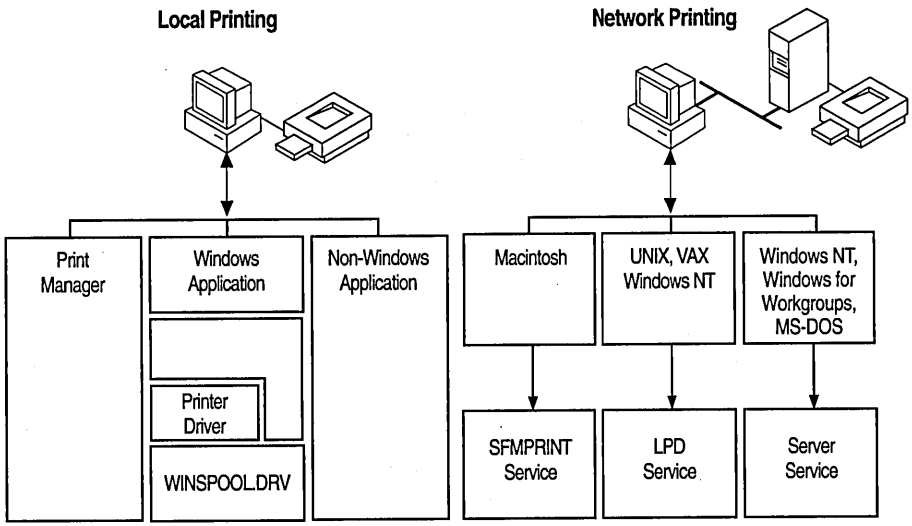


Figure 9.1 Local and Network Print Clients

16-bit Windows Network Clients

There is not much difference between the different versions of the 16-bit Windows platforms. They can all support printing from MS-DOS-based applications and from Windows-based applications, by using the 16-bit printer driver installed on the computer. They typically send jobs to a Windows NT print server by using one of the following MS Network client redirectors:

- LAN Manager
- Microsoft Network Client 3.0 for Windows
- Windows for Workgroups built-in redirector

The MS-Network redirectors send jobs by using the following protocols:

- NetBEUI
- NWLink
- TCP/IP

The Windows NT print server receives the job through the Windows NT Server service, which typically does not alter the print job. These clients might also run third-party software for sending jobs to other print servers, such as LPR software for sending print jobs to UNIX systems. This software is often able to send jobs to Windows NT as well, in which case, the TCP/IP Print Server (LPD) service receives the print job and alters it.

The following are common problems that occur in this situation:

- Local printer drivers are not correctly configured.
- Client application has a program bug.
- MS-DOS-based application tries to manipulate the parallel port directly in a network printing environment.

MS-DOS Network Clients

MS-DOS clients usually use an MS Network client redirector to send print jobs to Windows NT print servers. They typically use the NetBEUI or TCP/IP protocol, but may also print with third-party software like an LPR utility. Some MS-DOS-based applications do not use a printer driver; they create print jobs that contain only raw ASCII text. Unlike Windows clients that share the same driver for a particular print device, those MS-DOS-based client application that contain an internal version of the driver for each specific print device. This internal driver may, or may not be the correct version of the driver. Often the driver and printer settings can modified within the MS-DOS-based application, or by an accompanying utility.

MS-DOS-based applications may not be *network-aware* meaning that the application does not allow for a network redirector that might be forwarding the print job data over the network to a print server. Some network-unaware applications try to print by directly affecting the parallel port hardware, and may not be able to print correctly to a network print server. Other network-unaware applications assume that they have exclusive control of the printer, and therefore don't worry about properly terminating their print jobs. As a result, nothing prints until the application terminates and the operating system stops the print device.

The following are common problems that occur in this situation:

- MS-DOS driver is misconfigured.
- MS-DOS-based application is not network-aware.

For more information, see "Print Monitors" later in this chapter.

Windows NT Network Clients

Both the Windows NT Workstation computer and the Windows NT Server computer can be a client of a Windows NT print server. Windows NT computers can be configured in several ways to utilize print devices are managed by a Windows NT print server. Client computer configuration is determined by the Printer connection methods. These connection methods and their results are summarized in the following table.

Table 9.3 Establishing Printers on Client Computers

Connection Method	Windows NT Sending Software	Required Network Protocols	Windows NT Receiving Software
Connect To	Remote Procedure Call (RPC)	TCP/IP, NetBEUI, NWLink	RPC Server service
Create Printer and Print To options for a UNC device	Windows NT Workstation Service	TCP/IP, NetBEUI, NWLink	Windows NT Server service
Create Printer and Print To options for a LPR port	LPR (LPR.EXE) Port Print Monitor	TCP/IP	TCP/IP (LPD) Print Service
Create Printer and Print To options used for an AppleTalk device	AppleTalk Print Monitor	AppleTalk	Services For Macintosh

Windows NT Local Clients

You can print from applications running locally on a Windows NT print server.

Windows-based applications and MS-DOS-based applications print in different ways, and follow different rules. Windows-based applications send print jobs to printers defined in Print Manager. They typically use the printer driver associated with the printer they're printing. However, there are exceptions to this rule; some high-end desktop publishing or CAD applications may have their own internal program copy of printer drivers.

In contrast, MS-DOS-based applications are unaware of the printers defined in Print Manager. They print to ports instead of to printers. This can cause problems, because while Print Manager might have a few dozen printers, most MS-DOS-based applications are limited to ports like LPT1-LPT3, and COM1-COM2. Windows NT accommodates MS-DOS-based applications in the following ways:

- If the port is controlled by the network redirector (for example, when a port is assigned to a shared resource with a NET USE command), the redirector determines where the job goes.
- If the port is not controlled by a network redirector, but a printer defined in Print Manager prints to that port, the job is submitted to that printer, and that printer's spooling options take effect.
- If the port is not controlled by a network redirector, and no printer in Print Manager prints to that port, the job goes directly to the port device driver.

EXAMPLE:

Assume you are interactively logged on to a print server, and that there are two printers in Print Manager and two print devices. The server is named \\PSERVER1, the printer named HPV prints to an LPT2 port supplied by a separate IO card, and the printer named HPIIISI prints to the FILE port. Both HPV and HPIIISI are shared on PSERVER1 over the network. There are print devices connected to both LPT1 and LPT2. You open a command prompt and type the following command:

```
NET USE LPT3: \\PSERVER1\HPIIISI.
```

When you copy a file to each of LPT1, LPT2, and LPT3, the following results:

COPY TEST.TXT LPT1:

NTVDM checks with the redirector and finds that the redirector is not managing LPT1. It checks with the spooler and finds that neither of the defined printers prints to LPT1, so the job goes to the parallel port device driver.

COPY TEST.TXT LPT2:

NTVDM checks with the redirector and finds that the redirector is not managing LPT2. It finds that the printer PSERVER1 is printing to LPT2. NTVDM submits the job to HPV and that printer's spooling options take effect as the job is printed on LPT2.

COPY TEST.TXT LPT3:

NTVDM checks with the redirector and finds that the redirector is managing LPT3, so the redirector takes control. The redirector is set so that data sent to LPT3 is actually sent to the print share \\PSERVER1\HPIIISI. It sends the job to that share and once it arrives, HPIIISI's spooling options take effect. Although you usually use the NET USE command to assign a local port to a remote shared resource, it is perfectly legal to assign a local port to a local shared resource. This is often useful in testing and troubleshooting.

UNIX Clients

Windows NT supports UNIX printing clients by providing the TCP/IP Print Server (LPD) service. This serv ice can receive print jobs from UNIX systems or other operating systems, including Windows NT, that have LPR client software. The client software must support RFC 1179. This is problematic, because many systems do not support this specification, and many of the systems that claim to support it do not implement it entirely, or supply private extensions to the specification that their users have come to rely on, but that don't exist except on that system.

The following are common problems that occur in this situation:

- UNIX sends an **f** control command, so the Windows NT print server reformats the job.
- The client computer does not implement all of LPR specification, RFC 1179.
- Users on the client system depend on local extensions to RFC 1179 that exist only in the vendor's UNIX implementation on the client computer.

Macintosh Clients

Macintosh clients send jobs over AppleTalk to a Windows NT print server. To the Macintosh client, the Windows NT computer looks like another AppleTalk device in a zone; either an Apple Share print server or a standalone print device. On the Windows NT Server computer, the Services for Macintosh service receives the jobs. Using the Services For Macintosh service is described in the section of the same title. Additional information is also included in "Services for Macintosh Print Processor (SFMPSPRT)" and "Macintosh Print Monitor (SFMMON)".

Note Because the Services for Macintosh service (SFMSRV component) is only available with the Windows NT Server software, the Windows NT print server for Macintosh clients must be a Windows NT Server computer, not a Windows NT Workstation computer.

Print Spooler Modules

The preceding sections discussed local and network print clients, print jobs, print data types, and print devices. These are entities at the beginning and end of the printing process. The following section discuss the Windows NT print spooler modules and the processing that takes place on a Windows NT print server after the print job is created and before the print job reaches the print device.

The following figure shows the main spooler components used to process jobs on a Windows NT print server. The components are arranged from top to bottom; the components on top use the services of the components below them. For example, print clients use the services of the Router, which in turn uses the services of a print provider, which uses a print processor, which uses a print monitor, before the print job is sent to the print device.

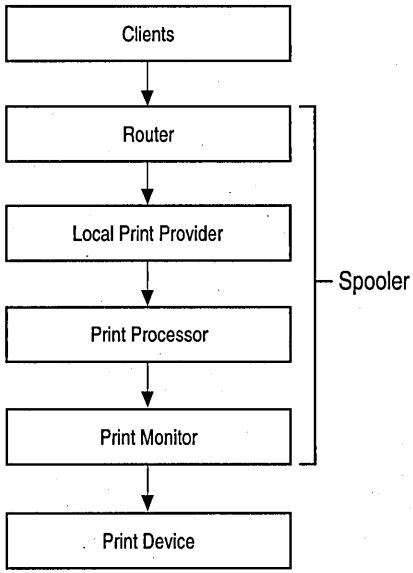


Figure 9.2 Print Spooler Components

As shown in the figure, the components below the clients are collectively called the *spooler*. In Windows NT, the spooler components are implemented as a service that you can stop and restart from the Services icon on the Control Panel, or from the command line, by using the Net Stop Spooler and Net Start Spooler commands.

The spooler in Windows NT is actually a collection of programs and files that enable the Windows NT print server to support 952 unique print devices. The components of the Windows NT spooler are listed in the following table.

Table 9.4 Windows NT Print Spooler Components

Spooler Component	Desciption
Router	Comprised of the two main spooler files, SPOOLSS.EXE and SPOOLSS.DLL.
Local Print Provider	LOCALSPL.DLL for locally run clients.
Remote Print Providers	WIN32SP.DLL for Windows 3.x clients.
	NWPROVAU.DLL for Netware clients.
Print Processors	WINPRINT.DLL, which supports print jobs of data type RAW or TEXT. SFMPSPRT.DLL, which supports print jobs of data type PSCRIPT1.
Monitor-related Utilities	DECMON, which sends print jobs to Digital PrintServer, DEClaser, and DECcolorwriter print devices.
	Universal printer driver
	Printer Drivers
	PostScript printer driver
	HPGL/2 plotter driver
	Note: A version of the printer driver exists for each hardware-platform tree; x86, Alpha, MIPS, and PPC.

Each spooler component is described in more detail in the following sections.

Router (SPOOLSS)

The two main spooler files are SPOOLSS.EXE and SPOOLSS.DLL. Collectively these files are referred to as the Router. This is because all spooler APIs resolve to the server-side of SPOOLSS (SPOOLSS.DLL). In a sense, SPOOLSS is the glue that holds the rest of the spooler together, because not only do external modules' API calls resolve to SPOOLSS, the API calls that the other spooler components use to communicate among themselves also resolve to SPOOLSS. SPOOLSS routes each of these requests for service to the correct spooler component.

SPOOLSS.DLL receives jobs from the local Windows NT Server services. It receives jobs from the client side SPOOLSS (SPOOLSS.EXE) running on remote Windows NT print clients, when those clients use the Connect To option.

Do not confuse the software Router component of the Windows NT spooler with the physical hardware router used in networks.

Remote Print Providers

Remote print providers are used when a Windows NT-based computer uses the Connect To option to establish a connection to a Windows NT-based print server, and send print jobs to that print server. This usually happens when your Windows NT-based computer acts as a workstation on the network, but it can also occur if your Windows NT-based computer acts as a print server that forwards incoming jobs to another print server.

Windows NT supplies the following remote print providers:

- WIN32SPL.DLL transfers jobs to Windows Network print servers (such as print servers running Windows NT or Windows for Workgroups).
- NWPROVAU.DLL transfers jobs to Novell NetWare print servers.

Generally, if you are sending a print job to another print server, you have established the printer in Print Manager using the Connect To option. When a client sends a job to such a printer, the Router polls each of the remote print providers in turn, in effect asking each one whether it recognizes the printer name. The Router passes control to the first network provider that recognizes the printer name.

You can set the polling order using the Network icon in the Control Panel window. To do this perform the following steps:

1. Click the Network icon in the Control Panel window.
2. Click on the Networks button.
3. Click on Print Provider in the Network Providers Search Order dialog box.
4. Select the name of a network and use the Up and Down buttons to change the polling order.

Note Neither of these remote print providers performs spooling. When you send a print job by using these remote print providers, the job does not spool locally.

MS-Network Print Provider

The Connect To option can be used to connect a client computer to a down-level print server or a Windows NT print server. In either case, the print job is sent directly to the remote print server, and is not spooled locally.

When the Connect To option is used with a down-level print server, the client computer is configured to send jobs using the Windows NT NetBIOS redirector. When the Connect To option is used with a Windows NT print server, the client is configured to send jobs by using the remote procedure call (RPC) interface.

If the local copy of the Windows network print provider, WIN32SPL.DLL, recognizes the printer name, it performs additional processing based on the type of print server to which the job is going. If the print server is running Windows NT, WIN32SPL.DLL makes remote procedure calls to the Router component on the remote server. The remote server's Router receives the print job over the network, and begins processing the job as if one of its own local clients had submitted the job.

If the remote print server is not running Windows NT, WIN32SPL.DLL sends a message to the local Windows Network redirector. The redirector forwards the job over the network to the down level server. The downlevel server prints the job.

The functions provided by the Windows network print provider are illustrated in the following figure.

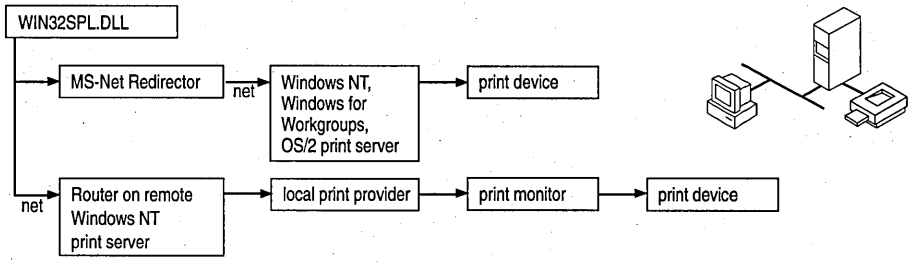


Figure 9.3 The Windows Network Print Provider (WIN32SPL.DLL)

Netware Remote Print Provider NWPROVAU.DLL

To use a NetWare print server, the client computer must use the **Connect To** option. This option configures the client computer to send print jobs using the Window NT Netware redirector. The print job is sent directly to the remote print server; it is not spooled locally.

If the NetWare print provider (NWPROVAU.DLL) recognizes the server name when polled by the Router, it takes control of the print job. The NetWare print provider sends a message to the NetWare workstation service, NWWKS.DLL, which in turn passes control to the NetWare redirector. The NetWare redirector transmits the print job over the network to the NetWare print server.

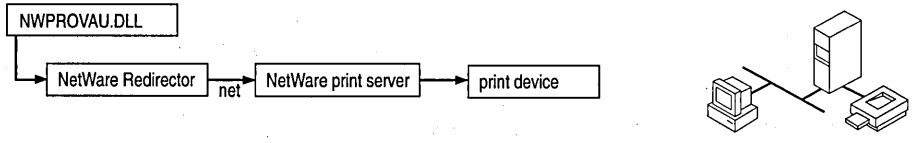


Figure 9.4 The NetWare Print Provider (NWPROVAU.DLL)

Local Print Provider (LOCALSPL.DLL)

The local print provider handles printers established in Print Manager by the Create Printer option. It writes the print job contents to a spool (.SPL) file, and tracks administrative information like user name, document name, and data type in a shadow (.SHD) file. By default, both files are written to: *%systemroot%*\SYSTEM32\SPOOL\PRINTERS.

Next, LOCALSPL polls the installed print processors, such as WINPRINT.DLL and SFMPSPRT.DLL, to see if one of them recognizes the job's data type set. If the data type is not set, the WINPRINT.DLL print processor receives the job and uses the default data-type set in Print Manager.

Most of the local print provider's options are configurable in Print Manager, by clicking on the Details button in the Printer Properties dialog box. These options are documented thoroughly in Help, but you should note some options new to Windows NT in version 3.5:

- You can select whether you want the spooler to hold mismatched jobs. If you do, the spooler makes sure that incoming jobs from locally run Windows-based applications are requesting features that are currently available. For instance, suppose that you have configured the printer driver assuming Letter-sized paper is loaded in the print device. Without the Hold Mismatched Jobs option, the printer will pause indefinitely waiting for a legal paper cassette to be loaded if legal-sized paper is requested. With this option enabled, the local print provider holds this job, enabling correctly-configured jobs to print, until a Legal cassette is loaded.
- You can specify whether to keep spool files after the print device accepts the whole print job. This situation is often undesirable, because spool files accumulate on disk. However, it is possible that between the time that the print device accepts the end of the job and the time it completes printing the job, the print device could be turned off. If you have a very important print job, or one that would be very difficult to recreate, deselecting this option lets you keep the spool file on disk until you're sure the print device has finished printing. Also, this option makes capacity planning easier. The size of a print job is a significant factor affecting the work a print server must do to receive, process, and distribute that job. To estimate the number of megabytes of print jobs that a print server must process, pick a typical printer, disable its Delete Jobs After Printing option, and edit the Registry to force the printer to spool all of its jobs in a directory that no other printer is spooling to. Wait an hour, or a day, and get a directory list to find the minimum and maximum job sizes, the total size of all jobs, and a record of what times the jobs were spooled.
- You can tell the print provider whether it should send data to the print device while the spool file is being written to disk, or wait for the whole spool file to be written before starting to send the job to the print device. Sending data to the print device while the .SPL file is written can improve printing speed on large jobs.
- You can decide which job to send to the print device first if two or more are spooling at the same time. You can select whether the job that *starts* spooling first will print first or whether the job that *finishes* spooling prints first, regardless of how long each job takes to finish spooling.

The local print provider creates two files for each print job sent to the spooler. These files are:

- A spool file, which contains the print job itself, but no administrative information about the job. Spool files have an .SPL extension.
- A shadow file, which contains information such as the name of the destination printer, the job's priority, the name of the user who sent the job, and other administration information. Shadow files have an .SHD extension.

If the print server is shut down while print jobs are spooled and waiting to print, the spool and shadow files remain on the disk and are used to restart the print job when the print server is restarted. The local print provider uses the information in the shadow file to determine how to print the print job, and the content of the job is contained in the spool file.

The spool file and shadow file for a job are kept in the same directory on disk. By default, these files are written to:

%systemroot%\SYSTEM32\SPOOL\PRINTERS

You can set a new default location or override the default location on a printer-by-printer basis by manually editing the Registry. However, before doing so, read the "Spooler File Security" section later in this chapter.

Print Processors

Print processors are the components that make necessary alterations to print jobs, based on the data type of the print job. A print processor might recognize only one data type, or it might recognize several data types. Windows NT supplies the following print processors:

- Windows print processor (WINPRINT.DLL).
- Macintosh print processor (SFMPSPRT.DLL).

Additional print processors may be supplied by third party software vendors to support custom data types.

Windows Print Processor (WINPRINT)

This print processor supports four data types: RAW, RAW [FF Auto], RAW [FF Appended], and TEXT. These are described in the following paragraphs.

RAW

The RAW data type tells the spooler not to alter the job at all. If an appropriate print server service assigns this data type, the spooler passes the print job through without altering it. When you create a printer in Print Manager, the Default Datatype value is initially set to RAW. Most jobs from MS Network clients are assigned the RAW data type.

RAW [FF Appended]

This data type tells the spooler to assume the job is from an application that does not append a form-feed character (0x0C) to the end of each job. Without a trailing form-feed, the last page of the job does not print when sent to a PCL print device. The spooler appends a form-feed character to the end of the print job, but makes no other alterations. None of the appropriate print server services supplied with Windows NT assign this data type, but you can make this data type the system default, so that it affects jobs from MS Network clients.

RAW [FF Auto]

This data type is similar to the RAW [FF Appended] data type, but RAW [FF Auto] tells the spooler to check for a form-feed character at the end of the job. It does not add a form feed if one is already present, and makes no other alterations. None of the appropriate print server services supplied with Windows NT assign this data type, but you can make this data type the system default, so that it affects jobs from MS Network clients.

TEXT

The TEXT data type tells the spooler that the job consists of ANSI text that the user wants to printed on the page. The spooler uses the current printer driver to create a new print job that prints the text of the original job using the print device's factory default font, form, orientation, and resolution listed in the Document Properties dialog box in Print Manager. This is very useful when the client application print job consists of simple text, but the target print device (for example, a PostScript printer) cannot interpret simple text jobs.

Text files actually consist of numeric values from 0 to 255, where each value is mapped to a particular character or symbol. There are several character mapping schemes (character sets) in common use, and text files contain no indication of which character set to use when displaying or printing the file. The TEXT data type assumes the ANSI character set, so it may print some characters incorrectly if the application that created the job does not use the ANSI character set. Most character sets are identical for the values 0 through 127, so this problem usually affects extended characters (those with values from 128 through 255).

The following table shows five examples in which common character sets use different numbers to represent the same character. The PC-850 character set is commonly used by MS-DOS-based applications in Europe; ANSI is used by Windows-based applications; PC-437 is commonly used by MS-DOS-based applications in the United States; Roman-8 is the default PCL character set.

Table 9.5 Character Sets

Character	PC-850	ANSI	PC-437	Roman-8
Lowercase C Cedilla	135	231	135	181
Lowercase AE Diphthong	145	230	145	215
Lowercase N Tilde	164	241	164	183
Lowercase Eth	208	240	-	228
Lowercase Es-zet	225	223	225	222

Services for Macintosh Print Processor (SFMPSPRT)

This print processor is only available with Windows NT Advanced Server version 3.1 or Windows NT Server version 3.1, 3.5, and 3.51. By default, this print processor (SFMPSPRT) is only installed when Services For Macintosh is installed on the Windows NT Server computer.

SFMPSPRT supports the PSCRIPT1 data type. This data type indicates that the job is Level 1 PostScript code from a Macintosh client, but the target printer is not a PostScript printer. The spooler sends the PostScript code through a TrueImage raster image processor (RIP), supplied with Services for Macintosh. The raster image processor creates a series of one-page, monochrome bitmaps at a maximum of 300 DPI. The Windows NT print spooler sends the RIP bitmap to the print driver for the target printer. The print driver returns a job that prints those bitmaps on the page.

Because the RIP bitmaps are monochrome and not more than 300 DPI, the target printer driver produces final output that is monochrome and not more than 300 DPI, even if the target printer driver supports color or supports higher resolutions. Similarly, the Windows NT PostScript driver is a Level 2 driver, but the RIP does not rely on the PostScript driver to generate its bitmaps. In short, the RIP restrictions, and therefore the PSCRIPT1 restrictions, are in the RIP software itself, not in the Windows NT printer drivers.

These restrictions do not affect most business users. For people who need a more full-featured RIP, several third-party Win32 RIP packages are commercially available for Windows NT version 3.1 and above.

Print Monitors

Print monitors are the code modules that send print jobs over a specific communications channel, regardless of the make and model of the print device on the far side of that channel.

Statistically, print monitors are one of the most troublesome parts of the spooler. Problems with print monitors include:

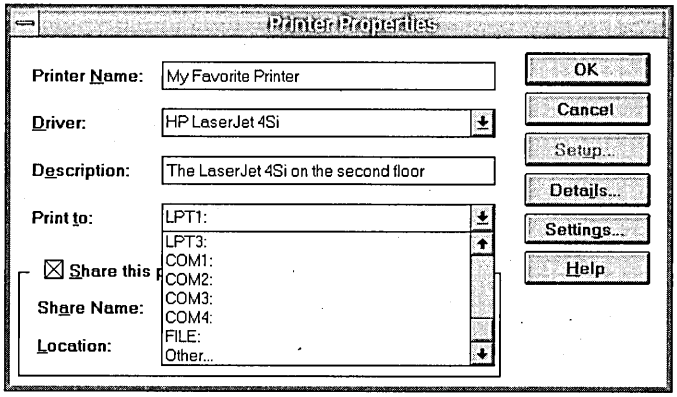
- Jobs that are successfully sent to a printer, but do not print.
- Jobs that print, but are truncated.

The print monitors supplied with Windows NT are described in the following sections. These print monitors are:

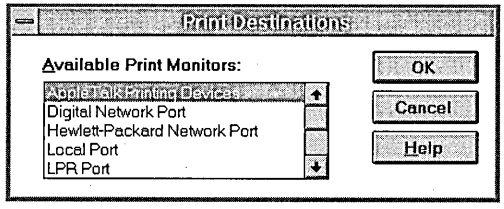
- Local print monitor (LOCALMON)
- Hewlett Packard print monitor (HPMON)
- Line Printer (LPR) Port print monitor (LPRMON)
- Macintosh print monitor (SFMMON)
- Digital Network Port print monitor (DECMON)

Specifying a Print Monitor

In the Printer Properties dialog box, the Print To listbox lists the default Windows NT ports. See the following example.



By default, the list in the Print To listbox includes only standard ports controlled by the local print monitor, LOCALMON.DLL. When you want to print over other communications channels (to a network-attached printer, for example), you must create a new port. To create a port, select Other from the Print To dialog box. The Print Destinations dialog box lists the available print monitors.



Monitors often depend on other software components and do not appear in this list unless you have loaded the components they require. For example, the Hewlett-Packard Network print monitor transmits print jobs by using the DLC network protocol. You see this monitor in the list only if you have installed the DLC protocol.

Select the monitor that controls the type of communications channel you want to use, and choose OK. The monitor displays its own user interface, which you use to create a new port. After you have created the new port and configured a printer to use that port, the Settings option in the Printer Properties dialog box launches the monitor user interface again, if the monitor allows reconfiguration of a port.

When you read details about each print monitor in the following sections, remember that each print monitor is concerned with a data communications channel, not with the print device at the other end of that channel. In most cases, the print monitor is not aware of the make or model of print device it is communicating with, nor does it need this information. Also, different print monitors may use the same network protocol, but this does not make them interchangeable. For example, both the Digital Network Port print monitor and the LPR Port print monitor use the TCP/IP protocol, but they send data over that protocol in very different ways.

Local Print Monitor (LOCALMON)

The local print monitor, LOCALMON.DLL, sends print jobs to local devices. These include familiar ports like LPT1 and COM1. Use of the less familiar FILE and Other ports are described below.

The FILE port appears in the default port list in the Printer Properties dialog box. When you send jobs to a printer that uses this port, the local print monitor prompts you for the name of a file in which the print job should be stored.

If you select Other from the list of ports in the Print To box on the Printer Properties dialog box, and select the Local Port option, the local print monitor prompts you to enter a port name. Some possibilities include:

- An explicit filename, such as C:\DIR\FILENAME. All jobs sent to this port are written to the named file. Each new job overwrites the last one.
- The UNC name of a print share, such as \\SERVER\PRINTER. Jobs sent to this port are transferred over the network to the named share by the network redirector. This can be useful if you need to send jobs to a network print server, but you want the job to spool locally as well as on the print server.

- The NULL port. You can use this port to test whether network clients are able to send jobs. Simply pause the printer set to use this port, send a job from a network client, look at the printer in Print Manager to confirm that it arrived, and resume the printer. Jobs sent to NULL are simply deleted from the system, without wasting paper or delaying real print jobs.

Hewlett-Packard Network Print Monitor (HPMON)

The Hewlett-Packard print monitor, HPMON.DLL, sends print jobs to HP JetDirect adapters. This includes both the network adapters commonly installed in print devices such as the LaserJet 4 Si and the JetDirect device, which connect a parallel print device to the network.

Many JetDirect devices can communicate over several different network protocols, including DLC, IPX, TCP/IP, and AppleTalk. HPMON.DLL is specific to DLC; you must load the DLC protocol to use this print monitor, and it is not able to transmit jobs over other protocols.

This monitor has several operating parameters as described below:

- The DLC protocol is bridgeable, but not routable. This means that if a Windows NT print server is on one physical subnet, and a JetDirect device is on another physical subnet, the server can send jobs to the JetDirect if the two subnets are joined by a bridge, but cannot send jobs if the two subnets are joined by a router.
- The DLC protocol can be bound to multiple network adapters, but the HP print monitor software can only manage printers over one network adapter, and it must be either adapter 0 or adapter 1. If your Windows NT computer has multiple network adapters, make sure all the HP JetDirect-equipped printers are on the same physical subnet. Also, note that adapter numbering can change during an upgrade. If you install one adapter initially, it is adapter 0. If you later install a second adapter, it is adapter 1. If you upgrade or reinstall, and adapter 1 has a lower IO base address than adapter 0, adapter 1 is changed to adapter 0 during the reinstall process, and adapter 0 is changed to adapter 1. The Registry entries for HPMON refer to the adapter only by number. In this case, HPMON looks for JetDirect cards through the wrong network adapter on the server.

- Ports managed by this print monitor are configurable for either Job Based or Continuous connection. The setting affects all ports at once. The Job Based connection option configures the print server to connect to the JetDirect adapter, send a print job, and disconnect when the print job is completed. The Job Based connection option enables other print servers to connect to the JetDirect adapter. The Continuous connection option configures the print server to connect to the JetDirect adapter and maintain the connection. The Continuous connection option prevents other servers from connecting and sending print jobs. The Continuous connection is held until either the Windows NT print server or the JetDirect print device is rebooted. The main advantage of Continuous connection is that all users are validated by the Windows NT security model, and every print job access can be audited.

Note If you configure two Windows NT print servers to send jobs to the same JetDirect device, configure both servers for Job Based connections. If you configure one of the print servers for Continuous connection, it prevents the Job Based server from connecting to the print device.

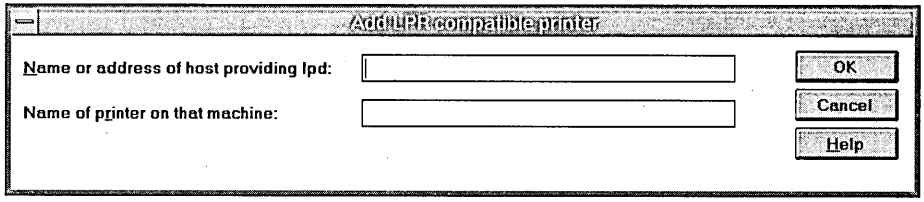
Line Printer Monitor (LPRMON/LPR.EXE)

LPR is one of the network protocols in the TCP/IP protocol suite. It was originally developed as a standard for transmitting print jobs between computers running Berkeley UNIX. The LPR standard is published as Request For Comment (RFC) 1179. Windows NT complies with this standard, as do most Berkeley UNIX operating systems. However, most System V UNIX operating systems do not comply with this standard, so in most cases Windows NT is not be able to send print jobs to System V computers, or receive print jobs from them. Exceptions are System V computers that are configured to accept BSD jobs; these computers can accept Windows NT print jobs.

The LPR protocol lets a client application on one computer send a print job to a print spooler service on another computer. The client application is usually named LPR and the service (or daemon) is usually named LPD. Windows NT 3.5 supplies a command line application, the LPR.EXE utility, and the LPR Port print monitor. Both act as clients sending print jobs to an LPD service running on another computer. As mentioned previously, Windows NT also supplies an LPD service, the TCP/IP Print service, so it can receive print jobs sent by LPR clients, including UNIX computers and other Windows NT computers.

