



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

This site is brought to you by
EarthWeb



[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[KEYWORD SEARCH](#)



Mastering Windows 2000 Registry
by Peter D. Hipson
Sybex, Inc.
ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)

[Advanced Search](#)

[PUBLICATION LOOKUP](#)

[JUMP TO TOPIC](#)



Desktop Library

[Click to access!](#)

[Introduction](#)

[PART I—Registry Basics](#)

[CHAPTER 1—What Is a Registry— and Why?](#)

[Organization](#)

[Hives and Their Aliases](#)

[Data Values](#)

[How the Registry Is Used](#)

[A Note on Terminology](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 2—Readme.1st: Preventing Disaster!](#)

[What's the Deal with the Registry, Anyway?](#)

[Where Exactly *Is* the Registry?](#)

[Are Two Copies Better Than One?](#)

[Backup Techniques](#)

[Back Up to Tape or Other Media](#)

[Backing Up Using copy or xcopy](#)

[What's on My ERD?](#)

[Using RegEdit to Back Up the Registry](#)

[Restoring the Registry](#)

[Restoring from Tape](#)

[Restoring from Other Media Supported by Backup](#)

[Recovering a Copied Registry](#)

[ERD Strikes Again: Using Setup to Recover](#)

[Loading a .reg file](#)

[The Recovery Console](#)

[Installing the Recovery Console](#)

[What's in the Recovery Console?](#)

[Using the Recovery Console](#)

[Starting the Recovery Console from the Installation CD-ROM](#)

[Recovery Console Commands and Options](#)

[Other Backup and Restore Programs](#)

[Hints and Kinks from the Experts](#)

CHAPTER 3—Anatomy of the Registry: The Blood, Gore, and Guts

[Of Hives and Bees—A Registry Overview](#)

[The Registry Hives](#)

[Hives, Keys, and Values](#)

[Registry Key Data Types](#)

[HKEY_LOCAL_MACHINE: The Machine's Configuration](#)

[HKEY_LOCAL_MACHINE\Hardware: The Installed Hardware Key](#)

[HKEY_LOCAL_MACHINE\SAM: The Security Access Manager](#)

[HKEY_LOCAL_MACHINE\Security: The Windows 2000 Security Manager](#)

[HKEY_LOCAL_MACHINE\Software: The Installed Software Information Manager](#)

[HKEY_LOCAL_MACHINE\System: The System Information Manager](#)

[HKEY_USERS: Settings for Users](#)

[HKEY_CURRENT_CONFIG: The Current Configuration Settings](#)

[HKEY_PERFORMANCE_DATA: The Performance Monitor Settings](#)

[NTUSER: The New User Profile](#)

[Hints and Kinks from the Experts](#)

CHAPTER 4—Registry Tools and Tips: Getting the Work Done

[RegEdit](#)

[Using RegEdit](#)

[Using RegEdit from the Command Line](#)

[Tips for RegEdit Users](#)

[Installing Remote Registry Editing on Windows 95 and Windows 98](#)

[RegEdt32](#)

[Using RegEdt32](#)

[Tips for RegEdt32 Users](#)

[Backup's Emergency Repair Disk Features](#)

[Creating an Emergency Repair Disk](#)

[Saving to the repair Directory](#)

[The Windows 2000 Resource Kit](#)

[Reg.exe](#)

[Hints and Kinks from the Experts](#)

CHAPTER 5—Policies, Good for One, Good for All

[An Introduction to Policies](#)

[Just More Confusion?](#)

[Software Settings](#)

[Windows Settings](#)

[Administrative Templates](#)

[The Microsoft System Policies for Windows 2000](#)

[System Policy Editor](#)

[Microsoft Management Console \(MMC\)](#)

[Active Directory Users and Computers](#)

[The Microsoft System Policy Editor for Windows NT 4](#)

[Hints and Kinks from the Experts](#)

[PART II—Advanced Registry Stuff](#)

[CHAPTER 6—Associations, Linkages, and OLE: How Confusing Can This Get?](#)

[Introduction to OLE](#)

[The Clipboard](#)

[DDE](#)

[What Is OLE?](#)

[OLE Controls, a.k.a. ActiveX](#)

[Client-Server OLE Applications](#)

[OLE and the Registry](#)

[How Linkages Are Made between Applications](#)

[Embedded Documents](#)

[Fixing an OLE Disaster](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 7—Why, Oh Why, Are There System.ini and Win.ini Files?](#)

[System.ini](#)

[Win.ini](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 8—Getting Rid of the Unwanted](#)

[RegClean](#)

[Running RegClean](#)

[Undoing RegClean](#)

[RegMaid](#)

[CLSID View](#)

[ProgId View](#)

[TypeLib View](#)

[Interface View](#)

[Recommendations for RegMaid](#)

[CleanReg](#)

[Hints and Kinks from the Experts](#)

CHAPTER 9—Recovering from Disaster, or Making the Best of a Bad Situation

To Repair or to Replace?

Stabilizing the System

Step 1: Can You Boot into a Spare Operating System?

Step 2: Can You Boot the System in Normal Mode?

Step 3: Does the System Run without Crashing?

Step 4: Do the System Components and Subsystems Run OK?

Step 5: Do Installed Applications Run OK?

Step 6: Is the System Generally Stable?

Step 7: Then What Is the Problem?

Analysis

Fixing Things Up

Possible Problems, Quick Fixes

An Application or a System Component Was Deleted

Another Application Has Overwritten an Application's Files

There Is an Error Reading the Application's Files

There Is an Error Reading the Drive

Manually Removing Registry Entries

Finding Entries

Removing Entries

Hints and Kinks from the Experts

CHAPTER 10—Programming and the Registry: A Developer's Paradise?

Windows 2000 Registry API Functions

To Use MFC or Not to Use MFC? That Is the Question

Hints and Kinks from the Experts

CHAPTER 11—The Performance Monitor Meets the Registry

PerfMon1: A Program to Access HKEY_PERFORMANCE_DATA

An Example Portion of PerfMon1 Output

PerfMon1's Performance Counters

Adding Performance Data to the Registry

Hints and Kinks from the Experts

PART III—Windows and Office Registry Entries

CHAPTER 12—The Windows 2000 User Interface: Changing How It Looks

Desktop Settings

Backgrounds and Wallpapers

Task Switching

Moving Windows

[The Cursor](#)

[Menus and the Windows 95 User Interface](#)

[Keyboard Settings](#)

[The Mouse and the Microsoft IntelliMouse](#)

[Other User Interface Settings](#)

[Displaying Your Favorites](#)

[Customizing the Properties Pop-up Menu](#)

[Custom Icons on the Desktop](#)

[Enhancing the Start Menu](#)

[Sounds Microsoft Never Gave Us](#)

[Make .dll Files Show Their Own Icons](#)

[Easter Egg Hunt?](#)

[Console and Command-Prompt Settings](#)

[Foreground and Background Colors](#)

[Memory Used by Command-Prompt Windows](#)

[Cursors](#)

[Fonts](#)

[What the Window Looks Like](#)

[The Command History Buffer](#)

[Miscellaneous Settings](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 13—Networking and Registry System Entries](#)

[System Entries](#)

[Networking Entries](#)

[Persistent Connections](#)

[RestoreConnection](#)

[OptionalNames](#)

[Improving Network Performance](#)

[Disk, Directory, and Related Entries](#)

[Moving Windows to a New Directory](#)

[Upgrade Blues](#)

[Where Was Windows 2000 Installed From?](#)

[I'm Full, Burp](#)

[Why Is Windows Asking for a Disk in the Drive?](#)

[Removing Context Menu Items](#)

[Using More Than Two IDE Controllers](#)

[Saving Share Information](#)

[Other Hardware Support Entries](#)

[Serial Ports](#)

[Printer \(Parallel\) Ports](#)

[Other Software Configuration Entries](#)

[Password Expires in *n* Days](#)

[Domain Refresh Interval](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 14—Microsoft Office Entries](#)

[Microsoft Office Shared Components](#)

[Changes Made by Microsoft Office Setup](#)

[The .reg Files](#)

[Microsoft Office System Configuration Information](#)

[The Access Entries](#)

[The Common Entries](#)

[The Excel Entries](#)

[The Outlook Entries](#)

[The PowerPoint Entries](#)

[The Publisher Entries](#)

[The Registration Entries](#)

[The Word Entries](#)

[Microsoft Office User Configuration Information](#)

[Access User Configurations](#)

[Binder User Configurations](#)

[Common User Configurations](#)

[Draw User Configurations](#)

[Excel User Configurations](#)

[Graph User Configurations](#)

[Outlook User Configurations](#)

[PowerPoint User Configurations](#)

[Publisher User Configurations](#)

[Query User Configurations](#)

[Word User Configurations](#)

[Using the Registry from Microsoft Office Applications](#)

[Hints and Kinks from the Experts](#)

[PART IV—The Registry Reference](#)

[CHAPTER 15—Introduction to HKEY_CLASSES_ROOT](#)

[GUIDs, UUIDs, and Other Funny Numbers in Windows](#)

[HKEY_CLASSES_ROOT](#)

[Managing File Types and File Extensions](#)

[Managing OLE and Embedding](#)

[Delving into the Unknown](#)

[Hints and Kinks from the Experts](#)

[CHAPTER 16—Introduction to HKEY_CURRENT_USER and HKEY_USERS](#)

[HKEY_CURRENT_USER](#)

[AppEvents](#)

[Console](#)

[Control Panel](#)

[Environment](#)

[EUDC](#)

[Identities](#)

[Keyboard Layout](#)

[Network](#)

[Printers](#)

[RemoteAccess](#)

[Software](#)

[Unicode Program Groups](#)

[Windows 3.1 Migrations Status](#)

[Volatile Environment](#)

[HKEY_USERS](#)

[Hints and Kinks from the Experts](#)

CHAPTER 17—Introduction to HKEY_LOCAL_MACHINE

[HKEY_LOCAL_MACHINE](#)

[Hardware](#)

[SAM](#)

[Security](#)

[Software](#)

[System](#)

[Hints and Kinks from the Experts](#)

CHAPTER 18—Introduction to HKEY_LOCAL_MACHINE\Software

[Classes](#)

[Clients](#)

[Microsoft](#)

[Windows](#)

[Windows Messaging Subsystem](#)

[Windows NT](#)

[ODBC](#)

[Policies](#)

[Program Groups](#)

[Secure](#)

[Voice](#)

[Windows 3.1 Migration Status](#)

[Hints and Kinks from the Experts](#)

CHAPTER 19—Introduction to HKEY_LOCAL_MACHINE\System and HKEY_CURRENT_CONFIG

[CurrentControlSet](#)

[Control](#)

[Enum](#)

[Hardware Profiles](#)

[Services](#)

[ControlSet001](#)

[ControlSet002](#)

[DISK](#)

[Mounted Devices](#)

[Select](#)

[Setup](#)

[HKEY_CURRENT_CONFIG](#)

[HKEY_CURRENT_CONFIG\Software](#)

[HKEY_CURRENT_CONFIG\System](#)

[CurrentControlSet](#)

[Hints and Kinks from the Experts](#)

[Appendix A](#)

[Appendix B](#)

[Appendix C](#)

[Appendix D](#)

[Appendix E](#)

[Appendix F](#)

[Index](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

ACKNOWLEDGMENTS

An acknowledgments section is always hard to write; there are just so many people who have helped. An author's greatest fear is forgetting someone, so I always start off by saying thanks to everyone. If I didn't list you, please don't hate me!

Thanks go to Tracy Brown and Ellen Dendy, of course, who served as acquisitions and developmental editors for this book. Tracy's help and advice were instrumental in making this a good book. Ellen Dendy also helped greatly as our second developmental editor. (Of course, if you don't like this book, the blame falls on me and only me!)

Thanks to the Sybex editorial staff, especially Bonnie Bills (who really is an independent editor). Thanks also to Susan Berge, project editor, for her skillful work and management; to Robin Kibby, Nila Nichols, Grey Magauran, and Franz Baumhackl, electronic publishing specialists, for their expert and speedy layout skills; to Shannon Murphy, project team leader, for her management of the book's production; and to David Nash, Jennifer Campbell, Patrick J. Peterson, and Nelson Kim, proofreaders, for their proficient proofreading of the pages.

John Savill (SavillTech Ltd.) served well as our technical editor. It was John's job to make sure that I told no lies, made no mistakes. John also provided some of our tips from the experts. John's Web site at <http://www.ntfaq.com/> is a major stopping point for users of Windows 2000; check it out.

Jerold Schulman (JSI, Inc.) maintains the Web page at <http://www.jsiinc.com/reghack.htm>. He provided a lot of expert hints for this book. If you need assistance with your Windows 2000 installation, check out Jerold's Web pages for his tips, tricks, and registry hacks.

Special thanks to Laura Belt at Adler & Robin Books. Laura is the person who makes this a business and not a hobby.

Thanks to Barry and Marcia Press for their input on the book's contents. Barry asked for a number of things to be covered, and I've covered as many as I could.

Thanks to ClubWin (and my team members who put up with my slow responses), and everyone at Microsoft who helped, too.

Of course, I would be remiss if I didn't thank my family, especially my wife, Nang, who has supported me through thick and thin, and the folks at CMC and MCH who made sure that I survived the experience.



INTRODUCTION

The registry has evoked emotions from terror to mystery. Few Windows 2000 users consider the registry their friend. After all, think of it: the registry is the heart and soul of the Windows 2000 operating system. The registry is everything—it is the brain of the operating system. Damage the registry, and Windows 2000 quickly develops brain damage and needs major surgery.

This is it—the only book on the Windows 2000 registry that you will need. Now, I won't kid you; there are a few other books on the Windows registry. However, most books try to cover the Windows 2000/NT 4 and Windows 95/98 registries together. Though these Microsoft operating systems all share a very similar registry structure, sufficient differences between them make it difficult for one book to cover everything well.

Will you need another book or tool besides this book? Maybe not. But I do recommend that you get Microsoft's Windows 2000 Resource Kit, too; it has a lot of good utilities that you will find invaluable. The Windows 2000 Resource Kit also has a lot of good non-registry stuff.

This book covers the Windows 2000 registry from A to Z. I've covered the standard stuff, from things that most of us should know to things that are not documented at all and are probably only known by a very few first-rate system administrators.

Who Is This Book For?

This book is valuable to all Windows 2000 users. Even users of Windows NT 4 and Windows 95/98 may find good information in this book, though it is primarily oriented toward Windows 2000.

This book is intended for:

- General users who use Windows 2000 Server or Professional at their desks and are responsible for their own computer(s). Typically, these users don't have responsibility for other users' computers, though they may help their friends out from time to time.
- System administrators who are responsible for an organization's computers (and perhaps thousands of Windows 2000 installations). Administrators will be presented with virtually every conceivable problem over a given period of time. Whatever can go wrong will; Murphy's Law is applied double to system administrators.
- Help desk staff who support users, even if they don't usually administer the system. Help desk staff roam throughout the organization, providing help and assistance as needed. All help desk people are going to find this book very useful.

If you are a user who wants to get the most out of your Windows 2000 installation (either Professional, Server, or Advanced Server), this book is a very good starting point. Think of it this way: If you are a system administrator, this book is one of the tools that you will need to manage and administer your Windows 2000 network. Manning the help desk? If so, having this book close at hand can save you lots of time and effort.

Overview of the Contents

This book is made up of four major sections. Part I, "Registry Basics," discusses ways to avoid problems, do backups, and restore the registry, and it covers some of the tools that are used with the registry. The first chapter, "What Is a Registry—and Why?," introduces the registry. You'll learn about the registry's major sections, called hives. This chapter also tells you about the registry's history.

TIP:

The fastest way to access the registry is to use `RegEdit.exe`, which comes with Windows 2000. To access `RegEdit.exe`, simply click the Start button, then click Run. Type **RegEdit** in the dialog box and press Enter. The `RegEdit` window will appear.

Chapter 2 is called "Readme.1st: Preventing Disaster!" It jumps right into one of the most important topics in this book: how to avoid getting into trouble. Most Windows 2000 disasters are registry related, and they are also preventable. Registry problems often arise because we don't have a good backup of the registry, and something comes along and damages it. Once damaged, the registry can be very difficult to recover.

Chapter 3, “Anatomy of the Registry: The Blood, Gore, and Guts,” is an in-depth analysis of what’s in the registry. Each major hive is covered in detail. We’ll discuss the way the hives relate to each other, along with how Windows 2000 manages users in the registry.

Tools, tools, and more tools. Chapter 4, “Registry Tools and Tips: Getting the Work Done,” takes a close look at the registry tools that are included with Windows 2000. The registry editors are covered, as well as the Backup utility and the registry software that is included in the Windows 2000 Resource Kit.

In Chapter 5, “Policies, Good for One, Good for All,” you learn all about policies in Windows 2000. Policies affect specific computers, users, and groups.

Part II, “Advanced Registry Stuff,” covers OLE (object linking and embedding), some history of the Win.ini and System.ini files, removing excess baggage from the registry, registry programming interfaces, and the Performance monitor entries. Getting into the advanced stuff, we jump right into the issues of OLE, associations, and such. Chapter 6 is called “Associations, Linkages, and OLE: How Confusing Can This Get?” It tries to clear the often muddy water that swirls around the OLE registry components. A major part of the registry is OLE related, with Windows 2000 using OLE to manage much of the user interface.

Even though the System.ini and Win.ini files have not been used for some time, we still have them. Chapter 7 is called “Why, Oh Why, Are There System.ini and Win.ini Files?” Here we delve into why these two files are still found under Windows and what makes them necessary.

If you want to get rid of that memo from your boss telling you that your project is due, you toss it into the trash can. Something in the registry that is not needed can be more difficult to get rid of. Chapter 8, “Getting Rid of the Unwanted,” introduces the problem of registry clutter and describes some very useful tools to clean up this excess.

By following the advice in Chapter 9, “Recovering from Disaster, or Making the Best of a Bad Situation,” you can make sure that disaster doesn’t strike. However, sometimes disaster just happens. Recovery, whether from backups or from manually cleaning the registry, is vital.

My name’s Peter, and I’m a programmer. Ah, there I said it, and I feel much better. I felt even better after writing Chapter 10, “Programming and the Registry: A Developer’s Paradise?” This is where the programming interface to the registry is unveiled. Examples in C/C++ and a lot of information about Microsoft’s MFC registry interface come to light in this chapter.

The Windows 2000 Performance monitor allows analysis of the system’s performance and the development of performance-enhancement strategies. In Chapter 11, “The Performance Monitor Meets the Registry,” you begin to understand how the Windows 2000 Performance monitor interacts with the registry and how you can add performance-monitoring technologies to your own applications.

Part III, “Windows and Office Registry Entries,” discusses the UI (user interface), networking, and internal Windows 2000 entries. What we see as users is all stored in the registry. Chapter 12, “The Windows 2000 User Interface: Changing How It Looks,” delves into the various registry entries that control the look and feel of Windows 2000. This chapter covers both the graphical Desktop and the Windows command windows.

Under the hood of Windows 2000 are entries in the registry for both networking and other internal Windows 2000 components. Chapter 13, “Networking and Registry System Entries,” digs into these less visible entries in the registry and explains them to you.

Chapter 14, “Microsoft Office Entries,” covers changes that Microsoft Office has made to the registry. Sometimes Microsoft Office components are installed and then removed. Sadly, not all registry entries for these products are removed. How do you get them out of there? Also, how do you create a configuration so those new users of Microsoft Office will get a predefined configuration? Care to program the registry using Visual Basic for Applications? (It’s easy, really.) Check this chapter for the answers to these questions.

Part IV, “The Registry Reference,” is a reference to many of the registry entries, arranged by hive. Program associations, OLE associations, and file-type management are all part of HKEY_CLASSES_ROOT. Chapter 15, “Introduction to HKEY_CLASSES_ROOT,” covers this hive’s contents.

User information that is stored in HKEY_USERS and used in HKEY_CURRENT_USER is the subject of Chapter 16, “Introduction to HKEY_CURRENT_USER and HKEY_USERS.” Windows 2000 keeps only the currently logged-on user and the .DEFAULT user in HKEY_USERS; other users are saved in HKEY_LOCAL_MACHINE’s SAM (Security Account Manager) sections.

HKEY_LOCAL_MACHINE is the hive that controls the system itself. This topic is so large that three chapters are dedicated to it. Chapter 17, "Introduction to HKEY_LOCAL_MACHINE," covers the major parts of HKEY_LOCAL_MACHINE. Information about installed software is found in Chapter 18, "Introduction to HKEY_LOCAL_MACHINE\Software." Virtually every installed application or component is found in HKEY_LOCAL_MACHINE\Software. The system configuration is covered in Chapter 19, "Introduction to HKEY_LOCAL_MACHINE\System and HKEY_CURRENT_CONFIG." System entries are critical to the health and welfare of Windows 2000.

Typesetting Conventions

This book is typeset so that it is readable. Otherwise the pages would all be blank.

OK, seriously. This book uses various conventions to present information. Notes, Tips, and Warnings, shown below, appear throughout the text in order to call attention to special details.



NOTE This is a Note. Notes contain additional comments and information related to the discussion.



TIP This is a Tip. Tips highlight important information that you need to know when working with the registry.



WARNING This is a Warning. Warnings call attention to trouble spots and things to watch out for. Speaking of which, have you backed up your registry lately?

This book also takes advantage of different font styles. **Bold font** in the text indicates something that the user types. A monospaced font is used for registry objects, program strings, entries, commands, and URLs.

To Contact the Author

If you so desire, you may contact me, the author, via e-mail. My e-mail address is phipson@acm.org. Please do not attempt to telephone, even if you find my phone number; my schedule really doesn't allow for answering the phone! If you have a question, and really must phone someone, call the publisher, as described below.

Sybex Technical Support

If you have questions or comments about this book or other Sybex books, you can contact Sybex directly. The following contact information for Sybex is listed in order of preference from the most preferred method to contact Sybex (email) to the least preferred method (snail mail).

For the Fastest Reply

E-mail us or visit the Sybex Web site! You can contact Sybex through the Web by visiting <http://www.sybex.com> and clicking Support. You may find the answer you're looking for on this site in the FAQ file, so check there too.

When you reach the support page, click Support@sybex.com to send Sybex an email. You can also email Sybex directly at Support@sybex.com.

It's important that you include all the following information to expedite a reply:

Name The complete title of the book in question.

ISBN number The ISBN that appears on the back cover of the book. This number appears at the bottom right corner on the back cover and looks like this:

0-7821-2615-4

Printing The printing of the book. You can find this near the front of the book at the bottom of the copyright page. You should see a line of numbers as in the following:

10 9 8 7 6 5 4 3 2

The lowest number in this line of numbers is the printing. The example here indicates that the book is from the second printing.



NOTE The ISBN number and printing are very important for technical support, because they indicate the edition and reprint you have in your hands. Many changes occur between printings and editions. Don't forget to include this information!

Page number or file name Include the page number where you have a problem.

PC details Include the following information:

- Name of your PC (the manufacturer)
- Operating system being used
- The software you have installed that relates to the book (indicate the exact version number)
- Whether your machine has any unique characteristics

Sybex technical support will try to answer your question quickly and accurately.

For a Fast Reply

Call Sybex Tech Support and leave a message. Sybex guarantees they will call you back within 24 hours, excluding weekends and holidays.

Tech support can be reached at (510) 523-8233, ext. 563.

After you dial the extension, press 1 to leave a message. Sybex will call you back within 24 hours with a reply. Make sure you leave a phone number where you can be reached!

Other Ways to Reach Sybex

The slowest way to contact Sybex is through the mail. If you do not have access to the Internet or a telephone, write Sybex a note and send it to the following address:

SYBEX Inc.
Attention: Technical Support
1151 Marina Village Parkway
Alameda, CA 94501

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

PART I

Registry Basics

Any sufficiently advanced technology is indistinguishable from magic.

—Arthur C. Clarke

CHAPTER 1

What Is a Registry— and Why?

FEATURING:

<i>Organization</i>	5
<i>How the registry is used</i>	8
<i>A note on terminology</i>	11
<i>Hints and kinks from the experts</i>	13

Some users of Windows NT know exactly what the registry is: a system designed to cause users and administrators to lose their hair. I know this is true because I can no longer feel the wind ruffling through my hair. Oh, I feel the wind; I just don't feel the hair.

The registry, like Windows, was evolutionary. The registry was preceded by a pair of flat-text files, called Win.ini and System.ini. These two files live on even today in Windows 2000, though they are virtually unchanged from Windows NT version 4. The first registry to appear in Windows was created to solve a number of problems: poor performance (retrieving information from the original flat-text .ini files was cumbersome), size limitations (the .ini files could be only so large), and maintenance problems (the .ini files were organizationally impaired!).

Today, the Windows 2000 .ini files contain only a few entries used by legacy 16-bit applications. They are of no importance to us, and we ignore them. It's the registry that is the most important system, because it contains the heart and soul of Windows 2000. Without the registry, Windows 2000 would be nothing more

than a collection of programs, unable to perform even the basic tasks that we expect from an operating system. Every bit of configuration information that Windows 2000 has is crammed into the registry. Information about the system's hardware, preferences, security, and users—everything that can be set is set in there.

The registry has a limit in size, too; it's not infinitely large. A message telling you that you are low on registry quota indicates that the registry has grown too large for the current size allocation. Unless you change it, the registry size is set to 25 percent of the paged pool size; for most computers, the paged pool size is approximately equal to the amount of installed RAM, up to a maximum of 192MB. The registry can be set to 80 percent of the paged pool size (80 percent of 192MB is just under 154MB, though good sense says to round down to 150MB).

Windows 2000 will adjust this size based on the currently installed RAM. There are several registry entries affecting registry size, though most users will find that the defaults are acceptable for their use. To create a very large registry, ensure that the amount of RAM installed is sufficient and set the `RegistrySizeLimit` and `PagedPoolSize` entries.



NOTE Microsoft limits the size of any object that is stored in a registry data key to 1MB. This limit is basically only meaningful for `REG_BINARY` objects, because strings and such are unlikely to become this large. If you find that you must store more than 1MB in a registry object, it will be necessary to store the information in a file and store a pointer to the file in the registry. Without this limitation, the registry could easily grow to be the largest file on your system.

Organization

The registry is organized into five major sections. These sections are called *hives*, which are analogous to root directories on your hard drive. Each hive, by definition, has its own storage location (a file) and log file. If necessary, a given hive could be restored without affecting the other hives in the registry.

Inside a hive you find both keys and subkeys (which are analogous to directories and subdirectories on your hard disk). A key may have information, or data, assigned to it (referred to as a *value entry*), making the key analogous to a file on your hard drive as well.

A key or subkey may have zero, one, or more value entries, a default value, and from zero to many subkeys. Each value entry has a name, data type, and a value:

- The entry's name is stored as a Unicode character string.
- The entry's type is stored as an integer index. The type is returned to the querying application, which must then map this type to the type that the application knows.
- The entry's value is stored as necessary to allow efficient retrieval of the data when needed.

Both the Windows 2000 operating system and applications store data in the Windows 2000 registry. This is both good and bad. It is good because the registry makes an efficient, common storage location. Here's the bad part: as more and more applications and systems store information in the registry, it grows larger, and larger, and larger.

It is most unusual for the registry to get smaller—I'm unaware of any application that does a really complete job of cleaning up all of its own registry entries when the application is uninstalled. Many applications leave tons of stuff in the registry when they are uninstalled, and not many applications clean up unused entries as a routine process. The end result is that the registry will grow, like Jack's magic beanstalk, as time goes on.



NOTE From time to time in this book, hives, keys, subkeys, and values will be referred to using the generic term *object*. When *object* is used, assume that the item could be any valid item in the registry!

Hives and Their Aliases

There are a number of hives in the Windows 2000 registry, and accepted abbreviations for each:

- `HKEY_CLASSES_ROOT`, a.k.a. `HKCR`
- `HKEY_CURRENT_USER`, a.k.a. `HKCU`
- `HKEY_LOCAL_MACHINE`, a.k.a. `HKLM`

- HKEY_USERS, a.k.a. HKU
- HKEY_CURRENT_CONFIG, a.k.a. HKCC



NOTE The HKEY_DYN_DATA hive, which has no abbreviation, disappeared in Windows 2000, though Microsoft had originally intended to include information about Plug and Play in this hive. So where is PnP data saved if the HKEY_DYN_DATA hive is gone? Windows 2000 supports PnP. Microsoft has decided to integrate PnP data with the main registry rather than have a separate hive.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

Each hive begins with HKEY_. HKEY is an abbreviation of “hive key,” though the significance of this is not terribly important in understanding the registry. The H also signifies that the name is a “handle” for a program to interface with the registry. These handles are defined in the file winreg.h, included with the Windows 2000 SDK (Software Development Kit).

The registry contains duplication—sort of. For example, you’ll notice that everything in HKEY_CURRENT_USER is also contained in the hive HKEY_USERS. But these aren’t two different sets of the same information; rather, they’re two names for the same set of information. Microsoft needed to make some parts of the registry appear to be in two places at one time. But they didn’t want to copy these sections, because that could have created problems with keeping each section current. Instead, they created an *alias*, or another name, for some registry components. The alias points to the original component and is updated whenever the original is. These aliases are created solely by Windows. You, as a user, can’t create an alias in the registry no matter how hard you try!

The most common alias is the registry hive HKEY_CURRENT_USER. It is an alias to either the .DEFAULT user in HKEY_USERS, or the current user in HKEY_USERS. If you take a quick peek at HKEY_USERS, you will see several keys there: one is .DEFAULT and the others are named with long strings of characters. These are SIDs (security identifiers), which Windows 2000 uses to identify users. One of these subkeys for the currently logged-on user consists of just the SID, while the other consists of the SID suffixed with *_Classes*. For example, on one Windows 2000 server, the administrator has the two subkeys HKEY_USERS\S-1-5-21-1004336348-842925246-1592369235-500 and HKEY_USERS\S-1-5-21-1004336348-842925246-1592369235-500_Classes. I’ll clear up what SIDs are and how they are used in Chapter 17.



NOTE The default user, used when no user is logged on, has only one subkey, named .DEFAULT.

There are also other aliases in the registry. For example, the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet is an alias to one of the other control sets—ControlSet001, ControlSet002, or sometimes ControlSet003. Again, this is that same magic; only one registry object is there, it just has two names. Remember, in modifying a specific registry key or subkey; don’t be surprised when another registry key or subkey seems to magically change also!

Data Values

A value may contain one or, in some instances, more than one data item. The only type of multiple-item

value entry that the registry editor can handle is REG_MULTI_SZ, which may contain zero, one, or more strings.

Data is stored in a number of different formats. Generally the system uses only a few simple formats, while applications, drivers, and so forth may use more complex types defined for a specific purpose. For example, REG_RESOURCE_LIST is a complex registry type used primarily by drivers. Though it would be inefficient, all registry data could be considered to be REG_BINARY data.

Data types for value entries include:

- REG_BINARY
- REG_COLOR_RGB
- REG_DWORD
- REG_DWORD_BIG_ENDIAN
- REG_DWORD_LITTLE_ENDIAN
- REG_EXPAND_SZ
- REG_FILE_NAME
- REG_FILE_TIME
- REG_FULL_RESOURCE_DESCRIPTOR
- REG_LINK
- REG_MULTI_SZ
- REG_NONE
- **REG_QWORD**
- **REG_QWORD_LITTLE_ENDIAN**
- REG_RESOURCE_LIST
- REG_RESOURCE_REQUIREMENTS_LIST
- REG_SZ
- REG_UNKNOWN



NOTE REG_QWORD is new to Windows 2000, and is a quad-word (64-bit) numeric entry; REG_QWORD_LITTLE_ENDIAN is the same as REG_QWORD.

Applications may access each of these data types. Additionally, some applications store data in formats that only they understand. Actually, there is a provision in the registry that allows the storing application to assign a specific type to the registry data. Any application or component that doesn't understand the format would simply treat the data as a REG_UNKNOWN type and read the data as binary.



NOTE Oops, did I say something special? Yes! Don't forget that applications can and do store data in the registry.

How the Registry Is Used

How does Windows 2000 use the registry? When is the registry first opened and used?

The registry is a tree-based hierarchical system that offers quick access to data stored in almost any format. Actually, the registry is a rather flexible database. Registry information comes from a number of sources:

- From installing Windows 2000
- From booting Windows 2000
- From applications, systems, and user interaction

Every component of Windows 2000 uses the registry, without exception. There is a set of APIs that are used to allow both Windows 2000 and other applications to access registry information easily and quickly.

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Windows 2000 starts to use the registry at the very beginning stages of system boot-up. The Windows 2000 boot process is based on which file format is installed, though the important parts are identical in either case. The unimportant parts are the loading of the specific drivers to read the NTFS file system.



NOTE Throughout this book, unless I say otherwise, I'm referring to Windows 2000 installed on an Intel x86 platform. There are differences in the boot process on RISC-based systems (such as the Digital Alpha system), though these differences are not terribly significant, considering how the registry is used.

The Windows 2000 boot process consists of the following steps:

1. The system is powered up, the video is initialized, and the hardware self-tests are performed. The BIOS performs these tests, which are called POSTs (power-on self-tests). Usually, the memory test is the most visible one; its progress is shown on most computer screens.
2. After running POST, the system will initialize each adapter. If the adapter has its own built-in BIOS, the adapter's BIOS will be called to perform its own initialization. Some adapters, such as Adaptec's SCSI adapters, will both display messages and allow the user to interact. Some adapters that don't have a BIOS won't be initialized until Windows 2000 loads their drivers much later in the boot-up process.
3. After all the adapters that have a BIOS have been initialized, the system boot loader reads in the sector located at the very beginning of the first bootable disk drive and passes commands to this code. This sector is called the *boot sector*, or the MBR (Master Boot Record), and it is written by the operating system when the operating system is installed.
4. The code in the MBR then loads the NTLDR file. (This file has no extension, though it is an executable file.) Once loaded, the MBR passes control to the code in NTLDR. When NTLDR is initialized, it displays the message "Windows NT Portable Boot Loader." Since our computers are so fast today, we never actually see this message—the screen is almost immediately cleared when NTLDR re-initializes the video system.
5. NTLDR then switches into 32-bit mode (remember an Intel x86 processor always boots into 16-bit real mode). It then loads a special copy of the necessary file system I/O files and reads in the file boot.ini.
6. The file boot.ini has information about each operating system that can be loaded. Remember, Windows 2000 supports multiboot configurations. It is trivial to create a Windows 2000 installation that can boot Windows NT, Windows 2000, and Windows 95 or Windows 98. The boot loader can even boot two different copies of Windows 2000 with either the same or different version numbers. NTLDR then processes the boot.ini file, displaying boot information to allow the user to select which


Desktop Library
 Click to access!

operating system will be loaded. At this point, let's assume that Windows 2000 will be loaded.

7. When you select Windows 2000 to be loaded, NTLDR will load the file NTDETECT.COM. This program then collects information about the currently installed hardware and saves this information for the registry. Most of this information is stored in the HKEY_LOCAL_MACHINE hive.
8. Once NTDETECT has detected the hardware, control is passed back to NTLDR and the boot process continues. At this point, the registry has been substantially updated with the current hardware configuration and stored in HKEY_LOCAL_MACHINE\Hardware.
9. The prompt to select the configuration is then presented. This prompt, "Press spacebar now to invoke Hardware Profile/Last Known Good menu," allows you to force Windows 2000 to use a specific configuration as stored in the registry hive HKEY_LOCAL_MACHINE.
10. Following the detection of NTDETECT, NTLDR will load and initialize the Windows NT kernel, load the services, and then start Windows.
11. When the kernel is loaded, the HAL is also loaded. (The HAL—Hardware Abstraction Layer—is used to manage hardware services.) Next, the registry system subkey HKEY_LOCAL_MACHINE\System is loaded into memory. Windows 2000 will scan the registry for all drivers with a start value of zero. This includes those drivers that should be loaded and initialized at boot time.
12. You can see the beginning of the next stage, kernel initialization. The screen will switch to a blue background, and you will see a message about the Windows 2000 build number and the number of system processors. Again, the system scans the registry and finds all drivers that must be started at the kernel initialization stage.
13. From this point, Windows 2000 starts various components and systems. Each component and system reads the registry and performs various tasks and functions. The final stage is to start the program that manages the user logon, WinLogon. WinLogon allows the user to log on and use Windows 2000.

Once Windows 2000 is booted, both the operating system and applications use the registry. The registry is dynamic, but usage of the registry may be dynamic or static. That is, some registry items are read one time and never reread until the system is restarted. Other items are read every time they are referenced. There is no fixed rule as to what is read each time it is needed and what is not, but to be on the safe side, follow these guidelines:

- Application-related data is probably read when the application starts. If you change application-based data, restart the application. In fact, the best path to follow is this: do not change application-based data while the application is running.
- User-interface data is sometimes dynamic, sometimes static. With user-interface data, the way to go is to change the data and wait to see the results of the change. If the change doesn't appear, try logging on again.
- System data is usually either static or otherwise buffered. Many system-related registry changes won't become effective until the system is restarted. Some system data is rewritten, or created, at start-up time, precluding changes by users. Many of the items in HKEY_LOCAL_MACHINE may be reset at system boot time, especially those items that are hardware related.

A Note on Terminology

The registry is made up of hives, keys, subkeys, and value entries. Well, actually, depending on the source, you may be faced with hives and data keys; or keys and items; or just data keys; or who knows what else.

There is some indication that Microsoft wants to drop the original term for a registry section—the *hive*—and replace this term with the word *key*. In the Windows NT Resource Kit, Microsoft makes the following definition:

The registry is divided into parts called *hives*. A hive is a discrete body of keys, subkeys, and values rooted at the top of the registry hierarchy. Hives are distinguished from other groups of keys in that they are permanent components of the registry; they are not created dynamically when the system starts and deleted when it stops. Thus, HKEY_LOCAL_MACHINE\Hardware, which is built dynamically by the Hardware Recognizer when Windows NT starts, is not a hive.

In the Windows 2000 documentation, Microsoft says a hive is:

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

A section of the registry that appears as a file on your hard disk...

These definitions are absolute and state exactly what is a hive and what is not. However, in the real world, no one follows this exact definition. Many authors call all holders of information *hives* (or *sub-hives*) and call data objects *keys*. Others never refer to hives at all, and instead call all holders *keys*, or *subkeys*, and refer to data objects as *values*.

Virtually every definition leaves something to be desired. To call the thing that holds data a “value entry” sometimes makes it awkward to refer to the contents. Consider these examples:

The value entry named asdf contains the value 1234.

The value called asdf contains the value 1234.

The following example is much more readable:

The value entry asdf is a REG_DWORD with a value of 1234.

Is there a need to distinguish between what Microsoft calls a “hive” (a top-level, permanent, registry component) and what Microsoft calls a “key”? When does a hive become a key, and is this important? I can’t think of any context in which anything is gained by making this distinction. Referring to the top-level objects as *hives* certainly frees up the term *key* to be used elsewhere, but why not stick to one term?

Table 1.1 compares registry terminology against the terminology used for the Windows file system—and gives the terminology I’ll be using in this book.

TABLE 1.1: REGISTRY TERMINOLOGY EXPLAINED

Context	Root Collections	Subcollections	Objects	Data
Disks	Root directories	Direcatoriers/subdirectories	Files	Data
Older registry terminology	Hives	Sub-hives	Data keys	Data
Newer registry terminology	Hives	Keys/subkeys	Value entry	Data
Registry termonology in this book	Hives	Keys/subkeys*	Value entry	Data

*Just to keep things easy to read, I’ll use the term *key* to refer to both keys and subkeys.



Desktop Library
Click to access!

Hints and Kinks from the Experts

In each chapter, I'll present a few hints and kinks from the experts. These experts are a number of people who have a lot of experience working with the Windows registry. They have learned from their experiences and the experiences of others.

For example, every expert will tell you the same thing: the minute you start tinkering with the registry, you will create a mess that is so bad that only a clean reinstall (or restoration from a backup) will fix it. To restore the backup, you would boot from the installation disks and choose Repair, insert the ERD, and restore the registry—not a full backup. (If you did want to restore a full backup, it would only work if you'd selected to back up System State data.)

The first time I had a serious registry problem, I'd change something, and things would just get worse. Some registry problems cannot be "hacked," or fixed manually. The only fix for these problems is to either reinstall or restore the system. However, this type of situation is unusual. My experience has been that these problems happen only when hardware (like the registry's drive) fails or an incredibly errant program totally trashes the registry. Neither of these happens very often at all.

Most users make minor tweaks or fixes in the registry. Most of the time, things go OK. Sometimes things go awry. Through it all, we toast the registry, and then it's back to the proverbial drawing board. Such is life.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 2

Readme.1st: Preventing Disaster!

FEATURING:

<i>What's the deal with the registry, anyway?</i>	17
<i>Are two copies better than one?</i>	21
<i>Backup techniques</i>	23
<i>Restoring the registry</i>	30
<i>The Recovery Console</i>	36
<i>Other backup and restore programs</i>	43
<i>Hints and kinks from the experts</i>	43

Preventing disaster is an important thing to do. No one wants a system failure or to have to reinstall Windows 2000.

You are reading this chapter for your own particular reason. Perhaps, as I am recommending, you are here because you want to do everything possible to prevent a disaster with your Windows 2000 installation. Or maybe you really, really want to recover from an existing disaster. If you are recovering from a problem, you may want to skip to the section later in this chapter titled "Restoring the Registry." For those of you who never do anything wrong, read on.

The registry has always been the one part of Windows that virtually every user has neither understood nor trusted. Just when things go well, the registry gets corrupted, and it is time to reinstall everything.

The Windows 2000 operating system is quite robust. However, many things can cause problems. For example, a hard drive failure (even a small soft error on the system drive in the registry files), a controller failure, or a more complex memory bit that sometimes doesn't set correctly all can cause many problems with Windows 2000 and the registry.





WARNING Windows 2000 is robust, but our hardware is not. Most Pentium systems do not have memory parity. Though earlier PC systems used memory parity, this feature disappeared quietly a few years back when memory prices skyrocketed and there was a serious effort to keep computer prices to a minimum. Most of the newest computers now do support parity for their memory; many of the systems still in use do not, and as a result, routine memory errors won't be detected until it is much too late.

In this chapter, we'll cover a number of potential problem areas:

Backup: You'll learn a number of ways to back up that pesky registry.

Restoration: What's difficult even under the best of conditions will be made simpler after you've perused these pages.

Recovery techniques: You'll discover ways to recover from a registry failure and retain as much of the existing installation of Windows 2000 as possible.

Hints and kinks from the experts: This is stuff from the Resource Kit and a few ideas from some experts on how to keep things going well.

What's the Deal with the Registry, Anyway?

One of the biggest problems with the registry is that Windows uses it constantly. The entire process of backing up and restoring the operating system is much more difficult because Windows must have the registry files open as a restore is being done.

There are several ways to solve this problem: One solution is to use the backup program supplied with Windows 2000. Another is to use an after-market backup program. Such a backup program has to contain the code necessary to do registry backups and restores.



TIP Oh, joy! The backup program that is included with Windows 2000 allows backing up to media other than tape drives. Now it is possible to back up to other hard drives (a technique that I use), Zip drives, and other storage media.



However, these techniques may not work well under your circumstances. You may already have had a registry failure, and there may be no registry backup to rely on for recovery. Backing up and recovering the registry without a tape backup is excruciatingly difficult using previous versions of the backup program.

Using the ERD (Emergency Repair Disk) is easy, but you cannot simply stick in a diskette, type restore registry, and expect it to work. Windows 2000 does not store any registry information on the ERD (Microsoft recognized that the registry was becoming too large to store on a typical diskette). The Windows 2000 ERD contains only three files: autoexec.nt, config.nt, and setup.log. The directory `%systemroot%\repair` (the same location in which they are stored in Windows NT 4) holds all the registry files that are backed up.

In fact, restoring the registry from the `%systemroot%\repair` directory requires the Windows 2000 installation program. It's not that bad; you don't have to reinstall Windows, but the installation program will restore the registry from the backup, if necessary.

The menu that is presented when you boot up Windows 2000 also allows a user to restore parts of the registry based on copies of the registry saved from previous sessions.



TIP Always, always make sure that you back up the registry whenever you install new software or hardware or remove anything from your computer. If you do not back up the registry, and you restore a previous copy from an old backup, the system will not work as expected!

Where Exactly Is the Registry?

In order to back it up, you need to know where the registry is located. Sometimes you get to the registry as if by magic—the standard registry editors don't tell you where the registry is; they simply load it automatically. However, many times you need to know where to find the registry files. They're not too difficult to find; the registry's files are in the directory `%systemroot%\System32\config`.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**Environment Variables**

Every Windows 2000 installation automatically has some shortcut variables installed that are accessible to the user and the system. These variables are called *environment variables*. One environment variable, `%systemroot%`, contains the drive, path, and directory name for the directory that Windows 2000 was installed in.

Using these environment variables makes it easy to write batch files and to otherwise locate components of your current Windows 2000 installation. For example, you might type at a command prompt:

```
CD %systemroot%
```

This command would then change to the directory that Windows 2000 was installed in.

Using the environment variables also can be very useful when writing software that must be run on a number of different Windows 2000 installations, especially when these installations are made to different drives or directories.

The `%systemroot%\System32\config` directory includes the following set of files, each of which is a critical component of the registry. These files are backed up to the repair directory, so that they may be restored as necessary in the event of a registry failure.

Autoexec.nt: Used to initialize the MS-DOS environment unless a different start-up file is specified in an application's PIF.

Config.nt: Used to initialize the MS-DOS environment unless a different startup file is specified in an application's PIF.

Default: The default registry file.

SAM: The SAM (Security Accounts Manager) registry file.

Security: The security registry file.

Setup.log: The file that contains a record of all files that were installed with Windows 2000. Service packs and other components of Windows 2000 use the information in this file to update the operating system.

Software: The application software registry file.

System: The system registry file.

Two additional files are used to reconfigure security when the registry must be repaired. These are contained

only in the repair directory, and not in the %systemroot%\system32\config directory:

SecDC.inf: The default security settings that have been updated for domain controllers.

SecSetup.inf: The out-of-the-box default security settings.

In a typical Windows 2000 installation, the %systemroot%\system32\config directory contains these files:

AppEvent.Evt: The application(s) event log file.

DEF\$\$\$\$.DEL: The default registry recovery file.

Default: The default registry file.

Default.sav: A backup copy of the information contained in the default registry file.

DnsEvent.Evt: The DNS server event log.

File Rep.evt: One of two File Replication Service event log files.

Netlogon.dnb: A NetLogon support file.

Netlogon.dns: A NetLogon support file.

NTDS.Evt: The Windows 2000 directory service event log.

NtFrs.Evt: The second of two File Replication Service event log files.

SAM: The Security Accounts Manager registry file.

SecEvent.evt: The security event log.

Security: The security registry file.

SOF\$\$\$\$.DEL: The software registry recovery file.

Software: The application software registry file.

Software.sav: A backup copy of the information contained in the software registry file.

SY\$\$\$\$.DEL: The system registry recovery file.

SysEvent.evt: The system events log.

System: The system registry file.

SYSTEM.ALT: A copy of the information contained in the system registry file.

System.sav: A backup copy of the information contained in the system registry file.

Userdiff: Migrates preexisting user profiles from previous versions of Windows NT to Windows 2000.

In the registry, the most important files are those with no extensions—these are the current registry files. Another important file is SYSTEM.ALT, a duplicate of the System registry file.

The files in the %systemroot%\System32\config directory that have the extensions .log or .sav contain a history that may be viewed with the Event Viewer program. For example, files with the extension .sav were saved using the Last Known Good booting process. Files with the .log extension are records of changes made to the registry when registry auditing is turned on. Though the .log and .sav files are not strictly necessary to have a working Windows 2000 installation, it is best to consider each of these files a member of a complete set.



WARNING Be careful not to replace one file in the registry without replacing all the others. It is simply too easy to get one file out of sync with the remaining registry files, and this would spell disaster.

Side Trip: Restoring Windows 2000

Restoring a copy of Windows 2000 from a backup can be a difficult process. First, without a working copy of Windows 2000, you can't run the backup and restore programs. This means you have to install a new copy of the OS to be able to run the restore program. You'd then use this copy of Windows 2000 to restore the original system from the backup. Some users will reformat the drive, reinstall Windows 2000 into the same directory that the original installation was made to, and restore on top of this new installation. There's nothing wrong with doing this, as long as you remember one critical point: If you installed any Windows 2000 service packs on your original installation, these service packs must also be installed on the new installation being used to run the restoration program. If you don't install the service packs, Windows 2000 restores system files from the original installation (with the service pack) on top of the new files (without the service pack); the files will be out of version sync with the existing operating system files and the registry. This will usually cause the restore to crash without much of a warning as to what happened.

To perform a full restore of Windows 2000 (and everything else on the drive) do the following:

1. Reformat the drive. Remember that you're doing a full restore here, and nothing that was on the drive is considered valuable at this point.
2. Install Windows 2000, using your original distribution CD-ROM.
3. Install the service packs that were installed with the version of Windows that is being restored. Remember that the service packs are cumulative, so you need only reinstall the last service pack. For example, if Service Pack 3 was installed, it will not be necessary to install Service Packs 1 and 2. You only need to reinstall Service Pack 3.
4. Reinstall your backup/restore program, if necessary, and begin your restoration process.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Are Two Copies Better Than One?

Generally, two of anything is better than one. It's easier to ride a bicycle than a unicycle. However, it is even easier to drive a car—you don't even have to keep your balance. Where the registry is concerned, keeping *at least* two copies of it is a good idea. I'd recommend that you keep at least four:

- The copy created by the Windows 2000 backup program, stored in %systemroot%\repair. The Windows 2000 Setup program is able to use this copy to restore the registry.
- A backup copy of the registry files found in %systemroot%\repair, saved in a safe and convenient location. Consider a Zip disk or some other type of removable storage media for this copy.
- One (or more) backup copies, created using a backup technique on a type of media that is compatible with the backup and restore program of choice. (I'll discuss backup methods to use in the next section.)
- A copy of the registry files contained in %systemroot%\System32\config stored on separate media, such as a different drive, diskettes, a Zip drive, CD-RW, or some other easily accessible, writeable media. Try to avoid media requiring special drivers and such, because these drivers may not work when you need to restore that pesky registry. This copy may only be made by dual-booting into another copy of Windows 2000 (or Windows 95/98 if the drive is FAT compatible).



NOTE In Windows NT 4, keep the special copy created by the RDisk utility that is stored in the Windows NT directory %systemroot%\repair. This copy of the registry can only be used by the Windows NT Setup program to repair an existing copy of Windows NT. Also keep the copy created by the RDisk utility that is stored on the Windows NT ERD. Again, this copy of the registry can only be used by the Windows NT Setup program to repair an existing copy of Windows NT. Windows 2000 doesn't support RDisk. Instead, the registry backup and ERD-creation functionality is incorporated into the finally-useful-for-everyone Backup program.

Be absolutely sure you keep these copies secure. Lock 'em up, stash 'em away. Oh, and by the way, that lock on your desk drawer is not good enough; use a good fireproof safe or strong box.



JUMP TO TOPIC

Danger, Will Robinson, Danger!

Throughout this chapter and this book we talk about backing up the registry, saving the registry to diskettes, other drives, and tapes. That's all well and good. However, you must remember that the registry contains sensitive information, especially if it is for a Windows 2000 server.

The registry is the heart and soul of the Windows 2000 operating system. It contains information critical to both the operation and security of Windows 2000. There are many ways that someone could use your backup registry files to breach your system's security, perhaps costing you money or (gasp!) your job.

Be absolutely sure you maintain the highest levels of security for any copies of the registry that you make. If saved to external media (diskettes, tapes, or Zip drives, for example), make sure these copies are securely locked up. Why? Someone could, with little effort, completely subvert system security and then use the backup copies of the registry to hide their actions.

I recommend you use a quality fireproof safe or a strong box for storing your registry backup copies. Me, I use a fireproof, locked strong box inside a federal government-rated Mosler safe—and I don't think I'm being overly protective, either.

Backup Techniques

Windows 2000 supports two different file systems. The first file system, called FAT (File Allocation Table), is identical to the file system used with both DOS and Windows 95/98. The FAT file system is not secure and offers no resistance to hackers and others who want to access files improperly. There are several flavors of the FAT file system: FAT12, FAT16, and FAT32. Windows 2000 fully supports FAT32 and FAT16. This support allows compatibility with Windows 98's large disk support.



NOTE Windows NT 4 does not support FAT32 except in a very limited, read-only manner. You cannot install Windows NT 4 onto a FAT32 drive. FAT12 is antiquated and is unlikely to be found on Windows NT systems.

The second file system, NTFS (NT File System), is unique to Windows 2000. Though it is possible to read an NTFS drive from DOS or Windows 95 using shareware utilities, it is generally not possible to write to an NTFS drive unless you are using Windows 2000. However, System Internals (see their Internet site at <http://www.sysinternals.com>) has two utilities that allow you to write to an NTFS volume from DOS or Windows 95/98.

Back Up to Tape or Other Media

The Windows 2000 backup program, Backup (NTBackup.exe), is one of a whole slew of compatible backup programs that allow backing up the system registry to tape, diskettes, other hard drives, or for that matter, any Windows 2000-supported writeable media. The process is straightforward and can be done as part of a regular backup cycle, or whenever desired. Just check System State in the backup tree to back up using Backup. When it stores the System State, Backup saves the following items:

- Active Directory—the database of information about objects on the network
- Boot files—the files used to boot the system
- COM+ Class Registration database—the COM+ classes' registration
- Registry—the set of files that comprise the configuration of Windows 2000 SysVol—a shared directory that contains the server's public files that will be replicated on all other domain servers on the network



NOTE In Windows 2000, to create an ERD, you use the Backup program. In the Tools menu, simply select Create an Emergency Repair Disk. Backup will prompt for diskettes as needed.

Using Backup is simple if you are familiar with creating and restoring tape backups. However, there are a few difficulties in using backups of the registry. First, to keep the registry backup easily accessible, it would be wise to place the registry backup on its own media. If the media is inexpensive, this is a viable practice, but if you are paying an arm and a leg for media, this can be costly, because each registry backup is relatively small as far as backups go.

Second, the registry backups must be kept secure; perhaps more secure than standard backups. Everyone's situation is different; just realize that unrestricted access to the registry would allow someone to have unrestricted, unaudited access to everything else as well.

Finally, tape backups are sometimes slow. Stick the tape in the drive and the first thing that happens is that the tape gets rewound (to re-tension it). This process alone can take some time—time that is not available when you are working on getting a server up and running. Consider instead backing up the registry to a local hard drive (a drive other than the system drive, however).

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!
Backing Up Using copy or xcopy

It is not possible to copy back the current registry while Windows 2000 is using the registry. Period. Therefore, to restore the registry using either `copy` or `xcopy`, it is necessary to shut down Windows 2000 and start another operating system, such as DOS, Windows 95/98, or a second copy of Windows 2000. Which operating system you use depends on which file system is being used on the computer. If the file system is FAT, you should start DOS or Windows 95/98. If the file system is NTFS, you should start a second copy of Windows 2000.

Backing up the registry with `copy` or `xcopy` is easier than using Backup. Run the Backup program and create an ERD. Then simply copy the backup of the registry found in the `%systemroot%\repair` directory to another location. Then (this is optional, but can't hurt), `xcopy` the current registry files in the `%systemroot%\system32\config` directory using the `/c` option, which tells `xcopy` to ignore errors. (Since the currently opened registry cannot be copied, these files will generate an error.)

Backing Up If You're Using FAT

Those Windows 2000 users who are using the FAT file system are able to simply boot a DOS, or Windows 95/98 (if FAT32 is used), diskette that was formatted with the `/sys` option. This will give you a DOS command prompt that allows you to read from and write to the hard drive quite easily.

To create a bootable disk, simply use the Windows 95/98 or DOS `FORMAT` command with the `/s` system option. Then copy the `xcopy` command's files (`xcopy*.*`) to the diskette, too. This disk may then be booted in the Windows 2000 computer, allowing unrestricted accesses to all FAT drives that are installed on that computer. When using Zip drives, it may be necessary to add DOS drivers for these drives to your boot diskette.



NOTE If the system is already configured for dual-booting, you probably can use the second operating system instead of using a boot diskette. It probably won't matter which alternate OS is installed (DOS, Windows 95/98, or even Windows NT); all will work fine for the purpose of backing up the registry. There is no need for boot diskettes in this situation.

After booting into a command prompt, it is a simple task to copy the registry files to a safe location, such as another hard drive, a set of diskettes (the registry won't fit on a single diskette), a Zip drive, a CD-RW drive, or other supported media.



NOTE Some computers will allow booting from the CD-ROM drive. If this is the case for your computer, then it is also possible, if you have a CD-RW drive, to create a bootable CD.

Backing Up If You're Using NTFS

Users with NTFS are presented with a much more difficult problem. The NTFS file system is a secure file system that may not be easily accessed using other operating systems such as DOS or Windows 95/98. Files on an NTFS drive may only be written by Windows 2000 and not by other operating systems. Sure, there are utilities that allow NTFS to be accessed from Windows 95/98. However, the mode of access is read-only; there is no chance of a restore that way.

To be able to access the registry files on an NTFS drive, it is necessary to install a second copy of Windows 2000. Actually, this is not major problem because everyone should have at least two installations of Windows 2000. Windows 2000 supports multiple boot configurations quite effectively. To create a multiple boot installation of Windows 2000, simply follow these steps:

1. Ensure that you have sufficient space on your hard drive for a second copy of Windows 2000. Your second copy of Windows 2000 will only need to be the basic operating system—only a minimal amount of hard disk space will be required. Figure 100MB to 150MB of hard disk space for this installation, depending on how much additional software and features you install.
2. Using the Windows 2000 installation boot diskettes, begin your installation. When prompted for a destination, simply specify a new directory. If you are farsighted enough, and are doing this before disaster has struck, you can install directly from the distribution CD without using the boot diskettes. To do so, run the Windows 2000 Setup program to begin the installation process.



WARNING Don't install to the same directory that your current working installation of Windows 2000 is installed into. That won't create a second copy of Windows 2000.

3. The Windows 2000 Setup program will configure the Boot Manager (creating new entries in the boot menu) so that you are able to choose which copy of Windows 2000 you want to boot.

Customizing the Boot Menu

Once you install a second copy of Windows 2000, your boot menu will list both copies of Windows 2000. This can be confusing since the descriptions will be almost identical.

There is a solution: the boot menu can be customized. The boot drive's root directory contains a file called `boot.ini`. This file includes the boot options for each copy of Windows 2000 that is installed. Before you can edit `boot.ini`, you need to remove the system, read-only, and hidden attributes by going to a command prompt and typing `c:\> attrib c:\boot.ini -r -s -h`. Don't forget to restore these attributes afterwards.

The `boot.ini` file includes a text string that describes the installation:

```
type boot.ini
[boot loader]
timeout=30
default=signature(9e2ebb84)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
signature(9e2ebb84)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows
2000 Server" /fastdetect
multi(0)disk(0)rdisk(0)partition(1)\WINNTBU="Windows NT Server"
multi(0)disk(0)rdisk(0)partition(1)\WINNTBU="Windows NT Server"
/basevideo /sos
```

You can modify anything in the quoted strings. A suggestion is to call your backup installation of Windows 2000 just that—"Windows 2000 B/U." For example:

```
multi(0)disk(0)rdisk(0)partition(1)\WINNTBU="Windows 2000 Server Registry
B/U"
multi(0)disk(0)rdisk(0)partition(1)\WINNTBU="Windows 2000 Server Registry
B/U [VGA model]" /basevideo /sos
```

Don't forget to use the Control Panel's System applet to change the default boot to the version of Windows 2000 that normally will be booted by default. After Windows 2000 is reinstalled, the latest installation is made the default operating system by the installation (Setup) program.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
To copy or to xcopy, That Is the Question

Users of FAT file systems can access the registry with a DOS boot disk, and users of either FAT or NTFS may gain access with a second copy of Windows 2000 as described above. Once a method to access the registry has been established, it is a simple task to completely back up the registry.

Typically, I'll use a command window (a "DOS box," or command prompt), because I use NTFS and have a second copy of Windows 2000 installed. I'll show how I back up the registry on my Windows 2000 server.

Using the md (make directory) or mkdir command, I create a new directory called \RegBU on another drive (my system has five hard drives):

```
md d:\RegBu
xcopy g:\winnt\system32\config\*.* D:\RegBu\*.* /s
```

I then use the copy command (or xcopy) to copy the registry files in g:\winnt\system32\config directory to the RegBU directory. The directory winnt is where my main copy of Windows 2000 is installed.

This example would save a backup to a subdirectory on the D: drive. This is a good solution if the system (G:) drive becomes unreadable, because the backup copy will still be accessible. Other alternatives include backing up to a removable (Zip) drive or a network drive on a different computer.

If things are going well, I also use PKZIP to back up the registry files to a set of diskettes. In my system, the files in my config directory are just over 16MB in size. Am I typical? No. I only have a few users in my user database, so my registry is smaller than most. PKZIP is able to compress the files down to only two diskettes, which is a reasonable number. Of course, if I used a Zip drive, I could put these files on a single cartridge, but in my case that would be a waste of space.

Once you've copied your registry files to a safe location, simply remove the boot diskette (if used) and reboot the computer. This will give you a copy of the registry that is restorable later using an almost identical technique: boot to DOS and restore the files.



TIP What the heck is a safe location? A safe location typically would be to another drive, a Zip drive, or perhaps even diskettes. Diskettes present a small problem in that the registry files are typically going to be a total of 10 to 20MB in size. Using a utility such as PKZIP allows you to write these large files to a number of diskettes while at the same time compressing them, reducing the number of diskettes required to a minimum.

What's on My ERD?
Desktop Library
[Click to access!](#)

The Windows 2000 ERD contains a number of files in addition to the registry files. The files found on a typical ERD include:

Autoexec.nt: Not part of the registry, this file is saved on the ERD. It is used to initialize the MS-DOS environment.

Config.nt: Not part of the registry, this file is saved on the ERD. It is used to initialize the MS-DOS environment.

Setup.log: This file contains information about the initial setup of Windows 2000. It is a critical file when restoring the registry using the Setup program's repair function.

Using RegEdit to Back Up the Registry

Using the Windows 95/98 Registry Editor, you can make an additional copy of the registry and restore it by double-clicking a single icon. The Windows 95/98 Registry Editor, RegEdit, is included with Windows 2000.



NOTE If you are a system administrator and you have Windows 95/98 systems, the technique described below will work for these computers as well. Actually, they work better with Windows 95/98 than with Windows 2000, but we'll keep that our carefully guarded secret.

If you follow the steps outlined shortly, you can create a copy of the system registry that includes everything except the Security and SAM registry keys. When backing up a Windows 2000 workstation on a network, RegEdit will usually use this technique to save everything needed. There are other methods to back up the security database, though those methods are awkward and somewhat difficult to manage: it is easier to use the techniques described earlier in the chapter to do a complete registry backup. Because the Security and SAM keys are not backed up, this is not a complete backup technique. Rather, this is an interesting technique for backing up the other major parts of the registry—one that is very easy and quick to do.

To use RegEdit to back up the registry:

1. Run RegEdit. Either go to a command window and type the command **RegEdit**, or choose Start ☪ Run to open the Run dialog box, type **RegEdit** in the Open input area, and click the OK button.
2. After RegEdit starts, note that My Computer is highlighted. If My Computer is not highlighted, click it to highlight it. This ensures that the entire registry, not just part of it, is backed up.
3. Select the Registry menu item Export Registry File.
4. RegEdit will display the Export Registry File dialog box. Using the dialog box's toolbar, navigate to the Desktop, type a name or the file (for example, **Registry**) and click Save.
5. Exit RegEdit.



Notice that unlike earlier versions of RegEdit, the version that is supplied with Windows 2000 will write the registry file out in Unicode format (each character is two-bytes long). Editors and utilities that do not understand Unicode character sets will have difficulty working with this file. To convert a Unicode text file to one-byte text format, use the type command, with the output redirected to a new file. For example:

```
type " file in unicode.reg " >" file in text.txt"
```

This is easy and almost painless. Using this technique to back up the registry immediately after installation allows you to restore the system to a known state very easily and quickly. Simply double-click the file you created in step 4, above, and this file will be reloaded as the current registry.



NOTE The saved registry file may be placed anywhere you desire. In some cases, placing a registry restore capability on the user's Desktop is tantamount to courting disaster. Some users will click it just to see what will happen. One solution is to hide the file or save it to an offline storage location.

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Restoring the Registry

To restore the registry, you must consider how the registry was saved. There are four ways to save a registry, each of which differs in just how much of the registry was saved and where the registry was saved:

- You can use a backup program to copy the registry to a tape or other offline location. The backup program will then restore the registry backup to its original location.
- You can copy the registry (as described above), creating identical copies of the registry that can then be recopied back to the original registry locations. This requires that you use a second operating system (or a second copy of Windows 2000) to copy the files back.
- The Backup program saves the registry to the %systemroot%\repair directory. You can then use the Windows 2000 Setup program to restore these files.
- You can use RegEdit to save the registry in a text file with an extension of .reg. Windows 2000 knows that this is a registry file (because the .reg file type is a registered extension) and will reload the file automatically into the registry if the file is double-clicked in Explorer or from the Desktop. From a command prompt, enter the command **start filename.reg**, where *filename* is the name of the registry backup file.

Restoring from Tape

Restoring a tape backup is a simple though time-consuming process. When you use a backup and restore program compatible with Windows 2000, make sure that you select the option to restore the local registry. You will have to make the decision about restoring other files at this time based upon your circumstances. If you suspect that other system files may be corrupted, or if you are simply not sure of the state of the system, then I would recommend repairing Windows 2000, or restoring the entire operating system and the registry at the same time. If you know that the registry is the only damaged component, simply restoring the registry and not other system files may save some time.

Restoring from Other Media Supported by Backup

Restoring backups saved on other media (such as disks, diskettes, Zip drives, etc.) is a simple and often fast process. Use the Windows 2000 Backup program, and select System State from the list of backed up items to restore.



NOTE It is not possible to restore only part of the System State data; you must restore it all.



Click to access!

If your backup includes additional files in addition to the System State, you may restore those files at the same time. You will have to make the decision about restoring these other files based on your circumstances. If you suspect that other system files may be corrupted, or if you are simply not sure of the state of the system, then I would recommend repairing Windows 2000 or restoring the entire operating system and the registry at the same time. If you know that the registry is the only damaged component, simply restoring the System State and not other system files may save a certain amount of time.

When Active Directory is running, it is not possible to restore the System State. This limitation requires that you stop the Active Directory services by doing the following: Reboot Windows 2000, and select the advanced start-up option Directory Services Restore Mode. Once the system has completed the boot, restore the System State. After restoring the System State, reboot Windows 2000 a second time.

If you're using another backup program, then simply follow the instructions provided with the program. The same general cautions about which files to restore (only the System State, or the entire operating system) still apply. The main difference between most backup programs is the user interface.



NOTE When restoring, be especially cautious that you do not restore the wrong version of the System State. Generally, you would want to make sure that you restore the most current working version of the registry.

Recovering a Copied Registry

A registry that has been backed up using copy or xcopy is restored in the opposite manner from which it was backed up. For example, if you have the NTFS file system, then you would have to restart the system using your backup copy of Windows 2000.

FAT and NTFS

When restoring a registry on a FAT-based file system running Windows 2000, it's necessary to boot DOS, Windows 95/98, or a second copy of Windows 2000. If you have a dual boot installed (either DOS or Windows 95/98), it will be OK if you use the dual boot to get to the other operating system.

If you are restoring the registry on an NTFS system, then dual-boot into the backup copy of Windows 2000 that you installed to back up the registry.



WARNING Once running the alternate operating system, find your latest working copy of the registry *before* you copy it over the registry that you think is corrupted, and back up the current registry to another location. Take this precaution just in case the current registry is not the problem (it happens) and the backup copy is not as good as you think it is.

You can follow these steps to restore your registry from a backup you have created:

1. Boot to another operating system: Windows 2000/NT, DOS, or Windows 95/98 for FAT; Windows 2000/NT for NTFS.
2. Save the current registry to a safe location just in case it is not the problem.
3. Copy your saved registry (from wherever it was stored) to the registry location.
4. Boot the original version of Windows 2000 and test to see if the restoration worked. If it didn't, keep reading, more golden tips are coming up soon.

ERD Strikes Again: Using Setup to Recover

If there are no other backup copies of the original registry, then you'll have to fall back on the ERD diskette and the repair directory. This technique is fraught with peril, including the fact that the saved registry may not have all the necessary information.

Properly restoring the system registry from the repair directory and the ERD diskette requires running the Windows 2000 Setup program. When it first starts, Setup examines the hard drive and looks for already-installed copies of Windows 2000 and their repair directories. Once the examination is complete, Setup will give you four choices:

- Help, which is invoked by pressing F1.
- Set up Windows 2000, invoked by pressing Enter.

- Repair a damaged Windows 2000 installation, by pressing R.
- Quit, because this was all just a big mistake, by pressing F3.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Now, we know that we are in trouble at this point—the only choice is whether it might be possible to recover from the problems without doing a complete reinstallation of Windows 2000.

Let's say that we are going to try to repair. First, we select the repair option, by pressing R. At this stage, the Setup program switches to repair mode and continues. The next screen will display four choices. You may choose any combination or all of them:

Inspect registry files: This choice allows the repair program to check and repair the registry files. This is the option that most of us will select. The repair program will need either an ERD diskette or the files stored in the %systemroot%\repair directory.

Inspect start-up environment: The start-up environment is the Boot Manager, which is called by the program contained in the boot sector. There are also other supporting files—including boot.ini, ntdetect.com, and others—that must be validated. The repair program will repair or replace these files as best as it can, but be prepared for some items to be restored to the state they were in when you installed Windows 2000.

Verify Windows 2000 system files: Verifying the system files is a process where the repair program will go through the root directory and all the system directories (such as the Windows and System directories) and verify that each and every file is valid. This process is used when a hard disk error (especially on an NTFS volume) has made one or more system files invalid. Careful! You will lose all service packs installed to this repair process. Reinstall your service packs immediately after choosing this option.

Inspect boot sector: There are several reasons to inspect (and repair) the boot sector. For example, if you inadvertently install another operating system with boot sector virus infections, this could damage the boot sector, especially with the FAT file system.

All four of these selections will be selected by default. You can use the selector bar (use the arrow keys) to highlight and deselect any option that is not desired; the Enter key is used to select or clear an option.

Once you have elected to continue, Setup will do a device check. This is the same check that is done prior to an installation of Windows 2000.

The next stage is to determine where the registry repair information will be coming from. Remember, you can use either the ERD or the copy stored in the repair directory. If you have multiple installations of Windows 2000, be sure to choose the correct repair directory to repair from.



TIP The ERD will tell Setup which copy of Windows 2000 you are attempting to repair. You *cannot* use the ERD from one installation of Windows 2000 to repair another installation of Windows 2000. It just won't work.



If you don't have an ERD (or you don't want to use it), then Setup will search your drive for Windows 2000. You may have multiple installations of Windows 2000; this is common, considering how many times I've recommended installing at least two copies. If this is the case, Setup will list each installation of the OS that it finds. Select the version of Windows 2000 you want to repair and press Enter to repair the selected installation.



WARNING Careful! Make sure you repair the right Windows 2000 installation if you have more than one copy of the OS installed. Nothing is worse than successfully repairing a copy of Windows 2000 that wasn't broken in the first place.

Next, Setup does a drive check. The message indicates that drives are being checked, and the status indicator at the bottom of the screen shows the progress. Actually, Setup only checks the boot (C:) drive, but that's probably all that is needed right now.

The next prompt, which is displayed when you have elected to have the registry repaired, is to determine which key or keys are to be repaired:

- System
- Software
- Default
- ntuser.dat
- Security
- SAM

Replacing some hives and not others might result in some problems if items in the registry have been updated since the registry was last saved. Typically, it is best to replace all files if possible to avoid any problems with different versions.

Once the registry has been updated, the Setup program will then prompt you to remove the diskette from its drive and reboot the computer. If all went well, the computer will reboot and run.

Loading a .reg file

Any .reg file created by RegEdit (discussed earlier) is usually loaded by simply double-clicking the .reg file in the Explorer program or on the Desktop.

You can also go into RegEdit to load the .reg file. From the RegEdit main menu, select Registry \emptyset Import Registry File. Actually, when you double-click a .reg file, Windows 2000 starts RegEdit to do the registry file load. The main advantage of loading a registry file from the RegEdit menu is that you're able to see the effect of the registry load in RegEdit.

A .reg file, being a text file, may be *carefully* edited. Did I emphasize *carefully* enough? Realize you are making a registry change if you modify the .reg file and then reload it. And make certain that the editor you use understands Unicode. Notepad will work fine, just remember not to use the default .txt file extension that Notepad uses when saving the file.

Realize that you will not be able to use this technique if you are unable to boot or run Windows. This is another good reason to have multiple backups of the registry in different formats.



NOTE When restoring the registry, several errors are displayed. Some errors will state "System Process - Licensing Violation" and advise the user that the system has detected tampering with the product registration component of the registry. Click OK when these messages appear and also when another error stating that it was not possible to write to the registry shows up. This final error is actually an artifact of the licensing violation errors and does not indicate a failure of the entire process.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

To make the restored registry active, it is necessary to restart Windows 2000. (Windows 2000 caches most of the registry while it is running.) There is no prompt to restart. However, some changes to the registry will not be reloaded until the system is restarted. Select Shut Down from the Start menu and then select Restart the Computer in the Shutdown dialog box.



NOTE It is not uncommon for applications to update the registry using a .reg file during program installation time. This is one method used by software developers. Why? Simple: this allows the registry to be repaired, restoring the application's default values without having to reinstall the entire program.

The Recovery Console



The Recovery Console is a new tool that has been added to Windows 2000 to allow recovery from a number of failures. Previously, all we could do was to boot another copy of Windows 2000 and hack our way around, replacing files, even registry components, in the blind hope that we would somehow fix the problem.

Now, with Windows 2000, we have two new tools to use: the Recovery Console and the Safe Mode feature.

The Recovery Console is a powerful, simple (no, that's not an oxymoron!) feature that is supplied with Windows 2000. Now, realize that the Recovery Console is not installed by default. You must install the Recovery Console before you can use it. Installing the Recovery Console after the system has failed is quite like locking the barn door after the horse has been stolen—it really won't work that well.

Installing the Recovery Console

The Recovery Console must be installed before disaster strikes. It will be difficult (maybe even impossible) to install it after a disaster has reared its ugly head. So, let's install the Recovery Console right now.

First, you must use the Windows 2000 distribution CD (or share, if installing from a network device). The Recovery Console is installed using the winnt32.exe program. The winnt32.exe program is the same program that is used to install Windows 2000; however, by selecting the correct option, you are able to tell winnt32.exe to not install Windows 2000, but to install the Recovery Console instead.



NOTE It is not possible to install the Recovery Console at the same time as Windows 2000. You must first install Windows 2000, then install the Recovery Console. If you have multiple copies of Windows 2000 installed, it is only necessary to install the Recovery Console one time—the Recovery Console will work with as many copies of Windows 2000 as are installed.

Desktop Library
[Click to access!](#)

The steps for installing the Recovery Console are:

1. Insert the distribution CD, and change into the i386 directory.
2. Run `winnt32.exe`, using the `/cmdcons` option. Typically, no other options are needed, though some users may wish to specify source options, especially if installing from a network share rather than a hard drive.
3. The `winnt32.exe` program will present the dialog box shown in Figure 2.1. This dialog box allows you to cancel the installation if you need to. Note that multiple installations of the Recovery Console will simply overwrite previous installations; in such cases, no error is generated.



FIGURE 2.1 Setting up the Recovery Console using `winnt32 /cmdcons` bypasses all other setup options.

4. If there are no errors, the dialog box shown in Figure 2.2 is displayed. The Recovery Console is ready for use at this point.



FIGURE 2.2 The Recovery Console has been successfully installed.

What's in the Recovery Console?

The Recovery Console consists of a minor modification to the `boot.ini` file, and the addition of a *hidden* directory on the boot drive. The added directory's name is `cmdcons`. The change to the `boot.ini` file is simply the addition of an additional line providing for a new boot option:

```
C:\CMDCONS\BOOTSECT.DAT="Microsoft Windows 2000 Recovery Console" /cmdcons
```

This option consists of a fully qualified file name (`c:\cmdcons\bootsect.dat`), a text description (Microsoft Windows 2000 Recovery Console), and a boot option (`/cmdcons`).

As everyone should be well aware, the Windows 2000 Boot Manager is able to boot virtually any operating system (assuming that the operating system is compatible with the currently installed file system). The Recovery Console does qualify as an operating system, though it is very simple—and limited.

A major question will always be this: is the Recovery Console secure? In most situations, the Recovery Console is actually quite secure. The user, at start-up of the Recovery Console, is prompted for two pieces of information:

- They must specify which Windows 2000 installation is to be repaired (assuming that there is more than one Windows 2000 installation!)
- They must successfully enter the Administrator's password for that installation. The Recovery Console will then use the installation's SAM to validate this password to ensure the user has the necessary permission to use the system.

A question comes to mind: If the Administrator's password is lost or otherwise compromised, not only may it be impossible to use the Recovery Console, but anyone with access to the compromised password would be able to modify the system with the Recovery Console. This is not really an issue, though. If the Administrator's password is lost, that's life. It will be difficult, if not impossible, to recover the password. If the security of the Administrator's password is compromised, then it will be necessary to repair the damage—changing the password is mandatory in this case. In either case, the Recovery Console is no less secure than Windows 2000 is.

In the `cmdcons` directory, there are over one hundred files. Most of these files are compressed, and will be uncompressed by the Recovery Console when needed. Here's a list of the uncompressed files found in this directory:

```
C:\cmdcons\autochk.exe
C:\cmdcons\autofmt.exe
C:\cmdcons\biosinfo.inf
C:\cmdcons\BOOTSECT.DAT
C:\cmdcons\disk101
```

```
C:\cmdcons\disk102
C:\cmdcons\disk103
C:\cmdcons\disk104
C:\cmdcons\kbdus.dll
C:\cmdcons\migrate.inf
C:\cmdcons\ntdetect.com
C:\cmdcons\setupldr.bin
C:\cmdcons\setupreg.hiv
C:\cmdcons\spcmdcon.sys
C:\cmdcons\system32
C:\cmdcons\txtsetup.sif
C:\cmdcons\winnt.sif
C:\cmdcons\system32\ntdll.dll
C:\cmdcons\system32\smss.exe
```

The files disk101, disk102, disk103, and disk104 are disk image identifier files, and they contain nothing but a single space and a carriage return/line feed. The BOOTSECT.DAT file is the bootable boot sector image file. The migrate.inf file contains information used to update the registry if needed. The setupreg.hiv file is used to update the registry; however, this file is in a special format usable only with certain applications. The cmdcons directory also contains the subdirectory system32. This subdirectory contains two files, ntdll.dll and smss.exe.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Using the Recovery Console

Once the Recovery Console is installed, it will appear in the start menu as the last item in the list, named "Microsoft Windows 2000 Recovery Console."



WARNING It is strongly recommend that the Recovery Console not be invoked unless absolutely necessary! The commands available in the Recovery Console are powerful, and if used improperly, they can destroy a Windows 2000 installation.

To use the Recovery Console, follow these steps:

1. Boot the system.
2. When the start-up screen is displayed, select "Microsoft Windows 2000 Recovery Console."
3. Select the installation to be repaired if there are multiple Windows 2000 installations.
4. Enter the correct administrator password for the installation to be repaired.
5. Use any Recovery Console commands (see the next section) needed to do the repair.

When you're done repairing the installation, simply enter the **exit** command to restart the computer.

Starting the Recovery Console from the Installation CD-ROM

The steps to start the Recovery Console for computers that either do not have the Recovery Console installed or cannot be booted (perhaps due to errors in the partition table, or MBR) are:

1. Boot the system, using the CD-ROM disk or diskettes as appropriate.
2. When the initial setup text screen is displayed, select Repair by pressing the R key.
3. At the prompt, select Recovery Console by pressing C.
4. Select the installation to be repaired if there are multiple Windows 2000 installations.
5. Enter the correct administrator password for the installation to be repaired.
6. Use any Recovery Console commands (see the next section) needed to do the repair.

When you're done repairing the installation, enter the **exit** command to restart the computer.

Recovery Console Commands and Options

When the computer is started in the Recovery Console mode, a prompt similar to a command prompt is the



only interface available to the user. The Recovery Console's functionality is limited, and there is only support for the following commands:

attrib	Changes file attributes. The read, hidden, and system attributes may be either set or cleared as desired.
batch	Allows execution of a set of Recovery Console commands that have been saved in a text file. The filename and extension both must be specified for the batch command to work. This command allows specifying an output file as well.
chdir (cd)	Works identically to the command session's cd command, changing the current working directory to the directory specified or, if no directory is specified, displaying the current working directory.
chkdsk	Works similarly to a command session's chkdsk command. Two options are available: /p specifies that the drive is to be checked regardless of whether the dirty flag is set; /r specifies that chkdsk should repair any bad sectors found.
cls	Works identically to the command session's cls command—clears the screen.
copy	Copies a file from a source location to a destination location. The file, if compressed, is uncompressed when copied. No wildcards are permitted with the copy command. There are no options to this command.
delete (del)	Works much like a command session's delete command. This command deletes the specified file or files. It will only work in the system directories of the installation being repaired, in hard drive root directories, and with local installation source files.
dir	Works similarly to a command session's dir command. This command will display the names of files and subdirectories in the location specified. The dir command has no options, listing file sizes, modification dates, and attributes.
disable	Used to disable a service or device driver. The service or device driver to be disabled is marked as SERVICE_DISABLED to prevent it from being started when the system is subsequently restarted.
diskpart	Manages partitions on disk devices. This command is able to add or delete partitions as desired. When adding a partition, a command parameter specifies the size of the partition in megabytes.
enable	Used to enable a service or device driver. The service or device driver to be enabled is marked with the user specified service type: SERVICE_AUTO_START, SERVICE_DISABLED, SERVICE_DEMAND_START, SERVICE_BOOT-START, or SERVICE_SYSTEM_START.
exit	Ends the Recovery Console session and reboots the computer.
expand	Works similarly to a command session's expand command. This command allows expanding files from a source CAB file. Two options are available: /d displays the contents of the CAB file; /y suppresses any overwrite warnings that may be given.
fixboot	Repairs or replaces the (optional) specified drive's boot sector.
fixmbr	Repairs or replaces the (optional) specified drive's master boot record.
format	Works similarly to a command session's format command. This command allows formatting disks using FAT, FAT32, and NTFS. One option, /q, allows quick formatting without a scan when the drive is known to be good.
help	Lists the available Recovery Console commands.
listsvc	Displays a list of services and drivers that are currently available on the computer.
logon	Run automatically when the Recovery Console is first started, this command is used to log on to an installation of Windows NT 4 or Windows 2000.
map	Used to display a list of all drive mappings. This command's output is very useful for the fixboot, fixmbr, and fdisk commands.
mkdir (md)	Works similarly to the command session's md (mkdir) command. This command allows creating directories within the system directories of the currently logged-on installation, removable disks, root directories of hard disk partitions, and local installation sources.
more	Works like the command session's type command. Displays the file's contents on the screen. There are no parameters for the more command.
rename (ren)	Allows the user to rename a file. This command does not support wildcard specifications.

- rmdir (rd) Works similarly to the command session's rd (rmdir) command. This command allows deleting directories within the system directories of the currently logged-on installation, removable disks, root directories of hard disk partitions, and local installation sources.
- set The Recovery Console supports a limited set of environment variables. These variables affect Recovery Console commands only.
- systemroot Changes to the current installation's %systemroot% directory. Functionally equivalent to cd %systemroot% in a normal command session.
- type Works like the command session's type command. This command displays the file's contents on the screen. There are no parameters for the type command.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The Recovery Console may be installed permanently so that whenever the system is booted, there is an option to select the Recovery Console. This works well for installations that will still boot to the start menu (where one selects the installation or operating system to be booted). The Recovery Console is placed into the directory cmdcons, located on the boot drive.



NOTE The cmdcons directory is always located on the boot drive, not on the system drive, unless the boot drive is also the system drive.

Other Backup and Restore Programs

There are other registry backup and restoration programs. One excellent source for them is the Windows 2000 Resource Kit's REG program, which has backup and restore functionality. Take a look at Chapter 4 for a listing of the registry tools found in the Windows 2000 Resource Kit.

Hints and Kinks from the Experts

Here's another installment of good stuff from the Windows gurus.

Why Don't My Changes to the Registry Take Effect?

Always reboot. Reboot after restoring any registry values. Windows 2000 does not reload many values except at boot time. There's nothing worse than wondering why your "fix" didn't work when Windows 2000 was simply not loading it.

Users Never Have a Current System State Backup!

In most sites, users rarely have a current System State backup when they need one. Give them one with this procedure. Use the scheduler (the AT command or a good one like OpalisRobot) on each workstation to schedule a save of the System State. The batch file to schedule is:

```
net use x: /delete
net use x: \\YourServer\RepairShare$ /persistent:no
if not exist x:\%computername% md x:\%computername%
REM - Use NTBackup to save the system state!
REM - '/l: f' used to create a full backup log.
```



Desktop Library
[Click to access!](#)

```
ntbackup backup systemstate /f "x:\%computername%\System State" /l:f  
  
net use x: /delete  
exit
```

where %computername% is a subdirectory of the hidden share on the server (one for each workstation). When you need the System State for that workstation, just reattach the share to the target system.

The scheduler must be run under the system context and allowed to interact with the desktop or under the context of an administrative user. If you use the system account, you can't schedule the copy because the system account has no network access. Use a ROBOT account that is a member of the Administrator group with a non-blank, non-expiring password. Use full path names for all files.

Here is a sample schedule for workstation wsA:

```
AT \\wsA 01:00 /interactive every:M,T,W,Th,F,S,Su  
\\YourServer\RepairShare$\Repair.bat
```

You can dress up the Repair.bat with logging, messaging, and so on.

What's the Difference between the Contents of the ERD and the Repair Directory?

The ERD and the repair directory contain the same files. One is as good as the other. However, you can have as many ERDs as you want while there can be only one repair directory. In fact, I'd recommend at least three or four ERDs for each Windows 2000 installation you have—more if you can find the diskettes.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

Search Tips

Advanced Search

PUBLICATION LOOKUP

JUMP TO TOPIC



Desktop Library

[Click to access!](#)
**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 3**Anatomy of the Registry: The Blood, Gore, and Guts****FEATURING:**

<i>Of hives and bees—a registry overview</i>	46
<i>HKEY_LOCAL_MACHINE: the machine's configuration</i>	53
<i>HKEY_USERS: settings for users</i>	62
<i>HKEY_CURRENT_CONFIG: the Current Configuration Settings</i>	66
<i>HKEY_PERFORMANCE_DATA: the Performance Monitor Settings</i>	67
<i>NTUSER: the new user profile</i>	67
<i>Hints and kinks from the experts</i>	69

In Chapter 1, we talked a little about what the registry is and the terminology used for its various components. In this chapter, we will get into more of the details of what actually is in the registry. If you're only interested in how to use (or recover) the registry, but not *what* the registry is, it's possible to skip this chapter. However, if you're unsure about this, I'd recommend reading it anyway.

Now humor me for just a moment; I think I'm going to back up my registry. In fact, it is a good time for *you* to do a backup as well, since it is entirely possible that at any time you might have some kind of problem (or disaster) with the registry and really need that backup copy to restore it. Start Backup, and select the System State in the Backup tab, and back up to a safe location. Then in the Welcome tab, select Emergency Repair Disk.

Next, let some time pass by...

Ah, that feels better. I've got a fresh backup copy of my registry just in case I do something stupid, and so do you—not that we ever do anything stupid, right?

The registry is subdivided into a number of clearly defined sections, called *hives*:

- HKEY_CLASSES_ROOT
- KEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG

Some hives are less important than others. For example, a damaged Security Accounts Manager key (SAM), can probably be recovered easily without serious, permanent problems. You could possibly lose the entire user database, so no users would be able to log onto the server. However, as long as you can log on as Administrator, the worst case is that you would have to enter the other user information again. The default SAM registry will contain at least the initial Administrator user ID and password, which you would have to know.

However, say you lose the system component of the registry without adequate backup. In that case, it is unlikely that you'll be able to recover without reinstalling Windows 2000, and that would be a painful experience at best.

Of Hives and Bees—A Registry Overview

As we discussed in Chapter 1, the Windows 2000/NT registry (and the registry for Windows 95/98) is arranged into logical units called *hives*. Though I can't vouch for its truth, legend has it that some unnamed programmer at Microsoft seemed to see a logical relationship between the various keys in the registry and the structure of a beehive. Now me, I just don't see this, so let's consider the two following alternative analogies:

- The registry is arranged just like the folders and files contained on your hard drive. Hives are analogous to root directories, and keys are like subdirectories and files. In fact, this relationship is almost 100 percent parallel: Hives are usually shown separated by backslashes (just like directories on the drive) from keys, and keys typically (but not always) have values. Remember, a file may also be empty.
- The registry is arranged as a hierarchical database, nothing more, and nothing less. If you are a database person, this view of the registry might make more sense to you. In truth, the database arrangement is more like the registry's actual construction.

Specific data is assigned to a key. As I've mentioned, some registry keys don't have a value set; this is also acceptable.



WARNING Be careful not to delete empty keys just because they are empty. Even though they don't have a value, their presence in the registry may be necessary.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

HKEY_CLASSES_ROOT

The HKEY_CLASSES_ROOT hive contains information about both OLE and various file associations. The purpose of HKEY_CLASSES_ROOT is to provide for compatibility with the existing Windows 3.x registry.

The information contained in HKEY_CLASSES_ROOT is identical to information found in HKEY_LOCAL_MACHINE\Software.

HKEY_CURRENT_USER

The HKEY_CURRENT_USER hive is used to manage specific information about the user who is currently logged on. This information includes:

- The user's Desktop and the appearance and behavior of Windows 2000 to the user.
- All connections to network devices, such as printers and shared disk resources.
- Desktop program items, application preferences, screen colors, and other personal preferences and security rights. They are stored for later retrieval by the system when the user logs on.

All other environment settings are retained for future use.

By accessing the roaming user profile, Windows 2000 is able to make any workstation that the user logs onto appear the same to the user. Domain users need not worry about having to set up or customize each workstation that they will be using.

The information contained in HKEY_CURRENT_USER is updated as users make changes to their environments.

HKEY_LOCAL_MACHINE

The HKEY_LOCAL_MACHINE hive contains information about the computer that is running Windows 2000. This information includes applications, drivers, and hardware. There are five separate keys contained within HKEY_LOCAL_MACHINE:

Hardware: The key used to save information about the computer's hardware. To allow new hardware to be added easily, the Hardware key is always re-created when the system is booted. Changes to this key are not meaningful. Contained within the Hardware key are the following four subkeys:

Description: Contains information about the system, including the CPU, FPU, and the system bus. Under the system bus is information about I/O, storage, and other devices.

DeviceMap: Contains information about devices (keyboards, printer ports, pointers, and so on).

ResourceMap: Contains information about the HAL (Hardware Abstraction Layer). Remember, as we approach the year 2001, HAL is not a talking computer on a spaceship, HAL is the hardware. Also contained are I/O devices, drivers, SCSI adapters, system resources, and video resources.

SAM: The Security Accounts Manager (SAM) stores information about users and domains in the SAM key. This information is not accessible using any of the resource editors. Rather, this information is better managed using the administrator's User Manager program.

Security: Contains information about local security and user rights. A copy of the SAM key is found in the Security key. As with SAM, the Security key is not accessible using the resource editors, and the information is best modified using the administrator's tools.

Software: Contains information about installed system and user software, including descriptions. There are generally subkeys for each installed product in which the products store information—including preferences, configurations, MRU (most recently used files) lists, and other application-modifiable items.

System: Contains information about the system start-up, device drivers, services, and the Windows 2000 configuration.

HKEY_USERS

The HKEY_USERS hive contains information about each active user who has a user profile. A minimum of two subkeys are in the HKEY_USERS key: .DEFAULT and information for the currently logged-on user. The purpose of the .DEFAULT key is to provide information for users who log on without a profile. Information for the current user is stored under the user's SID (security identifier).

Personal profiles are contained in the %systemroot%\Profiles folder or the %sysdrive%\Documents and Settings\Default User folder, unless roaming profiles are used, in which case a copy will be stored there, but the original will reside on a server.

HKEY_CURRENT_CONFIG

The HKEY_CURRENT_CONFIG hive contains information about the system's current configuration. This information is typically derived from HKEY_LOCAL_MACHINE\System and HKEY_LOCAL_MACHINE\Software, though HKEY_CURRENT_CONFIG does not contain all the information that is contained in the source keys.



NOTE As I noted in Chapter 1, the HKEY_DYN_DATA hive no longer exists in Windows 2000. In Windows NT 4, this hive was intended to contain information about the system's PnP (Plug and Play) status. However, since Windows NT 4 does not support PnP, this key is empty.

Registry Key Data Types

Values have different data types:

REG_BINARY: Represents binary values. They may be edited or entered as hexadecimal or binary numbers. Figure 3.1 shows RegEdt32's Binary Editor window. RegEdit has a similar edit window, though it is not as flexible in how data is entered.

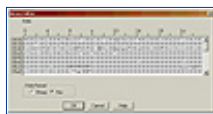


FIGURE 3.1 The Binary Editor window for RegEdt32

REG_SZ: Used for registry keys that contain strings. Editing is easy; just type in the new string. Case is preserved, but realize that the string is initially selected, so be careful not to inadvertently delete it. Strings are of fixed length and are defined when the key is created. Figure 3.2 shows a string being edited in the String Editor window. A string key may be made longer by adding more characters to the string; it will be reallocated if this happens.



FIGURE 3.2 The String Editor window for RegEdt32

REG_EXPAND_SZ: Used if the key is to contain an environment variable that must be expanded. Some keys need to contain values that reference environment variables, much like a batch file—for example, if a string contains the field `%systemroot%\System32`, and it is necessary for the `%systemroot%` part of the string to be replaced with the value that is assigned to it in the environment. To do this substitution, this string must be defined as a REG_EXPAND_SZ type string. The result of the expansion would then be passed to the requestor. `%systemroot%` is a standard environment variable containing the location, drive, and directory where Windows 2000 has been installed. RegEdt32 uses the same window as is used for REG_SZ for entering a REG_EXPAND_SZ key, as shown in Figure 3.3.



NOTE Any environment variable, either system created or created by the user, may be used in a REG_EXPAND_SZ key.

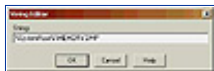


FIGURE 3.3 The String Editor window for RegEdt32

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

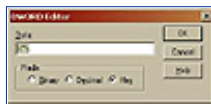
Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

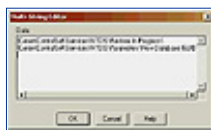
REG_DWORD: A 32-bit value. The value is entered as decimal, hexadecimal, or binary. The DWORD Editor window only allows us to enter valid numeric data to help save us from sloppy typing, as Figure 3.4 shows.

**FIGURE 3.4** The DWORD Editor window for RegEdt32

REG_MULTI_SZ: Used to store multiple strings in a single registry key. Normally, a string resource in the registry can contain only one line. However, the multi-string type allows a string resource in the registry to hold multiple strings as needed. Figure 3.5 shows multiple strings being edited, with two lines present in this example.



NOTE The Windows 95/98 registry editor, RegEdit, does not support the REG_MULTI_SZ type. If a REG_MULTI_SZ item is edited with RegEdit, it is possible to corrupt the data that is contained in it.

**FIGURE 3.5** The Multi-String Editor window for RegEdt32

REG_FULL_RESOURCE_DESCRIPTOR: Used to manage information for hardware resources. No one should edit the items that appear in the Resource Editor fields. Figure 3.6 shows a disk resource object displayed in RegEdt32. However, these objects are not ever changed manually with the resource editors.

**FIGURE 3.6** A disk resource shown in the Resources window for RegEdt32

REG_NONE: An identifier used when there is no data stored in the key. It doesn't take a rocket

scientist to figure out that there is no editor for the REG_NONE type.

REG_UNKNOWN: REG_UNKNOWN is used when the key's data type cannot be determined.

HKEY_LOCAL_MACHINE: The Machine's Configuration

The HKEY_LOCAL_MACHINE hive contains information about the current hardware configuration of the local computer. The information stored in this hive is updated using a variety of processes, including the Control Panel, hardware and software installation programs, and administrative tools, and is sometimes automatically updated by Windows 2000.

It is important not to make unintended changes to the HKEY_LOCAL_MACHINE hive. A change here could quite possibly render the entire system unstable.



NOTE All the settings in the HKEY_LOCAL_MACHINE hive are recomputed at boot time. If a change has been made, and the change is causing problems, first try rebooting the system. The Windows 2000 Boot Manager should rebuild the HKEY_LOCAL_MACHINE hive at reboot time, discarding any changes made.

HKEY_LOCAL_MACHINE\Hardware: The Installed Hardware Key

HKEY_LOCAL_MACHINE\Hardware contains information about the hardware configuration of the local machine. Everything hardware related (and I do mean everything) will be found in this hive.

In Windows 2000, the HKEY_LOCAL_MACHINE\Hardware key is subdivided into three keys:

Description: Contains descriptive information about each device, including both a general description and information about basic configurations and so on.

DeviceMap: Contains information about devices, including the location in the registry where a device's full configuration is saved.

ResourceMap: Contains translation information about each major component that is installed in the system. Most keys contain a set value entries named .Raw and .Translated.



NOTE In Windows NT 4, the Hardware key contains another subkey, OWNERMAP, which contains information about removable PCI-type devices. These are devices plugged into the system's PCI bus, but generally not permanently installed on the system's motherboard. However, not all PCI-type devices will be listed in OWNERMAP.

DESCRIPTION

Within HKEY_LOCAL_MACHINE\HARDWARE\Description is a wealth of information about the installed hardware. The only subkey, System, describes the CPU and I/O fully. Items in the Description key are always redetected at boot time. The following subkeys are contained in the System subkey:

CentralProcessor: The CentralProcessor subkey contains information about the CPU. This includes speed, which is an identifier that contains the CPU's model, family, and Stepping. Also included in this subkey is vendor information; for example, a "real" Intel CPU has the VendorIdentifier string "GenuineIntel".

FloatingPointProcessor: The FloatingPointProcessor subkey describes the system's FPU (floating point unit) in a set of entries similar to that of the CPU. The fact that the typical CPU has an integral FPU is not considered here; the FPU will be listed separately, regardless.

MultiFunctionAdapter: The MultiFunctionAdapter subkey describes the system's bus (PCI), any PnP BIOS installed, and other devices, including the controllers for disk drives, keyboards, parallel and serial ports, and the mouse. For a mouse that is connected to a serial port, the mouse will be found under the serial port, while a mouse that is connected to a PS/2 mouse port will be shown connected to a pointer controller as a separate device.

ScsiAdapter: The ScsiAdapter subkey describes the system's IDE interfaces, if there are any. Windows 2000 lists these as SCSI interfaces, and they include the controllers for IDE disk drives, IDE CD-ROM drives, and other supported IDE devices.



NOTE ScsiAdapter lists only the devices attached to the IDE controller. The IDE controller itself is described in HKEY_LOCAL_MACHINE\Hardware\DeviceMap.

Typically, the Description key can be used to determine what hardware is installed (and being used) and how the installed hardware is connected. However, some devices, such as storage devices (non-IDE hard drives, SCSI devices, non-IDE CD-ROM drives, video, and network interface cards), are not listed in HKEY_LOCAL_MACHINE\Hardware\Description. Instead, they are listed in HKEY_LOCAL_MACHINE\Hardware\DeviceMap. Why? Because these devices are not detected at boot-up stage; instead, they are detected when they are installed.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

DeviceMap

The HKEY_LOCAL_MACHINE\Hardware\DeviceMap subkey contains information about devices, arranged in a similar fashion to the HKEY_LOCAL_MACHINE\HARDWARE\Description subkey discussed earlier. In the Devicemap subkey, the following subkeys are found:

KeyboardClass: Contains the address of the subkey that manages information about the keyboard itself.

PARALLEL PORTS: Contains the address of the subkey that manages information about the parallel printer ports.

PointerClass: Contains the address of the subkey that manages information about the system mouse.

Scsi: A complex subkey with information about each SCSI interface found on the computer. A note about what is considered a SCSI port is in order. Actually, Windows 2000 pretends that IDE devices, as well as many CD-ROM devices that are connected to special interface cards, are SCSI devices. This is a management issue. Windows 2000 is not converting these devices to SCSI, nor is it using SCSI drivers; rather, Windows 2000 is simply classing all these devices under a common heading of SCSI.

SERIALCOMM: Contains the address of the subkeys that manage information about the available serial ports. In Windows NT 4, if the system mouse is connected to a serial port and not to a PS/2 mouse port, that port is not listed in the SERIALCOMM subkey.

VIDEO: Contains the address of the subkey that manages the video devices. There are typically two devices defined in VIDEO: one is the currently used adapter, and the second is a backup consisting of the previously installed (usually the generic VGA) adapter's settings to use as a backup in the event of a problem with the video system.



NOTE In NT 4, DeviceMap includes two additional subkeys. KeyboardPort contains the address of the subkey that manages information about the keyboard interface unit, often called the 8042 after the original chip that served as the keyboard controller in the original PC. PointerPort contains the address of the subkey that manages information about the port that the system mouse is connected to.

ResourceMap

The ResourceMap subkey is used by all the various hardware device drivers to map resources that they will use. Each ResourceMap entry will contain information about the usage of:

- I/O ports
- I/O memory addresses

- Interrupts
- Direct memory access (DMA) channels
- Physical memory installed
- Reserved memory

The ResourceMap subkey is divided into subkeys for each class of device (such as Hardware Abstraction Layer), and under these subkeys lie subkeys for different devices.



Windows 2000 includes a new key in ResourceMap, called PnPManager. This key contains Plug-and-Play information.

HKEY_LOCAL_MACHINE\SAM: The Security Access Manager

HKEY_LOCAL_MACHINE\SAM contains information used by the Windows 2000 security system. It also contains user information (permissions, passwords, and the like). The SAM key is mirrored in HKEY_LOCAL_MACHINE\Security\SAM; making changes to one changes the other.



NOTE Can't see the SAM or Security key? Use RegEdt32 to select the subkey you cannot see and then select Security Permissions from the main menu. Next, change the Type of Access from Special Access to Full Control.

In Windows 2000, this information is set using the Microsoft Management Console (MMC), Local Users and Groups branch. If the Windows 2000 system is a domain controller, the SAM is not used (we have the Active Directory services now). The SAM subkeys (both in HKEY_LOCAL_MACHINE\SAM\SAM and HKEY_LOCAL_MACHINE\Security\SAM) should only be modified using the MMC in Windows 2000 or the User Manager administrative programs in Windows NT. However, attempts to modify information that is in the SAM subkeys typically result in problems. For example, users will be unable to log on, wrong permissions will be assigned, and so on.



WARNING Don't attempt to modify the SAM or Security key unless you have made a full backup of your registry, including the SAM and Security keys, as described in Chapter 2.

HKEY_LOCAL_MACHINE\Security: The Windows 2000 Security Manager

The key HKEY_LOCAL_MACHINE\Security contains information relevant to the security of the local machine. This information includes:

- User rights
- Password policy
- Membership of local groups

This information is typically set using the Administrative User Manager or User Manager for Domains program.



NOTE Under Windows NT 4, the Security subkeys should only be modified using the User Manager or the User Manager for Domains. With Windows 2000 Active Directory, only the Active Directory administrative programs (Active Directory Users and Computers) should be used. Attempts to modify information in the Security key typically result in problems. For example, users are unable to log on, wrong permissions are assigned, and so on.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)**HKEY_LOCAL_MACHINE\Software: The Installed Software Information Manager**

The HKEY_LOCAL_MACHINE\Software registry key is the storage location for all software installed on the computer. The information contained in HKEY_LOCAL_MACHINE\Software is available to all users and consists of a number of standard subkeys as well as a few subkeys that may be unique to each computer.

One computer on my network, a Windows 2000 server, has the following subkeys in HKEY_LOCAL_MACHINE\Software. These subkeys correspond to items that I have installed on my computer:

Adobe: Contains information about the copy of Adobe's Acrobat program that was recently installed.

Federal Express: Contains information about the FedEx online access and support I have on my computer. All of my FedEx airbills are produced by computer, making shipments much easier.

INTEL: Contains information about the Intel 3D Scalability Toolkit that I installed at some point. I don't remember when or why, but it's there.

Intuit: Contains information specific to the financial software that is Intuit's specialty.

Qualcomm: Contains information specific to the Eudora email program. The nice thing about Eudora is that there is a free version for private use.

The following are system subkeys probably installed on your computer; however, some of these subkeys, such as ODBC and Clients, may not be present on some minimal installations:

Classes: Contains two types of items. First are file-type association items. For example, a typical association entry might have the name DIB, with a string that associates this name with the application Paint Shop Pro. Second are COM (Common Object Model) associations. For example, the extension .doc is associated with Microsoft Word for Windows or with WordPad, the default viewer for .doc files. Both WordPad and Word may be embedded in other applications. For instance, Outlook, Microsoft's upscale email system, can use Word-formatted documents and embed either Word for Windows or WordPad to display and edit these documents.

Clients: Contains client-server relationships. For example, Microsoft Outlook is a multipurpose program with email, a calendar, contact lists, news, and other features. Each of these parts of Outlook has a complex series of calling protocols that are defined in the Clients subkey.

Microsoft: Stores a number of items that pertain to Microsoft products or parts of Windows 2000. There can be as few as 20 or as many as 100 entries in the Microsoft subkey.

ODBC: Stores items that pertain to Open Database Connectivity, which allows applications to retrieve data from a number of different data sources. Many users install ODBC, either realizing that they are installing it or as a side effect of installing another product.

Program Groups: This subkey contains one value entry, `ConvertedToLinks`, which is used to indicate whether the program groups were converted. A value of one (0x1) shows that the conversion is complete. Even a system installed on a new computer that didn't require conversion will have this value.

Secure: If you say so. The `Secure` subkey is the location in which any application may store "secure" configuration information. Only an Administrator may modify this subkey, so mere mortal users can't change secure configuration information. Not many, if any, applications use the `Secure` subkey.

Windows 3.1 Migration Status: Used to indicate if the computer was upgraded from Windows 3.x to Windows NT 4/2000. Though at one time there were many upgrades, more users today are likely to be doing clean installations—virtually all existing Windows 3.x systems have already been upgraded. This key contains two subkeys: `IniFiles` and `REG.DAT`. These values show whether the `.ini` and `Reg.dat` files have been migrated successfully to later formats.



NOTE In NT 4, there's also a `Description` subkey that contains names and version numbers for software installed on the local computer. Though any vendor may use this subkey, the author can only see one entry, which is entered during installation of Windows 2000. Microsoft RPC (Remote Procedure Call) has several entries in this subkey.

HKEY_LOCAL_MACHINE\System: The System Information Manager

The `HKEY_LOCAL_MACHINE\System` subkey is used to hold start-up information used by Windows 2000 when booting. This subkey contains all the data that is stored and not recomputed at boot time.



NOTE A full copy of the `HKEY_LOCAL_MACHINE\System` information is kept in the file `System.alt`, found in the `%systemroot%\System32\config` directory.

The `HKEY_LOCAL_MACHINE\System` key (a.k.a., the `System` key) is organized into control sets (such as `ControlSet001`, `ControlSet002`, and `CurrentControlSet`) containing parameters for devices and services. (The key `Clone`, present in prior versions of Windows NT, is not found in Windows 2000.)

The main control sets are:

ControlSet001: The current and the default control set used to boot Windows 2000 normally. Mapped to `CurrentControlSet` at boot time, `ControlSet001` is the most critical component in the registry in the normal boot-up process.

ControlSet002: A backup control set from the Last Known Good boot that is used to boot from when the default control set (`ControlSet001`) fails or is unusable for some reason.

ControlSet003: `ControlSet003` (and `ControlSet00n` where n is greater than 3) is a backup control set from the Last Known Good boot that may be used to boot from when the default control set (`ControlSet001`) fails or is unusable for some reason.

CurrentControlSet: The `CurrentControlSet` is the control set Windows 2000 has booted from. It is usually mapped to `ControlSet001`.



NOTE The `Clone` control set found in NT 4 is the volatile copy of the control set (usually `ControlSet001`) that was used to boot the system. Created by the system kernel during initialization, this key is not accessible from the registry editor.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

There are three or four other items in the HKEY_LOCAL_MACHINE\System key:



MountedDevices: Contains items for each locally attached storage device that is available to the system.

DISK: Contains items for each mapped CD-ROM drive. For example, I map my CD-ROM drives to drive letters after S:, so I have three entries in this subkey mapping each CD-ROM drive to a different drive letter. This subkey is updated by the Disk Administrator tool.

Select: Contains four subkeys. It also has information on which control set was booted and which subkey is the Last Known Good set. Also, if there is a “failed” control set, the failed control set’s identity will be found in the Select subkey.

Setup: Contains information used by Setup to configure Windows 2000. This information includes locations of drives and directories, the setup command line, and a flag telling if setup is currently in progress.

The HKEY_LOCAL_MACHINE\System key is critical to both the boot process and to the operation of the system. Microsoft has created a number of tools and processes that help protect the HKEY_LOCAL_MACHINE\System key information. These include the Last Known Good boot process, which allows mapping in a known (or so we hope) copy of the control set, which in turn allows the system to boot if the original control set is too damaged to be booted.



WARNING Do not, I repeat, *do not*, boot using the Last Known Good control set unless it is necessary! Any changes made to the system during the previous session will be lost, gone, forever and ever!

When modifying the control sets, be aware of the process of booting and creating the control sets. Generally, modifying a backup control set won’t affect the system.

When Is the Current Control Set the Last Known Good Control Set?

At some point in the boot process, the current control set will be copied into the Last Known Good control set. In Windows 2000, the process of replacing the Last Known Good control set is done after the initial logon is performed. This allows the system to catch any problems related to the logon process.

HKEY_USERS: Settings for Users

Current user configurations are saved in HKEY_USERS. In HKEY_USERS there are three keys. The first key, .DEFAULT, is the default user profile. This profile is used when no user is currently logged on. Once a user logs on, their profile is loaded and stored as the second and third keys found in HKEY_USERS.



Desktop Library
Click to access!

The second key, the user profile for the user who is currently logged on, appears as something like this:

S-1-5-21-45749729-16073390-2133884337-500

The second key is the key for a specific user's profile. The profile either is the user's own profile or is copied from the default user profile (found in %systemdrive%\Documents and Settings\All Users) if the user has not established his or her own profile.

The third key looks something like:

S-1-5-21-45749729-16073390-2133884337-500_Classes

This key contains information about the various classes registered for the current user.

These last two long, magical registry keys need some explanation. The number, as a whole, is called a SID (security identifier). There is a lot of information in a SID. For example, the ending three- or four-digit number is used to identify both the user, and for some users, the type of user. Table 3.1 lists a number of general user types that might be assigned. In this book, the most commonly seen value is 500, which is assigned to me, the system administrator account.

TABLE 3.1: COMMON SID VALUES

User Group	SID
DOMAINNAME\ADMINISTRATOR	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-500
DOMAINNAME\GUEST	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-501
DOMAINNAME\DOMAIN ADMINS	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-512
DOMAINNAME\DOMAIN USERS	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-513
DOMAINNAME\DOMAIN GUESTS	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-514

General users might be assigned SIDs ending in four-digit numbers starting at 1000. My domain has a user called Pixel, whose SID ends in 1003, and another user, Long, whose SID ends in 1006. Get the picture?

There are also a number of built-in and special groups of SIDs, as shown in Tables 3.2 and 3.3.

TABLE 3.2: THE BUILT-IN LOCAL GROUPS

Built-in Local Groups	SID
BUILTIN\ADMINISTRATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-544
BUILTIN\USERS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-545
BUILTIN\GUESTS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-546
BUILTIN\POWER USERS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-547
BUILTIN\ACCOUNT OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-548
BUILTIN\SERVER OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-549
BUILTIN\PRINT OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-550
BUILTIN\BACKUP OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-551
BUILTIN\REPLICATOR	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-552

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

Click to access!

TABLE 3.3: THE SPECIAL GROUPS

Special Groups	SID
\CREATOR OWNER	S-1-1-0x-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxx
\EVERYONE	S-1-1-0x-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxx
NT AUTHORITY\NETWORK	S-1-1-2x-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxx
NT AUTHORITY\INTERACTIVE	S-1-1-4x-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxx
NT AUTHORITY\SYSTEM	S-1-1-18-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxx

Naturally, there are many more SID codes and definitions. Tables 3.1–3.3 simply show a few of the more commonly used SIDs.



NOTE Remember to differentiate between the HKEY_USERS hive and the HKEY_CURRENT_USER hive. HKEY_CURRENT_USER contains a pointer that references the current user in HKEY_USERS.

The content of a user's profile, as it is found in the HKEY_USERS hive, is interesting. For example, the following keys are present in a typical user's profile (usually, there is nothing to guarantee that they will all be present, or that others might not be added):

AppEvents: Contains information about events (an event is an action like closing, minimizing, restoring, or maximizing) in a key called EventLabels. This information includes a text label for the event, such as the label "Close program" for the event close. These labels are used for a number of purposes, but one that most of us see is in the Control Panel's Sounds applet. A second section in AppEvents is Schemes, which lists labels for each application that uses specific sounds for its own events.

Console: Contains the default command-prompt configuration. This configuration may be customized for each command prompt individually, or it is possible in this key to change the global default, which would be used for all new command prompts that are created. For an example of command-prompt customization, open a command window and select Properties from the System menu. There are more settings that may be configured in the registry than are found in the Properties dialog box.

Control Panel: Contains information saved by many of the Control Panel's applets. Typically, these are default, or standard, values that are saved here, not user settings, which are stored elsewhere.

Environment: Contains the user environment variables for a user. Generally, the System Properties applet, Environment tab, is used to set user and system environment values.



EUDC: Contains the definitions and other information about End User Defined Characters (EUDC). The program eudcedit.exe lets users edit/design characters that are specific to their needs.



Identities: Contains the information to link users and software configurations. Most configurations are Microsoft based, such as Outlook Express.

Keyboard Layout: Contains the keyboard configuration. Most users, at least those in the U.S., will have few or no substitutions. However, users who are using special keyboards or non-U.S. English keyboards will have some substitutions for special characters found in their languages.

Network: Contains mappings for each network drive connected to the computer. Information about the connections includes the host (server), remote path, and username used for the connection. The Network key is not typically found in the .DEFAULT key because users with no user profile are not automatically connected to a remote drive.

Printers: Contains mappings for each remote (network) printer connected to the computer. Information about the printer connection includes the host (server) and the DLL file used to manage the connection. The Printers key is typically not found in the .DEFAULT key because users with no user profile are not automatically connected to a remote printer.



RemoteAccess: Contains the various remote access configurations. The connections are managed using the Control Panel's Network and Dial-up Connections applet.

Software: Contains information about software installed, including components such as Schedule, Notepad, and so on. Also included in Software is Windows 2000 itself, with configuration information specific to the currently logged-on user.

System: Contains information about items such as backup configurations and files that are not to be backed up.

UNICODE Program Groups: Contains information about program groups that use Unicode. More commonly found on computers configured for languages other than English, Unicode is the scheme for displaying characters from both English and non-English alphabets on computers.



Volatile Environment: Contains information about the logon server that will be placed in the environment. One typical item is the *logonserver* environment variable. All items in Volatile Environment are dynamic; that is, they are created each time a user logs on. Other dynamic environment information might be contained in this key as well.

HKEY_CURRENT_CONFIG: The Current Configuration Settings

The registry hive HKEY_CURRENT_CONFIG is created from two registry keys, HKEY_LOCAL_MACHINE\System and HKEY_LOCAL_MACHINE\Software. As it is created dynamically, there is little value in modifying any of the objects found in the HKEY_CURRENT_CONFIG hive.

The HKEY_CURRENT_CONFIG hive is composed of two major subkeys:

Software: Contains current configurations for some software components. A typical configuration might have keys under Software for Microsoft Internet Explorer, for example.

System: Contains information about hardware. The most common device found in this key is the video display adapter (found in virtually all configurations) and sometimes information about the default video modes as well. The video mode settings contained here are typical for any video system: resolution, panning, refresh rates (didn't you wonder where refresh rates were saved?), and BitsPerPel (color depth).

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Generally, you would modify the source settings for a hardware device in HKEY_LOCAL_MACHINE\System\ControlSet001\Hardware Profiles\Current\System\CurrentControlSet\Services\\Device0, where <device> is the device being modified. For example, my Matrox Millennium is listed under the device name MGA64.



NOTE For more information about the source for HKEY_CURRENT_CONFIG, take a look at HKEY_LOCAL_MACHINE, described earlier in this chapter.

HKEY_PERFORMANCE_DATA: The Performance Monitor Settings

Ever wonder where the Windows 2000 Performance Monitor information is contained? Since Windows 95/98 uses HKEY_DYN_DATA to store performance data, and Windows 2000 does not, the performance data must be somewhere. There is a final “hidden” registry hive, named HKEY_PERFORMANCE_DATA. This hive, which is simply not accessible except to applications written specifically to access performance data, is primarily dynamic in nature. To find the answer to this question, check out Chapter 11, “The Performance Monitor Meets the Registry.”

NTUSER: The New User Profile

Windows 2000’s installation process will create a default user profile and configuration. This information is located in %systemdrive%\Documents and Settings\Default User. Whenever a new user logs onto a workstation or domain, this default user profile is copied to the user’s profile. After that, the user modifies their profile to their own requirements and needs.

As an example, Windows 2000’s default language is typically U.S. English. (There are other language editions of Windows 2000; for this example, I’m assuming the U.S. English version.) Whenever a new user logs on, the user will have U.S. English as his or her language, even if the system administrator has selected a different, non-English locale.

The default user profile is saved in the disk directory at \Documents and Settings\Default User [WINNT], where WINNT is the directory that Windows 2000 has been installed in. In Windows NT 4, the default user information was stored in %systemroot%\Profiles\Default User. User information is always saved in a file named NTUSER.DAT. There is an entire configuration for new users in this directory—check out the Start menu, Desktop, and other directories, too. You will find that interesting modifications can be made that enable new users to become proficient quickly without spending too much time customizing their computers.



Desktop Library
Click to access!



WARNING This technique is an advanced use of the registry editor and you must exercise care not to inadvertently modify the wrong registry or the wrong keys. Back up the registry *before* doing the following.

First, to make this new user profile accessible to remote users, (that is, all users other than those who log on locally) you must copy the Default User directory to the share named Netlogon. This share is typically located in the directory at %systemroot%\sysvol\sysvol, in a directory that is named for the server. (For Windows NT 4 users, look in %systemroot%\System32\Repl\Import.) One way to copy these files is to create a new custom profile and copy the new custom profile using the User Profiles tab in the Control Panel's System applet.