The LPR protocol was not designed to pass detailed error status information back to the LPR client. If anything goes wrong, from severe problems (such as the server being too busy to process requests) to print device problems (such as running out of paper), the LPR protocol reports the same error condition. As a result of this protocol limitation, Print Manager cannot provide detailed information when an error occurs while printing to an LPR port.

To send print jobs, the LPR client needs the network address of the LPD server computer, and it needs the name that the LPD service associates with its print device. Given this information, LPR sends print jobs to the LPD service, along with instructions on how to process the print job, and the name of the print device that should receive the job. The user interface as shown below enables you to use the Windows NT LPR Port print monitor to specify the Windows NT print server and printer (queue in UNIX terminology) which should receive the print job.



Use the Name Or Address of Host Providing LPD box to tell the LPR Port print monitor which UNIX computer it should send print jobs to. You can supply either the IP address or the host name of the UNIX computer.

For example, to send jobs to a printer named LABLASER on a UNIX computer whose IP address is 11.22.33.44, and whose name, defined in the hosts file on your Windows NT computer, is UNIXBOX. In the dialog box above, you can enter either "UNIXBOX" or "11.22.33.44" (without the quotation marks) in the Name Or Address Of Host Providing LPD box. You would enter LABLASER in the Name Of Printer On That Machine box.

If you don't know a valid name for the printer, you can often find it by looking at the /etc/printcap file on the UNIX computer. The printcap file is a flat-file text database of print queue information. Each entry corresponds to a print queue on the UNIX computer. Fields in these entries are separated by : characters, and for readability an entry may be broken over several lines by ending a line with a \ character and beginning the next line with a space or tab character. The first field of each entry lists valid names for the queue, separated by | characters. The remaining lines in each printcap file entry describe the queue's characteristics, such as communications parameters, spool file location, and error log file location.

Continuing the LABLASER example, we might find entries like the following in the printcap file on the computer named UNIXBOX:

```
lp|lablaser|The_Lab_Printer:\
:lp=/dev/ttya:br#9600:\
:lf=/usr/spool/lpd/lablaser-err:\
:sd=/usr/spool/lpd/lablaser:
```

The first line in this example defines a print queue with three valid names: lp, lablaser, and The_Lab_Printer. You can use any of these names in the second field of the LPR Port dialog box previously shown.

Note This example is provided for illustrative purposes only. The UNIX system documentation is your best source of detailed information on your system's printcap file.

When the LPR Port print monitor receives the LPD server's network address and the proper queue name, it can send print jobs (data files) and processing instructions (control commands contained in a control file). RFC 1179 defines 29 control commands. The three control commands described below are particularly important.

- The **f** command causes the data file to be printed as a plain text file, providing page breaks as necessary. Any ASCII control characters which are not in the following list are discarded: HT, CR, FF, LF, and BS. LPD should filter out most of the nonprinting control characters.
- The **l** command causes the specified data file to print without filtering the control characters (as is done with the **f** command).
- The **o** command prints the data file to be printed, treating the data as standard Postscript input.

Note Notice that all the control commands defined in RFC 1179 are case sensitive. Notice also, that many printer languages, including PCL, rely heavily on the ESC control character, which the **f** control command causes to be filtered from the print job. Do not use the **f** control command when sending print jobs that contain printer commands.

The LPR Port print monitor sends the **l** command by default, while the command line LPR.EXE utility sends the **f** command by default. With the LPR.EXE utility, you can use the **-o** command if you want to override the default on a job-by-job basis. If you want to change the default command for a particular printer controlled by the LPR Port print monitor, you need to modify a Registry parameter. Use the Registry Editor (REGEDT32.EXE) to find the key named:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\
LPRPort\Ports
```

Next, select the port whose default control command you wish to change, and select its Timeouts key. In this key, add a value named PrintSwitch with type REG_SZ, and enter the control command you want to use. For instance, enter the letter "f" (without the quotation marks) if you want to use the **f** command by default.

Some UNIX computers do not follow the control commands alone when deciding how to process a print job. For instance, if you send an ASCII text file directly to a PostScript printer, it does not print correctly. As a result, many UNIX systems have additional software that converts ASCII text jobs into PostScript jobs that print correctly. System administrators are wary of jobs that arrive with an **l** command, because they could be non-PostScript jobs accidentally sent with an **l** command, which would let them bypass the PostScript software and print incorrectly. To avoid this possibility, some LPD services scan jobs that arrive with the **l** control command, to find PostScript commands. If the scanner finds these commands, it passes the job directly to the printer as requested; otherwise, it assumes the user sent the wrong control command, and it sends the job through the PostScript software.

If you send PostScript jobs from a Windows NT computer using LPR, and the printer controlled by the UNIX server prints the PostScript code instead of interpreting it, the UNIX server may have a scanner that does not recognize the output from the Windows NT PostScript driver as valid PostScript code. If this happens, you may need to reconfigure Windows NT to use the **o** control command by default.

To configure a Windows NT print server to print using the LPR service on a LPD-compliant print device, you must provide a name in the Name of Printer On That Machine dialog box. The name value must be a valid LPD print queue (printer in Windows NT terminology) name.

LPD-compliant print devices must expose one or more names for each of their print queues, even though these devices often have only one possible destination for the print jobs they receive. Each print device manufacturer chooses naming conventions independently. Some, like the HP JetDirect adapter, accept any string as a legal print queue name. In comparison, the Emulex NetJet adapter accepts only two strings, and these are case sensitive, and default to TEXT and PASSTHRU. These strings are configurable. To find the names supported by any specific adapter, check its documentation, or contact the vendors technical support group.

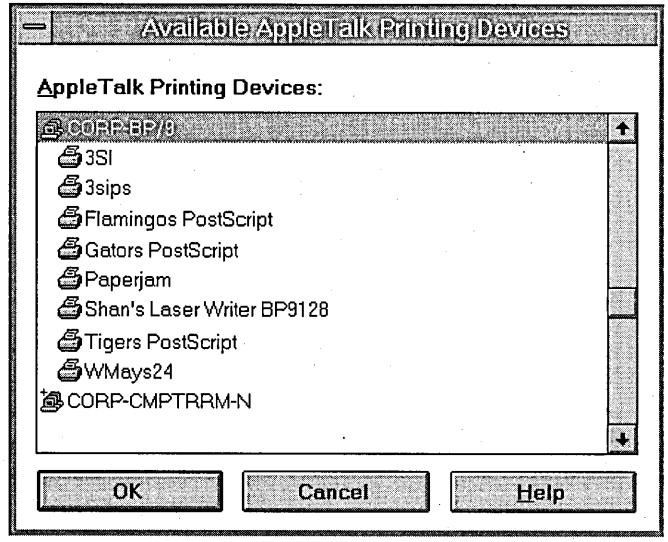
Remember that the local print provider spools print jobs, and that it despools them to the print monitors. In Windows NT 3.5 and 3.51, after LPRMON receives a job from the local print provider, it spools it a second time as a temporary file in the SYSTEM32 subdirectory. This is because LPR must send an accurate byte count in the control file, and it cannot get that byte count from the local print provider. Instead, it respools the data to a temp file, finds the size of that temp file, and sends that size in the control file to the LPD server.

The most common problem people encounter when printing from UNIX systems to a Windows NT print server is that their print jobs are processed as TEXT data type instead of RAW data type, as they would be on a UNIX system. This happens because the UNIX systems almost always send the **f** control command, expecting that the control command isn't too important, because the TCP/IP (LPD) server parses the job to identify what data type to use and how to alter the job. However, Windows NT relies on the control command to determine the data type. As a result, the LPD service on Windows NT assigns the TEXT data type to most jobs. See the section titled "Print Processors" earlier in this chapter for more details on symptoms of this problem.

Macintosh Print Monitor (SFMMON)

The Macintosh print monitor, SFMMON.DLL, transmits jobs over a network, using the AppleTalk protocol, to network-attached print devices such as the Apple LaserWriter family. It also lets you send jobs to AppleTalk spoolers, regardless of the print device that the spooler is attached to.

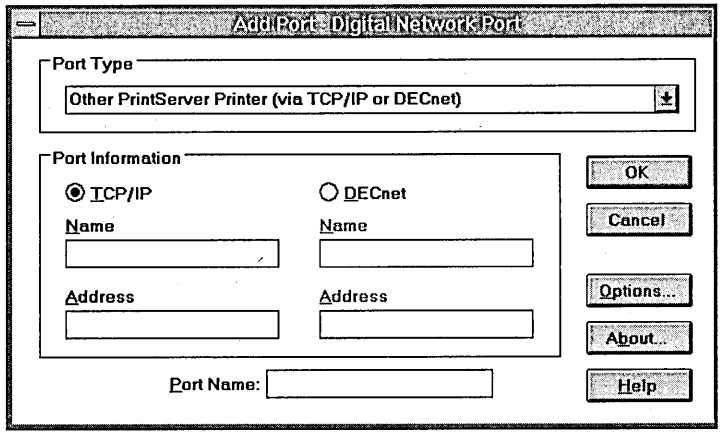
The following configuration dialog box example, displays the available network zones, shows the available printers in that zone, and lets a user choose a zone.



This monitor is available on both Windows NT Workstation and Windows NT Server computers, and enables any Windows NT-based computer to send local print jobs to AppleTalk printers. However, only Windows NT Server has a Macintosh print server component, so only a Windows NT Server computer can receive print jobs from Macintosh clients.

Digital Network Port Print Monitor (DECMON)

The Digital Network Port print monitor, DECMON.DLL, sends print jobs to Digital Equipment Corporation's Digital PrintServer print devices, and other Digital Equipment Corporation print devices such as the DEClaser 5100 and the DECcolorwriter 1000. Use this monitor user interface to select the print devices on which you want to print, and the network protocol you want to use.



Windows NT supplies the TCP/IP network protocol, but does not supply the DECnet™ protocol. If you want to use DECnet, you must contact Digital Equipment Corporation to obtain it.

Using Print Manager

The preceding sections described the components of the Windows NT print spooler. This section begins discussing how to implement advanced features of Windows NT printing.

Users familiar with Print Manager in versions of 16-bit Windows often misunderstand Print Manager in Windows NT which is an interface that lets you observe and configure the spooler. It is not the spooler itself and properly speaking, it is not part of the spooler. Use Print Manager to configure initially the spooler. Once the spooler is configured, Print Manager does not effect the spooler, unless the spooler needs to be re-configured.

Note Problems in Print Manager are possible but infrequent. Replacing the PRINTMAN file rarely solves Windows NT printing problems.

Configuring Printer Drivers Using Printer Properties

There are two dialog boxes in Print Manager that you can use to configure Windows NT printer drivers. You access both dialog boxes by selecting the printer you want to configure and selecting Properties on the Printer menu.

The resulting Printer Properties dialog box includes two buttons that take you to these two dialog boxes.

Table 9.6 Printer Details Dialog Box

Button	Description
Setup Button	Displays a **Printer Setup** dialog box that you can use to tell the spooler how the print device's hardware is configured. For instance, you can specify which forms are loaded in the device's trays, how much memory is installed in the device, or how a plotter's pens are arranged.
Details Button	Displays a **Printer Details** dialog box from which you can specify a separator file, priority, print processor, default data type, special spooling options, and additional ports.

The Printer Details dialog box also enables you to restrict the time the printer is available. On the Printer Details dialog box, the Defaults button displays the Document Properties dialog box, which you can use to define default settings such as:

1. Which form Windows-based applications should use by default.
2. Whether to print portrait or landscape.
3. What resolution to print.

Note Many Windows-based applications have a Print Setup option that usually displays the Job Defaults dialog box.

As a general rule, use the Setup button to set options that affect every job sent to the print device. Use the Job Defaults dialog box option to provide default values that network users are free to change from one job to the next.

Managing Print Forms

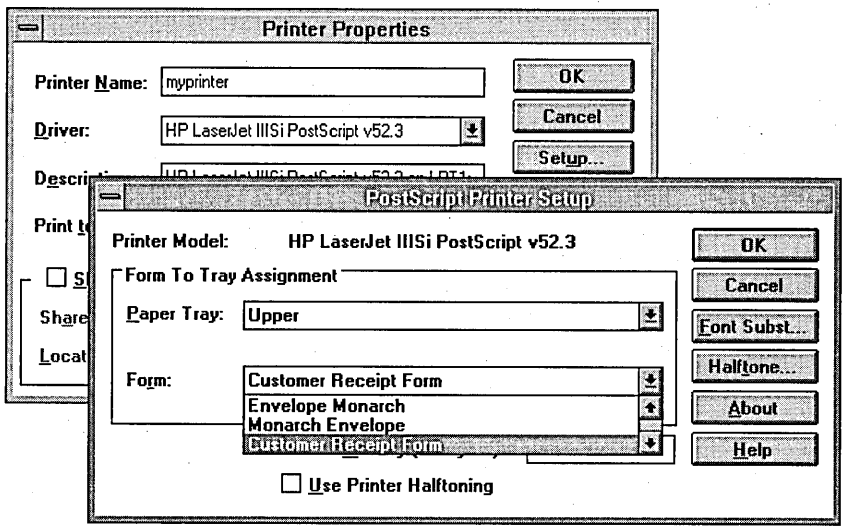
A major difference in printing between Windows 3.1 and Windows NT is the move from tray-based printing to forms-based printing. Under this model, the print server administrator configures the Windows NT print server, defining the form that is currently loaded in each paper source (tray). The form is defined in Windows NT using the following criteria:

- Size
- Image area
- Form name

Using Windows-based applications running on a Windows NT-based computer, each user can choose a desired print form. This frees the user from having to know which tray contains which form. The form selected by the user is identified in the print job source code when an application is used to create a print job. The Windows NT print server process the print job, checks the print device form-to-tray assignments, and sends commands to the print device to use the tray containing the user-specified form.

Any user with Full Control privilege can define a new form by using the Forms dialog box in Print Manager. For example, you could create a form called Customer Receipt Form that uses Letter-size paper and nonstandard margins. You can also can create multiple forms with the same paper size, or margins, or both, to meet specific user needs. For example, you can create forms that have unique names but the same paper size and image area (margins) to identify different departmental letterhead paper stock.

New form definitions are added to the print server's database and are stored per server, not per printer. To assign forms to a specific print device and tray, use the Printer Properties dialog box in Print Manager. Select the Select button to display the Printer Setup dialog box. Select the Form drop-down list to display a list of the available forms. This list contains only those forms that can be used by the printer; form sizes the printer cannot accommodate are not displayed. See the following illustration.



Note To set the default form, do not use the Setup button. Choose the Details button on the Printer Properties dialog box, select Job Defaults. Define the default form in the Form field on the Job Defaults dialog box.

Users who want to print a document can select the new form from the list shown in the application's Print Setup dialog box. The Windows NT print server spooler modules contain tray and form assignment data and send instructions to the print device to select the correct tray.

Windows-based applications can use different forms within a document. For example, you might use an envelope for the first page, Letterhead for the second page, and Letter for the third and following pages.

Note If an odd-sized form is needed, specify Manual Feed in the Paper Tray.

Managing Separator Page Files

The local print provider manages printers established in Print Manager by the Create Printer option. This option is typically used on the Windows NT print server computer.

The local print provider contains an interpreter, that reads commands from a separator file and produces one or more pages of text or graphics. These pages are added to the front of the print job. These pages typically show who submitted the job, when the job printed, and what server it printed on. Separator pages are sometimes called *header pages* or *burst pages.*

By default, separator page files are stored in the *%systemroot%*\SYSTEM32 directory. To use a separator page file, type its name in the Separator File text box of the Printer Details dialog box in Print Manager. Leave this text box blank, if no separator file is to be used.

The following table lists the separator files included with Windows NT. This table supersedes similar tables in the *Windows NT Server System Guide* and the *Windows NT Server Concepts and Planning Guide.*

Table 9.7 Separator Files Included with Windows NT

Filename	Purpose	Compatible with
SYSPRINT.SEP	Prints a page before each document	PostScript
PCL.SEP	Switches dual-language HP printer to PCL printing	PCL
PSCRIPT.SEP	Switches dual-language HP printer to PostScript printing	PostScript

To create your own separator file, you can copy and rename one of the supplied separator files. The following table shows the escape codes you can include in a separator file. The first character of the separator page file must always be the escape character. This character is used throughout the separator page file in escape codes. The separator file interpreter replaces these escape codes with appropriate data that is sent directly to the printer.

Table 9.8 Separator Page Escape Codes

Escape code	Function
\	The first line of the separator file is a single character. The separator file interpreter considers this the separator file command delimiter. This table assumes that character is a \ character.
\N	Prints the user name of the person that submitted the job.
\I	Prints the job number.
\D	Prints the date the job was printed. The representation of the date is the same as the Date Format in the International section on the Control Panel.
\T	Prints the time the job was printed. The representation of the time is the same as the Time Format in the International section on the Control Panel.
\L*xxxx*	Prints all the characters (*xxxx*) following it until another escape code is encountered.
\Fpathname	Prints the contents of the file specified by path name, starting on an empty line. The contents of this file are copied directly to the printer without any processing.
\H*nn*	Sets a printer-specific control sequence, where *nn* is a hexadecimal ASCII code sent directly to the printer. To determine the specific numbers, see your printer manual.
\W*nn*	Sets the width of the separator page. The default width is 80; the maximum width is 256. Any printable characters beyond this width are truncated.
\B\S	Prints text in single-width block characters until \U is encountered.
\E	Ejects a page from the printer. Use this code to start a new separator page or to end the separator page file. If you get an extra blank separator page when you print, remove this code from your separator page file.
n	Skips *n* number of lines (from 0 through 9). Skipping 0 lines simply moves printing to the next line.
\B\M	Prints text in double-width block characters until \U is encountered.
\U	Turns off block character printing.

Implementing Print Security

The Security menu in Print Manager is used to assign permissions to users and groups. Additional security can be assigned on the print spool file directory and sections of the Registry that affect printing. Additional security can be provided by managing print jobs from Macintosh clients or by forwarding print jobs to other print servers.

Printer Security

You can specify which printers have which security attributes, by using the Security Menu in Print Manager. Security attributes can be specified on each printer established on the Windows NT print server with the Create option. These security attributes are established by assigning permission levels by user groups. For example, all nonadministrative users in a user department can be given the Print level of permission, managers can be given Full Control permission with complete access and administrative control.

The following table lists the security permissions that can be assigned.

Table 9.9 Printer Security - User Permissions

Type of permission	Level of access
Full Control	Enables complete access and administrative control.
Manage Documents	Enables a person to change the status of any print job submitted by any user. Does not permit control of the printer status.
Print	Allows user to send print jobs to the printer and to control pause, resume, or delete for his or her own jobs.
No Access	Explicit denial of access to a specific printer.

The installation-default printer permissions are different for Windows NT Server and a Windows NT Workstation computers. By default, the following print permissions are assigned on a Windows NT Server computer:

- Full Control permission for the Administrator, Server Operator, and Print Operator groups
- Manage Documents permission for the Creator Owner group
- Print permission for all users

By default, the following print permissions are assigned on a Windows NT Workstation computer:

- Full Control permission for Administrator and Power User groups
- Manage Documents permission for Creator Owner group
- Print permission for all users

Security for Macintosh Clients

Although native Macintosh networking provides support for file security, it does not provide support for print device security. In the AppleTalk protocol there is no mechanism that supports client-user name or password. Macintosh clients therefore cannot identify themselves on the network and the Windows NT print server cannot impose user-level security on Macintosh clients. If a Macintosh client is physically able to send a job to a print device or print server, that client implicitly has permission to do so.

You can, however, enforce one set of printer permissions on all Macintosh users as a group. The Macintosh client must start the MacPrint service by always logging on using a user account, by default, it logs on as the System account. The System account has Print permission on all local print devices, so, by default, any Macintosh client can send a job to any of the Windows NT computer's local printers. If you want Macintosh clients to have a different set of permissions, you must create a new user account, give this user account the printer permissions you want Macintosh users to have, and set the Macintosh client MacPrint service to log on using this account. To do this, perform the following steps:

1. Open Control Panel and select the Services icon.
2. Select Print Services for Macintosh from the list, and then select the Startup button.
3. In the Startup dialog box, choose the This Account button and type the name of the user account you created in the box.

Note The System account on one computer does not have permission to access resources on other computers. Macintosh clients that start the MacPrint service by logging on as System user cannot send jobs to printers that forward jobs to other print servers. The solution is to configure the Macintosh client MacPrint service to log on as another user, one who has permission to print on all the print servers to which print jobs are forwarded.

Spool File Security

Spool file security can be created by changing the default spool directory to a NTFS partition and directory where write permission is limited by user.

When a print job prints locally, the local print provider spools the job to disk during processing. By default, the Everyone group has Change permission in the default spool directory. This allows all user print jobs write access to the default spooler directory.

A print job that cannot be spooled to disk during processing does not print. If the spool directory location is changed, all users who should print must have, or be assigned, Change permission for the new spool directory.

As described in the previous section "Local Print Provider", the spool file (SPL_) and the shadow file (SHD) are written to the default spooler directory:

```
%WINNT%\SYSTEM32\SPOOL\Printer
```

To set a new default location by manually editing the Registry

1. Start the Registry Editor (REGEDT32.EXE).
2. Find the following key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
   Print\Printers
   ```

3. Add a DefaultSpoolDirectory setting and as its value provide the full path to the spool directory that all printers should use by default.

The change in the Registry takes effect after you stop and restart the spooler service.

▷ **To override the default location for one specific printer**

1. Start the Registry Editor (REGEDT32.EXE).
2. Find the following key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
   Print\Printers
   ```

3. Find the key for the printer:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
   Print\<Printername>
   ```

4. Add a new SpoolDirectory setting, and as its value provide the path to the spool directory that this printer should use.

The change in the Registry takes effect after you stop and restart the Spooler service.

Registry Security

Most printing-related Registry settings reside in the subkey of:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print
```

An administrator can use the Registry Editor to assign read-only access to these subkeys. Users who are assigned read-only access are not able to install or configure printers using Print Manager, because the read-only access does not allow changes on these subkeys.

Windows-based applications also use the following subkey in the Registry to find information about available printers. This subkey is:

```
HKEY_CURRENT_USER\Software\Microsoft\WindowsNT\
CurrentVersion\PrinterPorts
```

Users who do not have permission to write to this subkey cannot add new printers that are recognized by Windows-based applications.

Forwarding Jobs

By default, job forwarding from one Windows NT print server to another, is disabled. This is because the Windows NT print server that forwards the print job to another Windows NT print server, uses a *null session* to forward the job. Under Windows NT 3.5, the null-session is disabled by default and prevents job forwarding.

You can enable job forwarding and null-session support by manually editing the following Registry subkey:

```
\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
LanmanServer\Parameters
```

In this subkey is a value named NullSessionShares. Edit this value and add a new line containing the sharename for the printer. This change does not take effect until you stop and restart the Spooler service.

Implementing Print Auditing

Windows NT Print Manager provides an auditing option to track successful or unsuccessful printing and administrative events. This tracking can be done by specified groups or individuals. However, you must first enable the audit feature.

▸ **To start the audit feature using the User Manager**

1. Select Audit on the Policies menu .
2. Select Audit These Events, and then choose OK.

You do not need to select any specific events in this dialog box to enable print auditing.

Troubleshooting Print Problems

Troubleshooting Windows NT printing problems can be difficult because of the number of variables involved in printing and the number of different clients and print devices that Windows NT supports.

Windows NT has a modular printing architecture. There is a module for each major task, such as receiving jobs from network clients, and it is easy to add new modules to add functionality such as supporting a new type of network client. This modularity gives Windows NT a great deal of flexibility; it is able to support a wide variety of client operating systems, applications, data objects, network configurations, spooling options, and print devices. That flexibility comes at a cost, though, because the wide array of possible configurations creates a huge number of possible points of failure.

Successful troubleshooting depends on your ability to quickly rule in or rule out general categories of points of failure. The modularity makes this fairly easy: there are seven processes involved in network printing, that always occur in the same order. By testing one of the processes, you can determine whether the problem is occurring in that process, before it, or after it.

The seven basic processes involved in network printing are:

1. An administrator creates a print share on the print server.
2. A client system connects to that share.
3. The client system creates a print job.
4. The client system sends the print job to the print share on the print server.
5. The print server receives, spools, and sometimes modifies the print job.
6. The print server sends the job to the print device.
7. The print device interprets the job and produces hardcopy output.

The basic strategy for troubleshooting printing problems is to use problem symptoms to identify the process, or processes, that are creating the problems.

To troubleshoot printing problems,

1. Identify which of the seven processes is failing.
   - Analyze symptoms to identify the most likely process.
   - Reconfigure that process.
   - Retest with the new configuration.
   - If the problem changes with the new configuration, you probably have the right process. Otherwise, you probably picked the wrong process.
2. Look for documented explanations or solutions from the following sources:
   - Product hardcopy documentation
   - Product online Help documentation.
   - Microsoft Product Support Knowledge Base
3. If possible, reconfigure the process to avoid the problem in the short term.
4. Implement long term solutions.

To help you implement this strategy, this section presents the seven basic printing processes and gives the following information about each process:

- A basic description of what the process involves
- A list of variables that can impact the process
- Symptoms that would suggest a problem in this process
- Tests to prove or disprove the suspicion

Printer Definition and Configuration

A print server administrator can use Print Manager to do the following:

- Create a local printer
- Connect to a remote printer

Both procedures install and configure a printer driver. Create Printer gives more control over driver configuration than Connect To, and the administrator can create new printer ports by entering values in the Print To field on the Printer Properties dialog box. The administrator can also define form-to-tray mapping, security, and spooler options.

Suspect a problem here if...	Confirm the problem is here if...
You cannot start Print Manager.	You cannot browse printers in Print Manager.
You cannot create a new local printer or a new local printer port.	The port name and port does not show up or is intermittent
The problem is specific to one printer or to a subset of the printers.	You can create another printer (with the same driver, port, and configuration options, but a different name) and the problem does not occur with that new printer.
You cannot install a particular printer driver.	The driver for the printer does not exist.
You cannot share a printer.	You cannot get security on printers.
You have problems with forms, separators, pages, fonts, halftones, or printer options.	You can delete the printer, stop and restart the spooler, recreate the printer with the same name and configuration, and the problem no longer occurs.

Client Computer Connects to a Shared Printer

A printer is established on a print client computer.

Suspect a problem here if...	Confirm the problem is here if...
You have problems with forms, separators, pages, fonts, halftones or printer options	You can connect to another shared printer, on the same print server, and the problem goes away.
You cannot connect to a remote printer.	You cannot browse printers in Print Manager

Client Application Creates A Print Job

A user on a client system runs an application that composes text and/or graphics to create output, a print job. The application may interact with a printer driver to create output in a printer language such as PCL, PostScript, or HP-GL/2.

Suspect a problem here if...	Confirm the problem is here if...
The error in one particular print job is not reproducible.	You create a simple test document, and print it to a file. Transfer that file to a different client system, and print it to a different printer on a different print server, and there, and it continues to fail.
One particular user, or the users in a specific group.	A different driver works fine.
One particular client computer.	What you see on the screen is not the output that you get.
Client computers using a particular operating system.	
Client computers using a particular vendor's printer driver, or one version of that driver.	
A particular font, or fonts from a particular vendor.	
Certain characters especially the extended-ASCII characters common in languages other than American English.	
A particular graphic object, or a type of graphic object, for example EPS, TIFF, BMP, or graphics generated by a particular application generates an error problem.	
All jobs created by a particular application, or version of an application.	
FF or LF is not sent or improper ASCII codes are sent.	
Color and shading problems.	
Incorrect dots per inch (DPI).	

The problem is most likely caused by another printing process if the problem persists when you send a simple test text job from each of several clients systems.

Client Sends Job to Spooler

The user on the client system sends the job over the network to the print server. The client system's application software or operating system sends the job to the client's transport protocol, to the network adapter, over the network hardware, to the transport software on the print server, and finally to the appropriate print server service on the print server. The print server service (Windows NT Server, Services for Macintosh, or TCP/IP Print) assigns a data type to the print job and submits the print job to the spooler, or leaves it blank.

Suspect a problem here if...	Confirm the problem is here by...
One particular client redirector (one vendor's software, or one version of that software).	Sending job again.
One particular network transport for example, TCP/IP, NWLink, NetBEUI, AppleTalk.	Checking disk space.
One particular make or model of network adapter, or one firmware level.	
One particular intermediate system for example, one specific router, bridge, hub, or gateway.	
One particular kind of intermediate system, for example, all jobs from clients on the other side of any router, bridge, or gateway.	
One particular kind of client, for example MS Network, LPR, or Macintosh clients.	
The print job is not started until the application is exited.	
Pages come out incomplete.	

Suspect a problem elsewhere if...

- The print job prints fine on another printer of the same make, model, or revision.
- You send a print job to the spooler in a different way and the problem does not go away.

Print Server Spooler Processes Print Job

The spooler receives the job from the print server service. If the job is targeted to a printer that was established with the Connect To option, the remote print provider sends the job to the print server. Otherwise, the job goes to the local print provider. Unless otherwise configured, the local print provider spools the job to disk and checks the data type assigned by the print server service. The data type (for example, RAW or PSCRIPT1) determines which print processor receives the job. The data type also effects whether the print processor alters the job or not, and if so, how it alters the job. If the spooler is configured to append a separator file, it does so before sending the job to the print monitor for delivery to the print device.

Suspect a problem here if...	Confirm the problem is here if...
Page size, font or character set is wrong because of incorrect data type (UNIX).	Disk space is limited.
Extra FF or no FF	Changing default data type affects problem with MS-Network.
Job gets stuck in printer.	Problem is specific to one type of client.
Macintosh jobs print wrong resolution, font problems, black and white, instead of color.	Problem occurs only when using with Connect To or Create Printer.

Print Server Spooler Sends Job to Print Device

The print monitor receives the job and interacts with local hardware drivers and transport drivers to send the job to its destination. The components in this process are:

- Print monitors, LOCALMON, redirector, LPRMON, SFMMON, DECMON, or HPMON
- Transport protocols, either TCP/IP, NWLink, NetBEUI, or AppleTalk
- Network hardware (routers, bridges)

Suspect a problem here if...	Confirm the problem is here by...
All print devices accessed by a print monitor	Sending same job to same device by a different communication channel by using a different protocol, serial, or parallel port produces a successful print.
All print devices on one segment of the LAN	Trying different parallel or serial cable produces a successful print.
One make/model/rev of print device	
One parallel/serial cable	

Print Device Interprets Job

The final processing at this point is completed by the print device. The print device receives the print job from a hardware port. The print device interpreter interprets the job and produces hardcopy output.

Suspect a problem here if...	Confirm the problem is here by...
The problem is specific to a specific print device, or print device that is a variation of the same model, make, or revision.	Sending same print job to another print device of same make, model, and revision produces a successful print.

Questions and Answers

Specific questions and their answers are provided in the following sections.

Does the Windows NT print server support UNIX clients using LPSSCHED?

No, it does not support UNIX computers running LPSSCHED. UNIX client computers must have an LPD program installed when used with a Windows NT print server.

How can platform specific printer drivers be installed for a print client on a hardware platform different than the Windows NT print server?

Microsoft places all new or updated printer drivers onto its electronic services for public download.

It is also likely that a platform-specific printer driver is available on another computer in the network. It can be used to install the printer driver on the Windows NT print server.

▸ **To install platform specific printer drivers from another computer in the network,**

1. Log on the client computer using an user account that has Full Control permission.
2. Start Print Manager.
3. Select Server Viewer to display a list of available print servers.
4. Select a print server to display the printers (print devices) managed by the print server.
5. Select a printer.
6. Choose Printer Properties and choose OK.
7. Type the location of the printer driver in the Printer Properties dialog box. The printer driver is installed on the print server where it is available to any client computer with the same hardware platform, that is connected to the Windows NT print server.
8. If no additional printer drivers need to be installed, go to Step 11.
9. To install additional printer drivers for each unique print device serviced by the selected Windows NT print server, repeat Steps 5 through 7.
10. To install platform specific printer drivers on additional Windows NT print servers, repeat Steps 3 through 7.
11. Close all dialog boxes and Print Manager.

What type of problems occur when a print job has been assigned an incorrect data type?

Typical problems that occur when there is some error in the data type assignment and consequent job alteration of a print job include the following:

- LPR client print jobs include PCL or Postscript code, include incorrectly printed extended characters, or print in the print device's default font.
- Last page of a Microsoft network-based client print job does not print.
- Extra page prints after a Microsoft network client print job.
- Microsoft Network client print jobs include PCL or Postscript code, include incorrectly printed extended characters, or print in the print device's default font.
- Postscript print jobs sent from Macintosh clients do not print in color, print at a lower resolution, or fail.

How many printers can be supported by a Windows NT print server?

The limitation on the number of printers that can be attached to Windows NT print server is dependent on whether the print server is a Windows NT Workstation computer or a Windows NT Server computer. Windows NT Workstation and Windows NT Server have been optimized for different roles in the network. Windows NT Workstation is limited to 10 connections from other computers, it should be used as a print server in small-network situations. Windows NT Server has been optimized as a print, file, and application server. Windows NT versions 3.5 and 3.51 are capable of supporting more printers than Windows NT version 3.1.

The limitation on total printers attached to a Windows NT Server print server should be determined by daily requirements for print throughput. Print throughput is determined by the processor capabilities. For example, a print server with approximately 64 meg RAM and a 486 66 processor or higher processor with a high throughput network card can support 25 - 30 DLC printers or 40 TCP/IP printers.

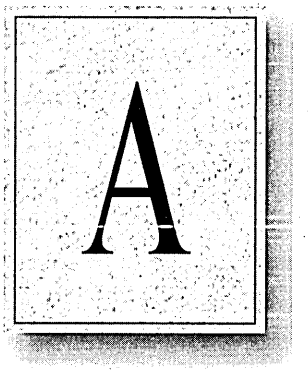
PART III

APPENDIXES

Part Three includes appendixes with information on major and minor revisions to the *Windows NT Resource Kit* documents, reference charts and tables for RAS, and reference lists for Microsoft's implementation of the UDP and TCP/IP protocol suite.

APPENDIX A

Major Revisions to Windows NT Update 1



Two of the appendixes from *Windows NT Update 1* have been revised and are included here as follows::

- Appendix B, "Major Revision to Windows NT Messages" is replaced with an enhanced version titled "Debugging Windows NT."
- The section titled "LAN Manager MIB II for Windows NT Objects" from Appendix A has also been replaced.

Debugging Windows NT

This section merges all the information about debugging Windows NT into one place. In addition to replacing Appendix B, as mentioned above, it also replaces the "Windows NT Debugger" section of Chapter 2, "Windows NT Executive Messages," in *Windows NT Messages*. The first two major sections in that chapter remain unchanged.

New material about the debugger and information about using the output from the **dumpexam** utility is also included here.

For Windows NT version 3.51, **windbg**, the utility used for reading memory dump files in earlier Windows NT releases, was replaced with a set of utilities that automatically read and interpret memory dump files. These new utilities simplify the process of dealing with kernel memory dump files and aid in sending memory dump files to support personnel for advanced analysis.

The debugging section of this appendix first defines terminology and provides an overview of debugging on Windows NT Server and Windows NT Workstation. Next, it describes setting up the computers for debugging. The remainder of this section describes how to create a memory dump file, the utilities that you can use to process the memory dump file, and interpreting the information in the memory dump file.

Terminology

This section defines some common terms and procedures that you should be familiar with before debugging Kernel STOP Errors.

Kernel STOP Error, Blue Screen, or Trap

When a Windows NT computer encounters hardware problems, inconsistencies within data necessary for its operation, or other similar errors, the operating system processes the error based upon the information entered in the Recovery dialog box. See the section titled "Creating a Memory Dump File" later in this chapter for information about the Recovery dialog box.

If the user did not check Automatically Reboot in the Recovery dialog box (which is the typical choice for a Windows NT Workstation), Windows NT displays a blue screen containing error information, and then stops. Seeing a blue screen typically brings a sense of despair to the user, and will usually bring a request for assistance in finding the cause of the problem.

Knowledge Base articles and other Windows NT documentation may refer to this type of error as *blue screen*, *kernel error*, or even *trap*. This chapter generally uses the term *Kernel STOP Error* for this type of error. When the context specifically refers to the Windows NT computer being stopped, with the blue screen being displayed, the term *blue screen* is used instead. The term *trap* is used in this chapter to means that the kernel has detected an error and will write a memory dump file as part of its processing of the error.

Symbols and Symbol Trees

Usually, when code is compiled to create executable files, one of two different versions of the executables can be created: a debug (also known as checked) version or a nondebug (also known as free) version. The checked version contains extra code that enables a developer to debug problems, but this means a larger and possibly slower executable file. The free version of the executable is smaller and runs at a normal speed, but cannot be debugged.

Windows NT combines the speed and smaller size of free versions with the debugging capabilities of the checked versions. All executables, drivers, dynamic-link libraries, and other program files in Windows NT are the free versions. However, each program has a corresponding symbol file, which contains the debug code that is normally part of the checked file. These symbol files are on the Windows NT CD, in the SUPPORT\DEBUG*platform*\SYMBOLS directories, where *platform* is I386, ALPHA, MIPS, or PPC. Within each SYMBOLS directory, there is one directory for each type of file (.EXE, .DLL, .SYS, and so forth). This structure is referred to as a *symbol tree*. The following directories exist in a standard symbol tree:

Directory	Contains symbols for
ACM	MSACM files
COM	.COM executable files
CPL	Control Panel applets
DLL	Dynamic-Link Libraries (.DLL files)
DRV	.DRV driver files
EXE	.EXE executable files
SCR	Screen Saver files
SYS	.SYS driver files

All of the utilities used to debug Windows NT or interpret memory dump files require a symbol tree containing the symbol files for the version of Windows NT you were running at the time of the Kernel STOP Error. With some utilities you need the SYMBOLS directory to be on your hard drive, in the *\systemroot* directory. With other utilities, you can specify the path to the SYMBOLS directory as a command line option or in a dialog box.

Target Computer

The term *target computer* refers to the computer on which the Kernel STOP Error occurs. This computer is the one that needs to be debugged. It can be a computer located within a few feet of the computer on which you run the debugger, or it can be a computer that you dial in to by using a modem.

Host Computer

The term *host computer* refers to the computer on which you run the debugger. This computer should be running a version of Windows NT that is at least as recent as the one on the target computer. It can use a later version of Windows NT, although that introduces some complications when setting up the debugger.

Debugging Overview

With Windows NT, there are two approaches you can take to finding the cause of Kernel STOP Errors:

- Debug the problem interactively on the target computer, using the kernel debuggers. This involves connecting a host computer to the target computer and entering debugging commands from the host computer. This approach lets you look in the memory of the target computer for the cause of the Kernel STOP Error.
- Set up the target computer to write the contents of its RAM to a memory dump file when a Kernel STOP Error occurs. You can then use the dump analysis utilities to analyze the memory dump, or send the memory dump file to technical support personnel for their analysis.

Kernel Debuggers

The Windows NT kernel debuggers, I386KD.EXE, ALPHAKD.EXE, MIPSKD.EXE, and PPCKD.EXE, are 32-bit EXEs used on the host computer to debug the kernel on the target computer. Each target hardware platform has its own set of utilities, which are provided on the Windows NT CD in the SUPPORT\DEBUG directory.

The kernel debuggers can be used for either remote or local kernel debugging. With local kernel debugging, the host computer is located within a few feet of the target computer, and the two computers communicate through a null-modem serial cable. With remote kernel debugging, the host computer can be any distance from the target computer, since communication takes place through modems.

The two computers send debugging (troubleshooting) information back and forth through communications ports that must be running at the same baud rate on each computer.

In general, you will use the kernel debugger approach to debug a problem with a computer that is running Windows NT Workstation. When you have a blue screen on the computer, you should restart it after recording the important information in the message. You can then continue running Windows NT until the message is redisplayed. When that happens, you should call your technical support group and request assistance with the debugging. They will decide whether to debug the Kernel STOP Error locally or remotely and will have you configure your system appropriately.

Dump Analysis Utilities

To use the dump analysis utilities, you must first configure your Windows NT computer to write a memory dump file when it gets a Kernel STOP Error. You use the Recovery dialog box to configure the target computer to write the memory file, as described in the section titled "Creating a Memory Dump File" presented later in this chapter. This file preserves the state of the computer at the time of the Kernel STOP Error, and the memory dump file can be used later by the dump analysis utilities to troubleshoot the problem. By using this option, you can run the dump analysis utilities on any Windows NT computer after you load the memory dump file, including the computer on which the Kernel STOP Error occurred.

This approach is usually the one to use for a Windows NT Server computer, because it minimizes the amount of time the server is unavailable. The default for a Windows NT Server computer is to automatically restart after writing an event to the system log, alerting administrators, and dumping system memory to the file called MEMORY.DMP. Therefore, to preserve memory dump files, you should rename the newest one each time a Kernel STOP Error occurs. You can then run the dump analysis utilities and send the information to your technical support group for processing.

Setting Up for Debugging

If you decide to use the kernel debugger to analyze the Kernel STOP Error, you need to set up the host to be debugged and connect your host and target Windows NT computers, using either a local null-modem cable or a modem. There are several procedures you need to do before you can start debugging:

- Set up the modem connection.
- Configure the target system for debugging.
- Set up a symbol tree on the host system.
- Set up the debugger on the host system.
- Start the debugger on the host system.

Note You do not need to do any of the procedures in this section if you are going to use the Recovery dialog box to create a memory dump file. See the section titled "Creating a Memory Dump File" later in this chapter for information about that alternative.

Setting up the Modem Connection

To do either local or remote debugging of your target computer, you need a connection between the target and host computers.

Setting Up for Local Debugging

To debug a Windows NT target computer by using a local host system, you need to connect the two computers with an industry-standard null-modem serial cable. The procedure for setting up the null-modem cable is the same on both the host and target computers. Be sure to start the host computer before restarting the target computer.

A standard, commercially available null-modem serial cable has the following configuration:

- Transmit Data connected to Receive Data
- Receive Data connected to Transmit Data
- Ground connected to Ground

For 9-pin and 25-pin D-subminiature connectors (known as db9 and db25, respectively), the cable connects as follows:

- Pin 2 to pin 3
- Pin 3 to pin 2
- Pin 7 to pin 7

The debugger on the host does not depend on any control pins (such as Data Terminal Ready, Data Set Ready, Request To Send, or Clear To Send). However, you may have to put a jumper from Data Terminal Ready to Data Set Ready and from Request To Send to Clear To Send in the connectors on both ends of the cable, as follows:

- On a db9 connector, this would be a jumper from pin 4 to pin 6 and a jumper from pin 7 to pin 8.
- On a db25 connector, this would be a jumper from pin 20 to pin 6 and from pin 4 to pin 5.

Setting Up for Remote Debugging

For remote debugging, you need to use a modem. Which port you use (COM1 or COM2) depends on how you prepare your target and host computers. The default connection for the target depends on the platform. The section titled "Configuring the Target System for Debugging" discusses setting the com port on the target computer. You set the com port on the host in an environment variable, as described in the section "Setting Up the Debugger Files on the Host."

Consult your modem documentation for specific information on the signals.

Connect the modem on the host first. When you are ready to connect to the modem on the target computer, see the last paragraph in the section "Setting Up the Debugger Files on the Host."

To set up the modem on the target computer for remote debugging:

- Connect the modem to one of the target computer's communication ports.
- Turn on auto-answer.
- Turn off flow control, hardware compression, and error detection.

Configuring the Target System for Debugging

Ordinarily, you will run Windows NT Workstation or Windows NT Server with the debug mode turned off, which is the default when Windows NT Setup installs the system. In that mode, a Kernel STOP Error will not enable the debugger that is part of the operating system. To switch Windows NT into debug mode, you have to edit the Windows NT startup file, and set debugging variables.

On an x86-based computer, you edit BOOT.INI, and include one of the two debug-mode switches: **/crashdebug** or **/debug**. On a RISC-based computer, you edit the firmware environment variable OSLOADOPTIONS to include **debug** or **crashdebug**.

If you use **crashdebug**, the debugger is loaded when you start, but remains inactive unless a kernel error occurs. This mode is useful if you are experiencing random, unpredictable kernel errors. When you include **debug**, the debugger is loaded when you start and can be activated at any time by a host debugger connected to the computer. This is the standard mode used when debugging problems that are regularly reproducible.

You can also change the default communications port and baud rate by editing the BOOT.INI file or the OSLOADOPTIONS variable. Because the code that enables remote kernel debugging resides in the Hardware Abstraction Layer (HAL), the defaults for the communications port and baud rate may vary from one computer to another.

When you are finished debugging the computer, you should turn the debug mode off again. To do this, repeat the process of editing the BOOT.INI file or the OSLOADOPTIONS environmental variable, and delete any values you set to enable debugging. On a RISC-based computer, this may mean the OSLOADOPTIONS variable has no values, which is acceptable.

If your target computer is in the following situation:

- It is stopped at a blue screen.
- You have edited the startup file to enable the debugger.
- You haven't configured the modem on the target computer.

You can still use the debugger, without rebooting the computer, by

- Connecting the modem to a running computer.
- Using terminal mode to send the appropriate commands to the modem to configure it.
- Moving it to the target computer while the modem is still powered up.

Preparing an X86-Based Computer

If you have more than one communications port, the default debug port is set to COM2. However, if you have a serial mouse attached to COM2, the default debug port is set to COM1. If you do not set the baud rate, the default baud rate is 9600 if a modem is attached, and 19200 for a null-modem cable. If necessary, you can verify later which baud rate has been set.

The procedures later in this section describe how to modify your startup file.

Note The modem connection should always be at 9600 baud, because this is the fastest you can communicate reliably over a modem with hardware compression, flow control, and error correction turned off.

For local debugging, the fastest reliable speed that will work on all systems is 19200. However, if both computers have a 16550 UART, it is possible to enable speeds as high as 115,200 baud, although such speeds can cause problems with the debugger on some systems. Also, since only text is being transferred, such high speeds will not decrease the time needed for debugging very much.

▸ **To prepare an x86-based target computer for remote or local debugging**

1. For remote debugging, connect a modem to the communications port, turn the power on, and set the modem to auto-answer.

   For local debugging, connect the null-modem serial cable to any available communications port.

You don't need to restart the system until step 8, unless the system is currently stopped at a blue screen.

If the target computer stops at a blue screen every time you boot it, or does not keep running long enough for you to edit the BOOT.INI file to enable the debugger, you have these options:

- If your boot partition is FAT, you can boot MS-DOS from a boot floppy disk and use the MS-DOS editor to edit BOOT.INI.
- If your boot partition is NTFS (or, if you are running Windows NT version 3.1 or 3.5, you may be using HPFS), you can install Windows NT into a different partition and boot from that partition. You have to use this method, because you cannot access files on an NTFS or HPFS partition from MS-DOS.

2. To turn off the system, hidden, and read-only attributes of the BOOT.INI file, you can use the Properties dialog box from the File menu in File Manager, or type the following at the command prompt:

```
attrib -s -h -r boot.ini
```

The BOOT.INI file is usually located in the root directory of the partition from which the Windows NT NTLDR program was loaded, which is ordinarily C:\.

3. Use the MS-DOS editor or a text editor such as Notepad to edit the BOOT.INI file, as follows:

edit boot.ini

The BOOT.INI file appears within the MS-DOS Editor window, and normally looks similar to the following example:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT35

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT35="Windows NT Workstation
Version 3.51"
multi(0)disk(0)rdisk(0)partition(1)\WINNT35="Windows NT Workstation
Version 3.51 [VGA mode]" /basevideo /sos
C:\="MS-DOS"
```

Each entry in the [operating systems] section should correspond to the options listed in the boot menu during a normal system startup.

4. Select the startup option that you normally use, and add either the **/debug** or **/crashdebug** switch at the end of the line.

An alternative to using the **/debug** switch is to add switches at the end of the line that explicitly set the communications port and baud rate your computer will use to send debugging information.

5. To specify the communications port, add the switch **/debugport=com***x*, where *x* is the communications port that you want to use.
6. To specify the baud rate, add the switch **/baudrate=<***baudrate***>**.

   See the section titled "Preparing an X86-Based Computer" earlier in this chapter for information about the baud rate.

   The following is an example of a startup-option line specifying the communications port and baud rate:

```
multi(0)disk(0)rdisk(0)partition(1)\WINNT35="Windows NT Workstation
Version 3.51" /debugport=com1 /baudrate=9600
```

7. Save the file and quit the text editor or the MS-DOS Editor.
8. Restart the computer to run under Windows NT.

   You may now contact your technical support group or a trained technician and have them call the modem to establish a remote debugging session.

Preparing an RISC-Based Computer

Preparing a RISC-based computer for remote or local kernel debugging also involves editing one line in a startup file, but you access that file in a different manner. The procedure for all Alpha systems should be the same. However, the path to the firmware menus may vary for MIPS-based and PPC-based systems. The options you use to configure the PPC-based system are the same as the options you select to configure the MIPS-based system.

On RISC-based computers, the default debug port is set to COM1, and the baud rate is always 19200.

To prepare a RISC-based target computer for remote or local debugging

1. For remote debugging, connect a modem to the communications port, turn the power on, and set the modem to auto-answer. As with x86-based computers, this could also be done on a second computer and then the modem could be moved to the target computer after a Kernel STOP Error occurs.

   For local debugging, connect the null-modem serial cable to any available communications port.
2. Restart the computer.

   The ARC System screen appears, displaying the main menu, from which you can select an action.

3. On a MIPS RISC-based system, choose Run setup to display the Setup menu and then choose Manage startup to display a menu of the boot options.

   On a Digital Alpha AXP RISC-based system or a PPC RISC-based system, select the options listed in the following table to get to the Boot selections menu.

Menu	Select this option:
System Boot	Supplementary menu
Supplementary	Setup the system
Setup	Manage boot selections

4. Choose Change a boot selection to display a list of the operating systems that are installed on this computer.
5. Choose the Windows NT operating system. If you have more than one version of Windows NT installed, select the one that you want to debug.

   A two-part screen appears for changing the current settings of the environment variables used to start the RISC-based computer. The environment variable that controls whether or not the RISC-based computer starts up in debug mode is the OSLOADOPTIONS variable.
6. To edit the value for the OSLOADOPTIONS variable, use the arrow keys to select it from the list of variables.

   Once selected, it appears in the Name box at the top of the screen.
7. Press ENTER to display the Value box.
8. Type **debug** or **crashdebug** in the Value box and press ENTER to save it and turn the debug mode on.

   You may also add a value that explicitly sets the communications port, as in the following example:

   ```
   OSLOADOPTIONS debug debugport=com2
   ```

   If you do not specify the debug port, the default debug port is set to COM1. Since RISC-based computers allow only a default baud rate of 19200, you do not need to specify the baud rate.
9. Press ESC to stop editing.

10. On a MIPS RISC-based system, choose Return to main menu and then Exit to return to the ARC System screen.

    On a Digital Alpha AXP RISC-based system, choose Supplementary menu, save your changes, and then choose Boot menu to return to the ARC System screen.

    If this is the first time that you have debugged a Digital Alpha AXP RISC-based system, after connecting the local host computer, you must do the following:

    - Shut down both computers.
    - Restart the host (debugger) computer.
    - Run ALPHAKD.EXE on the local host.
    - Restart the target (Digital Alpha AXP RISC-based) computer while ALPHAKD.EXE is running on the host computer to set up configuration information on the target computer, and prepare it for either local or remote debugging.

    Note Once you have carried out the preceding four steps by using a local host, you can use either a local or a remote host to debug the target.

11. Restart the RISC-based computer to run under Windows NT.

    You may now contact your technical support group or a trained technician and have them call the modem to establish a remote debugging session.

Setting Up the Symbol Tree on the Host

You set up the symbol tree on the host to match the version of Windows NT that you are running on the target computer.

The Windows NT Server and Windows NT Workstation CDs come with symbol trees already created. They are in SYMBOLS directories on the CD under SUPPORT\DEBUG*platform*, where *platform* is I386, ALPHA, MIPS, or PPC. The *platform* needs to match your target computer.

If you have not installed any service packs or hotfixes and do not have a multiprocessor system, then you might need to only specify the path to the correct SYMBOLS directory on the CD, or copy that directory to *\systemroot* and use this as the symbol path.

If you have installed service packs or hotfixes to Windows NT, or are using any HAL other than the standard, single processor HAL, you must construct a symbol tree.

▸ To construct a symbol tree

1. Copy the correct tree from the SUPPORT directory on the CD to your hard drive.
2. Copy the symbols for the updates you have applied into this tree in the order that you applied the updates, so that the later versions overwrite the earlier versions.
3. If you are using kernel debuggers to debug a multiprocessor or a single processor system that is using a special HAL, you must rename some of the symbol files.

   The kernel debuggers always load the files named NTOSKRNL.DBG for kernel symbols and HAL.DBG for HAL symbols. Therefore, you need to determine which kernel and HAL you are using, and rename the associated files to these filenames.

   If you have a computer with a multiprocessor, you need only rename NTKRNLMP.DBG to NTOSKRNL.DBG. These files are in the EXE subdirectory of the symbol tree.

   If your computer uses a special HAL, there are a number of possibilities. The following tables list the possible HAL files for each hardware platform. These tables list the actual name of the .DLL file as it exists on the CD and the uncompressed size of the file in bytes. Each .DLL file has a corresponding .DBG file, which is in the DLL subdirectory of the symbol tree. Determine which HAL you are using, and rename the associated .DBG file to HAL.DBG. If you are not sure which HAL you are using, compare the file size in the table with the HAL.DLL file on the target system. The HAL.DLL file can be found in *systemroot*\SYSTEM32.

Table A.1 HAL files for I386 systems

Filename	Uncompressed Size (bytes)	Description
HAL.DLL	48,416	Standard HAL for Intel systems
HAL486C.DLL	47,376	HAL for 486 c Step processor
HALAPIC.DLL	63,616	Uniprocessor version of HALMPS.DLL
HALAST.DLL	46,416	HAL for AST SMP systems
HALCBUS.DLL	79,776	HAL for Cbus systems
HALMCA.DLL	45,488	HAL for MCA-based systems (PS/2 and others)
HALMPS.DLL	65,696	HAL for most Intel multiprocessor systems

Table A.1 HAL files for I386 systems *(continued)*

Filename	Uncompressed Size (bytes)	Description
HALNCR.DLL	79,392	HAL for NCR SMP computers
HALOLI.DLL	40,048	HAL for Olivetti SMP computers
HALSP.DLL	52,320	HAL for Compaq Systempro
HALWYSE7.DLL	40,848	HAL for Wyse7 systems

Table A.2 HAL files for DEC Alpha systems

Filename	Uncompressed Size (bytes)	Description
HAL0JENS.DLL	56,800	Digital DECpc AXP 150 HAL
HALALCOR.DLL	69,120	Digital AlphaStation 600 Family
HALAVANT.DLL	66,752	Digital AlphaStation 200/400 Family HAL
HALEB64P.DLL	70,528	Digital AlphaPC64 HAL
HALGAMMP.DLL	72,896	Digital AlphaServer 2x00 5/xxx Family HAL
HALMIKAS.DLL	67,040	Digital AlphaServer 1000 Family Uniprocessor HAL
HALNONME.DLL	65,376	Digital AXPpci 33 HAL
HALQS.DLL	65,088	Digital Multia MultiClient Desktop HAL
HALSABMP.DLL	72,736	Digital AlphaServer 2x00 4/xxx Family HAL

Table A.3 HAL files for MIPS systems

Filename	Uncompressed Size (bytes)	Description
HALACR.DLL	43,648	ACER HAL
HALDTI.DLL	68,288	DESKStation Evolution
HALDUOMP.DLL	41,728	Microsoft-designed dual MP HAL
HALFXS.DLL	42,016	MTI with an r4000 or r4400
HALFXSPC.DLL	42,176	MTI with an r4600
HALNECMP.DLL	44,736	NEC dual MP
HALNTP.DLL	116,000	NeTpower FASTseries
HALR98MP.DLL	127,232	NEC 4 processor MP
HALSNI4X.DLL	95,520	Siemens Nixdorf UP and MP
HALTYNE.DLL	68,032	DESKStation Tyne

Table A.4 HAL files for PPC Systems

Filename	Uncompressed Size (bytes)	Description
HALCARO.DLL	169,504	HAL for IBM-6070
HALEAGLE.DLL	206,208	HAL for Motorola PowerStack and Big Bend
HALFIRE.DLL	136,576	Hal for Powerized_ES, Powerized_MX, and Powerized_MX MP
HALPOLO.DLL	169,152	HAL for IBM-6030
HALPPC.DLL	169,184	HAL for IBM-6015
HALWOOD.DLL	95,616	HAL for IBM-6020

In some cases, you might have a HAL file that was supplied by your computer manufacturer. If so, you need to obtain symbols for these files from the manufacturer, rename that symbol file to HAL.DBG, and place it in the DLL subdirectory of the symbol tree. For example, Compaq provides updated HAL files for their Proliant systems. This also applies if you have drivers from third party sources; obtain symbols from your third party vendor, and put them in the appropriate directory.

Setting Up the Debugger Files on the Host

To set up the debugger on the host, first ensure that you have the correct files available. These files should be copied from the SUPPORT\DEBUG*platform* directory to a debug directory on the hard drive, where *platform* matches the platform of the host computer.

If you are debugging Windows NT version 3.1 or 3.5 from a Windows NT computer running version 3.51, copy the files from a CD containing the version of Windows NT being used on the target computer.

Some files that you copy from the directory must match the platform of the target computer, as described in the following text. These files are necessary for kernel debugging:

- *platform*KD.EXE

  Where *platform* matches the platform of the target computer. The files are from the following list:

  - ALPHAKD.EXE
  - I386KD.EXE
  - MIPSKD.EXE
  - PPCKD.EXE
- IMAGEHLP.DLL
- KDEXT*platform*.DLL

  Where *platform* matches the platform of the target computer. The files are from the following list:

  - KDEXTALP.DLL
  - KDEXTX86.DLL
  - KDEXTMIP.DLL
  - KDEXTPPC.DLL

For instance, if your host computer is a 486 computer and the target computer is a MIPS RISC-based system, you would copy the following files from the SUPPORT\DEBUG\I386 directory:

- MIPSKD.EXE
- IMAGEHLP.DLL
- KDEXTMIP.DLL

Once you have set up the symbol tree and copied the necessary files to it, use a batch file or command line to set the following environment variables on the host:

Variable	Purpose
_NT_DEBUG_PORT	COM port being used on host for debugging
_NT_DEBUG_BAUD_RATE	Max baud rate for debug port. On x86-based computers, 9600 or 19200 for modems, 19200 for null-modem serial cables. On RISC-based computers, always 19200.
_NT_SYMBOL_PATH	Path to symbols directory
_NT_LOG_FILE_OPEN	Optional, the name of the file to which to write a log of the debug session

Once these environment variables have been set, you can start the host debugger.

Note Setting the _NT_LOG_FILE_OPEN variable does not always result in a log file being written. You can also create the log file from the debugger. The command is:

```
.logopen <pathname>
```

You may also need to issue the **!reload** command to get this to work.

Starting the Debugger on the Host

You can start the host debugger from the command line or a batch file, using the name of the executable as the command. Each debugger supports the following command-line options:

-b
Causes the debugger to stop execution on the target computer as soon as possible, by causing a debug breakpoint (INT 3).

-c
Causes the debugger to request a resync on connect. Resynchronization ensures that the host and target computers are communicating in sequence.

-m
Causes the debugger to monitor modem control lines. The debugger is only active when the carrier detect (CD) line is active; otherwise, the debugger is in terminal mode, and all commands are sent to the modem.

-n
Causes symbols to be loaded immediately, rather than in a deferred mode.

-v
Verbose mode; displays more information about such things as when symbols are loaded.

-x
Causes the debugger to break in when an exception first occurs, rather than letting the application or module that caused the exception deal with it.

The most commonly used switches are **-v** (verbose) and **-m** (for modem debugging).

Generally, the best way to start the debugger is to create a batch file with the necessary commands to set the environment variables, followed by the command to start the correct kernel debugger.

Using the Remote Utility to Start the Debugger

If the host computer is on a network, you might choose to use the **remote** utility, included in the *Windows NT Resource Kit*, to start the debugger. **Remote** is a server/client utility that provides remote network access via named pipes to applications that use STDIN and STDOUT for input and output. This allows users at other computers on the network to connect to your host debugger session and either view the debugging information or enter commands themselves. The syntax for starting the server (host) end of the remote session is as follows:

remote /s "*command*" *Unique_Id* [**/f** *foreground_color*|**/b** *background_color*]

For example:

```
REMOTE /S "i386kd -v" debug
```

The server session is ended with **@K**.

To interact with this session from some other computer, use the **remote /c** command. The syntax of this command is as follows:

remote /c *ServerName Unique_Id* [**/l** *lines_to_get*|**/f** *foreground_color*|**/b** *background_color*]

To exit from the remote session on a client and leave the debugger running on the host computer, use **@Q**.

For example, if a session with the ID **debug** had been started on the host computer \\Server1 by using the **remote /s** command, you could connect to it with the command

```
REMOTE /C server1 debug
```

For more information on using the remote command, see the RKTOOLS. HLP file on the *Windows NT Resource Kit* CD.

Examples

Let us suppose the following:

- Debugging needs to take place over a null-modem serial cable on COM2.
- The symbols are on a CD on the E drive.
- A log file called DEBUG.LOG is to be created in C:\TEMP.

Note The log file holds a copy of everything you see on the debug screen during your debug session. All input from the person doing the debugging, and all output from the kernel debugger on the target system, is written to the log file.

A sample batch file for local debugging is:

```
REM Target computer is local
set _NT_DEBUG_PORT=com2
set _NT_DEBUG_BAUD_RATE=19200
set _NT_SYMBOL_PATH=e:\support\debug\i386\symbols
SET _NT_LOG_FILE_OPEN=c:\temp\debug.log
remote /s "i386kd -v" debug
```

The last line of the batch file uses the **remote** utility to start the host debugger. This lets people on Windows NT computers who are networked to the host computer (and who have a copy of the **remote** utility) connect to the debug session by using the following command:

remote /c *computername* **debug**

where *computername* is the name of the host computer.

To allow remote debugging, which requires the use of a modem, begin with the batch file in the previous example. You should change the baud rate to 9600, and add the **-m** switch to the last line. The result is as follows:

```
REM Target computer is remote from the host
set _NT_DEBUG_PORT=com2
set _NT_DEBUG_BAUD_RATE=9600
set _NT_SYMBOL_PATH=e:\support\debug\i386\symbols
SET _NT_LOG_FILE_OPEN=c:\temp\debug.log
remote /s "i386kd -v -m" debug
```

The batch file should be executed from the directory that contains the debugger files.

When you start the debugger, one of two screens appears, depending upon whether you are doing local debugging or remote debugging.

When doing local debugging, the following screen appears:

```
************************************
***********   REMOTE  ************
***********   SERVER  ************
************************************
To Connect: Remote /C BANSIDHE debug

Microsoft(R) Windows NT Kernel Debugger
Version 3.51
(C) 1991-1995 Microsoft Corp.

Symbol search path is:
KD: waiting to connect...
```

Once at this screen, you can use CTRL+C to break in to the target computer, if it is still running. If the target is currently stopped at a blue screen, you will probably break in automatically. If you have any problems, try using CTRL+R to force a resync between the host and target computers.

If you are doing remote debugging, the same screen appears, with the following extra line:

```
KD: No carrier detect - in terminal mode
```

In this case, the debugger is in terminal mode, and you can issue any of the standard AT commands to your modem. Begin by sending commands to disable hardware compression, flow control, and error correction. These commands will vary from modem to modem, so consult your modem documentation. Once you connect to the target system and have a carrier detect (CD) signal, you are returned to the debugger.

Creating a Memory Dump File

If you do not want to or are unable to do local or remote debugging, you can configure your Windows NT Server or Windows NT Workstation computer to write a memory dump file each time it gets a Kernel STOP Error. This file contains all the information needed by the **dumpexam** utility to troubleshoot the Kernel STOP Error as if you were connected to a live computer experiencing the problem.

Using the memory dump file enables you to examine the error at any time, so you can immediately restart the computer that failed instead of having your target computer unavailable while you are using the debugger. The only drawback to this method is that you must have sufficient space on a hard disk partition for the resulting memory dump file, which will be as large as your RAM memory. Therefore, whenever a Kernel STOP Error occurs, a computer with 32 MB of RAM will produce a 32 MB memory dump file. You must also have a page file on your *systemroot* drive that is at least as large as your RAM memory.

▸ **To configure Windows NT to save STOP information to a memory dump file**

1. In Control Panel, choose the System option.
2. In the System dialog box, choose the Recovery button.
3. In the Recovery dialog box, select the Write Debugging Information To check box, and either accept the default path and filename (C:*systemroot*\MEMORY.DMP) or type your own names in the text box.

   To have this memory dump file overwrite any file of the same name, select the Overwrite Any Existing File check box. If you set the option to overwrite an existing file, you should rename or move the file so it does not get overwritten before you have time to process it. If you clear this check box, Windows NT will not write a memory dump file if there is already a file by that name.

Using Utilities to Process Memory Dump Files

Included on the Windows NT Server and Windows NT Workstation version 3.51 CDs are three utilities for processing memory dump files: **dumpflop**, **dumpchk**, and **dumpexam**. All three utilities are on the CDs in the SUPPORT\DEBUG*platform* directories, where *platform* is I386, ALPHA, MIPS, or PPC.

The primary purpose of these utilities is to create floppy disks or a text file that you can send to technical support personnel for their analysis.

Dumpflop

Dumpflop is a command-line utility that you can use to write a memory dump file in segments to floppy disks, so it can be sent to a support engineer. This is rarely the most efficient way to send a memory dump file, but it is sometimes the only way. **Dumpflop** compresses the information it writes to the floppy disks, so a 32 MB memory dump file will generally fit onto 10 floppy disks, rather than the 20 or more disks you might expect. **Dumpflop** does not require access to symbols.

To store the crash dump onto floppy disks, use **dumpflop** with the following command-line syntax:

dumpflop *[options] <CrashDumpFile> [<Drive>:]*

To assemble a crash dump from floppy disks, use DUMPFLOP with the following command-line syntax:

dumpflop *[options] <Drive>: [<CrashDumpFile>]*

In either case, the options are as follows:

-?
Displays the command syntax.

-p
Only prints the crash dump header on an assemble operation.

-v
Shows compression statistics.

-q
Formats the floppy disk, when necessary, before writing the memory dump file to the floppy disk. When reading the floppy disks to assemble the file, overwrites an existing memory dump file.

If executed with no parameters, **dumpflop** attempts to find a memory dump file in the *\systemroot* directory (the default location for creating a memory dump file) and writes it to floppy disks on the A drive.

Dumpchk

Dumpchk is a command-line utility that you can use to verify that a memory dump file has been created correctly. **Dumpchk** does not require access to symbols.

Dumpchk has the following command line parameters:

dumpchk *[options] CrashDumpFile*

where the options are as follows:

-?
Displays the command syntax.

-p
Prints the header only (with no validation).

-v
Specifies verbose mode.

-q
Performs a quick test.

Dumpchk displays some basic information from the memory dump file and then verifies all the virtual and physical addresses in the file. If any errors are found in the memory dump file, it reports them. The following is an example of the output of a **Dumpchk** command:

```
Filename . . . . . .memory.dmp
Signature. . . . . .PAGE
ValidDump. . . . . .DUMP
MajorVersion . . . . .free system
MinorVersion . . . . .807
DirectoryTableBase . .0x00030000
PfnDataBase. . . . . .0xffb7e000
PsLoadedModuleList . .0x80196d40
PsActiveProcessHead. .0x80196c38
MachineImageType . . .i386
NumberProcessors . . .1
BugCheckCode . . . . .0xc000021a
BugCheckParameter1 . .0xe17b7b68
BugCheckParameter2 . .0xc0000005
BugCheckParameter3 . .0x00000000
BugCheckParameter4 . .0x00000000

ExceptionCode. . . . .0x80000003
ExceptionFlags . . . .0x00000001
ExceptionAddress . . .0x8015f015

NumberOfRuns . . . . .0x3
NumberOfPages. . . . .0x3f9e
Run #1

 BasePage . . . . . .0x1
 PageCount. . . . . .0x9e
Run #2

 BasePage . . . . . .0x100
 PageCount. . . . . .0xec0
Run #3

 BasePage . . . . . .0x1000
 PageCount. . . . . .0x3040

**************
**************--> Validating the integrity of the PsLoadedModuleList
**************
```

```
**************
***************--> Performing a complete check (^C to end)
**************
**************
***************--> Validating all physical addresses
**************
**************
***************--> Validating all virtual addresses
**************
```

In this example, the most important information (from a debugging standpoint) is the following:

```
MajorVersion . . . . .free system
MinorVersion . . . . .807
MachineImageType . . .i386
NumberProcessors . . .1
BugCheckCode . . . . .0xc000021a
BugCheckParameter1 . .0xe17b7b68
BugCheckParameter2 . .0xc0000005
BugCheckParameter3 . .0x00000000
BugCheckParameter4 . .0x00000000
```

This information can be used to determine what Kernel STOP Error occurred and, to a certain extent, what version of Windows NT was in use.

Dumpexam

Dumpexam is a command-line utility that examines a memory dump file, extracts information from it, and writes it to a text file. This text file can then be used by support personnel to determine the cause of the Kernel STOP Error. In many cases, the **dumpexam** analysis provides enough information for support personnel to determine the cause of the error without directly accessing the memory dump file.

Three files are required to run **dumpexam**, and they all must be in the same directory. You can find them on the Windows NT Server or Windows NT Workstation CD in the directory SUPPORT\DEBUG*platform*, where *platform* is I386, ALPHA, MIPS, or PPC. The first two files are:

- DUMPEXAM.EXE
- IMAGEHLP.DLL

The third file is one of the following, depending on the type of computer on which the memory dump file was generated:

- KDEXTX86.DLL
- KDEXTALP.DLL
- KDEXTMIP.DLL
- KDEXTPPC.DLL

You can run **dumpexam** directly off the CD with no parameters, if

- The computer on which the dump occurred was running Windows NT version 3.51.
- You have not applied any hotfixes or service packs on that computer.
- The memory dump file you want to examine is in the location specified in the Recovery dialog box.

Dumpexam creates a text file called MEMORY.TXT, located in the same directory as the MEMORY.DMP file, that contains information extracted from the memory dump file.

You can also use **dumpexam** to examine memory dump files created on computers running earlier versions of Windows NT. However, it will only execute on a system running Windows NT version 3.51, so you will need to move the memory dump file or access it over the network. Additionally, you will need to replace the KDEXT*.DLL files listed above with copies from the version of Windows NT that was running on the computer on which the dump occurred. These files contain debug information specific to that version of Windows NT. You must also specify the path to the symbols for the operating system version that was running on that computer.

Syntax for Dumpexam

The syntax for **dumpexam** is as follows:

dumpexam *[options] [CrashDumpFile]*

where

-?
Displays the command syntax.

-v
Specifies verbose mode.

-p
Prints the header only.

-f *filename*
Specifies the output file name.

-y *path*
Sets the symbol search path.

You need to specify the memory dump file path (using the **-f** option) only if you have moved the memory dump file.

You need to specify the symbol search path (using the **-y** option) only if you are using an alternate symbol path. The symbol path for **dumpexam** can contain several directories, separated by semicolons(;). These directories are searched in the order in which they are listed, so you should list directories with the most recently installed hotfixes or service packs first.

Examples

In the first example, the memory dump file was created on a computer running Windows NT Workstation version 3.51, and no service packs were installed. The symbols are all in the directory C:\SYMBOLS. The memory dump file is in the directory C:\DUMP and is called MACHINE1.DMP. The command line reads as follows:

```
dumpexam -y c:\symbols c:\dump\machine1.dmp
```

The results of the exam will be in *\systemroot*\MEMORY.TXT.

In the next example, the memory dump file was created on a DEC Alpha computer running Windows NT Server version 3.5, with Service Pack 2 installed. The Service Pack 2 symbols are in D:\SP2\SYMBOLS. The Windows NT Server 3.5 symbols are on the product CD, which is in the E drive. The memory dump file MEMORY.DMP is in D:\TEMP. The output file is to be put in the same directory as the memory dump file. The command line reads as follows:

```
dumpexam -y d:\sp2\symbols;e:\support\debug\alpha -f d:\temp\memory.txt
d:\temp\memory.dmp
```

Using the Dumpexam Output File

Dumpexam reads a memory dump file, executes debugger commands on it, and writes the output in a text file, called MEMORY.TXT, by default. The same debugger commands are executed on each memory dump file.

A full interpretation of the output generally requires knowledge of Windows NT kernel processes and the ability to read assembly language; however, there are some guidelines you can follow to get an idea of what the output means. This section first describes each section of the memory dump file output, giving sample output and a description. Then several common traps are discussed, along with guidelines on which sections of the MEMORY.TXT file will aid in determining what caused the Kernel STOP Error.

Since the primary purpose of this utility is to create a text file to send to support personnel, the descriptions in this section do not provide complete details of the contents of the MEMORY.TXT file.

Systemwide Information in MEMORY.TXT

The output of a particular memory dump file may not have information for all of the sections listed below, although most sections will have some information in them.

The following sections of the MEMORY.TXT file each occur once, as they include information that applies to the whole system. These sections are listed in the order in which they appear in MEMORY.TXT.

Windows NT Crash Dump Analysis

The first section of output is Windows NT Crash Dump Analysis, which looks like the following:

```
*****************************************************************
**
** Windows NT Crash Dump Analysis
**
*****************************************************************
*
Filename . . . . . .c:\temp\dumps\mac.dmp
Signature. . . . . .PAGE
ValidDump. . . . . .DUMP
MajorVersion . . . . .free system
MinorVersion . . . . .1057
DirectoryTableBase . .0x0006f005
PfnDataBase. . . . . .0x83fce000
PsLoadedModuleList . .0x800ee5c0
PsActiveProcessHead. .0x800ee590
MachineImageType . . .alpha
NumberProcessors . . .2
```

```
BugCheckCode . . . . .0x0000002e
BugCheckParameter1 . .0x00000000
BugCheckParameter2 . .0x00000000
BugCheckParameter3 . .0x00000000
BugCheckParameter4 . .0x00000000
ExceptionCode. . . . .0x80000003
ExceptionFlags . . . .0x00000001
ExceptionAddress . . .0x800bc140
```

Most of the information here is only useful for determining whether or not the memory dump file is corrupt, but the following items are most important, especially if you did not record any information from the blue screen generated when the computer trapped:

BugCheckCode: Lists the number of the stop that occurred, which can be used by support personnel to determine what trap occurred. For information on bug check codes, see Chapter 4, "Message Reference," in *Windows NT Messages*. Descriptions of the STOP code message start on page 441 in chapter 4 and are in numerical order. In the preceding example, the code was 0x0000002e, which is a DATA_BUS_ERROR.

BugCheckParameters: These are the four parameters that are normally included with each STOP code. The description of the stop code in *Windows NT Messages* includes the meaning of the parameters for some of the Kernel STOP Errors.

Symbol File Load Log

This section of the MEMORY.TXT file includes any errors that were generated when the symbols were loaded. If no errors were generated, this section will be blank.

!drivers

The **!drivers** command is a debug command that is used to list information on all the device drivers loaded on the system. The information for the device drivers looks like this:

```
*******************************************************************
** !drivers
*******************************************************************
*
Loaded System Driver Summary

Base    Code Size     Data Size    Driver Name    Creation Time
80080000 f76c0 (989 kb) 1f100 (124 kb) ntoskrnl.exe Fri May 26 15:13:00 1995
80400000 d980 ( 54 kb) 4040 ( 16 kb)    hal.dll Tue May 16 16:50:34 1995
80654000 3f00 ( 15 kb) 1060 ( 4 kb) ncrc810.sys Fri May 05 20:07:04 1995
8065a000 a460 ( 41 kb) 1e80 ( 7 kb) SCSIPORT.SYS Fri May 05 20:08:05 1995
```

The following information can be determined from the above output:

Base: The starting address of the device driver code, in hex. When the code that causes a trap falls between the base address for a driver and the base address for the next driver in the list, then that driver is frequently the cause of the fault. For instance, the base for ncrc810.sys is 0x80654000. Any address between that and 0x8065a000 belongs to this driver.

Code Size: The size of the driver code in hex and in decimal kilobytes.

Data Size: The amount of space allocated to the driver for data, in hex and in decimal kilobytes.

Driver Name: The driver filename.

Creation time: The link date of the driver. This should not be confused with the file date of the driver, which can frequently be set by external utilities. The link date is set by the compiler when a driver or executable is compiled. It should be close to the file date, but it won't always be the same.

!locks

The **!locks** command is a debugger command that displays all locks held on resources by threads. A lock can be shared or exclusive, which means no other threads can access that resource. This information is frequently useful when a deadlock has occurred on a system, because a deadlock is caused when one non-executing thread holds an exclusive lock on a resource needed by an executing thread.

```
*****************************************************************
** !locks -p -v -d
*****************************************************************
*
**** DUMP OF ALL RESOURCE OBJECTS ****
KD: Scanning for held locks.................

Resource @ 0xffb6ed14  Shared 2 owning threads
   Threads: ffb3bb70-01
0012fb50: Unable to read ThreadCount for resource

Resource @ 0xffb6ecdc  Shared 2 owning threads
   Threads: ffb3bb70-02
0012fb50: Unable to read ThreadCount for resource
```

!memusage

The **!memusage** command gives a short description of the current memory use of the system. Then it gives a much longer list of the memory usage summary. The output looks something like this:

```
*******************************************************************
** !memusage
*******************************************************************
*
 loading PFN database.....................................................

       Zeroed:  405 ( 3240 kb)
        Free:   0 (   0 kb)
      Standby:  3242 ( 25936 kb)
      Modified:  135 ( 1080 kb)
  ModifiedNoWrite:   0 (   0 kb)
    Active/Valid:  4410 ( 35280 kb)
     Transition:   0 (   0 kb)
      Unknown:   0 (   0 kb)
       TOTAL:  8192 ( 65536 kb)

Usage Summary in KiloBytes (Kb):
Control Valid Standby Dirty Shared Locked PageTables name
80975548   0   56   0   0   0   0 mapped_file(oemnxpip.inf)
80975248   0   16   0   0   0   0 mapped_file(oemnxpnb.inf)
8096aa68   0  160   0   0   0   0 mapped_file(SFMATALK.SY_)
80974f48   0  104   0   0   0   0 mapped_file(oemnxpsm.inf)
809758e8   0   96   0   0   0   0 mapped_file(utility.inf)
```

While this section provides some information for some memory leak issues, it is more useful to refer to the **!vm** section for memory information for most common Kernel STOP Errors.

!vm

The **!vm** command lists the systems virtual memory usage. **!vm** output commonly looks like this:

```
*******************************************************************
** !vm
*******************************************************************
*
*** Virtual Memory Usage ***
    Physical Memory:  32784  (131136 Kb)
    Available Pages:  27435  (109740 Kb)
    Modified Pages:    33  (  132 Kb)
    NonPagedPool Usage:  461  ( 1844 Kb)
    PagedPool 0 Usage:  1519  ( 6076 Kb)
    PagedPool 1 Usage:  125  (  500 Kb)
    PagedPool 2 Usage:  149  (  596 Kb)
```

```
PagedPool Usage:    1793  ( 7172 Kb)
Shared Commit:      173   (  692 Kb)
Process Commit:     254   ( 1016 Kb)
PagedPool Commit:   1793  ( 7172 Kb)
Driver Commit:      321   ( 1284 Kb)
Committed pages:    4261  ( 17044 Kb)
Commit limit:       80792 (323168 Kb)
```

All memory usage is listed in pages and in kilobytes. The most useful information in the **!vm** section for diagnosing problems is:

Physical memory: The total physical memory in the system.

Available Pages: The number of pages of memory available on the system, both virtual and physical. If this is low, it might indicate a problem with a process allocating too much virtual memory.

NonPagedPool Usage: The amount of pages allocated to the nonpaged pool. The nonpaged pool is memory that cannot be swapped out to the pagefile, so it must always occupy physical memory. This number should rarely be larger than 10% of the total physical memory. If it is larger, this is usually an indication that there is a memory leak somewhere in the system.

!errlog

The debugger sometimes keeps track of kernel errors logged by the system when a problem occurs. The **!errlog** section contains a dump of this log. In most cases, the error log will be empty, but if it isn't, it can sometimes be used to determine the component or process that caused the blue screen.

!irpzone full

An Interrupt Request Packets (IRP) is a data structure used by device drivers and other kernel mode modules to communicate information to each other. The **!irpzone full** command displays a list of all the pending IRPs on the system. The following information is displayed in this section:

```
*******************************************************************
** !irpzone full
*******************************************************************
*
Small Irp list
Irp is from zone and active with 1 stacks 1 is current
 No Mdl System buffer = fb564000 Thread fb5688a0: Irp stack trace.
 cmd flg cl Device  File   Completion-Context
> d  0 1 fb56a030 fb56cd48 00000000-00000000  pending
       \FileSystem\MacSrv
            Args: 00001000 00000000 00121020 00000000
```

```
Large Irp list
Irp is from zone and active with 4 stacks 5 is current
 No Mdl Thread fb4b6860: Irp is completed. Pending has been returned
 cmd flg cl Device  File   Completion-Context
  0  0 0 00000000 00000000 00000000-00000000

            Args: 00000000 00000000 00000000 00000000
  0  0 0 00000000 00000000 00000000-00000000

            Args: 00000000 00000000 00000000 00000000
  0  0 0 00000000 00000000 00000000-00000000

            Args: 00000000 00000000 00000000 00000000
  d  0 0 fb5e3020 00000000 f8a8c711-fb48df10
      \FileSystem\Ntfs SrvCompleteRfcbClose
            Args: 00000000 00000000 00000000 00000000
```

Each entry lists information about a different IRP and points to the driver that currently owns the IRP. This information can be useful when the trap analysis (which occurs later in the MEMORY.TXT file) points to a problem with a corrupted or bad IRP. Generally, the IRP listing will contain several entries in both the small and large IRP list.

!process 0 0

This command lists all processes and their headers. The process header list will contain entries like the following:

```
*******************************************************************
** !process 0 0
*******************************************************************
*
**** NT ACTIVE PROCESS DUMP ****
PROCESS fb667a00 Cid: 0002  Peb: 00000000 ParentCid: 0000
  DirBase: 00030000 ObjectTable: e1000f88 TableSize: 112.
  Image: System

PROCESS fb5edde0 Cid: 0018  Peb: 7ffdf000 ParentCid: 0002
  DirBase: 01587000 ObjectTable: e11d59a8 TableSize: 48.
  Image: SMSS.EXE
```

The important information in the **!process 0 0** section is:

Process ID: The 8 character hex number after the word PROCESS is the process ID. This is used by the system to track the process. For the first process in the example, this is fb667a00.

Image: The name of the module that owns the process. In the above example, the first process is owned by System, the second by SMSS.EXE.

!process 0 7

This command also lists process information, but instead of just listing the process header, it lists all information about the process, including all threads owned by each process. This listing is generally very long, because each system has a large number of processes and each process will have one or more threads. In addition, if the stack from a thread is resident in kernel memory (as opposed to swapped to the page file), it will be listed after the thread information. Most process and thread listings will look like the following:

```
*****************************************************************
** !process 0 7
*****************************************************************
*
**** NT ACTIVE PROCESS DUMP ****

PROCESS fb667a00 Cid: 0002  Peb: 00000000 ParentCid: 0000
  DirBase: 00030000 ObjectTable: e1000f88 TableSize: 112.
  Image: System
  VadRoot fb666388 Clone 0 Private 4. Modified 9850. Locked 0.
  FB667BBC MutantState Signalled OwningThread 0
  Token                e10008f0
  ElapsedTime             15:06:36.0338
  UserTime              0:00:00.0000
  KernelTime             0:00:54.0818
  QuotaPoolUsage[PagedPool]     1480
Working Set Sizes (now,min,max) (3, 50, 345)
  PeakWorkingSetSize         118
  VirtualSize              1 Mb
  PeakVirtualSize            1 Mb
  PageFaultCount           992
  MemoryPriority           BACKGROUND
  BasePriority             8
  CommitCharge             8

    THREAD fb667780 Cid 2.1 Teb: 00000000 Win32Thread: 80144900 WAIT:
(WrFreePage) KernelMode Non-Alertable
    80144fc0 SynchronizationEvent
    Not impersonating
    Owning Process fb667a00
    WaitTime (seconds)   32278
    Context Switch Count  787
    UserTime          0:00:00.0000
    KernelTime         0:00:21.0821
    Start Address Phase1Initialization (0x801aab44)
    Initial Sp fb26f000 Current Sp fb26ed00
    Priority 0 BasePriority 0 PriorityDecrement 0 DecrementCount 0
```

```
ChildEBP RetAddr Args to Child
    fb26ed18 80118efc c0502000 804044b0 00000000 KiSwapThread+0xb5
    fb26ed3c 801289d9 80144fc0 00000008 00000000
KeWaitForSingleObject+0x1c2
```

The following entries in the process information can be important:

UserTime: Lists the amount of time the process has been running in user mode.

KernelTime: Lists the amount of time the process has been running in kernel mode. If either of the values for User or Kernel time is exceptionally high, it might identify a process that is taking up all the resources and starving out the system.

Working Set sizes: Lists the current, minimum, and maximum working set size for the process, in pages. An exceptionally large working set size can also be a sign of a process that is leaking memory or using too many system resources.

QuotaPoolUsage: These two entries list the paged and nonpaged pool used by the process. On a system with a memory leak, looking for excessive nonpaged pool usage on all the processes can tell you which process has the memory leak.

In addition to the process list information, the thread information also contains a list of the resources on which the thread has locks. This information is listed right after the thread header. In this example, the thread has a lock on one resource, a SynchronizationEvent with an address of 80144fc0. By comparing this address to the list of locks shown in the **!locks** section, you can determine which threads have exclusive locks on resources.

Processor Specific Information in MEMORY.TXT

The following sections in the MEMORY.TXT file occur once for each processor on the system. In a four-processor system you will find these sections repeated for processors 0 through 3. In addition, some traps will generate a few extra sections, most noticably a STOP 0x0000001E.

Register Dump For Processor #X

A dump of the state of all registers at the time of the trap is included in this section. For an x86-based system, it will appear as follows:

```
*******************************************************************
** Register Dump For Processor #0
*******************************************************************
*
eax=ffdff13c ebx=00000000 ecx=00000000 edx=fb5a7db4 esi=00000d31 edi=00000d31
eip=8013b446 esp=f88b6de4 ebp=f88b6df8 iopl=0     nv up di pl nz na pe nc
cs=0008 ss=0010 ds=0023 es=0023 fs=0030 gs=0000       efl=00000286
```

```
cr0=8001003b cr2=00000d31 cr3=00030000 dr0=00000000 dr1=00000000 dr2=00000000
dr3=00000000 dr6=ffff0ff0 dr7=00000400 cr4=00000000
gdtr=80036000  gdtl=03ff idtr=80036400  idtl=07ff tr=0028 ldtr=0000
```

For a RISC-based system, the register dump will vary from processor type to processor type. The following example is from a DEC Alpha system:

```
v0=80006000   t0=00000000   t1=00000000   t2=800ef538
t3=00000008   t4=00000000   t5=800ec440   t6=00000000
t7=00000000   s0=c53f2000   s1=00000002   s2=00000001
s3=00000000   s4=00000001   s5=0018da83   fp=fc90f940
a0=00000002   a1=c53f2000   a2=c53f2000   a3=00000000
a4=00000000   a5=00000002   t8=800ed580   t9=80a4752c
t10=c53f2000  t11=80a4752c   ra=8009b0bc  t12=80a61ecc
at=a0000000   gp=800ed430   sp=fc90f890  zero=00000000
pcr=0000000008000000  softfpcr=0000000000000000    fir=800bf2fc
psr=0000000a
mode=0 ie=1 irql=2
```

In general, the register dump will be valuable only if you are skilled in reading assembly language on the system you are debugging.

Stack Trace for Processor X

The next section includes a trace of the stack for that processor. The stack trace is very important, because it tells you what functions were called and can be used to trace back from a trap to determine why it happened. Included right after each stack trace is a section of disassembled code from the area in memory around the last instruction in the stack. This information will also look different, depending upon the platform type you are running on.

The first example is an excerpt from an x86-based computer on which a STOP 0x0000000A occurred:

```
*******************************************************************
** Stack Trace
*******************************************************************
*
ChildEBP RetAddr Args to Child
f88b6e00 f89805b0 fb55ea88 fb55e988 fb55ea88 KiTrap0E+0x252 (FPO: [0,0,0])
f88b6df8 fb4a71a0 fb4a6028 f89805b0 fb55ea88 NTSend+0x142

  8013B430: 8B 4D 64      mov     ecx,dword ptr [ebp+64h]
  8013B433: 83 E1 02      and     ecx,2
  8013B436: D1 E9       shr     ecx,1
  8013B438: 8B 75 68      mov     esi,dword ptr [ebp+68h]
  8013B43B: 56          push    esi
  8013B43C: 51          push    ecx
```

```
  8013B43D: 50          push    eax
  8013B43E: 57          push    edi
  8013B43F: 6A 0A       push    0Ah
  8013B441: E8 00 C6 FD FF   call    KiTrap0E+24Eh
--->8013B446: F7 45 70 00 00 02 test    dword ptr [ebp+70h],offset KiTrap0E+255h
      00
  8013B44D: 74 0D       je      KiTrap0E+268h
  8013B44F: 83 3D EC 05 14 80 cmp     dword ptr [KiTrap0E+25Dh],0
      00
  8013B456: 0F 85 29 FE FF FF jne     KiTrap0E+264h
  8013B45C: 83 3D 38 49 14 80 cmp     dword ptr [KiTrap0E+26Ah],0
      00
  8013B463: 0F 85 1C FE FF FF jne     KiTrap0E+271h
  8013B469: 83 3D C0 4D 14 80 cmp     dword ptr [KiTrap0E+277h],0
      00
  8013B470: 0F 85 0F FE FF FF jne     KiTrap0E+27Eh
  8013B476: B8 FF 00 00 00   mov     eax,offset KiTrap0E+283h
  8013B47B: EB AC       jmp     KiTrap0E+235h
  8013B47D: A1 52 F0 DF FF   mov     eax,[KiTrap0E+28Ah]
  8013B482: C6 05 52 F0 DF FF mov     byte ptr [KiTrap0E+290h],0
```

The ---> indicates the line in the assembly code at which the system trap occurred.

The most important information here is the stack trace at the top. This will tell you the part of the code in which the system trapped. Each line of a stack trace is a different instruction that has been pushed on the stack, with the first line being the last thing pushed on the stack. The following information is included in each line of an x86 stack trace:

ChildEBP: The base pointer. This is an address on the stack.

RetAddr: The return address. This is the address that the processor will return to when it finishes executing the current thread. This is also the address of the instruction on the next line of the stack.

Args to Child: The first three arguments passed to the function when it was called will be listed here. These are generally pointers, but can also be other values.

Function name and offset: The final piece of information is a function name and an offset into that function that identifies the location, in code, whose address was pushed on to the stack.

This next example is from a Dec Alpha system that experienced a STOP 0x0000002E:

```
Callee-SP      Arguments to Callee         Call Site
fc8e4f90 80403e08 : 80ae1060 00000000 00000000 00000000
KeBugCheckEx+0x58
fc8e5290 800c3ce8 : 80ae1060 00000000 00000000 00000000
HalMachineCheck+0x198
fc8e52d0 800c33b8 : 80ae1060 00000000 00000000 00000000
KiMachineCheck+0x28
fc8e52e0 800c1c20 : 80ae1060 00000000 00000000 00000000
KiDispatchException+0x68
fc8e55e0 800c1bcc : 80ae1060 00000000 00000000 00000000
KiExceptionDispatch+0x50
fc8e5680 80409d4c : 80ae1060 00000000 00000000 00000000
KiGeneralException+0x4
fc8e5880 f7361344 : 80ae1060 00000000 00000000 00000000
READ_REGISTER_UCHAR+0x6c
fc8e5880 f71313c4 : 80ae1060 00000000 00000000 00000000
AtalkReceiveIndication+0x654
fc8e5930 f71361a4 : 80ae1060 00000000 00000000 00000000
EthFilterDprIndicateReceive+0x234
fc8e5990 f713218c : 80ae1060 00000000 00000000 00000000
MiniportSendLoopback+0xb14
fc8e5a30 f71308d8 : 80ae1060 00000000 00000000 00000000
MiniportSyncSend+0x20c
fc8e5a70 f73628c0 : 80ae1060 00000000 00000000 00000000 NdisMSend+0x158

  800BC12C: B21DF170 stl      a0,KeBugCheckEx+80x4(gp)
800BC130: 0000001C call_pal   rdpcr
  800BC134: A0000CA0 ldl      v0,KeBugCheckEx+80x4(v0)
  800BC138: 22000060 lda      a0,KeBugCheckEx+80x5(v0)
  800BC13C: D3406778 bsr      ra,RtlCaptureContext
--->800BC140: 0000001C call_pal   rdpcr
  800BC144: A0000CA0 ldl      v0,KeBugCheckEx+t0x5(v0)
  800BC148: 22000060 lda      a0,KeBugCheckEx+t0x6(v0)
  800BC14C: D34006DC bsr      ra,KiSaveProcessorControlState
  800BC150: 0000001C call_pal   rdpcr
  800BC154: 45299801 xor      s0,76,t0
  800BC158: 221E00D0 lda      a0,KeBugCheckEx+o0x7(sp)
  800BC15C: A0000CA0 ldl      v0,KeBugCheckEx+o0x7(v0)
  800BC160: 223F0230 mov      KeBugCheckEx+E0x78,a1
  800BC164: 22400060 lda      a2,KeBugCheckEx+o0x7(v0)
  800BC168: D340803D bsr      ra,OtsMove
  800BC16C: 47EB0402 mov      s2,t1
```

In an Alpha stack trace, the Callee-SP serves the same purpose as the ChildEBP in the x86 stack. The number right after the Callee-SP is the return address, and the next four numbers are the arguments that were pushed onto the stack. These will generally be 0, because a RISC-based system uses special registers and does not pass arguments on the stack.

!process

A **!process** command without any parameters will list information on the process currently running on the active processor. The output here will look exactly like the output in the **!process 0 7** section, except that it is only for one process, and no thread information is listed.

!thread

A **!thread** command without any parameters behaves exactly as a **!process** command without any parameters, and lists the thread that is currently running. The thread output will look exactly like the output in the **!process 0 7** section.

Note There are three very similar versions of the same information so it is easier to find which thread(s) are currently executing. A **!process 0 7** will list out all process and thread information, which generally results in about 10–15 pages of data just for the process and thread output. Picking out the process or thread that is currently running from this long list can be difficult.

Dump Analysis Heuristics for Bugcode XXXXXXXX

This section is included only for the processor that actually caused the trap. This section will include information specific to the STOP code and can be very important. The exact information here will vary for different STOP codes, but this will generally list the address at which the STOP occurred and any more information that is available.

This an example from a STOP 0x0000000A:

```
*******************************************************************
** Dump Analysis Heuristics for Bugcode IRQL_NOT_LESS_OR_EQUAL
*******************************************************************
*
Invalid Address Referenced: 0x00000020
IRQL:           2
Access Type:        Write
Code Address:       0xfa6325a5
```

This example is from a STOP 0x0000001E:

```
*************************************************************
** Dump Analysis Heuristics for Bugcode KMODE_EXCEPTION_NOT_HANDLED
*************************************************************
*
Exception Code:        0xc0000005
Address of Exception:     0x801704a7
Parameter #0:         0x00000001
Parameter #1:         0x00000001
```

Common STOP Codes

Looking through the MEMORY.TXT output of common STOP codes will sometimes allow you to identify the module or driver that caused the problem. Using this information, you might be able to determine whether or not a service pack or update to Windows NT will fix the problem. In many cases, you will still need to contact support personnel, but looking at the MEMORY.TXT output gives you an idea about what is wrong.

STOP 0x0000000A IRQL_NOT_LESS_OR_EQUAL

STOP 0x0000000A indicates that a kernel mode process or driver attempted to access a memory address that it did not have permission to access. The most common cause of this error is a bad or corrupt pointer that references an incorrect location in memory. A pointer is a variable used by a program to refer to a block of memory. If the variable has a bad value in it, then the program tried to access memory that it should not be using.

When this occurs in a user mode application, it generates an access violation.

When it occurs in kernel mode, it generates a STOP 0x0000000A message. This trap can be caused by either hardware or software, and you will generally want to contact support personnel to determine the exact cause.