If there are BDCs (Backup Domain Controllers), you would actually edit the file in the Export directory (same initial path) because this directory is locally replicated to the Import directory and to the other BDC Import directories, although it might be located elsewhere. The NetLogon share can be located quickly by typing the following command:

```
net share
```

at a command prompt. The computer's shares will be displayed.

Follow these steps to modify the default new user profile in your new Default User directory (remember to create a new Default User directory, saving the current Default User directory as a backup):

1. Start RegEdt32 using either a command prompt or the Start menu's Run command. Don't use RegEdit for this process, as RegEdit is unable to load the NTUSER.DAT file.
2. Click on the title bar of the HKEY_USERS on Local Machine window to make the window active.
3. Choose Registry Load Hive from the RegEdt32 menu.
4. Open the hive found in %systemroot%\Profiles\Default User or %systemdrive%\Documents and Settings\Default User. This hive has the filename NTUSER.DAT.
5. RegEdt32 will prompt for a new key name. Use the name **NTUSER**.
6. Change whatever keys in NTUSER need to be modified. There will be a slew of changeable items in the new profile, including AppEvents, Console, Control Panel, Environment, Keyboard Layout, Software, and Unicode Program Groups. When adding new keys, do be careful to ensure that all users have at least read access to the new keys. No read access means that the key won't be accessible to the user.



NOTE To set the permissions for a key, select the key, and then select Security Permissions from the RegEdt32 menu. Ensure that the group Everyone has at least read access. Resist the urge to give everyone more than read access to this key, too. Too much power can be a dangerous thing!

7. After making all modifications to NTUSER, choose Registry Unload Hive from the RegEdt32 menu.
8. Exit RegEdt32.

Once this profile is saved in the NetLogon share location, new users will get this new profile each time they log on.

Hints and Kinks from the Experts

Another installment of good stuff from the Windows gurus. In your search for more information, frequent their sites on the Internet (see Appendix C, "Where Can I Get More Help?").

How Can I Tell What Changes Are Made to the Registry?

Using the regedit.exe program, it is possible to export portions of the registry. This feature can be used as follows:

1. Start the registry editor (RegEdit.exe).
2. Select the key you want to monitor.
3. Select Registry Export Registry File.
4. Enter a filename (if you want to export the whole registry, select Export Range All) and click OK.
5. Perform the change (install some software or change a system parameter).
6. Perform steps 1–4 again using a different filename.

7. Run the two files through a comparison utility such as windiff.exe.
8. If you are using WinDiff, select File \emptyset Compare Files, and you will be prompted to select the two files to compare.
9. Once compared, a summary will be displayed stating whether or not there are any differences. To view the changes, double-click the message.
10. Press F8, or select View \emptyset Next Change, to view the next change. You have now found what changed.
(Courtesy of John Savill.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 4**Registry Tools and Tips: Getting the Work Done****FEATURING:**

<i>RegEdit</i>	72
<i>RegEdt32</i>	85
<i>Backup's Emergency Repair Disk features</i>	104
<i>The Windows 2000 Resource Kit</i>	106
<i>Hints and kinks from the experts</i>	118

There are a number of excellent registry tools for users of Windows 2000. First, Microsoft provides two registry editors as part of Windows 2000: RegEdit and RegEdt32. Two—did I say *two* registry editors? Why do we need two of them, anyway? Actually, each registry editor has its advantages and disadvantages. RegEdit is the registry editor created for Windows 95. This editor offers a few functions that RegEdt32 doesn't, such as importing and exporting registry files and pretty good search capabilities.

RegEdt32 is the "native" Windows 2000 registry editor. An MDI (multiple document interface) application, RegEdt32 displays each of the main hives in the registry in its own window. RegEdt32 has powerful administrative tools that RegEdit doesn't support, including read-only mode and security configuration, that allows you to restrict access to some registry hives, keys, and subkeys.

Both of these registry editors are valuable. I use either one depending on what I am doing, and my mood. I find RegEdit easier to use, while RegEdt32's got much more power. Not included with Windows 2000, the Windows 2000 Resource Kit offers a number of excellent tools, too. The registry tool, REG, is run at the command prompt. REG allows flexible manipulation of the registry, replacing earlier versions of a number of the other Resource Kit components.

If you are still using older Resource Kit components in legacy support systems, there is no urgent need to change or migrate to the newer tools that are contained in the Windows 2000 Resource Kit. However, it is not recommended that the older utilities be used when updating support facilities, but that the new tools be



integrated wherever possible.

Many of the Resource Kit utilities are command-prompt driven. However, being experienced users, we are not afraid of a command prompt, are we?



TIP Found a program you don't know about? When in doubt, enter a command with either no options or a `/?` option, and the command should display some form of help. Not all commands display significant help, and some do not provide any help at all. However, the Resource Kit utilities won't cause damage if this help convention is used.

RegEdit

RegEdit is the Windows 95/98 registry editor. Microsoft was smart enough to make RegEdit work well with Windows 95/98 and Windows 2000, allowing Windows 2000 users the ability to choose between registry editors. One of the nicest things about RegEdit is its simplicity. A quick user interface, easy to understand options, and a clean, uncluttered look make RegEdit a favorite for many users.

Registry Changes Are Permanent!

All changes made with RegEdit are immediate and, for all intents, permanent! Though you can go back and manually undo a change made with RegEdit, everything that you change with RegEdit affects the current registry. Unlike RegEdt32, RegEdit does not have a read-only mode. There is no safety net, nothing to catch your bloopers, and generally you'll have to clean up your own mess.

In other words, you are editing the real, working, live, honest-to-goodness registry—not a copy. There is no Save command in RegEdit; you type in a change, and it is saved right then and there.

So, make sure you have a backup of the registry files before diddling with registry.

Using RegEdit

Using RegEdit is as simple as starting it. From a command prompt, typing **regedit** will start the program. You can also select Start Run, type in **RegEdit**, and click the OK button to start RegEdit.

Once started, RegEdit will display the current registry (see Figure 4.1). By default, RegEdit opens the local registry. However, it is possible to open a registry on a remote computer by selecting Registry Connect Network Registry and entering the name of the computer whose registry is to be opened.

Making RegEdit Do What It Used to Do!

A change made to RegEdit in Windows 2000 is the display of the last open key from RegEdit's previous session. Some users like this feature; others do not. Currently there is no easy way to disable this functionality, though perhaps Microsoft will give us the option to do so at a later time. Until that time, try this to disable the feature:

1. Using RegEdt32 (you cannot use RegEdit to do this), open `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit`.
2. Edit the LastKey value, and change its contents to an empty string.
3. Select the key RegEdit.
4. Select Security Permissions.
5. Uncheck the Full Control permission for every user in the list.

This prevents RegEdit from saving a value in this key. (Note that this also prevents RegEdit from saving any defaults or favorites.)

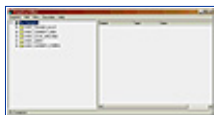


FIGURE 4.1 RegEdit opens the current, local registry automatically.

RegEdit has a straightforward set of menus. The Registry menu allows you to save and load text-based .reg (registry) files, connect to and disconnect from a network registry, and print the current branch or the entire registry.

One improvement made to RegEdit in Windows 2000 is the addition of a new column in the right-hand display of values and data. This new column lists each value's type. Although RegEdit will display the names of all the data types available to Windows 2000, the user is still restricted to editing string, binary, and DWORD data types.



Also, new to the Windows 2000 version of RegEdit is the Favorites menu. In this menu, you are able to place subkeys into a list of favorites. This gives you the ability to quickly navigate to a subkey.

The Edit menu allows the user to create a new key or value entry. Because data types in RegEdit are restricted to string, binary, and DWORD, if you need to create a registry data type unique to Windows 2000, you'll have to use RegEdt32. The Edit menu also allows the user to delete an object, rename a key or subkey, or copy a key name to a new name. At the bottom of the Edit menu are Find and Find Next options.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

Importing and Exporting Registry Hives and Keys

The ability to export a registry hive or key (or the entire registry, if necessary) is a powerful feature of RegEdit. Once a registry is open, select a hive or key (or My Computer to export the entire registry) and choose Registry \emptyset Export Registry File to open the Export Registry File dialog box (see Figure 4.2).



NOTE The typical Windows 2000 registry will be several thousand to hundreds of thousands of lines long. The registry on my server has over 130,000 lines. At 66 lines per page, there would be at least 2,000 pages in the report. At least, you say? Yes, many registry lines would require more than one line to print, so it would actually be much more than 2,000 pages.

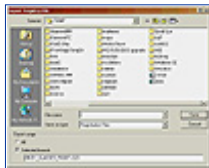


FIGURE 4.2 Exporting the currently selected hive or key is easy!

A hive is exported into a Unicode text-based file. This file has no comments; some of the Resource Kit registry tools do comment exported sections of the registry. However, the file may be opened with most any text editor (such as Notepad), searched, and even (carefully) modified. Any changes made to the exported text file may be incorporated into the registry by simply importing the modified file.

Importing a file that RegEdit had previously exported is as simple as selecting Registry \emptyset Import Registry File and entering the name of the registry file to import.

What Is an Exported Registry File?

A registry file exported by RegEdit will start with a line that reads “Windows Registry Editor Version 5.00.” The line following will be the first hive exported in a hierarchical format:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE]
[HKEY_LOCAL_MACHINE\Hardware]
[HKEY_LOCAL_MACHINE\HARDWARE\Description]
```

Generally, a full export of a registry will start with an export of the HKEY_LOCAL_MACHINE hive, as the above example shows.

The contents of an exported registry are arranged in the file as a hive and key combination (fully qualified, enclosed in brackets), with the data value name in quotes and the value following the equal sign:

```
[HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\FloatingPointProcessor\0]
"Component Information"=hex:00,00,00,00,00,00,00,00,00,00,00,00,00,00,01,00,
00,00
"Identifier"="x86 Family 5 Model 4 Stepping 3"
"Configuration Data"=hex(9):ff,ff,ff,ff,ff,ff,ff,ff,ff,00,00,00,00,00,00,
00,00
```

This example shows the three value entries that FloatingPointProcessor contains.

Why export the registry? First, none of the search capabilities in any of the registry editors is optimal. (Well, that’s my opinion!) Loading an exported registry file into an editor allows you to quickly search for strings using the editor’s search capability.

Another benefit is that it is easy to export the registry before installing an application or system extension. After an installation, it is also a good idea to export the registry. Then, using one of the system comparison tools (such as FC or WinDiff), you can compare the two versions of the registry and see what the installation has changed. Bingo—a quick way to see what’s happening to the registry on installations.

Connecting to and Disconnecting from Remote Registries

When RegEdit starts, the local registry is opened. Once started, it is then possible to open the registry on a remote computer (see Figure 4.3). Actually, it is possible to connect to many remote registries at one time. With a remote registry connected, you may not close, or disconnect, the local registry. This leaves it up to the user to make sure that if changes are made, they are made in the correct registry.



NOTE The computer with the remote registry must be on the network, have remote administration enabled, share x\$ where x is the drive that the registry is on (for Windows 2000/NT computers only), and be running the remote registry service.

Hopefully, this will make remote registry maintenance somewhat easier. Some functionality doesn’t span multiple registries (such as searching), but generally everything that may be performed on a local registry may be also performed on a remote registry.



TIP Once finished with a remote registry, it is a very good idea to disconnect from it. This may help prevent unexpected modifications to the wrong registry.



FIGURE 4.3 A remote registry and the local one open at the same time in RegEdit

Printing the Registry

Printing a registry hive or key is possible in RegEdit. As mentioned previously, printing an entire registry is not a swell idea—you’d have to make a major investment in paper and printer supplies. Typically, a registry

would require thousands of pages to print.

Printing sections of a registry hive can be very useful if a paper record is needed, or if you need something to take to a meeting. The limit of a printed registry hive or key is that searching it might be difficult.


Printing is easily done if you select the hive or subkey to print, then select Registry  Print from RegEdit's main menu. The Print dialog box, shown in Figure 4.4, allows you to edit the branch to be printed (with the currently selected object as the default). The results of printing a registry report are almost identical to exporting, with the exception that a printed report lacks the initial header line that's found in an exported registry file.



FIGURE 4.4 RegEdit's Print dialog box printing the hive HKEY_CLASSES_ROOT



TIP Is it very readable? Generally not. The RegEdit print facility is basic and simply wraps lines at 80 characters. Any line more than 80 characters long will wrap and be difficult to read. A better solution is to export the registry to a file, load the file into a word processor, format it so that it is readable, and print it from the word processor.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Creating, Deleting, and Renaming Entries

RegEdit allows you to quickly create, delete, or rename an entry. Entries may consist of keys, subkeys, or value entries.

Creating a New Key You can quickly create a new key by following these steps:

1. Select the hive or key in which the new key is to be created.

Why Can't I Create a Key Here?

Not all hives allow you to create keys directly under the hive itself. For example, it is not possible to create a key under HKEY_LOCAL_MACHINE, though it is possible to create a key under HKEY_CURRENT_USER.

The first question that comes to mind is, "Why?" Simply put, the HKEY_LOCAL_MACHINE hive is not "saved" when Windows 2000 shuts down. Rather it is re-created anew each time Windows boots—therefore, any key or subkey created would be lost at the next boot-up time.

2. RegEdit will create the new subkey, giving it a default name of New Key #*n* where *n* is a number beginning with 1. You will have the opportunity to edit the new subkey's name, which you should do at this time. Give the subkey a meaningful name or the name that is expected for this subkey.

Once the new subkey has been created, it may then be populated with additional subkeys and value entries.



NOTE A hive, key, or subkey may contain both value entries and other subkeys at the same time.

Creating a Value Entry You can quickly create a new value entry by following these steps:

1. Select the hive or key in which the new value entry is to be created.
2. Select Edit ∅ New and then either String Value, Binary Value, or DWORD Value, depending on the type of data that this value entry will have. You should select the data type for your key.
3. RegEdit will create the new value entry, giving it a default name of New Value #*n* where *n* is a number beginning with 1. You will have the opportunity to edit the new value entry's name, which you should do at this time. Give it a meaningful name or the name that is expected for it. Press the Enter key to save the new name.



Click to access!



TIP At any point, you may rename a key or value entry by right-clicking the item to be renamed and selecting Rename from the menu.

- To enter data into the new entry, double-click the entry. The correct edit box will be displayed, allowing you to edit the data.

Once the new value entry has been created, its data may be entered as necessary.



NOTE A key need not have a data value entered. A key is valid without any data, though no-data defaults vary depending on the type of data the key contains: String values have a zero-length string as their default. Binary values have a zero-length binary value (which is different from having a value of zero). DWORD values have a value of zero.

Figure 4.5 shows RegEdit with a new subkey containing another subkey, a string value, a binary value, and a DWORD value, exactly as created by RegEdit. Note that I've named the initial subkey Test Key.

In the example, each of the three new value entries were given names to match the type of data stored in them. Each one can be edited at any time. Selecting the key or value entry and choosing Edit ▾ Rename allows you to change the name; selecting the value entry and choosing Edit ▾ Modify allows you to change the value entry's contents. You may also double-click the value entry or right-click (also known as a *context-click*) on the item and choose Modify to change the value.



FIGURE 4.5 RegEdit after creating the subkey called Test Key and a further subkey called Test sub-key

Copying Key Names

Is this as simple as it seems? A long, convoluted name without having to type it? Yes, it is!

Copy Key Names, found in RegEdit's Edit menu (and from the key's context menu if you right-click the key and select Copy Key Name), will copy the key's name to the Clipboard. The information is copied in text format and may then be pasted into other applications or word processors as needed. For example, when the new key created for Figure 4.5 is copied, the following text will be placed into the Clipboard:

```
HKEY_LOCAL_MACHINE\Hardware\Description\System\Test Key\Test sub-key
```

This means it is not necessary to manually type in long registry keys into other applications and documents. This feature, for example, was a great help when writing this book.

Searching: Find and Find Next

Searching a registry is one of the most important tasks you'll have to undertake. Before you make a modification, do debugging, or start browsing, it is usually necessary to search for something.

Now, as I've mentioned previously, RegEdit's search capabilities are a bit limited. However, RegEdit does have the best search of the two registry editors, as RegEdt32 is even more limited in its searching capabilities.



TIP RegEdit searches downward only. If what you are searching for is located above the current selection, you'll never find it. When in doubt, start at My Computer, and you can be assured that the search will include the entire registry.

Searching allows you to look at keys, data value names, and data value contents. You may choose to search any or all of these (see Figure 4.6), and the search may also be limited to whole strings only, which applies to searching text strings exclusively.

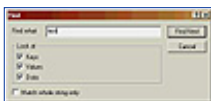


FIGURE 4.6 RegEdit searches downward only.



NOTE RegEdit's search is not case specific, so strings to be searched for may be entered in lowercase if desired. This is nice, since the case of many registry entries is rather mixed.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Once the search finds the item searched for, it will stop on the word(s) found. If RegEdit's search is unable to find the string entered, RegEdit will display an error dialog box. Use F3 to continue the search.

Using RegEdit from the Command Line

RegEdit may be used from the command line, without user interaction. The commands that RegEdit uses include those described below. (Note that not all commands may be available under all operating systems.)

To import a registry file into RegEdit, use the command:

```
REGEDIT [/L:system] [/R:user] filename1
```

To create a registry object from a file, use the command:

```
REGEDIT [/L:system] [/R:user] /C filename2
```

To export a registry (or part of the registry), use the command:

```
REGEDIT [/L:system] [/R:user] /E filename3 [regpath1]
```

To delete part of a registry, use the command:

```
REGEDIT [/L:system] [/R:user] /D regpath2
```

In all the above commands, the parameters are:

/L:system	Specifies the location of the system.dat file. Note that there is a colon between the /L and the parameter system.
/R:user	Specifies the location of the user.dat file. Note that there is a colon between the /R and the parameter user.
filename1	Specifies the file(s) to import into the registry.
/C filename2	Specifies the file to create the registry from. Note that there is a space between the /C and the parameter filename2.
/E filename3	Specifies the file to export the registry to. Note that there is a space between the /E and the parameter filename3.
regpath1	Specifies the starting registry key to export from (defaults to exporting the entire registry).



JUMP TO TOPIC

/D regpath2

Specifies the registry key to delete. Note that there is a space between the /D and the parameter *regpath2*.



WARNING Be careful; be very careful. Running RegEdit in the command-line mode can be damaging to the registry—it is possible to utterly destroy the registry with a single command.

Tips for RegEdit Users

RegEdit offers the ability to export the registry, or parts of the registry, to a text file. This file may be used for any of the following:

- A snapshot of the condition of the registry at the time the export was made.
- A limited backup of the registry that might have value in restoring the registry in the event there is a failure.
- A file that when compared with another export file, using FC or WinDiff, can quickly show differences between the two versions of the registry.
- A file that can be edited using any text editor (Notepad, for example); the results of the editing could then be incorporated into a registry using RegEdit's import facilities.

RegEdit allows adding simple keys, subkeys, and values (with limited data types) to any registry. Though RegEdit won't add, edit, or create the more complex data types that Windows 2000 supports (such as data types REG_MULTI_SZ, REG_FULL_RESOURCE_DESCRIPTOR, and so on), much work with the registry is done using the simple character and numeric data types.

Installing Remote Registry Editing on Windows 95 and Windows 98

Though Windows NT Workstation and Windows 2000 Professional have remote registry editing installed already, Windows 95 and Windows 98 do not. The installation process is similar on both operating systems, though the source of the necessary drivers differs with each version.

You have to install a network service to enable remote registry editing. This service, REGSERV, is found in the following location:

- Windows 95: Look on the Windows 95 distribution CD, in the directory `\tools\reskit\netadmin\remotreg`, for the regserv program files.
- Windows 98: Look on the Windows 98 distribution CD, in the directory `\admin\nettools\remotreg`, for the regserv program files.

In each operating system, the installation is identical:

1. Open the Control Panel.
2. Start the Network applet.
3. Click the Add button in the Configuration tab.
4. Select Service from the list, and click the Add button.
5. Click the Have Disk button, and provide the directory information as given above.
6. Select Microsoft Remote Registry.
7. Install the Remote Registry service, and reboot the computer when prompted.



TIP The Remote Registry service files are identical in Windows 95 and Windows 98. Either will work with either version of the operating system.

RegEdt32

RegEdt32 is specifically designed for the Windows 2000/NT registry. (RegEdit was actually designed for the Windows 95/98 registry.) Unlike RegEdit, RegEdt32 works as an MDI program, opening each of the main hives in the registry in a different window. This application model offers both a good side and a bad side. Each hive is kept separate, which means you don't inadvertently move from one hive to another, especially when doing searches; but at the same time, searching, can be difficult because things are not always where one expects them.

RegEdt32 is actually more powerful than RegEdit. Additional functionality includes management of security (noticeably missing on Windows 95/98 machines), more control of how data is displayed, more options, and the ability to add more data types to the registry.

Using RegEdt32

Using RegEdt32 is not difficult. You can start it from a command prompt or from the Windows 2000 Start menu's Run command. Like you can with any other command or program, you can add a shortcut to the Start menu for RegEdt32, although I don't recommend doing so. RegEdt32 is a dangerous tool that should not be too easy for the uninitiated user to have access to.

When started, RegEdt32 opens the local registry. It is possible to open the registry on a remote computer as well; however, when editing a remote registry, not all hives will be available.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Opening and Closing Registries

RegEdt32 allows you to open and close both the local and remote registries. To open the local registry, select Registry \emptyset Open Local. This will open the local registry without further ado. To close the currently selected registry, which may be either the local registry or a remote registry, click on any window in the registry to be closed, then select Registry \emptyset Close.



NOTE RegEdt32 allows you to close all registries so that nothing is open. If this happens, you can either restart RegEdt32 or select Registry \emptyset Open Local to open the local registry.

Remote Registries

To open a remote registry, select Registry \emptyset Select Computer. The Select Computer dialog box that appears (see Figure 4.7) shows all computers accessible on the network. Select the computer whose registry is to be opened and click OK, or just double-click it. The registry hives that can be opened on the remote computer will be displayed, and a warning dialog box will advise you if not all hives could be opened.



NOTE Not all remote computers will allow you to edit, or open, their registries remotely. Windows 95/98 machines must specifically authorize remote registry editing (Windows NT systems do this automatically). See "Installing Remote Registry Editing on Windows 95 and Windows 98," above, for more information on enabling remote registry editing on these two operating systems.

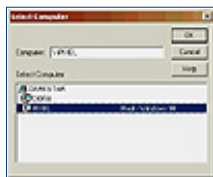


FIGURE 4.7 RegEdt32 is ready to open the registry on the remote computer Pixel.



WARNING As with a local registry, any changes made to a remote registry take effect immediately. There is no saving, no tossing the changes away, no just closing RegEdt32 and losing the changes. Once entered, the change takes effect, with possible disastrous results for remote users who may not even realize that their registry was modified.

Loading and Unloading



RegEdt32 allows a subkey to be loaded into the current registry. This subkey may be modified and later unloaded. Why?

There are several reasons for loading and unloading keys into RegEdt32. A classic example is given in Chapter 3. This example, configuring a modified new user profile, concerns the file NTUSER.DAT. In NTUSER.DAT is the HKEY_CURRENT_USER hive. Within this hive are settings, such as internationalization, colors, schemes, and other items. Windows 2000's installation process will create a default user profile—nothing spectacular, a very plain configuration. Whenever a new user logs onto a workstation (or domain), this default user profile will be copied to the user's profile. After that, the user may modify this default profile to his or her requirements and needs. Of course, you might want to establish some organizational defaults, such as a company scheme.



WARNING The techniques shown next are advanced use of the registry editor. Back up the registry *before* doing the following.

The default user profile is saved in the following disk directory:

- For new Windows 2000 installations: %sysdrive%\Documents and Settings\Default User (this directory may have the hidden attribute set, so that it is not displayed when using either Explorer or a command session)
- For Windows NT 4, and Windows 2000 installations that are upgraded from Windows NT 4: %systemroot%\Profiles\Default User\

The name of the user profile is NTUSER.DAT. There is an entire configuration for new users in the directory %sysdrive%\Documents and Settings\Default User; check out the Start Menu, Desktop, and other directories, too. You will find that interesting modifications can be made that enable new users to become proficient quickly without spending too much time customizing their computers.

First, to make this new user profile accessible to remote users (users other than those who log on locally), you must copy the Default User directory to the share named NetLogon. This share is typically located in the directory at C:\WINNT\sysvol\sysvol.

Placing files in Export will cause replication to copy them locally to Import, along with any BDCs (Backup Domain Controllers). Note that the share might be located elsewhere. The NetLogon share can be located quickly by typing the following command at a command prompt:

```
net share
```

The computer's shares will be displayed.

One process to copy these files is to create a new custom profile, and then copy the new custom profile using the System applet's User Profiles tab.



WARNING If you are even slightly smart, you'll make a backup copy of the NTUSER.DAT file *before* you make any changes in it!

Do the following to modify the default new user profile. (Remember to create a new Default User directory, saving the current Default User directory as a backup.)

1. Start RegEdt32 using either a command prompt or by selecting Start ⌘ Run. Don't use RegEdit for this process.
2. Click on the title bar of the HKEY_USERS on Local Machine window to make it active.
3. Choose Registry ⌘ Load Hive from the RegEdt32 menu.
4. Open the hive file in %sysdrive%\Documents and Settings\Default User. (If your system is configured, or installed, with different directory names, choose the correct name.) This hive has a filename of NTUSER.DAT.

Can't Find the Location for the NTUSER.DAT?

Remember that the NTUSER.DAT file will have the hidden attribute, so it will not normally be displayed in either a command window or in Explorer. Either tell Explorer to display hidden files or, at a command prompt, use the **dir** command with the **/ah** option to display hidden files and directories.

Worst comes to worst, open a command window (tough to do this in Explorer) and, in the root of the system drive, use the command:

```
DIR /ah /s ntuser.dat
```

This command will list all copies of the NTUSER.DAT file, allowing you to change the appropriate one. One thought though: don't change the "current user" NTUSER.DAT file—it won't work! Windows will rewrite the file when the user next logs off, causing any changes you made to disappear!

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

5. RegEdt32 will prompt for a new Key Name. Use the name NTUSER.
6. Change whatever keys in NTUSER need to be modified. There will be a slew of changeable items in the new profile, including AppEvents, Console, Control Panel, Environment, Keyboard Layout, Software, and Unicode Program Groups. When adding new keys, do be careful to ensure that all users have at least read access to the new keys. No read access means that the key won't be accessible to the person named "user."



TIP To set the permissions for a key, select the key, and then select Security \emptyset Permissions from the RegEdt32 menu. Ensure that the Everyone group has at least read access. Resist the urge to give everyone more than read access to this key. Too much power can be a dangerous thing!

7. After making all modifications to NTUSER, choose Registry \emptyset Unload Hive from the RegEdt32 menu. Unload the hive to the file NTUSER.DAT. (You did back up the original file, right?)
8. Exit RegEdt32.

Once this profile is saved in the Netlogon share location, each time a new user logs onto the network, the user will get this new profile.

Save Subtree As

The Save Subtree As command, chosen from the Registry menu, allows you to save a hive, key, or subkey, and all descendants' contents. The data is saved in text format and may be edited with an editor such as Notepad. Information about the hive and keys is found in the saved file. Extensive data about the registry hive, including the date the registry file was last written to, is also saved in this file. Relatively easy to read, the file's size will be many, many times larger than the registry, as the small example in Figure 4.8 shows.



NOTE Just a suggestion: Don't print these files without first determining just how long they are. For me, saving a simple object generated almost seven hundred lines of output.



FIGURE 4.8 An example of the Save Subtree As file output



Save Key

The Save Key command allows you to save a hive, key, or subkey and the saved object's contents. RegEdt32 writes the data saved in a binary format; no hacking or editing is allowed here. This file may be later reread using the Restore option in the Registry menu. The file is saved to the filename and extension you specify, unlike with RegEdit, which automatically uses an extension of .reg.



NOTE Actually, to save an object, you must have sufficient privilege to read the entire object. If you do not have this privilege, you will get an error message, "Insufficient privilege to save the key." Use the Permissions option in the Security menu to alter permissions if this becomes a problem.

Information saved by Save Key contains unqualified hive and key information. For example, if you save the HKEY_LOCAL_MACHINE\Hardware\Description\System key, the only name saved to the file will be System. The HKEY_LOCAL_MACHINE component of the object's name will *not* be saved. Again, think about how file and directory names are sometimes either fully qualified or not.

Restoring

Restoring is what Joe and Ed on the Learning Channel do to old furniture, right?

Well, maybe so, but it's also possible to restore an object in the registry using RegEdt32. The process is straightforward, although like everything else, you must have something to restore from. As explained just above, using Save Key (in the Registry menu), you can save a registry object to a file. Since the file extension is user determined, it will be a really good idea to keep filenames as descriptive as possible.

A suggestion: If you have a strong desire to play with the save and restore functionality of RegEdt32, install a practice copy of Windows 2000. Don't do this on a working version—at least not a copy of Windows 2000 that you, or anyone else, care about.



NOTE When an object is restored, the data overwrites the existing object. It becomes permanent, as everything that RegEdt32 does is immediately written to the registry.



WARNING More important—when an object is restored, it is written on top of the currently selected object. Make sure that the object you are restoring belongs at the current selection. Again, make sure you name your file well so that you know exactly which object a given file represents. Imagine coming back to a saved file, perhaps weeks later, and trying to restore it without knowing which object it was saved from.



WARNING Even *much* more important: Restoring an object may override the read-only mode option—it will write to the registry no matter what! Care to guess how I found that out?.

When an object is restored, the selected object itself will not be renamed, even though the contents of the object will be replaced.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Printing the Registry

In RegEdt32, if you select Registry \emptyset Printer Setup, this will take you to a standard printer configuration dialog box. Printing subkeys of the registry with RegEdt32 results in a much nicer printout than you get with RegEdit. The printout consists of virtually the same data that the Save Subtree command in the Registry menu saves. An example of what a printout might look like is shown in Figure 4.8, earlier in this chapter.



TIP Be careful not to select too much to print. RegEdt32's printout capabilities can generate massive reports..

Adding Keys and Subkeys

Adding keys, subkeys, and value entries is easy enough with RegEdt32. First, make sure the registry is not in read-only mode (see "Setting Options," later in this chapter). Next, select the location where the new subkey or key is to be located. The RegEdt32 Edit menu then provides the tools to add the new item.

Adding Keys and Subkeys You can create a new key by selecting Edit \emptyset Add Key. The Add Key dialog box (see Figure 4.9) allows you to specify the name, and optionally, a class.



FIGURE 4.9 Adding subkeys with the Add Key dialog box

A subkey may be assigned a class. The class may be any of the following:

- REG_BINARY
- REG_SZ
- REG_EXPAND_SZ
- REG_DWORD
- REG_MULTI_SZ
- A user-supplied string describing the key's type

However, since subkeys are not normally assigned a value, the class attribute is rarely used, and often it is left blank.

Adding Values and Data You can add a new value entry, with a specified data type, by selecting Edit \emptyset



Desktop Library
Click to access!

Add Value. The Add Value dialog box (see Figure 4.10) requires that you enter a value name and select a data type.



FIGURE 4.10 Adding a value with the Add Value dialog box

Unlike a key, a value *must* have a data type. Valid selections for the data type are:

- REG_BINARY
- REG_SZ
- REG_EXPAND_SZ
- REG_DWORD
- REG_MULTI_SZ

The specialized types, such as REG_FULL_RESOURCE_DESCRIPTOR, may not be added or created using RegEdt32's Add Value dialog box, shown in Figure 4.10. (I'll leave it up to you to figure out how to create a key with a non-standard data type, with this hint: it can be done, but it's not that easy.)



NOTE OK, I'll tell... Creating a data value with a data type not supported by RegEdit or RegEdt32 is done using RegEdit. Create a .reg file (export the subject key) and edit the .reg file with any compatible text editor. Entries may be created in the format: "entry name"=hex(n):hh hh... Where *entry name* is the name for the data value, *n* is a number corresponding to the type desired, and *hh* is one or more pairs of hex data. Data types are described in Appendix B..

After creating a new key, one of the data value editors appropriate for the data type selected will be displayed (see Figure 4.11).

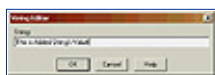


FIGURE 4.11 The RegEdt32 String Editor, just after entering a text string

Once a key has been created, its data may be edited with a different editor. However, the contents may not make much sense when displayed using the wrong format.

Each of the editor choices displays the data in the format for that editor, but does not change the data type of the data—it only allows editing the data using a different format. Were I designing the registry editor, I'd make data editing using all formats in a single dialog the default.

Select a key, and then in the Edit menu, select one of the following choices:

Binary: Edits the item in binary format.

String: Edits the item in string format.

DWORD: Edits the item in the DWORD format. It will truncate objects longer than 4 bytes.

Multi String: Edits the item using the multiline string editor.



WARNING Realize that editing an object that is more than four bytes long with the DWORD editor will truncate the object to four bytes—permanently, irrevocably, without recourse—when you click OK..



NOTE It is not possible to change a key's data type without creating a new key. Start by renaming the key to be changed to a temporary name. Next, create a new key with the original name, using the new data type. Finally, using the Clipboard, edit the original key, copy the key's data to the Clipboard, then edit the new key, and paste the data into the new key..

Deleting the Unwanted Getting rid of the unwanted is easy. Select the object, either a key or a value entry, to be deleted and then either select Edit \emptyset Delete or just press the Delete key. RegEdt32 will prompt you to confirm that the object is to be deleted, if the Confirm on Delete option is selected (see "Confirm on Delete," later in this chapter).



WARNING Once deleted, 'tis gone forever! Be careful not to delete anything that you will want later. Prior to deleting, it's appropriate to back up the registry. It also might be a good idea to rename the object, just in case it is necessary to restore it at a later time..

Searching

The RegEdt32 search is somewhat different from that found in RegEdit. Unlike RegEdit, RegEdt32 is able to search either up or down and match case. Figure 4.12 shows the Find dialog box.

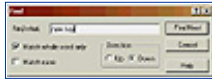


FIGURE 4.12 The RegEdt32 Find dialog box has better search criteria.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

However, there is one big *gotcha*: RegEdt32 is able to search only on hive, key, and subkey names. RegEdt32 is not able to find value names, or data, which is a major limitation if you ask me. (OK, they did ask me, but I didn't say anything at the time.)



NOTE Perhaps it's a bug, but the check box Match Whole Word Only actually forces the search tool to match the contents of the entire name, not just a single word in the name. The check box probably should actually read Match Whole Name Only..

If the search fails (which it usually does for me, since I'm always searching for data names or values, not keys), a small dialog box will tell you that the item was not found. Don't forget to search in both directions if you are in the middle of the registry.

Security

Security is paramount in a Windows 2000 installation. The registry, just like the NTFS file system, can be protected from unauthorized access. This can be a critical issue, because Windows 2000 supports remote registry editing.



NOTE It is possible to make changes in a registry from another computer without the recipient of these changes even knowing that a change has been made (that is, until they see the results of the change)..

RegEdit does not support any security modifications. If a hive is not accessible to RegEdit, the user is unable to view the hive or change it, depending on the level of access granted by the system. However, the RegEdt32 Security menu allows you to change the security attributes for a hive and any keys, if you have sufficient authority to do so.

Initially, when you select Security \emptyset Permissions, the Permissions For dialog box is displayed (see Figure 4.13). Basic security is set in this dialog box, while advanced functionality (permissions, auditing, and owner) is set in the Access Control Settings dialog box.





FIGURE 4.13 Setting the permissions for Test Key in RegEdt32

Clicking the Advanced button of the Permissions For dialog box displays the Access Control Settings dialog box, shown in Figure 4.14. The Access Control Settings dialog box has three tabs: Permissions, Auditing, and Owner.

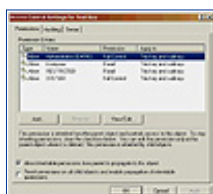


FIGURE 4.14 Specific users and administrative units can have their own permissions.

Permissions The currently selected object is displayed along with the current permissions granted. Default permissions are typically, but not always, ones that everyone can read; the Administrator accounts and the system both have full control.

The Permissions tab will list the object's name in the dialog box's title bar. The "Allow inheritable permissions from parent to propagate to this object" check box allows the current object to include its parent's permissions. The "Reset permissions on child objects and enable propagation of inheritable permissions" check box allows changing permissions for both the selected item and any subkeys it contains.

You set detailed permissions by clicking the View/Edit button in the Permissions tab of the Access Control Settings dialog box (see Figure 4.15). In the list box are the current permissions, organized by name. Select one name (each must be modified separately) and set the type of access. The selections include the following:

Query Value: Allows the selected user to have read access.

Set Value: Allows the selected user to have write access.

Create Subkey: Allows the selected user to create a subkey.

Enumerate Subkeys: Allows the selected user to obtain a list of subkeys contained within the object.

Notify: Tells Windows 2000 to notify the owner when the object is modified.

Create Link: Allows the selected user to create a link to the object from another object.

Delete: Allows the selected user to delete the object.

Write DAC: Allows the selected user to modify Discretionary Access Control information.

Write Owner: Allows the selected user to modify the owner record information.

Read Control: Combines the Standard Read, Query Value, Enumerate Subkeys, and Notify Permissions.



FIGURE 4.15 Permissions are customized on a user-by-user basis in the Permission Entry dialog box.



WARNING Of course, the standard warnings apply: *Do not grant more permission than is necessary to do the job.* Understand what permissions are being granted (see the above list) and consider granting permissions temporarily, removing anything granted as soon as it is not necessary..

Auditing The word *auditing*, when mentioned with the word *government*, generally gets us weak in the knees and starts us sweating profusely. However, auditing registry interaction can be somewhat less troublesome and very beneficial to the user.

Auditing, like permissions, is based on users. You set up auditing in the Auditing tab of the Access Control Settings dialog box (see Figure 4.16). For an object that has not had any auditing set, the list will be blank. The first thing to do is to check “Allow inheritable auditing entries from parent to propagate to this object.” Next, click the Add button to add new users to the list (see Figure 4.17). In the Select User, Computer, or Group dialog box, both groups and individual users can be selected. Select one name in the Name list and click the Add button to add that name to the list of names to be audited. Once all names to be audited have been added, click OK.



FIGURE 4.16 The Auditing tab, in which you set auditing permissions

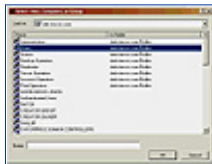


FIGURE 4.17 Add users or administrative units to be audited in this dialog box.

Next, in the Auditing tab, select one of the names in the Auditing Entries list, and click the View/Edit button. Then set specific permissions in the Auditing Entry For dialog box. Events that may be audited include:

Query Value: Audited whenever the user or group in the Name list reads the object.

Set Value: Audited whenever the user or group in the Name list writes to the object.

Create Subkey: Audited whenever the user or group in the Name list creates a key.

Enumerate Subkeys: Audited whenever the user or group in the Name list enumerates a list of keys contained within the object.

Notify: Audited whenever the user or group in the Name list does anything that generates a notification to the owner.

Create Link: Audited whenever the user or group in the Name list creates a link to the object from another object.

Delete: Audited whenever the user or group in the Name list deletes the object.

Write DAC: Audited whenever the user or group in the Name list modifies the Discretionary Access Control information.

Write Owner: Audited whenever the user or group in the Name list modifies the owner record information.

Read Control: Audited whenever the user or group in the Name list does anything that includes the standard Read, Query Value, Enumerate Subkeys, or Notify Permissions.

[Previous](#) [Table of Contents](#) [Next](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

When auditing, success and/or failure may be audited. Either or both may be selected if d

Success: Whenever a successful operation is done, auditing information is saved. This mode is useful when creating a log of information about changes to the registry. Success auditing can help you go back and determine what changes were made to the registry to try to fix the problem.

Failure: Whenever an unsuccessful operation is done, auditing information is saved. Whenever security is an issue (any time there is more than one user), failure auditing can help point to attempts to compromise system security.



TIP Select audit success for critical objects that shouldn't be changed often. Select audit failure for any object that is security related..

Owner I own things, you own things. To keep the records straight, there are things like titles for cars, deeds for property, and other documents that trace ownership of anything that is non-trivial. With computers, especially Windows 2000, ownership is an important thing. I "own" my computer, and probably I don't want you messing with it.

When using NTFS, ownership may be set for files. In addition, objects in the registry may have ownership, too. Ownership implies ultimate control: the owner can restrict access, audit, do whatever he or she wants.

In RegEdt32, the Owner tab in the Access Control Settings dialog box allows you to take "ownership" of a registry object (see Figure 4.18).

The owner of any object may allow or disallow another user from taking ownership; however, once another user has ownership, the original owner's rights are terminated.



NOTE Both the current owner and the system administrator may assign ownership of the object to a user or to the system administrator..

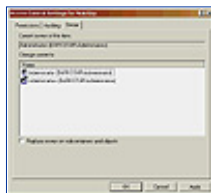


FIGURE 4.18 The Owner tab lists the current owner and allows ownership to be set to the current user.

Setting Options

RegEdt32 has a number of options that you can set through the Options menu.

Font The font used to display all registry information can be changed as necessary. Most often the font you choose is a personal preference, or you might need to change fonts due to multiple language support—for example, you might be running RegEdt32 on one computer and remotely editing a registry on another computer on which the registry has strings that are in a different language or character set.

Selecting Options ▸ Font displays a standard Font dialog box (see Figure 4.19), allowing you to select the font, style (regular, italic, bold, or bold italic), and font size. You can also select the script (language). Choices for script include Western, Hebrew, Central European, and others.

Auto Refresh Auto Refresh is used to update all the RegEdt32 registry windows for the local registry. You can make Auto Refresh active from the Options menu:

- If Auto Refresh is checked, Refresh All and Refresh Active on the View menu will be disabled.
- If Auto Refresh is not checked, Refresh All and Refresh Active on the View menu will be enabled.

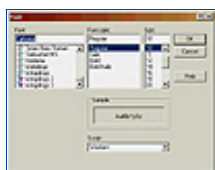


FIGURE 4.19 The font can be a personal preference, or it may support a different language or character set.



NOTE Auto Refresh cannot be used with a remote registry. Even if Auto Refresh is checked, remote registries will not be updated, nor will the Refresh All and Refresh Active selections be enabled in the View menu. (With the View menu refresh options disabled, it is rather difficult to refresh the remote registry!) When editing remote registries, always turn off Auto Refresh..

Read-Only Read-only is used to make the current registry un-editable. This functionality serves as a safety net, making it possible to browse the registry without being able to make any changes to it. When read-only is active, it will be checked in the Options menu.



WARNING Unless specifically changing the registry, it is best to keep Read-Only checked at all times. Making an inadvertent change to the registry can be the beginning of the end of the Windows 2000 installation!.

Confirm on Delete Whenever Confirm on Delete is active, RegEdt32 will notify the user to confirm the deletion with a Warning dialog box (see Figure 4.20). The amount of time lost using Confirm on Delete is minimal compared to inadvertently deleting an object that is necessary to the well-being of Windows 2000.

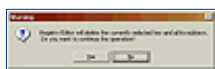


FIGURE 4.20 Confirm on Delete warns you each time you delete an object.

Confirm on Delete is active whenever there is a check next to it in the Options menu. Confirm on Delete is not available when RegEdt32 is in read-only mode.

Save on Exit Options, such as those previously listed, may be either temporary (for the current session) or permanent. When checked, the Save on Exit selection in the Options menu has RegEdt32 save options whenever it is exited.



WARNING Only the settings of options are saved on exit. All changes to the registry are saved as they are made. There is no separate saving of the registry..

Tips for RegEdt32 Users

Several tips come to mind when using RegEdt32:

- First, when saving a hive using Save Key (under the Registry menu), make absolutely sure that the filename saved to is descriptive enough to enable the successful restoration of the hive at a later time.

RegEdt32 doesn't check whether a hive being restored is the same hive as the one being replaced.

- Second, as with RegEdit, be aware that printing can create reports of incredible size. Do not print the entire registry, especially if you are over the age of 22 or so: life is just too short.
- Finally, the RegEdt32 Save Subtree As functionality allows saving a detailed text report, identical to the printed report, to a disk file. This report could then be loaded into a text editor or word processor, allowing editing and printing.

Backup's Emergency Repair Disk Features

RDisk is not available under Windows 2000. When using Windows 2000, the Backup program contains the RDisk functionality. With Windows 2000, the ERD (Emergency Repair Disk) has slightly different contents than under previous versions of Windows NT.

The repair disk is used to hold some of the system configuration components. Backup is capable of backing up registry files to a location on the hard drive (the repair directory) and configuration files to a diskette. The diskette version of this information is called the ERD (Emergency Repair Disk), and it contains files used to help Windows 2000 restore the system to a known state in the event of damage to the working copy of the registry.

Generally, copies of the registry contained in the repair directory are only usable with the Setup program's repair facility. This may seem to limit their usefulness. However, when disaster strikes, anything is better than nothing. Actually, spending half an hour running the Setup repair function is a small price to pay to recover from a damaged registry.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

There can be only one repair directory on a Windows 2000 system, always at %systemroot%\repair. However, there may be many ERDs in existence at one time. Since an ERD's contents are (generally) matched to the registry, it is best to simply keep one or two copies of the most recent registry backed up.



TIP Actually, you can copy the files in the repair directory to another safe location, as well, then copy them back to the repair directory if necessary. Make sure that *all* files are copied or backed up and restored as a set—don't attempt to back up only some of the files in the repair directory. If you copy the repair directory files to another location, also create a copy of the ERD and save that as well..

Creating an Emergency Repair Disk

To create an ERD, follow these steps:

1. Start Backup without any options.
2. Select the Emergency Repair Disk button in the Welcome tab.
3. At the prompt to insert a diskette (see Figure 4.21), insert a diskette containing nothing of value. The diskette must already be formatted.



FIGURE 4.21 Backup will write backup files to the ERD diskette, and optionally to the repair directory.

4. Once Backup is done, it will display a dialog box prompting you to label the ERD and place it in safekeeping. You may then create another ERD, update the repair information on the hard disk, perform other backup tasks, or exit.



NOTE Remember to remove the floppy diskette from the drive once Backup finishes writing the repair information to it. Attempting to boot this diskette won't cause a problem; however, it will have to be removed before the system can be rebooted..

Saving to the repair Directory

Backup can save the entire registry to the repair directory. To update the %systemroot%\repair directory, follow these steps:



Desktop Library
Click to access!

1. Start Backup without any options.
2. Select the Emergency Repair Disk button in the Welcome tab.
3. At the prompt to insert a diskette, insert a diskette containing nothing of value. The diskette must be already formatted.
4. Select the “Also backup the registry...” check box (see Figure 4.22).
5. Once Backup is done, it will display a dialog box prompting you to label the ERD and place it in safekeeping. You may then create another ERD, perform other backup tasks, or exit.



FIGURE 4.22 Backup allows backing up the entire registry when the repair directory option is selected.

The Windows 2000 Resource Kit

The Windows 2000 Resource Kit contains a number of very useful tools. Many of these tools run from a command prompt, although one has a Windows-type interface. The Resource Kit has changed substantially in Windows 2000. Gone are all the old registry utilities, leaving only the multipurpose reg.exe program.



NOTE There are two resource kits: one is included with the operating system, on the distribution CD, and has only limited contents. The second version has both a book and a CD with many more utilities and is available from Microsoft Press. Try the URL <http://mspress.microsoft.com/reslink> for more information..

If nothing else, the Windows 2000 Resource Kit is an excellent source of both information and a whole bunch of really neat utilities and tools for the Windows 2000 user.



NOTE While I've got you in support mode, make a link on your Desktop for the URL <http://support.microsoft.com/support/search/c.asp?SPR=>. This URL links to the online TechNet search support. TechNet contains a vast amount of technical information oriented toward system administrators. I don't know what I'd do without TechNet..



WARNING Many of the earlier versions of the Windows Resource Kit utilities work with both Windows NT 4 and Windows 2000. However, be most cautious when using the older utilities with Windows 2000, as they have not been well tested on the Windows 2000 platform!.

Reg.exe

Reg.exe is a tool combining the functionality of a number of the other command-line driven Windows NT 4 Resource Kit registry tools. It improves the interaction between the command line and the registry, and is somewhat easier (and a whole lot more consistent) to use than the handful of other utilities. Reg.exe has the following functions:

- Add
- Backup
- Copy
- Delete
- Load
- Query
- Restore
- Save
- Unload
- Update

In the following sections, I'll cover each of the functions, showing parameters and results as examples of how to use Reg.exe.

Add

The add function, invoked with the command `reg add <options>`, is used to add an object (key or value entry) to the registry. Options include the registry object to be added with the object's value, an optional machine name (additions may be made to remote registries), and an optional data type, as described next.

The command line for add is:

```
REG ADD RegistryPath=value [data type][\Machine]
```

As with other registry tools, the registry path may be a ROOTKEY or a hive (with or without a value entry).

The ROOTKEY may be one of the following (HKLM is assumed if none is entered):

- HKLM (for HKEY_LOCAL_MACHINE)
- HKCU (for HKEY_CURRENT_USER)
- HKCR (for HKEY_CLASSES_ROOT)
- HKU (for HKEY_USERS)
- HKCC (for HKEY_CURRENT_CONFIG)

The hive will be further qualified to determine the object to be added.

The data type parameter will be one of the following (the default, if the data type is not specified, is to use REG_SZ):

- REG_SZ
- REG_DWORD
- REG_EXPAND_SZ
- REG_MULTI_SZ

Here's an example of executing the add command:

```
Windows 2000 8:56:09 C:\
REG ADD HKLM\Software\MyCo\MyApp\Version=1.00
The operation completed successfully.
```

```
Windows 2000 9:00:48 C:\
REG query HKLM\Software\MyCo\MyApp\Version
REG_SZ      Version 1.00
```

```
Windows 2000 9:00:59 C:\
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Backup

The backup function, invoked with the command `reg backup <options>`, is used to save the registry object specified to the file specified. Options include the registry path to be saved, the output filename, and an optional machine name (saves may be made on remote registries).

The command line for backup is:

```
REG BACKUP RegistryPath OutputFileName [\\Machine]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

Only HKLM and HKU may be specified when copying objects to a remote registry.



NOTE Notice that `reg save` and `reg backup` are identical in functionality..

An example of executing the save command is shown below. In this example, I've saved a small key to the file `C:\temp\MyCo.reg`:

```
Windows 2000 9:34:19 C:\
REG backup HKLM\Software\MyCo\MyNewApp c:\temp\MyCo
The operation completed successfully.
```

```
Windows 2000 9:34:21 C:\
dir c:\temp\myco.*
Volume in drive C is (c) - Boot drive
Volume Serial Number is CC56-5631
```



Desktop Library
Click to access!

```
Directory of c:\temp
```

```
07/17/99 09:34a          8,192          MyCo
      1 File(s)          8,192 bytes
                        183,407,104 bytes free
```

```
Windows 2000 9:34:27 C:\
```

Copy

The copy function, invoked with the command `reg copy <options>`, is used to copy the registry object specified to a new name. Options include the registry path to be copied (the source) and a destination name.

The command line for copy is:

```
REG COPY OldPath [\\Machine] Newpath [\\Machine]
```

As with other registry tools, the registry path to be copied (both the old path and the new path) may be a ROOTKEY or a hive. The path may be specified with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

Only HKLM and HKU may be specified when copying objects to a remote registry.



NOTE Consider the case where a registry object is copied from one registry to another registry on a different machine. This command is more powerful than is apparent at first glance..

The hive may be further qualified to determine the contents of a specific key or value entry. If no value entry is specified, all the value entries in the key will be copied. Here's an example of executing the copy command:

```
Windows 2000 9:10:52 C:\
REG query HKLM\Software\MyCo\MyApp\
```

```
Listing of [Software\MyCo\MyApp\]
```

```
REG_SZ      Version 1.00
Windows 2000 9:15:18 C:\
REG copy HKLM\Software\MyCo\MyApp\ HKLM\Software\MyCo\MyNewApp
The operation completed successfully.
```

```
Windows 2000 9:15:43 C:\
REG query HKLM\Software\MyCo\MyNewApp
```

```
Listing of [Software\MyCo\MyNewApp]
```

```
REG_SZ      Version 1.00
```

```
Windows 2000 9:15:51 C:\
```

Delete

The delete function, invoked with the command `reg delete <options>`, is used to delete the specified registry object. Options include the registry path to be deleted, an optional machine name (queries may be made on remote registries), and an optional parameter, `/F`, that forces the deletion without recourse.

The command line for delete is:

```
REG DELETE RegistryPath [\\Machine] [/F]\
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive (with or without a value entry). The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

Only HKLM and HKU may be specified when deleting objects from a remote registry.

The hive deletion may be forced by using the /F option, which will force the deletion without any prompt or confirmation. Microsoft recommends that the /F option be used only with extreme care. I agree.

An example of executing the delete command is shown next. Notice that I had to respond with a y to the prompt to delete the specified object.

```
Windows 2000 9:05:30 C:\
REG query HKLM\Software\MyCo\MyApp\Version
REG_SZ      Version 2.00
```

```
Windows 2000 9:09:30 C:\
REG delete HKLM\Software\MyCo\MyApp\Version
Permanently delete registry value Version (Y/N)? y
The operation completed successfully.
```

```
Windows 2000 9:09:40 C:\
REG query HKLM\Software\MyCo\MyApp\Version
The system was unable to find the specified registry key.
```

```
Windows 2000 9:09:43 C:\
```

Load

The load function, invoked with the command `reg load <options>`, is used to load the registry object from the file specified. The object must have been saved using the `reg save` or `reg backup` command. Options include the name of the file to load from, the registry path to be restored, and an optional machine name (restorations may be made to remote registries).

The command line for restore is:

```
REG LOAD FileName keyname [\\Machine]\
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a data key. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU

Only HKLM and HKCU may be specified in this command.

Objects in the key will be loaded, overwriting existing objects if there are any. Here's an example of executing the load command:

```
Windows 2000 9:47:58 C:\
REG load c:\temp\myco HKLM\TEMP\
The operation completed successfully.
```

```
Windows 2000 9:48:01 C:\
reg query HKLM\TEMP /s
```

```
Listing of [TEMP\]
```

```
REG_SZ      Version 1.00
```

```
Windows 2000 9:48:35 C:\
```

Query

The query function, invoked with the command `reg query <options>`, is used to search the registry for a specific value entry and display its contents. Options include the registry path to be queried, an optional machine name (queries may be made on remote registries), and an optional parameter, `/S`, that forces a query of all keys.

The command line for query is:

```
REG QUERY RegistryPath [\\Machine] [/S]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

The hive may be further qualified to determine the contents of a specific key or value entry. If no value entry is specified, all value entries in the key will be retrieved. Here's an example of executing the query command:

```
Windows 2000 8:54:08 C:\
REG QUERY HKLM\Software\Microsoft\ResKit\Setup\InstallDir
REG_SZ      InstallDir      C:\NTRESKIT
```

```
Windows 2000 8:54:11 C:\
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Restore

The restore function, invoked with the command `reg restore <options>`, is used to restore the registry object from the file specified. The object must have been saved using the `reg save` or `reg backup` command. Options include the name of the file to restore from, the registry path to be restored, and an optional machine name (restorations may be made to remote registries).

The command line for restore is:

```
REG QUERY FileName RegistryPath [\\Machine]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

Only HKLM and HKU may be specified when copying objects to a remote registry.

Objects in the key will be restored and overwritten by the information contained in the specified file. Here's an example of executing the restore command:

```
Windows 2000 9:39:17 C:\
REG backup HKLM\Software\MyCo\MyNewApp c:\temp\MyCo
The operation completed successfully.
```

```
Windows 2000 9:40:20 C:\
REG restore c:\temp\myco HKLM\Software\MyCo\MyNewApp
Are you sure you want to replace Software\MyCo\MyNewApp (Y/N) y
The operation completed successfully.
```

```
Windows 2000 9:40:44 C:\
```



Save

The save function, invoked with the command `reg save <options>`, is used to save the registry object specified to the file specified. Options include the registry path to be saved, the output filename, and an optional machine name (saves may be made on remote registries).

The command line for save is:

```
REG SAVE RegistryPath OutputFileName [\\Machine]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive (with or without a value entry). The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

Only HKLM and HKU may be specified when copying objects to a remote registry.

An example of executing the save command is shown next. In this example, I've saved a small key to the file `C:\temp\MyCo.reg`:

```
Windows 2000 9:16:27 C:\
REG save HKLM\Software\MyCo\MyNewApp c:\temp\MyCo.reg
The operation completed successfully.
```

```
Windows 2000 9:18:35 C:\
dir c:\temp\myco.reg
Volume in drive C is (c) - Boot drive
Volume Serial Number is CC56-5631
```

```
Directory of c:\temp
```

```
07/17/99 09:18a           8,192           MyCo.reg
          1 File(s)       8,192 bytes
          183,407,104 bytes free
```

```
Windows 2000 9:19:08 C:\
```

Unload

The unload function, invoked with the command `reg unload <options>`, is used to unload (delete) the registry object specified. The object must be a single-level key, such as `HKLM\TEMP`. Options include the name of the key to unload and an optional machine name (objects may be unloaded from remote registries).

The command line for unload is:

```
REG UNLOAD keyname [\\Machine]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU

Only HKLM and HKCU may be specified in this command.

Objects in the key will be unloaded and will not be saved. There is no recovery in the event of a user error with this command. Here's an example of executing the unload command:

```
Windows 2000 9:47:58 C:\
REG unload HKLM\TEMP\
The operation completed successfully.
```

```
Windows 2000 9:48:01 C:\
reg query HKLM\TEMP /s
```

The system was unable to find the specified registry key.

```
Windows 2000 9:48:35 C:\
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Update

The update function, invoked with the command `reg update <options>`, is used to update an existing object (key or value entry) to the registry. Options include the registry object to be added (with the object's value) and an optional machine name (updates may be made to remote registries).

The command line for update:

```
REG UPDATE RegistryPath=value [\\Machine]
```

As with other registry tools, the registry path to be queried may be a ROOTKEY or a hive, with or without a value entry. The ROOTKEY may consist of one of the following (HKLM is assumed if none is entered):

- HKLM
- HKCU
- HKCR
- HKU
- HKCC

The hive will be further qualified to determine the object to be added.

Below is an example of executing the update command. First I show the original value, then I update the object, then I show the new value.