To determine the general cause of a STOP 0x0000000A, look at the "Stack Trace for Processor X" section of the MEMORY.TXT file. If you have a multiprocessor system, you will need to check the output for all processors and look for a stack trace that has a line similar to the following at the top of the stack:

```
ChildEBP RetAddr Args to Child
f88b6e00 f89805b0 fb55ea88 fb55e988 fb55ea88 KiTrap0E+0x252 (FPO:
[0,0,0])
```

This is the processor on which the trap occurred. After the stack trace section, you will find additional information on the trap in the "Dump Analysis Heuristics" section. To determine the module that caused the trap, look at the line on the stack trace occurring immediately after the line in the preceding example. This line will generally be the line of code that caused the trap. From this information, you can identify the module in which the trap occurred. For instance, if the top lines of the stack trace read:

```
ChildEBP RetAddr Args to Child
fa679758 fa6325a5 fcdb0b58 fccd3770 02611e6c KiTrap0E+0x252
fa6797e0 fa63ae8e fcc37528 fa67992e fccd3770 FindNameOrQuery+0x141
fa679838 fa6444a5 fa679854 fa6a33d0 fa6798d0 NbtConnect+0x3ae
fa679860 fa630393 fccd3770 fcdb2e08 fa679900 NTConnect+0x2b
```

The first line contains the reference to KiTrap0E and the second line contains FindNameOrQuery+0x141, which means the processor trapped in the function FindNameOrQuery.

STOP 0x0000001E KMODE_EXCEPTION_NOT_HANDLED

A STOP 0x0000001E can also be caused by both hardware and software. It is caused by hardware more often then a STOP 0x0000000A, but can still be caused by software.

When looking at **dumpexam** output from a STOP 0x0000001E, you will actually see two stack trace listings for the processor on which the STOP occurred. The first listing is the stack after the trap occurred, which only shows the kernel calls made to handle the trap and does not include any information about what code caused the trap.

The second listing shows the stack just before the trap occurred and is the one you actually want to use for your analysis. The register dump for the processor will also be duplicated, with the first dump showing the status of the registers after the trap and the second showing the state of the registers when the trap occurred. These two sets of information will be separated by a section that looks like the following:

```
*******************************************************************
** !exr fca49c20
*******************************************************************
*
Exception Record @ FCA49C20:

  ExceptionCode: c0000005
 ExceptionFlags: 00000000
 Chained Record: 00000000
ExceptionAddress: 801704a7
```

```
NumberParameters: 00000002

  Parameter[0]: 00000001
  Parameter[1]: 00000001
```

This section includes the following information:

ExceptionCode: This is a status code that identifies what type of exception occurred. In this case, the code is c0000005, which indicates an access violation. To find out what a particular status code means, contact support personnel.

Exception address: The address of the instruction that caused the STOP.

The first stack trace from a STOP 0x0000001E, the one that does not provide any useful information, will generally look like the following:

```
ChildEBP RetAddr Args to Child
fca49968 8013387e fca49990 801367ab fca49998
PspUnhandledExceptionInSystemThread+0x18 (FPO: [0,0,0])
fca49970 801367ab fca49998 00000000 fca49998 PspSystemThreadStartup+0x4a
(FPO: [0,0,0])
fca49f7c 8013e452 fca54bae 00000001 00000000 _except_handler3+0x47
00000000 00000000 00000000 00000000 00000000 KiThreadStartup+0x16
```

To determine where the trap occurred, ignore this stack and look at the second listing, which you will find after the **!exr** entry. The first line in this listing will indicate the location in code that caused the trap. With a STOP 0x0000001E, it is also useful to compare the exception address listed in the **!exr** section to the list of device drivers in the **!drivers** section of the MEMORY.TXT file. If the trap was caused by a specific driver, this address will fall in the address range in the drivers list. If this is the case, it can indicate either a problem with the device that the driver controls or with the driver itself. Here is an example:

```
FramePtr RetAddr  Param1  Param2  Param3  Function Name
fa1bcda4 8010e244 fcff3940 00000000 00000220 NT!PsReturnPoolQuota+0xe
fa1bcdd4 80117085 fcbee668 fcddf648 fcbff020 NT!ExFreePool+0x16c
fa1bce24 8011c60b fcddf648 fa1bce58 fa1bce54 NT!IopCompleteRequest+0xbd
fa1bce5c 8013de15 00000000 00000000 00000000 NT!KiDeliverApc+0x83
fa1bce7c 8011a1ce 00000000 00000000 80179a01 NT!@KiSwapThread@0+0x15d
fa1bcea0 80179b3f fcc4bf60 00000006 80179a01 NT!KeWaitForSingleObject+0x1c2
fa1bcef0 80139b09 00000114 00000001 00000000 NT!NtWaitForSingleObject+0xaf
fa1bcef0 77f893eb 00000114 00000001 00000000 NT!KiSystemService+0xa9
00000000 00000000 00000000 00000000 00000000 NTDLL!ZwWaitForSingleObject+0xb
```

STOP 0x0000007F UNEXPECTED_KERNEL_MODE_TRAP

A STOP 0x0000007F generally occurs in the processor itself and almost always indicates a hardware fault. There are several different kinds of STOP 0x0000007Fs, which you can determine by the first parameter of the STOP code, found in the "Windows NT Crash Dump Analysis" section at the beginning of the MEMORY.TXT file.

The following are common kernel mode traps:

First Parameter	Meaning
0x00000000	Divide by Zero Error
0x00000004	Arithmatic overflow
0x00000006	Invalid Opcode
0x00000008	Double Fault

Divide by Zero Error: A divide by zero is caused when a DIV instruction is executed and the divisor is 0. This can be caused by memory corruption, hardware problems, or software failures, which need to be investigated farther.

Here's an example of a divide by zero error:

```
ChildEBP RetAddr Args to Child
8019d778 8013cdcc fe483688 00000000 00000000 NT!_KiSystemFatalException+0xe
 (FPO: [0,0] TrapFrame @ 8019d778)
8019d7e8 fbb053be 0001440d 000004a9 000004a9 NT!_RtlEnlargedUnsignedDivide+0xc
 (FPO: [4,0,0])
8019d80c 8010f613 0001440d 000004a9 fe482bd0 bhnt!_BhStationQueryTimeout+0x44
 (FPO: [4,0,1])
8019d820 fb910aa6 fe50a000 fe44255a fe44254c NT!_KeSetTimer+0x8f
8019d85c fb9409b3 fe4820c8 fe44255a fe44254c
 NDIS!_EthFilterDprIndicateReceive+0x111
8019d894 fb94044a fe482b98 fe483688 ffdff401 netflx!NetFlexProcessEthRcv+0x85
8019d8ac fb910ba1 fe482aa8 fb910b30 00000001
 netflx!_NetFlexHandleInterrupt+0x4a
8019d8c4 80137c06 fe482bac fe482b98 00000000 NDIS!_NdisMDpc+0x71 (FPO: [EBP
 0xfb910b30] [4,0,4])
fb910b30 18247c8b 8b34778b 4e8d106f d015ff30 NT!_KiIdleLoop+0x5a
kd> !trap 8019d778
eax=0001440d ebx=00000003 ecx=8019d81c edx=000004a9 esi=fe4820c8 edi=fe46a188
eip=8013cdcc esp=8019d7ec ebp=8019d820 iopl=0     nv up ei pl zr na po nc
cs=0008 ss=0010 ds=0023 es=0023 fs=0030 gs=0000       efl=00010246
ErrCode = 00000000
8013cdcc f774240c     div   dword ptr [esp+0xc]
```

Arithmetic Overflow: An overflow occurs when the result of a multiplication operation is larger then a 32 bit integer. This error could be caused by a software failure, but is also frequently a hardware problem.

Invalid Opcode: An invalid opcode occurs when the processor attempts to execute an instruction that is not defined. This error is almost always caused by hardware memory corruption. If you are receiving this error, you should run memory diagnostics on your regular memory and both L1 and L2 cache memory.

Double Fault: This trap occurs when two kernel mode traps occur simultaneously and the processor is unable to handle them. This trap is almost always caused by some kind of hardware failure.

If a particular trap can be caused by either software or hardware, more analysis is required to determine which is the cause. If you suspect a hardware problem, try the following hardware troubleshooting steps:

1. Run diagnostic software and test the RAM in the computer. Replace any RAM reported to be bad. Also, make sure that all the RAM in the computer is the same speed.
2. Try removing or swapping out controllers, cards, or other peripherals.
3. Try a different motherboard on the computer.

LAN Manager MIB II for Windows NT Objects

The LAN Manager MIB II for Windows NT contains a set of objects specifically designed to support computers running Windows NT. Notice that, due to differences in the operating systems, there are fewer objects in the LAN Manager MIB II for Windows NT than the LAN Manager MIB II.

All LAN Manager MIB II objects apply to computers running Windows NT Workstation and Windows NT Server.

Common Group

The object name and object identifier for this group is:
iso.org.dod.internet.private.enterprise.lanmanager.lanmgr-2.common
(1.3.6.1.4.1.77.1.1)

comVersionMaj {common 1}
The major release version number of the Windows NT software.
SYNTAX OCTET STRING
ACCESS read-only

comVersionMin {common 2}
The minor release version number of the Windows NT software.
SYNTAX OCTET STRING
ACCESS read-only

comType {common 3}
The type of Windows NT software this system is running.
SYNTAX OCTET STRING
ACCESS read-only

comStatStart {common 4}
The time at which the Windows NT statistics on this node were last cleared. The time is the number of seconds since January 1, 1970.
SYNTAX INTEGER
ACCESS read-only

The **comStatStart** object applies to the following statistical objects:

comStatNumNetIOs	**svStatErrorOuts**	**wkstaStatSessStarts**
comStatFiNetIOs	**svStatPwErrors**	**wkstaStatSessFails**
comStatFcNetIOs	**svStatPermErrors**	**wkstaStatUses**
svStatOpens	**svStatSysErrors**	**wkstaStatUseFails**
svStatDevOpens	**svStatSentBytes**	**wkstaStatUseFails**
svStatQueuedJobs	**svStatRcvdBytes**	**wkstaStatAutoRecs**
svStatSOpens	**svStatAvResponse**	

comStatNumNetIOs {common 5}
The number of network I/O operations submitted on this node.
SYNTAX Counter
ACCESS read-only

comStatFiNetIOs {common 6}
The number of network I/O operations on this node that failed issue.
SYNTAX Counter
ACCESS read-only

comStatFcNetIOs {common 7}
The number of network I/O operations on this node that failed completion.
SYNTAX Counter
ACCESS read-only

Server Group

The object name and object identifier for this group is:
iso.org.dod.internet.private.enterprise.lanmanager.lanmgr-2.server
(1.3.6.1.4.1.77.1.2)

svDescription {server 1}
A comment describing the server.
SYNTAX DisplayString (size (0..255))
ACCESS read-write

svSvcNumber {server 2}
The number of network services installed on the server.
SYNTAX INTEGER
ACCESS read-only

svSvcTable {server 3}
A list of service entries describing the network service installed on the server.
SYNTAX SEQUENCE OF svSvcEntry
ACCESS not-accessible

svSvcEntry {svSvcTable 1}
The names of the network services installed on the server.
SYNTAX svSvcEntry
ACCESS read-only

svSvcName {svSvcEntry 1}
The name of a Windows NT network service.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

svSvcInstalledState {svSvcEntry 2}
The installation status of a network service.
SYNTAX INTEGER {
uninstalled(1),
install-pending(2),
uninstall-pending(3),
installed(4)
}
ACCESS read-only

svSvcOperatingState {svSvcEntry 3}
The operating status of a network service.
SYNTAX INTEGER {
active(1),
continue-pending(2),
pause-pending(3),
paused(4)
}
ACCESS read-only

svSvcCanBeUninstalled {svSvcEntry 4}
Indicates whether the network service specified by this entry can be removed.
SYNTAX INTEGER {
cannot-be-uninstalled(1),
can-be-uninstalled(2)
}
ACCESS read-only

svSvcCanBePaused {svSvcEntry 5}
Indicates whether the network service specified by this entry can be paused.
SYNTAX INTEGER {
    cannot-be-paused(1),
    can-be-paused(2)
    }
ACCESS read-only

svStatOpens {server 4}
The total number of files that have been opened on the server.
SYNTAX Counter
ACCESS read-only

svStatDevOpens {server 5}
The total number of communication devices that have been opened on the server.
SYNTAX Counter
ACCESS read-only

svStatQueuedJobs {server 6}
The total number of print jobs that have been spooled on the server.
SYNTAX Counter
ACCESS read-only

svStatSOpens {server 7}
The number of sessions that have been started on the server.
SYNTAX Counter
ACCESS read-only

svStatErrorOuts {server 8}
The number of sessions disconnected because of an error on the server.
SYNTAX Counter
ACCESS read-only

svStatPwErrors {server 9}
The number of password violations encountered on the server.
SYNTAX Counter
ACCESS read-only

svStatPermErrors {server 10}
The number of access-permission violations encountered on the server.
SYNTAX Counter
ACCESS read-only

svStatSysErrors {server 11}
The number of system errors encountered on the server.
SYNTAX Counter
ACCESS read-only

svStatSentBytes {server 12}
The number of bytes sent by the server.
SYNTAX Counter
ACCESS read-only

svStatRcvdBytes {server 13}
The number of bytes received by the server.
SYNTAX Counter
ACCESS read-only

svStatAvResponse {server 14}
The mean number of milliseconds it took the server to process a workstation I/O request (for example, the average time an NCB sat at the server).
SYNTAX INTEGER
ACCESS read-only

svSecurityMode {server 15}
The type of security running on the server.
SYNTAX INTEGER{
share-level(1),
user-level(2)
}
ACCESS read-only

svUsers {server 16}
The number of concurrent users the server can support.
SYNTAX INTEGER
ACCESS read-only

svStatReqBufsNeeded {server 17}
The number of times the server has needed a request buffer in the process of handling a client request and could not allocate one.
SYNTAX Counter
ACCESS read-only

svStatBigBufsNeeded {server 18}
The number of times the server needed, but could not allocate, a big buffer while processing a client request.
SYNTAX Counter
ACCESS read-only

svSessionNumber {server 19}
The number of sessions on the server.
SYNTAX INTEGER
ACCESS read-only

svSessionTable {server 20}
A list of session entries corresponding to the current sessions that clients have with the server.
SYNTAX SEQUENCE OF svSessionEntry
ACCESS read-only

svSessionEntry {svSessionTable 1}
A session that is currently established on the server.
SYNTAX svSessionEntry
ACCESS read-only

svSesClientName {svSessionEntry 1}
The name of the remote computer that established the session.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

svSesUserName {svSessionEntry 2}
The name of the user account that established the session on the remote computer.
SYNTAX DisplayString (size (1..20))
ACCESS read-only

svSesNumConns {svSessionEntry 3}
The number of connections to server resources that are active in the current session.
SYNTAX INTEGER
ACCESS read-only

svSesNumOpens {svSessionEntry 4}
The number of files, devices, and pipes that are open in the current session.
SYNTAX INTEGER
ACCESS read-only

svSesTime {svSessionEntry 5}
The length of time, in seconds, since the current session began.
SYNTAX Counter
ACCESS read-only

svSesIdleTime {svSessionEntry 6}
The length of time, in seconds, that the session has been idle.
SYNTAX Counter
ACCESS read-only

svSesClientType {svSessionEntry 7}
The type of client that established the session.
SYNTAX INTEGER {
down-level(1), old clients (such as PC-LAN or XENIX)
dos-lm(2), LAN Manager 2.0 for MS-DOS basic clients
dos-lm-2(3), LAN Manager 2.0 for MS-DOS enhanced clients
os2-lm-1(4), LAN Manager 1.0 for OS/2 clients, or LAN Manager 2.0 for OS/2 with Microsoft OS/2 1.1
os2-lm-2(5), LAN Manager 2.0 for OS/2 clients
dos-lm-2-1(6),
os2-lm-2-1(7),
afp-1-1(8),
afp-2-0(9),
NT-3-1(10)
}
ACCESS read-only

svSesState {svSessionEntry 8}
Indicates the state of this session. Currently, the state can only be active. When secure SNMP is available, the deleted state can be used in set requests to delete a session.
SYNTAX INTEGER{
active(1),
deleted(2)
}
ACCESS read-write

svAutoDisconnects {server 21}
The number of sessions that the server automatically disconnected because of inactivity.
SYNTAX INTEGER
ACCESS read-only

svDisConTime {server 22}
The number of seconds the server waits before disconnecting an idle session.
SYNTAX INTEGER
ACCESS read-write

svAuditLogSize {server 23}
The maximum size, in kilobytes, of the server's audit log.
SYNTAX INTEGER
ACCESS read-write

svUserNumber {server 24}
The number of users who have accounts on the server.
SYNTAX INTEGER
ACCESS read-only

svUserTable {server 25}
A table of active user accounts on the server.
SYNTAX SEQUENCE OF svUserEntry
ACCESS not-accessible

svUserEntry {svUserTable 1}
A user account on the server.
SYNTAX svUserEntry
ACCESS not-accessible

svUserName {svUserEntry 1}
The name of a user account.
SYNTAX DisplayString (size (1..20))
ACCESS read-only

svShareNumber {server 26}
The number of shared resources on the server.
SYNTAX INTEGER
ACCESS read-only

svShareTable {server 27}
A table of the shared resources on the server.
SYNTAX SEQUENCE OF svShareEntry
ACCESS not-accessible

svShareEntry {svShareTable 1}
A table corresponding to a single shared resource on the server.
SYNTAX svShareEntry
ACCESS not-accessible

svShareName {svShareEntry 1}
The name of a shared resource.
SYNTAX DisplayString (size (1..12))
ACCESS read-only

svSharePath {svShareEntry 2}
The local name of a shared resource.
SYNTAX DisplayString (size (1..255))
ACCESS read-only

svShareComment {svShareEntry 3}
A comment associated with a shared resource.
SYNTAX DisplayString (size (0..255))
ACCESS read-only

svPrintQNumber {server 28}
The number of print queues on the server.
SYNTAX INTEGER
ACCESS read-only

svPrintQTable {server 29}
A table of the print queues on the server.
SYNTAX SEQUENCE OF svPrintQEntry
ACCESS not-accessible

svPrintQEntry {svPrintQTable 1}
A table entry corresponding to a single print queue on the server.
SYNTAX svPrintQEntry
ACCESS not-accessible

svPrintQName {svPrintQEntry 1}
The name of a print queue.
SYNTAX DisplayString (size (1..12))
ACCESS read-only

svPrintQNumJobs {svPrintQEntry 2}
The number of jobs currently in a print queue.
SYNTAX INTEGER
ACCESS read-only

Workstation Group

The object name and object identifier for this group is:
iso.org.dod.internet.private.enterprise.lanmanager.lanmgr-2.workstation
(1.3.6.1.4.1.77.1.3)

wkstaStatSessStarts {workstation 1}
The number of sessions the workstation initiated.
SYNTAX Counter
ACCESS read-only

wkstaStatSessFails {workstation 2}
The number of failed sessions the workstation had.
SYNTAX Counter
ACCESS read-only

wkstaStatUses {workstation 3}
The number of connections the workstation initiated.
SYNTAX Counter
ACCESS read-only

wkstaStatUseFails {workstation 4}
The number of failed connections the workstation had.
SYNTAX Counter
ACCESS read-only

wkstaStatAutoRecs {workstation 5}
The number of sessions that were broken and then automatically re-established.
SYNTAX Counter
ACCESS read-only

wkstaErrorLogSize {workstation 6}
The maximum size, in kilobytes, of the workstation error log.
SYNTAX INTEGER
ACCESS read-write

wkstaUseNumber {workstation 7}
The number of entries in wkstaUseTable (active uses the workstation is currently maintaining).
SYNTAX INTEGER
ACCESS read-only

wkstaUseTable {workstation 8}
The table of active uses made by this workstation.
SYNTAX SEQUENCE OF wkstaUseEntry
ACCESS not-accessible

wkstaUseEntry {wkstaUseTable 1}
A use of a remote network resource.
SYNTAX wkstaUseEntry
ACCESS not-accessible

useLocalName {wkstaUseEntry 1}
The name of the local devicename (for example, e: or lpt1:) that is redirected.
SYNTAX DisplayString (size (0..8))
ACCESS read-only

useRemote {wkstaUseEntry 2}
The name of the remote shared resource to which the redirection has been made (for example, \\server\share).
SYNTAX DisplayString (size (1..255))
ACCESS read-only

useStatus {wkstaUseEntry 3}
The status of this connection.
SYNTAX INTEGER {
use-ok(1),
use-paused(2),
use-session-lost(3),
use-network-error(4),
use-connecting(5),
use-reconnecting(6)
}
ACCESS read-only

Domain Group

The object name and object identifier for this group is:
iso.org.dod.internet.private.enterprise.lanmanager.lanmgr-2.domain
(1.3.6.1.4.1.77.1.4)

domPrimaryDomain {domain 1}
The name of the primary domain to which the computer belongs.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

domLogonDomain {domain 2}
The name of the domain to which this machine is logged on.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

domOtherDomainNumber {domain 3}
The number of entries in domOtherDomainTable.
SYNTAX INTEGER
ACCESS read-only

domOtherDomainTable {domain 4}
The list of other domains that this machine is monitoring.
SYNTAX SEQUENCE OF domOtherDomainEntry
ACCESS not-accessible

domOtherDomainEntry {domOtherDomainTable 1}
An entry in the table of other domains.
SYNTAX domOtherDomainEntry
ACCESS not-accessible

domOtherDomainName {domOtherDomainEntry 1}
The name of an additional domain that this machine is monitoring.
SYNTAX DisplayString (size (1..15))
ACCESS read-write

domServerNumber {domain 5}
The number of entries in domServerTable.
SYNTAX INTEGER
ACCESS read-only

domServerTable {domain 6}
The list of nonhidden servers that are on all of the domains this machine is monitoring.
SYNTAX SEQUENCE OF domServerEntry
ACCESS not-accessible

domServerEntry {domServerTable 1}
An entry in the domain server table.
SYNTAX domServerEntry
ACCESS not-accessible

domServerName {domServerEntry 1}
The name of a server on one of the domains that this machine is monitoring.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

domLogonNumber {domain 7}
The number of entries in domLogonTable.
SYNTAX INTEGER
ACCESS read-only

domLogonTable {domain 8}
The list of domain logons that this machine has processed. Available only on servers acting as primary or backup domain controllers.
SYNTAX SEQUENCE OF domLogonEntry
ACCESS not-accessible

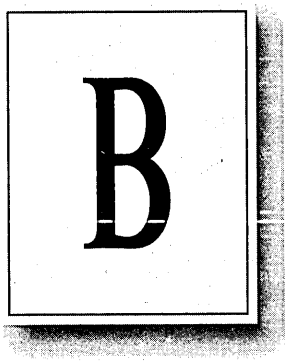
domLogonEntry {domLogonTable 1}
An entry in the logon table.
SYNTAX domLogonEntry
ACCESS not-accessible

domLogonUser {domLogonEntry 1}
The name of the user who is logged on to this domain.
SYNTAX Display String (size (1..20))
ACCESS read-only

domLogonMachine {domLogonEntry 2}
The name of the machine from which the user logged on.
SYNTAX DisplayString (size (1..15))
ACCESS read-only

APPENDIX B

Minor Revisions to Existing Resource Kit Books



The *Microsoft Windows NT Resource Kit* for Windows NT Workstation and Windows NT Server version 3.51 contains slightly updated editions of the version 3.5 set of four volumes. For those customers who already have the version 3.5 set and will only receive the *Windows NT Update 1* and *Windows NT Update 2* books, we have included appendixes in both update books with lists of the changes that were made when we reprinted the books for this current version.

Resource Guide

The following changes were made in the *Windows NT Resource Guide*:

- Chapter 3, "Customizing Windows NT Setup"
  - page 97, first paragraph under the section "Using the TXTSETUP.SIF File to Update the Registry." Add the following to the beginning of the first sentence, "When upgrading, you..."
  - page 97, fourth paragraph. Delete the words "in order."
  - page 103, second paragraph under the section "Uploading the Master System to the Distribution Server." Change the filename "PROFILE.TXT" to "CPS.HLP."
- Chapter 4, "Windows NT Files"
  - page 126, Table 4.4. Delete the following filename and description: NETBIOS.DLL Network DDE
  - page 145, Table 4.9. Delete the superscript "2" after the filename _DEFAULT.PIF.
  - page 146, Table 4.9. Delete the superscript "2" after the filename OSO001.009.

Networking Guide

The following changes were made in the *Windows NT Networking Guide*:

- Chapter 12, "Networking Concepts for TCP/IP"
  - page 189, first paragraph. Delete the last sentence:
    For additional information about these topics, see the books listed in the "Welcome" section of this manual.
  - page 204, third sentence of the first paragraph under the section "M-Node." Change "successful" to "unsuccessful."
  - page 206, first sentence of the last paragraph under the section "WINS in a Routed Environment." Change "a router" to "routers."
- Chapter 13, "Installing and Configuring DHCP Servers"
  - page 243, third paragraph under the section "Managing Client Leases." Change "tell you want you want" to "tell you what you want."
  - page 254, the Registry key example. Delete "\current." The new key example should then read as follows:

    ```
    ...SYSTEM\currentcontrolset\services\DHCPServer\Parameters
    ```

  - page 255, under **DatabaseCleanupInterval**. The default value should be as follows: 0x15180 (86,400 minutes -- 24 hours)
  - page 256, first paragraph after the Registry key example. Change the last word "defineC" to "defined."
- Chapter 14, "Installing and Configuring WINS Servers"
  - page 276, Table 14.3, the description for the Backup On Termination option. Change "WINS Manager" to "the WINS service."
  - page 278, first paragraph. Delete the words "in order."
  - page 278, the two-item bulleted list. Change both instances of "ServerB" to "Server3," and delete the word "WINS."
  - page 281, under the procedure, **"To define push partner properties."** Change the minimum value for Update Count from "5" to "20." The example in the screen shot should also be changed.
  - page 288, fourth paragraph under the section "Multihomed Names." Change both instances of "Netbt" to "NetBT."
- Chapter 16, "Using the Microsoft FTP Server Service"
  - page 323, first bulleted item under the section "Configuring the FTP Server Service." Change "FTP connection" to "FTP connections," and "Allow Anonymous Connection" to "Allow Anonymous Connections."
  - page 323, second bulleted item under the same section. Change "Allow Anonymous Connection" to "Allow Anonymous Connections."

Windows NT Update 1

The following changes were made in *Windows NT Update 1*:

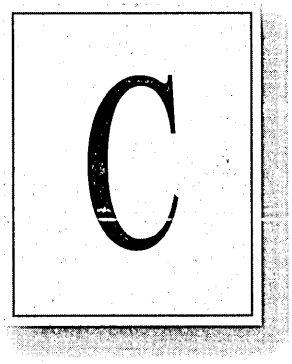
- Chapter 1, "Setup"
  - page 6, last paragraph. Change "NTDETEC.COM" to "NTDECT.COM."
  - page 9, last sentence under the section "Auto-Joining a Domain." Change "use User Manager on a domain controller" to "use User Manager for Domains on a primary domain controller."
- Chapter 4, "Internet Services and Security"
  - page 49, first and third paragraphs. Change the title "*Windows NT 3.51 Resource Kit*" to "*Windows NT Resource Kit* version 3.51."
  - page 50, third paragraph under the section "Mail Server Service." Change "via User Manager" to "via User Manager for Domains."
  - page 50, fourth paragraph under the same section. Change "the Windows NT server" to "the Windows NT Server computer."
- Chapter 6, "Troubleshooting"
  - page 86, first paragraph under the section "Event Viewer Log File Information." Change "%SYSTEMROOT%" to "*\systemroot*."
  - page 86, first paragraph after the first bulleted list under the same section. Change the second sentence to read as follows:
    "These APIs are documented in the Microsoft Development Library on the MSDN CD."
  - page 90, after the first paragraph under the section "Altering Startup Boot Menu." Change the rest of the section to read as follows:
    "On an x86-based computer, modify c:\boot.ini. You have to unhide the file first by typing

    c:\attrib -r -h -s boot.ini

    On RISC-based computers, choose Setup from the ROM menu."
  - page 90, last sentence. Change "help" to "Help."
  - page 91, first paragraph under the section "Problems in WINDIFF.EXE." Change the title "*Windows NT 3.5 Resource Kit*" to "*Windows NT Resource Kit* version 3.5."
  - page 91, last paragraph. Change "to manually change this parameter" to "to change this parameter manually." Also, delete the words "so you should upgrade to it" and replace them with " which might make it useful for you to upgrade to it."

APPENDIX C

RAS Reference



This appendix is split into three sections. The first is an overview of the most important modem compatibility standards and how they work within RAS. The second section is a series of quick-reference charts to give you a high-level perspective of how Remote Accesss Service (RAS) works during a call to a Windows NT RAS server. The last section containsn reference tables for RAS server and client computers that detail the different versions of RAS and the features they support.

RAS and Modem Compatibility Standards

The following information is an overview of the most important modem standards, how they can be categorized into four compatibility levels, and how they pertain to RAS and RAS data compression. This information, along with the RAS online Help topic "Modifying MODEM.INF," also enables you to implement unsupported modems with RAS versions 1.x, RAS for Windows for Workgroups 3.11, and RAS for Windows NT versions 3.1, 3.5, and 3.51.

You can apply most of the information to RAS for Windows 95; however, RAS for Windows 95 does not use a MODEM.INF file. Instead, it uses the Telephone Application Programming Interface (TAPI) that relies on entries in the Windows 95 registry for modem initialization. To implement unsupported modems with RAS for Windows 95, it is recommended that you obtain a modem driver from the modem manufacturer.

Supported Media

Depending on the RAS version, RAS supports X.25, ISDN, Null Modem, and asychronous modem communication. RAS does not support synchronous communication and therefore cannot communicate with a synchronous serial port. Synchronous communciation synchronizes the transmission by controlling the timing and the duration of data signals.

Asychronous communication means that each byte is framed with a start and stop bit. RAS sends data asynchronously to the serial port, and the serial port sends data asychronously to the modem. The asynchronous modem then strips the start and stop bit from each byte (a byte is also referred to as character) and converts the characters into blocks that are then sent synchronously (not asychronously) to the other modem by using an error control protocol. The other modem disassembles these synchronous blocks, and frames each character with a start and stop bit before sending it on to the serial port of the RAS server. The serial port then sends it to the RAS server service. This process occurs during a communication call similiar to figure C.1.

Leased line communication using asychronous serial ports and either asychronous or synchronous modems can be implemented under some circumstances, as long as RAS can treat the connection as an asynchronous null modem connection. Some modems support both synchronous and asynchronous communication.

Asynchronous Communication

To understand asynchronous modem communication using RAS, you should understand the following four compatibility levels in relation to RAS:

- Modem Command Language
- Modem Modulation Standards
- Modem Error Control Standards
- Modem Data Compression Standards and RAS Data Compression

Figure C.1 shows these four compatibility levels being implemented during a WAN connection. The different levels are shown below the section of the asynchronous RAS link to which they apply: modem command language affects command compatibility between RAS and the local modem, modem modulatation standards affect the telephone line speed compatibility between two modems, modem error control standards affect the local and remote modem level of error control compatibility, modem compression standards affect the local and remote modem data compression compatibility, and RAS data compression affects data compression between the local and remote computer. In the diagram, the RAS client is on the right side and the RAS server on the left side, and they are connected over external modems.

Notice that a RAS client or server acts as Data Terminal Equipment (DTE), and a modem acts as Data Communications Equipment (DCE).