```
Windows 2000 9:00:48 C:\
REG query HKLM\Software\MyCo\MyApp\Version
REG_SZ      Version 1.00
```

```
Windows 2000 9:01:33 C:\
REG update HKLM\Software\MyCo\MyApp\Version=2.00
The operation completed successfully.
```

```
Windows 2000 9:03:47 C:\
REG query HKLM\Software\MyCo\MyApp\Version
REG_SZ      Version 2.00
```



Windows 2000 9:03:53 C:\

Hints and Kinks from the Experts

In this chapter's hints and kinks, we provide a few hints for using the two registry editors.

How Do You Restrict Access to the Registry Editor?

In RegEdt32, follow these steps:

1. Highlight HKEY_USERS and choose Load Hive from the Registry menu.
2. Browse to their profile directory and select NTUSER.DAT.
3. When prompted for Key Name, input their user ID.
4. Navigate to \Software\Microsoft\Windows\CurrentVersion\Policies.
5. If no System subkey exists in Policies, add it. Then select Add Value in the menu, and add a value named DisableRegistryTools under the System key using type REG_DWORD and set the value to 1.
6. Choose Unload Hive from the Registry menu.

(Courtesy of John Savill.)

Should You Use RegEdit or RegEdt32?

You can use either. RegEdit does have a few limitations; the largest limitation is that it does not support the full RegEdt32 data types, such as REG_MULTI_SZ. If you edit this type of data with RegEdit, RegEdit will change its type.

RegEdit is based on the Windows 95 version and has features that RegEdt32 lacks, such as a good search feature. In general, RegEdit is nicer to work with. RegEdit also shows your current position in the registry at the bottom of the window.

(Courtesy of John Savill.)

How Do You Restrict Access to a Remote Registry?

Access to a remote registry is controlled by the ACL on the key winreg. To set access, follow these steps:

1. Start the registry editor (regedt32.exe).
2. Move to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers.
3. Check for a key called winreg. If it does not exist, create it (select Edit \emptyset Add Key).
4. Select the winreg key by clicking it.
5. Select Security \emptyset Permissions.
6. Click the Add button and give the user read access.
7. Once added, click the user and select Special Access.
8. Double-click the user and you can select which actions the user can perform.
9. Click OK when finished.

It is possible to set up certain keys to be accessible even if the user does not have access by editing the value HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg\All (using RegEdt32). You can add paths to this list.

(Courtesy of John Savill.)



NOTE For more information, see Knowledge Base article Q153183 at <http://www.microsoft.com/kb/articles/q153/1/83.htm>

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 5

Policies, Good for One, Good for All

JUMP TO TOPIC



Desktop Library

Click to access!

FEATURING:

<i>An introduction to policies</i>	122
<i>Just more confusion?</i>	124
<i>The Microsoft system policies for Windows 2000</i>	129
<i>The Microsoft System Policy Editor for Windows NT 4</i>	135
<i>Hints and kinks from the experts</i>	142

There are several different ways to modify the registry. Many applications do it, and also there are the registry tools. Windows 2000 stores configurations for all users, and computers, as policies. By default, no policies are set, but an administrator can easily set policies for a group of users or for the entire system. It's possible to set some policies by manually "hacking" the registry. However, that's the hard way. An easier way to change policies is to use one of the policy tools that Microsoft has provided for us.

The first question on your mind is, "What does this have to do with the registry?" Well, of all things in Windows 2000, policy affects (and changes) the registry more than anything else. With policy settings, it is possible to change the way hardware and software behave and can be used.

An Introduction to Policies

Policies govern a site, a domain, or an organizational unit (often referred to as an OU), but not a specific user or computer. Policy is applied in a hierarchy, with higher-level policies used where no lower-level policy exists. For example, policy is applied as site (the highest level), then domain, organizational unit, and user. Policies in the domain override those they conflict with in the site, while conflicts between the domain and organizational unit will be resolved with the organizational unit taking precedent. Conflicts between nested organizational units are resolved with the lower-level organizational unit taking precedent.



NOTE Policy objects and settings can be set, unset, or not configured.

Policy settings are configured in objects called group policy objects (GPOs). GPOs are edited using the Microsoft Management Console (MMC).

The Official Order of Policy Implementation Is...

When Windows 2000 implements system policies, they are applied in this order:

1. Policies inherited from Windows NT 4. The Windows NT 4 policies are contained in the NTConfig.pol file. Note that Windows NT 4 policies need not exist, and will not exist on a clean installation of Windows 2000.
2. The policies contained in the local group policy object.
3. Site group policy objects, in the order specified by the administrator.
4. Domain group policy objects, in the order specified by the administrator.
5. Organizational unit group policy objects, from higher-level to lower-level organizational unit (parent to child organizational unit), and in administratively specified order at the level of each organizational unit.

Organizational units may be nested. That is, you can have an organizational unit called Students. Within Students, you then might have Freshmen, Sophomores, Juniors, and Seniors, representing the four classes. (You might also have Graduate Students, Masters, or Doctoral.) Nesting can be as simple or as complex as your organization is.

When nesting organizational units, policy may be either inherited or not. You, the administrator, will specify inheritance rules, within the following framework:

- Inheritance is downward only. In the example above, Freshmen inherit from Students, but Students would *never* inherit from Freshmen.
- Settings that have not been configured are not inherited.
- Settings that are disabled will be inherited as disabled.
- When a setting is configured in the higher-level organizational unit, and not configured in the lower-level organizational unit, then the lower-level organizational unit will inherit the setting from the parent organizational unit.
- When settings between the higher-level organizational unit and a lower-level organizational unit don't conflict (are compatible), but are not the same, both will be used to form the lower-level organizational unit's policy.
- When settings between the higher-level organizational unit and a lower-level organizational unit do conflict (are incompatible), but are not the same, the lower level organizational unit's policy will be used.



NOTE Well, almost always... There is an attribute called No Override that, if selected at the higher-level organizational unit, will cause the lower-level organizational unit to always inherit the higher level OU's policy.

Just More Confusion?

In the previous section, we saw that policies are set for sites, domains, and organizational units. Now, I'm going to confuse things a bit and say that policies are divided into two parts, Computer Configuration and User Configuration. Computer Configuration specifies policies that are applied to a computer without regard to who the user is. User Configuration specifies policies that are applied to a user without regard to which computer the user logs onto.

Both the Computer Configuration and the User Configuration are made up of three sections:

Software Settings: Everything in Software Settings deals with software installation policy—for example, what can be installed, what must be uninstalled, and when.

Windows Settings: Settings for Windows 2000 are controlled in this section. There are more items in

the Windows Settings section for the User Configuration than for the Computer Configuration.

Administrative Templates: The extensible section, almost a catchall for everything that doesn't fall into the other two sections, is the Administrative Templates section. Items are added to Administrative Templates using .adm files.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Software Settings

The Software Settings section contains policies that deal with software installation and maintenance, such as what applications can be installed, what must be uninstalled and when, what maintenance must be done, and so on.

For example, if you configure Microsoft Word 2000 under Software Settings, and a user logs on to a computer that doesn't have Microsoft Word 2000 installed already, the user will still see a Start menu selection (shortcut) for Microsoft Word 2000. If the user selects this shortcut, Microsoft Word 2000 will install itself (from a network share) for the user to use.

Windows Settings

In both Computer Configuration and User Configuration, you'll find a Windows Settings section. For Computer Configuration, the settings are applied to each user who logs on to the computer. For User Configuration, the settings are applied to users who log on regardless of the computer they log onto.

Administrative Templates

The Administrative Templates section contains all registry-based information. There are two hives that are used:

- HKEY_CURRENT_USER—the location where user configuration settings are saved
- HKEY_LOCAL_MACHINE—the location where computer configuration information is saved

Both user application policy items and policy for Windows 2000 are managed in Administrative Templates. Adding policy items to Administrative Templates is simple. Most applications (that support policy) come with .adm files that contain information about which registry settings can be configured. For example, Microsoft Word 2000 comes with an .adm file named word9.adm (it seems that Microsoft Word 2000 is also known as Microsoft Word version 9).

A small fraction of the Microsoft Word 2000 .adm file is:

CLASS USER

CATEGORY !!Word

KEYNAME Software\Policies\Microsoft\Office\9.0\Word\Options



JUMP TO TOPIC


```

CATEGORY !!Tools
KEYNAME Software\Policies\Microsoft\Office\9.0\Word\Options\vpref
CATEGORY !!View
CATEGORY !!WShow
POLICY !!WHighlight
    PART !!StdCheckT CHECKBOX
    VALUENAME fShowHighlight_533_1
    VALUEON NUMERIC 1
    VALUEOFF NUMERIC 0
    END PART
    END POLICY
POLICY !!WBookmarks
    PART !!StdCheckT CHECKBOX
    VALUENAME grpvisi_146_1
    VALUEON NUMERIC 1
    VALUEOFF NUMERIC 0
    END PART
    END POLICY
(deleted lines)
[Strings]
Word = "Microsoft Word 2000"
Tools = "Tools | Options..."
Misc = "Miscellaneous"

```

```
StdCheckT = "Check to enforce setting on; uncheck to enforce setting off"
```

```

View = "View"
WHorzScroll = "Horizontal scroll bar"
WVertScroll = "Vertical scroll bar"
WHighlight = "Highlight"
WBookmarks = "Bookmarks"
WStatusBar = "Status bar"
WShow = "Show"
WPictPlace = "Picture placeholders"

```

In this example, we see that the registry section being changed is HKEY_CURRENT_USER. The key being set (or changed) is Software\Policies\Microsoft\Office\9.0\Word\Options\vpref (vpref stands for "view preferences"). The key to be changed is shown in a KEYNAME line in the .adm file.

Checking Microsoft Word 2000's menu structure, we see that there is a top-level menu item called Tools, and under Tools is a menu selection called Options. Clicking Options will display Microsoft Word 2000's Options dialog box (Figure 5.1), which has a tab named View. On the View tab is a section called Show. Each of these items correlates to a CATEGORY line in the .adm file, shown partially in the above listing.

Finally, in the .adm file, we have the POLICY line. The first POLICY line shown is coded with a substitutable variable, called WHighlight. Looking at the bottom of the file, we see that WHighlight has a value of "Highlight" and this value is used as the policy prompt.

Now, let's link this information together into one figure. Figure 5.2 shows the MMC Group Policy window with Microsoft Word 2000 opened. Inside, we see that the next selection is Tools | Options, and under that selection is View, and finally under View is Show. When Show is selected, each of the settable policy items is shown in the right pane of the MMC Group Policy window.



FIGURE 5.1 User options in Microsoft Word 2000 are set in the Options dialog box.

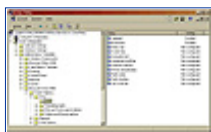


FIGURE 5.2 Microsoft Word 2000's group policy for a user.

Back to the above listing, under POLICY, there's a PART line with the substitute text variable StdCheckT (which has the value "Check to enforce setting on; uncheck to enforce setting off"), followed by the keyword CHECKBOX. CHECKBOX specifies that this policy is toggled on and off (that is, checked and unchecked). If on, the value of the key will be 1 (specified with the line VALUEON NUMERIC 1) or 0 (specified by the next line, VALUEOFF NUMERIC 0).

The next two lines end blocks that start at PART (ends at the END PART line) and POLICY (ends at the END POLICY line). Each policy has one or more parts; most policies have only a single part, however.

Armed with this information, you can go out and create policy files for other applications. Granted, if you are not the applications creator, it will be difficult—but not impossible—to set things like defaults using policies.

The changes made under Administrative Templates are saved in two Registry.pol files. These files are stored in subdirectories under %systemroot%, one called Machine (which contains the Registry.pol file that will be used to update HKEY_LOCAL_MACHINE), and User (which contains the Registry.pol file that will be used to update HKEY_CURRENT_USER).

Finding Registry.pol Files

I can't say where the Registry.pol files are, except to tell you to use the Windows Find feature, or a command session's dir command. Why? Well, each installation is unique in the locations where these files are stored; for example, on my Windows 2000 computer, I have the following Registry.pol files:

```
G:\WINNT\system32\GroupPolicy\Machine\Registry.pol
G:\WINNT\system32\GroupPolicy\User\Registry.pol
G:\WINNT\sysvol\domain\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Registry.pol
G:\WINNT\sysvol\domain\Policies\{6AC1786C-016F-11D2-945F-00C04FB984F9}\USER\Registry.pol
G:\WINNT\sysvol\domain\Policies\{EE520C60-1F3E-11D3-A6E8-00A024D2DD82}\User\Registry.pol
G:\WINNT\sysvol\sysvol\darkstar.mv.com\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Registry.pol
G:\WINNT\sysvol\sysvol\darkstar.mv.com\Policies\{6AC1786C-016F-11D2-945F-00C04FB984F9}\USER\Registry.pol
G:\WINNT\sysvol\sysvol\darkstar.mv.com\Policies\{EE520C60-1F3E-11D3-A6E8-00A024D2DD82}\User\Registry.pol
```

Notice the use of GUIDs (globally unique identifiers) for some of the directory names—those may be different for each installation.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

The Microsoft System Policies for Windows 2000

The System Policy Editor is a tool that allows users to set policy in Windows NT 4. In Windows 2000, we have new tools for policies. Now, let's be clear—the stand-alone version of System Policy Editor is still included, and may be used, but it is definitely not recommended. Now we use the Active Directory Users and Computers to manage policy. Changes made to policy are made to the registry, either immediately or when a given user or member of an organizational unit logs on, or when the computer starts. Policy is a registry issue, and a complex one, at that.



WARNING Wait a minute—I know I changed that registry entry! With policies it is possible to "hack," or change, the registry and have the change go nowhere fast. That's right: the policy will be reapplied automatically, wiping out whatever changes you have made to the registry, all without even telling you it is happening! If you ever find your changes mysteriously disappearing, round up the usual suspects and make sure that policy is high on the suspect list!

Policy may be edited in a number of different ways:

System Policy Editor: This utility, which is becoming obsolete, is retained for compatibility with Windows NT 4. Neither Microsoft nor I recommend that you use the System Policy Editor.

Microsoft Management Console (MMC): This is a program that is used to manage many facets of Windows 2000. MMC is able to load whatever functionality is needed through the use of a custom extension called a snap-in. With Windows 2000, Microsoft provides about 40 different snap-ins to use with MMC.

Active Directory Users and Computers: This administrative tool (see Start Ø Programs Ø Administrative Tools) allows management of computers, users, groups, domain controllers, and policy. Actually this is MMC, using a snap-in to do Active Directory. When you choose to edit policies, MMC will be used with the policy snap-in.

Active Directory Site and Services: This administrative tool (see Start Ø Programs Ø Administrative Tools) allows management of sites (an Active Directory organizational level) and services.



NOTE Each of these Windows 2000 administrative tools actually uses MMC as a common interface. And each in turn uses MMC to edit policy. It's not uncommon to have many copies of MMC open at the same time.

Each of these tools requires that you select certain objects to enable editing of group policies. The next section describes how to use each policy-editing tool.

System Policy Editor

The System Policy Editor is obsolete, retained only for compatibility with NT 4. Neither Microsoft nor I recommend that you use the System Policy Editor if you can possibly avoid using it. (Uh, no, I can't see any circumstances where you'd need to edit or manage policies using the System Policy Editor.)

To start the System Policy Editor, from the Start menu's Run command, enter **poledit** and click OK. See "The Microsoft System Policy Editor for Windows NT 4" later for information on how to use the Microsoft System Policy Editor.

Microsoft Management Console (MMC)

MMC is a "universal" management tool that Microsoft has created to manage Windows 2000. Using MMC is easy, and since MMC presents a standardized appearance and operating methods, it will become the preferred tool to use for management.

To start MMC, from the Start menu's Run command, enter **MMC** and click OK. Once started, MMC is able to load whatever functionality is needed.

Using MMC to manage group policy is easy! First, it is trivial to create an MMC console file that is configured to display a given GPO policy object. Just follow these steps:

1. Open the administrative tool Active Directory Users and Computers.
2. Right-click the organizational unit that uses the policy to be edited to display its context menu. If you're creating a new policy, select the organizational unit that will use the policy once it is created. If creating policy for a domain, select the domain. In Figure 5.3, darkstar.mv.com is the domain, and Domain Controllers, Students, Freshmen, Sophomores, Juniors, Seniors, Faculty and Staff, Faculty, Staff, Student Staff, and Temporary Staff are all organizational units.

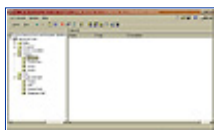


FIGURE 5.3 You configure user policies using Active Directory Users and Computers.

3. Select Properties from the context menu, or Properties from the Action menu.
4. In the properties dialog box for the object, select the Group Policy tab (see Figure 5.4).
5. Select a policy from the Group Policy Object Links list and click Edit, or click New to create a new group policy object. Clicking Edit displays the Group Policy window, which is actually MMC displaying the Group Policy add-in (see Figure 5.5).



FIGURE 5.4 The Group Policy tab allows group policy to be assigned. Add, Edit, New and other commands help with policy management.

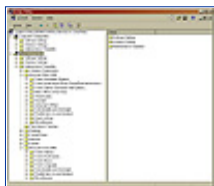


FIGURE 5.5 The Group Policy window in MMC allows you to manage group policy. Drilling down through the hierarchy gets you to the right place.

6. Select the policy item to change. Note in Figure 5.5 that I have opened both Microsoft Office 2000 and Microsoft Word 2000 policies. Note how the policies are tied to menu selections and functionality groups.

7. Now for the hard part. To create an MMC configuration file (these files have an extension of .msc) select **Console** \emptyset **Save As**. Enter a new filename when prompted (I used the name **Student Policies**). Click **Save** to save this file.

Well, now you have created your policy MMC configuration file; next is what to do with it. (Actually, you have probably saved a bunch of MMC configuration files; one for each group policy object.) My recommendation is to create a folder to hold these files, maybe called **Local Policy**. Then create a second folder under **Administrative Tools**, again called **Local Policy**. Then place shortcuts to each of the MMC configuration files in this folder. This will give you one-click access to each group policy object.

Figure 5.6 shows an example of just this type of policy control: You can see my Start menu on the main server, DORA, here at DarkStar. Notice that I have a policy for each organizational unit that I created to manage my users. I also have other policies (such as the default policies that come with Windows 2000), which I can add to the same folder. Easy, fast, and efficient—sure beats threading through Active Directory Users and Computers, properties, and everything else.

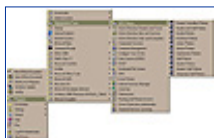


FIGURE 5.6 My Start menu's Administrative Tools folder contains a folder called GPO Policies, where I can start MMC for each policy that I have.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)
Active Directory Users and Computers

The Active Directory Users and Computers administrative MMC tool allows you to manage computers, users, groups, domain controllers, and policy. If you select a computer domain (back in Figure 5.3, darkstar.mv.com is one domain, while DORA is the domain controller) or an organizational unit, then select Properties in either the context menu or the Action menu, you will be able to view the Group Policies properties tab.

Setting Policy for a User

To set policy for a given user, you need the user, an organizational unit to assign the user to, and a policy to apply to the organizational unit. For example, I have students, and some students are seniors (and seniors are much more responsible than freshmen!). I have a student, Marie Theplama, who is a senior. This creates two levels of organization. Your organization may have more (or fewer) levels of organization, but the process is similar.

If I've not yet set up any policies, then to apply the global policy object to Marie Theplama, I must do the following:

1. I first create an organizational unit called Students.
2. I display the properties for the Students organizational unit properties, and click the Group Policy tab.
3. I then click the New button to create a new policy. The policy is created, and I am placed into rename mode to name the new global policy object.
4. I select the new global policy object and click the Edit button to change whatever policies are applicable.
5. Next, I repeat steps 1–3 to create another organizational unit under Students named Seniors.
6. I then select the new global policy object, and click the Edit button to change whatever policies are applicable so that seniors have appropriate privileges and policies.
7. I then create Freshmen, Sophomores, and Juniors organizational units and global policy objects in the same manner.
8. I create a user for Marie Theplama. She's a senior; nothing else is special about her. I create her user under the organizational unit Seniors.

Setting Policy for a Computer

To set policy for a given computer, the process is very similar to the process for users, above. You need to create a computer record, an organizational unit to assign the computer to, and a policy to apply to the organizational unit. Here at DarkStar, computers are named after famous science fiction characters. We have a computer called Pixel (who is a cat that can walk through walls).

Just like for users, we need to have an organizational unit for computer policy. For this example, we'll use Students for our computers that will be accessible to and used by students.



NOTE A computer's organizational unit is no different than an organizational unit for a user; in fact the same organizational unit could be used for both if appropriate.

I create a computer under Students for Lazarus. Once created, as a computer, I can set a description, assign membership into security groups and organizational units, and specify the computer's location and who is responsible for this computer. I assign membership to the Students organizational unit, any other applicable organizational units, and security groups.

Just like with users, you have a lot of latitude when configuring computers. While users are typically assigned to organizational units based on the administrative hierarchy of the organization, computers are often assigned based on physical location, or how they are to be used.

The Microsoft System Policy Editor for Windows NT 4

The System Policy Editor for Windows NT 4 is a tool that allows users to set policy. Many of the changes made by the System Policy Editor are to the registry, so although the System Policy Editor is not thought of as a registry tool, I'll document it here anyway. Actually, modifying the Windows NT 4 registry using the System Policy Editor is a wise move—it will validate your changes, preventing you from doing something that may have seemed logical to you, but actually is not.

The System Policy Editor allows you to open either a policy file (with the extension of .pol) or a computer. It uses a simple user interface, as shown in Figure 5.7. When you click an object, the object's Properties dialog box will be displayed. In Figure 5.7, Local Computer and Local User are both objects that can be opened.



FIGURE 5.7 The System Policy Editor displays a properties dialog box when you click an icon. Then you can open the tree to see specific settings.

With the System Policy Editor, the Local Computer entry should display eight items, all applicable for a Windows NT system. For a Local User, the Properties dialog box should have six items. In both cases, the items displayed are unique; there is no overlap.

WYSIWYG? From System Policy Editor?

What is displayed in the Properties dialog boxes is dependent on which template(s) are loaded. With Windows NT, there are three templates supplied by default:

- common.adm:** Contains user interface options common to both Windows NT and Windows 95/98.
- winnt.adm:** Contains specific settings for Windows NT and Windows 2000.
- windows.adm:** Contains specific settings for Windows 95/98.

There are two sections in all .adm files, CLASS MACHINE and CLASS USER, which define how settings are applied.

.adm files are text files, that can be modified to suit the user's needs. A competent user should be able to write an .adm file, or modify an existing one without too much trouble. However, those pesky "make sure you have good backup" warnings also apply if you customize your .adm files.

The System Policy Editor is usable for Windows 95/98 clients, enabling some remote administration of these machines. However, the System Policy Editor has not been well tested on these two platforms.

Typically, for all machines (Windows NT and Windows 95/98), you can modify the following categories:

Control Panel: Allows you to restrict the display of the Control Panel.

Desktop: Allows/disallows you to change wallpaper and/or color schemes.

Shell: Allows you to

- remove the Run command from the Start menu.
- remove folders from Settings on the Start menu.
- remove the Taskbar from Settings on the Start menu.
- remove the Find command from the Start menu.
- hide drives in My Computer.
- hide Network Neighborhood.
- hide the Entire Network in Network Neighborhood.
- hide all items on the Desktop.
- disable the Shut Down command.
- not save settings at exit.

System: Allows you to

- disable registry editing tools.
- run only allows Windows applications.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

This site is brought to you by
EarthWeb



HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH



Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

For Windows NT, you can modify the following categories:

Windows NT Shell: Consists of three sections, which allow setting the following:

- | | |
|-------------------------------|--|
| Custom User Interface: | Set custom shell |
| Custom Folders: | Set custom Programs folder |
| | Hide Start menu subfolders |
| | Set custom Startup folder |
| | Set custom Network Neighborhood |
| | Set custom Start menu |
| Restrictions: | Use only approved shell extensions |
| | Remove File menu from Explorer |
| | Remove common program groups from Start menu |
| | Disable context menus for the Taskbar |
| | Disable Explorer's default context menu |
| | Remove the Map Network Drive and |
| | Disconnect Network Drive options |
| | Disable Link File Tracking |

Windows NT System: Consists of four choices:

- Parse Autoexec.bat
- Run logon scripts synchronously
- Disable Task Manager
- Show welcome tips at logon

For Windows 95, you can modify the following categories:

Windows 95 Control Panel: Consists of four sections, which allow setting the following:

Network: Restrict Network Control Panel

Printers: Restrict printer settings

Passwords: Restrict Passwords Control Panel

System: Restrict System Control Panel

Windows 95 Shell: Consists of one section, Custom Folders, which allows you to set the following:

- Custom Programs folder

- Custom Desktop icons
- Hide Start menu subfolders
- Custom Startup folder
- Custom Network Neighborhood
- Custom Start menu

Windows 95 System: Consists of one section, Restrictions, which allows you to

- disable the MS-DOS prompt.
- disable the single-mode MS-DOS apps.

Windows 95 Network: Consists of one section, Sharing, which allows you to

- disable file sharing.
- disable print sharing.

For any type of machine (Windows NT or Windows 95/98), you can modify the following:

Network: Consists of one choice:

System policies update: Remote update.

System: Consists of two sections, SNMP and Run, which allow you to set the following:

SNMP: Communities
Permitted managers
Traps for Public community

Run: Items that are executed at startup

For Windows NT-only machines, you can modify:

Windows NT Network: Consists of one section, Sharing, which allows you to

- create hidden drive shares (workstation).
- create hidden drive shares (server).

Windows NT Printers: Consists of three choices:

- Disable browse thread on this computer
- Scheduler priority
- Beep for error enabled

Windows NT Remote Access: Consists of four choices:

- Max number of unsuccessful authentication retries
- Max time limit for authentication
- Wait interval for callback
- Auto Disconnect

Windows NT Shell: Consists of one section, Custom Shared Folders, which contains four choices:

- Custom shared Programs folder
- Custom shared Desktop icons
- Custom shared Start menu
- Custom shared Startup folder

Windows NT System: Consists of two sections:

Logon: Allow logon banner
Enable shutdown from
Authentication dialog box
Do not display last logged on
username
Run logon scripts synchronously

File System: Do not create 8.3 filename for long
filenames
Allow extended characters in 8.3
filenames

Do not update last access time

Windows NT User Profiles: Consists of four sections:

- Delete cached copies of roaming profiles
- Automatically detect slow network connections
- Slow network connection time-out
- Timeout for dialog boxes

For Windows 95 machines, you can modify the following:

Access Control: Consists of one section:

- User-level access control

Logon: Consists of three sections:

- Custom logon banner
- Require validation by network for Windows access
- Allow logon without name or password

Passwords: Consists of four sections:

- Hide share passwords with asterisks
- Disable password caching
- Require alphanumeric Windows password
- Min Windows password length

Microsoft Client Service for NetWare Networks: Consists of four sections:

- Preferred server
- Support long filenames
- Search mode
- Disable automatic NetWare login

Microsoft Client for Windows Networks: Consists of three sections:

- Log on to Windows NT
- Workgroup
- Alternate workgroup

File and Printer Sharing: Consists of two sections:

- Disable file sharing
- Disable printer sharing

Dial-up Networking: Consists of one section:

- Disable dial-in

Windows 95 System: Consists of three sections:

SNMP: Enables Internet MIB (RFC1156).

Network Paths: Enables network path for Windows setup.

Enables network path for Windows tour.

Profiles: Enables user profiles.

For each item, choices may range from a simple disable or enable of the property to setting of text, additional options, and so on. For instance, Figure 5.8 shows the Local Computer Properties dialog box, where I've selected Local computer, Network, System Policies Update, and Remote Update. Displayed in the Settings for Remote Update at the bottom of the dialog box are some additional settings that I can change for this item.

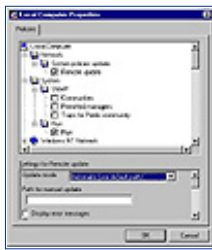


FIGURE 5.8 Setting remote update properties displays additional choices.

Hints and Kinks from the Experts

In this chapter's hints and kinks, we cover a few hints on using the Windows 2000 policy.

Eliminate Policy Editor Restrictions

The Windows policy editors (either Windows 2000's MMC-based Policy Editor or Windows NT 4's System Policy Editor) don't let you remove an entry you added in error, nor do they allow the setting of REG_BINARY values.

You can overcome these deficiencies by using the Registry Editor, RegEdt32:

1. Select the HKEY_LOCAL_MACHINE window.
2. On the Registry menu, select Load Hive.
3. Navigate to and open a .pol file, such as %systemroot%\System32\USER.POL.
4. Enter a name in the Key Name dialogue box, such as USERPOL.
5. Make any additions, deletions, and changes you desire.
6. Select the key name you used in step 4.
7. On the Registry menu, select Unload Hive.
8. Respond Yes to "Registry Editor will unload the currently selected key," and you're done.

(From <http://www.jsiinc.com/reghack.htm>. Copyright © 1997–1999 Jerold Schulman [JSI, Inc.]. Used with permission.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

PART II

Advanced Registry Stuff

How's this for good government?

If the U.S. government has no knowledge of aliens, then why does Title 14, Section 1211 of the Code of Federal Regulations, implemented on July 16, 1969, make it illegal for U.S. citizens to have any contact with extraterrestrials or their vehicles?

CHAPTER 6

Associations, Linkages, and OLE: How Confusing Can This Get?

FEATURING:

<i>Introduction to OLE</i>	148
<i>Client-server OLE applications</i>	157
<i>How linkages are made between applications</i>	168
<i>Hints and kinks from the experts</i>	172

First, let's start out with a few ground rules:

- There is no way to learn all about OLE in one chapter.
- Even OLE experts are not really experts.
- There are a number of good books on OLE; but unless you are programming, avoid them.
- If you don't understand everything about OLE after reading this chapter, don't feel bad.
- The author takes no responsibility for what happens when you wake up at 2 A.M. and shout, "Now I understand!"

Most programmers don't build their OLE applications from the ground up. Instead, for the difficult parts, they use development systems such as Microsoft's Developer Studio. Today, a programmer can create an OLE application almost as quickly as any other type of application.

Most applications manage their initial OLE setup by themselves. Some applications rely on their installation programs to do the OLE setup. And some applications both use the installation program to set up OLE and, if the configuration becomes damaged, repair the damage to the extent that they can reconfigure the OLE components.

This brings up some questions. First, what is OLE? Second, what does OLE have to do with the registry? Moreover, why do we have to worry about it? Do we mention DDE? And where, oh where, does the Clipboard fit into this mess?

OK, stay tuned for answers to all of these questions.

Introduction to OLE

OLE (Object Linking and Embedding) is a technology that allows applications to share data and functionality easily. I like that. Sounds good. It's quick and easy to understand, and it's basically accurate.

Kraig Brockschmidt is probably the best-known expert on OLE. Here's how he describes its evolution:

Windows API (Application Program Interface) evolved into Windows Objects, which eventually became what we know as OLE.

Kraig admits it's not that simple, but OLE developed by evolution, not by revolution.

Way back in the good old days, Windows was much simpler and easier to understand. In its first incarnation, Windows allowed virtually no interprocess communications. There was the Clipboard (which we still know and love) to which one program could post data, and another program could then (hopefully) read that data. However, that interaction required user interaction. The user was required to take steps to put the selected data in the Clipboard, then in the recipient application, take steps to retrieve the data stored in the Clipboard.

Problems arose. First, the basic Clipboard supported only a very limited range of data types. Programs could exchange data in various basic formats (text and binary, for the most part), but these formats were sorely lacking the flexibility to express any object that was composed of compound data.

Compound Data?

Compound data is data that contains information in multiple formats. The easiest type of compound data to envision is a word processing document that includes some images. At this point in the evolution of the Clipboard, the word-processing program couldn't just toss that document and its images on the Clipboard. After all, how would the program identify the format of that data? If it said binary, no other application would be able to understand or use the data. If it said text, what would happen when an application tried to use the data and encountered the images? Would it delete the images? Sure, that would work, but if the user wanted the complete document, including the images, he or she would be most unhappy about the results.

Microsoft realized quickly that applications needed a direct, application-to-application communication method that didn't rely on the Clipboard. Quickly or slowly, depending on your point of view, the concept of DDE (Dynamic Data Exchange) was born. Actually "conceived" would be a better description, because DDE wasn't viable in its original format. As it grew, DDE did allow applications to communicate data. However, there were still problems. With DDE, there was no way for applications to find out about their partners. Developers created most DDE applications specifically as pairs. For applications from two independent sources, DDE was unlikely to be useful, because the developers would have to cooperate in order to take advantage of DDE.

OLE became the next stage in the development of interapplication communications and data sharing. OLE allowed an application to interact with another one without knowing, in advance, about the other application. Magic, really.

The Clipboard

The Clipboard is the original and most basic method to transfer data between applications. The Clipboard supports both interapplication transfers (between two applications) and intra-application transfers (within the

same application).

There is only one object in the Clipboard at any one time. There are some complex rules on the Clipboard, such as the following:

- An application cannot assume that an object placed in the Clipboard will remain there after the application releases the Clipboard. Therefore, it is not possible to use the Clipboard as a temporary storage location.
- The format of the object in the Clipboard must be in one of the standard formats (listed below), or the application placing the data on the Clipboard must be prepared to render or display the Clipboard's contents.
- Some objects in the Clipboard are in a format that is not native to Windows. These objects require the application that places the object to be available to display or render the object if necessary.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Windows 2000 is able to support the following types of data in the Clipboard, without creating custom formats:

CF_BITMAP: a bitmap (image)**CF_DIBV**: a DIB (Device Independent Bitmap)**CF_DIBV5**: a version 5 bitmap (only available on Windows 2000)**CF_DIF**: a DIF (Data Interchange Format) object**CF_DSPBITMAP**: a private format bitmap**CF_DSPENHMETAFILE**: an enhanced metafile display format object**CF_DSPMETAFILEPICT**: a metafile-picture display format object**CF_DSPTEXT**: a text display format object, with private format**CF_ENHMETAFILE**: an enhanced metafile object**CF_GDIOBJFIRST** through **CF_GDIOBJLAST**: a range of integer values for application-defined GDI (Graphical Device Interface) objects**CF_HDROPV**: a handle of type HDROP, identifying a list of files**CF_LOCALE**: locale information**CF_METAFILEPICT**: a metafile picture object**CF_OEMTEXT**: a text format in the OEM (original equipment manufacturer) character set**CF_OWNERDISPLAY**: an object of owner display format**CF_PALETTE**: a color palette object**CF_PENDATA**: an object containing data for the pen extensions to the Microsoft Windows for Pen Computing**CF_PRIVATEFIRST** through **CF_PRIVATELAST**: a range of integer values for private Clipboard formats**CF_RIFF**: a sound object too complex for the CF_WAVE format**CF_SYLK**: an object in Microsoft Symbolic Link (SYLK) format**CF_TEXT**: a plain-text format object**CF_WAVE**: an audio object, using PCM (Pulse Code Modulation)**CF_TIFF**: a Tagged Image File Format object**CF_UNICODETEXT**: a text object using the two-byte Unicode character set

JUMP TO TOPIC



Desktop Library

Click to access!

As this list shows, Windows supports many different formats, without any programmer intervention. However, there are many situations in which these formats are not adequate. In these cases, the application serving (placing) the data on the Clipboard may register a new format with Windows. To enable viewing of the Clipboard data, it will be necessary to also have code that will display the Clipboard data.

DDE

DDE, or Dynamic Data Exchange, has been part of Windows since the early days. An Excel spreadsheet (the client) for managing stock market information is an example of DDE. A second software application that actually retrieves the stock prices (quotes) is the server. In addition, there is an application that goes to the Internet and gets current stock market quotes (the server). The two programs need to interact dynamically (after all, prices change), so using the Clipboard is not optimal: you want your spreadsheet updated dynamically and efficiently, without any user interaction.

Through a process of broadcasting, Excel (the client) establishes a communications link with the server. Excel broadcasts its request and the server responds that it is able to fulfill this request. A DDE linkage is established, allowing Excel to request information from the server as necessary.

As an example, you may be interested in a particular list of stocks. Excel would tell the server to check these stocks and provide the current quote for them. Excel might also have a timer loop that repeats this process every five minutes, providing you with up-to-date stock quote information.

As another example, you might request a one-time quote on a stock of interest. Maybe you're interested in just how well Microsoft (MSFT) is doing on the stock exchange. Perhaps your spreadsheet has a section where you type in the stock name. You enter the name and the quote comes back.

Either the client or the server can perform automatic updating. Client-initiated updates might occur on a time-based basis, or when the user makes a change if the data retrieved was relatively static. Servers might initiate an update whenever the server recognizes that the information the user is requesting has become out of date.

OK, no one said DDE was easy. If they did, they didn't tell the truth. DDE is complex and very difficult to understand or use. Programmers exposed to DDE shuddered and desperately searched for better alternatives. Some programmers kludged together broadcast messages to pass simple data, but for many, DDE was still the best (only) method to exchange data between two applications.

Why Is It Difficult to Exchange Data?

Memory protection causes most of the problems; one application cannot access memory objects belonging to other applications.

When an object is placed in the Clipboard, the memory that the object occupies is given to Windows. From that point onward, Windows owns the object, and the application that placed the object in the Clipboard loses control of the object. This means that whenever an object is placed in the Clipboard, the application will usually make a copy of the object and place the copy in the Clipboard, keeping the original object for the application's use.

The DDE process uses the Clipboard to transfer large blocks of data, too. Typically, the server application would place the data on the Clipboard and use DDE to tell the client application about the data. Server applications are able to pass small data objects to the client application as part of the DDE conversation.

What Is OLE?

OLE means Object Linking and Embedding. That says it all. With DDE, and with the Clipboard, applications only passed data and did not pass any functionality. With OLE, we expand on what the server application is able to do for the client.

Not only may applications pass data back and forth using OLE, but OLE also allows the server and the client to pass programming functionality between them. The server is able to do something that the client wishes done. However, the client program's developer does not have to develop all this functionality if it exists already.

As an example, take the email system called Outlook. Outlook has a simple, built-in email editor. However, some users want (demand, actually) more functionality in their email editors. They want formatting, macros,

even included images, and other nifty stuff. Microsoft Word actually would do everything that the user wants.

Wouldn't it be nice if the Outlook development team could borrow part of Word? Now, it would make little sense for the Outlook development team to sneak into the Word group's office and steal the code for Word. After all, they'd then have to maintain it, and Word's one big puppy—major maintenance blues there.

What's the next best thing? First, let's let the Word developers continue to maintain Word. Second, let's get Word to work for us. We know that the developers on the Word team included OLE server technology into Word; Word has client OLE technology too, in case you wondered. However, we find that the Outlook team can't really expect the Word team to put special stuff into Word for them, so what can they do?

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

Things are not so bad here: because Word is an OLE-compliant application, all Outlook has to do is to ask Word, "What can you do for me?" Outlook does this by first checking with the server at the most basic OLE level (a level that all OLE applications must support). This level allows the client to ask the server what functionalities are supported.

Realize that when we talk about supported functionalities, we are not talking about "Do you support italic text?" Rather, we are asking such questions as "Do you support embedding?" or "Do you support automation?" The server is then able to tell the client exactly what it is able to do. In the case of Outlook using Word to edit email, Outlook asks, "Can you be embedded?" and Word responds, "Yes, I can."



NOTE You might ask, "Peter, why are you adding yet another term, *OLE automation*?" This process allows the client application to take control of the server, and it lets the server see the client as a user. The client is able to actually click buttons and otherwise interact with the server application.

Now read on.

Embedding

With embedding, an object (which could be either a data object or server functionality) is embedded into the client application or the client application's data. When we embed Word into Outlook, we create a window, and using OLE, we tell Word to use this window to interact with the user. We also tell Word how it should appear to the user; for example, Outlook customizes Word's toolbars.

This embedding works regardless of whether Word is running or not. If Word is running, anything that Word is currently doing is unaffected by having Word embedded into Outlook's email editing system. In fact, the OLE server treats these as separate instances of the program, and keeps them separate. There are advantages, however. If the server is already running, it is not necessary to load a second copy of the server. Instead, the two instances share the executable code.

With embedded objects, there is a private copy of the object that the client owns. The server may update the client's object, though the server won't change any other instances of the data.

Each time an embedded object is used, there will be a new copy of the object. For complex objects, graphics and so on, this can consume substantial system resources.

Object Linking

Object linking is a mysterious technology where one application creates an object used by another

application later. A linked object remains the property of the creating application, and there is only one copy of the object.

The server is the creating application. The server links to the client application. When the server updates the object, the client gets a message and updates the object display in the client as necessary. Some objects are not visible, so there is no display update necessary.

The closest thing to showing how linking works is to look at Windows itself. There are a number of icons on your Desktop. Most are called shortcuts, which are denoted by that funny up-pointing arrow image in the lower-left corner. Think of these shortcuts as links. Open the properties for a shortcut and go to the Shortcut tab. In the Target edit box, you will see the name of the file that is associated with this shortcut (link). If you have a dozen shortcuts to the same file, each shortcut will open the same copy of the program. There won't be a dozen copies of the program.

OLE Controls, a.k.a. ActiveX

In our previous examples, the server application was a typical Windows program. Applications like this are native Windows applications. For example, Word for Windows is a server application. Word has a user interface and it runs on its own, without needing any client to embed the Word object.

Sometimes the server application doesn't have a native, stand-alone mode. That is, such an application doesn't have a user interface—no window, no direct way for the user to interact with the program. Applications like this are ActiveX controls; they used to be called OLE controls. ActiveX controls are commonly used with programs such as Internet Explorer and other Web browsers; however, many programs can use ActiveX controls.



NOTE An ActiveX control must be embedded and may never be run alone.

A typical user could have a large number of ActiveX controls installed, and the user might never know it. It is common for a user to download ActiveX controls from the Internet without ever realizing that this has happened.

VBX, What's a VBX?

VBX controls, or Visual Basic controls, were the first generation of ActiveX controls. When VBX controls were first developed, they served in dialog boxes as custom controls, things such as progress bars, and so on.

Generally, a VBX control doesn't handle data, while an ActiveX control might. In addition, only Visual Basic was able to create VBX controls. Programmers who developed in C/C++, for example, had difficulty creating their own VBX controls. However, Microsoft eventually did develop a system to create VBX controls using development platforms other than Visual Basic.

Eventually, Microsoft realized that the concept of VBX (embeddable controls) was a good one, and that these controls were here to stay. In came the OCX (OLE Control) technology; it was development-platform independent, usage-platform independent, and more flexible.

Evolution and the name game reared their heads again. Microsoft moved to ActiveX controls more as a change in name than in function. It is common to see ActiveX controls referred to as OCX controls, and vice versa.

Some ActiveX controls display data. Some don't do anything other than provide some form of user interface. For example, these controls were on one of my computers:

- BtnMenu Object
- CarPointProximityCtrl
- ChatShowClt Object
- DirectAnimation Java Classes
- HHCtrl Object
- Internet Explorer Classes for Java
- IPTDImageControl.SImage

- Label Object
- Microsoft MSChat Control Object
- Microsoft Search Settings Control
- Microsoft XML Parser for Java
- PopupMenu Object
- Win32 Classes

All of these controls were installed in the %systemroot%\Occache directory. If you are not using Internet Explorer or are not active on the Internet, you probably won't have many of these controls.



NOTE If you don't find an Occache directory, don't panic. It is probably because there are not any ActiveX controls installed on your computer!

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)**Liar, Liar, Pants on Fire!**

Remember when I said previously that OLE controls don't have a user interface? Well actually, I lied a little. It is possible to use RunDll32 to execute some OLE controls. RunDll32 doesn't have a user interface either, and any control that works with RunDll32 must be written specifically for this type of usage. For example, the OLE Active Movie control will run with the command:

```
%systemroot%\System32\rundll32.exe amovie.ocx,RunDll
```

This will open the Active Movie OLE control (RunDll provides a main window for the control), and Active Movie will then display an Open File dialog box. You might select an Active Movie file (try clock.avi in the Windows 2000 %systemroot% directory) and run it using amovie.ocx. This is possible because Active Movie was written to work with RunDll, and as such, it works. Try this trick with most any other OLE control, and you will get the message, "Missing entry point RunDll," which indicates that the entry point passed in the command was not found.

Oh, yes, you can also pass parameters to your OLE control with the command:

```
RunDll : %systemroot%\System32\rundll32.exe amovie.ocx,RunDll  
Ø %systemroot%\clock.avi
```

This command would load Active Movie, load clock.avi, and allow the user to interact with the control. Try it. Better yet, try this:

```
: %systemroot%\System32\rundll32.exe amovie.ocx,RunDll /play /close  
Ø %systemroot%\clock.avi
```

Don't mistakenly insert spaces between the executable file (amovie.ocx in the previous example), the comma, and the entry point (RunDll in the previous example). This will break RunDll without telling you why it failed.

Get the hint? We passed a parameter to the Active Movie control to play the clock.avi file, then to close it when the .avi file is finished. Active Movie loaded the file specified, played the file, and closed it—all without user intervention.

Oh, don't blame me if the clock.avi file is a bit annoying.

Actually, RunDll will run more than OLE controls—RunDll will (or will at least attempt to) execute any

executable file, including DLL (Dynamic Link Library) and EXE (executable) files. This is true as long as you know the file's entry point and the file to be executed follows the RunDll protocol. For more information, see Microsoft Knowledge Base article Q164787, which can be viewed at <http://support.microsoft.com/support/kb/articles/q164/7/87.asp>. Though originally written for Windows 95/98, it contains information helpful to Windows 2000 users.

Client-Server OLE Applications

Client-server OLE applications make up a substantial number of programs on most Windows computers. Even though the user may not be aware of what client-server OLE applications are installed, there are many.

One of the best-designed and best-integrated sets of applications is Microsoft Office, currently released as Office 2000. It combines word processing (Word 2000 for Windows), spreadsheets (Excel 2000), a database system (Access 2000), a presentation program (PowerPoint 2000), and a host of utilities (such as Chart). Each of the main applications in Microsoft Office works as both a client and a server application. Some applications—such as the Word Art and Chart utilities—are not designed to run as simple clients. For example, take Word 2000 (a program that at least I know how to use).

Word, as a client is . . . Word. Open Word and edit a document. Write a short letter to someone, it doesn't matter whom. Create something, about a page long, three or four paragraphs. You have Word's functionality in all these paragraphs; you did everything using Word and nothing else.

Now things start to get exciting. Insert an object. For grins, insert a drawing into a Word document. Click Insert \emptyset Object. Word displays the Object dialog box that lists all the embeddable OLE server objects (see Figure 6.1). Actually, OLE uses an API call to display the dialog box.



FIGURE 6.1 Inserting a Microsoft Clip Gallery object is as easy as selecting it from the Object dialog box.

Some servers work by totally embedding themselves into Word. For example, Microsoft Paint will replace Word's menu with the Microsoft Paint menu and display the Microsoft Paint tool bars and so on—all without any user intervention (whammo, check out Figure 6.2).

It is quite incredible that Microsoft Paint works without Microsoft Word having prior knowledge of it. Actually, select a Microsoft Paint object and Word gives control to Microsoft Paint. Microsoft Paint then displays its own menu items along with Word's Window menu so that the user can switch between Word documents if necessary—just like I'm doing while I write this chapter.

When the Microsoft Paint object is not selected, Word takes control of the user interface, and the Word menu structure is restored (see Figure 6.3). You do context-switching between Word and Microsoft Paint whenever you select something in the document. If you select something from Microsoft Paint, Microsoft Paint is put in control; otherwise, Word takes control.



FIGURE 6.2 Microsoft Paint gives the user a new menu, new toolbars—it just plain takes over.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

So, in a nutshell:

- Client applications may have OLE objects embedded into their documents, and/or have OLE server functionality embedded into their basic functionality.
- This embedding is done at runtime, so the developer knows nothing about what embedding will be done when the program is being written.
- When a client application wants to embed an OLE object, the client application will call OLE to display the Insert Object dialog box to the user. The user then selects the embedded object.
- By selecting the object, OLE allows the client's user interface (menus and toolbars, for example) to be turned over to the server application.
- Server applications may edit the object in place, or may create a special editing window, which may have menu/toolbar support, as appropriate. Usually, complex objects have their own windows for editing just to keep things simpler for the user.
- OLE uses the registry to learn about embeddable server applications.
- OLE server and client applications are identified by CLSIDs; call 'em UUIDs, or GUIDs, if you want. A CLSID is a unique long string of numbers.
- The server application is able to use OLE to tell the client what capabilities the server has. This allows the client to behave in a predictable manner.



NOTE It is possible to embed a purely functional OLE object into a document. From time to time with database programming, OLE controls (ActiveX) and OLE applications (without instance-specific data, such as Microsoft Comic Chat) are used.

OK, I've prattled on about OLE long enough (is that a wild cheer I hear?), so let's get to the registry component of OLE.

OLE and the Registry

Wow, now we're back to the registry. That was a lot of stuff to cover, just to get a handle of the basics of OLE.

OLE works extensively with the registry. When an application registers itself with OLE as a potential server application, this registration process consists of adding a number of entries into the registry. For OLE applications, such as ActiveX controls, these entries are relatively simple and easy to follow. More complex OLE applications, take Microsoft Word as an example, have hundreds of entries in the registry and are

typically difficult to understand.

Let's look at a simple OLE control—the ActiveX control called the Comic Chat control. Comic Chat is an application available from Microsoft to allow users to use the Internet chat facility. There are hundreds of chat servers—try chat.msn.com, chat.microsoft.com, or any other IRC chat server. Chat can be fun.

Yes, Comic Chat is embeddable into a Word document. This usage, though not typical, is legitimate. Check out Figure 6.4 to see Word and Comic Chat working together. In the session that I used for this figure, I actually was chatting with two computer users in Australia and one user in North Carolina in the U.S.



FIGURE 6.4 Embedding Microsoft Comic Chat adds a new dimension to your Word documents.

Size and Placement...

Some versions of ActiveX controls, such as Comic Chat, don't work well when embedded into Word. They do not properly size themselves and tend to resize their display in unexpected ways. This is not acceptable behavior, I might add.

In Comic Chat's case, the control creates a base window, and then sizes the main Comic Chat window to fit the base window. Unfortunately, when a user tries to resize the Comic Chat window, Word restores it to the size of the underlying base window.

One way to resize Comic Chat is to first make it smaller, then quickly before Word resizes it, click on the underlying base window. This locks the base window, and then you may resize it as appropriate. Once satisfied with the Comic Chat window size, a simple double-click anywhere inside the base window will restore the normal display.

Microsoft Comic Chat is a large application that doesn't allow a lot of interaction with other documents or other applications. Nevertheless, why would you, a user, want other interaction? Easy! One, classic example is to embed Comic Chat into a Web page, a document whose application is the Web browser. Another example is to embed Comic Chat into an email message. Ding! Did the light go off? E-mail everyone on your team and include in the message the details of a virtual meeting both on an IRC chat server and on the client.

So, we have a Microsoft Comic Chat OLE server application implemented as an ActiveX control. Let's look at the registry entries for Comic Chat.

First are the entries in HKEY_LOCAL_MACHINE\Software\Classes\CLSID. These entries define much of the OLE interface:

```
{ 241AF500-8FB6-11CF-ADC5-00AA00BADF6F } ]
@="Comic Chat Room"
```

The lines above are the hive (HKEY_LOCAL_MACHINE), key (Software), and subkeys (Classes\CLSID), followed by any values that these keys might contain. A value in the form of @=data denotes the default value entry found in every registry key and subkey.



NOTE The Windows 2000 CLSID is identical the Windows NT 4 CLSID. Surprised? Don't be, as the CLSID comes from the application, not the operating system. Also, Comic Chat version 2.1 has the same CLSID as Comic Chat version 2.5.

The first subkey contains the CLSID for the Comic Chat server. This CLSID happens to be 241AF500-8FB6-11CF-ADC5-00AA00BADF6F, although other versions of Comic Chat might have different CLSIDs. The default data variable contains a string describing the program. Notice that this string is also found in the second section of the registry, HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1, described next.

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\AuxUserType  
@=""
```

Windows uses the AuxUserType subkey for short, people-readable names for the application. Menus, both regular and pop-up, use these short names. Microsoft recommends that the names in AuxUserType be limited to not more than 15 characters.

This example shows one name for the application—the string Room:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\AuxUserType\2  
@="Room"
```

This example shows another name for the application—the string chat:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\AuxUserType\3  
@="chat"
```

Take this entry:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\DefaultExtension  
@=".ccr,Microsoft Chat Room (*.ccr)"
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

In this entry, the DefaultExtension subkey holds the suggested extension (file type) for the default when File Save As or File Open is selected in the menu.

Take this next entry:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\DefaultIcon
@="C:\\PROGRA~1\\MICROS~2\\CChat.exe,1"
```

In this entry, the DefaultIcon subkey holds information about the icon. This icon is an icon index in the executable file listed. Though the executable file containing the icon need not be the same executable file as the OLE control, it is good if it is.

The following entry, which works with the Microsoft OLE DocObject technology, can contain information about the capabilities of the OLE object:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\DocObject
@="0"
```

In the next entry, the InprocHandler32 subkey tells the system what in-process handler will be used.

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\InprocHandler32
@="ole32.dll"
```

Many applications use Ole32.dll as their in-process handler, although this is not a requirement. Another commonly used in-process handler is MAPI32.DLL, which is used by many mail-enabled objects:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\Insertable
@=""
```

Intended for use with Windows 95/98 and/or Windows 2000, the following entry indicates to the system that the application is listed in the insert list of the Insert New Object dialog box:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\LocalServer32
@="C:\\PROGRA~1\\MICROS~2\\CChat.exe"
```

The next entry contains the application's fully qualified path and executable filename. This string is not a REG_EXPAND_SZ, so don't use substitution variables:




```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\MiscStatus
@="32"
```

Table 6.1 lists the flag values allowed in MiscStatus.

TABLE 6.1: Flag Values Used in the MiscStatus Object

Flag (Value) in Decimal	Flag (Value) in Hex	Description
1	0x0001	When resizing, the object is recomposed.
2	0x0002	The object is only available as an icon.
4	0x0004	The object is used in insert mode, not replace mode.
8	0x0008	The object is static.
16	0x0010	The object can't link inside.
32	0x0020	OLE 1 can link the object.
64	0x0040	The object is a link object.
128	0x0080	The object is inside out.
256	0x0100	Activate the object when it is visible.
512	0x0200	The object's rendering is device independent.



NOTE In MiscStatus, combine values using binary or bitwise addition; the easiest way to do a bitwise is to simply add the values. For example, an application with the flags “OLE 1 can link the object” (32) and “Activate the object when it is visible” (256) would store a value of $(256 + 32) = 288$ in MiscStatus.

In this entry, the Printable subkey denotes an OLE object that will support the IPrint method:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\Printable
@=""
```

For an insertable object, there must be an associated ProgID value (ProgID is shorthand for “programmatic identifier”). This value consists of a short name, a type, and a numeric value:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\ProgID
@="ComicChat.Room.1"
```

A registry section is created with this name (see the next entry), where more registry values will be stored for this object:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\Verb
@=""
```

Verbs indicate types of action that the object may take. Always numbered consecutively in the registry, there are three components to verb entries, as shown here:

```
{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\Verb\0
@="&Edit,0,2"
```

This sample verb, Edit, shows three things. First, the text used in the menu, &Edit. The & indicates that the letter following it will be underscored and used as a hotkey value.

Second, the first number, 0, is the menu flag's value. Shown in Table 6.2 are the valid values. (Not all are used with OLE menus, such as MF_OWNERDRAW.)

TABLE 6.2: Flag Types Allowed

Flag Name	Value	Description
-----------	-------	-------------

MF_STRING	0x000	The menu item is a string.
MF_ENABLE	0x000	The menu item is enabled.
MF_UNCHECKED	0x000	The menu item is unchecked.
MF_INSERT	0x000	The menu item is an inserted item.
MF_BITMAP	0x004	The menu item is a bitmap.
MF_CHECKED	0x008	The menu item is checked.
MF_DISABLED	0x002	The menu item is disabled.
MF_GRAYED	0x001	The menu item is dimmed.
MF_OWNERDRAW	0x100	The menu item is an owner-draw item.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.


Datamation
Profit and Value from Information Technology
[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

 This site is brought to you by
 EarthWeb

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)
[KEYWORD SEARCH](#)

Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)
[Advanced Search](#)
[PUBLICATION LOOKUP](#)
[Previous](#) [Table of Contents](#) [Next](#)

Third, the second number, 2, is the verb flag. There are only two possible values for this entry, as shown in Table 6.3.

TABLE 6.3: Verb Flag Names

Verb Flag Name	Value	Description
OLEVERBATTRIB_NEVERDIRTIES	1	Indicates that the verb does not modify the object, so the object will not require storing in persistent storage.
OLEVERBATTRIB_ONCONTAINERMENU	2	Indicates that the verb should appear on a pop-up menu.

There is a second section of the registry for the Comic Chat OLE object. This section, in HKEY_LOCAL_MACHINE\Software\Classes, is called ComicChat.Room.1.

In the ComicChat.Room.1 subkey, there are two value entries:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1]
@="Comic Chat Room"
"EditFlags"=hex:00,00,01,00
```

The first value entry is the default value (@=) that contains the name ("Comic Chat Room") used in the insert list of the Insert Object dialog box. The second value, EditFlags, contains the edit flags, expressed as hex values:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\CLSID]
@="{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}"
```

The CLSID subkey contains the object's CLSID.

The next subkey, which works with the Microsoft OLE DocObject technology, may contain information about the capabilities of the OLE object:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\DocObject]
@="0"
```

Intended for use in Windows 95/98 and/or Windows 2000, the following entry indicates to the system that the application should be listed in the insert list of the Insert New Object dialog box:



```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\InsetValue]
@=""
```

The protocol subkey is used for compatibility with OLE 1 container (client) applications:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol]
@=""
```

There is one subkey in protocol, called StdFileEditing. Within StdFileEditing, there are a number of items, as shown here:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol\StdFileEditing]
@=""
```

The default entry in StdFileEditing is an empty string.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol\StdFileEditing\server]
@="C:\\PROGRAM~1\\MICROS~2\\CChat.exe"
```

The first subkey in StdFileEditing is the server subkey. Inside server is the default string that contains the fully qualified name of the server executable file. Because this string is REG_SZ, do not use any substitutable variables, such as %systemroot%, in it.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol\StdFileEditing\verb]
@=""
```

The next subkey in StdFileEditing is verb. Inside verb are one or more numbered subkeys; numbers begin with 0 and should be consecutive. Each verb that the OLE application uses in a menu will be included, as shown here:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol\StdFileEditing\verb\0]
@="&amp;Edit"
```

This verb is the Edit menu selection. The text used in the menu is &Edit. The & indicates that the letter following it will be underscored and used as a hotkey value.

Finally, a version-independent ProgID is created. Even when the control is updated, this entry won't change:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\VersionIndependentProgId
@="ComicChat.Room"
```

Like ProgID, this identifies the program, without any version references.

How Linkages Are Made between Applications

OK, now we'll look at a few of the mechanisms that Windows 2000 uses to manage OLE applications, CLSIDs, and the user interface.

First, let's confuse applications and documents. Considering them identical for now will ease some of the issues here. OLE is one complex puppy, so anything we can do to understand it is OK. Later in this chapter, I'll spend some time pointing out what the differences are between a document and an application.

OK, so the user's application wants to use OLE. There are a couple of ways that applications can use OLE:

- Write the application from the get-go to use OLE controls. Some applications do this; however, many do not.
- Write the application to allow the user to embed OLE objects into it. Many OLE applications do this.

Neither of these two scenarios is mutually exclusive. For example, an application could have both methods built into it. In either case, it is necessary to register the server of the OLE object that the client will be using.

When registered, the server's basic properties will be listed in the registry, in the HKEY_LOCAL_MACHINE\Software\Classes and HKEY_LOCAL_MACHINE\Software\Classes\CLSID sections. This information in the registry provides the client with the minimum (got that, *minimum*) amount of information needed to interact with the OLE server.

However, the client application needs to know more about the server. Questions that must be answered include what the server does, expectations of the server, whether support for in-place editing exists, and what information or data is communicated between the server and the application.

An ActiveX control, for example, probably won't have any data that is stored in the client's document. Most ActiveX controls display information for the user. However, the displayed information varies greatly. Some ActiveX controls display contents that vary only in detail. A classic example of this type of ActiveX control is a real-time clock control—the control retrieves the time from the system and displays the time in a specified format. Another controls data changes in content, but not type. For example, the Comic Chat control always displays IRC chat sessions, though the sessions might vary widely. However, a Microsoft Paint server's data and type both would change from invocation to invocation. Who knows what the user might try to display in the Microsoft Paint control? The display could be anything from a company logo to a cheery holiday greeting.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

Regardless, each server must communicate with the client application. The client always initiates communications between a server and a client; otherwise, how'd the server know a client needed it? This communication uses a technique called *querying the interface*. The server will respond with information about exactly what the server can do.

Everyone Uses OLE

Everyone uses OLE, we just don't realize it. The Windows 2000/NT and Windows 95/98 operating systems use OLE to perform a number of useful tasks. OLE is a built-in, not an added-on-later, part of Windows 2000.

Explorer, the Windows 2000 user interface, relies on OLE for many of its abilities. For example, look at your Desktop. Do you understand what is going on there? Probably not. Do you care? Maybe, and a bit of understanding can help later when you decide to customize it. One program, Explorer, is responsible for much of the functionality that you see on your Desktop. Explorer is the program that paints your Desktop background; puts up those icons (such as the pesky and difficult to remove Recycle Bin, My Briefcase, and My Computer); and manages aspects of the user interface, such as property sheets and context menus. This is all done with the very valuable assistance of OLE.

Let's give OLE a big hand—it does a lot for us.

Embedded Documents

Embedded documents will have references to each OLE object that they have. Unlike when OLE controls are used with an application (remember, we blurred this distinction in the previous section), OLE objects in a document can and do vary greatly. Each document is unique—one document may contain no OLE objects, while the next may contain many different objects.

Transportability is a critical issue. Say I create a chapter for my publisher and embed an OLE object into the document. Then I email that document to my editor. When the editor opens the document and wants to have access to the object, the OLE server application will have to display the object on the computer. It is not necessary that the OLE server be in the same directory, or in any specific directory. OLE uses the registry to take care of locating the server and activating it as necessary. I might have the OLE server installed in a directory on my Q: drive, while my editor might have the same server located on the C: drive, and the executable filename may well be different in each installation, too. Regardless, as long as the ProgID value is identical, Windows 2000 will be able to locate the server and launch it.

Critical items in the registry are those entries shown in the previous sections of this chapter. If you find it

necessary to move an OLE server's files from one location to another, it may be possible to edit the registry and change the file locations that are stored in entries, such as:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\DefaultIcon]
```

@=

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{241AF500-8FB6-11CF-ADC5-00AA00BADF6F}\LocalServer32]
```

@=

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\ComicChat.Room.1\protocol\StdFileEditing\server]
```

@=



NOTE Before making any change such as this, be sure your backups are up-to-date.



WARNING Don't even consider moving system OLE servers and objects. Leave anything supplied with Windows 2000 that is OLE related where it is. This is because it is possible that there are references to these objects in places other than the registry.

Fixing an OLE Disaster

Common OLE problems arise when a user inadvertently deletes the OLE server files, often in an ill-advised attempt to clean up hard disk space, while the OLE registry entries remain in the registry. There are several tricks to recover from this. First, attempt to reinstall a new copy of the OLE server in the original location. This will probably work in most cases. However, if you cannot reinstall—maybe you don't know where the source files are located—consider restoring the files from a backup.

As a last resort, try to remove the registry entries for the OLE server. This probably will result in your registry and system becoming unstable, but if it is unusable anyway, what do you have to lose? Check the sections listed here for entries about the OLE server:

- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\
- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\

Doing this will require some detective work. You will have to search the registry using either RegEdit or by exporting the registry to a text file and using a text editor's search. While searching, note all locations listing the OLE server. There will be at least the two mentioned in the previous list, although some OLE components may have more entries.

Disaster typically raises its ugly head when there are multiple dependencies between a numbers of OLE objects. The fix here is to restore if possible.

Another disaster point is when a new application installs an OLE object that conflicts with an existing one. Typically, the two OLE objects would have different CLSIDs. However, it is possible that the CLSIDs are identical, although in theory this should not happen. Installing a second copy of an OLE object modifies the object's ProgID. The user will frequently see two OLE objects in the Insert Object dialog box with the same name. Often, only one of the objects will work correctly.

Hints and Kinks from the Experts

OK, there is a shortage of hints from the experts on OLE. I thought about making up some hints, but decided not to. I've included one hint from John Savill.

A Service or Driver Failed to Start and the Event Viewer Has Taken a Vacation

If you receive a message that a service or driver has failed to start, you are instructed to check the Event

Viewer for details.

If Event Viewer has gone on vacation (all your All Users folders are gone) and/or no Network icon displays in the Control Panel, you may have a missing or corrupted Ole32.dll in the %systemroot%\System32 folder. Install a copy from the CD-ROM or from your latest service pack or hotfix, and then reboot your system. (Courtesy of John Savill.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 7**Why, Oh Why, Are There System.ini and Win.ini Files?**

JUMP TO TOPIC

**Desktop Library**

Click to access!

FEATURING:

<i>System.ini</i>	176
<i>Win.ini</i>	177
<i>Hints and kinks from the experts</i>	179

OK, so we have a registry, and that registry was supposed to replace the System.ini and Win.ini files that the first 16-bit versions of Windows were plagued with.

If you have been a Windows user for more than a few years, you're probably well aware of the issues that have evolved concerning the Win.ini and System.ini files. These files contained almost all of the information used to configure earlier versions of Windows; other configuration files, such as Protocol.ini, were used to store network information as well. When the time to design Windows NT arrived, those wonderful software guys at Microsoft decided that there were some problems with using .ini files. Several problems were apparent:

- Users would edit these files, often without regard for the consequences of making changes. Sometimes these changes were totally inadvertent.
- Some editors (typically those used to doing word processing) would add, remove, or even change some characters without explicitly telling the user. An example is quotes used around strings, for which the editor would stick in stylized quotes.
- Another problem that became more apparent as time went on was that the System.ini and Win.ini files were growing at an alarming rate. As users added software, fonts, and system components, these files grew. The result was that the primitive search routines employed in the early versions of Windows could not efficiently search for entries in the files.
- A fourth problem was that applications were able to modify system entries in the Win.ini and System.ini files with impunity. A rogue application could butcher these files and no one would be the wiser until the damage caused a failure of the operating system—no protection or security was

available.

These problems with Win.ini and System.ini prompted Microsoft to move to a more efficient method of storing information that both Windows and user applications could access easily and efficiently. The registry—a binary, tree-oriented database—is quick and easy to work with. Changes to existing products as they were migrated to 32-bit environments, such as Windows NT and Windows 95/98, presented a few problems for programmers to resolve. Moving to the registry-based model also presented a few problems for those applications that were already in existence. These applications are called *legacy applications*. They already exist either on the user's computer(s) or as products that are being sold but will not be updated. These problems include:

- Existing 16-bit applications must be supported in executable form. That is, an application that expects Win.ini and System.ini files to exist must be able to use them.
- Some 16-bit applications do not have access to registry manipulation APIs. These applications must be supported using the preexisting Win.ini and System.ini files also.



NOTE Today, there are still many 16-bit applications being sold on the Windows platform. This is over three years after the introduction of Windows 95, a 32-bit platform that supports 32-bit applications very well. In the foreseeable future, there will always be at least one 16-bit application being sold or used somewhere. Old habits, and old software, die hard.

To handle these problems, Microsoft wisely decided to retain support for both Win.ini and System.ini. Windows would no longer use these files, but they would be available to any applications that chose to access or utilize them.

In this chapter, we'll take a look at the System.ini and Win.ini files that Windows 2000 provides. The default files are not too large, although the Win.ini file might become larger when the user installs more 16-bit applications. The System.ini file might also grow as the user adds items that are not designed to work with Windows.

When Windows applications write to the Win.ini or System.ini file, and those applications use the Windows 2000 registry-updating APIs, the information that would have been stored in the .ini file will be stored in the registry. This is subject to the exclusions discussed next.

Any file listed in the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping section (that is, an .ini file or registry entries) will also be updated using the Windows 2000 registry-updating APIs. Windows 2000 will search the IniFileMapping section for the application's section. If the section is found, it is used. If no application section is found, Windows will search for an .ini file to use.



NOTE Any application that directly opens an .ini file, perhaps using the 16-bit .ini file-processing APIs, will bypass the registry file entirely.

System.ini

Located in the System.ini file are a few entries that the Windows 2000 Setup program supplies by default. Here is a typical, basic System.ini file:

```
; for 16-bit app support
[drivers]
wave=mmdrv.dll
timer=timer.driv
```

```
[mci]
[driver32]
[386enh]
woafont=dosapp.FON
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

This file contains four sections—[drivers], [mci], [driver32], and [386enh]—and only a few entries in these sections. Entries are primarily for fonts (in the [386enh] section), as well as two drivers used with Windows 2000:

mmdrv.dll—a driver that is used for multimedia (sound) support

timer.driv—a driver that is used to provide timer support

In addition, your System.ini file may contain other entries and other sections if you are using 16-bit-incompatible applications. These applications would use the Win.ini file to write application-specific information, typically in sections created for the application.

Most of the Windows 2000 system entries were moved from the System.ini file to the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WOW. This key contains many entries that would be found in a Windows 3.1x installation.



NOTE Windows 2000 cannot use any 16-bit screen savers because they do not perform correctly when used in the Windows NT environment. Any entry found in the [BOOT] section of the System.ini file will not be migrated to Windows 2000.

Win.ini

Few entries are located in the *Win.ini* file, except for computers that have been used for some time and have had additional software or components installed.

The default Win.ini file contains only four sections with no entries, and two sections containing entries:

```
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
[MCI Extensions.BAK]
asf=MPEGVideo2
asx=MPEGVideo2
ivf=MPEGVideo2
```



```
lsf=MPEGVideo2
lsx=MPEGVideo2
m3u=MPEGVideo
mp3=MPEGVideo
mpv2=MPEGVideo
```

A computer with a few more miles on it will have additional entries, like these:

```
[WinZip]
win32_version=6.0a
Name=Peter D. Hipson
SN=99999999
[Collage Capture]
Save On Exit=1
Prompt On Entry=0
Collage Capture Settings File=G:\COLLWIN\ORIGINAL.SET
```

This computer has some additional applications installed. These applications are a mixture of system components and added-on programs from a variety of sources:

Mail: The entries in the [Mail] section describe the mail interface that is installed on this computer.

WinZip: WinZip is a front end to the very popular DOS-based PKZIP program. WinZip adds both a Windows interface and the ability to handle long filenames to PKZIP.

The first application, Mail, is most certainly part of the 32-bit email system. Why does it have entries in the Win.ini file? This allows 16-bit applications to know something about the already-installed email interface. Remember: 32-bit applications can use .ini files, including Win.ini, although it is strongly recommended that they do not.

Win.ini system-based settings are stored in the registry in a number of subkeys. Table 7.1 shows some of these settings and their locations. This is mostly of interest to users who are using dual environments, such as Windows 2000 and Windows NT.

TABLE 7.1: SECTIONS FOUND IN Win.ini

Section in Win.ini	Registry Path	Description
[extensions]	HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Extensions	File associations used by Explorer
[fonts]	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Fonts	Fonts used by Windows
[fontsubstitutes]	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\FontSubstitutes	Fonts used by Windows
[mci extensions.bak]	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\MCI Extensions	The Media Control Interface settings and extensions
[mci extensions]	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\MCI Extensions	The Media Control Interface settings and extensions

Some items are never migrated in a dual-environment system. These items are usually not moved to the registry either due to their complexity or for other reasons. Other items are migrated, but not used. Such items include:

- [Ports], [Devices], and [PrinterPorts] which are migrated during the migration process as part of installation; these settings are not used for any purpose.
- Persistent shares and users as used by Windows for Workgroups, but not Windows 3.1x.
- The default domain and user ID from Windows for Workgroups or the LANMAN.ini.
- Individual user profiles that are maintained by WinLogin.
- Changes that users make in their copies of the Main, Startup, Games, and Accessories Program Manager groups.

- MS-DOS drive letters, which are managed using the Windows Disk Administrator. (Drive letters usually vary between Windows 3.1x and Windows NT or Windows 2000 due to how drives are detected and the possible presence of Windows 3.1x–incompatible drive formatting, such as NTFS drives.)
- Auto Arrange, Minimize on Run, and Save Settings on Exit options for Program Manager. These settings are not type compatible. (Program Manager uses strings, while Windows 2000 uses DWORD values for these settings.)
- DOS command window font details.

Hints and Kinks from the Experts

Here are some System.ini and Win.ini hints from our experts.

Where Does Windows 2000 Store the System.ini Info for 16-Bit Programs?

The WOW key at HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WOW stores configuration data for the Win16 on Win32 subsystem, a.k.a. Windows on Windows.

The WOW key contains subkeys that have the same names as headings in the System.ini file. The value entries in these subkeys are the same as the values in the 16-bit Windows 3.x System.ini file.

(Courtesy of Jerold Schulman.)

Does Unattended Setup Sometimes Ignore Win31Upgrade=no in Your Unattended Answer File?

If you do not want to upgrade an existing Windows 3.x installation during Unattended Setup, you should have a Win31Upgrade=no statement in your Unattended Windows 2000 Setup Answer file.

You must also hide the old Windows installation from Windows 2000 Setup. If you don't hide it, Setup will only honor your Win31Upgrade=no if it has insufficient room on the Windows drive, or if it suspects that Windows 95 is installed. If it finds Shell32.dll, User32.dll, Kernel32.dll, and Gdi32.dll in the system directory, Setup will think Windows 95 is installed.

If Setup finds Win.com, Win.ini, or System.ini in the Windows directory and autoexec.bat in the root, it will think Windows 3.x is installed.

Since Setup only searches for Windows in the PATH as defined in C:\autoexec.bat, the best way to hide the Windows 3.x installation is to edit autoexec.bat and remove all references to Windows from the path.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 8

Getting Rid of the Unwanted

JUMP TO TOPIC


Desktop Library
 Click to access!
FEATURING:

<i>RegClean</i>	183
<i>RegMaid</i>	187
<i>CleanReg</i>	194
<i>Hints and kinks from the experts</i>	199

Sometimes we don't have what we want in the registry. Other times, we have too much of what we *don't want* in the registry. This chapter covers the second case.

We install software, try it, don't like it, and remove it. Things come and things go, sometimes intentionally, sometimes by accident. But whatever the cause or reason, any computer that has been running Windows 2000 for more than a few months will probably have a few entries in the registry that do nothing more than clutter it up. Additionally, there will be a few unlucky users who will have some entries that are doing something that they really don't want to happen.

The classic problem is that we are not always good at removing things we install. Many software programs come with uninstall programs, but many others don't. Sometimes we lose track of an application—usually because, in a moment of weakness, we delete the application's directory without properly uninstalling the application. Desperation for even a few more MB of hard disk space will make us do strange things.

Have you ever installed an application on a secondary drive only to later have that secondary drive fail? Maybe you have a good backup, maybe not. Perhaps you just want to do a general housecleaning. For whatever the reason, this chapter will deal with the very difficult task of trying to remove unwanted things from the registry without having to reinstall Windows 2000.



WARNING Have I already said this? *Back up your registry before doing anything described in this chapter.* Manually removing items from a registry is perhaps the *easiest way to trash everything.* Back up, back up, and back up again.

This chapter covers three utilities that help clean up the registry: RegClean, RegMaid, and CleanReg.



WARNING Windows 2000 is a new operating system—utilities specific to Windows 2000 are only beginning to become widely available. Utilities described in this chapter were originally created for Windows NT, and should be used with caution. I've tested each of the described utilities; however, that is no guarantee that these utilities will work with your Windows 2000 configuration.

RegClean

Microsoft created a program that automates cleaning the registry. Called RegClean, it is available from several sources; I recommend that you retrieve it directly from Microsoft's Internet site at <http://support.microsoft.com/download/support/mslfiles/regclean.exe>.

There are several versions of RegClean; this URL leads to the most recent version, RegClean 4.1a. This version was released in early 1998. It is fully compatible with Windows 2000, although there may be a need to update to a later version of the OLE driver OLEAUT32.DLL. This update is included with the distribution of RegClean. Documentation on how to install the driver is also included in the RegClean readme file.

Using RegClean is simple—just follow these steps:

1. Download the RegClean.exe file from the URL given above, or from Microsoft's FTP site at <ftp://ftp.microsoft.com/Softlib/MSLFILES>.
2. Execute the RegClean.exe file to start the self-extractor program. Alternatively, you may use either WinZip or PK_UNZIP on the RegClean.exe distribution file to extract the program and other files. Files contained in the RegClean.exe distribution file include:
 - OADIST.exe—the update for OLEAUT32.dll, if needed
 - Readme.txt—a text file with instructions on how to use RegClean and information about OADIST.exe
 - RegClean.exe—the real RegClean.exe program, which is an executable Windows 2000 application
3. Execute the RegClean.exe program that is extracted.



NOTE RegClean.exe writes a program called RegClean.exe. Confused? Well, you should be. The file RegClean.exe that you download (about 800KB in size) is a self-extracting Zip file. One of the files contained in RegClean.exe will be RegClean.exe—the actual program. In order for both files to coexist, the self-extracting RegClean.exe file must write its output to a different directory or drive. RegClean.exe cannot extract to its own directory.

Do I Need to Update OLEAUT32?

If you receive the message(s):

REGCLEAN.EXE is linked to missing export OLEAUT32.DLL:421

and/or

A device attached to the system is not correctly functionin


it is probable that you will need an updated OLEAUT32.dll file.

OLEAUT32.dll is installed with Internet Explorer 3.x or later, so most of the users who are affected by this problem have earlier versions of Windows, such as Windows NT.

Installing OLEAUT32.dll is a simple process—just execute the OADIST.exe file that is extracted from the RegEdit.exe file (see step 2 in the “RegClean” section).

Running RegClean

Executing RegClean is simple; it doesn't care what directory it is run from. However, RegClean will save undo information to the directory that it has been executed from.

Start RegClean either by choosing Start  Run, by using a command-prompt window, or from Explorer. Once started, RegClean will display a window similar to the one shown in Figure 8.1. In this window, the lower status bar and the descriptive text just above it will indicate the progress of RegClean's initial pass

through the registry.



FIGURE 8.1 RegClean has just two buttons: Fix Errors and Cancel. The Fix Errors button is initially labeled Start, even though RegClean starts automatically.

Once RegClean finishes the scan of the registry, it will advise the user either that it has not found any registry errors (this usually happens if you run RegClean frequently) or that RegClean can correct the errors found. Clicking the Fix Errors button will tell RegClean to clean the registry. Clicking on Cancel will cause RegClean to exit without doing anything else.

If RegClean doesn't find any errors, the message shown in Figure 8.2 will appear. This message tells you that there were no errors detected in the registry.

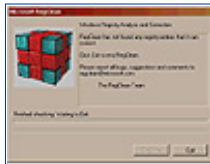


FIGURE 8.2 If RegClean finds no errors, this message is shown.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

As RegClean cleans the registry, it will write a registry file to the drive that RegClean was run from. This registry file may be used to restore the registry to the same condition that it was in before running RegClean.

The registry save file created by RegClean is named in the following manner:

```
UNDO computer yyyyymmdd hhmmss.reg
```

Here, *computer* is the name of the computer whose registry was cleaned; you may keep a single copy of RegClean and then link to and execute it from many other computers. The *yyyyymmdd* is the year, month, and day that RegClean was executed; and *hhmmss* is the time of day that RegClean was executed.

Undoing RegClean

After RegClean runs, it is important to make sure that all applications and systems are still functioning correctly. If you find that something has broken (this is unlikely, but could happen), it is imperative that you restore the registry to its original state immediately. To do this, simply use Explorer and double-click the Registration Entries backup file created by RegClean (see Figure 8.3).

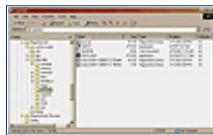


FIGURE 8.3 RegClean creates the Registration Entries files whose filenames all start with Undo

Be careful that you select the correct .reg backup file if there is more than one. Remember, you can tell Explorer to list files in date/time order, making the selection process much simpler.

Sometimes, users find that they are unable to undo the changes. Windows 2000 will give an error when the registry backup file created by RegClean is double-clicked. The user (that's you) will get one or more errors that indicate a problem has occurred. These errors are caused by a problem with the registry, not with the .reg file.

To fix this problem, follow these steps:

1. Open Explorer and select View Options.
2. In the Options dialog box, select the File Types tab.
3. In the Registered File Type list box, select Registration Entries, then click the Edit button.
4. In the Edit File Type dialog box, select Merge in the Actions list box. Then click the Edit button to



Desktop Library
Click to access!

open the Editing Action for Type: Registration Entries dialog box.

5. In the Application Used to Perform Action box, enter the name **regedit.exe “%1”** (including the double quotes).

6. Click OK in all open dialog boxes.

After doing this, you should be able to restore registry entries from a .reg registry backup file. It is rare that the Registration Entries configuration becomes corrupted. However, Microsoft mentions that this may be a problem with RegClean.

RegMaid

Like RegClean, RegMaid is a utility that helps Windows 2000 users clean up their registries. RegMaid is much more interactive than RegClean; RegMaid actually has a user interface. The RegMaid program is available from several sources. I suggest that you retrieve it directly from Microsoft’s Internet site at <ftp://ftp.microsoft.com/Softlib/MSLFILES>. Like with RegClean, there may be other versions of RegMaid or other products called RegMaid, but I recommend that you use the Microsoft version found at this address.

The current version of RegMaid, 1.1, was released in 1995. This version is fully compatible with Windows NT 4 and was actually last revised in late 1997. The changes in the revision were slight.



NOTE RegMaid, unlike RegClean, comes with source code. That’s right, you can customize RegMaid to do specific cleanups as desired. To rebuild RegMaid, you will need a copy of Microsoft Visual C++. However, to ensure that the correct directory structure for Visual C++ is maintained, be sure to use either the RegMaid self-extractor or the /d PKUNZIP option.

Is RegMaid Compatible with Windows 2000?

I am not convinced that RegMaid works correctly under Windows 2000.

OK, I’ll be honest; I know that RegMaid has problems with Windows 2000. One problem that I found is that RegMaid doesn’t expand TypeLib entries in REG_EXPAND_SZ format. (It is probable that RegMaid doesn’t expand any REG_EXPAND_SZ objects at all.) On my computer, RegMaid won’t find the file %systemroot%\speech\Xtel.dll (where %systemroot% is set to G:\WinNT), but it will find the file G:\WinNT\speech\Xtel.dll!

Since Windows 2000 stores many TypeLib entries in REG_EXPAND_SZ format, RegMaid fails to find these entries.

If you use RegMaid with Windows 2000, I strongly recommend that you back up the registry and have a second copy of Windows 2000 installed so that you are able to repair any damage to the registry that RegMaid may inflict. Hopefully, Microsoft will introduce a new version of RegMaid after Windows 2000 is released. Check Microsoft’s Web site for more information on the current release status of RegMaid.

Using RegMaid is simple—just follow these steps:

1. Download the RegMaid.exe file from Microsoft’s Internet site at <ftp://ftp.microsoft.com/Softlib/MSLFILES>. (If Microsoft moves the file, you can search for it from any point in Microsoft’s Web site.)
2. Execute the RegMaid.exe file to start the self-extractor program. Alternatively, you may use either WinZip or PK_UNZIP on the RegMaid.exe distribution file to extract the program and other files. If you are using PK_UNZIP, use the /d option to force the creation of subdirectories.



NOTE If you use PK_UNZIP to extract the RegMaid programs without using the /d option, you will receive a message that there are two copies of RegMaid.hlp. Select Overwrite to retrieve the correct help file.

3. Files contained in the RegMaid.exe distribution file include an executable copy of RegMaid.exe, help files, and the program’s source files. There are just under one hundred files contained in the RegMaid distribution package.
4. Execute the RegMaid.exe program that is extracted. When RegMaid is extracted properly, you will be provided with a directory called RegMaid\Release. RegMaid and the necessary support files are located in the Release directory. They may be copied to any location you desire.

RegMaid's primary user interface is the toolbar; like almost all Windows applications, RegMaid has a full function menu, too. The toolbar buttons allow you to quickly navigate through the registry objects that RegMaid has found suspect. You may easily and quickly delete any of these objects.



WARNING Careful. Unlike RegClean, RegMaid doesn't create a recovery file. Once RegMaid removes a registry entry, it will be difficult to restore it. Before running RegMaid, it would be very wise to fully back up the registry. This will also facilitate recovery from any blunders that RegMaid might make.

RegMaid contains four views:

- CLSID view
- ProgId view
- TypeLib view
- Interface view

These views are discussed next. The views are used in order CLSID, ProgId, TypeLib, then Interface. There is a Refresh button on RegMaid's toolbar, and it is recommended that you refresh after deleting objects, before moving to a new view, and after moving to a new view.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**CLSID View**

The first view that RegMaid displays is the CLSID view. This view lists objects, their names, and CLSIDs. The CLSID view looks for CLSIDs (OLE components) that don't have a handler or server or for which the handler or server specified is missing, probably because the file or directory was deleted.

Valid handlers are:

- InprocHandler
- InprocHandler32
- InprocServer
- InprocServer32
- LocalServer
- LocalServer32



NOTE The missing item is shown in the first column in the CLSID view, Missing. This column has six positions, with five dashes and one X. The position where the X is found is the type of the missing handler. For example, if the Missing column shows - - - X - -, this indicates that the InprocServer32 is missing.

Notice that each handler or server comes in two flavors, either 16-bit or 32-bit. Generally, Windows 2000 components will be 32-bit. However, some systems and components do use the 16-bit entries, including some versions of Microsoft Word Basic.

Take a look at Figure 8.4. RegMaid found over 340 items that were not correct in the registry of a relatively stock Windows 2000 installation. Some items were the result of installing aftermarket applications, others come with Windows 2000.

Items listed in Figure 8.4 include:

- Window List in Shell Process, an InprocServer32 object
- History, another InprocServer32 object
- Voice Text Object, a LocalServer32 object
- EnumTelnetClientsSvr Class, another LocalServer32 object

I can tell RegMaid to clean up these entries automatically. To do this, I must select an entry (see Figure 8.5), then click the Delete button in the toolbar or select Clean Up Delete Entries.

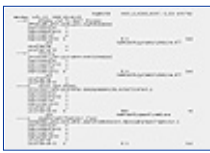


FIGURE 8.4 RegMaid's report for the CLSID view shows some objects that have problems with their handlers.



FIGURE 8.5 RegMaid's CLSID view showing objects that could be fixed

A second, and perhaps better, way is to simply uninstall the problem application. Start the Add/Remove Programs applet in the Control Panel and select the program, application, component, or whatever it is that you want to remove. Do this only if the product is not in use anymore; if the product were still in use, this would not be an option.

Regardless of what I did, after fixing the problem, I'd next click the Refresh button in RegMaid and make sure that no new entries show up in the CLSID view. If nothing new shows up, I'd go on to the next view: ProgId. If any new entries do appear, I'd follow this process a second time.

Once you are happy with the items in CLSID view, move on to ProgId view.

ProgId View

The ProgId view contains items that are associated with the registry's ProgID entries. Entries in ProgId view show a name, a CLSID, and a ProgID name (see Figure 8.6). As with the CLSID view, it is imperative to determine exactly what each entry listed is for and why there is an error. Unlike CLSID problems, the ProgID entries are not simply a matter of a missing file—in this case, we are dealing with registry entries that are corrupt or, more likely, missing.

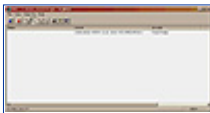


FIGURE 8.6 RegMaid's ProgId view shows those entries with invalid ProgID entries.

Generally, it is safe to remove these entries. As with any other registry change, back up the registry first.



NOTE Right from the start, some Windows 2000 systems (those upgraded from Windows NT 4, for example) have several invalid ProgID entries. The Scheduler Queue Object and Scheduler Job Object entries are found in all Windows 2000 systems. No documentation exists with regard to their use or necessity, other than that they are used to process .job or .que file types. No actions are specified for either.



NOTE A clean installation of Windows 2000 typically has one invalid ProgID entry, called TimeStamp. There is no documentation on this entry.

TypeLib View

RegMaid will search all entries in the HKEY_CLASSES_ROOT\TypeLib section of the registry to determine if there is an associated .tlb (TypeLib) file. If the file cannot be found based on the entry, RegMaid will report that entry.

Here, I have to disagree with RegMaid's documentation (regarding whether to delete the entry or not). My recommendation is to do the following:

- For any TypeLib entry with an entry in one (or more) of the file columns, search for the file on the hard drive. If the file is found, but at a different location from where the registry entry says it should be, you may consider updating the registry manually to show the correct path name (see Figure 8.7). I've found that about half of the entries flagged as being bad in the TypeLib view are marked this way because the path to the file was incorrect.

- For any TypeLib entry that lists two or more versions, it is possible that one version has improper entries, while the other version may be OK. Typically, when a new version of a product is installed, the older version may not be completely removed from the registry. In this situation, I'd recommend leaving these entries in the registry without change, or deleting the version that has incomplete values. RegClean actually does a proper job of cleaning up this type of registry chaff.



WARNING Generally, my recommendation is to err on the cautious side. If in doubt, don't use RegMaid to delete the entry. RegClean does a much better job of cleaning and repairing the TypeLib entries than RegMaid does.



FIGURE 8.7 This TypeLib entry originally had a data value of %systemroot%\speech\Xtel.dll. Changing the value to g:\winnt\speech\Xtel.dll solved the problem that RegMaid found.

Interface View

The Interface view searches the HKEY_CLASSES_ROOT\Interface entries. Each entry that has a TypeLib subkey is checked to determine that there is a match between the TypeLib entry's CLSID and a valid OLE object found in the HKEY_CLASSES_ROOT\Typelib subkey. If no entry is found in HKEY_CLASSES_ROOT\Typelib, RegMaid will flag the line.

RegMaid claims that entries that don't match may be safely deleted. However, I recommend that you don't allow RegMaid to fix this error—RegMaid will delete the entire subkey in HKEY_CLASSES_ROOT\Interface instead of deleting the suspect TypeLib entry. RegClean also does not flag this discrepancy as an error.



WARNING If you wish to invoke RegMaid's delete on an Interface view item, back up the registry or subkey in question before continuing. Blind deleting like this will probably lead to disaster, sooner or later.

Recommendations for RegMaid

There are several recommendations to follow when using RegMaid:

- Make a full backup of the registry before starting RegMaid.
- Be careful about what is removed with RegMaid. RegMaid does not have any methodology to recover from errors, either its own or yours.
- The CLSID view entries may be safe to delete, but do review each of them first.
- The ProgId view entries are probably safe to delete, although you should review each of them first as well.
- The TypeLib and Interface view entries probably should not be deleted unless you are absolutely sure that these entries are not being used.
- Run RegClean before running RegMaid. RegClean will clean many problem entries that RegMaid would find. RegClean will create a .reg file that allows restoring these entries if desired, so there is an additional recovery path that RegMaid doesn't offer.
- Consider rewriting RegMaid to write a recovery file for each item deleted. Since Microsoft supplies the source code file for RegMaid, a recovery file would not be difficult to create (if you are a C/C++ programmer).

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**WARNING** Due to the lack of any restore methodology, always use RegMaid with caution!

CleanReg

Matt Pietrek, a columnist for *Microsoft Systems Journal* and a developer at Compuware's NuMega Labs, created a program called CleanReg for an article he wrote for *MSJ*. This very clever utility may be obtained from several sources. If you have access to a subscription to MSDN, the source for CleanReg is available on the MSDN CD-ROM. Or, if you have a subscription to *MSJ* and still have the September 1996 issue, the source is located on page 77. The source code is also available at <http://www.microsoft.com/msj/codetop.htm>.



NOTE Checking out Microsoft's entire *MSJ* Web site at <http://www.microsoft.com/msj/> will reveal lots of good information, especially for programmers. For example, all source code from back issues is available by clicking on the Back Issues link and scrolling to the desired issue.

CleanReg works a bit differently from RegClean and RegMaid. CleanReg looks at registry entries and attempts to find filenames. Whenever CleanReg finds what it thinks is a filename, Nit searches for the file.

Pietrek had to overcome several difficulties when he wrote CleanReg. For one thing, he had to determine what constitutes a valid filename. With long filenames, Pietrek correctly states that the following is actually a valid filename (try it, I did):

```
foo -p .exe
```

So You Say CleanReg Won't Compile Right?

There is a problem with CleanReg and some later versions of Microsoft Visual C++: Microsoft Visual C++ will indicate an error in `clnregui.cpp` with the `WinMain` function. The error indicates that `WinMain` has been either redefined or overloaded. The error is in the types assigned to the parameters of the `WinMain` function. To correct this problem, change the `WinMain` parameter list to what I have shown here. Simply add all characters and lines shown in bold in this listing fragment to your version of `clnregui.cpp` (and don't forget the comment characters, //):

```
TEXT("CD-ROM) for all documentation questions.");
// int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
//                    PSTR lpszCmdLine, int nCmdShow )
// Function parameters cleaned 6/8/98 by Peter D. Hipson
int PASCAL WinMain(
    HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPSTR lpszCmdLine,
    int nCmdShow )
{
    InitCommonControls(); // Gotta do this for treeview controls
```

Fix the `WinMain` function before correcting any errors, such as an error calling the `DialogBox()` function a bit later in the `WinMain` function, because these other errors are caused by the incorrect `WinMain` parameters.

I've successfully compiled CleanReg with Microsoft Visual Studio 6.0 after fixing the `WinMain` line.

But, in the registry, what's to differentiate the filename `foo -p .exe` from the executable `foo` taking the parameter, `-p .exe`? Is there a standard in the registry? No, not really. Is there a standard anywhere else? Yes, somewhat. For a command passed to the operating system, it is expected that the executable filename will be enclosed in double quotes ("") if it is not a short (8.3) name. That is, if you have the file `foo -p .exe`, and you want to execute this file, you must enter the command exactly as:

```
"foo -p .exe"
```

There will be an error if you enter the name like this, without quotes:

```
foo -p .exe
```

In this case, the operating system will assume that the name of the executable file is `foo`, and will then attempt to pass the parameter(s) `-p .exe` to that file.



NOTE Sometimes Windows 2000 is able to correctly determine the filename even if it is a long name. In these cases, Windows 2000 usually is able to figure it out if the name doesn't contain any spaces or other special characters.

Entries in the registry don't have set, fixed rules. Programmers of applications have been known to code exactly what they expect and not to bother considering any other application or system convention. Now, some programmers have adopted a convention that Microsoft uses for many registry entries. It involves filenames and parameters—don't quote the filename, but instead, quote the parameters, if there are any, such as in this example:

```
foo "-p .exe"
```

This works well, but unless you know this rule is being followed, it is difficult to determine whether the programmer is following this rule or simply being lazy about including quotes in the following string:

```
foo -p .exe
```

This leads us right back to the original problem: what constitutes a valid filename in a registry entry and what does not? In the end, you, the user, will have to determine whether a filename is valid when running CleanReg. Some simple tricks of the trade will be helpful. When given a path, take the name up to the first

non-alpha character, append *.* to that name, and try to find the file with the **dir** command in a command window. (An alpha character is a letter, a number, or one of the allowed special characters.)

For example, when searching for

```
C:\temp\foo -p .exe
```

take the first part, up to the first invalid character (the space)

```
C:\temp\foo
```

append *.* to this name

```
C:\temp\foo*.*
```

and do a dir command

```
dir C:\temp\foo*.*
```

The dir command will list all files beginning with foo, allowing you to determine if the file in question is foo, foo.exe, foo -p.exe, or whatever. Then you may make an educated guess with CleanReg as to whether to remove the file's entry in the registry or not, depending on the search.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

CleanReg allows you to remove either a single value or a key. When removing a key, CleanReg will delete all subkeys contained in the subject key. This should raise a note of caution with you—be careful not to delete too much when using CleanReg.

Figure 8.8 shows CleanReg running on a Windows 2000 system. There were probably almost two hundred entries that CleanReg found that were suspect. A manual check showed that about 80 percent of these entries really were bad.

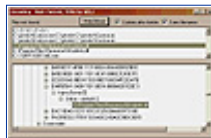


FIGURE 8.8 CleanReg listing a bad reference to the file C:\Program Files\Resource Kit\sidwkr.dll

In Figure 8.8, the file that CleanReg thought was missing (C:\Program Files\Resource Kit\sidwkr.dll) was actually found on the G: drive. It seems that the Resource Kit installation program “made a mistake” while updating the registry and that the installation would always be to the C: drive. Bad assumption!

CleanReg has two check boxes in the user interface:

Update after Delete This option tells CleanReg to update the display after the user deletes anything. Checking this option could slow things down a bit when the user’s registry is large, so the use of this option is up to the user, based on experience.

Sane Filenames A “filename” means that there’s a \ near the beginning of the string that CleanReg is checking. If Sane Filenames is checked, CleanReg assumes that characters like / and - aren’t part of the filename, although they’re technically legal.

Some entries to suspect and delete are those that point to the temp directory. These files are usually artifacts of checking out a file and having the file appear in a program’s MRU (most recently used) list. Or perhaps a program was temporarily installed into the temp directory to be checked out. My temp directory is C:\temp. My rule is that anything in the temp directory may be deleted at any time. Nothing to be saved should ever be placed in the temp directory.

Many applications store their MRU in the registry in a subkey called Recent File List. Removing entries from such a subkey manually usually results in few bad side effects. However, using the application to clear the MRU list is the best alternative, if possible; some applications don’t have a mechanism to clear the MRU list.

Some applications save work or other files in the temp directory, too. Generally, these applications are robust enough that they will not fail should these files be deleted. Any critical work file will typically be kept open

by the application just so the user is unable to delete the file.



NOTE CleanReg doesn't see hidden files or directories. Be careful that you don't mistake a file that is hidden—one that has the hidden attribute—with a file that is truly missing. In a command-prompt window, you can determine a file's attributes with the **attrib** command. In Windows, use Explorer's options to turn on the display of files with the hidden attribute.

When Matt Pietrek wrote CleanReg, he wisely decided not to check for files on floppy drives or other drives with removable media, such as CD-ROM drives. CleanReg does check for files on currently accessible network drives. However, be careful of the case where a CD-ROM is accessed over a network.

To use or make changes to Pietrek's program, download the original source from Microsoft and compile it, or see if you can find an executable version of CleanReg on the Internet.

Hints and Kinks from the Experts

Here is a registry cleanup hint from our experts.

How Can I Clean Up/Remove Invalid Entries from the Registry?

Microsoft has released a utility called RegClean that will go through your machine's registry and delete any unused/unnecessary keys. The current version is 4.1a and can be downloaded from <http://support.microsoft.com/download/support/mslfiles/RegClean.exe>.

Once downloaded, just click on the executable, and it will check your registry. After the check is complete, you will be given an option to fix errors. You may do so by clicking the Fix Errors button. To exit, click the Exit button.

RegClean creates an uninstall file in the directory where RegClean is located. RegClean will use the name Undo <machine name> <yyyymmdd> <hhmmss> .reg—for example, Undo workstation 19980320 104323.reg. To undo the changes, just double-click this file.

See <http://support.microsoft.com/support/kb/articles/q147/7/69.asp> for more information.

(Courtesy of John Savill.)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!
CHAPTER 9**Recovering from Disaster, or Making the Best of a Bad Situation****FEATURING:**

<i>To repair or to replace</i>	203
<i>Stabilizing the system</i>	204
<i>Analysis</i>	219
<i>Fixing things up</i>	220
<i>Possible problems, quick fixes</i>	221
<i>Manually removing registry entries</i>	224
<i>Hints and kinks from the experts</i>	226

Disaster usually strikes when least expected. There it is, usually late at night, just when things are sailing smoothly along, and whammo! A server fails, maybe with an infamous “blue screen of death.” (When a system error occurs in Windows that is so severe that the operating system cannot continue, it displays a blue text mode screen with white characters text mode telling about the error.)

Disaster can also strike when Windows 2000 boots. Maybe the system starts fine, then mysteriously crashes in any of a thousand ways after a few minutes, a few hours, or even a few days. On the other hand, maybe, just maybe, out of the blue something happens and the system becomes unstable.

Sometimes we have to figure out what happened. Maybe a hardware problem precipitated the failure. Or perhaps a failing hard drive with a bad sector in an infrequently used section of a system file caused the problem. Maybe, and this one is nasty, there’s a bad spot in the registry. When disaster strikes, a methodical approach to recovery is the only reasonable path to follow. You can try the shotgun technique: replace things randomly until something fixes the problem. Or you can use a more logical technique: analyze the problem and apply fixes in a systematic method. I vote for the latter; I’ve tried shotgun type repairs, and they are so

difficult to do that in the end, only the most inexperienced user will try to fix a problem using such a random technique.



WARNING Have I already said this? *Back up your registry* before doing anything described in this chapter. Manually removing items from a registry is perhaps the *easiest way to trash everything!* Back up, back up, and back up again.



NOTE Remember to check the Event Viewer. The event log can contain valuable information about failures of both applications and system components!

If I haven't mentioned it, read Chapter 8, "Getting Rid of the Unwanted," as well. Sometimes the unwanted is the root cause of all of our problems.

What Fails Most Often?

The things that cause serious problems with Windows 2000 registries and installations are:

- Removing software without using the software's uninstall program. If you don't use the uninstall program, entries are left in the registry that point to files that are no longer there.
- Improperly or incompletely installed software. Again, the problem is often due to insufficient disk space for the software's files, or an installation program that fails. In this case, the installation program probably updated the registry before the file copy process had completed; it then failed to undo the registry update after the file copy process failed.
- Damaged software files, caused by installing either the wrong software or wrong software version into a directory where an existing software program or version exists. Some problems arise when software versions become mixed, or when the installation process cannot properly update one or more files.

To Repair or to Replace?

The Windows 2000 installation program allows you to repair a broken installation. This is nice. This is good. This can be dangerous, too.

Generally, the repair options in the Windows 2000 installation program simply allow you to replace the Windows 2000 system files with fresh copies from the distribution media. These system files are the same files installed at the original installation. Now how can that be bad? Well, if you have installed a service pack, such as Service Pack 4, you may find that when you refresh the Windows 2000 installation, part of your service pack goes away. However, some things, like registry entries, won't go anywhere. This can result in some rather strange problems, to say the least. Sometimes it becomes a catch-22 situation. A service pack is installed, so you can't refresh the Windows 2000 installation; but you can't remove the service pack (most service packs have an uninstall process) because the system won't run; and if you force a refresh of the system files, the system won't run to allow you to install the service pack. Oops, you're stuck, again.



NOTE *Catch-22* is the title of a popular book by Joseph Heller. This term describes a situation in which two actions are mutually dependent and cannot be done separately. However, they can't be done at the same time, either. Like how you can't reinstall the Windows 2000 system files and the service pack at the same time, although to run Windows 2000, you might need the service pack.

What do you do? Try refreshing the Windows 2000 installation using the repair options in the installation program. Immediately after that, install the same or a higher-level service pack as was installed originally on the system. That should refresh the Windows 2000 installation and the service pack installation. Of course, if you refresh the installation and the system won't run afterwards, you do have a problem; it may be time to reinstall Windows 2000 from scratch.

[Previous](#) [Table of Contents](#) [Next](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**

Click to access!



NOTE At least one supplier of Windows backup and restore software noted the following scenario and problem: Let's say you have a system for which the system drive has totally failed. You replace the drive and install a minimum copy of Windows 2000 to run the restore program to recover the original disk's contents from backup. The original system included a service pack. You will probably find that you can't complete the restore. The problem is that the minimum copy of Windows 2000 must have the same service pack installed as the original copy of Windows 2000; otherwise, you are restoring mismatched files into the system directories.

Stabilizing the System

Once a disaster has occurred, the first step is to stabilize the system. It is important that you prevent further problems or damage. After stabilizing the system, it will be much easier to fix the problem and get everything performing at its best.

Consider stabilization a systematic analysis. Start with the first step, discussed next. Can you do what this step calls for? If not, go to step 2. If so, does the system work right? If not, go through step 1 and see if any of the hints and suggestions might apply to your system. I can't list every possible problem or fix, but I'll try to cover the most common ones here in this chapter.

When this chapter doesn't help, consider Microsoft's Internet news server at msnews.microsoft.com. This server is accessible using one of the Microsoft news programs, such as Outlook Express or Microsoft New, or an Internet news program, such as Agent from Forte or any of the other Internet news programs available to users.

A few of the newsgroups to check on msnews.microsoft.com include:

- [microsoft.public.windowsnt.misc](#)
- [microsoft.public.windowsnt.apps](#)
- [microsoft.public.windowsnt.setup](#)

Posting a query in one of these newsgroups will certainly create some response. Whether the respondents are able to assist you is something that you won't know until you try. I've posted a number of questions over the years; I've gotten help about half the time, and usually when I did not receive a useful reply, I did get the feeling that people on the newsgroup had at least tried to assist with a solution.



WARNING Be careful not to lose your Last Known Good configuration. When Windows 2000 boots successfully, it will overwrite the Last Known Good configuration with the current configuration. This could cause great gnashing of teeth later on. Try very hard to back up the registry and the operating system if possible (discussed next).

Step 1: Can You Boot into a Spare Operating System?

Can you boot the system into a different operating system or a different copy of Windows 2000? If not, go to step 2.

By booting into a different operating system or a different copy of Windows 2000, you will possibly be able to preserve (back up) the existing registry and hard drives and even do tests on the system's hardware. Once you've booted and are running, *back up immediately*.



WARNING When backing up, do not back up to existing backup tapes. Use new tapes so that you do not overwrite any existing backups. There is a very high probability that you will be making a backup of information that is not good, while any existing backup (especially older backups) may have valid copies that you will have to restore later. If necessary, go out and buy a new set of backup tapes.

Once the computer boots another copy of the operating system, do the following:

1. Back up the registry files using the techniques described in Chapter 2. Copy the files from the copy of Windows 2000 that failed. You will find this copy of the registry in the Windows 2000 installation directory, C:\winnt\system32\config, of the failed Windows installation. Any process used to back up this directory and its files will be useful. Copy the directory to removable media, such as a Zip drive or a network drive. Using diskettes is a possibility, although the size of many registry files (a total of 30 or more megabytes) will necessitate the use of many diskettes.
2. Back up the entire system. Use the booted operating system's backup program to create a copy of the system exactly as it was when it failed. Don't delete anything, don't rename anything, and don't change anything. Get a backup—just in case you are wrong about the problem and need to restore everything to the state that it was in when it failed. More than one time, I've hacked about on a failing system only to realize after I've done considerable damage and that the problem was somewhere else. When this happens, it is nice to be able to restore the drive to undo your own self-induced damage.
3. Back up any drives used to hold components and applications. This generally will mean doing a complete backup of all of the system's hard drives.
4. Run diagnostic software on the computer. Check the drives (fully, including a surface scan if possible), memory, and CPU before going any further. Sometimes a system will boot another operating system even when there is a hardware failure—perhaps the other operating system doesn't have any critical components in the area of memory that is bad. (Windows 2000 pushes the hardware very hard, while Windows 95/98 is less demanding on the system and memory.) If you suspect bad memory, many computers will allow you to set, in the BIOS, the maximum amount of memory allowed. However, if the bad memory is in the first few MB, it is unlikely that there will be enough memory to boot the system. In this case, swapping the bad RAM with good units can help diagnose the problem.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)


NOTE Diagnostic software? Where does one get diagnostic software? There are several good commercial test programs, such as Q&A Plus, used to test computer hardware. These programs allow the user to determine if the system is performing correctly. Be careful with any diagnostic software, especially when checking storage media. Some diagnostic program functions may be destructive to data on drives. Be sure to follow all program instructions carefully and heed all warnings.

JUMP TO TOPIC


Desktop Library
[Click to access!](#)
Step 2: Can You Boot the System in Normal Mode?

If you can boot the system in its normal mode, go to step 3. Otherwise, read on.

Windows 2000 has a different boot-up manager than Windows NT 4. Differences include only one selection for the initial boot (the option to boot to a VGA mode has been moved) for each installed copy of Windows 2000. Figure 9.1 shows the Windows 2000 boot menu. Compare this menu with the Windows NT 4 boot menu (shown in Figure 9.2), and note the subtle changes made.



FIGURE 9.1 The Windows 2000 boot menu allows you to boot in one mode only. Booting problems require pressing F8 to get to the debugging screen.



FIGURE 9.2 The NT boot menu allows you to boot in VGA mode.

In Figure 9.1, you can see the two Windows 2000 installations on the system. Each installation is on a different drive so that drive failure won't prevent booting the system.



TIP In each menu where there is an automatic selection, you will see the message "Seconds until highlighted choice will be started automatically" and a countdown timer. The countdown timer will stop whenever an arrow key (either up or down) is pressed. Even if there is only one selection, pressing an arrow key will still stop the timer, giving you time to read the menu's text.

If you have a problem and you can boot to the boot menu, you can press F8 and set debugging modes as appropriate (see Figure 9.3).



FIGURE 9.3 The Windows 2000 Advanced Options Menu allows you to choose how Windows 2000 will boot.

The Advanced Options Menu has eight choices:

- **Safe Mode:** This mode starts the system with a minimal set of files and drivers. Drivers loaded include only mouse, monitor, keyboard, mass storage, base video, and default system services. There is no network support in safe mode.
- **Safe Mode with Networking:** This mode adds network support to standard safe mode. This is useful when debugging tools reside on a network drive.
- **Safe Mode with Command Prompt:** With the command-prompt mode, the same configuration is loaded as with safe mode, but instead of starting the GUI, Windows displays a command prompt. Users who are familiar with the command prompt may find this mode more stable and easier to use.
- **Enable Boot Logging:** Using boot logging allows you to determine which drivers and other objects are loaded when Windows 2000 boots. Listing 9.1 shows part of a typical boot log. The boot log is stored in the `%systemroot%` directory, in the file `ntbtlog.txt`. Use Notepad to edit or display this file, which typically has several hundred entries.
- **Enable VGA Mode:** This starts Windows 2000 using the default VGA driver, in 640×480 mode. This driver is compatible with all display adapters supported by Windows 2000. The default VGA driver is not an optimal driver. It lacks support for higher resolutions, higher color depth, and any high-performance features of the display adapter. However, the default driver will usually work regardless of the hardware installed.
- **Last Known Good Configuration:** This starts Windows using the Last Known Good configuration. `HKEY_LOCAL_MACHINE\System\Select\LastKnownGood` is a pointer to the Last Known Good configuration. This value contains an index to one of the `ControlSet n m` subkeys. Use the Last Known Good configuration when a bad configuration change (such as improperly adding new hardware) happens. The Last Known Good configuration will not help when system configuration files are missing or damaged.
- **Directory Services Restore Mode (Windows 2000 domain controllers only):** Use this option to restore the Active Directory. The Directory Services Restore Mode is usable on a domain controller, not on Windows 2000 Professional (workstation) or member servers.
- **Debugging Mode:** This mode sends status messages to the default communications port, which is COM1. Connect a terminal or other serial device to the communications port, and configure the device correctly.

Listing 9.1: Excerpts from a Typical ntbtlog.txt Boot Log File

```
Microsoft (R) Windows 2000 (R) Version 5.0 (Build 2072)
 7 20 1999 19:23:08.500
Loaded driver \WINNT\System32\ntoskrnl.exe
Loaded driver \WINNT\System32\hal.dll
Loaded driver \WINNT\System32\BOOTVID.DLL
Loaded driver pci.sys
Loaded driver isapnp.sys
Loaded driver intelide.sys
Loaded driver \WINNT\System32\DRIVERS\PCIINDEX.SYS
Loaded driver MountMgr.sys
Loaded driver ftdisk.sys
Loaded driver Diskperf.sys
Loaded driver \WINNT\System32\Drivers\WMILIB.SYS
Loaded driver dmload.sys
Loaded driver dmio.sys
Loaded driver PartMgr.sys
Loaded driver atapi.sys
Loaded driver aic78xx.sys
```