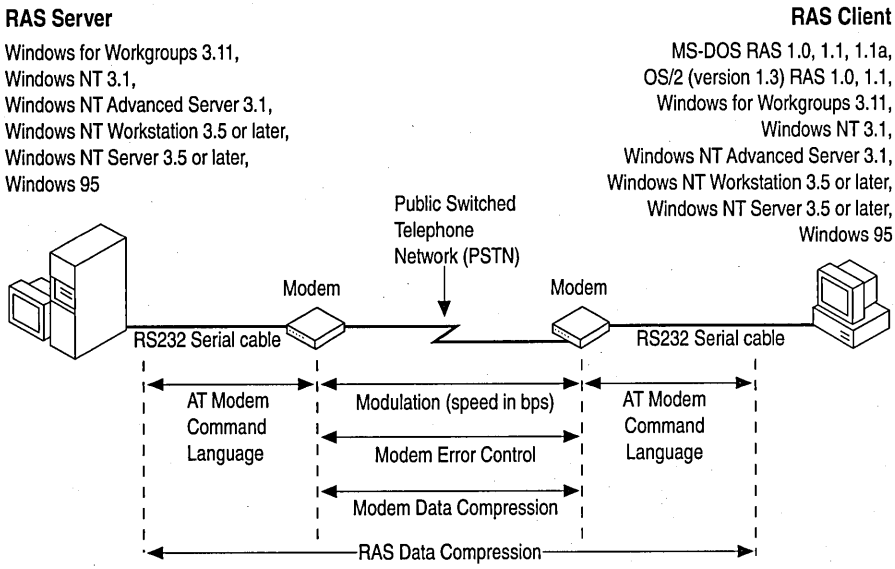


Figure C.1 Modem Compatibility Standards

RAS and the Modem Command Language

For a modem to be compatible with any communication software running on the local computer, the software must be compatible with the *modem command language*, also referred to as the *modem command set*. The command set configures the other three compatibility levels of the modem according to the needs of the local software (RAS), the supported modem standards of the modem being called, and the users' preferences. For certain modems, the modem command set works between the local and remote modems as well as between the local communications software and the local modem.

Command set compatibility between RAS modem drivers and the local modem is advantageous, but not a requirement. Many manufacturers advertise that their modem is 100% compatible with the popular Hayes AT command set, or that it is compatible with another popular modem's command set. This generally means that the modem uses the same commands as Hayes modems for basic operations such as dial, hang up, reset, and answer. It does not imply that the modem uses the same commands for configuration of modulation, error control, or data compression. However, if your modem is not compatible with any supported modem's command set, you can customize the RAS modem driver by modifying the MODEM.INF file (or MODEMS.INF file in RAS versions 1.x) to make it compatible with virtually any modem's command set. For additional information on modifying the MODEM.INF file, see the RAS Help topic "Modifying MODEM.INF."

Command set compatibility between the local modem and the remote modem is not a requirement for RAS. Command set compatibility between the local and remote modem is only important if both modems are designed to be configurable by a command received from the other modem. For example, an administrator may call from home and issue a command to change the default settings during the next power up of the modem in the office; however, this specialized feature is not important during RAS communication.

RAS and Modem Modulation Standards

The modem modulatation standards affect the telephone line speed compatibility between two modems, as shown in figure C.1. For two modems to communicate, their modulation standards must be compatible.

The modulation standard only defines the speed or permitted speed range between the modems. The speed between the computer and the modem can be different and depends on the serial hardware, the microprocessor speed in the computer, and whether or not hardware flow control or XON/XOFF software flow control is enabled.

Table C.1 shows the most popular modulation schemes in the left column and their corresponding speed range in bits per second (bps) in the right column.

Table C.1 Modulation Schemes and Modem Speeds

Popular Modulation Schemes	Modem to Modem speed
V.22 (ITU-T (formerly CCITT) Standard)	1200 bps
V.22 *bis* (ITU-T (formerly CCITT) Standard)	2400 bps
V.32 (ITU-T (formerly CCITT) Standard)	4800 - 9600 bps
V.32 *bis* (ITU-T (formerly CCITT) Standard)	4800 - 14400 bps
V.fc and V.fast (Proprietary Modulation Schemes)	2400 - 28800 bps
V.34 (ITU-T (formerly CCITT) Standard)	2400 - 28800 bps

Figure C.2 shows that modulation standards take effect only on the modem-to-modem (DCE-to-DCE) link. Speeds between DTE and DCE are independent of modem modulation standards and often differ from the DCE-to-DCE speed. When speeds are different, *hardware flow control* or *software flow control* support is required so that data is not lost when transmission speed between the modems is less than transmission speed between the DTE and DCE, or when transmission between the modems is temporarily delayed due to retransmission of data that was corrupted during transmission. RAS versions 1.0 and 1.1 do not support hardward or software flow control. RAS versions 1.1a and later support hardware flow control.

Figure C.2 also displays the different modulation standards below the DCE-to-DCE link to which they apply. The column to the right of the center column displays the RAS clients' DTE-to-DCE speeds that occur in relation to the DCE-to-DCE speeds on the same row in the center column. Depending on your modem's capabilities, these speeds should be set in the corresponding RAS client versions that are displayed in the far right column. The highest supported DCE-to-DTE speeds that should be set on the RAS server (depending on its modem[s] capabilities) are displayed in the column to the left of the center column. The right most column lists the RAS client versions and the left most column lists the RAS server versions whose highest supported modulation rate corresponds to the modulation standard in the center column.

For example, in the "9600 bps, V.32" entry in the center column, the columns to the left and right show that the DTE-to-DCE speeds are also 9600 bps, and that this was the highest supported modulation mode of RAS versions 1.0 and 1.1.

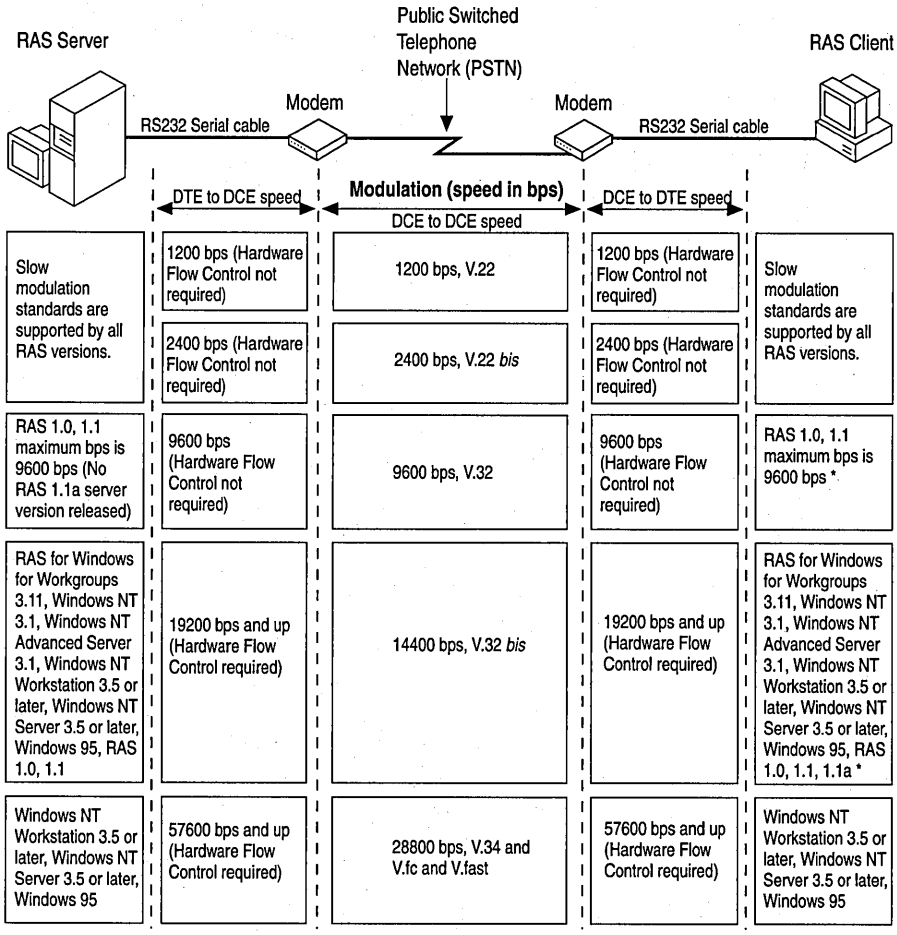


Figure C.2 Modulation Standards and RAS

*RAS 1.0 and 1.1 were tested only with US Robotics V.32 *bis* modems at 14400 bps line speed. The modems were configured for V.42, V.42 *bis*, and V.32 *bis*. The OS/2 1.3 LAN Manager RAS 1.0 or 1.1 server had an intelligent DigiBoard installed.

Modem Speed and Modulation Change During a Connection

When connecting, modems do not always negotiate their fastest built-in modulation rate. Depending on the quality of the phone line connection and the quality of the modem hardware, two modems may negotiate a certain modulation standard, but may not use the maximum speed defined in that standard. For example, two V.34 modems made by the same or different manufacturers, may negotiate a modulation rate in 2400 bps increments somewhere between 2400 bps and 28800 bps. They may agree to use 2400 bps because of noise in the telephone line.

Even though the line speed may be only 2400 bps, these V.34 modems are still using the V.34 modulation scheme. V.34 modulation also enables the modems to change the line speed dynamically during a call, in response to changes in the phone line quality. Other modulation standards may also enable the modems to connect at slower speeds, but may not enable the modems to dynamically adjust the line speed during a connection.

To change to a different modulation scheme, as opposed to just changing the speed within a modulation scheme while the modems have a connection, the modems need to support an error control protocol that supports this functionality. The non-standard MNP10™ error control protocol is an example. However, the MNP10 protocol is only needed if the modem is not a V.34 modem and only if the modem needs to downshift from 4800 bps V.32/V.32 *bis* to 2400 bps V.22 *bis* (or upshift).

RAS is usually oblivious to the dynamic speed changes. However, MNP10 modulation changes and very slow connections that may occur with cellular modems or with low quality telephone lines may cause time-out problems in the RAS network protocols. Only Windows 95 supports cellular modem connections.

Table C.2 displays the fastest possible modulation mode likely to be negotiated when modems configured with the same or different modulation modes attempt to make a connection. This table assumes that modems with different modulation configurations and capabilities are configured to enable them to negotiate up or down to the highest modulation standard the other modem supports. If both modems are not configured to negotiate to a different modulation, and their modulation settings differ, they cannot establish a connection.

For example, if a V.22 *bis* (2400bps) modem and a V.32 *bis* (14400 bps) modem try to establish a connection and the V.32 *bis* modem is not configured to negotiate down to V.22 *bis*, the modems are unable to establish a connection. If the V.22 *bis* modem is not configured to negotiate down, but the V.32 *bis* modem is configured to negotiate down, then the modems can connect at V.22 *bis* (but not at V.22).

Table C.2 Modulation Modes

Modulation Selected on Calling Modem	V.34	V.fc/V.fast	V.32 *bis*	V.32	V.22 *bis*	V.22
V.22 (1200bps)	V.22	V.22	V.22	V.22	V.22	V.22
V.22 *bis* (2400 bps)	V.22 *bis*	V.22 *bis*	V.22 *bis*	V.32 *bis*	V.22 *bis*	V.22
V.32 (9600 bps)	V.32	V.32	V.32	V.32	V.22 *bis*	V.22
V.32 *bis* (14400 bps)	V.32 *bis*	V.32 *bis*	V.32 *bis*	V.32	V.22 *bis*	V.22
V.fc / V.fast (28800 bps)	V.fc/V.fast (V.32 *bis* if V.fc/V.fast is not supported by answering modem)	V.fc/V.fast	V.32 *bis*	V.32	V.22 *bis*	V.22
V.34 (28800 bps)	V.34	V.fc/V.fast (V.32 *bis* if V.fc/V.fast is not supported by calling modem)	V.32 *bis*	V.32	V.22 *bis*	V.22

Note V.fc and V.fast are proprietary modulations that use modulation technology similar to V.34. Therefore, two V.fc or two V.fast modems from different manufacturers may not be able to connect at V.fc or V.fast respectively and may have to negotiate down to V.32 *bis*. Also, the use of the V.fc and V.fast names may be inconsistent between manufacturers, making it difficult for a buyer to determine compatibility between these modems. The V.34 standard does not include support for V.fc and V.fast. Therefore, some V.34 modems that do not support V.fc or V.fast fall back to V.32 *bis* when connecting with a V.fc or V.fast modem.

How to Make Unsupported Modems Work in Pre-Windows NT 3.5 RAS Versions

You may be able to make RAS 1.1a, RAS for Windows for Workgroups 3.11, and RAS for Windows NT version 3.1 work with an unsupported modem (for example, a V.fc,V.fast, or V.34 modem, which is unsupported in these versions). If you want to use a V.fc, V.fast, or V.34 modem, it is recommended that you use a RAS client computer that has a 486 processor or later, and a serial port that has a 16550 UART chip or later. If problems occur, configure your modem to use V.32 *bis*.

The easiest way to get an unsupported modem supported under RAS for Windows for Workgroups 3.11 or RAS for Windows NT 3.1 is to obtain a modem that is supported in RAS for Windows NT 3.5 or 3.51. The following procedure assumes that you have a modem that is supported in Windows NT 3.5 or 3.51. If your modem is not supported in Windows NT 3.5 or 3.51 or you have RAS 1.1a, skip the following procedure, but read the rest of this chapter, and then follow the instructions in the Help topic "Modifying MODEM.INF."

To make an unsupported modem work

1. From the RAS for Windows NT 3.5 or 3.51 MODEM.INF file, copy the section that applies to your modem.
2. Load MODEM.INF into a text editor and append the copied section to the RAS for Windows for Workgroups 3.11 or Windows NT 3.1 MODEM.INF file.
3. Remove the lines that start with the following words from the section you appended:

   DETECT_STRING=

   DETECT_RESPONSE=

   Note RAS modem autodetection is not supported in RAS for Windows for Workgroups 3.11 and RAS for Windows NT 3.1.

4. Save MODEM.INF, and then quit the text editor.
5. Restart your computer.
6. Depending on whether you have Windows NT 3.1 or Windows for Workgroups 3.11, continue with the corresponding section below.

To make an unsupported modem work in Windows for Workgroups 3.11

1. Restart the RAS client software, and then choose Configure from the Setup menu.
2. Select your modem from the Device field, and then choose OK.

   RAS is now ready to use your modem.

▷ **To make an unsupported modem work in Windows NT 3.1**

1. Start Control Panel, and then choose the Network icon.
2. Select Remote Access Service from the list of Installed Network Software, and then choose Configure.
3. In the Remote Access Setup dialog box, choose Configure.
4. In the Attached Device box, select the modem name that corresponds to the section you just copied to MODEM.INF, and then choose OK.

   RAS is now ready to use your modem.

RAS and Modem Error Control Standards, Modem Data Compression Standards, and RAS Data Compression

Error control and modem compression are optional features that are available with most modems made in the last few years. For reliable connections, the local and remote modem error control standards must be compatible; and for improved throughput, their modem data and RAS data compression standards must be compatible.

Table C.3 shows the error control and modem data compression standards available with most modems. If you use modem compression, use the error control standard in the cell above the compression value. For instance, if you want to enable error control and modem compression, MNP4 is used with MNP5 or V.42 is used with V.42 *bis*. MNP4 error control cannot be combined with V.42 *bis* compression, nor can V.42 error control be combined with MNP5 compression. MNP4 and MNP5 are older standards and less efficient than V.42 and V.42 *bis*.

Table C.3 Modem Error Control and Compression Protocols

Modem Error Control Protocol	**MNP4™**	**V.42 w/ LAPM**	**MNP10™ for cellular and land modem connections**
Modem Compression Protocol	**MNP5™ (up to 2:1 compression)**	**V.42 *bis* (up to 4:1 compression)**	**V**.42 *bis* or MNP5

(MNP = Microcom Networking Protocol)

Note MNP10 is a new nonstandard reliable connection protocol not available on most modems. It was designed to overcome cellular modem signal distortion; however, it also makes land connections more reliable. It is not compatible with the Motorola MC2™ cellular protocol, nor the AT&T EC2™ cellular protocol, as of July 1995. MNP10 is combined with V.42 *bis* compression.

Figure C.3 has three horizontal sections whose titles you can find in the center column directly below the modem-to-modem link picture at the top of the figure. To read the information in this figure, decide which of the following three sections you want to focus on: Modem Error Control, Modem Data Compression, or RAS Data Compression, then find that section in the center column and read the entries to the left and right that are in the same horizontal row. This enables you to determine whether the standard or feature also applies to the modem-to-computer link, and what RAS client versions and server versions support the modem or RAS feature of this section.

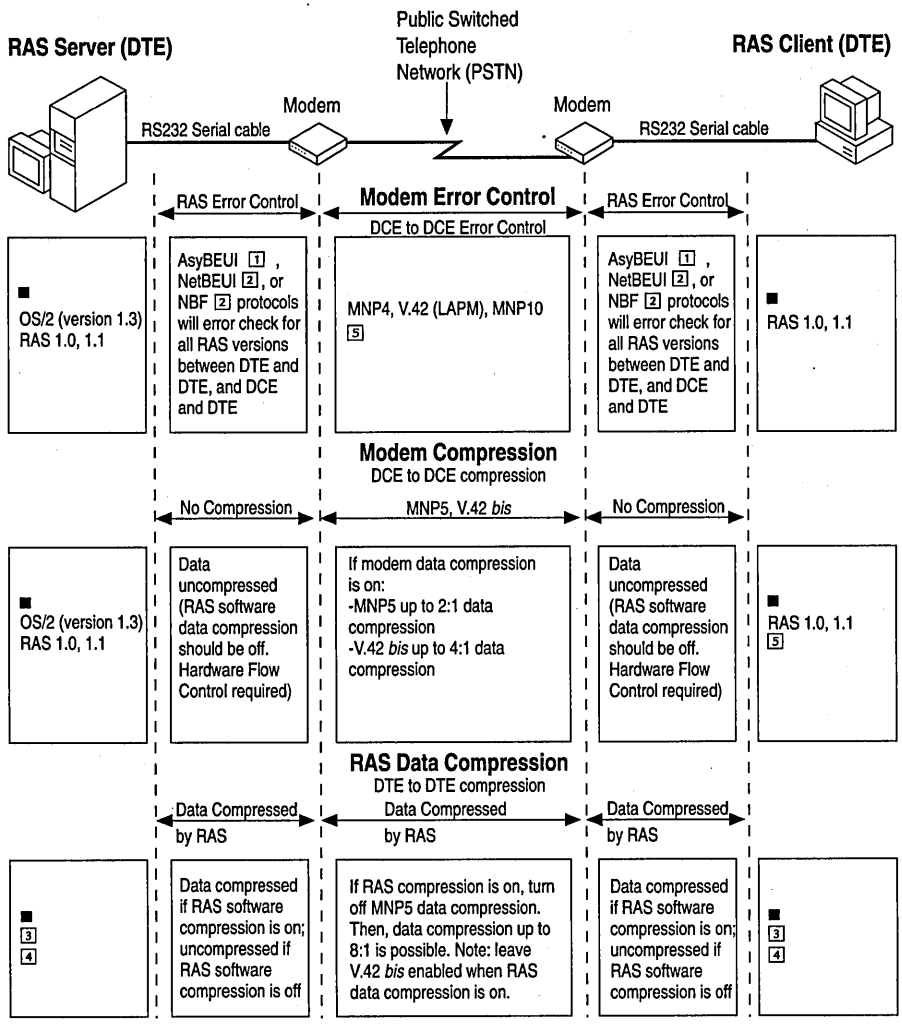


Figure C.3 Modem Error Control Standards, Modem Compression Standards, and RAS Software Data Compression

■ RAS for Windows for Workgroups 3.11, Windows NT 3.1, Windows NT Advanced Server 3.1, Windows NT Workstation 3.5 or later, Windows NT Server 3.5 or later, Windows 95.

[1] AsyBEUI applies to RAS versions 1.x only.

[2] NetBEUI applies to RAS for Windows for Workgroups 3.11, and NBF applies to Windows NT versions 3.1, 3.5, and 3.51.

[3] In case you have not disabled your RAS software compression in the Options menu of your RAS client, check the size of the RASMAC.386 files. If it has only 27193 bytes, it does not have RAS compression. If RASMAC.386 has 49209 bytes, then it has RAS compression support. The file with RAS compression support can be obtained on the Windows NT 3.5 U.S. Service Pack 2 CD in the SUPPORT\RAS directory.

[4] In case you have not disabled your RAS software compression in the Options menu of your RAS client, check the size of the ASYNCMAC.SYS file. If it has only 33732 bytes it does not have RAS compression. The releases of ASYNCMAC.SYS with 53188 bytes and 53716 bytes have RAS compression support. The files with RAS compression support can be obtained on the Windows NT 3.5 U.S. Service Pack 2 CD in the SUPPORT\RAS directory .

[5] RAS 1.0 and 1.1 were tested only with US Robotics V.32 *bis* modems at 14400 bps line speed. The modems were configured for V.42, V.42 *bis*, and V.32 *bis*. The OS/2 1.3 LAN Manager RAS 1.0 or 1.1 server had an intelligent DigiBoard installed.

Modem Error Control

The first section of figure C.3 shows that modem error control (MNP4, V.42 [LAPM], MNP10) standards take effect only between modems (DCE-to-DCE link). For RAS versions 1.0 and 1.1, these standards are not supported; however, in RAS version 1.1a and later these standards are activated.

MNP4 is the oldest and least efficient of the three error control standards; V.42 is newer and more efficient than MNP4; and MNP10 is the newest (nonstandard) protocol, although it is not available in many modems. MNP10 is designed to enable change of modulation modes during connections and operate under extremely adverse environments where V.42 fails. Error control between computer and modem (DTE and DCE) is handled automatically by the RAS protocol by encompassing the entirety of the connection; not only the connection between DTE and DCE, but the connection between RAS client and RAS server (DTE-to-DTE).

Modem Data Compression

The second section of figure C.3 shows that modem data compression (MNP5, V.42 *bis*) standards take effect only between modems (DCE-to-DCE link). For RAS versions 1.0 and 1.1, these standards are not supported. However, if you are using RAS version 1.1a or later, you should activate modem compression (along with hardware flow control) if you are calling a Windows for Workgroups 3.11 or Windows NT RAS server. You should not activate modem compression if you are calling a RAS 1.0 or 1.1 OS/2 version 1.3 server with LAN Manager 2.1 or later. For more information, see the ⧈ note below figure C.3.

RAS Data Compression

The third section of figure C.3 shows that RAS data compression and decompression happen between computers (DTE-to-DTE link) without involving modem compression. The modem receives RAS-compressed data and just passes it along to the other modem. Thus, modem compression is not involved in this part of the figure. With RAS data compression, MNP5 modem compression must be turned off in order to prevent redundancy, because data that are already highly compressed by RAS might grow larger before transmission if MNP5 modem compression is enabled simultaneously.

RAS data compression can be four times as efficient as MNP5, and about twice as efficient as V.42 *bis* modem data compression, unless a previously compressed file is transmitted. V.42 *bis* can detect whether it is about to expand data that it is supposed to compress. V.42 *bis* can also automatically suspend itself until the data is once again compressible. It is advised that you leave V.42 *bis* enabled at all times, but not use MNP5 with RAS data compression.

To avoid data loss due to data buffer overflow, you must enable hardware flow control in the RAS client whenever you are using modem compression or RAS data compression. You may turn hardware flow control on in the RAS user interface of the following RAS versions: Windows for Workgroups 3.11, Windows 95, Windows NT versions 3.1, 3.5, and 3.51. RAS versions 1.0 and 1.1 do not support hardware flow control. RAS version 1.1a requires that a hardware flow control command be set in the MODEMS.INF file on the command.init line containing the modem initialization string for your modem.

Note RAS 1.1a ships with the MCOMP.INF file that issues hardware flow control and maximum baud rate commands for all modems supported by that file. To use hardware flowcontrol with a supported modem, you must save the MODEMS.INF file (which contains modem commands that disable hardware flow control) and copy the MCOMP.INF file to the name MODEMS.INF.

RAS and Unsupported Modems

If you have an unsupported modem or want to customize the MODEM.INF file and want to use compression, and you are using RAS versions later than 1.x on the RAS server and client, it is recommended that you turn on V.42 *bis* modem compression and RAS software data compression. If you want to use RAS data compression, leave the most efficient error control protocol that is available on your modem turned on. For additional information, see the RAS Help topic "Modifying MODEM.INF."

To use modem compression with an unsupported modem on the RAS 1.1a client (that does not have the RAS data compression feature) append a custom modem section to the MODEMS.INF file and create an initialization string that enables hardware flow control, V.42 *bis* compression, and V.42 error control, if available. Be sure to check that your modem is not already supported in the RAS 1.1a MCOMP.INF file that ships with RAS 1.1a. To create your custom modem section, model your section after an existing section in MCOMP.INF.

Modem Standard Combinations Supported by the Different RAS Versions

Table C.4 is a historical view of the different RAS versions and the modem standard combinations they supported when they were released. Newer RAS versions support all down-level modem compression and error control combinations. Only the most important standard combinations are shown. If there is more than one column for a RAS version, the right most column displays the highest modem standards supported by that RAS version.

The left-most column of that RAS version shows a combination of standards that may occur with less capable modems. For example, the highest modulation rate RAS 1.1a supported at the time of release is V.32 *bis*; this means it also supports V.32 and V.22 *bis* which were supported by RAS 1.1 and 1.0. Because this table is only a historical view, it does not reflect that an older RAS version can possibly support a newer modulation standard than shown in the table. For example, the table shows RAS for Windows for Workgroups 3.11 as supporting only V.32 *bis* as the highest modulation scheme. This is because V.fc, V.fast, and V.34 modulation scheme modems did not exist yet and therefore were not available for testing. However, RAS for Windows for Workgroups 3.11 may work properly with a V.34 modem in the right configuration. To make it work properly you may have to use a computer with one or more of the following components:

- 486 or later processor
- 4 MB of RAM
- An asychronous serial board that has a 16550 UART chip

Table C.4 History of RAS versions and supported modem standards

RAS Versions/ Modem Standards	RAS 1.0	RAS 1.1a*	RAS 1.1a*	RAS for Windows for Workgroups version 3.11	RAS for Windows NT versions 3.1, 3.5 and 3.51	Remote Network Access for Windows 95(RNA)
Modem Command Set	Hayes AT, etc.	Hayes AT, etc.	Hayes AT, etc.	Hayes AT, etc.	Hayes AT, etc.	Hayes AT, etc.
Modulation	V.22 *bis* (2400 bps) and V.32 (9600 bps)	V.32 (9600 bps)	V.32 *bis* (14400 bps)	V.32 *bis*	Windows NT 3.1: V.fc, V.fast (28800 bps); Windows NT 3.5 and 3.51: V.34 (28800 bps)	V.34 (28800 bps)
Error Control	Not Supported	MNP4 (may also use V.42)	V.42 with LAPM (may also use MNP4)	V.42 with LAPM (may also use MNP4)	V.42 with LAPM (may also use MNP4)	V.42 with LAPM (may also use MNP4)
Modem Compression	Not Supported	MNP5 (may also use V.42 *bis* if V.42 is enabled	V.42 *bis* (may also use MNP5 if MNP4 is enabled)	Use RAS datacompression along with V.42 *bis* [1] (may also use MNP5 with MNP4 enabled and RAS data compression disabled)	Use RAS data compression along with V.42 *bis* [1] (may also use MNP5 with MNP4 enabled and RAS data compression disabled)	RAS data compression is automatically used. Enable V.42 *bis* [1] (may also use MNP5 with MNP4 enabled)

*RAS 1.0 and 1.1 were tested only with US Robotics V.32 *bis* modems at 14400 bps line speed. The modems were configured for V.42, V.42 *bis*, and V.32 *bis*. The OS/2 1.3 LAN Manager RAS 1.0 or 1.1 server had an intelligent DigiBoard installed.

[1] If an uncompressed file is transmitted between two Microsoft RAS computers, RAS software compression increases throughput significantly more than V.42 *bis* compression. Do not enable modem compression (MNP5) along with RAS software compression, because it may decrease throughpu

Note RAS clients running version 1.0 or 1.1 on the MS-DOS platform should be upgraded to at least RAS version 1.1a, which can be obtained from Microsoft at no charge. RAS server version 1.0 and 1.1 on the OS/2 1.3 platforms were never updated. In general, for RAS 1.x versions on MS-DOS or OS/2 1.3, it is recommended that you upgrade to RAS for Windows for Workgroups 3.11, RAS for Windows 95 or Windows NT 3.51, because of improved performance, new features, and the new, easier-to-use interface.

RAS Communication Quick Reference

If you are a network administrator, consultant, or support engineer, you can use the following quick reference tables to gain a high level perspective of how RAS works over a WAN connection. These quick reference tables are high-level diagrams that show you at what point which RAS client features (of any Microsoft RAS version to date) execute when you make a call to a Windows NT RAS server.

How to Read the Diagrams

Specifically, these diagrams show how a call flows from a RAS client over different media (telephone lines, an ISDN line, or an X.25 network) through third-party security (or other) devices into the RAS server's port (serial or ISDN) and through the RAS server's architectural software layers to the RAS server service and, eventually, the RAS server. For a specific RAS client, you can also see whether it supports modem pools and other third party pre- or post-connect devices (via pre- and post-connect scripts in the SWITCH.INF file or RAS Terminal pop-up screens), whether it supports PAD.INF scripts for X.25 communication, and at what point during a call these scripts or Terminal screens execute.

The tables include the full path of all possible, but not necessary, events that can happen during a RAS call. To see the possible flow of events when a RAS client initiates a call to a Windows NT RAS server, first determine what type of communication medium you will be using: telephone lines, an ISDN line, an X.25 dial-up, or an X.25 with an Eicon card. Then, using the table specific to your situation, follow the path from the RAS client to the RAS server.

Information Not Included in the Diagrams

The diagrams do not show specific Windows NT RAS client security features for calling up third-party PPP or SLIP servers. To learn more about available Windows NT security features see the Help topic "Configuring Security." Note that Windows NT RAS can function as a SLIP client, but not as a SLIP server.

The diagrams also do not show direct serial (null modem) connections. For additional information about null modem connections, see the "Direct Serial Connections" section in the Windows NT Server version 3.5 *Remote Access Service* book.

ISDN Notes

When you call from a RAS client by using ISDN, the Pre-connect script/Terminal and Post-connect script/Terminal options are not available. This is because there is no ISDN standard defined on how ASCII characters are to be transmitted. For a detailed explanation of ISDN, see the Microsoft Knowledge Base article "Integrated Services Digital Network (ISDN)" (Q99767).