```
Loaded driver \WINNT\System32\DRIVERS\SCSIPTORT.SYS
Loaded driver disk.sys
Loaded driver \WINNT\System32\DRIVERS\CLASSPNP.SYS
Loaded driver Dfs.sys
Loaded driver KSecDD.sys
Loaded driver Ntfs.sys
Loaded driver NDIS.sys
Loaded driver Mup.sys
Loaded driver \SystemRoot\system32\drivers\ctlsb16.sys
Loaded driver \SystemRoot\System32\DRIVERS\gameenum.sys
Loaded driver \SystemRoot\System32\Drivers\NDProxy.SYS
Loaded driver \SystemRoot\System32\Drivers\EFS.SYS
Did not load driver \SystemRoot\System32\Drivers\NDProxy.SYS
Loaded driver \SystemRoot\System32\DRIVERS\usbhub.sys
Loaded driver \SystemRoot\System32\DRIVERS\flpydisk.sys
Did not load driver \SystemRoot\System32\Drivers\Sfloppy.SYS
Loaded driver \SystemRoot\System32\DRIVERS\ipsec.sys
Loaded driver \??\G:\WINNT\System32\Drivers\syment.sys
Loaded driver
    \??\G:\PROGRA~1\COMMON~1\SYMANT~1\VIRUSD~1\19990712.001\NAVENG.Sys
Loaded driver
    \??\G:\PROGRA~1\COMMON~1\SYMANT~1\VIRUSD~1\19990712.001\NavEx15.Sys
Loaded driver \??\G:\WINNT\System32\Drivers\navap.sys
Loaded driver \SystemRoot\system32\drivers\kmixer.sys
```

Safe Mode

When you experience problems with your Windows 2000 system, try booting in the safe mode first. This is the default, and often the most useful, debugging mode. Safe mode is much like Windows 95/98's safe mode, in that only a minimum system is loaded.

There are three safe modes in Windows 2000. First, there's safe mode with no networking (the default). This mode loads only the basic files and drivers: the base video (VGA for most systems), basic mouse, monitor, services, and storage.

The next level of safe mode is safe mode with networking. In this mode, drivers and files loaded are still the same basic ones loaded with safe mode without networking. However, Windows 2000 attempts to load the networking support as well. Using safe mode with networking allows you to connect to other computers if necessary.

Safe mode with command prompt does not start the Desktop, Start Menu, or Taskbar. Instead, you're presented with a command prompt to work from. This final mode is similar to the Recovery Console described in Chapter 2.

Try safe mode without networking first; if that works, and you need networking, try safe mode with networking. If safe mode without networking fails, try safe mode with command prompt. If safe mode with command prompt fails, then it will be necessary to fall back to the Recovery Console.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

Using the Last Known Good Configuration

To use the Last Known Good Configuration menu, choose it from the Advanced Options Menu shown in Figure 9.3 and press Enter. The system will continue the boot by displaying the initial boot menu (with the bottom line indicating the selected boot option). Use the arrow keys and press Enter to boot the desired version of the operating system.

Once the system starts, the boot process displays a menu called the Hardware Profile/Configuration Recovery Menu (shown in Figure 9.4). A Windows 2000 installation can have multiple hardware configurations. (The most common applications for multiple configurations are a notebook computer, a computer with PCMCIA, or PC, cards, or an active USB or IR bus configuration.) Those of us with standard PC configurations, without easily removable hardware, will have only a single hardware profile (by default named Profile 1), while anyone with removable hardware should have a profile for each configuration that may be used.



FIGURE 9.4 Use the Hardware Profile/Configuration Recovery Menu to select the hardware profile and Last Known Good configuration.

If it is necessary to change to the default configuration, press D, which turns off the Last Known Good selection. You can reenable the Last Known Good selection by pressing the L key.

For systems with multiple hardware configurations, select the boot configuration from the list. For a system with one default configuration, the configuration name is Profile 1, and it is automatically selected for you.

The Profile 1 entry, by the way, comes from the System applet in the Control Panel. In that applet, there is a tab for configuring hardware, and on that tab is a button to display the dialog box for configuring hardware profiles. In this dialog box (see Figure 9.5), you can also set the time delay before taking the default selection. Though many users will have only one hardware configuration, anyone using Windows 2000 on a dockable portable platform will certainly have at least two profiles, one for when docked, one for when not.

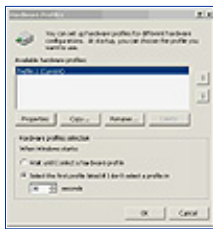


FIGURE 9.5 Configure hardware profiles in the Hardware Profiles dialog box.

After selecting the Last Known Good configuration (and hardware profile, if necessary), you still have to press Enter to continue the boot process. The boot process for Windows waits indefinitely until Enter is pressed.



WARNING Remember, once the Last Known Good configuration is booted, it becomes the current configuration (the current control set), and the current configuration that would have been booted will be discarded. Anything installed after the previous boot will be lost.

Control Sets, Control Sets, and More Control Sets

After booting using the Last Known Good configuration option, your registry will “grow” a new control set. This control set will be numbered one higher than the currently known highest control set. For example, if your system has ControlSet001 and ControlSet002, a new control set called ControlSet003 will also be created. In this situation, one control set is the one that failed, one is the current control set, and one is the Last Known Good configuration. After booting my system, the Last Known Good configuration had the following control sets:

ControlSet001 Was marked as the control set that failed. This control set would have been booted if the Last Known Good configuration had not been chosen.

ControlSet002 Was marked as the Last Known Good control set. This control set will be booted if the Last Known Good configuration is selected at the next boot.

ControlSet003 Was marked as the current control set—the control set used to boot the system. Prior to booting, this control set was marked as Last Known Good.

If you manage to boot the Last Known Good configuration, consider yourself lucky; the system should be stable, although it probably will be missing whatever software and hardware installed during the last session. However, this should be only a minor problem. In this case, consider everything installed during the last session. Think very carefully as to whether it makes sense to reinstall the same item a second time. Consider set



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
The System Boots, Then Crashes Almost Immediately

This situation is virtually as bad as a system that won't boot. Possibly the cause of the crash is something that is starting up when the system starts. Try this: Start the system, but don't log on. Just sit and watch for at least twice as long as it normally takes to crash. Does it crash? If it does, this is probably due to some system component. You are probably stuck with little or no hope except to reinstall, or to restore from a backup.

If the system doesn't crash immediately, the crash is probably due to something that the user is loading. Log on as another user. Does it crash? If it does, the problem is probably something that is common to all users. Check out the common Startup directory (%systemroot%\Profiles\All Users\Start Menu\Programs\Startup or %systemdrive%\Documents and Settings\All Users\Start Menu\Programs\Startup), and clean it out. Try starting the system again. If it fails again, you are probably stuck with either a restore or a reinstallation.

If the system only crashes when you log on as a particular user, you may be saved yet. Check the failing user's Programs \ Startup directory in the Start menu. Check all Programs\Startup directories for that matter, cleaning out each one; put anything contained in the Startup directories into temporary directories. Once you have cleaned out the Startup directories, log on again as the user who causes the system to fail.

If the system doesn't fail once you've cleaned out a user's Startup directory or the Startup directory for all users (you're almost home free now), check the entries that were in the Startup directory. Consider manually starting each one, then wait for a reasonable period to see if the system fails or not. This will almost certainly help localize the problem to a single entry in the Startup directory.

How do you get to the Startup directory if the system keeps failing? Again, you can rely on your old friend, the dual boot. (You did create a dual-boot system as I described in Chapter 2, right?) Boot the backup operating system and use it to allow you to clear out the Startup directories. Just make sure you are deleting the *correct* Startup directories.

Some additional locations that items may run from include:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run



JUMP TO TOPIC

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
- HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
- HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load

The Crash Comes Sometime Later

How much later? Minutes, hours, or days? A crash that comes many hours or days later is probably not related to the registry. A crash that happens a few minutes later is almost identical to the above situation where the crash is virtually immediate. Nevertheless, a crash that happens some minutes or even an hour later could easily be a registry entry gone awry. How does this happen? When Windows 2000 starts it will start up many services and devices. Some services are slow to start and other services start but then spend some time initializing.

Try this: in either `%systemroot%\Profiles\All Users\Start Menu\Programs\Startup` or `%systemdrive%\Documents and Settings\All Users\Start Menu\Programs\Startup` (depending on whether the current installation was an upgrade or a clean installation), put in a link to `taskmgr.exe`. This will launch the Windows 2000 Task Manager application. Look at what Task Manager is saying. Sort the entries in the Processes tab by CPU usage. Do you see an application that is jumping up in CPU utilization just before the system crashes? This may be the problem.

OK, let's say you have a suspect. Task Manager shows a big chunk of CPU utilized by a particular application. Let's call this application `badapp.exe`. (Great name, isn't it?) What do you do? First, it would be nice to simply tell Windows 2000 not to load or execute `badapp.exe`. However, it may be virtually impossible to do that, since this application may be launched by a registry entry. Remember, there are six places in the registry that function much like the various `Programs\Startup` directories.

What is the next best thing? If you have nothing to lose, consider temporarily renaming the file. Boot into your backup operating system and use it to rename the file, giving it a new temporary filename. I would add the prefix `bad_` to the original filename, making it easy to find later. Just make sure you are renaming the *correct* file.

After renaming the file, restart the original or backup Windows 2000 installation. You should expect to see at least one message informing you that the file you renamed can't be found, and you can *probably* ignore this. Probably, but not always. If the file is a necessary part of the operating system, Windows 2000 probably won't start. Arrgggg! Such is life; in this case, an operating system restoration or repair is the only solution.

Can You Cause the System to Crash?

Sometimes the system will remain stable until you do a specific thing. In this case, there are two possible courses of action. If the application worked at one time and just recently started to fail, something has happened either to the application's files or to the application's registry entries. In either case, a good course of action is to simply try reinstalling the application.

If possible, try removing the application before doing the reinstallation; be sure to back up any user data files first, though. Sometimes installation programs don't write over certain files that already exist.

If the application never worked on your system, again, there is but one alternative: uninstall the application, posthaste. Sadly, some applications are poorly written and don't have an uninstall program. With due caution (make backups), rename the application's directory to something you'll notice, so that in a week or so, if the system remains stable, you can delete the directory.

After renaming the application's directory, restart the system and see if there is any instability. If things are stable after a day or two, use a registry cleanup tool such as RegClean, CleanReg, or RegMaid to extract any registry entries for this application. My choice would be to use CleanReg (see Chapter 8), because CleanReg will check more than just the OLE entries.

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

The Crash Happens after a Specific Time

If the system always seems to crash after a specific time, check to make sure that there are no time-based applications or commands that run. (The Windows 2000 AT command is a suspect here.) What other things happen at the time? Is the time absolute or relative to boot? If absolute, suspect that something is being started at the specific time or shortly before. If relative, look for something that is being started with the system boot up, but maybe taking a very long time to initialize because it fails. Note that some systems are timing interdependent, which means that process A must start after process B. Again, beware of any catch-22 situations where two processes are mutually dependent.

Step 4: Do the System Components and Subsystems Run OK?

If you find that your system will run indefinitely without failing, you may have good reason to suspect that an installed application is the problem. You randomly run applications and eventually something fails.

At this point, you say *voila*, "I've found the problem." Alas, it is not that simple. You may find the problem's trigger, but the odds are high that the problem itself is somewhere else.

Narrow down interdependencies between applications by running only one at a time. Start Windows 2000, then start and use one application. (This works well for most applications, but when you have two applications designed to work together, this may not be a viable way to troubleshoot the problem.)

Review your list of recently installed applications. Anything installed just before the system became unstable should be suspect. If an application has never worked on your system, again there is but one alternative: uninstall the application if possible.



NOTE A possible test is to create a second, clean installation of Windows 2000 and install the suspect application under the second copy of the OS. That will be a good indicator as to whether the application will be able to run under Windows 2000 without problems. Using a clean installation of Windows 2000 will help minimize unwanted interaction between two applications.

If your application doesn't have an uninstall program, make a backup and rename the application's directory. Rename the directory to something you'll notice, so that in a week or so, if the system remains stable, you can delete the directory.

After renaming the application's directory, restart the system and see if there is any instability. If things are stable after a day or two, use a registry cleanup tool such as RegClean, CleanReg, or RegMaid to extract any registry entries for this application. My choice would be to use CleanReg (see Chapter 8), because CleanReg

will check more than just the OLE entries.

Step 5: Do Installed Applications Run OK?

If installed applications run OK, go to step 6. Otherwise, read on.

What is happening? Probably something has corrupted the registry or there is a hardware problem. First, back up the system fully. Then, run sufficient diagnostics to rule out any hardware problems. Finally, try restoring the registry. Start with the most recent backup—not the one you made before running diagnostics, but the most recent regularly scheduled backup. If the most recent backup doesn't solve the problem, continue working back through older backups to see if one of them will restore system stability.

Be aware that by going back through older backups, you only want to restore system files and the registry—for example, you do not want to restore user files.

Step 6: Is the System Generally Stable?

If the system is generally stable, go to Step 7. Otherwise, read on.

If a system is unstable, and the instability cannot be traced to a specific application or component, this usually points to a hardware problem. In this situation, analysis of the failures is important. These steps may help in diagnosing and fixing the problem:

- Run all possible hardware checks and diagnostics.
- Swap out whatever hardware parts may be replaced easily.
- Install and run a second copy of Windows 2000 with all the software and components that the failing system uses.
- Reinstall (repair) the failing installation of Windows 2000.
- Reinstall the applications and optional components.

Step 7: Then What Is the Problem?

What is the problem, then, if the system will start, run, and shut down OK, and it doesn't crash or otherwise fail? There can be serious problems even when a system doesn't crash.

Take the situation in which the computer's hardware is simply overwhelmed by the demands that the operating system and applications place on it. Running some applications—for example, server components such as SMS, SQL Server, and Exchange Server—will quickly bring a substandard system to its knees.



TIP Windows 2000 is even more demanding on hardware performance than earlier versions of Windows! Do not be surprised if upon upgrading you find that the system doesn't run as well as it did prior to the upgrade.

Use the Windows 2000 Performance monitor to analyze system performance problems. This program is able to monitor all Windows 2000 performance indicators and indicators for a number of add-on components, such as Exchange Server, SQL Server, and others.

Analysis

First, do an analysis. Ask yourself what changed. Analysis of the problem means that you must determine why the computer worked yesterday but doesn't work today. For example, did you:

- Remove any software or system components?
- Clean up the drive, deleting files that you thought were unneeded?
- Install any new applications?
- Upgrade any applications?
- Upgrade the operating system (install any service packs?)
- Change system hardware?
- Experience a power failure or fluctuation?

To keep this chapter from becoming a general system failure analysis tool, I'll limit the effects of these items to what might happen to the registry.

Fixing Things Up

Next, decide if it is better to try to restore things to their original states either by reinstalling the component or application or by removing the offending item.

If there is a backup of the registry and the item in question, restoring to get the system back to a working state will probably be a good starting point. A stable system that is not having trouble is much easier to work on than a system that fails for unexplained reasons.

Once the system is restored, try the established method for removing the component, such as the Add/Remove Programs applet in the Control Panel or the application's uninstall program.

If there is no backup of the registry or component, a different tack must be taken. There are three possible avenues of attack:

- Try reinstalling the component. Typically, the installation program will restore any registry entries that are necessary for the component to run. Often, any customization done since the last installation will be lost, but that's life.
- Try finding an uninstall program. First, check the Add/Remove Programs applet in the Control Panel. If the component is listed, run uninstall from there. If the component is not listed, then check the component's directories. List all the executable (.exe) files. If there is one named uninstall or remove, this may be the program that you need. Don't forget to check the component's documentation regarding uninstallation procedures, too.
- If there is no uninstall program, and the application must be removed, and you are going to have to do this manually, read on.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

NOTE Some components, especially those that are system components, make so many changes to the registry that it is impossible to remove them manually. This is especially true for components that have replaced already existing components, as in the case of upgrading to a new version. Though you can remove the entries for the component in question, you cannot restore the entries that the component has changed to their original state; this is especially true if you don't know their original state. Changes to the registry are usually not well logged, so there is typically little to tell you what has changed from time to time.

Possible Problems, Quick Fixes

Some possible problems that cause the system to fail include those listed next. There are other problems too, so don't consider this list to be exhaustive.

An Application or a System Component Was Deleted

Say an application or a system component was deleted, perhaps in error. In this case, you would do the following:

- Try restoring the application's files. Running the application's installation program may be the best way to restore files, though many applications allow a single file to be restored from the distribution media. Be aware that some applications store the files on the distribution media in compressed format, so that the only way to restore a single file may be to reinstall the entire application.
- If that fails, try reinstalling the application. Reinstalling the application may be necessary when the application's files are not accessible on the distribution media. Be aware that some installation programs will delete user configurations and other items that either you or other users have modified since the original installation.
- If that fails, try removing the application with the application's uninstall program, then reinstall the application. Some applications try to be smart and only reinstall those files and components that have not already been installed. But you may be trying to replace a file that you suspect has been corrupted or trying to restore registry entries, and the setup program doesn't realize that. It's just trying to save some time! (Some time-saver, huh?) In this case, it will probably be necessary to remove the original application (use its uninstall program, if there is one) before reinstalling it.

Another Application Has Overwritten an Application's Files

A new application has been installed, and this new installation has overwritten a previously installed application's files. OK, this was probably an error, but you inadvertently installed the new application in the

existing application's directory. This sometimes happens when the two applications have the same default installation directory. More often, we simply make a mistake and choose the wrong directory. Most application setup programs won't warn that the path already exists. Major bummer. When you suspect an application's files have been overwritten, here are some things to do:

- First, use the new application's uninstall program to uninstall the new application. If the new application has been used, and there are user document or data files, back up these files. However, get rid of that new application; you can reinstall it later. If there is no automated uninstall for the new application that you are removing and you must remove it manually, make certain to clean up as many of the new application's registry entries as possible. If you don't, and you reinstall the new application into a new directory, the setup program may not properly update the registry because it thinks the application has already been installed.
- Next, restore the application's files, perhaps from a known good backup. If that fails, try reinstalling the application from the original distribution media.
- If that fails, try removing the original application with the application uninstall program and then reinstall the original application.

There Is an Error Reading the Application's Files

If there is an error reading the application's files, or the application crashes (faults) when executed, the application's files are probably damaged. What happened? There are several possibilities and some of them are very ugly, by the way.

Maybe a user error caused one or more files to be overwritten. In this case, things don't look too bleak. Generally, a restore of the application's files will allow you to recover from this situation. Use a known good backup or reinstall from the distribution media.

Maybe another application or the operating system overwrote one or more files. This is rare, but it could happen. Check file dates to try to determine when the file overwrites occurred and see if there is a way to determine the culprit. Restore the correct files and consider setting permissions to read/execute for everyone but an administrative userID that you won't use except to manage these files. Using file system permissions allows you to get immediate notification when a file overwrite occurs.



WARNING Permissions are the Windows 2000 way to protect applications and system files from unauthorized changes. Always set permissions so that most users, other than those who must have higher-level permissions, have read/execute permissions only. Allowing all users to have write permissions for system and application executable files is not a very good move, no matter how trusted the users are. Eventually, someone will unintentionally overwrite something, delete a file, or do some other damage.

There Is an Error Reading the Drive

Well, actually this is the beginning of the end of the world.

First, run `chkdsk` and determine what Windows 2000 is able to do to fix the problem. Realize that when Windows 2000 fixes a file on an NTFS drive, it doesn't fix the file; it only makes the file readable. Windows 2000 is not able to recover the file's contents—if it could, everything would be all right. Instead, it gives a message that says file so-and-so has been repaired, which is somewhat misleading in this respect. However, you must do this repair to be able to replace the file with the right one.

When `chkdsk` runs, it will tell you if there are any damaged files. Windows 2000 is able to recover from minor problems and errors on the drive. Don't worry about these types of errors; it is not unusual to have a drive reported as having minor errors.

After running `chkdsk`, you must make a decision. A backup at this point can't hurt, but don't back up over any existing backups. Use a fresh tape, or whatever your backup program backs up to, and put this backup to the side. Here are the actions I'd take, in order of preference:

1. Replace the drive and restore to the new drive from the most recent known good backup. Since drives usually fail in stages, a little bit at a time, it is possible that your backups are not going to help as much as you'd like. This is a judgment call—if you are confident that a recent backup is OK, try it. If you are not confident of your more recent backups—often errors develop over time and contaminate all backups long before they are discovered—don't use the backup.
2. Reformat the failing drive and restore from a known good backup.

3. Restore the entire drive from a known good backup without reformatting. This is sometimes necessary if, for some reason, the drive can't be formatted.
4. Try to restore specific files known to be defective, either from backups or from the application's distribution media.

If there is an error reading the application's files, or the application crashes (faults) when executed, the application's files are probably damaged. What happened? There are a few possibilities. Maybe one or more files were overwritten by user error. As I mentioned earlier, a restore of the application's files will generally allow you to recover from this situation.

Maybe another application or the operating system overwrote one or more files. Again, check file dates to try to determine when the file overwrites occurred, and see if there is a way to determine the culprit. Restore the correct files and consider setting permissions to read/execute for everyone but an administrative userID that you won't use except to manage these files. Using file system permissions allows you to get immediate notification when a file is overwritten.

Manually Removing Registry Entries

In Chapter 8, I described three programs that will automate the process of registry entry removal. But sometimes when repairing a problem, it is necessary to remove entries manually. Here I'll cover manual removal techniques.

Manual removal techniques are even more dangerous than using a program to clean out entries. Removing things by hand is tedious, and you won't be able to fully check registry integrity this way. Backups are in order before even thinking of starting to manually remove an entry from the registry.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**Finding Entries**

The first thing that you must do is find all the entries relative to the problem. This means you have to do a search.

Searching the registry with the registry editors is possible but not optimal. RegEdit has the best search capabilities, but rather than using it to search, try the following technique: Launch RegEdit and select My Computer. Next, select Registry \oslash Export Registry File. This will write the entire registry, excluding items such as the security hives, to a text file. Next, use a text editor to find your problem application. Sounds too easy, doesn't it? However, finding the application may present a few problems. What do you search for? Try searching for the executable name or directory name. Or try searching for the known name of the application. If none of these work, search for things such as the application's document file extension, if it has one.

There may be entries for applications in HKEY_LOCAL_MACHINE\Software. Many applications install subkeys here, but others do not. If you're looking for a potentially optional component of Windows 2000, check the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion and HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion subkeys. Virtually everything that is part of the Windows 2000 operating system and from Microsoft should have entries in these two subkeys.

Still having problems finding your application? Try reading through the registry line by line. Start in the CLSID section, HKEY_LOCAL_MACHINE\Software\Classes\CLSID in HKEY_CLASSES_ROOT.

Visually scan the registry, starting with HKEY_CLASSES_ROOT, then HKEY_LOCAL_MACHINE, to see if any entries match anything that you can associate with the errant application. Look at the program's name, its publisher—anything that might be a link. At this point, you are in detective mode.



NOTE Ever wonder how hackers break into systems? Oftentimes, it's by doing things just like this. They read anything about the system they can find. In short, they do just what you'll be doing.

Most of the time, the application will have entries grouped together under a subkey. Some applications will have other entries that tend to float, but these are rather unusual. Once you find something that matches what you are looking for, see if there is a CLSID (class ID) for it. Searching for a CLSID will be helpful in finding other entries in the registry for that application or component.

Removing Entries

Warning: If you are trying to remove entries from the registry, you should have exhausted all other alternatives; removing these entries is your last resort short of reinstalling Windows 2000. Got that? The odds are very good

that if you start hacking away at the registry, you'll destroy it.

But if you have nothing to lose, and you want to learn about the registry, this can be a way to do so. Back up the registry. I'd recommend having a parallel installation of either Windows NT, Windows 2000, or Windows 95/98 that you can boot to when you have totally destroyed your installed registry; this will allow you to restore the registry with a minimum of grief. If you don't have a parallel installation of Windows NT, Windows 2000, or Windows 95/98, now would be as good a time as any to install one.

To remove items, use RegEdit or RegEdt32. The delete capabilities of both of these programs are adequate. Select the entry (key, subkey, or value) to be deleted and remove it. Don't forget that the registry editor is editing the actual working registry; once you delete something; there is no easy way to restore it.

With the registry editors, you may want to consider saving any major subkeys to disk files before deleting them. By saving these subkeys to the disk, you will be able to restore them should you find that you've deleted the wrong thing. It is possible to delete items from the registry that will make it impossible to start or run Windows 2000. Having a complete backup of the registry that is restorable without using the affected copy of Windows 2000 is a very good idea.

Hints and Kinks from the Experts

Here are some hints from our experts.

When You Run a Repair, Setup Refuses to Recognize or Repair Your Installation

While hacking the registry one day, I destroyed my installation so thoroughly that Windows 2000 would not boot and Setup would not repair it. The NTFS file system was still intact, but the registry was so damaged that Setup did not recognize it as a valid installation. While I could have restored, I would have lost four hours of work.

To fix this problem, boot to an alternate installation of Windows 2000 and copy each key in the original %systemroot%\Repair\regback folder to the original %systemroot%\System32\Config folder as shown here:

```
copy c:\winnt\Repair\regback\Software._ C:\WINNT\system32\config\Software
copy c:\winnt\Repair\regback\Software._
O C:\WINNT\system32\config\Software.sav
copy c:\winnt\Repair\regback\Default._ C:\WINNT\system32\config\default
copy c:\winnt\Repair\regback\Default._ C:\WINNT\system32\config\default.sav
copy c:\winnt\Repair\regback\SAM._ C:\WINNT\system32\config\sam
copy c:\winnt\Repair\regback\SAM._ C:\WINNT\system32\config\SAM.sav
copy c:\winnt\Repair\regback\Security._ C:\WINNT\system32\config\security
copy c:\winnt\Repair\regback\Security._
O C:\WINNT\system32\config\security.sav
copy c:\winnt\Repair\regback\System._ C:\WINNT\system32\config\system
copy c:\winnt\Repair\regback\System._ C:\WINNT\system32\config\SYSTEM.ALT
copy c:\winnt\Repair\regback\System._ c:\winnt\system32\config\system.sav
```

Here is an example of the response:

```
copy c:\winnt\Repair\regback\Software._ C:\WINNT\system32\config\Software
C:\WINNT>copy C:\WINNT\repair\regback\sam ..
O C:\WINNT\repair\system32\config\*. *
Overwrite ..\system32\config\SAM.? (Yes/No/All): a
    1 file(s) copied.
```

After expanding these files, I was able to boot; and in my case, I was fully recovered. You may need to repair boot records using the Recovery Console.

(Courtesy of Jerold Schulman.)

Build an NTFS or FAT Boot Floppy

The Microsoft Knowledge Base article Q119467 describes the process of building a boot floppy for an NTFS partition. This is useful if you accidentally replace the boot disk hardware driver or lose your Boot Manager and

no ERD is available.

The procedure in the article did not work in my environment, but the following process did allow me to successfully boot. Try the method in Q119467 first.

If the Q119467 method fails, try this:

1. Since some of these files are hidden/system/read only, go to Explorer Options View and check the box labeled Show All Files and uncheck the box labeled Hide Files.
2. Use DISKCOPY to copy the first setup disk.
3. Delete all files on this new boot floppy.
4. If DOS is installed, copy NTDetect.com and Bootsect.dos to the floppy from your root.
5. Copy NTLDR from your root to the floppy, renaming it Setupldr.bin.
6. Copy NTBootdd.sys from your root to the floppy. This is a copy of your SCSI driver. If you don't have a SCSI NT disk, you don't need this. If you have a SCSI NT disk, and the SCSI BIOS is enabled, you don't need this; but it is a good idea to protect against SCSI BIOS failure, which will prevent booting.
7. Create a Boot.ini as follows, or just copy your C:\Boot.ini; spacing is very important:

```
[boot loader]
timeout=10
default= scsi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
scsi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows 2000 Professional"
```

In this listing, WINNT is my NT directory; no drive letter is allowed. "Windows 2000 Professional" could be any character string, such as "Glad I had this BOOT FLOPPY!"

It is a good idea to have a second instance of Windows 2000 installed on a different partition, preferably on a different disk. This will ensure that you can always boot if the second instance of Windows is referenced in this Boot.ini. You will also be able to boot to this alternate instance to repair your primary instance.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

Click to access!

CHAPTER 10

Programming and the Registry: A Developer's Paradise?

FEATURING:

<i>Windows 2000 registry API functions</i>	231
<i>To use MFC or not to use MFC?</i>	247
<i>Hints and kinks from the experts</i>	251

Disclaimer # 1: I'm a C/C++ programmer, so this chapter will deal with C/C++ programming. However, to be fair, I've included some Visual Basic for Applications registry programming in Chapter 14, "Microsoft Office Entries." All of the programming techniques discussed in that chapter are usable with virtually any version of Visual Basic.

Disclaimer # 2: I'm a Microsoft Visual C++ programmer. However, any development platform that uses MFC (Microsoft Foundation Classes) will be compatible with this chapter's content. Also, those registry manipulation techniques that are part of the Windows 2000 API are exposed in all development platforms as standard Windows 2000 API calls. So, if you are not using Visual C++, don't despair: Your system will be sufficiently similar. You should experience only minor problems in using everything discussed in this chapter with other languages and compilers on your system.

Disclaimer # 3: I could write an entire book on programming for the Windows registry. Remember, programming is an art, not a science, and there are many, many different ways to write your applications. Use MFC, don't use MFC, use C++ and classes, don't use C++ and classes, use a dialog interface, use a window interface, use a command-prompt interface, and so on. I don't spend a lot of time on the interface in this chapter; instead, I work more on the actual calls and functions that you, a programmer, would be using.



NOTE Much of what this chapter covers is directly applicable to both Windows 2000/NT and Windows 95/98. In Windows 95/98, many registry entries are in different locations, although the basic concepts are identical for the programmer. The operating system does a good job of masking these differences.

Remember the registry's history. You see, the history of the registry is important in understanding how the various registry functions work and the parameters that are passed to these functions. What is now the registry was, once upon a time, a set of .ini files (specifically Win.ini and System.ini). In addition, each application had its own .ini file. An application could store information in the Win.ini and System.ini files, but that practice didn't gain much acceptance for a number of reasons, including performance and file bloat.

Much of the code that updated .ini files was reworked so that applications could easily work with the registry. In some cases, the applications didn't need to be modified at all; in other cases, there were minor modifications. However, all in all, you will see a lot of excess baggage in some registry functions. In some cases you will see that, even today, the same functions will still work with .ini files if need be.



NOTE The .ini files of old were divided into sections called *profiles*. Typically, a profile section is dedicated to a specific application or module.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Windows 2000 Registry API Functions

A program manipulates the registry using a number of registry functions. These functions are prefixed with Reg, and the rest of the function name describes the function's actual purpose in life. Table 10.1 lists the Windows 2000 registry functions, along with a short description of each one's functionality.



NOTE Two new functions have been added to Windows 2000, and the newer versions of the Windows SDK (Software Development Kit) show the registry hive HKEY_PERFORMANCE_DATA as a predefined type.

TABLE 10.1: WINDOWS 2000 REGISTRY FUNCTIONS

Function	Description
RegCloseKey	Closes the connection between the application and a specific registry object. The function RegOpenKey opens this connection.
RegConnectRegistry	Allows an application to modify a remote registry. It will establish a connection with the registry on a specified remote computer.
RegCreateKey	Creates a new registry subkey. This simple function allows no options; see RegCreateKeyEx for a more powerful version of this API.
RegCreateKeyEx	Creates a new registry subkey. This function allows setting security, options, and classes.
RegDeleteKey	Deletes an existing subkey that opened with RegOpenKey.
RegDeleteValue	Deletes an existing data key that opened with RegOpenKey.
RegEnumKey	Enumerates all the subkeys starting with the specified key or subkey. One object will be returned for each call to RegEnumKey, until the function returns the value ERROR_NO_MORE_ITEMS. This function exists for compatibility with earlier versions of Windows; programmers for Windows 2000 should use RegEnumKeyEx.


Desktop Library
 Click to access!

RegEnumKeyEx	Enumerates all the subkeys, starting with the specified key or subkey. One object will be returned for each call to RegEnumKeyEx until the function returns the value ERROR_NO_MORE_ITEMS. This function will retrieve the class name, the time of last modification, and the object's name.
RegEnumValue	Enumerates all the data keys in the specified key or subkey. One object will be returned for each call to RegEnumValue until the function returns the value ERROR_NO_MORE_ITEMS. This function will retrieve the name, the value, and the type for the object.
RegFlushKey	Causes any changes made to a registry entry to be written to the actual registry. This implies only simple buffering because, generally, changes to the registry are immediate.
RegGetKeySecurity	Retrieves the security attributes for a given registry object; the security may be set (changed) if the user has sufficient privileges.
RegLoadKey	Creates a new subkey under either HKEY_USERS or HKEY_LOCAL_MACHINE; the information to create the new subkey is contained in a file, the name of which is passed to the function.
RegNotifyChangeKeyValue	Tells the system to inform the caller if the specified object is changed or if the object's attributes are changed. If the object is deleted, no notification is sent. An event handler in the application processes the notification.
RegOpenKey	Used to open a registry object. This function is called before many other registry functions. The handle returned by RegOpenKey is then passed to other registry functions that require a registry handle. Microsoft recommends that RegOpenKeyEx be called by Windows 2000, Windows NT, and Windows 95/98 applications.
RegOpenKeyEx	Used to open a registry object. This function is called before many other registry functions. The handle returned by RegOpenKeyEx is passed to other registry functions that require a registry handle. RegOpenKeyEx handles security and other options that RegOpenKey does not handle.
RegQueryInfoKey	Returns information about the specified object.
RegQueryMultipleValues	Returns information about the data keys in a specified subkey.
RegQueryValue	Returns the value of the default (unnamed) value entry associated with each key and subkey. Microsoft recommends that RegQueryValueEx be called by Win32 applications.
RegQueryValueEx	Returns the value of the default (unnamed) value entry associated with each key and subkey. RegQueryValueEx handles security and other options that RegQueryValue does not handle.
RegReplaceKey	The registry is stored as a series of files, one file for each of the main keys. Upon restarting, the RegReplaceKey function tells the operating system to use a different file for this key. Use this function to back up and restore the registry and for disaster recovery.
RegRestoreKey	Restores the key's or subkey's contents from a file. The RegRestoreKey function will restore multiple objects, as many as are contained in the registry file provided.
RegSaveKey	Saves the key's or subkey's contents to a file. The RegSaveKey function will save multiple objects, as many as are specified to the registry file provided.

RegSetKeySecurity

Sets the specified object's security attributes. The user must have sufficient privileges to use this function.

RegSetValue

Sets the value of the default (unnamed) value entry associated with each key and subkey. Microsoft recommends that RegSetValueEx be called by Win32 applications.

RegSetValueEx

Sets the value of the default (unnamed) value entry associated with each key and subkey. RegSetValueEx handles security and other options that RegSetValue does not set.

RegUnLoadKey

Removes from the registry the specified object(s).



RegOpenUserClassesRoot

This function retrieves the HKEY_CLASSES_ROOT hive for a specific user. This function is useful when managing one user while not logged on as that user.



RegOverridePredefKey

Allows mapping of a predefined key or hive name (such as HKEY_CLASSES_ROOT) to a different key or hive. For example, you could map HKEY_CLASSES_ROOT to HKEY_CURRENT_USER\Temp\DLL.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

A number of different functions that work with the older .ini files are obsolete by Microsoft's standards, although they still allow support for legacy applications. These functions should not be incorporated into new code, although they may be encountered in legacy code. Use the functions described in Table 10.1 for new work. Table 10.2 lists the now obsolete registry functions.

TABLE 10.2: OBSOLETE WIN32 REGISTRY FUNCTIONS

Function	Description
GetPrivateProfileInt	Returns an integer value entry value from the specified location
GetPrivateProfileSection	Returns an entire section's contents
GetPrivateProfileSectionNames	Returns the names in a section
GetPrivateProfileString	Returns a string value entry value from the specified location
GetPrivateProfileStruct	Fetches a private structure from the specified location, comparing the checksum retrieved with the checksum that was written when the object was saved
GetProfileInt	Returns an integer value entry value from the specified location
GetProfileSection	Returns an entire section's contents
GetProfileString	Returns an integer value entry value from the specified location
WritePrivateProfileSection	Saves or writes to the specified location an entire section's contents
WritePrivateProfileString	Writes to the specified location a value entry string value
WritePrivateProfileStruct	Writes to the specified location, saving a checksum written with the object
WriteProfileSection	Writes an entire section's contents
WriteProfileString	Writes to the specified location a value entry string value


Desktop Library
[Click to access!](#)

In many cases, these functions will map directly into the registry, in the entry under `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping`. This mapping allows many legacy applications that used the `Win.ini`, `System.ini`, or `Control.ini` files to continue to function correctly. Support for this functionality is available under Windows 2000/NT only and does not apply to Windows 95/98 or any other version of Windows. However, for new code, do not use these functions: use the newer functions described earlier in Table 10.1.

Writing an application that uses the registry API calls is simple and straightforward. For example, an application that queries the registry for a certain object's value might be as simple as:

1. Open the object.
2. Query the object's contents.
3. Close the object.

Let's try that. In Windows 2000 (actually, all versions of Windows NT too), we have some advantages in that we can write console applications that interact with the registry. OK, Windows 95/98 has many of these advantages, too. Although console applications are not always the most user friendly, they are very quick and easy to write; and since this is not a programming book, we'll develop our example program as a console application.

To develop any application using Visual C++, use the New Project Wizard. Why not—after all, this wizard saves us a lot of work. Follow these steps:

1. In Visual C++, select File ã New.
2. Select the Projects tab in the New dialog box.
3. Select Win32 Console Application in the Project Type list.
4. Provide a name for the project (**Reg1**, say) and a location, then click OK.
5. Open the newly created `Reg1.cpp` file and drop in the code shown in Listing 10.1. It is best if you download the code from <http://www.sybex.com>, and cut and paste to save time and to avoid typing errors. However, if you do not have Internet access, you may type in this code directly.



TIP You may download the entire project from the Sybex Web site, at <http://www.sybex.com>. Click on Catalog, type the name of the book or the reference number from the book's ISBN (2615), and press Enter. From the main page for this book, click Downloads to go to the code. All of the files in the project are zipped into a single file called `reg1.zip`.

6. Build the project.
7. Correct your typing errors.
8. Rebuild the project and try out `Reg1`.

Slight modification of these steps will be necessary if you are not using Visual C++. Regardless, the basics are the same: create a new, empty console application and, in the main source file, add the code from Listing 10.1.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
LISTING 10.1: Reg1.cpp, a Program to Access the Registry

```
// Reg1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "windows.h"
#include "winreg.h"
#include <winerror.h>
#include "stdio.h"

int main(int argc, char* argv[])
{
#define MAX_VALUE_NAME 4096 // How big things can get
CHAR ClassName[MAX_PATH] = ""; // Buffer for class name.
CHAR KeyName[MAX_PATH]; // Name for the data value entry
char *szHive = "HARDWARE\\DESCRIPTION\\System";
char szBufferReturn[MAX_VALUE_NAME];
char szData[MAX_VALUE_NAME]; // Data value returned
DWORD dwcClassLen = MAX_PATH; // Length of class string.
DWORD dwcMaxClass; // Longest class string.
DWORD dwcMaxSubKey; // Longest sub key size.
DWORD dwcMaxValueData; // Longest Value data.
DWORD dwcMaxValueName; // Longest Value name.
DWORD dwcSecDesc; // Security descriptor.
DWORD dwcSubKeys; // Number of sub keys.
DWORD dwcValues; // Number of values for this key.
DWORD dwType = 0; // Type of data such as REG_SZ;
DWORD i = 0;
DWORD nBufferReturnSize = sizeof(szBufferReturn);
DWORD nDataSize = MAX_PATH; // Data value buffer size
DWORD dwcValueName = MAX_VALUE_NAME;
DWORD retCode;
FILETIME ftLastWriteTime; // Last write time.
HKEY hKey = NULL; // Handle for the registry key
```

**Desktop Library**[Click to access!](#)

```

HKEY      hKeyResult;
long      nReturnCode = 0;
PHKEY     phkResult = &amphKeyResult; // Result code hole!

printf("Reg1: version 2000 - Windows NT 4/Windows 2000\n");
hKey = HKEY_LOCAL_MACHINE;
hKeyResult = HKEY_LOCAL_MACHINE;

// First open the key specified in szHive:
if ((nReturnCode = RegOpenKeyEx(hKey,
    szHive,
    0,
    KEY_ENUMERATE_SUB_KEYS|KEY_EXECUTE|KEY_QUERY_VALUE,
    &amphKeyResult)) == ERROR_SUCCESS)
{
    // Get Class name, Value count. Display for the user.
    retCode = RegQueryInfoKey (hKeyResult, // Key handle.
        ClassName, // Buffer for class name.
        &dwClassLen, // Length of class string.
        NULL, // Reserved.
        &dwSubKeys, // Number of sub keys.
        &dwMaxSubKey, // Longest sub key size.
        &dwMaxClass, // Longest class string.
        &dwValues, // Number of values for this key.
        &dwMaxValueName, // Longest Value name.
        &dwMaxValueData, // Longest Value data.
        &dwSecDesc, // Security descriptor.
        &dwftLastWriteTime); // Last write time.

    printf("\n\nLooking at HKEY_LOCAL_MACHINE\\%s\n\n", szHive);

    printf (
        "ClassName, '%s' \n"
        "dwClassLen, '%ld'\n"
        "dwSubKeys, '%ld'\n"
        "dwMaxSubKey, '%ld'\n"
        "dwMaxClass, '%ld'\n"
        "dwValues, '%ld'\n"
        "dwMaxValueName, '%ld'\n"
        "dwMaxValueData, '%ld'\n"
        "dwSecDesc, '%ld'\n",
        ClassName, // Buffer for class name.
        dwClassLen, // Length of class string.
        dwSubKeys, // Number of sub keys.
        dwMaxSubKey, // Longest sub key size.
        dwMaxClass, // Longest class string.
        dwValues, // Number of values for this key.
        dwMaxValueName, // Longest Value name.
        dwMaxValueData, // Longest Value data.
        dwSecDesc); // Security descriptor.

    printf("\n\n");

    for (i = 0, retCode = ERROR_SUCCESS;
        retCode == ERROR_SUCCESS; i++)
    {
        retCode = RegEnumKey (hKeyResult, i,
            KeyName, MAX_PATH);

        if (retCode == (DWORD)ERROR_SUCCESS)
            printf("Sub-key name = '%s'\n", KeyName);
    }
}

```

```

    }

    retCode = ERROR_SUCCESS;

    printf("\n\n");

    // Next get the value stored in Identifier:
    for (i = 0; i < 100 && nReturnCode == ERROR_SUCCESS; i++)
    {
        nBufferReturnSize = sizeof(szBufferReturn);
        szBufferReturn[0] = '\0';
        nDataSize = sizeof(szData);
        szData[0] = '\0';

        if ((nReturnCode = RegEnumValue(
            hKeyResult, i,
            szBufferReturn, &nBufferReturnSize,
            NULL,
            &dwType,
            (LPBYTE)szData, &nDataSize
        )) == ERROR_SUCCESS)
        {
            printf("Identifier is '%s'\n\n", szBufferReturn);
            nBufferReturnSize = sizeof(szBufferReturn);

            if (dwType == REG_SZ)
            {
                printf("Identifier contains '%s' REG_SZ \n\n",
                    szData);
            }
            else
            {
                printf("Identifier contains a non-string'\n\n");
            }
        }
        else
        {
            // We're done, check for errors now:
            if (nReturnCode != ERROR_NO_MORE_ITEMS)
            {
                // No need to tell we are at end of list...
                nBufferReturnSize = sizeof(szBufferReturn);

                FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM, NULL,
                    nReturnCode, 0, szBufferReturn,
                    nBufferReturnSize, NULL);

                printf("RegEnumValue() %ld failed '%s'!\n\n",
                    nReturnCode, szBufferReturn);

                printf("RegEnumValue() failed!\n\n");
            }
        }
    }
    // When done, always close the key!
    RegCloseKey(hKey);
}
else
{
    nBufferReturnSize = sizeof(szBufferReturn);

```



```

        FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM, NULL, nReturnCode,
                    0, szBufferReturn, nBufferReturnSize, NULL);

        printf("RegOpenKey() %ld failed '%s'!\n\n",
              nReturnCode, szBufferReturn);
    }

    return 0;
}

```



NOTE This is when I usually regale the reader with stories about my first Windows program. It took me about six months to get the basics of my first Windows interface displayed on the screen, and there was no functional code in that interface. Today, with Visual C++ and the wizards, I can do that six months of work in about ten minutes. Progress, ah progress—and to think there are those who'd choose to stifle this innovation.

A programmer is able to write a program that interacts with the registry using only a few lines of source code. Listing 10.1 shows the main source file for just such a program. The program in Listing 10.1 also requires simple stdafx.cpp and stdafx.h files. Listing 10.2 shows the stdafx.cpp file and Listing 10.3 shows the stdafx.h header (include) file for Reg1.cpp.

LISTING 10.2: stdafx.cpp, the Support Precompiled Header File for Reg1

```

// stdafx.cpp : source file that includes just the standard includes
//     Reg1.pch will be the pre-compiled header
//     stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

[Previous](#)
[Table of Contents](#)
[Next](#)

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)

[Products](#) |
 [Contact Us](#) |
 [About Us](#) |
 [Privacy](#) |
 [Ad Info](#) |
 [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**LISTING 10.3: stdafx.h, the Support Precompiled Header File for Reg1**

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#ifdef AFX_STDAFX_H__BD7FBDE9_14B4_11D2_88CB_0060970BB14F__INCLUDED_
#define AFX_STDAFX_H__BD7FBDE9_14B4_11D2_88CB_0060970BB14F__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

// TODO: reference additional headers your program requires here

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations
// immediately before the previous line.

#endif
//!defined(AFX_STDAFX_H__BD7FBDE9_14B4_11D2_88CB_0060970BB14F__INCLUDED_
```

To create your own Reg1 program, simply plug these files into a project. Output for this program uses simple printf statements, because this program doesn't have a Windows user interface.



NOTE The Reg1 program, though nominally a C++ program, is really a standard C program. Although it would be very easy to include additional C++ (and even MFC) code, I chose to keep this program as simple as possible.

Now, let's take a closer look at the Reg1 program. The first step after basic program initialization is to open a registry subkey:

```
if ((nReturnCode = RegOpenKeyEx(hKey,
    szHive,
    0,
```

```

        KEY_ENUMERATE_SUB_KEYS | KEY_EXECUTE | KEY_QUERY_VALUE ,
        &amp;hKeyResult
    )) == ERROR_SUCCESS)
    { // Get Class name, Value count. Display for the user.

```

In this code, I call `RegOpenKeyEx` and save the return code; the error handler will use the return code to display an error message, if appropriate. If `RegOpenKeyEx` returns `ERROR_SUCCESS`, the registry subkey was opened successfully. In `hKey` is the base key given to `RegOpenKeyEx`. We initialize this to `HKEY_LOCAL_MACHINE`. We initialize the desired key to open in `szHive`. The desired key to open is hard coded as “`HARDWARE\DESCRIPTION\System`”. Finally, `hKeyResult` will contain the handle to the key opened if the function is successful.

Once opened, the next step is to get some information about our key:

```

retCode = RegQueryInfoKey (hKeyResult,           // Key handle.
    ClassName,           // Buffer for class name.
    &dwcClassLen,        // Length of class string.
    NULL,               // Reserved.
    &dwcSubKeys,        // Number of sub keys.
    &dwcMaxSubKey,     // Longest sub key size.
    &dwcMaxClass,      // Longest class string.
    &dwcValues,        // Number of values for this key.
    &dwcMaxValueName, // Longest Value name.
    &dwcMaxValueData, // Longest Value data.
    &dwcSecDesc,      // Security descriptor.
    &ftLastWriteTime); // Last write time.

```

The call to `RegQueryInfoKey` returns the information about the key that’s shown in Table 10.3.

TABLE 10.3: INFORMATION RETURNED BY REGQUERYINFOKEY()

Variable in Geg1 (the User May Specify a Different Name)	Description
--	-------------

<code>ClassName</code>	Class name (this field may be blank under Windows 95/98)
<code>dwcClassLen</code>	Length of class string buffer and the returned length of the class string
<code>dwcSubKeys</code>	Number of subkeys in this key
<code>dwcMaxSubKey</code>	Longest object name
<code>dwcMaxClass</code>	Longest class string
<code>dwcValues</code>	Number of value entries in this subkey
<code>dwcMaxValueName</code>	Longest value name
<code>dwcMaxValueData</code>	Longest value data
<code>dwcSecDesc</code>	Security descriptor
<code>ftLastWriteTime</code>	Last write time for Windows 2000 and Windows NT systems.

Once we have some information about the subkey, we display this information for the user and carry on.

The next step in our simple program is to display all the subkeys that are contained within our target key. A simple loop that enumerates all the subkeys and prints the results of this enumeration does the job. We monitor the results of the `RegEnumKey` function call until an error is returned. Most loops would check the return value to determine what the error was, in order to build in error recovery; in our simple program, this is unnecessary.

```

for (i = 0, retCode = ERROR_SUCCESS;
    retCode == ERROR_SUCCESS; i++)
{
    retCode = RegEnumKey (hKeyResult, i,
        KeyName, MAX_PATH);

```

```

        if (retCode == (DWORD)ERROR_SUCCESS)
            printf("Sub-key name = '%s'\n", KeyName);
    }

```

The next step is to get each value entry's name and value. Due to the simple nature of this program, I only display keys that have a data type of REG_SZ and skip other keys. However, adding a more complex case statement would allow displaying all the different data types.

As in code used to enumerate subkeys, the printing of value entry values uses a loop and a test to ensure that the enumeration function, RegEnumValue, returns successfully.

This loop is composed of two steps. The first step is to get the value entry's name; the second step is to get the actual data value contained in the entry. Separate printf statements display this data for the user, as appropriate:

```

for (i = 0; i < 100 && nReturnCode == ERROR_SUCCESS; i++)
{
    nBufferReturnSize = sizeof(szBufferReturn);
    szBufferReturn[0] = '\\0';
    nDataSize = sizeof(szData);
    szData[0] = '\\0';

    if ((nReturnCode = RegEnumValue(
        hKeyResult, i,
        szBufferReturn, &nBufferReturnSize,
        NULL,
        &dwType,
        (LPBYTE)szData, &nDataSize
    )) == ERROR_SUCCESS)
    {
        printf("Identifier is '%s'\n\n", szBufferReturn);
        nBufferReturnSize = sizeof(szBufferReturn);

        if (dwType == REG_SZ)
        {
            printf("Identifier contains '%s' REG_SZ \n\n",
                szData);
        }
        else
        {
            printf("Identifier contains a non-string'\n\n");
        }
    }
    else
    {
        // We're done, check for errors now:
        if (nReturnCode != ERROR_NO_MORE_ITEMS)
        {
            // No need to tell we are at end of list...
            nBufferReturnSize = sizeof(szBufferReturn);

            FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM, NULL,
                nReturnCode, 0, szBufferReturn,
                nBufferReturnSize, NULL);

            printf("RegEnumValue() %ld failed '%s'!\n\n",
                nReturnCode, szBufferReturn);

            printf("RegEnumValue() failed!\n\n");
        }
    }
}
// When done, always close the key!
RegCloseKey(hKey);

```

}

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

This is error handling at its simplest. We save the return code from a registry function call. If the return is not ERROR_SUCCESS, something went wrong. For such a case, we can use FormatMessage to create a more user-friendly error message, which we can print on the screen for the user:

```

else
{
    // Could not open the registry object!
    nBufferReturnSize = sizeof(szBufferReturn);

    FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM, NULL, nReturnCode,
        0, szBufferReturn, nBufferReturnSize, NULL);

    printf("RegOpenKey() %ld failed '%s'!\n\n",
        nReturnCode, szBufferReturn);
}

```

A Windows program uses a message box to display the error-message text.

Does FormatMessage Always Return the Best Message?

It probably does, but not always; the problem is exactly as it seems. Regardless, whatever FormatMessage does return is better than just displaying an error code value to the user.

Take this example of an error message:

```
"Error number 259 occurred".
```

Descriptive? No.

Useful? No.

User friendly? No.

The better result comes from FormatMessage, formatted in a string:

```
"The Error 'No more data is available.' occurred in the call to
RegEnumValue"
```

This message, though not perfect, is much better and provides useful information to the user of the program. Programmers who display meaningless numbers in their error messages without explanatory


Desktop Library
[Click to access!](#)

text should be banned from ever using a computer again.

One caution, however: The error strings returned by `FormatMessage` may contain a trailing newline. It may be necessary to pare these from your error messages.

Figure 10.1 shows the results of an execution of `Reg1`.



NOTE Even when run under Windows 95/98, `Reg1` will provide useful output (although the output is different between Windows NT 4 and Windows 2000). This shows the compatibility between the Windows NT registry and the registry found in Windows 95/98.



NOTE For complete source code for the `Reg1` program, check this book's Web site at <http://www.sybex.com>; click Catalog, enter the book title or reference number from the book's ISBN (2615), and press Enter. From the main page for this book, click Downloads to go to the code.

FIGURE 10.1 `Reg1`, a simple command-prompt application, provides lots of interesting information about a registry subkey.

To Use MFC or Not to Use MFC? That Is the Question

C++, MFC (Microsoft Foundation Classes), and the concept of object-oriented programming (a.k.a. OOP) have all hit the big time. Some programmers actually believe that it is not possible to use old C calls in a C++ program. Actually, it is fully possible to use the Windows API calls in any program, whether the program is C, C++, MFC, or whatever. However, Microsoft did bundle a few of the registry functions into MFC to make programming a bit easier. For example, the `CWinApp` class contains a number of both documented and undocumented registry manipulation functions.

First, the good news: some registry functions are available in `CWinApp`. Now, the bad news: there are not many registry functions available in `CWinApp`. The functions listed in Table 10.4 are available to programmers directly. Don't despair, however—you can just call the plain old Windows API registry functions as well.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
TABLE 10.4: REGISTRY FUNCTIONS THAT ARE PART OF CWINAPP

Function, with Parameters Passed	Documentation	Description
void SetRegistryKey(LPCTSTR lpszRegistryKey)	Yes	This overloaded function fills in the m_pszRegistryKey variable using the passed string. This string would typically contain the company name. m_pszRegistryKey is used to create the necessary key(s) under HKEY_CURRENT_USER\Software\m_pszRegistryKey\m_pszProfileName. m_pszProfileName is set to m_pszAppName by SetRegistryKey.
void SetRegistryKey(UINT nIDRegistryKey)	Yes	This overloaded function fills in the m_pszRegistryKey variable from a string contained in the application's string resources. This string would typically contain the company name. m_pszRegistryKey is used to create the necessary key(s) under HKEY_CURRENT_USER\Software\m_pszRegistryKey\m_pszProfileName. m_pszProfileName is set to m_pszAppName by SetRegistryKey.
HKEY GetSectionKey(LPCTSTR lpszSection)	No	Returns hKey for HKEY_CURRENT_USER\Software\RegistryKey\AppName\lpszSection, creating it if it doesn't exist, where RegistryKey is the company name as stored in m_pszRegistryKey and AppName is the application name as stored in m_pszAppName. The caller must close the Hkey returned.
HKEY GetAppRegistryKey()	No	Returns Hkey for HKEY_CURRENT_USER\Software\RegistryKey\AppName, creating it if it doesn't exist, where RegistryKey is the company name as stored in m_pszRegistryKey, and AppName is the application name as stored in m_pszAppName. The caller must close the Hkey returned.
UINT GetProfileInt(LPCTSTR lpszSection, LPCTSTR lpszEntry, int nDefault)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegQueryValueEx() to get the value for the key specified in lpszEntry. Returns nDefault if the entry is not found. Works on an .ini file if a call to SetRegistryKey() has not previously been made.
BOOL WriteProfileInt(LPCTSTR lpszSection, LPCTSTR lpszEntry, int nValue)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegSetValueEx() to set the value for the key specified in lpszEntry. Returns FALSE if the entry cannot be set. Works on an .ini file if a call to SetRegistryKey() has not previously been made.

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

CString GetProfileString(LPCTSTR lpszSection, LPCTSTR lpszEntry, LPCTSTR lpszDefault = NULL)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegQueryValueEx() to get the value for the key specified in lpszEntry. Returns lpszDefault if the entry is not found. Works on an .ini file if a call to SetRegistryKey() has not previously been made.
BOOL WriteProfileString(LPCTSTR lpszSection, LPCTSTR lpszEntry, LPCTSTR lpszValue)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegSetValueEx() to set the value for the key specified in lpszEntry. Returns FALSE if the entry cannot be set. Works on an .ini file if a call to SetRegistryKey() has not previously been made.
BOOL GetProfileBinary(LPCTSTR lpszSection, LPCTSTR lpszEntry, LPBYTE* pData, UINT* pBytes)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegQueryValueEx() to get the value for the key specified in lpszEntry. The size of the buffer to return the data in is specified by pBytes. The parameter pBytes is set to the size of the returned data. Works on an .ini file if a call to SetRegistryKey() has not previously been made.
BOOL WriteProfileBinary(LPCTSTR lpszSection, LPCTSTR lpszEntry, LPBYTE pData, UINT nBytes)	Yes	Calls GetSectionKey() to open lpszSection, then calls RegSetValueEx() to set the value for the key specified in lpszEntry. The buffer containing the data to save is pData, and the data's size is specified by nBytes. Works on an .ini file if a call to SetRegistryKey() has not previously been made.
LONG DelRegTree(HKEY hParentKey, const CString& strKeyName)	No	Deletes the specified subkey from the specified parent. Since a registry subkey may not be deleted unless it is empty, a helper function is used to recursively delete further subkeys and keys.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)


NOTE A bright programmer could write a wrapper around the Windows API registry functions if desired. However, there's a reason that Microsoft didn't already do that: you'd actually gain no additional functionality or usability. On the other hand, it might be possible to improve the registry access, especially searching for and retrieving specific keys, with a C++ registry class. I'll leave it up to you to design your own registry class.

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

The process for using the CWinApp registry functions is simple:

1. Call CWinApp::SetRegistryKey() to tell MFC that your application is going to work with the registry rather than a separate .ini file.
2. Call the functions to retrieve or set values in the registry.

There is no closing code needed unless a call has been made to one of the following CWinApp functions:

- HKEY GetSectionKey(LPCTSTR lpszSection)
- HKEY GetAppRegistryKey()

If one of these functions is used, be sure that your application does a proper close of the registry key returned. Of course, check to ensure that the function didn't fail.

Hints and Kinks from the Experts

Our experts are not programmers, but here we go.

How Do I Migrate My Older Application to Support the Registry?

There are two problems here: program conversion and user migration. First, modify the application to use the registry. Replace all calls that access the application's .ini file with calls that access the registry. Establish the application's registry key, using the following convention:

```
HKEY_CURRENT_USER\Software\company\application
```

Here, *company* is your company name, or a name or word closely associated with your company; and *application* is the name of this particular application. If your company produces many applications, and they share some items, it may be useful to create both a subkey for each specific application and a separate subkey (called Common) for items shared between two or more applications or components.

The second issue is migration of user data from the .ini file to the registry. Do this at either of two points—during installation/upgrade time or during the first execution of the application. In either case, the

process is identical.

Create a helper program or function. This helper will read the .ini file using .ini file I/O routines. Then the helper will write registry entries based on the .ini file's contents.

Hint: After transferring the .ini file to the registry, consider marking it read only. This will prevent additional manual modifications by the user. In addition, add a comment at the top of the .ini file describing the move to the registry and advising the user that the .ini file is no longer used. Say something like: "The information in this .ini file has been moved to the registry. Items in this file will be ignored by *application name version n* and later." (Of course, fill in *application name* and *version n* with your application's name and version.)

(Courtesy of Peter D. Hipson.)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 11

The Performance Monitor Meets the Registry

JUMP TO TOPIC



Desktop Library

Click to access!

FEATURING:

<i>PerfMon1: a program to access HKEY_PERFORMANCE_DATA</i>	254
<i>A look at the program</i>	260
<i>Adding performance data to the registry</i>	271
<i>Hints and kinks from the experts</i>	273

A part of the registry that we've not discussed yet is the registry hive HKEY_PERFORMANCE_DATA, the registry's performance hive. This registry hive contains the necessary information to allow an application to successfully interact with and display performance data. Hidden from the registry editors, it is contained in a place in the registry that is only accessible programmatically; otherwise, it's not visible or editable.

That HKEY_PERFORMANCE_DATA does not actually exist is an interesting part of Windows. Windows stores this hive temporarily, mostly in memory. Quick updating of these performance-monitoring counters is necessary to avoid impacting performance.

Although it is somewhat more difficult to access the performance hive than other registry hives, this difficulty can be overcome using the tools provided in the Windows header file, winperf.h, distributed with Visual C++. In addition, in this chapter I'll show you a simple program I've created for browsing HKEY_PERFORMANCE_DATA. PerfMon1 is a program that views the information in HKEY_PERFORMANCE_DATA. Keep in mind that the program as presented in this chapter is only an example and doesn't actually do any useful retrieval of data—you will have to add that functionality.



NOTE Are the performance data and HKEY_PERFORMANCE_DATA part of the registry? Well, that's a good question. Hardly ever mentioned, they show up in only 11 TechNet articles. It seems obvious that they weren't part of Microsoft's original conception of the registry. HKEY_PERFORMANCE_DATA is an example of Microsoft *extending* the registry functionality and interface to provide special services.

PerfMon1: A Program to Access HKEY_PERFORMANCE_DATA

The PerfMon1 program is a simple console application, displaying its voluminous data using printf() statements. To keep this example as simple as possible, I've forgone any semblance of a user interface. The example program's inability to do any real monitoring is for the same reason—simplicity!



NOTE When you're using PerfMon1, I suggest you use I/O redirection and capture the data into a file. Then edit or browse the file. PerfMon1 might typically print over 50,000 lines of output; watching all of this scroll past on the screen won't be any fun.

The performance data is entirely contained within the HKEY_PERFORMANCE_DATA hive, with the exception of the object and counter names and help information, which are contained in the key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib\009. There are only two values in this key: Counter and Help. Both of these values are REG_MULTI_SZ, with many entries in each.



NOTE Appendix D lists a typical installation's counters and objects. You can use the information in Appendix D to determine which items are objects and what counters are found in each object.

An Example Portion of PerfMon1 Output

First, let's take a look at PerfMon1's output:

```
PerfMon1 - Check out HKEY_PERFORMANCE_DATA! Version 2000
+-----+
Index: 10386
```

```
Counter: 10388
Counter: 10390
Counter: 10392
Counter: 10394
Counter: 10396
Counter: 10398
Counter: 10400
Counter: 10402
Counter: 10404
Counter: 10406
Counter: 10408
Counter: 10410
Counter: 10412
Counter: 10414
Counter: 10416
Counter: 10418
Counter: 10420
Counter: 10422
Counter: 10424
Counter: 10426
Counter: 10428
Counter: 10430
Counter: 10432
Counter: 10434
Counter: 10436
Counter: 10438
Counter: 10440
Counter: 10442
Counter: 10444
Counter: 10446
Counter: 10448
Counter: 10450
```

Counter: 10452
Counter: 10454
Index: 10548
Counter: 10550
Counter: 10552
Counter: 10554
Index: 10524
Counter: 10526
Counter: 10528
Counter: 10530
Counter: 10532
Counter: 10534
Counter: 10536
Counter: 10538
Counter: 10540
Counter: 10542
Counter: 10544
Counter: 10546

PerfMon1's Performance Counters

To make sense of PerfMon1's output, you can look up the object and counter IDs in Appendix D. It'll also help to start off with a good understanding of what *counters* are. A counter is an item that indicates how many times, or at what rate, a given event happens. For example, there is a counter named Page Faults/Sec. This counter indicates the number of page faults (that is, the number of times the system needed to use virtual memory) that occurred in the previous second. Most, if not all, counters are based on a time interval (that you, the user, may set). The process is that a temporary counter is used, and when the time interval expires, the counter in the registry is updated. Then, any performance-monitoring application is able to retrieve and display the updated counter.

And a performance object? A *performance object* is a collection of performance counters used to monitor a specific functionality.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

This site is brought to you by
 EarthWeb



- HOME
- SUBSCRIBE
- SEARCH
- FAQ
- SITEMAP
- CONTACT US

KEYWORD SEARCH

- ▶ Search Tips
- ▶ Advanced Search

PUBLICATION LOOKUP

JUMP TO TOPIC



Desktop Library
 Click to access!



Mastering Windows 2000 Registry
 by Peter D. Hipson
 Sybex, Inc.
ISBN: 0782126154 **Pub Date:** 01/01/00

Search this book:

[Previous](#) [Table of Contents](#) [Next](#)

The output begins with the Active Server Pages performance object at index 10386. This object measures various parameters regarding Active Server Pages. Counters in Active Server Pages (see Figure 11.1) include those shown below:

Counter	Name
10388	Debugging Requests
10390	Errors During Script Runtime
10392	Errors From ASP Preprocessor
10394	Errors From Script Compilers
10396	Errors/Sec
10398	Request Bytes In Total
10400	Request Bytes Out Total
10402	Request Execution Time
10404	Request Wait Time
10406	Requests Disconnected
10408	Requests Executing
10410	Requests Failed Total
10412	Requests Not Authorized
10414	Requests Not Found
10416	Requests Queued
10418	Requests Rejected
10420	Requests Succeeded
10422	Requests Timed Out
10424	Requests Total
10426	Requests/Sec
10428	Script Engines Cached
10430	Session Duration
10432	Sessions Current
10434	Sessions Timed Out
10436	Sessions Total
10438	Templates Cached

10440	Template Cache Hit Rate
10444	Template Notifications
10446	Transactions Aborted
10448	Transactions Committed
10450	Transactions Pending
10452	Transactions Total
10454	Transactions/Sec



FIGURE 11.1 The Performance monitor's Add Counters dialog box showing Active Server Pages and some counters

The next object is number 10548, Indexing Service Filter. This object has only three counters:

Counter	Name
10550	Total indexing speed (MB/hr)
10552	Binding time (msec)
10554	Indexing speed (MB/hr)

Moving on to the next object in the PerfMon1 report, the next index in the Performance monitor system is number 10524, or in more friendly terms, Indexing Service. This object is a bit different from Indexing Service Filter, but why? It's because the actual indexing service is different from the indexing service filter. The counters in the Indexing Service performance object are shown here:

Counter	Name
10526	Word lists
10528	Saved indexes
10530	Index size (MB)
10532	Files to be indexed
10534	Unique keys
10536	Running queries
10538	Merge progress
10540	# documents indexed
10542	Total # documents
10544	Total # of queries
10546	Deferred for indexing

The Performance monitor itself is another example of the MMC (Microsoft Management Console). In Figure 11.2, it's showing the first few counters for the system's Active Server Pages usage.

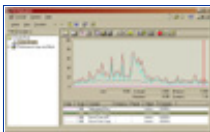


FIGURE 11.2 The Performance monitor displaying some CPU usage counters.

A Look at the Program

Now let's take a look at the PerfMon1 program itself. Listing 11.1 contains the main program, PerfMon1.cpp, a very simple program that is used to access performance counters and objects stored in the registry. To create PerfMon1, see the instructions for creating a project in Chapter 10.



NOTE You may download the PerfMon1 code from the Sybex Web site, at <http://www.sybex.com>. Click on Catalog, type the name of the book or the reference number from the book's ISBN (2615), and press Enter. From the main page for this book, click Downloads to go to the code.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
LISTING 11.1: PerfMon1.cpp

```
// PerfMon1.cpp : Defines the entry point for the console
//application.
```

```
#define _UNICODE
#include "stdafx.h"
```

```
#include <windows.h> // Standard windows header
#include <winperf.h> // Performance monitor definitions
#include <stdio.h> // printf() and other I/O stuff
#include <malloc.h> // memory allocation definitions.
```

```
#define BUFFERSIZE 8192 // initial buffer size,
```

```
#define INCREMENT 4096 // If too small, increment by 4096
```

```
int main(int argc, char* argv[])
```

```
{
// These objects are shown in winperf.h:
PPERF_DATA_BLOCK      PerfDataBlock = NULL;
PPERF_OBJECT_TYPE     PerfObjectType;
PPERF_INSTANCE_DEFINITION PerfInstanceDefinition;
PPERF_COUNTER_DEFINITION PerfCounterDefinition;
PPERF_COUNTER_DEFINITION PerfCurrentCounter;
PPERF_COUNTER_BLOCK   PerfCounterBlock;
```

```
// Program variables:
```

```
DWORD BufferSize = BUFFERSIZE; // Size of our buffer.
DWORD nShort; // If TRUE, display minimal data
DWORD i; // Index
DWORD j; // Index
DWORD k; // Index
char szOutput[512]; // temporary output buffer
```



Click to access!


```

// Create a reference, and initialize test buffer:
szOutput[0] = '\\0';

printf("PerfMon1 - Check out HKEY_PERFORMANCE_DATA! ");
printf("Version 2000\n");

// Check options, /S for short output display!

nShort = FALSE;
if (argc > 1)
{
    if (argv[1][0] == '/' &&
        (argv[1][1] == 'S' || argv[1][1] == 's'))
    {
        nShort = TRUE;
    }
}

// Allocate an initial buffer, which we'll resize later.
PerfDataBlock = (PPERF_DATA_BLOCK) malloc(BufferSize);

printf("+");
while (RegQueryValueEx(HKEY_PERFORMANCE_DATA,
    "Global", NULL, NULL, (LPBYTE) PerfDataBlock,
    &BufferSize) == ERROR_MORE_DATA)
{
    // The buffer is too small, so expand it!
    printf("-");
    BufferSize += INCREMENT;
    PerfDataBlock = (PPERF_DATA_BLOCK)
        realloc(PerfDataBlock, BufferSize);
}
printf("!\\n\\n");

// Buffer is sized OK now, let's get the first object!
PerfObjectType = (PPERF_OBJECT_TYPE)
    ((PBYTE)PerfDataBlock + PerfDataBlock->HeaderLength);

// loop through objects in HKEY_PERFORMANCE_DATA
for (i = 0; i < PerfDataBlock->NumObjectTypes; i++)
{
    if (nShort)
    {
        printf("Index: %ld\\n",
            PerfObjectType->ObjectNameTitleIndex);
    }
    else
    {
        printf("\\n");
        printf("Index to name in Title Database %ld\\n",
            PerfObjectType->ObjectNameTitleIndex);
        printf("Length of this object definition %d\\n",
            PerfObjectType->TotalByteLength);
        printf("Length of object definition %ld\\n",
            PerfObjectType->DefinitionLength);
        printf("Length of this header structure %ld\\n",
            PerfObjectType->HeaderLength);
        printf("use by analysis program to point to "
            "retrieved title string %ld\\n",
            PerfObjectType->ObjectNameTitle);
        printf("Index to Help in Title Database %ld\\n",

```

```

    PerfObjectType->ObjectHelpTitleIndex);
printf("Used by analysis program to point to "
    "retrieved title string %ld\n",
    PerfObjectType->ObjectHelpTitle);
printf("Object level of detail %ld \n",
    PerfObjectType->DetailLevel);
printf("Number of counters in each "
    "counter block %ld \n",
    PerfObjectType->NumCounters);
printf("Default counter to display %ld \n",
    PerfObjectType->DefaultCounter);
printf("Number of object instances %ld\n",
    PerfObjectType->NumInstances);
printf("Instance name Code page, "
    "or 0 if UNICODE %ld\n",
    PerfObjectType->CodePage);
printf("Sample Time in 'Object' units %ld\n",
    PerfObjectType->PerfTime);
printf("Frequency of 'Object' units %ld\n\n",
    PerfObjectType->PerfFreq);
}

// next get the counter block,
// containing counter information!
PerfCounterDefinition = (PPERF_COUNTER_DEFINITION)
    ((PBYTE)PerfObjectType +
    PerfObjectType->HeaderLength);
if (PerfObjectType->NumInstances > 0)
{ // first instance:
    PerfInstanceDefinition =
        (PPERF_INSTANCE_DEFINITION)
        ((PBYTE)PerfObjectType +
        PerfObjectType->DefinitionLength);

    // Next instance loop:
    for(k = 0;
        k < (DWORD)PerfObjectType->NumInstances; k++)
    {
        if (nShort)
        {
            printf("\n\tInstance '%S'\n", (char *)
                ((PBYTE)PerfInstanceDefinition +
                PerfInstanceDefinition->NameOffset));
        }
        else
        {
            printf("\n\tUnicode name of "
                "this instance '%S'\n",
                (char *)((PBYTE)PerfInstanceDefinition +
                PerfInstanceDefinition->NameOffset));
            printf("\tLength including the "
                "subsequent name %ld\n",
                PerfInstanceDefinition->ByteLength);
            printf("\tTitle Index to name "
                "of 'parent' object %ld\n",
                PerfInstanceDefinition->
                ParentObjectTitleIndex);
            printf("\tIndex to instance "
                "of parent object %ld\n",

```

```

    PerfInstanceDefinition->
    ParentObjectInstance);
printf("\tA unique ID used "
    "instead of matching the "
    "name to identify this "
    "instance, -1 = none %ld\n",
    PerfInstanceDefinition->UniqueID);
printf("\tLength in bytes "
    "of name; 0 = none %ld\n\n",
    PerfInstanceDefinition->NameLength);
}

PerfCurrentCounter = PerfCounterDefinition;

// Get first counter in this instance
PerfCounterBlock = (PPERF_COUNTER_BLOCK)
    ((PBYTE)PerfInstanceDefinition +
    PerfInstanceDefinition->ByteLength);
// Then retrieve all counters in this
// instance with a loop:
for(j = 0; j < PerfObjectType->NumCounters; j++)
{
    if (nShort)
    {
        printf("\t\tCounter: %ld\n",
            PerfCurrentCounter->
            CounterNameTitleIndex);
    }
    else
    {
        printf("\t\tLength in bytes of this "
            "structure %ld \n",
            PerfCurrentCounter->ByteLength);
        printf("\t\tIndex of Counter name "
            "into Title Database %ld\n",
            PerfCurrentCounter->CounterNameTitleIndex);
        wprintf(L"\t\tretrieved name string '%s'\n",
            PerfCurrentCounter->CounterNameTitle);
        printf("\t\tIndex of Counter Help into "
            "Title Database %ld\n",
            PerfCurrentCounter->CounterHelpTitleIndex);
        wprintf(L"\t\tretrieved help string '%s'\n",
            PerfCurrentCounter->CounterHelpTitle);
        printf("\t\tPower of 10 to scale %ld\n",
            PerfCurrentCounter->DefaultScale);
        printf("\t\tCounter level of detail "
            "(for controlling display complexity %ld\n",
            PerfCurrentCounter->DetailLevel);
        printf("\t\tType of counter %ld\n",
            PerfCurrentCounter->CounterType);
        printf("\t\tSize of counter in bytes %ld\n",
            PerfCurrentCounter->CounterSize);
        printf("\t\tOffset to the first "
            "byte of this counter %ld\n",
            PerfCurrentCounter->CounterOffset);

        printf("\n\n");
    }
}

// Get next counter.

```

```

    PerfCurrentCounter = (PPERF_COUNTER_DEFINITION)
        ((PBYTE)PerfCurrentCounter +
        PerfCurrentCounter->ByteLength);
} // for loop

// next instance, coming up next!
PerfInstanceDefinition = (PPERF_INSTANCE_DEFINITION)
    ((PBYTE)PerfCounterBlock +
    PerfCounterBlock->ByteLength);
} // for loop

} // if (PerfObjectType->NumInstances > 0)
else
{ // Get the first counter.
    PerfCounterBlock = (PPERF_COUNTER_BLOCK)
        ((PBYTE)PerfObjectType +
        PerfObjectType->DefinitionLength);

    // Get counters in a loop:
    for(j = 0; j < PerfObjectType->NumCounters; j++)
    {
        if (nShort)
        {
            printf("\tCounter: %ld\n",
                PerfCounterDefinition->CounterNameTitleIndex);
        }
        else
        {
            printf("\tLength in bytes of "
                "this structure %ld \n",
                PerfCounterDefinition->ByteLength);
            printf("\tIndex of Counter name "
                "into Title Database %ld\n",
                PerfCounterDefinition->CounterNameTitleIndex);
            printf("\tretrieved title string '%S'\n",
                PerfCounterDefinition->CounterNameTitle);
            printf("\tIndex of Counter Help "
                "into Title Database %ld\n",
                PerfCounterDefinition->CounterHelpTitleIndex);
            printf("\tretrieved help string '%S'\n",
                PerfCounterDefinition->CounterHelpTitle);
            printf("\tPower of 10 by which to scale %ld\n",
                PerfCounterDefinition->DefaultScale);
            printf("\tCounter level of detail (for "
                "controlling display complexity %ld\n",
                PerfCounterDefinition->DetailLevel);
            printf("\tType of counter %ld\n",
                PerfCounterDefinition->CounterType);
            printf("\tSize of counter in bytes %ld\n",
                PerfCounterDefinition->CounterSize);
            printf("\tOffset to the first "
                "byte of this counter %ld\n",
                PerfCounterDefinition->CounterOffset);
            printf("\n\n");
        }
    }

    // Data is (LPVOID)((PBYTE)PerfCounterBlock +
    // PerfCounterDefinition->CounterOffset);

```

```
        PerfCounterDefinition = (PPERF_COUNTER_DEFINITION)
        ((PBYTE)PerfCounterDefinition +
        PerfCounterDefinition->ByteLength);
    } // for loop
} // else if (PerfObjectType->NumInstances > 0)

// Get the next object to monitor
PerfObjectType = (PPERF_OBJECT_TYPE)
((PBYTE)PerfObjectType +
PerfObjectType->TotalByteLength);

} // Done! Go home and be sweet about it.

return(0);

}
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH



Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The program includes references to stdafx.cpp and stdafx.h. These two files are very simple. The file stdafx.cpp contains the lines:

```
// stdafx.cpp : source file that includes just the standard includes
// PerfMon1.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information
```

```
#include "stdafx.h"
```

```
// TODO: reference any additional headers you need in STDAFX.H
// and not in this file
```

The include file stdafx.h contains:

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//
```

```
#if !defined(AFX_STDAFX_H__766546C6_18BD_11D2_88CB_0060970BB14F__INCLUDED_)
#define AFX_STDAFX_H__766546C6_18BD_11D2_88CB_0060970BB14F__INCLUDED_
```

```
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
```

```
#define WIN32_LEAN_AND_MEAN // Exclude rarely-used
// stuff from Windows headers
#include <stdio.h>
```

```
// TODO: reference additional headers your program requires here
```

```
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately
// before the previous line.
```

```
#endif // !defined(AFX_STDAFX_H__766546C6_ . . .
```


[Click to access!](#)

)

Our performance information access program is simple and does not do much more than list the counters found in the registry. Of course, most performance-monitoring programs will want the actual performance data values too.

If you look at the `PERF_COUNTER_DEFINITION` structure, you will see that the last three items defined in this structure are `CounterType`, `CounterSize`, and `CounterOffset`. These three items represent the specific information needed to access a particular performance counter. Of course, to use the counter in a meaningful way, you'd also have to (at least) scale and format the counter properly, and then display it.

The following code segment shows the definition of the `PERF_COUNTER_DEFINITION` object:

```
typedef struct _PERF_COUNTER_DEFINITION {
    DWORD      ByteLength;
    // Length in bytes of this structure
    DWORD      CounterNameTitleIndex;
    // Index of Counter name into Title Database
    LPWSTR     CounterNameTitle;
    // Initially NULL, for use by analysis
    // program to point to retrieved title string
    DWORD      CounterHelpTitleIndex;
    // Index of Counter Help into Title Database
    LPWSTR     CounterHelpTitle;
    // Initially NULL, for use by analysis program
    // to point to retrieved title string
    LONG       DefaultScale;
    // Power of 10 by which to scale chart line
    // if vertical axis is 100
    // 0 ==> 1, 1 ==> 10, -1 ==> 1/10, etc.
    DWORD      DetailLevel;
    // Counter level of detail (for controlling
    // display complexity)
    DWORD      CounterType;
    // Type of counter
    DWORD      CounterSize;
    // Size of counter in bytes
    DWORD      CounterOffset;
    // Offset from the start of the PERF_COUNTER_BLOCK
    // to the first byte of this counter
} PERF_COUNTER_DEFINITION, *PPERF_COUNTER_DEFINITION;
```

Notice the last item in `PERF_COUNTER_DEFINITION`: `CounterOffset`. The offset from the start of the `PERF_COUNTER_BLOCK` is `CounterOffset`. This finds the counter in question. The `CounterSize` variable (just above `CounterOffset`) tells you how many bytes the counter in question occupies. Many counters are four bytes (a `DWORD`) in length.

Another important item for the counter is `CounterType`, a `DWORD` that describes the counter's type. The `CounterType` field is bitmapped and contains a lot of valuable information about the counter, as shown in Table 11.1. Figure 11.3 shows a mapping of the bits in `CounterType`.



FIGURE 11.3 CounterType bits are all significant.

TABLE 11.1: THE BITS IN COUNTERTYPE

First Bit	End Bit	Description
0	7	Reserved
8	9	Size field indicating the size, ranging from 0 to variable-length binary
10	11	The counter's type: number, counter, text, or 0

12	15	Reserved
16	19	The counter's type: hex, decimal, or decimal scaled (for counters); value, rate, fraction, and so on (for numbers); and other values (for text or 0 counter types)
20	21	The time base for the counter: either timer ticks, timer 100 nanoseconds, or an object-based timer
22	23	Calculation modifier, used for delta counters
24	27	Calculation modifier, used with inverse or multi-counters
28	31	Display format flag describing how to display this counter's data

The best information on these values is contained in `winperf.h`, one of the better-documented header files for Windows 2000.

To access a counter, see the section of Listing 11.1 where `PerfCounterBlock` is used. Set this initially to the first counter:

```
PerfCounterBlock = (PPERF_COUNTER_BLOCK)
  ((PBYTE)PerfInstanceDefinition +
  PerfInstanceDefinition->ByteLength);
```

You can access the data using the following line of code:

```
Data = (LPVOID)((PBYTE)PerfCounterBlock +
  PerfCounterDefinition->CounterOffset);
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

When using the code described here, be certain to properly de-reference the pointers.

Adding Performance Data to the Registry

The first part of this chapter delves into accessing preexisting performance data in the registry. It is also possible to add performance data to the registry for your own systems and applications. To do this, you must design your application or system to keep counters and modify the application or system to write these counters into the registry. Use the techniques described in the Microsoft Resource Kit version 3.5, Chapter 13, "Adding Application Performance Counters." This reference is available on the TechNet CD from Microsoft.

The process of adding performance data to the registry involves creating a .dll for collecting performance information. This .dll must have (at least) the Collect entry point functionality. The entry point names may be determined at the product design phase. When installed, your installation program *must* specify the entry point names. Here are some possible entry points used to support performance monitoring:

Library: The name of the .dll file that your application must have to collect performance data.

Collect: The routine that is used to report performance data upon request; this function is required.

Open: The function that is used to initialize the performance monitoring; this function is not required if the application is able to update performance counters without requests.

Close: The function that terminates the collection of performance data; as with Open, the Close function is not strictly required.

Adding counters to the registry is done using the LODCTR function. This function updates the registry and adds performance counters for an application or service as necessary. To use LODCTR, it is necessary to write an .ini file. The Microsoft Resource Kit version 3.5, Chapter 13, documents this process.

Here is an example of performance counters added to the registry of one Windows 2000 Server installation for the IIS Web service:

```
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:00002890
"Last Help"=dword:00002891
"First Counter"=dword:000027ee
```



```
"First Help"=dword:000027ef
```

In this example, the library to monitor performance data is `w3ctrs.dll`. There are function entry points defined for `Open`, `Close`, and `Collect`. Your application may add these entries at program installation time or the first execution of the program.

LODCTR automatically creates the final four entries—Last Counter, Last Help, First Counter, and First Help—when counters are loaded; your application should not need to modify these four entries.

An example of a LODCTR .ini file follows:

```
[info]
applicationname=MyApp
symbolfile=symfile.h

[languages]
009=English

[text]
OBJECT_1_009_NAME=MyApp
OBJECT_1_009_HELP=Monitor performance statistics for MyApp

DEVICE_COUNTER_1_009_NAME=Transactions
DEVICE_COUNTER_1_009_HELP=Number of transactions processed/second

DEVICE_COUNTER_2_009_NAME=LostCustomers
DEVICE_COUNTER_2_009_HELP=Number customers lost due to slow transactions
```

This example is simple. There are two counters with descriptive text and only one language. A real application might have many counters, depending on the application's complexity.

In summary, to create performance monitoring, follow these steps:

1. Create the necessary registry entries as described earlier in this section for `Library`, `Open`, `Close`, and `Collect`.
2. Move or copy the file specified in the `Library` entry to the `%systemroot%\System32` directory.
3. Use the LODCTR program to integrate the .ini file's counter entries into the registry.

Hints and Kinks from the Experts

Although not exactly programming oriented, the following hints are performance related.

Finding That Memory Leak Using Windows 2000

Much has been written about using Performance monitor to detect and isolate memory leaks. Two Microsoft Knowledge Base articles on the subject are Q130926 and Q150934. While these standard protocols work, the hit-and-miss method of finding the leaking process can be very time consuming. Here is an alternate method:

1. Start `PMON.EXE` from the Resource Kit.
2. Monitor paged and non-paged pool usage; these are the last two items in the second row. If these are increasing over time, you have a memory leak.
3. Monitor the commit counters in the second row. Increasing numbers over several hours indicate a probable leak.
4. Monitor the Commit Charge column. The process with the leak will have an increasing value.
5. To make it easier to monitor, copy the output to the Clipboard and paste it into Notepad. You can save the output as a file and compare it with later outputs.

Do this about once an hour over the duration of your testing.

(Courtesy of Jerold Schulman.)

Allowing Ordinary Users to Monitor Server Performance

How do you allow ordinary users to monitor server performance? Start by granting read access to the following server files:

- %Windir%\system32\PERFCxxx.DAT
- %Windir%\system32\PERFHxxx.DAT

The *xxx* part of the filename is the basic language ID for the system, 009 for U.S. English. Subtract 400 from HKEY_USERS*<SID of local server user>*\Control Panel\International\Locale to determine the basic language ID.

If the files are missing or corrupt, expand them from the CD-ROM. Using RegEdt32, give the user at least read access to the following:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib and all of its subkeys
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg

When adding objects to monitor, the user can select the server desired.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

PART III

Windows and Office Registry Entries

Have you ever noticed? Anybody going slower than you is an idiot, and anyone going faster than you is a moron.

—George Carlin

CHAPTER 12

The Windows 2000 User Interface: Changing How It Looks

FEATURING:

<i>Desktop settings</i>	279
<i>Other user interface settings</i>	294
<i>Console and command-prompt settings</i>	300
<i>Hints and kinks from the experts</i>	314

The Windows user interface is probably the most modified part of Windows. Virtually all users change some part of the user interface at some time. With Windows 2000, Microsoft has changed how Windows works with relation to the Desktop, Explorer, and Internet Explorer. For example, for a logged-on user, some Desktop functionality is stored in a different location in the registry than it was in previous versions of Windows NT.

Why? Well, for one thing, Internet Explorer replaces the Windows Explorer application. Now, this is not all bad, as it tends to integrate functions that were originally separate. In addition, Windows uses Internet Explorer for a number of other areas, including the Control Panel, the MMC (Microsoft Management Console), and other minor applications throughout Windows 2000.



Regardless of where they are done, many user interface registry modifications are easy to do and are relatively safe. For instance, to figure out what is where, I make changes both from the properties dialog boxes and in the registry and then just look at what's changed.

Messing up a registry entry for the user interface will usually not break the operating system; your display may not look like it should, but that's correctable. Typically, it will just make things either less friendly or not so pretty. However, some changes can cause serious damage to the system. Restore a damaged registry from a backup.

The Control Panel is the best tool to use to change the display properties in the registry; open the Display applet in the Control Panel (or right-click an unused place on the Desktop and select Properties). The registry controls everything—the Desktop, how things look, fonts, dialog box styles, colors, and DOS command-prompt windows.

There are some shortcuts for modifying the registry, other than using RegEdit or RegEdt32. For example, you can use the MMC Policy Editor snap-in that is part of Computer Management (Start ⌘ Programs ⌘ Administrative Tools), the Reg utility in the Resource Kit (see Chapter 4), and a few other tools to implement many of the settings. But the really interesting settings that are often desired, but seldom used, must be set using either very special tools or by direct, manual manipulation of the registry.

In this chapter, I'll cover each of the user interface items, describing each one, its location in the registry, and any tool available to manipulate it.

When you need to create a new key for setting values, make sure that you choose the correct data type for the key. Many keys are REG_SZ but hold a numeric value. Be careful not to create these keys with the wrong data type, such as REG_DWORD, because Windows 2000 won't recognize the value. Many of the settings for the user interface won't take effect until you log on the next time.



WARNING Do I have to say it? Back up your registry before making changes in it. Admittedly, changing the HKEY_USERS hive is less dangerous than changing HKEY_LOCAL_MACHINE, but there are hazards in making any changes to the registry.

Desktop Settings

The Desktop is the single most modified part of the Windows user interface. Users quickly put on a bitmap for their wallpaper, set a background pattern, customize the size and configuration of windows, and change colors—all with wild abandon.

Windows NT 4 has a subkey in the registry that holds the Desktop settings. However, Windows 2000 changes this (and in a complicated way, too!).

There are two users to consider: There's the currently logged-on user. For me, the currently logged-on user is the Administrator; you may have used a different name for your normal logons, but the principles are the same regardless. The other user is the one your computer considers to be the user when no user is actually logged on. This user is very limited in what they can do. And, yes this user has a name—the user is .DEFAULT. Notice the period, the first character in that user's name; it is significant in that Windows knows that this is the user to use when no other user is logged on. I'll refer to this user as the "default user."

Unfortunately, these two users are somewhat different. For example, the default user does not use Internet Explorer to manage the Desktop. (What management is there, anyway—no icons, only wallpaper, background, and a few color settings.) This means that techniques that are usable on normal users may not work on the default user. However, this is not a major issue, since you configure the default user the same way that you did in Windows NT 4.

Themes are predefined configurations consisting of Desktop configurations, colors, sounds, icons, cursors, and other visual customizations for the user interface. With Windows NT 4, one of the first things to do is load the Microsoft Windows NT Themes application, which is available from Microsoft as part of the Windows NT Resource Kit. Themes allows you to load any Windows 95/98-compatible theme; this saves time when configuring the Desktop.

Users can create themes themselves or get them from a variety of sources. Many users rely on the themes that come with the Windows NT Resource Kit, Windows 95/98, or from a number of sites on the Internet.



NOTE I won't get into the issues of copyrights and themes. There are a number of themes for Windows that surely impinge upon the rights of others. For example, *Star Trek*, *Star Wars*, *Three Stooges*, *The Simpsons*, and a host of other themes floating around on the Internet. Some are licensed, and most are the creation of the fans that have made these themes available to the public, usually for free. If you use a particular theme, whether you pay for it or not is a value decision that you will have to make. Myself, I'd prefer that everyone respected the intellectual property rights of others.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

Although you can make many of the modifications mentioned in this chapter using various Properties dialog boxes, some modifications can only be done by changing the registry. Of course, I recommend using the easiest method to change things when possible—and that would be the Properties dialog box, in most cases.

The registry key HKEY_USERS\DEFAULT\Control Panel\Desktop contains the settings used by Windows 2000 when there is no logged-on user. Notice that this subkey includes many settings that probably won't mean much to the system. In the "no user logged on" mode, Windows 2000 *normally* doesn't display icons or much of anything else. Notice I said *normally*; there are ways to force Windows 2000 to do just this. For example, you can allow services to interact with the Desktop and start services before a user logs on.

The registry key HKEY_USERS\<<SID>\Control Panel is the key used to hold settings for the current user.



NOTE Note that <SID> is a placeholder for a user's SID (security identifier). Usually, finding your SID is easy; just peek at the HKEY_USERS hive. There will be two or three subkeys: .DEFAULT (for use when no user is logged on) and one or more subkeys identified by a SID value—this is your SID.

Backgrounds and Wallpapers

At some point, every user has probably wondered about the difference between a background and wallpaper. Both are bitmaps. The background bitmap is typically small, consisting of one or two colors. Wallpaper is usually an image that covers some, or all, of the exposed Desktop; this image can be as large as the Desktop, tiled, or stretched to fit.

The background is under everything on the Desktop. When drawing, the first thing that Windows 2000 does with the Desktop is to draw the background bitmap in the background color(s) selected. The Windows 2000 default is pea green—not my favorite color. When there is no background bitmap, Windows 2000 by default draws the background as a solid color.



NOTE We'll talk about changing that pea green background color later; don't worry. Having a solid-color background can improve system performance when compared to having a bitmap for the background.

Once the background is drawn, Windows 2000 draws the wallpaper. Windows draws the wallpaper in one of three modes:

Centered: One copy of the wallpaper image is drawn in the center of the screen. If the wallpaper image is too small to fit, then the background will be visible on those areas the wallpaper image does not cover. If the wallpaper image is too large, Windows clips parts to fit.

Tiled: The first copy of the wallpaper is drawn in the upper left corner of the screen; additional copies are drawn below and to the left to fill the entire screen with the wallpaper bitmap. Some bitmaps, by design, fit together well, hiding this seam; other bitmaps present a jagged, clumsy look when tiled.

Stretched: This refers to the Windows 2000 Plus! feature, which is built into Windows 2000 and is not an option as it was with Windows 95. Windows 2000 automatically stretches the wallpaper bitmap to fit the screen dimensions. This is the most commonly used mode, and the most inefficient mode.



NOTE If you primarily use one resolution, you can improve performance by doing the following: Instead of using a bitmap as wallpaper in stretched mode, use a graphics editor to stretch the bitmap to the screen's resolution. Then, select the Centered option to display the wallpaper. Substantial improvements in Desktop-update performance may be achieved by changing the mode.

After drawing the wallpaper, Windows 2000 draws objects, such as Desktop icons and other windows. Windows handles icons and their labels in a special manner. Windows draws the icon images over the wallpaper. Next, Windows fills the area under the icon's label with the background color, but not the background pattern; the label's background is a solid color. (Why? Because the labels would be very difficult to read if a pattern were used.) Finally the text for the icon is drawn, so what you see under the text of an icon is not the wallpaper, but the background color.

In addition, there are two sets of background/wallpaper settings. Windows 2000 maintains one set for the current user, and the other set for all users who do not have personalized values. Those users without predefined configurations are supported using the HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor subkey.

Wallpaper

The Wallpaper value entry is a string that must contain the filename of the image file for the wallpaper. The value to modify is in HKEY_USERS\\Software\Microsoft\Internet Explorer\Desktop\General.

Value entry:	Wallpaper
Type:	REG_EXPAND_SZ
Typical value:	"" (or a fully qualified filename)

The file should be any file that Internet Explorer is able to display, including HTML, HTT (Hypertext Template), JPG, and BMP files. The resolution of the image should be compatible with the current display mode. If the resolution is different from the display mode, it is not the end of the world, but the quality of the display, and system performance, will be compromised.

If you are specifying a file that is not in the %systemroot% path, be sure to include a fully qualified path name. Generally, the Display Properties dialog box will include path-name information regardless of where the file is located.

This parameter is compatible with logged-on users only. For the .DEFAULT user, the registry value to modify is in HKEY_USERS\.DEFAULT\Control Panel\Desktop.

Value entry:	Wallpaper
Type:	REG_SZ
Typical value:	"" (or a fully qualified bitmap image filename)

The file should be a standard (non-compressed) Windows 2000 bitmap, and the resolution should be compatible with the current display mode. If the resolution is different from the display mode, it is not the end of the world, but the quality of the display, and system performance, will be compromised.

If you are specifying a file that is not in the %systemroot% path, be sure to include a fully qualified path name. Since this variable is REG_SZ, do not use environment variables to specify the file's path. Alternatively, modify this value with Reg.exe.

[Previous](#)
[Table of Contents](#)
[Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

This parameter is compatible with the default user only. It allows you to configure the Desktop display before any user has logged on. When configuring the Desktop display before a user has logged on, consider placing your company logo there!



NOTE For users other than the default user, the image file type for a background is any type that Internet Explorer is able to display. This means that even hypertext files can be used for backgrounds. For the default user, the image type must be a bitmap (.bmp) file.

WallpaperStyle

The WallpaperStyle entry tells Windows 2000 to stretch or compress a wallpaper bitmap that is different from the Desktop in size or resolution to fit the Desktop fully. For all users except for the default user, this value is located in HKEY_USERS\<<SID>\Software\Microsoft\Internet Explorer\Desktop\General. For the default user, this value is located in HKEY_USERS\.DEFAULT\Control Panel\Desktop.

Value entry:	WallpaperStyle
Type:	REG_SZ
Typical value:	0

A value of 0 will result in unstretched wallpaper, while a value of 2 will stretch the wallpaper to fit the screen's resolution.

If carried to extremes, this mode can result in an unattractive Desktop. Generally, if the wallpaper image is close to the size of the Desktop, the appearance will be acceptable. Certain bitmaps stretch better than others, so if in doubt, try it.

The WallpaperStyle parameter is compatible with both specific users and the default user subkeys. It allows you to configure the Desktop display before any user has logged on. For the default user configuration, consider stretching either wallpaper that is a different size from the default logon screen or wallpaper designed to be stretched.

TileWallpaper

The TileWallpaper entry tells Windows to either tile or center the Desktop's wallpaper. For all users except the default user, this value is located in the subkey HKEY_USERS\<<SID>\Software\Microsoft\Internet Explorer\Desktop\General. For the default user, this value is located in HKEY_USERS\.DEFAULT\Control Panel\Desktop.



Value entry:	TileWallpaper
Type:	REG_SZ
Typical value:	0

A value of 0 will not tile the wallpaper, while a value of 1 will tile the wallpaper to fit the screen's resolution.

It is possible to have a single copy of the wallpaper that is not centered or located at the upper left of the Desktop; see the next two sections.

This parameter is compatible with both specific users and the default user subkeys. It allows you to configure the Desktop display before any user has logged on. For the default user configuration, when tiling, use either a small bitmap or a tileable bitmap.

WallpaperOriginX

The WallpaperOriginX entry allows you to set the origin for both tiled and un-tiled wallpaper displays as shown here. Located in HKEY_USERS\DEFAULT\Control Panel\Desktop, it may be necessary to create the value entry.

Value entry:	WallpaperOriginX
Type:	REG_SZ
Typical value:	0

This entry is useful if, for example, you want to set your wallpaper to be in one of the corners of the Desktop. This parameter works with both centered and tiled wallpapers.

This parameter is compatible with the default user only. It allows you to configure the Desktop display before any user has logged on. For the default user configuration, consider recentering the wallpaper to provide an aesthetically pleasing Desktop. I find that having a small company logo in the lower right or upper left corner of the screen can be pleasing!

WallpaperOriginY

The WallpaperOriginY entry allows setting the origin for both tiled and un-tiled wallpaper displays. Located in HKEY_USERS\DEFAULT\Control Panel\Desktop, it may be necessary to create the data value.

Value entry:	WallpaperOriginY
Type:	REG_SZ
Typical value:	0

This entry is useful if, for example, you want to set your wallpaper to be in one of the corners of the Desktop. This parameter works with both centered and tiled wallpapers.

This parameter is compatible with the default user only. It allows configuration of the Desktop display before any user has logged on. For the default user configuration, consider recentering the wallpaper to provide an aesthetically pleasing Desktop. I find that having a small company logo in the upper left or lower right corner of the screen can be pleasing!

Pattern

The Pattern entry contains a pattern drawn using the background Desktop color. It is stored as a string containing eight numbers. This value is located at HKEY_USERS\\Control Panel\Desktop for all but the default user, and at HKEY_USERS\DEFAULT\Control Panel\Desktop for the default user.

Value entry:	Pattern
Type:	REG_SZ
Typical value:	(None)

Internet Explorer converts each number to a line in a bitmap. Each one in the binary number will represent the Desktop color.

The patterns are stored in the key HKEY_USERS\\Control Panel\Patterns. There is a simple editor for modifying (and creating new) patterns. You start this editor in the Background tab of the Display Properties dialog box, by clicking the Pattern button.

This parameter is compatible with both specific users and the default user subkeys. This allows you to configure the Desktop display before any user has logged on. For the default user configuration, consider using a pattern when not using stretched or tiled wallpaper.

Task Switching

Most users switch tasks with the keystroke combination Alt+Tab, although some use the Taskbar. You can configure the Task Switch dialog box, displayed when Alt+Tab is pressed, if desired.

Settings for this dialog box are simple to implement. Since there is no task switching before a user logs on, these settings are only meaningful to logged-on users, and not the default user. Though the settings can be set for the default user, they will have no useful effect—there is no Task Switch dialog box for the default user!

Each of the task-switching values is located in the registry subkey HKEY_USERS\\Control Panel\Desktop.

CoolSwitch

The CoolSwitch entry controls whether Windows displays the task-change window.

Value entry:	CoolSwitch
Type:	REG_SZ
Typical value:	1

Versions of Windows earlier than NT 4 used two styles of task switching. In *direct switching*, the system cycles through the running applications as the task-switch keystroke combination is pressed. Usually, only the application's title bar is active until the user releases the task-switch key. The second method of task switching, *CoolSwitch*, is used to display a dialog box, similar to that used in Windows NT 4 and Windows 2000, to allow the user to select the application to switch to. Windows NT 4 and later only support the second method. Therefore, Windows 2000 does not use the CoolSwitch-enabling registry key. Setting this entry to any value other than 1 seems to have no effect. Microsoft recommends not changing this key, so that the key will be compatible with future upgrades.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CoolSwitchColumns

The number of columns in the CoolSwitch dialog box is set using CoolSwitchColumns, as shown here.

Value entry:	CoolSwitchColumns
Type:	REG_SZ
Typical value:	7

The CoolSwitch dialog box displays icons for each running application that has a displayable window; it displays these icons in rows and columns. The CoolSwitch dialog box does not display applications and components that have no window to display.

This registry key sets the number of columns (the number of icons across) displayed in the CoolSwitch dialog box. The default value, 7, is a reasonable choice for most resolutions. However, users may find that with low-resolution displays, fewer columns may be more appropriate. Users with high-resolution displays, running a large number of applications concurrently, may want more columns displayed.



NOTE Are more applications running than will fit in the CoolSwitch dialog box? No problem, CoolSwitch will scroll the icons automatically as the user presses the task-switch keystroke combination. However, setting both CoolSwitchColumns and CoolSwitchRows to 1 won't create a single-icon CoolSwitch dialog box.

CoolSwitchRows

The number of rows in the CoolSwitch dialog box is set using CoolSwitchRows.

Value entry:	CoolSwitchRows
Type:	REG_SZ
Typical value:	2

The CoolSwitch dialog box displays icons for each running application with a displayable window in rows and columns. Windows will not display applications and components without a main window, as they cannot be activated using the task-switch keys.

This registry key sets the number of rows (the number of icons up and down) displayed in the CoolSwitch dialog box. The default value, 2, is a reasonable choice for most resolutions. However, users may find that with low-resolution displays, fewer rows may be more appropriate. Users with high-resolution displays, running a large number of applications concurrently, may want more rows displayed.


[Click to access!](#)

Moving Windows

One of the nice things about Windows is that it gives you the ability to drag things. Windows may be moved using a drag operation, and objects may be dragged and dropped. You can move windows either by using the keyboard or by simply clicking the title bar and dragging the window itself.

Both icons and objects are draggable. This includes selections in documents for those applications that support drag-and-drop. Clicking on the object selects it, and moving the mouse begins a drag-and-drop operation.

There are three registry entries for dragging that are specific to Windows 2000: DragFullWindows, DragHeight, and DragWidth. DragFullWindows specifies whether a window's contents are displayed (a full-content drag) while moving it, or whether only the window's outline is displayed. The other two entries specify the size the box in a drag operation must be before Windows 2000 will actually consider the object as something to be dragged.

Why change the size of the drag box? Some users who have difficulty controlling the amount that they move the mouse may find it preferable to set a larger drag area.

Each of the drag-related values is located in the registry subkey HKEY_USERS\<>SID>\Control Panel\Desktop. Both logged-on users and the default user can have these values applied to them.

DragFullWindows

Earlier versions of Windows only supported a mode called outline dragging. This was necessary primarily due to the incredible lack of performance of early CPUs and video systems. Newer hardware, improvements in video driver technology, and other changes have brought full-window dragging to Windows.

DragFullWindows allows dragging both a window and the window's contents.

Value entry:	DragFullWindows
Type:	REG_SZ
Typical value:	1

To drag the entire window with contents, set this parameter to 1; to drag only a window outline, set this value to 0. This value may also be set in the Effects tab of the Display Properties dialog box. Look for the check box called Show Windows Contents While Dragging.

There is not much dragging done before a user logs on, so these entries probably won't matter much to the default user's configuration.

DragHeight

The DragHeight entry determines the height of the rectangle used to detect the start of a drag operation.

Value entry:	DragHeight
Type:	REG_SZ
Typical value:	2

Click an object and move the mouse pointer more than the distance set for DragHeight and/or DragWidth, and Windows 2000 will assume that a drag operation is being performed.

DragWidth

The DragWidth entry determines the width of the rectangle used to detect the start of a drag operation.

Value entry:	DragWidth
Type:	REG_SZ
Typical value:	2

Again, when you click an object and move the mouse pointer more than the DragWidth (and/or DragHeight), Windows 2000 will assume that a drag operation is being performed.

The Cursor

The cursor can be configured somewhat. The only parameter that can be set for the cursor (sometimes called the *text caret*) is the blink rate. Some video systems, such as portables and video projection systems, don't react well to fast blink rates. Setting the CursorBlinkRate entry (discussed next) to a higher value may make the cursor easier to see.

CursorBlinkRate

The CursorBlinkRate entry specifies, in milliseconds, the blink rate of the cursor. It's found in the subkey HKEY_USERS\

Value entry:	CursorBlinkRate
Type:	REG_SZ
Typical value:	530

A smaller value will make the cursor blink faster, while larger values make it blink slower.



NOTE Where'd that odd value of 530 milliseconds come from? Why not just use 500 milliseconds, which is half a second? Got me. I know this is a holdover from the earliest versions of Windows, though.

Menus and the Windows 95 User Interface

The Windows 95/98 user interface allows users more flexibility when selecting menu items. Menu selections follow the cursor better in Windows 2000 and display cascading menus as the cursor is positioned over them. Try your Start menu for a good example of menu cascading. The MenuShowDelay entry controls cascading-menu delays; I'll cover it next.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

MenuShowDelay

The MenuShowDelay entry controls how long Windows 2000 will delay before showing a cascading menu.

Value entry:	MenuShowDelay
Type:	REG_SZ
Typical Value:	400

If the user pauses on a menu item that has a cascading menu under it, after the time specified in milliseconds, Windows 2000 displays the cascading menu automatically.

Slower processors may work better with a smaller value. However, the default of 400 milliseconds works well in most Windows 2000 installations.

Keyboard Settings

There are several value entries used to configure the keyboard under Windows, which I cover below. For most systems, they will contain default values, though they are all easily modified by the user in the Control Panel's Keyboard applet.

InitialKeyboardIndicators

The InitialKeyboardIndicators entry is used to set or clear keyboard toggle keys, such as Num Lock and Caps Lock.

Value entry:	InitialKeyboardIndicators
Type:	REG_SZ
Typical value:	0

The Num Lock key can be problematic because many users wish to have it turned on at the time they log on. There are two ways to ensure that this happens. One way is to turn on the Num Lock key, then press Ctrl+Alt+Delete and select the Logoff button. Alternately, in the registry, change the user's setting in either HKEY_USERS\<<SID>\Control Panel\Keyboard or HKEY_CURRENT_USER\Control Panel\Keyboard. Change the InitialKeyboardIndicators entry from whatever value it already has to 2. This will force on the Num Lock key.



NOTE Other values in the Keyboard key control the other toggle keys, including the Caps Lock and Scroll Lock keys.



Desktop Library
Click to access!

KeyboardDelay

KeyboardDelay sets the delay before a pressed keyboard key will begin to repeat.

Value entry:	KeyboardDelay
Type:	REG_SZ
Typical value:	1

The auto-repeat function can be very valuable to anyone who needs to draw a line in their word processor, move the cursor with the arrow keys, or needs to repeat the same key a number of times over. With this benefit comes a problem: we sometimes inadvertently hold keys down longer than we should, leading to multippppple characters in our text. Yep, I paused a bit too long on the *p* in that word, trying to decide if my editor would let me do it!

On the other hand, if you are a good typist, then a quickly working key repeat with minimal delay will speed things up. Each has their own value that they'd like to see, and for this one, 0 is about a quarter of a second, and 3 is a full second, more or less.



NOTE Times are approximate. Though a computer is capable of measuring even very small amounts of time, it cannot accurately repeat such measurements.

KeyboardSpeed

The KeyboardSpeed entry sets how quickly keys will be repeated when a key is held down for longer than KeyboardDelay (above).

Value entry:	KeyboardSpeed
Type:	REG_SZ
Typical value:	31

The repeat rate can be between 2 and 30 characters per second, where the value of KeyboardSpeed ranges from 0 to 31. Too high a repeat rate causes the user to often type more repeating characters than were desired, while a low rate will cause the user to be frustrated and cranky.

Keyboard Layout

Keyboard Layout (notice the space in the name) holds what the user's current keyboard layout is. For example,

I can switch to a Thai keyboard layout and type in Thai script, like this **หหดำกญ** (no, that word doesn't mean anything). This registry subkey is located at HKEY_USERS\<>SID>\Keyboard Layout. Within this subkey are a number of other subkeys, used to hold information about using and customizing keyboards.



TIP It is easiest to use the Control Panel's Keyboard applet to modify Keyboard Layout objects.

The Mouse and the Microsoft IntelliMouse

There are a number of settings for the mouse and the new Microsoft IntelliMouse. The IntelliMouse has a design with a new feature—a wheel that scrolls windows. As Microsoft's IntelliMouse support improves, more and more applications will work with the scroll wheel. The scroll wheel has a switch functioning as a separate button. A user may assign functionality to this button.

WheelScrollLines

Only meaningful for the IntelliMouse, the WheelScrollLines entry specifies the number of lines to scroll whenever the user uses the mouse wheel. It's found in HKEY_USERS\<>SID>\Control Panel\Desktop.

Value entry:	WheelScrollLines
Type:	REG_SZ
Typical value:	3

The wheel has discrete degrees of movement that provide tactile feedback to the user. The default value is to scroll three lines.



TIP Windows 2000 at this point doesn't allow an easy method to modify this value; however, since you have this book, you can change it as needed.

It takes some time to experiment with and get used to using the IntelliMouse's wheel for scrolling. Before long, it can become second nature, and then it can be very fast and easy to use.



NOTE If you're using Windows NT 4, be sure to get Microsoft's latest drivers for the IntelliMouse. Although there is some native support for wheel mice in Windows NT, the latest drivers offer improved performance.

DoubleClickHeight and DoubleClickWidth

The DoubleClickHeight and DoubleClickWidth settings control how much the mouse may move before Windows 2000 won't consider two clicks in quick succession to be a double-click.

Value entry:	DoubleClickHeight or DoubleClickWidth
Type:	REG_SZ
Typical value:	4

For most users, the default values are fine. However, users with notebook computers that don't have good pointer resolution and users with handicaps may wish to make the double-click tolerance higher, especially when working with a high-resolution screen.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Other Mouse Values

Table 12.1 shows some of the other values in the registry that affect mouse performance. Each of these values is located in the HKEY_USERS\<>SID>\Control Panel\Mouse key. Some are changed using the Control Panel's Mouse applet; others require registry surgery.

TABLE 12.1: MISCELLANEOUS MOUSE VALUES

Value Entry	Type	Default or Typical value	Description
ActiveWindowTracking	REG_DWORD	0	Allows the user to bring a window to the top, as can be done in X Windows. Typically turned off, set this value to 1 to turn on this feature.
DoubleClickSpeed	REG_SZ	500	Sets how much time between mouse clicks before Windows won't consider two clicks in succession to be a double-click.
MouseSpeed	REG_SZ	1	The degree of scaling for the mouse-pointer movement. A setting of 1 doubles the speed; 2 quadruples the speed.
MouseThreshold1	REG_SZ	6	Used to change the acceleration-to-speed ratio for the mouse.
MouseThreshold2	REG_SZ	10	Used to change the acceleration-to-speed ratio for the mouse.
SnapToDefaultButton	REG_SZ	0	When set to 1, the mouse will automatically move to the current default button whenever Windows displays a new window.


Desktop Library
[Click to access!](#)

SwapMouseButton	REG_SZ	0	If set to 1, the right and left mouse buttons are swapped, marginally useful for left-handed users (ever seen a left-handed mouse?).
-----------------	--------	---	--

Other User Interface Settings

You can use a few other user interface settings with Windows 2000. Some of these just don't fit well in the previous section, so I'll give them their own home.

Displaying Your Favorites

With Windows 2000, displaying Internet Explorer's Favorites list in the Start menu allows you to quickly jump to an Internet Explorer favorite item. Simply open the Taskbar's properties dialog box, and in the Advanced tab, turn on Display Favorites. To do the same change in the registry, follow these steps:

1. At a command prompt, create a new directory (using the command MD) named WWW in the following location: "%userprofile%\Start Menu\WWW". Be sure to include the quotes in this command because this is a long filename.
2. Copy all your favorites, typically in %userprofile%\Favorites, to your new directory at %userprofile%\Start Menu\WWW.
3. In the registry, go to HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders.
4. Edit the value entry Favorites, changing its value to "%userprofile%\Start Menu\WWW". This will force Internet Explorer to use your new favorites directory instead of your original favorites.

Customizing the Properties Pop-up Menu

It is possible to add functionality to the properties pop-up menu that Explorer displays for a file or link. To customize the Windows 2000 generic properties pop-up menu to add an Open With menu selection, do the following:

1. In HKEY_CLASSES_ROOT*, create a new subkey, called openas.
2. In HKEY_CLASSES_ROOT*\openas, created in step 1, create a subkey called command.
3. In HKEY_CLASSES_ROOT*\openas\command, change the value entry, which is named (default) in RegEdit or <no name> in RegEdt32, to have the value %systemroot%\System32\rundll32.exe %systemroot%\System32\shell32.dll,OpenAs_RunDLL %1.

Custom Icons on the Desktop

You can change the icons for certain objects—My Computer, My Network Places, Recycle Bin (full), and Recycle Bin (empty)—using the Display Properties dialog box in the Control Panel. Fine and dandy. Nevertheless, what about changing the other system icons?

The icon that is most often changed is that yellow folder used when Windows 2000 displays a directory on the Desktop. With all those bright colors and complex icons, the yellow folder icon is just a bit plain. That one's easy to fix; you can change the icon in the Properties dialog box for the specific folder.

Other icons can be more problematic. For example, changing some of the icons on the Desktop can be most intimidating. Table 12.2 lists some of the Windows 2000 Desktop objects that have icons that are difficult to change.

TABLE 12.2: CLSIDS FOR SOME DESKTOP OBJECTS

Name	CLSID	Windows NT 4 Compatible	Windows 2000 Compatible
Inbox	00020D75-0000-0000-C000-000000000046	ü	ü
Internet Explorer 1.0	0002DF01-0000-0000-C000-000000000046	ü	ü

Internet Explorer 2 or later	FBF23B42-E3F0-101B-8488-00AA003E56F8	ü	ü
Microsoft Outlook	00020D75-0000-0000-C000-000000000046	ü	ü
My Computer	20D04FE0-3AEA-1069-A2D8-08002B30309D	ü	ü
Network Neighborhood	208D2C60-3AEA-1069-A2D7-08002B30309D	ü	
My Network Places	208D2C60-3AEA-1069-A2D7-08002B30309D		ü
Recycle Bin	645FF040-5081-101B-9F08-00AA002F954E	ü	ü
The Internet	3DC7A020-0ACD-11CF-A9BB-00AA004AE837	ü	ü
The Microsoft Network (MSN)	00028B00-0000-0000-C000-000000000046	ü	

To change the name for one of the objects mentioned in Table 12.2, go to HKEY_CLASSES_ROOT\CLSID and scroll down until you find the CLSID for the item that you wish to change. Open the subkey and note that there are by default three values:

(default), the default registry value for this subkey that contains the title text

InfoTip, a REG_SZ string containing the tip text

LocalizedString, a REG_SZ string containing a path to shell32.dll, a resource locator, and a second copy of the title text

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Modify the (default) and LocalizedString values to include your title rather than the default text. You can also modify the InfoTip text to change the text displayed when the user moves the mouse cursor over the item.



TIP I always create three new strings, prefixing each original name with old, and then store the original values. This allows me to restore the original text easily.

To change the Desktop icon for one of these objects, go to HKEY_CLASSES_ROOT\CLSID and scroll down until you find the CLSID (from Table 12.2) for the item that you wish to change. Open the subkey and then under that subkey open the subkey named DefaultIcon.

In the DefaultIcon subkey, there will be a default REG_SZ entry, containing the path to the icon to display. Simply change that path to another icon path.

Want More Icons?

There are two main sources of icons in Windows 2000. The first, shell32.dll, contains the icons used by Windows 2000 for many components. Most of the program's executable files contain icons, too.

Another file that contains only icons is moricons.dll. This file is located in the %systemroot%\System32 directory. This file contains hundreds of icons of all different types. If you find that none of the icons in shell32.dll are to your liking, check out the moricons.dll file.

For fun, I searched my %systemroot% directory for all files with the word icon in them, using the command `dir *icon*.*`. I found about 20 files that had a multitude of icons that I could use for applications on my Desktop. Just be sure that when you use the icon in another file, the file doesn't get uninstalled at some time in the future—it may be a good idea to create a new folder in system32 called something like My Icons, and place a copy of the file in there.

Figure 12.1 shows the registry entry for the Recycle Bin. This entry is the most complex of the Desktop icons, in that the Recycle Bin will automatically switch between an icon representing the full or empty state as necessary. Windows 2000 always displays the icon that is in the (default) entry and doesn't know about any other entries in the DefaultIcon subkey. Getting ideas here? You can hide a few icon definitions in the DefaultIcon subkey for later manual retrieval if you want.


Desktop Library
 Click to access!

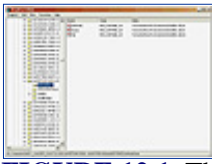


FIGURE 12.1 The Recycle Bin has two icons, plus the (default) entry for icons. You can change both the empty and full icons if you want.

The Recycle Bin's Icons

The Recycle Bin has two icons: one for empty and one for full. You may change either or both. If you check the Recycle Bin's subkey, you will see their arrangement (see Figure 12.1). Either the empty icon or the full icon will match the icon specified in the default entry. You must maintain this relationship. For example, if the empty icon entry matches the default entry, change both at the same time to the same value. The Recycle Bin automatically changes the default icon depending on its state by copying either the full or empty icon description to the default entry.

Windows 2000 displays only the icon in the value entry named (default) in RegEdit or <no name> in RegEdt32, because Windows knows nothing about whether the Recycle Bin has anything in it.

If the Recycle Bin's icons get out of sync, drop a file into the Recycle Bin, then empty it. This should force the Recycle Bin to resynchronize the displayed icon.

Enhancing the Start Menu

You may add a number of new entries to the Start menu, such as quick shortcuts to specific Control Panel applets. This process is not really a registry modification; but you do use CLSIDs, so we'll pretend that it is.

A folder named with <any name>.{<CLSID>}, as shown in the following format example, is handled differently from other folders by Windows 2000. Explorer will display the part of the name before the period. (Remember, Explorer displays the Start menu, too.) Explorer then uses part after the period, the CLSID number, to fill in the directory structure.

For example, a directory named Control Panel.{21ec2020-3aea-1069-a2dd-08002b30309d} will display the name "Control Panel" and all the items in the folder in Explorer and in the Start menu. The following Windows 2000 components support this behavior:

- Control Panel.{21EC2020-3AEA-1069-A2DD-08002B30309D}
- Printers.{2227A280-3AEA-1069-A2DE-08002B30309D}
- Dial-Up Networking.{992CFFA0-F557-101A-88EC-00DD010CCC48}
- Recycle Bin {645FF040-5081-101B-9F08-00AA002F954E}



NOTE When creating one of these special folders, don't forget to enclose the CLSID in curly braces. It won't work otherwise. The name, the portion before the period, may be any name that you desire.

Sounds Microsoft Never Gave Us

It is possible to add new sounds to Windows 2000. For example, every time you start RegEdit, you could have Ringin.wav play a difficult-to-ignore bell sound.

To add new sounds, follow these steps:

1. Start RegEdit (or RegEdt32) and open the subkey HKEY_CURRENT_USER\AppEvents\Schemes\Apps.



NOTE This process requires setting the (default) value and works best if done with RegEdit. RegEdt32 will work, though. Instead of performing step 5 in these instructions, use the Sounds applet in the Control Panel to set the value. RegEdt32 will not automatically create a (default) value. RegEdt32 calls the (default) value <No Name>. However, the Sounds applet will allow you to fill in the (default) value.

2. Create a new subkey named RegEdit, or whatever the name is of the program that you are adding sounds to.
3. In your new key, create a subkey called Open.

4. In your new Open key, create a subkey called .Current. Don't forget the leading period.
5. In your .Current subkey, set the (default) value to Ringin.wav, or whatever sound file you want. Remember that when using RegEdt32, the default value will display as <No Name>, if it exists.

After completing the entries in steps 2, 3, and 4, modify sounds played using the Control Panel's Sounds applet. The Sounds applet permits browsing and previewing sounds, making setting the sounds easier.

Make .dll Files Show Their Own Icons

Windows displays .dll files in Explorer with a generic .dll icon. This generic icon conveys no information about the .dll file, other than the fact that the file is a .dll.

Many .dll files have one or more icons. You can force Explorer to display the .dll's first icon, if there is one, or the generic Windows file icon, if there is no icon in the .dll file. Change the value contained in HKEY_CLASSES_ROOT\dllfile\DefaultIcon to the string "%1". The original value, "%SystemRoot%\System32\shell32.dll,-154", is the generic icon for .dll files and won't be used anymore.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Easter Egg Hunt?

As many of us know, Microsoft programmers put little credits screens and other goodies into each Microsoft product. These screens are popularly called Easter eggs because they are intended to be found by users, perhaps by accident.

Windows 2000 contains a couple of Easter eggs. They are in the OpenGL 3D Text screen saver.

To display the Easter egg, follow these steps:

1. Open the Display Properties dialog box either by context-clicking (right-clicking) the Desktop or by using the Control Panel.
2. In the Screen Saver tab, select the 3D Text (OpenGL) screen saver.
3. Click the Settings button.
4. In the Display group's Text box, enter **Volcano**.
5. Change any other settings you wish and click OK.
6. Click the Preview button.

You will see a list of all the volcanoes on the U.S. west coast.

Alternatively, try this:

1. Open the Display Properties dialog box either by context-clicking (right-clicking) the Desktop or by using the Control Panel.
2. In the Screen Saver tab, select the 3D Text (OpenGL) screen saver.
3. Click the Settings button.
4. In the Display group's Text box, enter the words **Not Evil**. That's two words, folks!
5. Change any other settings you wish and click OK.
6. Click the Preview button.

You will see a list of all the Windows 2000 developers.

Console and Command-Prompt Settings

Windows 2000 installations have a subkey under HKEY_CURRENT_USER called Console. Under Console, there is a subkey called %SystemRoot%_system32_ntvdm.exe. This section describes customizing these areas of the user interface. The values discussed in this section affect console and command-prompt windows that do not



have a custom configuration created.

All changes to the Console subkey will change the default values for all command-prompt windows created after the change takes effect. After opening a window, use the Properties dialog box to change the window's attributes.

The user may create additional subkeys under the HKEY_CURRENT_USER\Console key. Name each subkey created with the same name as a console window's title. When Windows 2000 creates a console window with the same name as a subkey found in HKEY_CURRENT_USER\Console, it will use the setting in this subkey to configure the window's default view.

More Than One Way to Set Command-Prompt Options!

Windows 2000 adds a new functionality to command-prompt customization. It is possible to have three different values set for many options:

1. The first level, found in the HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor key, affects all users.
2. The next level found, in HKEY_CURRENT_USER\Software\Microsoft\Command Processor, affects only the currently logged-on user (and is saved in that user's profile).
3. The final level, where the configuration is entered as a parameter passed to the command processor when it is invoked, affects only that particular session.

For more information about which command-prompt options and parameters may be set in these three ways, enter the command `CMD /?` at any command prompt.

Foreground and Background Colors

You can change a command prompt's foreground and background colors. I prefer a light gray background with black characters, or sometimes dark blue characters. There are three areas that affect the colors used for a command-prompt window: the color table entries, the command-prompt window colors, and the pop-up window colors.

By modifying the color table entries (ColorTable00 through ColorTable15) you create a custom color palette. Windows 2000 allows modification of the color palette in the Properties dialog box, although some users may be able to use the registry for this.

Setting the foreground and background indexes into the color table entries changes the window colors. Indexes are stored for both foreground and background as a single DWORD entry.

ColorTable00 through ColorTable15

The color table entries (Figure 12.2) allow users to select colors for fonts and backgrounds.

Value entries: ColorTable00 through ColorTable15

Type: REG_DWORD

Typical value: (RGB value, varies)

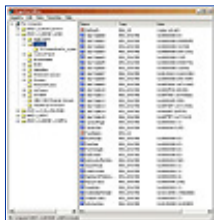


FIGURE 12.2 The HKEY_CURRENT_USER\Console key includes the color entries used in command windows.

The default colors for a command window are white on a black background. You can display the command window's properties dialog box from the window's System menu, or you can right-click the window's title bar and choose Properties.

The properties dialog box contains four tabs. Choose the final tab, Colors. In this tab, you can choose the

colors for the window's or the pop-up window's background and foreground from the standard 16-color palette (see Figure 12.3).



FIGURE 12.3 The Colors tab allows setting colors and color palettes.

The standard palette allows selecting the rather common and mundane colors. It also allows the user to customize colors using a set of edit controls. Some users will want the custom colors to be available. An alternative to setting these colors manually, one by one, is to change them in `HKEY_CURRENT_USER\Console`.

Each color index is a DWORD, consisting of red, green, and blue values; e.g. 00RRGGBB. Each color value may range from 0 to 255. The initial two digits are zeros.

PopupColors

Windows uses a pop-up window to inform you of some action or problem. You can set its colors independently from the colors of the command window itself.

Value entry:	PopupColors
Type:	REG_DWORD
Typical value:	0xF5

The DWORD value contains two bytes; one byte is used and the other is ignored. This allows specifying both the foreground and background color indexes. These colors are indexes to the `ColorTable` entries. The first four bits, 5 in the preceding typical value, are the foreground color index. The second four bits, the F in the preceding typical value, are the background color index.

ScreenColors

Command windows may have both foreground and background colors set using `ScreenColors`.