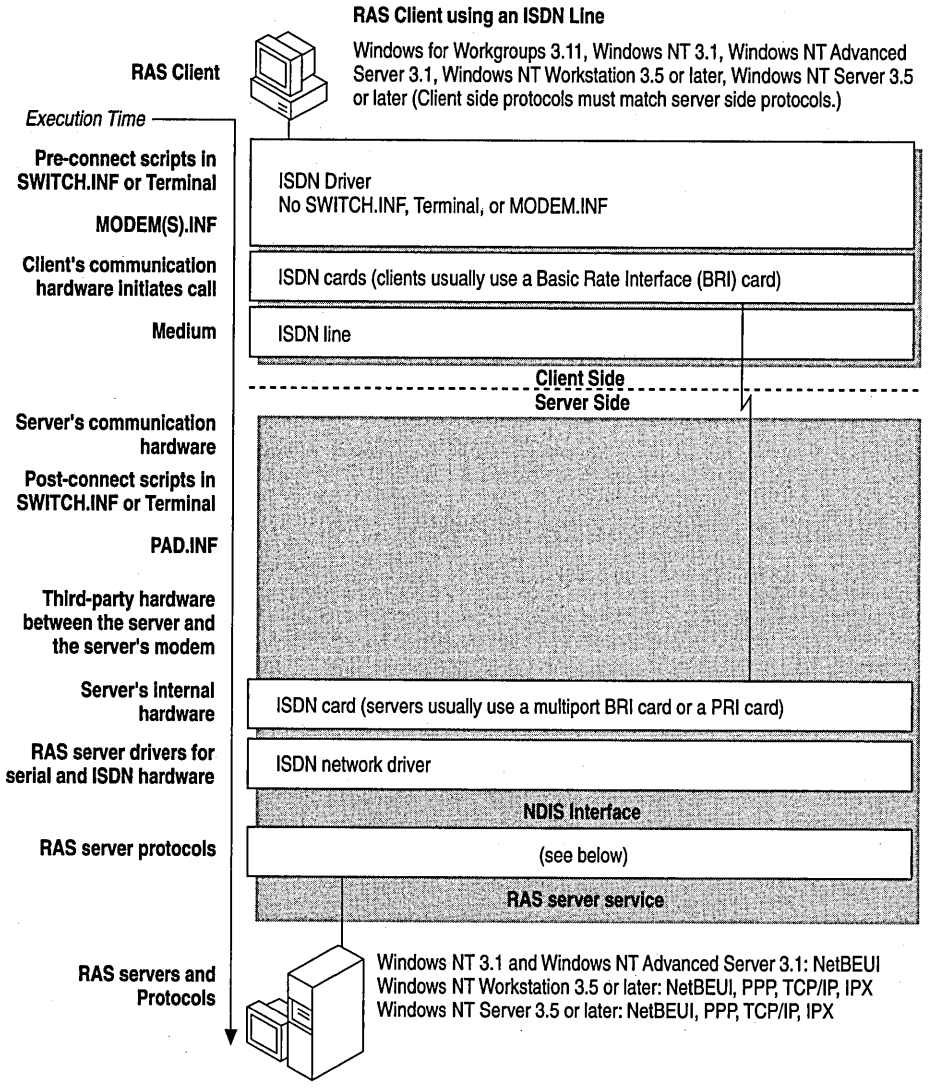


Figure C.4 RAS Client using an ISDN Line

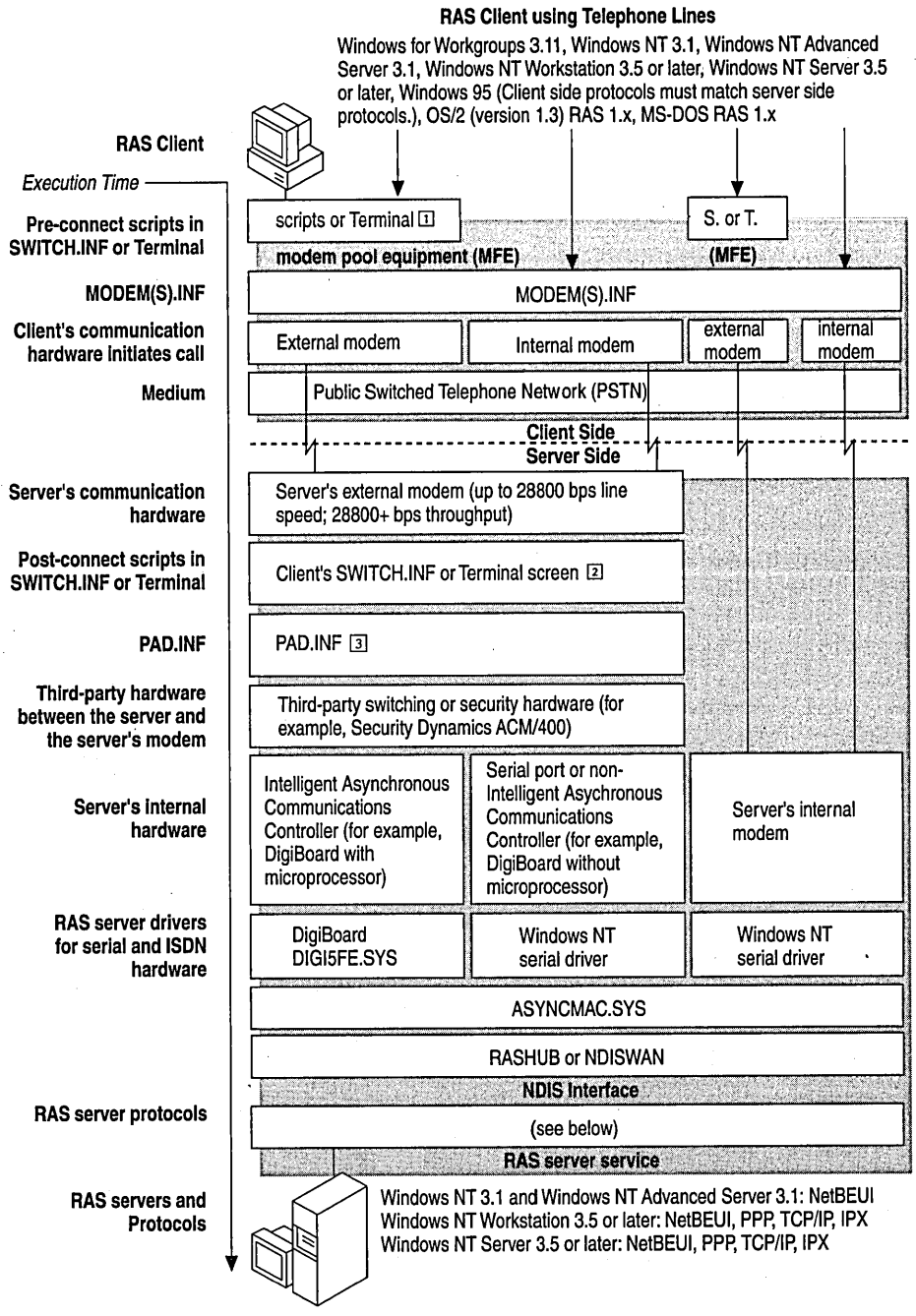


Figure C.5 RAS Client using Telephone Lines

1 RAS 1.x: no scripts or Terminal; Windows 95: no scripts 2 not in RAS 1.x 3 not in RAS 1.0

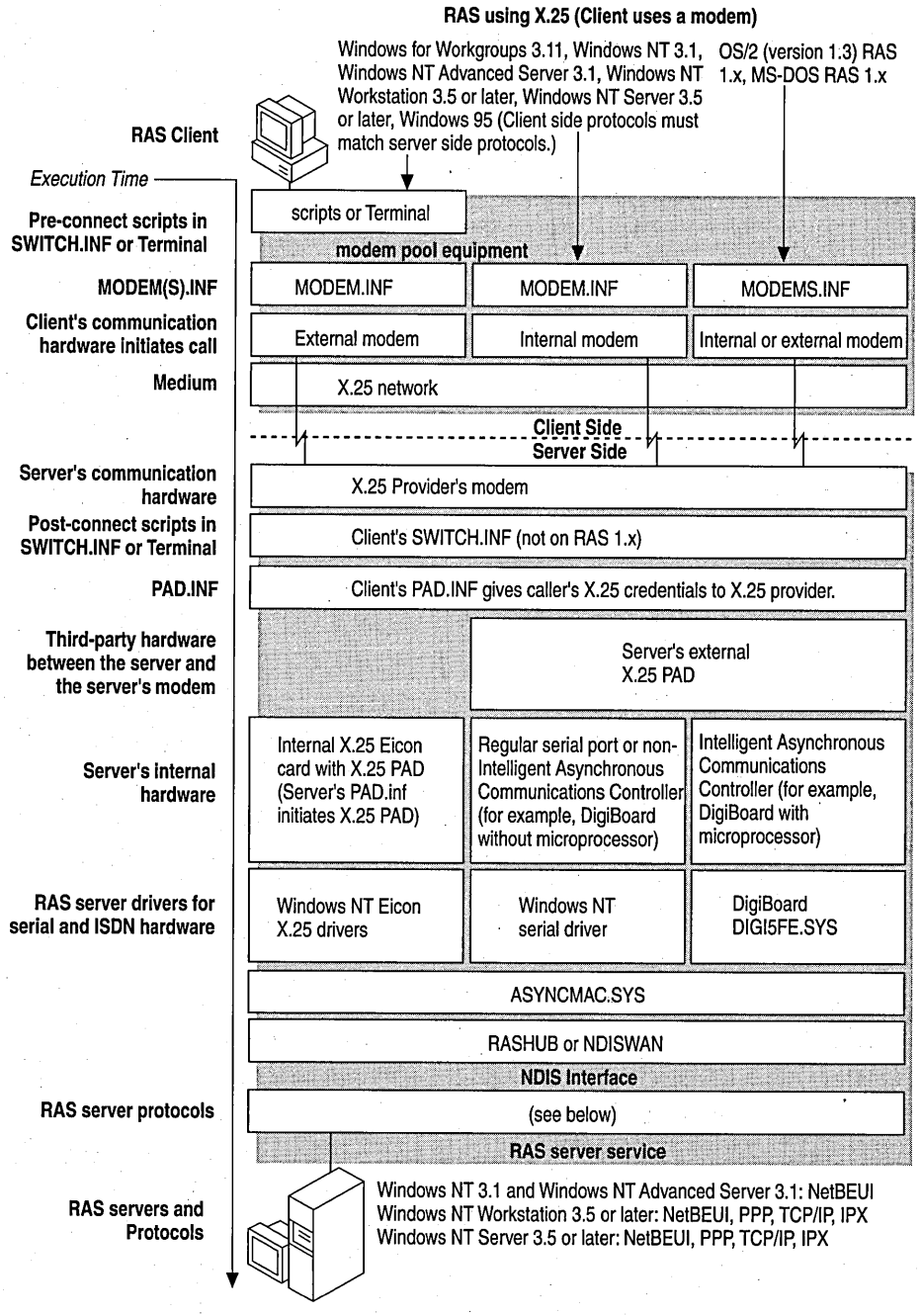


Figure C.6 RAS using X.25 (Client uses a modem)

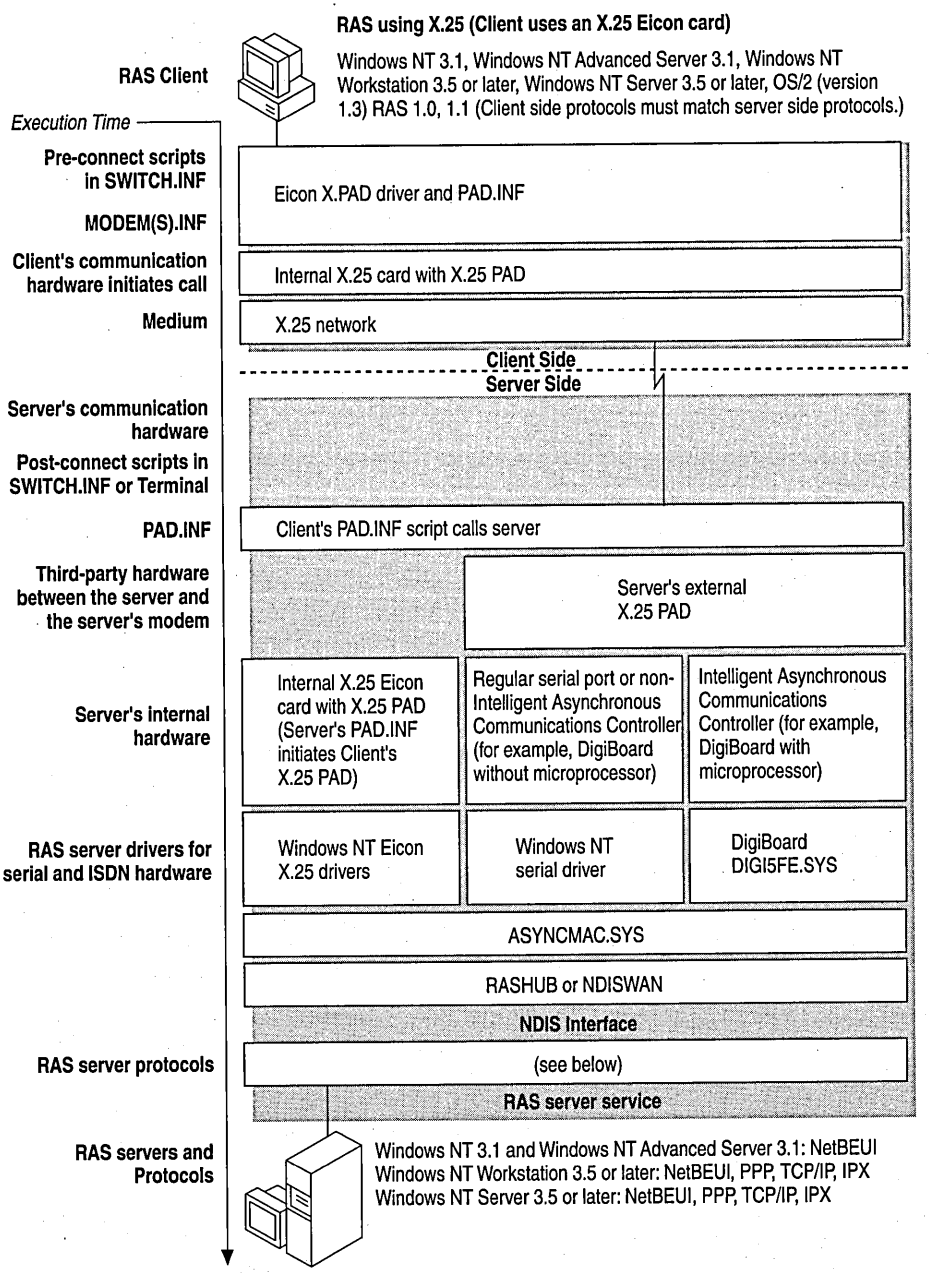


Figure C.7 RAS using X.25 (Client uses an X.25 Eicon card)

Microsoft Remote Access Version Features

The following tables list the most important features of all RAS server and client versions that were released before October 1995. The first table lists the features of RAS server versions, the second table lists the features of RAS client versions.

Table C.5 RAS Server Versions

Operating System	RAS version	Proprietary MS Protocol Support	PPP Support (Protocols tunneled through PPP Protocol)	SLIP Support (Protocols Tunneled through SLIP Protocol)	Medium
OS/2 1.31 (not OS/2 2.x) with LAN Manager 2.x	RAS 1.0	AsyBEUI [1]	No	No	Serial
OS/2 1.31 (not OS/2 2.x) with LAN Manager 2.x	RAS 1.1 (no 1.1a available)	AsyBEUI	No	No	Serial; X.25
Windows for Workgroups 3.11	Windows for Workgroups 3.11 "Point-to-Point Server"; Supports only access to its own resources	NetBEUI, which is backward compatible with AsyBEUI	No, but third-party PPP TCP/IP software is available	No	Serial; ISDN; No internal X.25 board support
Windows NT 3.1 (acting as a RAS server)	RAS for Windows NT 3.1	Uses NBF [2]	No	No	Serial; ISDN; X.25
Windows NT Advanced Server 3.1	RAS for Windows NT 3.1	Uses NBF	No	No	Serial; ISDN; X.25

RAS Software Compression	Flow Control Support	Highest Tested bps Rate for Modems (not ISDN lines)	Number of Simultaneous RAS Ports (Incoming Calls) Supported	Types of Modems Tested
No	No	9600 bps	16 ports	Analog; Null Modem
No	No	9600 bps (V.32) [3]	16 ports (13 with x.25 card)	Analog; Null Modem
A	Yes	19200 bps (using V.32 *bis*, V.42, and V.42 *bis*)	1 port only (RAS client may access only server's local drives. No network access)	Analog; Null Modem
A	Yes	57600 bps (using V.32 *bis*, V.42, and V.42 *bis*)	1 port only	Analog; Null Modem
A	Yes	57600 bps (using V.32 *bis*, V.42, and V.42 *bis*)	64 ports	Analog; Null Modem

Table C.5 RAS Server Versions *(continued)*

Operating System	RAS version	Proprietary MS Protocol Support	PPP Support (Protocols tunneled through PPP Protocol)	SLIP Support (Protocols Tunneled through SLIP Protocol)	Medium
Windows NT 3.5 Server and Windows NT 3.5 Workstation (Both can act as a RAS server)	RAS for Windows NT 3.5	Supports NBF	NBF, IPX, TCP/IP	No SLIP server functionality	Serial; ISDN; X.25
Windows NT 3.51 Server and Windows NT 3.51 Workstation (Both can act as a RAS server)	RAS for Windows NT 3.51	Supports NBF	NBF, IPX, TCP/IP	No SLIP server functionality	Serial; ISDN; X.25
Windows 95	Remote Network Access for Windows 95 (RNA)	Supports NBF	NBF, IPX	No SLIP server functionality	Serial; No internal X.25 board support

RAS Software Compression	Flow Control Support	Highest Tested bps Rate for Modems (not ISDN lines)	Number of Simultaneous RAS Ports (Incoming Calls) Supported	Types of Modems Tested
B	Yes	57600 bps (using V.34, V.fc/V.fast, V.42, and V.42 *bis*)	Windows NT 3.5 Server: 256 ports. Windows NT 3.5 Workstation: 1 port	Analog; Null Modem
Compatible with all RAS software compression methods	Yes	57600 bps (using V.34, V.fc/V.fast, V.42, and V.42 *bis*)	Windows NT 3.51 Server: 256 ports. Windows NT 3.51 Workstation: 1 port	Analog; Null Modem; certain PCMCIA modems [4]
B	Yes	57600 bps (using V.34, V.fc/V.fast, V.42, and V.42 *bis*)	1 port only (RAS client may access server's drive and network resources)	Analog; Null Modem; PCMCIA modems; certain cellular modems [5]

A: Compatible with Windows NT 3.1, Windows NT Advanced Server 3.1, and Windows for Workgroups 3.11 RAS compression. To achieve RAS software compression compatibility with Windows NT 3.5 Workstation or Server clients, you must install Windows NT 3.5 SP2 on the Windows NT 3.5 Workstation or Server computer that is acting as the RAS client.

B: Compatible with a Windows NT 3.5 Workstation or Server RAS client. To achieve RAS software compression compatibility with Windows NT 3.1, Windows NT Advanced Server 3.1, or Windows for Workgroups 3.11 RAS clients, you must install Windows NT 3.5 SP2 on your Windows NT 3.5 Workstation or Server computer that is acting as a RAS server.

[1] Used in RAS 1.x.

[2] NetBEUI Frame Protocol (NBF) which is backward compatible with AsyBEUI and NetBEUI (Windows for Workgroups 3.1)

[3] Modem error control (MNP4, V.42) and modem compression (MNP5, V.42 *bis*) are not supported, with the following exception: 19200 bps was tested with US Robotics V.32 *bis* modems and an intelligent DigiBoard on a server running OS/2 version 1.3 with LAN Manager 2.1 and RAS 1.1 using V.42 error checking and V.42 *bis* modem compression.

[4] Consult the Windows NT 3.51 Hardware Compatibility List (HCL) for a listing of supported PCMCIA modems. If your modem is not listed, you can check the Windows NT 3.51 MODEM.INF file contents to see whether your PCMCIA modem was included after the HCL was printed.

[5] Consult the updated Windows 95 Hardware Compatibility List (HCL) on the Microsoft Internet Web server.

Table C.6 RAS Client Versions

Operating System	RAS version	Proprietary MS Protocol Support	PPP Support (Protocols tunneled through PPP Protocol)	SLIP Support (Protocols tunneled through SLIP Protocol)	Medium
MS-DOS 5.0 and later	RAS 1.0	AsyBEUI [1]	No	No	Serial
MS-DOS 5.0 and later	RAS 1.1	AsyBEUI	No	No	Serial; X.25
MS-DOS 5.0 and later	RAS 1.1a	AsyBEUI	No	No	Serial; X.25
OS/2 1.31 (not OS/2 2.x) with LAN Manager 2.x	RAS 1.0	AsyBEUI	No	No	Serial
Windows for Workgroups 3.11	RAS for Windows for Workgroups 3.11	NetBEUI; compatible with AsyBEUI	No, but third-party PPP TCP/IP software is available	No	Serial; ISDN; X.25 dial-up only [3]
Windows NT 3.1	RAS for Windows NT 3.1	Uses NBF [2]	No	No	Serial; X.25; ISDN
Windows NT Advanced Server 3.1 (acting as a RAS client)	RAS for Windows NT Advanced Server 3.1	Uses NBF	No	No	Serial; X.25; ISDN
OS/2 1.31 (not OS/2 2.x) with LAN Manager 2.x	RAS 1.1	AsyBEUI	No	No	Serial; X.25

RAS Software Compression	Flow Control Support	Support for Scripts to interact with third party devices (X.25 dial-up and security hosts)	Interactive Terminal Screen [5]	Highest Tested bps Rate for Modems (not for ISDN)	Types of Modems Tested
No	No	No	No	9600 bps	Analog; Null Modem
No	No	X.25 dial-up: Use PAD.INF	No	9600 bps (V.32) [4]	Analog; Null Modem
No; use modem compression	Yes	X.25 dial-up: Use PAD.INF	No	19200 bps (tested with V.32 *bis*, V.42, and V.42 *bis* enabled	Analog; Null Modem
No	No	No	No	9600 bps	Analog; Null Modem
No	No	X.25 dial-up: Use PAD.INF	No	9600 bps [4]	Analog; Null Modem
A	Yes	X.25 dial-up: Use PAD.INF; Third-party intermediary security devices: Use SWITCH.INF file	Yes	19200 bps	Analog; Null Modem
A	Yes	X.25 dial-up: Use PAD.INF; Third-party intermediary security devices: Use SWITCH.INF file	Yes	57600 bps (using V.32 *bis*, V.42, and V.42 *bis*)	Analog; Null Modem
A	Yes	X.25 dial-up: Use PAD.INF; Third-party intermediary security devices: Use SWITCH.INF file	Yes	57600 bps (using V.32 *bis*, V.42, and V.42 *bis*)	Analog; Null Modem

Table C.6 RAS Client Versions *(continued)*

Operating System	RAS version	Proprietary MS Protocol Support	PPP Support (Protocols tunneled through PPP Protocol)	SLIP Support (Protocols Tunneled through SLIP Protocol)	Medium
Windows NT 3.5 Server and Windows NT 3.5 Workstation (Both can act as a RAS client)	RAS for Windows NT 3.5	Supports NBF	NBF, IPX, TCP/IP	TCP/IP	Serial; X.25; ISDN
Windows NT 3.51 Server and Windows NT 3.51 Workstation (Both can act as a RAS client)	RAS for Windows NT 3.51	Supports NBF	NBF, IPX, TCP/IP	TCP/IP	Serial; X.25; ISDN
Windows 95	Remote Network Access for Windows 95 (RNA)	Supports NBF	NBF, IPX TCP/IP	TCP/IP	Serial; X.25 dial-up only [3]

[1] Used in RAS 1.x.

[2] NetBEUI Frame Protocol (NBF) which is compatible with AsyBEUI and NetBEUI (Windows for Workgroups 3.11)

[3] Internal X.25 card not supported on Windows for Workgroups 3.11

[4] Modem error control (MNP4, V.42) and modem compression (MNP5, V.42 *bis*) are not supported, with the following exception: 19200 bps was tested with US Robotics V.32 *bis* modems and an intelligent DigiBoard on a server running OS/2 version 1.3 with LAN Manager 2.1 and RAS 1.1 using V.42 error checking and V.42 *bis* modem compression.

[5] Interactive Terminal screen can be invoked to log on to third party PPP hosts or intermediary security devices.

[6] Consult the Windows NT 3.51 Hardware Compatibility List (HCL) for a listing of supported PCMCIA modems. If your modem is not listed, you can check the Windows NT 3.51 MODEM.INF file contents to see whether your PCMCIA modem was included after the HCL was printed.

RAS Software Compression	Flow Control Support	Support for Scripts to interact with third party devices (X.25 dial-up and security hosts)	Interactive Terminal Screen [5]	Highest Tested bps Rate for Modems (not for ISDN)	Types of Modems Tested
B	Yes	X.25 dial-up: Use PAD.INF; Third-party intermediary security devices: Use SWITCH.INF file	Yes	57600bps (using V.34, V.fc/V.42, and V.42 *bis)*	Analog; Null Modem
Compatible with all RAS software compression methods.	Yes	X.25 dial-up: Use PAD.INF; Third-party intermediary security devices: Use SWITCH.INF file	Yes	57600bps (using V.34, V.fc/V.42, and V.42 *bis*	Analog; Null Modem; certain PCMCIA modems [6]
B	Yes	Use SCRIPTER.EXE ▣ for scripting connections with:- X.25 dial-up providers; third-party intermediary security devices; SLIP dial-up servers; PPP dial-up servers	Yes	57600 bps	Analog;Null Modem;PCM CIA modems; certain cellular modems ■

A: Compatible with Windows NT, Windows NT Advanced Server 3.1, and Windows for Workgroups 3.11 RAS compression. To achieve RAS software compression compatibility with Windows NT 3.5 Workstation and Server RAS servers, you must install the Windows NT 3.5 SP2 on the Windows NT 3.5 Workstation or Server that is acting as the RAS server.

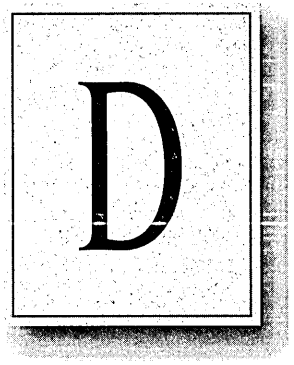
B: Compatible when calling a Windows NT 3.5 Workstation or Server that is acting as a RAS server. To achieve RAS software compression compatibility with Windows NT 3.1, Windows NT Advanced Server 3.1, or Windows for Workgroups 3.11 RAS servers, you need to install Windows NT 3.5 SP2 on your Windows NT 3.5 Server or Workstation that is acting as a RAS client.

▣ Windows 95 ships with SCRIPTER.EXE in the \ADMIN\APPTOOLS\DSCRIPT directory. The Windows 95 PLUS! software package of tools contains an upgraded version of SCRIPTER.EXE. **Note** SCRIPTER.EXE does not support pre-connect scripts to communicate with modem pool equipment. Instead, check the Bring Up Terminal Window Before Dialing option in the Option dialog box of the Properties of your connection to communicate interactively with your modem pool or pre-connect device.

■ Consult the updated Windows 95 Hardware Compatibility List (HCL) on the Microsoft Internet Web server*

APPENDIX D

RFC and Port Reference for Microsoft TCP/IP



Microsoft TCP/IP is compatible with TCP/IP implementations on LAN Manager and other networks that implement the standard TCP/IP protocol suite. This protocol defined in Requests for Comments (RFC) published by the Internet Engineering Task Force (IETF).

This paper contains a list of the RFCs implemented in Microsoft TCP/IP for Windows NT and the port numbers used for UDP and TCP connections.

This chapter provides information about the following:

- Microsoft TCP/IP RFC Reference
- Microsoft TCP/IP Port Reference
- RFC Source Reference

The information contained in this document represents the current view of the Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft TCP/IP RFC Reference

This section lists the RFCs and related TCP ports that are implemented in Windows NT version 3.5 and later when Microsoft TCP/IP is installed. Most of these RFCs are also implemented in Microsoft TCP/IP under Windows 95®.

RFC	Title	Comments
768	User Datagram Protocol (UDP)	
783	Trivial File Transfer Protocol (TFTP)	
791	Internet Protocol (IP)	
792	Internet Control Message Protocol (ICMP)	
793	Transmission Control Protocol (TCP)	
826	Address Resolution Protocol (ARP)	
854	Telnet Protocol (TELNET)	
862	Echo Protocol (ECHO)	
863	Discard Protocol (DISCARD)	
864	Character Generator Protocol (CHARGEN)	
865	Quote of the Day Protocol (QUOTE)	
894	IP over Ethernet	
919, 922	IP Broadcast Datagrams (broadcasting with subnets)	
959	File Transfer Protocol (FTP)	
1001, 1002	NetBIOS Service Protocols	NetBIOS over TCP/IP in Microsoft TCP/IP uses the following TCP and UDP ports: UDP port 137 (name services) UDP port 138 (datagram services) TCP port 139 (session services) Microsoft WINS listens on port 137; NetBIOS name servers from other vendors may listen on different ports.

RFC	Title	Comments
1034, 1035	Domain Name System (DNS)	Domain Name System over TCP/IP in Microsoft TCP/IP uses TCP port 53 for DNS queries larger than 512 bytes.
	IP over Token Ring	
1055	Transmission of IP over Serial Lines (IP-SLIP)	
1112	Internet Gateway Multicast Protocol (IGMP)	
1122, 1123	Host Requirements (communications and applications)	
1134	Point to Point Protocol (PPP)	
1144	Compressing TCP/IP Headers for Low-Speed Serial Links	
1157	Simple Network Management Protocol (SNMP)	A protocol entity receives messages at UDP port 161 on its associated host for all messages, except for those that report traps. Messages that report traps are received on UDP port 162 for further processing.
1188	IP over FDDI	
1191	Path MTU Discovery	
1201	IP over ARCNET1201	
1231	IEEE 802.5 Token Ring MIB (MIB-II)1231	
1332	PPP Internet Protocol Control Protocol (IPCP)	
1334	PPP Authentication Protocols	
1533	DHCP Options and BOOTP Vendor Extensions	

RFC	Title	Comments
1534	Interoperation Between DHCP and BOOTP	The BOOTP Client uses UDP port 68 and BOOTP Server uses UDP 67. IP multicasting uses these ports as the source and destination ports, respectively.Interoperation Between DHCP and BOOTP
1541	Dynamic Host Configuration Protocol (DHCP)	DHCP uses UDP as its transport protocol. DHCP messages from a client to a server are sent to the DHCP server port (67), and DHCP messages from a server to a client are sent to the DHCP client port (68).
1542	Clarifications and Extensions for the Bootstrap Protocol	
1547	Requirements for Point to Point Protocol (PPP)	
1548	Point to Point Protocol (PPP)	
1549	PPP in High-level Data Link Control (HDLC) Framing	
15521	PPP Internetwork Packet Exchange Control Protocol (IPXCP)	
1553	IPX Header Compression	
1570	Link Control Protocol (LCP) Extensions	
Draft RFCs	NetBIOS Frame Control Protocol (NBFCP); PPP over ISDN; PPP over X.25; Compression Control Protocol	

Microsoft TCP/IP Port Reference

This section presents information from the SERVICES file provided with Windows NT and the port numbers for well-known services as defined by RFC 1060. From RFC 1340 (J. Reynolds and J. Postel, July 1992):

> The Well Known Ports are controlled and assigned by the Internet Assigned Numbers Authority (IANA), and on most systems can only be used by system (or root) processes or by programs executed by privileged users.
>
> Ports are used in TCP to name the ends of logical connections that carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port."
>
> To the extent possible, these same port assignments are used with UDP. The assigned ports use a small portion of the possible port numbers. For many years, the assigned ports were in the range 0 – 255. Recently, the range for assigned ports managed by the IANA has been expanded to the range 0 – 1023.

Port Assignments for Well Know Ports

The following table describes port assignments for well-known ports.

Table D.1 Port Assignments for Well Known Ports

Decimal	Keyword	Description and Microsoft networking alias
0/tcp, udp		Reserved
1/tcp, udp	tcpmux	TCP Port Service Multiplexer
2/tcp, udp	compressnet	Management Utility
3/tcp, udp	compressnet	Compression Process
4/tcp, udp		Unassigned
5/tcp, udp	rje	Remote Job Entry
6/tcp, udp		Unassigned
7/tcp, udp	echo	Echo
8/tcp, udp		Unassigned
9/tcp, udp	discard	Discard; alias=sink null
10/tcp, udp		Unassigned
11/udp	systat	Active Users; alias=users
12/tcp, udp		Unassigned
13/tcp, udp	daytime	Daytime
14/tcp, udp		Unassigned
15/tcp, udp		Unassigned [was netstat]

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
16/tcp, udp		Unassigned
17/tcp, udp	qotd	Quote of the Day; alias=quote
18/tcp, udp	msp	Message Send Protocol
19/tcp, udp	chargen	Character Generator; alias=ttytst source
20/tcp, udp	ftp-data	File Transfer [Default Data]
21/tcp, udp	ftp	File Transfer [Control]
22/tcp, udp		Unassigned
23/tcp, udp	telnet	Telnet
24/tcp, udp		Any private mail system
25/tcp, udp	smtp	Simple Mail Transfer; alias=mail
26/tcp, udp		Unassigned
27/tcp, udp	nsw-fe	NSW User System FE
28/tcp, udp		Unassigned
29/tcp, udp	msg-icp	MSG ICP
30/tcp, udp		Unassigned
31/tcp, udp	msg-auth	MSG Authentication
32/tcp, udp		Unassigned
33/tcp, udp	dsp	Display Support Protocol
34/tcp, udp		Unassigned
35/tcp, udp		Any private printer server
36/tcp, udp		Unassigned
37/tcp, udp	time	Time; alias=timserver
38/tcp, udp		Unassigned
39/tcp, udp	rlp	Resource Location Protocol; alias=resource
40/tcp, udp		Unassigned
41/tcp, udp	graphics	Graphics
42/tcp, udp	nameserver	Host Name Server; alias=nameserver
43/tcp, udp	nicname	Who Is; alias=nicname
44/tcp, udp	mpm-flags	MPM FLAGS Protocol
45/tcp, udp	mpm	Message Processing Module
46/tcp, udp	mpm-snd	MPM [default send]
47/tcp, udp	ni-ftp	NI FTP
48/tcp, udp		Unassigned
49/tcp, udp	login	Login Host Protocol

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
50/tcp, udp	re-mail-ck	Remote Mail Checking Protocol
51/tcp, udp	la-maint	IMP Logical Address Maintenance
52/tcp, udp	xns-time	XNS Time Protocol
53/tcp, udp	domain	Domain Name Server; alias=nameserver, dns
54/tcp, udp	xns-ch	XNS Clearinghouse
55/tcp, udp	isi-gl	ISI Graphics Language
56/tcp, udp	xns-auth	XNS Authentication
57/tcp, udp		Any private terminal access
58/tcp, udp	xns-mail	XNS Mail
59/tcp, udp		Any private file service
60/tcp, udp		Unassigned
61/tcp, udp	ni-mail	NI MAIL
62/tcp, udp	acas	ACA Services
63/tcp, udp	via-ftp	VIA Systems - FTP
64/tcp, udp	covia	Communications Integrator (CI)
65/tcp, udp	tacacs-ds	TACACS-Database Service
66/tcp, udp	sql*net	Oracle SQL*NET
67/tcp, udp	bootpc	DHCP/BOOTP Protocol Server
68/tcp, udp	bootpc	DHCP/BOOTP Protocol Server
69/tcp, udp	tftp	Trivial File Transfer
70/tcp, udp	gopher	Gopher
71/tcp, udp	netrjs-1	Remote Job Service
72/tcp, udp	netrjs-2	Remote Job Service
73/tcp, udp	netrjs-3	Remote Job Service
74/tcp, udp	netrjs-4	Remote Job Service
75/udp		Any private dial out service
76/tcp, udp		Unassigned
77/tcp, udp		Any private RJE service; alias=netrjs
78/tcp, udp	vettcp	Vettcp
79/tcp, udp	finger	Finger
80/tcp, udp	www	World Wide Web HTTP
81/tcp, udp	hosts2-ns	HOSTS2 Name Server
82/tcp, udp	xfer	XFER Utility
83/tcp, udp	mit-ml-dev	MIT ML Device

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
84/tcp, udp	ctf	Common Trace Facility
85/tcp, udp	mit-ml-dev	MIT ML Device
86/tcp, udp	mfcobol	Micro Focus Cobol
87/tcp, udp		Any private terminal link; alias=ttylink
88/tcp, udp	kerberos	Kerberos
89/tcp	su-mit-tg	SU/MIT Telnet Gateway
89/udp	su-mit-tg	SU/MIT Telnet Gateway
90/tcp, udp		Default WINS name server destination port
91/tcp, udp	mit-dov	MIT Dover Spooler
92/tcp, udp	npp	Network Printing Protocol
93/tcp, udp	dcp	Device Control Protocol
94/tcp, udp	objcall	Tivoli Object Dispatcher
95/tcp, udp	supdup	SUPDUP
96/tcp, udp	dixie	DIXIE Protocol Specification
97/tcp, udp	swift-rvf	Swift Remote Virtual File Protocol
98/tcp, udp	tacnews	TAC News
99/tcp, udp	metagram	Metagram Relay
100/tcp	newacct	[unauthorized use]
101/tcp, udp	hostname	NIC Host Name Server; alias=hostname
102/tcp, udp	iso-tsap	ISO-TSAP
103/tcp, udp	gppitnp	Genesis Point-to-Point Trans Net; alias=webster
104/tcp, udp	acr-nema	ACR-NEMA Digital Imag. & Comm. 300
105/tcp, udp	csnet-ns	Mailbox Name Nameserver
106/tcp, udp	3com-tsmux	3COM-TSMUX
107/tcp, udp	rtelnet	Remote Telnet Service
108/tcp, udp	snagas	SNA Gateway Access Server
109/tcp, udp	pop2	Post Office Protocol - Version 2; alias=postoffice
110/tcp, udp	pop3	Post Office Protocol - Version 3; alias=postoffice
111/tcp, udp	sunrpc	SUN Remote Procedure Call
112/tcp, udp	mcidas	McIDAS Data Transmission Protocol
113/tcp, udp	auth	Authentication Service; alias=authentication
114/tcp, udp	audionews	Audio News Multicast

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
115/tcp, udp	sftp	Simple File Transfer Protocol
116/tcp, udp	ansanotify	ANSA REX Notify
117/tcp, udp	uucp-path	UUCP Path Service
118/tcp, udp	sqlserv	SQL Services
119/tcp, udp	nntp	Network News Transfer Protocol; alias=usenet
120/tcp, udp	cfdptkt	CFDPTKT
121/tcp, udp	erpc	Encore Expedited Remote Pro.Call
122/tcp, udp	smakynet	SMAKYNET
123/tcp, udp	ntp	Network Time Protocol; alias=ntpd ntp
124/tcp, udp	ansatrader	ANSA REX Trader
125/tcp, udp	locus-map	Locus PC-Interface Net Map Server
126/tcp, udp	unitary	Unisys Unitary Login
127/tcp, udp	locus-con	Locus PC-Interface Conn Server
128/tcp, udp	gss-xlicen	GSS X License Verification
129/tcp, udp	pwdgen	Password Generator Protocol
130/tcp, udp	cisco-fna	Cisco FNATIVE
131/tcp, udp	cisco-tna	Cisco TNATIVE
132/tcp, udp	cisco-sys	Cisco SYSMAINT
133/tcp, udp	statsrv	Statistics Service
134/tcp, udp	ingres-net	INGRES-NET Service
135/tcp, udp	loc-srv	Location Service
136/tcp, udp	profile	PROFILE Naming System
137/tcp, udp	netbios-ns	NetBIOS Name Service
138/tcp, udp	netbios-dgm	NetBIOS Datagram Service
139/tcp, udp	netbios-ssn	NetBIOS Session Service
140/tcp, udp	emfis-data	EMFIS Data Service
141/tcp, udp	emfis-cntl	EMFIS Control Service
142/tcp, udp	bl-idm	Britton-Lee IDM
143/tcp, udp	imap2	Interim Mail Access Protocol v2
144/tcp, udp	news	NewS; alias=news
145/tcp, udp	uaac	UAAC Protocol
146/tcp, udp	iso-tp0	ISO-IP0
147/tcp, udp	iso-ip	ISO-IP
148/tcp, udp	cronus	CRONUS-SUPPORT

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
149/tcp, udp	aed-512	AED 512 Emulation Service
150/tcp, udp	sql-net	SQL-NET
151/tcp, udp	hems	HEMS
152/tcp, udp	bftp	Background File Transfer Program
153/tcp, udp	sgmp	SGMP; alias=sgmp
154/tcp, udp	netsc-prod	NETSC
155/tcp, udp	netsc-dev	NETSC
156/tcp, udp	sqlsrv	SQL Service
157/tcp, udp	knet-cmp	KNET/VM Command/Message Protocol
158/tcp, udp	pcmail-srv	PCMail Server; alias=repository
159/tcp, udp	nss-routing	NSS-Routing
160/tcp, udp	sgmp-traps	SGMP-TRAPS
161/tcp, udp	snmp	SNMP; alias=snmp
162/tcp, udp	snmptrap	SNMPTRAP
163/tcp, udp	cmip-man	CMIP/TCP Manager
164/tcp, udp	cmip-agent	CMIP/TCP Agent
165/tcp, udp	xns-courier	Xerox
166/tcp, udp	s-net	Sirius Systems
167/tcp, udp	namp	NAMP
168/tcp, udp	rsvd	RSVD
169/tcp, udp	send	SEND
170/tcp, udp	print-srv	Network PostScript
171/tcp, udp	multiplex	Network Innovations Multiplex
172/tcp, udp	cl/1	Network Innovations CL/1
173/tcp, udp	xyplex-mux	Xyplex
174/tcp, udp	mailq	MAILQ
175/tcp, udp	vmnet	VMNET
176/tcp, udp	genrad-mux	GENRAD-MUX
177/tcp, udp	xdmcp	X Display Manager Control Protocol
178/tcp, udp	nextstep	NextStep Window Server
179/tcp, udp	bgp	Border Gateway Protocol
180/tcp, udp	ris	Intergraph
181/tcp, udp	unify	Unify
182/tcp, udp	audit	Unisys Audit SITP

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
183/tcp, udp	ocbinder	OCBinder
184/tcp, udp	ocserver	OCServer
185/tcp, udp	remote-kis	Remote-KIS
186/tcp, udp	kis	KIS Protocol
187/tcp, udp	aci	Application Communication Interface
188/tcp, udp	mumps	Plus Five's MUMPS
189/tcp, udp	qft	Queued File Transport
190/tcp, udp	gacp	Gateway Access Control Protocol
191/tcp, udp	prospero	Prospero
192/tcp, udp	osu-nms	OSU Network Monitoring System
193/tcp, udp	srmp	Spider Remote Monitoring Protocol
194/tcp, udp	irc	Internet Relay Chat Protocol
195/tcp, udp	dn6-nlm-aud	DNSIX Network Level Module Audit
196/tcp, udp	dn6-smm-red	DNSIX Session Mgt Module Audit Redir
197/tcp, udp	dls	Directory Location Service
198/tcp, udp	dls-mon	Directory Location Service Monitor
199/tcp, udp	smux	SMUX
200/tcp, udp	src	IBM System Resource Controller
201/tcp, udp	at-rtmp	AppleTalk Routing Maintenance
202/tcp, udp	at-nbp	AppleTalk Name Binding
203/tcp, udp	at-3	AppleTalk Unused
204/tcp, udp	at-echo	AppleTalk Echo
205/tcp, udp	at-5	AppleTalk Unused
206/tcp, udp	at-zis	AppleTalk Zone Information
207/tcp, udp	at-7	AppleTalk Unused
208/tcp, udp	at-8	AppleTalk Unused
209/tcp, udp	tam	Trivial Authenticated Mail Protocol
210/tcp, udp	z39.50	ANSI Z39.50
211/tcp, udp	914c/g	Texas Instruments 914C/G Terminal
212/tcp, udp	anet	ATEXSSTR
213/tcp, udp	ipx	IPX
214/tcp, udp	vmpwscs	VM PWSCS
215/tcp, udp	softpc	Insignia Solutions
216/tcp, udp	atls	Access Technology License Server

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
217/tcp, udp	dbase	dBASE UNIX
218/tcp, udp	mpp	Netix Message Posting Protocol
219/tcp, udp	uarps	Unisys ARPs
220/tcp, udp	imap3	Interactive Mail Access Protocol v3
221/tcp, udp	fln-spx	Berkeley rlogind with SPX auth
222/tcp, udp	fsh-spx	Berkeley rshd with SPX auth
223/tcp, udp	cdc	Certificate Distribution Center
224-241		Reserved
243/tcp, udp	sur-meas	Survey Measurement
245/tcp, udp	link	LINK
246/tcp, udp	dsp3270	Display Systems Protocol
247-255		Reserved
345/tcp, udp	pawserv	Perf Analysis Workbench
346/tcp, udp	zserv	Zebra server
347/tcp, udp	fatserv	Fatmen Server
371/tcp, udp	clearcase	Clearcase
372/tcp, udp	ulistserv	UNIX Listserv
373/tcp, udp	legent-1	Legent Corporation
374/tcp, udp	legent-2	Legent Corporation
512/tcp	exec	Remote process execution; authentication performed using passwords and UNIX login names
512/udp	biff	Used by mail system to notify users of new mail received; currently receives messages only from processes on the same computer; alias=comsat
513/tcp	login	Remote login like telnet; automatic authentication performed based on privileged port numbers and distributed data bases that identify "authentication domains"
513/udp	who	Maintains data bases showing who's logged in to machines on a local net and the load average of the machine; alias=whod
514/tcp	cmd	Like exec, but automatic authentication is performed as for login server
514/udp	syslog	
515/tcp, udp	printer	Spooler; alias=spooler

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
517/tcp, udp	talk	Like tenex link, but across machine; unfortunately, doesn't use link protocol (this is actually just a rendezvous port from which a TCP connection is established)
518/tcp, udp	ntalk	
519/tcp, udp	utime	Unixtime
520/tcp	efs	Extended file name server
520/udp	router	Local routing process (on site); uses variant of Xerox NS routing information protocol;alias=router routed
525/tcp, udp	timed	Timeserver
526/tcp, udp	tempo	Newdate
530/tcp, udp	courier	RPC
531/tcp	conference	Chat
531/udp	rvd-control	MIT disk
532/tcp, udp	netnews	Readnews
533/tcp, udp	netwall	For emergency broadcasts
540/tcp, udp	uucp	Uucpd
543/tcp, udp	klogin	
544/tcp, udp	kshell	Krcmd; alias=cmd
550/tcp, udp	new-rwho	New-who
555/tcp, udp	dsf	
556/tcp, udp	remotefs	Rfs server; alias=rfs_server rfs
560/tcp, udp	rmonitor	Rmonitord
561/tcp, udp	monitor	
562/tcp, udp	chshell	Chcmd
564/tcp, udp	9pfs	Plan 9 file service
565/tcp, udp	whoami	Whoami
570/tcp, udp	meter	Demon
571/tcp, udp	meter	Udemon
600/tcp, udp	ipcserver	Sun IPC server
607/tcp, udp	nqs	Nqs
666/tcp, udp	mdqs	
704/tcp, udp	elcsd	Errlog copy/server daemon
740/tcp, udp	netcp	NETscout Control Protocol

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
741/tcp, udp	netgw	NetGW
742/tcp, udp	netrcs	Network based Rev. Cont. Sys.
744/tcp, udp	flexlm	Flexible License Manager
747/tcp, udp	fujitsu-dev	Fujitsu Device Control
748/tcp, udp	ris-cm	Russell Info Sci Calendar Manager
749/tcp, udp	kerberos-adm	Kerberos administration
750/tcp	rfile	Kerberos authentication; alias=kdc
750/udp	loadav	
751/tcp, udp	pump	Kerberos authentication
752/tcp, udp	qrh	Kerberos password server
753/tcp, udp	rrh	Kerberos userreg server
754/tcp, udp	tell	Send; Kerberos slave propagation
758/tcp, udp	nlogin	
759/tcp, udp	con	
760/tcp, udp	ns	
761/tcp, udp	rxe	
762/tcp, udp	quotad	
763/tcp, udp	cycleserv	
764/tcp, udp	omserv	
765/tcp, udp	webster	
767/tcp, udp	phonebook	Phone
769/tcp, udp	vid	
770/tcp, udp	cadlock	
771/tcp, udp	rtip	
772/tcp, udp	cycleserv2	
773/tcp	submit	
773/udp	notify	
774/tcp	rpasswd	
774/udp	acmaint_dbd	
775/tcp	entomb	
775/udp	acmaint_transd	
776/tcp, udp	wpages	
780/tcp, udp	wpgs	
781/tcp, udp	hp-collector	HP performance data collector

Table D.1 Port Assignments for Well Known Ports *(continued)*

Decimal	Keyword	Description and Microsoft networking alias
782/tcp, udp	hp-managed-node	HP performance data managed node
783/tcp, udp	hp-alarm-mgr	HP performance data alarm manager
800/tcp, udp	mdbs_daemon	
801/tcp, udp	device	
888/tcp	erlogin	Login and environment passing
996/tcp, udp	xtreelic	XTREE License Server
997/tcp, udp	maitrd	
998/tcp	busboy	
998/udp	puparp	
999/tcp	garcon	
999/udp	applix	Applix ac
999/tcp, udp	puprouter	
1000/tcp	cadlock	
1000/udp	ock	

Port Assignments for Registered Ports

The Registered Ports are not controlled by the IANA and on most systems can be used by user processes or programs. This list specifies the port used by the server process as its contact port. Although the IANA cannot control uses of these ports, it does register or list uses of these ports as a convenience to the TCP/IP community. To the extent possible, these same port assignments are used with UDP. The Registered Ports are in the range 1024–65535.

Table D.2 Port Assignments for Registered Ports

Decimal	Keyword	Description
1025/tcp, udp	blackjack	Network blackjack
1109/tcp	kpop	Pop with Kerberos
1167/udp	phone	
1248/tcp, udp	hermes	
1347/tcp, udp	bbn-mmc	Multimedia conferencing
1348/tcp, udp	bbn-mmx	Multimedia conferencing
1349/tcp, udp	sbook	Registration Network Protocol
1350/tcp, udp	editbench	Registration Network Protocol

Table D.2 Port Assignments for Registered Ports *(continued)*

Decimal	Keyword	Description
1351/tcp, udp	equationbuilder	Digital Tool Works (MIT)
1352/tcp, udp	lotusnote	Lotus Note
1524/tcp, udp	ingreslock	Ingres
1525/tcp, udp	orasrv	Oracle
1525/tcp, udp	prospero-np	Prospero nonprivileged
1527/tcp, udp	tlisrv	Oracle
1529/tcp, udp	coauthor	Oracle
1600/tcp, udp	issd	
1650/tcp, udp	nkd	
1666/udp	maze	
2000/tcp, udp	callbook	
2001/tcp	dc	
2001/udp	wizard	Curry
2002/tcp, udp	globe	
2004/tcp	mailbox	
2004/udp	emce	CCWS mm conf
2005/tcp	berknet	
2005/udp	oracle	
2006/tcp	invokator	
2006/udp	raid-cc	RAID
2007/tcp	dectalk	
2007/udp	raid-am	
2008/tcp	conf	
2008/udp	terminaldb	
2009/tcp	news	
2009/udp	whosockami	
2010/tcp	search	
2010/udp	pipe_server	
2011/tcp	raid-cc	RAID
2011/udp	servserv	
2012/tcp	ttyinfo	
2012/udp	raid-ac	
2013/tcp	raid-am	

Table D.2 Port Assignments for Registered Ports *(continued)*

Decimal	Keyword	Description
2013/udp	raid-cd	
2014/tcp	troff	
2014/udp	raid-sf	
2015/tcp	cypress	
2015/udp	raid-cs	
2016/tcp, udp	bootserver	
2017/tcp	cypress-stat	
2017/udp	bootclient	
2018/tcp	terminaldb	
2018/udp	rellpack	
2019/tcp	whosockami	
2019/udp	about	
2020/tcp, udp	xinupageserver	
2021/tcp	servexec	
2021/udp	xinuexpansion1	
2022/tcp	down	
2022/udp	xinuexpansion2	
2023/tcp, udp	xinuexpansion3	
2024/tcp, udp	xinuexpansion4	
2025/tcp	ellpack	
2025/udp	xribs	
2026/tcp, udp	scrabble	
2027/tcp, udp	shadowserver	
2028/tcp, udp	submitserver	
2030/tcp, udp	device2	
2032/tcp, udp	blackboard	
2033/tcp, udp	glogger	
2034/tcp, udp	scoremgr	
2035/tcp, udp	imsldoc	
2038/tcp, udp	objectmanager	
2040/tcp, udp	lam	
2041/tcp, udp	interbase	
2042/tcp, udp	isis	

Table D.2 Port Assignments for Registered Ports *(continued)*

Decimal	Keyword	Description
2043/tcp, udp	isis-bcast	
2044/tcp, udp	rimsl	
2045/tcp, udp	cdfunc	
2046/tcp, udp	sdfunc	
2047/tcp, udp	dls	
2048/tcp, udp	dls-monitor	
2049/tcp, udp	shilp	Sun NFS
2053/tcp	knetd	Kerberos de-multiplexer
2105/tcp	eklogin	Kerberos encrypted rlogin
2784/tcp, udp	www-dev	World Wide Web - development
3049/tcp, udp	NSWS	
4672/tcp, udp	rfa	Remote file access server
5000/tcp, udp	commplex-main	
5001/tcp, udp	commplex-link	
5002/tcp, udp	rfe	Radio Free Ethernet
5145/tcp, udp	rmonitor_secure	
5236/tcp, udp	padl2sim	
5555/tcp	rmt	Rmtd
5556/tcp	mtb	Mtbd (mtb backup)
6111/tcp, udp	sub-process	HP SoftBench Sub-Process Control
6558/tcp, udp	xdsxdm	
7000/tcp, udp	afs3-fileserver	File server itself
7001/tcp, udp	afs3-callback	Callbacks to cache managers
7002/tcp, udp	afs3-prserver	Users and groups database
7003/tcp, udp	afs3-vlserver	Volume location database
7004/tcp, udp	afs3-kaserver	AFS/Kerberos authentication service
7005/tcp, udp	afs3-volser	Volume management server
7006/tcp, udp	afs3-errors	Error interpretation service
7007/tcp, udp	afs3-bos	Basic overseer process
7008/tcp, udp	afs3-update	Server-to-server updater
7009/tcp, udp	afs3-rmtsys	Remote cache manager service
9535/tcp, udp	man	Remote man server
9536/tcp	w	

Table D.2 Port Assignments for Registered Ports *(continued)*

Decimal	Keyword	Description
9537/tcp	mantst	Remote man server, testing
10000/tcp	bnews	
10000/udp	rscs0	
10001/tcp	queue	
10001/udp	rscs1	
10002/tcp	poker	
10002/udp	rscs2	
10003/tcp	gateway	
10003/udp	rscs3	
10004/tcp	remp	
10004/udp	rscs4	
10005/udp	rscs5	
10006/udp	rscs6	
10007/udp	rscs7	
10008/udp	rscs8	
10009/udp	rscs9	
10010/udp	rscsa	
10011/udp	rscsb	
10012/tcp	qmaster	
10012/udp	qmaster	
17007/tcp, udp	isode-dua	

RFC Source Reference

All RFCs can be found on the Internet by using *ds.internic.net*. RFCs can also be obtained from the following primary sites by using FTP.

RFC repository	Comment[1]
nic.ddn.mil	Use the pathname ***rfc/rfcnnnn.txt***. Login with FTP username *anonymous* and the password *guest*.
ftp.nisc.sri.com	Use the pathname ***rfc/rfcnnnn.txt*** or ***rfc/rfcnnnn.ps***. Login with FTP username *anonymous* and the password *guest*. To obtain the RFC Index, use the pathname ***rfc/rfc-index.txt***.
nis.nsf.net	Login with FTP username *anonymous* and the password *guest*; then connect to the RFC directory (**cd RFC**). The filename is of the form ***RFCnnnn.TXT-***.
nisc.jvnc.net	Use the pathname ***rfc/RFCnnnn.TXT.v*** (where *v* refers to the version number of the RFC).
venera.isi.edu	Use the pathname ***in-notes/rfcnnnn.txt***. Login with FTP username *anonymous* and the password *guest*.
wuarchive.wustl.edu	Use the pathname ***info/rfc/rfcnnnn.txt.Z*** (where ***.Z*** indicates that the document is in compressed form).
src.doc.ic.ac.uk	Use the pathname ***rfc/rfcnnnn.txt.Z*** or ***rfc/rfcnnnn.ps.Z***. Login with FTP username *anonymous* and the password of your Internet e-mail name. To obtain the RFC Index, use the pathname ***rfc/rfc-index.txt.Z*** (where ***.Z*** indicates that the document is in compressed form.)
ftp.concert.net	Login with username *anonymous* and your Internet e-mail address as the password. The RFCs can be found in the directory /rfc, with filenames of the form: ***rfcnnnn.txt*** or ***rfcnnnn.ps***. This repository is also accessible by using WAIS and the Internet Gopher.

[1] In this table, *nnnn* refers to the number of the RFC.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

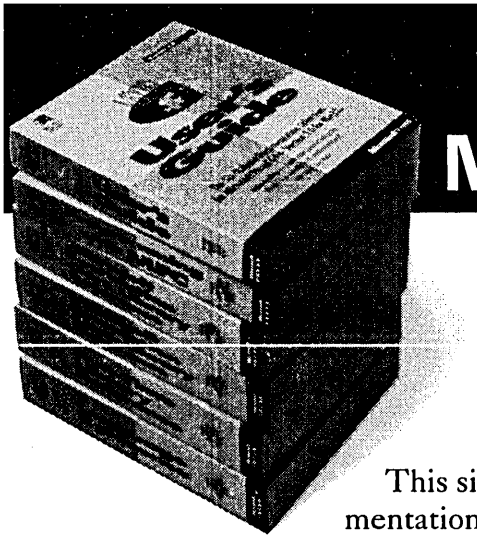
T

U

V

W

X



When you program in Microsoft® Visual C++™,

be ready for everything.

ISBN 1-55615-901-3, U.S.A. $159.95, (U.K. £147.99; Canada $215.95)

This six-volume collection is the complete printed product documentation for Microsoft Visual C++™ version 4, the development system for Win32®. In book form, this information is portable and easy to access and browse, a comprehensive alternative to the substantial online help system in Visual C++. Although the volumes are numbered as a set, you have the convenience and savings of buying only the volumes you need, when you need them.

- **Vol. 1: Microsoft Visual C++ User's Guide** ISBN 1-55615-915-3, U.S.A. $29.95 (U.K. £27.49; Canada $39.95)

This four-part tutorial provides detailed information about wizards, the Component Gallery, and the Microsoft Developer Studio with its integrated debugger and code browser—all essential instruments for building and using prebuilt applications in Visual C++.

- **Vol. 2: Microsoft Visual C++ Programming with MFC** ISBN 1-55615-921-8, U.S.A. $29.95 (U.K. £27.49; Canada $39.95)

This comprehensive tutorial gives you valuable information for programming with the Microsoft Foundation Class Library (MFC) and Microsoft Win32, plus details on building OLE Controls. You'll find out how MFC works, with an in-depth overview and a valuable compilation of over 300 articles on MFC programming. Win32 topics cover exception handling, templates, DLLs, and multithreading, with a Visual C++ perspective.

- **Vol. 3: Microsoft Foundation Class Library Reference, Part 1** ISBN 1-55615-922-6, U.S.A. $29.95 (U.K. £27.49; Canada $39.95)
- **Vol. 4: Microsoft Foundation Class Library Reference, Part 2** ISBN 1-55615-923-4, U.S.A. $29.95 (U.K. £27.49; Canada $39.95)

This two-volume reference is your Rosetta stone to Visual C++, providing a thorough introduction to MFC, a class library overview, and the alphabetical listing of all the classes used in MFC. In-depth class descriptions summarize members by category and list member functions, operators, and data members. Entries for member functions include return values, parameters, related classes, important comments, and source code examples. Valuable information about macros and globals, structures, styles, callbacks, and message maps is included.

- **Vol. 5: Microsoft Visual C++ Run-Time Library Reference** ISBN 1-55615-924-2, U.S.A. $29.95 (U.K. £27.49; Canada $39.95)

This volume contains complete descriptions and alphabetical listings of all the functions and parameters in both the run-time and iostream class libraries–with helpful source code examples–and full details on the 27 new debug run-time functions.

- **Vol. 6: Microsoft Visual C/C++ Language Reference** ISBN 1-55615-925-0, U.S.A. $27.95 (U.K. £25.99; Canada $37.95)

The C and C++ references in this volume guide you through the two languages: terminologies and concepts, programming structures, functions, declarations, and expressions. The C++ section also covers Run-Time Type Information (RTTI) and Namespaces, important new language features added to this version of Visual C++. The final section discusses the preprocessor and translation phases, which are integral to C and C++ programming, and includes an alphabetical listing of preprocessor directives.

Microsoft Press® books are available wherever quality books are sold and through CompuServe's Electronic Mall—GO MSP or our Web Page, http://www.microsoft.com/products/mspress. Call 1-800-MSPRESS for more information or to place a credit card order.*
Please refer to BBK when placing your order. Prices subject to change.

*In Canada, contact Macmillan Canada, Attn: Microsoft Press Dept., 164 Commander Blvd., Agincourt, Ontario, Canada M1S 3C7, or call 1-800-667-1115. Outside the U.S. and Canada, write to International Coordinator, Microsoft Press, One Microsoft Way, Redmond, WA 98052-6399, or fax +1-206-936-7329.

IMPORTANT—READ CAREFULLY BEFORE OPENING SOFTWARE PACKET(S). By opening the sealed packet(s) containing the software, you indicate your acceptance of the following Microsoft License Agreement.

MICROSOFT LICENSE AGREEMENT

(Book Companion CD)

This is a legal agreement between you (either an individual or an entity) and Microsoft Corporation. By opening the sealed software packet(s) you are agreeing to be bound by the terms of this agreement. If you do not agree to the terms of this agreement, promptly return the unopened software packet(s) and any accompanying written materials to the place you obtained them for a full refund.

MICROSOFT SOFTWARE LICENSE

1. GRANT OF LICENSE. Microsoft grants to you the right to use one copy of the Microsoft software program included with this book (the "SOFTWARE") on a single terminal connected to a single computer. The SOFTWARE is in "use" on a computer when it is loaded into the temporary memory (i.e., RAM) or installed into the permanent memory (e.g., hard disk, CD-ROM, or other storage device) of that computer. 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. You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or disassemble the SOFTWARE. If the SOFTWARE is an update or has been updated, any transfer must include the most recent update and all prior versions.

4. DUAL MEDIA SOFTWARE. If the SOFTWARE package contains both 3.5" and 5.25" disks, then you may use only the disks appropriate for your single-user computer. You may not use the other disks on another computer or loan, rent, lease, or transfer them to another user except as part of the permanent transfer (as provided above) of all SOFTWARE and written materials.

5. SAMPLE CODE. If the SOFTWARE includes Sample Code, then Microsoft grants you a royalty-free right to reproduce and distribute the sample code of the SOFTWARE provided that you: (a) distribute the sample code only in conjunction with and as a part of your software product; (b) do not use Microsoft's or its authors' names, logos, or trademarks to market your software product; (c) include the copyright notice that appears on the SOFTWARE on your product label and as a part of the sign-on message for your software product; and (d) agree to indemnify, hold harmless, and defend Microsoft and its authors from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of your software product.

DISCLAIMER OF WARRANTY

The SOFTWARE (including instructions for its use) is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT FURTHER DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION REMAINS WITH YOU.

IN NO EVENT SHALL MICROSOFT, ITS AUTHORS, OR ANYONE ELSE INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE SOFTWARE 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 OF OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/COUNTRIES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

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 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

If you acquired this product in the United States, this Agreement is governed by the laws of the State of Washington.

Should you have any questions concerning this Agreement, or if you desire to contact Microsoft Press for any reason, please write: Microsoft Press, One Microsoft Way, Redmond, WA 98052-6399.

097-000-680

Microsoft® Windows NT™ Resource Kit

Version 3.51 Update 2

Essential, new information exclusively for owners of the Microsoft Windows NT Resource Kit, Version 3.51

CD INCLUDED

New utilities on the VERSION 3.51 UPDATE 2 CD include:

- TextViewer—enables you to quickly search and view text files within multiple subdirectories on local or shared drives
- FTEDIT.EXE—helps you recover fault-tolerant volumes

The VERSION 3.51 UPDATE 2 book presents the latest detailed information on topics that are either new for version 3.51 or reflect issues the Microsoft Product Support staff consider timely and important.

This update provides owners of the *Microsoft Windows NT Resource Kit,* version 3.51, with additional must-have information and tools. It covers new, as well as old, features in Windows NT version 3.51 and updates and corrects the five volumes in the *Resource Kit.* It also contains an updated version of the *Resource Kit* CD, which includes many new utilities, technical updates of existing utilities, and support for the PowerPC.™

VERSION 3.51 UPDATE 2 adds new, in-depth technical information that covers:

- Getting your organization up and running on Windows NT—creating a rollout plan, testing the rollout plan, upgrading a small group in preparation for an organization-wide upgrade (working out the kinks), implementing the final rollout, automating the deployment process
- Using Windows NT—determining whether to use the Windows NT File System (NTFS) or the file allocation table (FAT) or both, using NTFS compression, moving and copying compressed files, understanding fault tolerance and creating a fault-tolerant disk configuration, recovering disk information after a hardware failure, printing with Windows NT, troubleshooting printing problems

Four appendixes cover "Major Revisions to Windows NT Update 1," "Minor Revisions to Existing Resource Kit Books," "RAS Reference"—an overview of the most important modem compatibility standards—and "RFCs and Port Reference for Microsoft® TCP/IP"—how to set up the software so it recognizes the hardware.

To use this update effectively, you must have the five-volume *Microsoft Windows NT Resource Kit* for version 3.51. If you have the four-volume *Microsoft Windows NT Resource Kit* for version 3.5, look for *Version 3.51 Update* (volume 5), which corrects and updates the four main volumes of the *Microsoft Windows NT Resource Kit* and gives owners of the *Microsoft Windows NT Resource Kit* for version 3.5 the information and tools needed to update their kits to version 3.51.

7 90145 52564 2

U.S.A. **$39.95**
U.K. £37.49 [V.A.T. included]
Canada $54.95
[*Recommended*]

ISBN 1-57231-256-4

90000

9 781572 312562

Microsoft Press

Operating Systems/Windows NT