Value entry:	ScreenColors
Type:	REG_DWORD
Typical value:	0x07

The DWORD value consists of two bytes; one byte is used and the other is ignored. This allows specifying both the foreground and background color indexes. These colors are indexes to the `ColorTable` entries. The first four bits, 7 in the preceding typical value, specify the foreground color index. The second four bits, the 0 in the preceding typical value, specify the background color index.

Memory Used by Command-Prompt Windows

A couple of settings control the memory used by a command-prompt window. This memory is only for the display and does not, for example, affect memory available for applications.

[Previous](#) [Table of Contents](#) [Next](#)

HOME SUBSCRIBE SEARCH FAQ SITEMAP CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CurrentPage

The CurrentPage entry specifies the current page to use. The user should not reset this system variable.

Value entry:	CurrentPage
Type:	REG_DWORD
Typical value:	0x3

ScreenBufferSize

The ScreenBufferSize entry specifies the size of the screen buffer. The buffer size specifies the height and width value.

Value entry:	ScreenBufferSize
Type:	REG_DWORD
Typical value:	0x190050

The DWORD value has two bytes, allowing you to specify both the width and height in characters of the screen buffer. The low-order word (a DWORD consists of two words) specifies the width, while the high-order word specifies the height. For example, 0x00190050 specifies a screen buffer that is 0x19 (25) lines high and 80 characters wide.

Cursors

The cursor attributes allow customizing the cursor size. The standard cursor for a Windows 2000 command-prompt window is a modified underline cursor that can be set to a block cursor of varying size. I suggest modifying the underline cursor because it is actually a very short block cursor that looks like an underline, and therefore may be difficult to see.

CursorSize

The CursorSize entry specifies the percentage of the character cell that is filled with the cursor.

Value entry:	CursorSize
Type:	REG_DWORD
Typical value:	0x19 (that's 25 in decimal)



The Options tab of the properties dialog box allows setting the cursor to three sizes: Small, Medium, and Large (see Figure 12.4). Actually, this value may be a number between 0, Windows displays no cursor, and 99, Windows displays a full block cursor.



FIGURE 12.4 The Options tab allows setting many different options, such as three different cursor sizes.

Keep in mind that the cursor consists of discrete lines based on the command-prompt window's font size. As the font size gets larger, the user has more control over the size of the cursor.



NOTE Windows 2000 does not have any provision for a non-blinking cursor. Such is life—it just blinks on and on and on.

Fonts

Font attributes may be set, in a limited fashion, from the Font tab of the command-prompt properties dialog box (see Figure 12.5). More control of fonts is available in the registry. Setting font values requires an understanding of fonts, especially when using complex ones, such as TrueType.



FIGURE 12.5 The Font tab allows setting some font specifications. There is more flexibility when directly manipulating the registry.

For simple changes, such as font size and so on, use the properties dialog box. To select fonts that are not available normally, or sizes that the dialog box doesn't allow you to set, direct manipulation of the registry is the way to go.

FaceName

The FaceName entry specifies the font used to display characters in a command-prompt window and is by default a raster font.

Value entry:	FaceName
Type:	REG_SZ
Typical value:	(None)

Windows creates a *raster font* character in a cell, say 8 dots wide by 12 dots high, producing a moonscape font. Raster fonts are faster for Windows to process, but usually don't have much size flexibility. They are also generally lower in quality due to size constraints. Complex fonts, such as the TrueType fonts, are infinitely variable in size and are typically of higher quality when displayed in larger sizes. However, a complex font, such as a TrueType font, requires more system resources (CPU capacity) to display as the font must be drawn, or rendered, when used.

Most command windows use the default font, which is an undefined raster font. The font size may vary depending on screen resolution, although a default size in most installations is 8×12 , providing a reasonable, readable display.

FontFamily

The FontFamily entry specifies the font family for the window's display font. There are a number of different families, such as TrueType and raster.

Value entry:	FontFamily
Type:	REG_DWORD
Typical value:	0

This entry is a DWORD, with values that include:

0	Don't care which is used
10	Roman family
20	Swiss family
30	Modern family
40	Script family
50	Decorative family

As the most flexible font-family specification is 0 or "don't care," most users do not change this value.



WARNING Before setting font-family values, be sure to understand what is, and how to specify, a family value. Setting an invalid family value may cause the display to be different from what was expected.

FontSize

The FontSize entry specifies the value for the font displayed.

Value entry:	FontSize
Type:	REG_DWORD
Typical value:	0

Windows divides The DWORD value into two halves, allowing both the width and height of the characters to be stored in one value. The low order word specifies the width, while the high order word specifies the height. For example, 0x0008000C specifies a character that is 8 × 12 in size. (Remember, 0x000C in hex is 12 in decimal.)

FontWeight

The FontWeight value specifies whether a font is bold or light.

Value entry:	FontWeight
Type:	REG_DWORD
Typical value:	0

A default value of zero specifies a default character that is not bold. Values range from 0 to 1000; typical values are shown here:

0	Don't care how bold
100	Thin
200	Extralight
300	Light
400	Normal
500	Medium
600	Semibold
700	Bold
800	Extrabold
900	Heavy



NOTE The “don’t care” (0) value will be equated with the normal level (400) of bolding.

A generic bold/non-bold may be set from the Font tab in the command prompt’s properties dialog box when displaying a TrueType font. Raster fonts do not support the bold attribute.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb’s [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!
What the Window Looks Like

You can change the appearance of the command-prompt window in a number of ways (see Figure 12.6). Direct manipulation is possible; for example, the window location can be set using a simple drag-and-drop procedure. Other window attributes can be set using the registry or the properties dialog box for the window.

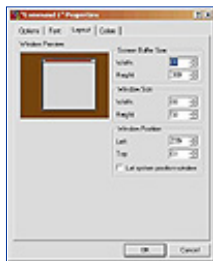


FIGURE 12.6 The Layout tab allows setting the size, position, and buffer size of the screen.

FullScreen

The FullScreen entry specifies whether the console session is full screen or windowed.

Value entry:	FullScreen
Type:	REG_DWORD
Typical value:	0

Most users put their command-prompt sessions in a window, and not full screen, for ease of use. The two values allowed for this entry are as follows:

1	Full-screen mode
0	Windowed mode



NOTE This option is usable only on Windows 2000 running on Intel x86 machines. RISC systems allow only windowed mode.

WindowSize

The WindowSize entry specifies the size of the command-prompt window. WindowSize is both a height and a

width value, each stored in the same DWORD.

Value entry:	WindowSize
Type:	REG_DWORD
Typical value:	0x190050

Windows splits the WindowSize DWORD into two bytes, allowing you to specify both the width and height of the screen buffer. The low-order word specifies the width, while the high-order word specifies the height. For example, 0x00190050 specifies a screen buffer that is 0x19 (25) lines high and 80 (0x50) characters wide.

WindowPosition

The WindowPosition entry specifies the Window's location on the Desktop relative to the upper left corner.

Value entry:	WindowPosition
Type:	REG_DWORD
Typical value:	0x00000000

This position is the number of pixels in the x- and y-axes. Windows splits the DWORD value into two halves, allowing both the x- and y-axes of the window to use the same DWORD. The low-order word specifies the width, while the high-order word specifies the height. For example, 0x00000000 specifies that the command prompt will be located in the upper left part of the Desktop at screen coordinates 0,0.



NOTE When setting WindowPosition in the properties dialog box, Windows 2000 keeps the user from entering a value that would move the window entirely off the Desktop. However, when setting these values manually, there is no safeguard to prevent placing the window off the screen. This presents some interesting possibilities in hiding a window. (When moving a window off the Desktop hides it, you have to maximize the window from the Taskbar to use it.)

The Command History Buffer

Windows 2000 maintains a command buffer that allows users to recall previously entered commands for re-execution. You can configure the command history buffer in both the command prompt's properties dialog box and the registry, setting the buffer size, the number of buffers, whether duplicates are stored, and so on.

Why do we set the number of commands stored in a buffer and the number of buffers, as well? I haven't found a satisfactory answer to this question. I am not aware of advantages of having multiple smaller buffers versus having a few large buffers. Microsoft has not clarified this.

DosKey performs the command history management, which is loaded by Windows 2000 every time a command-prompt session is started. DosKey allows the definition of keys, the creation of macros, and so on.

Windows 2000 has the command history buffer support built into the command processor, while in earlier versions, the buffer support was not as tightly integrated with the command processor. Regardless of where the command history buffer is maintained, the functionality does not change.

For more information on DosKey, enter the command **DosKey /?** at any command prompt. The help screen will assist you in using DosKey.

HistoryBufferSize

The command buffer is activated using the up and down arrow keys. The HistoryBufferSize entry specifies the number of commands stored.

Value entry:	HistoryBufferSize
Type:	REG_DWORD
Typical value:	0x32 (that's 50 in decimal)

A number of buffers can be set in the command prompt's properties dialog box, regardless of whether duplicate commands are saved or not.

HistoryNoDup

The `HistoryNoDup` entry specifies whether consecutive duplicate entries of a command will be stored in the command history buffer or not.

Value entry:	<code>HistoryNoDup</code>
Type:	<code>REG_DWORD</code>
Typical value:	<code>0x0</code>

This entry controls whether duplicate commands are saved or not. Values allowed in this entry are as follows:

<code>1</code>	Discard duplicates
<code>0</code>	Keep duplicates

NumberOfHistoryBuffers

`NumberOfHistoryBuffers` specifies the size of the Windows 2000 command buffer.

Value entry:	<code>NumberOfHistoryBuffers</code>
Type:	<code>REG_DWORD</code>
Typical value:	<code>0x4</code>

The command buffer is activated using the up and down arrow keys. This entry allows you to specify how many buffers are used (see also the previous section). The default value, 4, is usually adequate for most users.

Miscellaneous Settings

There are a few settings that don't seem to fit into the other categories I've discussed so far. Settings for the `InsertMode`, `QuickEdit`, and `CompletionChar` entries are helpful to users.

InsertMode

A command-prompt window allows a default insert/overwrite mode. (The default may be changed by pressing the Insert key on the keyboard.)

Value entry:	<code>InsertMode</code>
Type:	<code>REG_DWORD</code>
Typical value:	<code>1</code>

Most users set insert on, with a value of 1 (this is my preference), although some users find that overwrite mode is more convenient.

QuickEdit

`QuickEdit` is a mode that allows you to quickly mark information, copy it to the Clipboard, and paste the information from the Clipboard with the mouse.

Value entry:	<code>QuickEdit</code>
Type:	<code>REG_DWORD</code>
Typical value:	<code>0</code>

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

You can set this mode to QuickEdit disabled, which allows normal editing and is signified with a value of 0, or QuickEdit enabled, which is signified with a value of 1.

CompletionChar

CompletionChar is located in HKEY_USERS/Software/Microsoft/Command Processor. This value entry tells Windows 2000 to complete a partially typed filename when the user presses a specified key.

Value entry:	CompletionChar
Type:	REG_DWORD
Typical value:	0

Many users set this key's value to 9, the numeric value for the Tab key. Other keys could be used, but be careful not to select a key that is already used with Windows.

After setting this value and logging on again, open a command window. Next, in the root directory, type the command **dir w**. Next, press the Tab key or whatever key you assigned to the command-completion key. Notice how Windows now cycles through each directory or file that begins with the letter *w*.



NOTE The subkey `Command Processor` may not be present for users who have upgraded from previous versions of Windows NT. If this subkey is missing, create it and the `CompletionChar` value entry. Be sure to preserve both case and spaces in these names and to assign the key's data type as `REG_DWORD`, in order to ensure that they work correctly.

Hints and Kinks from the Experts

Here are some hints from our experts.

Do Your Users Close the Logon Script Window?

If your users tend to close the logon script window, and you want to prevent them from doing this, you may hide it. First, navigate to HKEY_USERS*each user*\Console. Next, select Edit or Add Value, as appropriate, and add/edit the two following `REG_DWORD` values:

- `WindowSize`, with the Radix at Hex; set it to 050005
- `WindowPosition`, with the Radix at Hex; set it to 04FF06FF

This makes the window very small and positions it off screen so users can't see it and, hence, can't close it.



If this window is invoked when running a command prompt, just maximize it, right-click the title bar, and size/position it using the Layout tab in the properties dialog box. When you click OK, check Modify Shortcut.

(Courtesy of Jerold Schulman.)

How Can I Get Windows 2000 to Use My Keyboard Layout During Logon?

To get Windows 2000 to use a specified keyboard layout during logon, perform the following steps:

1. Edit HKEY_USERS\DEFAULT\Keyboard Layout\Preload.
2. Double-click 1 and change the number to your local layout, which is at HKEY_CURRENT_USER\Keyboard Layout\Preload\1. (You may also change HKEY_USERS\DEFAULT\Control Panel\International\Locale to this value, but this is not mandatory.)
3. Log off and then log on.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 13

Networking and Registry System Entries

FEATURING:

<i>System entries</i>	317
<i>Networking entries</i>	317
<i>Disk, directory, and related entries</i>	321
<i>Other hardware support entries</i>	335
<i>Other software configuration entries</i>	338
<i>Hints and kinks from the experts</i>	339

For a typical user, the Windows 2000 internal registry entries probably occupy about half of the registry. Only after adding many applications does this proportion change much.

Many networking and system registry modifications are easy to do, although unlike changes to the user interface, they can be dangerous. Making an improper change can break the operating system, which may prevent you from booting your system. There are many changes that, when done improperly, will cause serious damage to the system, necessitating restoration of the registry from a backup.

You should make most networking changes using the Control Panel's Network applet. The main reason for this is that after making changes, the Network applet will check the registry to ensure that all networking registry entries are valid. This is not an exhaustive, problem-finding check; rather the Network applet just updates some entries if updating is necessary. In other words, don't rely on this updating to detect errors. Windows updates the registry during the binding phase.

Note that many of the settings for networking won't take effect until you restart the system. Simply logging on again isn't sufficient, as Windows reads some network settings only at boot time.



Do I Always Have to Reboot?

Windows 2000 has improved the updating of network settings. Generally, it is not necessary to reboot the computer unless prompted to.

Although there are times when Windows 2000 will work fine without being rebooted following a change made using the Control Panel, I recommend that you always reboot when the Network applet suggests that you do so. If you have systems or software that slow down the rebooting process, such as Microsoft Exchange Server, consider disabling these systems or software while making changes to the networking registry components.



WARNING Do I have to say it repeatedly? Back up your registry before making changes in it! Changing the system and networking sections of the registry is extremely dangerous. These sections are some of the most sensitive and difficult ones to modify.

System Entries

Many of the other registry hives contain system entries. Most system entries are located in hives HKEY_LOCAL_MACHINE, HKEY_CURRENT_USER (HKEY_USERS), and HKEY_CURRENT_CONFIG. You can ignore HKEY_CURRENT_CONFIG, since it contains only a reference to the CurrentControlSet subkey contained in the HKEY_LOCAL_MACHINE hive. That is, modifying an object in HKEY_CURRENT_CONFIG\System would simply modify the corresponding object in HKEY_LOCAL_MACHINE\System. Ditto for HKEY_USERS, since this information is available in HKEY_CURRENT_USER as well.

So at this point, if you examine the HKEY_LOCAL_MACHINE hive, you'll see most of the configurations that Windows 2000 uses for the system. Windows divides the entries into a number of major areas, including networking, disk, other hardware support, and other software configuration entries.

Networking Entries

Networking is a major component of Windows 2000. Networks connect virtually every Windows user, with many users connected to complex networks.

Often, we use the Control Panel's Network applet to make changes to the network configuration. In fact, I recommend that you use the Network a



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

When using Explorer to map a network drive, the dialog box used has a check box labeled Reconnect at Logon. When checked, this option creates a persistent connection. Regardless of the method used to create the connection, maintenance of persistence is the same: either Windows restores the connection when the user logs on the next time, generally the default, or the connection is lost when the user logs off.

Other than a value entry named Order, which specifies the order for the shared directory connections (if you edit it, the order in the drop-down changes), the only entry in HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Network\Persistent Connections that seems to do anything significant is SaveConnections. This value entry specifies the default value used with Explorer when mapping a network drive. The setting of the Map Network Drive dialog box's Reconnect at Logon check box is stored in this value entry. If this value is missing, Explorer will assume that the value is yes.

Just Because Microsoft Says It, Doesn't Mean It's Always So

Concerning persistent connections, the Microsoft Resource Kit for Windows NT 4 states the following about HKEY_CURRENT_USER\Network:

This object is no longer used. In previous versions of Windows NT, it stored persistent connections. Persistent connections are now stored in HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Network\Persistent Connections.

However, with some computers, I've noticed that Windows NT 4 and Windows 2000 still store persistent-connection information in HKEY_CURRENT_USER\Network, and not in the subkey Microsoft identifies in the Resource Kit. This is an example of how you should always check the system itself, rather than simply trusting the documentation.

RestoreConnection

Ghosted connections are persistent connections that exist when the actual connection to the server has not been reestablished after the user logs on. For example, say user John has persistent connections to 10 network drives. Each time John logs on, Windows 2000 could restore each network connection, establishing connections with each server. However, when doing this, what happens? First, restoring a number of connections is slow. Second, if one or more servers are unavailable at logon time, John will get some sort of message telling him that Windows could not make the connection. Both situations are problematic because John is always in a hurry. He knows that some of the servers are not available, but he doesn't care, because he won't be using those connections until he knows the server is accessible.

Windows uses ghosted connections when a user doesn't need or want an actual connection until there is a

need for the connection. Once the user uses the connection, Windows will make the necessary connection. In some instances, this technique can cause problems; for example, there will be a delay the first time that an inactive, ghosted connection is used. To avoid such problems, it is possible to disable ghosting with the following registry value:

Value entry:	RestoreConnection
Type:	REG_DWORD
Typical value:	0

Values used in this entry are as follows:

0	Windows will ghost the connection, restoring each connection as needed.
1	Windows restores the connection each time the user logs on.

This value entry is in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\NetworkProvider. Since this entry doesn't exist by default, you must create it in order to use it.

OptionalNames

Care to make your server appear as if it had a split personality? It is easier to make this change than to tell how to do it. In the registry key HKEY_Local_Machine\System\CurrentControlSet\Services\LanmanServer\Parameters, you can add the following entry:

Value entry:	OptionalNames
Type:	REG_SZ or REG_MULTI_SZ
Typical value:	SPLIT

You must reboot the system for this change to take effect, but that is a small price to pay to make your network appear to be larger than it really is.

Why do this? Several reasons. Say you add a new server. Eventually, this new server will replace an old, preexisting server. You know that many users have persistent connections to the old server. You create your new server with the name you choose, which will necessarily be different from the old server, since the old server is still in use on the network. You set up the new server, test it, and all is well.

At some quiet time, like when no clients are logged on, you migrate all resources from the old server to the new server. Then you turn off the existing server. Finally, you just add the OptionalNames entry using the name of the existing server. Tell users that you are migrating to the new server and that they should use the new server's name, not the old name, whenever they make new connections.

Users will eventually migrate to the new server's name, or you can migrate them manually without disrupting the system.

Improving Network Performance

Several networking settings will improve performance. Increasing buffering usually improves performance if sufficient memory is available. The following registry values, found in the subkey HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanWorkstation\Parameter, can help improve network performance. Some of these entries may not exist on your system. If they don't exist, you will need to create them.

First, modify or add a MaxCmds value:

Value entry:	MaxCmds
Type:	REG_DWORD
Typical value:	15

This registry entry may contain a value between 0 and 255. Since the default value is only 15, my recommendation is to increase it by steps of five, monitoring performance with each change.

Both MaxThreads and MaxCollectionCount also affect network performance:

Value entry:	MaxThreads
Type:	REG_DWORD

Typical value: 15

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

This registry entry should contain the same value as MaxCmds, shown in the previous example.

Value entry:	MaxCollectionCount
Type:	REG_DWORD
Typical value:	16

Specify the buffer used for the “character-mode named pipe” writes. You may choose a value up to 65535.

Disk, Directory, and Related Entries

There are probably a thousand different registry entries that affect disk drives. Unfortunately, many are specific to a given hardware configuration. The odds that any two computers would have the same hardware configuration are somewhat remote, unless you bought them all on the same day from the same vendor and had since made all changes yourself. In addition, the number of different permutations of hardware makes it difficult to localize common entries that would be significant to the majority of us. However, even with these staggering obstacles, I’ve plowed ahead and found as many generic disk registry entries as I could. Let’s hope that these will answer most of your questions.

Moving Windows to a New Directory

In Knowledge Base article Q154129, Microsoft outlines how to change the name of the Windows NT 4 installation directory. Microsoft has not given (at the time this book was written) a process to move Windows 2000; however, the techniques in this article should be useful for anyone who must move Windows 2000 to a new directory.

This is not an everyday action. However, for users who have upgraded earlier versions of Windows NT that were installed in a directory with the version number as part of the directory name, this process may make the installation look cleaner. For example, let’s say you upgraded an installation of Windows NT 4.0 installed in the directory WNT40. You’d like to rename this directory WNT2000 to reflect the current version number. Another example involves installing the new version of the operating system into a temporary directory, such as NewWNT, so that you have both versions of Windows NT installed at the same time.



NOTE No one has tested these renaming techniques with Windows 2000. Similar to moving Windows NT 4, moving Windows 2000 is probably possible, though the process will be difficult.

There are two distinct possibilities here. If you have installed Windows NT on an NTFS partition, you follow



a slightly complicated process that's described a little later. If Windows NT is on a FAT partition, you can change the name using another somewhat simpler process, described next.

FAT System Partitions: First Steps

Users who have installed Windows on a FAT partition have a somewhat simple task. FAT doesn't support security and is compatible with DOS and Windows 95/98. Use a boot diskette made on a DOS or Windows 95/98 computer to access the files on the hard drive.

I recommend that you have sufficient disk space to hold at least two copies of the operating system temporarily. This allows you to retain your original installation until you are able to ensure that the change in directory names is working correctly. If you retain two copies, be sure to rename the original so that the system won't see it.

Microsoft Says "No" to Be Safe

Microsoft doesn't recommend or support renaming the Windows system directory. (Can we blame them?) This means that if something goes wrong, you could be up the creek without a paddle. For this reason, before doing this, do a full backup to ensure that you are able to restore the original configuration just in case something goes wrong.

My own precaution is to carefully check installed applications to ensure that none are expecting Windows to install them in a fixed location.

One test to perform first is to dump the registry and search for the installation directory, such as C:\WNT351, in the registry. RegEdit and RegEdt32 could also do this search, but a dump edited with a good text editor may work better.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

Users of NTFS have to install a second copy of Windows to change the installation directory. Luckily, users of FAT partitions don't have to do this. For FAT-based systems, perform the following steps:



TIP If your system is on a FAT32 partition, then it will be necessary to use a later release of Windows 95 or any release of Windows 98. The initial release of Windows 95 is not compatible with FAT32.

1. Open a command-prompt window. Type the command **attrib -r-s c:\boot.ini**.
2. Create a bootable diskette, either from DOS or from Windows 95/98. Copy the `xcopy`, `edit`, and `move` command files to this diskette. Make sure they are compatible with the operating system version on the diskette.



TIP It is not necessary to use an ERD (Emergency Repair Disk) to boot a Windows 95/98 system. Windows 95/98 is capable of creating bootable diskettes with the `format` command, using the `/s` (for system) option.

3. Boot your computer from the bootable disk. Test to ensure that the `xcopy`, `edit`, and `move` commands are functioning correctly. If they aren't, correct this problem before continuing.
4. After ensuring that the necessary commands work, make a directory, using the `MD` command, with the new name that you wish to run Windows from—for example, type **MD WNT2000**.



WARNING An alternative technique would be to use the DOS command `move` to rename the directory. Dangerous since you would then have no backup to be able to go back to. I really do recommend making a full copy, just in case!

5. Use the `xcopy` command to copy all the files and subdirectories from the original Windows system directory to your new Windows system directory.



NOTE Use the `xcopy` command option `/e` to ensure that even empty subdirectories are copied. There may be empty subdirectories that are necessary for the system to work correctly.

6. Using the `edit` command, change the `boot.ini` file. Edit and change the lines with the original directory name to reflect the new directory name:

```
multi(0)disk(0)rdisk(0)partition(2)\WNT40="Windows NT Server Version 4.0"
```

Both lines contain a directory reference, in our example, it is WNT40. Change both to read:

```
multi(0)disk(0)rdisk(0)partition(2)\WNT2000="Windows 2000 Server"
```

In both cases, changing the text in quotes is a good idea. This is the prompt telling the user to select the operating system.



NOTE It is not necessary, but may be desirable, to change the attributes in the `boot.ini` file back to System and Read Only. If you do reset the attributes to System and Read Only, do so after everything is working correctly.

7. Remove your boot diskette and attempt to reboot the system. If the system reboots and runs correctly, continue.
8. Follow the steps outlined later under “Completing the Move.” When done with these steps, continue with the next step.
9. Again, reboot the system. If the system reboots and runs correctly, rename the original directory, WNT40 in our example, to a different name (say, WNT40_OLD). *Do not delete this directory yet*—wait until you have tested the change.
10. Now set the attributes back on `boot.ini`; use the command **attrib c:\boot.ini +r +s**.
11. After a suitable test period with no problems, typically several weeks, back up and then delete the original installation directory that you renamed in step 10.

NTFS System Partitions: First Steps

Users who have installed Windows 2000 on an NTFS partition have a somewhat more difficult task. NTFS is not accessible from DOS or Windows 95/98, at least not easily accessible in a read/write mode. Because of this limitation, it will be necessary to install a second operating system that is compatible with NTFS—Windows 2000 or NT.



NOTE I recommend that you have sufficient disk space to hold two copies of the original operating system temporarily, as well as a third, basic installation of the operating system. This allows you to retain your original installation until you are able to ensure that the change in directory names is working correctly. If you retain two copies, be sure to rename the original so that the system won't see it.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

To change the system directory name for Windows 2000, follow these steps:

1. Install a new, maintenance copy of Windows 2000 or NT into a new directory. (If you don't already have one installed, that is.) It is not necessary to install this copy of Windows 2000/NT on the boot drive. However, doing so will make things slightly easier.
2. Open a command-prompt window. Type the command **attrib -r-s c:\boot.ini**.
3. Restart the computer, and boot your new maintenance copy of Windows 2000/NT.
4. Log on as Administrator and open a command window.
5. Make a directory, using the MD command, with the new name that you wish to run Windows 2000 from; for example, type **MD WNT2000**.



WARNING An alternative technique would be to use the command `move` to rename the directory. This is dangerous since you would then have no backup to be able to go back to. I recommend doing a full copy, just in case!

6. Use the `xcopy` command to copy all the files and subdirectories from the original system directory to your new system directory.



NOTE Use the `xcopy` command option `/e` to ensure that even empty subdirectories are copied. There may be empty subdirectories that are necessary for the system to work correctly.

7. Using the `edit` command, change the `boot.ini` file. Edit the following line, where WNT40 is the original installation directory:

```
multi(0)disk(0)rdisk(0)partition(2)\WNT40="Windows NT Server Version 4.0"
```

This line contains a directory reference; in our example, it is WNT40. Change it to read:

```
multi(0)disk(0)rdisk(0)partition(2)\WNT2000="Windows 2000 Server"
```

In both cases, changing the text in quotes is a good idea. This is the prompt telling the user to select the operating system.



NOTE It is not necessary, but may be desirable, to change the attributes on the `boot.ini` file. If you do not immediately reset the attributes to System and Read Only, remember to do so after everything is working correctly.

8. Attempt to reboot the system, selecting your original installation of Windows 2000. If the system reboots OK, continue.

9. Follow the steps outlined under the next section, "Completing the Move." When done with these steps, continue with step 10.
10. Again, attempt to reboot the system, selecting your original installation of Windows 2000. If the original version of Windows 2000 reboots and runs correctly, rename the original Windows 2000 directory, WNT40 in our example, to a different name (say, WNT40_OLD). *Do not delete this directory yet—test the system thoroughly first!*
11. Now, set the attributes back on boot.ini, using the command **attrib c:\boot.ini +r +s**.
12. After a suitable test period with no problems, typically several weeks, back up and then delete the original installation directory that you renamed in step 10.

Completing the Move

Regardless of whether you have an NTFS or a FAT partition, it is necessary to perform the following steps. There is a file called setup.log, and you must modify this file. The Backup program's registry restoration and Windows 2000's Setup and Service Pack Setup programs use setup.log. Additionally, the registry itself will have many hard-coded references to the Windows 2000 system directory. You must modify these references, as well.

Perform the following steps on your FAT or NTFS system to complete the renaming process:

1. Back up the file setup.log to setup.bak using the **copy** command.
2. Open the file setup.log in the %systemroot%\Repair directory with a text editor, such as the command prompt's edit command or Notepad.
3. Globally change all references to the original installation directory with the new name that you have chosen.



WARNING Be careful not to change anything other than the installation directory name in this file, or the setup repair process will not be able to repair the system later.

A short section of a typical setup.log file is shown below. Assume that the original installation directory is WNT40. As an example, I've used underlines here to highlight the lines that would have to be changed:

```
[Paths]
TargetDirectory = "\WNT40"
TargetDevice = "\Device\Harddisk0\Partition1"
SystemPartitionDirectory = ""
SystemPartition = "\Device\Harddisk1\Partition1"
[Signature]
Version = "WinNt5.0"
[Files.SystemPartition]
NTBOOTDD.SYS = "ataboot.sys", "ad03"
NTDETECT.COM = "NTDETECT.COM", "11f1b"
ntldr = "ntldr", "3aae6"
arcsetup.exe = "arcsetup.exe", "3036c"
arclldr.exe = "arclldr.exe", "33a86"
[Files.WinNt]
\WNT40\system32\drivers\kbdclass.sys = "kbdclass.sys", "8a28"
\WNT40\system32\drivers\mouclass.sys = "mouclass.sys", "98d7"
\WNT40\system32\drivers\uhcd.sys = "uhcd.sys", "d727"
\WNT40\system32\drivers\usbd.sys = "usbd.sys", "9c73"
\WNT40\system32\drivers\hidparse.sys = "hidparse.sys", "6230"
\WNT40\system32\drivers\hidclass.sys = "hidclass.sys", "13b9c"
\WNT40\system32\drivers\usbhub.sys = "usbhub.sys", "b54b"
\WNT40\system32\drivers\intelide.sys = "intelide.sys", "4ae2"
\WNT40\system32\drivers\pci.sys = "pci.sys", "14ec5"
\WNT40\system32\drivers\isapnp.sys = "isapnp.sys", "12889"
\WNT40\system32\drivers\aic78xx.sys = "aic78xx.sys", "1ce69"
\WNT40\system32\drivers\i8042prt.sys = "i8042prt.sys", "c5b9"
```



NOTE There are typically about 3000 lines in a setup.log file. If your file is considerably shorter, or does not start as the above example shows, make sure you are changing the correct file!

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

It will be necessary to scan the system registry to ensure that there are no hard-coded references to the installation directory. I found that there were almost two thousand hard-coded references to the installation directory in a well-used installation. Each of these references would have to be manually changed. Follow these steps to determine all hard-coded references to the installation directory for Windows 2000:

1. Using RegEdit, export the entire registry to a file called orig.reg. To do so, select My Computer in the registry tree display.
2. Use a text editor's Search and Replace commands to change all occurrences of the original installation directory name to the new directory name.
3. Reintegrate your edited registry into the original registry; either double-click the exported registry in Explorer or issue the command **START orig.reg** at a command prompt.



NOTE This process is somewhat complex. There is a good chance that when you've finished, the system will not work correctly. Always make sure you have a good backup for restoring in case the change fails.

Upgrade Blues

Windows 2000 comes in two flavors: upgrade and full installation. You can usually get an upgrade for an existing product at a considerable discount over the cost of an entire new product license. Generally, the product is identical in both versions, but in the upgrade version, the Setup program will confirm that you actually have the original product.

The test to see if there is an original product to upgrade is relatively simple, but not flawless. The upgrade version of Windows 2000 will check the hard drive for a qualifying version of Windows. If Setup finds no prior installation of Windows, it will prompt you to insert a disk for the original product to prove you have an upgradeable product.

One problem comes about when you install the Windows 2000 upgrade on a system and you later need to reinstall a new copy of Windows 2000 in the same directory. It is possible that the Windows 2000 upgrade setup program won't work correctly, because it may think that you don't have a product that is included in the upgrade offer when it only finds Windows 2000 on the drive.



NOTE If you are installing a second copy of Windows 2000, the upgrade program will work. It only fails when reinstalling over the original installation.

There is a quick workaround for this problem:

1. Edit the registry subkey HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion. Change the value for CurrentVersion to 4.0.
2. Edit the registry subkey HKEY_LOCAL_MACHINE\System\Setup. Check the value entry SystemSetupInProgress. If necessary, reset its value to 0.
3. Edit the registry subkey HKEY_LOCAL_MACHINE\System\Setup. Check the value entry UpgradeInProgress. If necessary, reset its value to 0. If this value does not exist, don't worry about it; it is not necessary to add it.
4. If Windows has an installed service pack, it would be a very wise move to remove it before reinstalling Windows 2000. Otherwise, problems with the TCP/IP drivers may result in system instability. After reinstalling Windows 2000, reinstall the last service pack. Remember, service packs are cumulative, so you only need to install the highest numbered service pack. Finally, reinstall any hotfixes that were applied to the original system.
5. If you have RAS (Remote Access Service) on this computer, it is imperative to uninstall the service packs. (Service packs can substantially change RAS.) However, some users are unable to remove the service packs without breaking other critical parts of Windows 2000. In that case, restore the file %systemroot%\System32\drivers\tcpip.sys from the original distribution CD-ROM or from the service pack uninstall directory %systemroot%\\$NtServicePackUninstall\$.



NOTE To recover a file from the distribution CD-ROM, you must use the `expand` command from a command prompt. Typing `expand /?` gives you more information on using `expand`.

Where Was Windows 2000 Installed From?

Many of us change the drive letters assigned to the CD-ROM drives after the Windows installation is completed. It is a simple process and helps provide order in the system, especially if you are like me and add or remove drives frequently.

On my computers, I assign all CD-ROM drives to drive letters ranging from S: to Z:. There are four servers with between one and three CD-ROM drives each. Shares have the same drive letters on each networked computer. This way, a reference to S: on any computer on the network will always access the same CD-ROM drive and usually the same CD-ROM, too.

Reassigning the CD-ROM drive letters makes the system more manageable, but there is one problem. Every time you want to make a setup change to Windows 2000 and the Windows 2000 Setup or Configuration programs need to access the original Windows 2000 CD-ROM, the prompt will be for the Windows 2000 installation CD-ROM drive letter used. This drive letter will be different from the new, reassigned CD-ROM drive letter.

The location of the original installation source CD-ROM is stored in the registry subkey HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion in the value entry SourcePath. In addition, check the registry subkey HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Setup, in the value entries SourcePath and Installation Sources. The value entry Installation Sources is a binary value, so edit this one with caution. Change both instances of SourcePath to X:\I386, where X is the CD-ROM drive letter.

I'm Full, Burp

Windows 2000 will give a warning when the amount of free space on the drive falls to less than 10 percent. This percentage works well with smaller 1 or 2GB hard drives, but when the drive is large (20GB or more), the amount of free space can be several gigabytes when the warning is given.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

Search Tips

Advanced Search

PUBLICATION LOOKUP

JUMP TO TOPIC



Desktop Library

[Click to access!](#)**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Previous](#) [Table of Contents](#) [Next](#)

To fix this problem, it is possible to alter the percentage-free parameter, changing it from 10 percent to a more reasonable value. Edit the key
 HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters. Add a new
 REG_DWORD data value named DiskSpaceThreshold. Edit this data value and set its value to the percentage of free space at which you want the warning given. For example, set the value to 5 to give a warning when there is less than 5 percent free space remaining.



NOTE DiskSpaceThreshold will affect all drives. Consider the effect when your system has a mix of small and large drives.

Why Is Windows Asking for a Disk in the Drive?

From time to time, we get into a situation with Windows in which there is no disk or CD-ROM in the disk drive. This might happen when we start an application or a service or at some other time. After checking whether the drive specified is missing from the path statement, check something less obvious. Check HKEY_LOCAL_MACHINE\System\Setup. If it contains a value entry named WinnPath, delete this entry and restart Windows.



NOTE How does a CD-ROM or diskette drive get into the path? Most often, this happens due to either a user error or an application installation that has gone awry. Some applications allow execution from the CD-ROM drive, but don't realize that the application, or a disk, isn't always going to be available in the drive. If inserting any disk satisfies the message from Windows that no disk is in the drive, the message is not significant, and you should try the fix just discussed.

Removing Context Menu Items

It is easy to remove both the Map Network Drive and Disconnect Network Drive selections in the Explorer context menu (and the Tools menu). You might want to do this to make the context menus simpler and easier to use.

A simple change to the registry tells Explorer not to display either of these entries. Here's how:

In the registry key HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer, change (or add, if it doesn't exist) the following entry:

Value entry: NoNetConnectDisconnect

Type: REG_DWORD
Typical value: 1

This registry entry supports two values. When the value is 1, then the Map Network Drive and Disconnect Network Drive menu selections are available. When the value is 0, the Map Network Drive and Disconnect Network Drive menu selections are not available. The policy editing tools, as described in Chapter 5, are able to make this change.

Using More Than Two IDE Controllers

Most computers now come with two built-in IDE controllers. The hardware will usually map one controller to the PCI bus and one to the IDE bus. (The PCI bus IDE controller may exhibit better performance.) Windows 2000 is able to access both IDE controllers, if desired, without any modifications.



WARNING These techniques have not been well tested with Windows 2000. If you need to attempt these procedures, back up critical data before proceeding!

Several configurations are possible with the two IDE controllers. One configuration is to have four hard drives. Today, IDE drives are available in sizes that rival SCSI drives. (My server, DORA, has an 18GB IDE hard drive, in addition to the SCSI drives.) You can create a very reasonable configuration with as much as 75GB (or more, depending on when you read this) of hard disk space using all four IDE drives.





Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

Each IDE controller is numbered. The primary IDE controller is numbered 0 and the secondary IDE controller is numbered 1. An added third IDE controller would be numbered 2, a fourth would be numbered 3, and so forth. Keep this concept in mind as you go about adding a third or fourth (or fifth...) IDE controller:

1. In RegEdit or RegEdt32, open the subkey HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Atdisk and add a new subkey named PARAMETERS.
2. Open this newly created PARAMETERS key and create a subkey named with the number for the added controller. For example, create a subkey named 2 if you are adding a third new IDE controller, or 3 for a fourth new IDE controller.
3. Open the subkey that you named in step 2 and create the following three data values:

Value Entry	Type	Typical Value
BaseAddress	REG_DWORD	Use the physical address of the IDE controller's data register. Configure the controller so that this address does not conflict with any existing IDE controllers or other installed devices.
DriveControl	REG_DWORD	Use the physical address of the IDE controller's drive control register. Configure the controller so that this address does not conflict with any existing IDE controllers or any other installed devices. Typically, this address is at BaseAddress + 0xE.
Interrupt	REG_DWORD	Use the IDE controller's IRQ (interrupt request) address. Configure the controller so that this address does not conflict with any existing IDE controllers or any other installed devices.

Saving Share Information

Many people use Windows 2000 shares, which may be lost when making a clean installation. We clean install for a number of reasons. Perhaps it's because we find our system unstable. And since it is not possible to determine the starting date of the problems, we cannot depend on backups. In addition, sometimes the system hardware configuration changes (for example, a new server or disk assembly is installed), which necessitates a clean installation.

For servers with a large number of shares, reentering each share manually can be a time-consuming process. The following registry trick is easier.



NOTE Before following these steps, realize that the process described may overwrite any existing shares.

1. Start RegEdit.
2. Open the subkey HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Shares.
3. Select Registry ▾ Export Registry File.
4. Enter a filename for saving the Shares subkey. Preferably, you should save this file to a floppy disk or another nonvolatile location. Click the Selected Branch button, and the saved branch should read HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Shares.
5. After reinstalling Windows 2000, insert the diskette with the file saved in step 4 and type the command **START filename.REG** at the command prompt. Use the filename you saved in step 4.
6. Check to ensure Windows 2000 has properly incorporated these shares into the registry.



NOTE Macintosh shared volumes will not be saved using these techniques.

When using this technique at upgrade time, check that the new version of Windows saves share information in the same location as the previous version; otherwise the changes won't have the desired effect. At the time of this book's writing, this technique works with both Windows NT 4 and Windows 2000.

Other Hardware Support Entries

There are thousands of other hardware support entries. The key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Ports contains default information for each port on the system. A typical system has the following ports:

COM1, COM2, COM3, COM4: These are communications (serial) ports used with mice, modems, and other serial devices.

FILE: These ports, typically used for printer driver output when there is no matching physical device attached to the computer, redirect output to a disk file.

LPT1, LPT2, LPT3, LPT4: These are printer (parallel) ports used with printers, some other devices, special modems, etc.

Ne00, Ne01: These are the ports used with printers directly connected to the network. Some higher-performance printers include a built-in Ethernet port.

I've gathered a few for this chapter and grouped them by major components—serial ports, printer ports, and so on.

Serial Ports



In Windows 2000, the serial ports are contained in the subkey HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Ports. In addition, the key HKEY_LOCAL_MACHINE\System\ControlSet001\Enum\Root*PNP0501 is used to hold port information. Neither of these keys exists in previous versions of Windows. As Windows 2000 spreads the information throughout the registry, manual modification is more difficult!

Each communications port entry consists of a REG_SZ string containing the port's speed, parity, bits, and stop bits. The default values for these entries are given below:

Value Entry	Value
COM1:	"9600,n,8,1"
COM2:	"9600,n,8,1"
COM3:	"9600,n,8,1"
COM4:	"9600,n,8,1"

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

The PnP (Plug-and-Play) subkeys contain settings for virtually all hardware installed on a Windows 2000 system. For example, *PNP0501 is the identifier for serial ports (also see Appendix E, “Plug-and-Play Identifiers”).

The key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\Root*PNP0501 on my machine, DORA, consists of two subkeys, one for each of the two serial ports installed on the system. Inside this key are additional subkeys, one for each port. The names of these subkeys may vary, but could be PnPBIOS_14 and PnPBIOS_11. Inside these subkeys are three additional subkeys, Control, Device Parameters, and LogConf.

In the Control subkey, you’ll find a number of values that deal with device setup—for example, whether firmware (software on the device) is used or not and how resources are allocated. Modifying this subkey can be very tricky.

In the subkey Device Parameters, you’ll find the values shown below (a given Windows 2000 installation may not have all the values described):

Value Entry	Value
PortName	A string such as COM1
PollingPeriod	A REG_DWORD with a default value of 0
ForceFifoEnable	A REG_DWORD with a value of 0 if the FIFO buffers are not used.
RxFIFO	A REG_DWORD with the receive FIFO value set by the user. The value will range from 1 to 14.
TxFIFO	A REG_DWORD with the transmit FIFO value set by the user. The value will range from 1 to 16.

Windows 2000 seriously limits the user’s ability to alter items such as IRQ addresses, I/O addresses, and DMA channels. This limitation is necessary to comply with the PnP requirements.

These settings are contained in a REG_RESOURCE_LIST value in the registry key HKEY_LOCAL_MACHINE\Hardware\ResourceMap\PnP Manager\PnPManager. Every port (actually every device that is installed on the computer) has a REG_RESOURCE_LIST value. The device driver uses the REG_RESOURCE_LIST (a device driver resource list) to “find” the hardware.

Printer (Parallel) Ports



HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Parallel contains parameters that help control the system's utilization of the basic parallel or printer ports. Many Windows 2000 computers use one of these parallel ports connected to a standard printer. An Intel x86 system may have between zero and two parallel ports, although most systems have only one parallel, or printer, port.

Ports on virtually all Windows 2000 systems utilize a standard printer driver chip as the hardware interface. This chip is configurable in the BIOS, and may allow either one-way (to the printer) or two-way (both to the printer and from the printer to the computer) communications. Additionally, printer port configurations allow for high-speed communications. High-speed communications are important when you are printing complex images (bitmaps, for example) and transferring a large amount of data between the computer and the printer. Some printers also have a scanning mode. These printers require both high-speed printer ports and ports that support bidirectional data transfers. Many of the parallel port's settings are configurable with the Control Panel's Ports applet.

Other Software Configuration Entries

Some settings affect the user interface and the system equally. Where do we place these entries? WinLogon is a section in the registry that holds settings used for things like the users log.

Password Expires in *n* Days

Windows displays a password expiration message to the user a certain number of days before the user's password expires. Configure this message in Windows 2000 at the client by following these steps:

1. Start RegEdit or RegEdt32.
2. Open the subkey HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WinLogon.
3. Add this value entry, or modify it if it already exists:

Value entry:	PasswordExpiryWarning
Type:	REG_DWORD
Typical value:	14

This entry holds the number of days Windows will display the "password expires" warning.

Domain Refresh Interval

Windows refreshes the domain list whenever the workstation is unlocked, providing that the workstation has been locked for more than 120 seconds. (A user can lock their workstation by pressing Ctrl+Alt+Del.) On many networks, refreshing the domain list can result in a significant delay before the user regains control of their system.

This problem may be somewhat alleviated by increasing the minimum "locked time" setting (in essence, gambling that the domain list won't have changed during that time), which you do by modifying the following value entry:

Value entry:	DcacheMinInterval
Type:	REG_DWORD
Typical value:	120

This entry contains the number of seconds that the system must have been locked before the registry will force the system to refresh the domain list. Values range from a minimum of 120 seconds, the default, to a maximum of 86,400 seconds.

Hints and Kinks from the Experts

Here are some hints and kinks from our experts on Windows 2000 networking and system entries.

Where Do DHCP Clients Store Lease Information?

DHCP (Dynamic Host Configuration Protocol) clients store information locally so that they can attempt to

lease the same IP address and still function if the DHCP server is down. A Windows NT client stores DHCP information in the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\<Network Adapter>\Parameters\Tcpip.

A Windows 95 or Windows 98 client stores DHCP information in the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\DHCP\DHCPInfo00.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Windows for Workgroups with TCP/IP-32 stores the DHCP-related information in the DHCP.bin file in the <WINDOWS_SYSTEM_ROOT> subdirectory. This file is in binary format.

LAN Manager 3 and LAN Manager for MS-DOS 2.2c clients both store the local DHCP-related information in the DHCP.prm file in the <NETWORK_ROOT> directory. This file is in a binary format.

(Courtesy of Jerold Schulman.)

Does your 100MB Ethernet TCP/IP Network Perform Poorly?

If your 100MB Ethernet TCP/IP network performs poorly, it could be due to ACK collisions. You can contact your NIC manufacturer to see if there is a way to increase the interframe gap.

The Intel EtherExpress 100B adapters have the following registry parameter:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\{e100bx}\Parameters. The x represents the number of your interface card.

Add the following entry to enable an adaptive algorithm:

Value entry:	Adaptive_ifs
Type:	REG_DWORD
Value:	1

Setting the value to 0 will disable it. A value from 2 to 200 sets a predefined interframe gap if you want to measure collisions at 20, 40, 60, and so on; be sure to pick a value with a low collision rate and good performance.

If you have a different NIC, edit HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters instead, adding the following:

Value entry:	TcpWindowSize
Type:	REG_DWORD
Value:	2920 (decimal)



TIP This tip is for 100MB Ethernet TCP/IP networks only. Using this parameter on a 10MB network, WAN, FDDI, Token Ring, or anything else will impact performance.

(Courtesy of Jerold Schulman.)



Click to access!

Learn Common Registry Entries for Networking Programs

The key HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Network\Program name contains configuration data that is common for programs that view and maintain the network. Program names include Browser Monitor, Event Viewer, Server Manager, User Manager, and so on. Some of the common entries in the registry for networking programs are shown below. All the entries are type REG_SZ.

Value	Default	Description
SaveSettings	1	1 = Options are saved. The user's choices are restored when the user starts the program again. 0 = Options are not saved. This entry needs to be set to 1 for most other entries to work.
Confirmation	1	Is confirmation for deletion or changes required? 1 = Yes 0 = No
FontFaceName	(None)	This entry sets the font used if different from the default.
FontHeight	0	0 = Use the default point size for the font
FontItalic	0	0 = Not italic 1 = Italic
FontWeight	0	0 = Use the default font weight 400 = Standard weight 700 = Bold 900 = Heavy
SortOrde	(Depends on program)	For Event Viewer: 0 = Oldest first 1 = Newest first (default) For User Manager: 0 = Sort by full name 1 = Sort by username (default)
Window PosX PosY SizX SizY	xyxy0	This entry sets the four pixel coordinates that define the size and position of the window. When not minimized, 0 follows these values. When minimized, 1 follows these values.

Reduce Network Delay

You may encounter delays, when TCP/IP network activity is light, with the default request buffer size of 4356 (decimal). The range of this parameter is 512 to 65536 bytes. Testing has shown that in most standard Ethernet environments, 14596 (decimal) is a better choice, if the memory is available. To reduce network delay, perform the following steps:

1. Edit HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters.
2. Add the following:
Value entry: SizReqBuf
Type: REG_DWORD
3. Restart the computer.

(Courtesy of Peter D. Hipson.)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

CHAPTER 14

Microsoft Office Entries

FEATURING:

<i>Microsoft Office shared components</i>	345
<i>Changes made by Microsoft Office Setup</i>	346
<i>The .reg files</i>	347
<i>Microsoft Office system configuration information</i>	348
<i>Microsoft Office user configuration information</i>	358
<i>Using the registry from Microsoft Office applications</i>	363
<i>Hints and kinks from the experts</i>	368

This chapter is a bit different from some of the others. Although I could just list every registry entry for Microsoft Office, the chapter would quickly become boring, and it would be excessively long. Instead, I'm going to focus on particular Office-related topics that affect the registry:

- Microsoft Office shared components and how they are interlinked to form a cohesive product. Microsoft Office consists of many programs that interact with each other. In addition, there are some optional Office programs, installed only if the user wants to use them. And to confuse the issue even more, some programs are automatically installed on "first use"!
- Changes made by the Microsoft Office Setup program to the registry, which can be extensive in a typical system. Even one Microsoft Office program can result in hundreds of registry changes.
- The .reg files that come with Microsoft Office. These files are used to update the registry, making changes that are standardized for all Microsoft Office installations.

- How to modify Microsoft Office configuration information. Each configuration of Microsoft Office is unique. During the installation process, a user may select options and components to install.
- How to customize and copy user information between users. Many organizations choose to install Microsoft Office on an organization-wide basis. Usually this is done by installing one copy, then using cloning techniques to duplicate the installation across a network.
- Programming the registry using Microsoft Office's VBA (Visual Basic for Applications.) VBA offers a lot of power to the typical Office user. For example, it is capable of much of the power that Visual Basic provides. Moreover, all VBA applications are capable of registry interaction.



NOTE The information in this chapter refers to Microsoft Office 2000 Professional Edition, the release that is current at the time of this book's writing. Later versions of Office will be released every year or so, perhaps even in 2000. Hopefully, much of this chapter's information will be usable with these future releases.

This chapter tells you how to repair your Office 2000 registry entries. If Office 2000 is not running correctly, the problem may be more involved than just a damaged or missing registry entry. For example, it is entirely possible that *files* are either corrupted or missing. For this reason, don't look at this chapter as being a save-all. Rather, try fixing the registry, but if that doesn't have the desired effect, try reinstalling the malfunctioning Office components.



WARNING Never, under any circumstances, install any beta editions of Office 2000 on Windows 2000. This is virtually guaranteed to cause your system to crash and die horribly, with the only option left open to you a clean reinstallation of Windows 2000.

There have been some compatibility issues with Office 2000 and Windows 2000. These include problems with the Windows 2000 implementation of Internet Explorer. (Internet Explorer sometimes seems to be less compatible with Office 2000 than Microsoft would like it to be.)

I'd never suggest that you try to restore the Office products by first restoring a backup of the files and then adding registry entries. Though this might work, you could expect the need for other subtle things, such as adding critical shared .dll files to the Windows 2000 SYSTEM32 directory.

One of two processes performs most of Microsoft Office's registry modifications. The first process is the Setup program. This program will add, subtract, and otherwise modify a number of registry entries, all of which are critical to the running of Microsoft Office. The second process is a group of registry modification files, with the extension of .reg. These files are contained in directories on the Microsoft Office distribution CD-ROM.

Repairs to Microsoft Office are relatively easy. On the one hand, some components reinstall well. Reinstalling Microsoft Word, on the other hand, may overwrite your normal.dot file. We suggest, therefore, that you save or otherwise back up user-modified Microsoft Office files, such as the document templates like normal.dot, before reinstalling Microsoft Office.



NOTE Notice that there are components listed in this registry section that are *not* part of the basic Microsoft Office package, such as a listing for Microsoft Publisher.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

Microsoft Office Shared Components

Microsoft Office consists of a number of components. We all know about the big ones—Word, Excel, Access, and PowerPoint. Nevertheless, a number of small helper applications that we don't always see or know about are also included with Office. For instance, there is Microsoft Chart, a charting program used with Word, Excel, and Access. The Microsoft Office shared components are listed here:

Equation Editor	Used to create visually appealing equations.
WordArt	Used to embed simple drawings into documents. Word 2000 integrates the WordArt capability, although the original WordArt application, if previously installed, is retained for compatibility.
Graph	A basic graphing tool; used to graphically display data.
Organization Chart	Used for the drawing and maintenance of basic organizational charts.
Media Player	An embeddable media player; most useful for embedding video clips into PowerPoint presentations.
ClipArt Gallery	A collection of clip art from Microsoft PowerPoint that may be used to improve the visual appeal of documents.
Draw	A basic drawing package that can be used to create effects such as 3-D. In Word 2000, drawing is integrated into Word itself.

With the possible exception of the ClipArt Gallery, which consists mostly of images, the shared components are usually ActiveX embeddable components. Embedding uses CLSIDs inserted into the registry by the Microsoft Office installation process. Appendix F, "Office 2000 CLSIDs," lists the significant Microsoft Office CLSIDs.



NOTE Office does not install all of the CLSIDs listed in Appendix F. Most of us install only parts of Microsoft Office, and therefore only some of the CLSIDs will be present in the registry. A missing CLSID doesn't signify an error or problem in itself.

Changes Made by Microsoft Office Setup

Unlike earlier versions of Office, Office 2000 uses the newer Microsoft installer named MSI, or Microsoft Installer. The Microsoft Office Setup program first adds and sometimes removes a number of registry entries; actually, a full installation process could modify over a thousand registry entries. (Now you see

why I don't just list them all!) Each entry modified, deleted, or added by the Setup program is modified because the entry is based on information specific to the current installation. For example, Setup must handle the entry that has the installation directory, which the user may change at setup time. The .reg files cannot do this, because the .reg file technique cannot take into account user preferences.

The main controlling file for the Microsoft Office Setup program is DATA1.msi. It is located in the root directory of the Microsoft Office CD-ROM. This file is binary so that users may not alter it in any way. There is also a setup initialization file, SETUP.ini, which Setup uses to initialize itself. The .msi file contains entries that control which files are copied and where; which registry entries to delete, add, and modify; and everything else that must be done to install Microsoft Office.

The DATA1.msi file includes:

- 19 registry searches
- 53 retrievals of path names
- 1,418 additions to the registry
- 2 checks of the registry
- 28 checks for registry equality
- 9 registrations of type libraries
- 29 self-registrations
- 21 retrievals of the Windows path
- 73 creations of strings of REG_SZ type
- 6 creations of REG_DWORD type objects
- 35 registry entry removals
- 9 registry tree deletions
-



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Microsoft Office System Configuration Information

Microsoft Office 2000 stores information about common configuration settings in the registry key at HKEY_LOCAL_MACHINE\Software\Microsoft\Office (I'll refer to this location as the "system keys"). This key contains a subkey called 9.0 (for the version number of Microsoft Office 2000). The next version of Office will probably be stored in a subkey named 10.0. There is also a subkey called 8.0 with a single object: Outlook, which contains information on registration and setup.

Non-common configuration information, specific to a given user, is stored in the user's configuration at HKEY_CURRENT_USER\Software\Microsoft\Office (I'll refer to this location as the "user keys"). These settings are modifiable by users and are kept separate from one another, as each user will have their own configuration and defaults.

The structure of these two locations for Microsoft Office is virtually identical. Under Microsoft is the Office subkey. Under Office are one or more subkeys for each version of Microsoft Office installed. For example, many installations have a subkey, 8.0, for Office 97, and a subkey, 9.0, for Microsoft Office 2000. Under the version subkey are one or more product keys, and a few support keys.

Don't be surprised if there are subkeys for Microsoft Office components that aren't installed. Some components set items for other components regardless of whether they're installed or not. For example, on my computer, in the user keys, I have the following major subkeys. (Throughout this chapter, assume that an unqualified key is a user key, as I will fully qualify, or annotate, any system keys.)

Access: Microsoft Access, a full-featured desktop database system.

Common: Items common to more than one Microsoft Office component.

Excel: Microsoft Excel, a spreadsheet program.

Outlook: Microsoft's advanced e-mail client.

Registration: Microsoft's Office 2000 user-registration system. The first time a user installs Microsoft Office 2000, the registration program will contact Microsoft, relaying user and product statistics.

Word: Microsoft's well-known word processor, used to write this book.

Each of these keys contains more keys and information. For example, I don't actually have Access installed, but Microsoft Office Setup included the Access subkey, regardless. The Access subkey has entries for the following items:

- Clipboard Formats
- InstallRoot



JUMP TO TOPIC

- Jet
- Menu Add-Ins
- Options
- Report Formats
- Speller
- Wizards

I'll describe each of these subkeys in detail in the next section.

In the system keys, we find the subkeys listed below. Generally, at this level the system keys closely parallel the user keys.

Access: Microsoft Access, a full-featured desktop database system

Common: Items common to more than one Microsoft Office component

Excel: Microsoft Excel, a spreadsheet program

Outlook: Microsoft's advanced e-mail client

Shortcut Bar: The Office 2000 shortcut bar configuration information

Word: Microsoft's well-known word processor, used to write this book (didn't I already say that?)

Like the user keys, each of the system keys contains more keys and information. For example, the Access subkey has entries for the following items:

CustomizableAlerts: Access gives an alert whenever a certain event occurs.

Settings: Access uses an MRU list to allow the user to quickly open a previous file. The MRU file list is contained in this subkey.

The Access Entries

The Microsoft Access database program is Microsoft's main entry into desktop database systems. Though Microsoft also offers a product called FoxPro (and more appropriately, Visual FoxPro), FoxPro is not part of the Office suit. We'll look into some subkeys of the Access key in the following sections.

Clipboard Formats

The entries in the Clipboard Formats subkey describe special formats that Microsoft Access is able to process. These entries include formats and descriptive information on the handler for each format. A typical entry might be:

Value entry: HTML (*.html)

Type: REG_SZ

Typical value: soa900.dll,30,html,HTML,HTML(*.html),1

This entry indicates that Access will use soa900.dll to read this type of data. The entry also provides information to Access about requirements necessary to invoke the code in the .dll file. A typical installation might define six or more Clipboard formats.

InstallRoot

The InstallRoot subkey has a single data value, named Path. Path, a REG_SZ object, contains the fully qualified path to the Microsoft Access program files. A typical entry might be C:\Program Files\Microsoft Office\Office, where C: is the drive that Microsoft Office was installed on.

Jet

Microsoft Jet is the engine that Microsoft Access uses to access the actual database files. Microsoft has opened the interface to the Jet engine to other application software, allowing developers to create programs. These programs are able to create, read, and write Access-compatible databases.

Microsoft Jet is a complex, high-performance database engine. There are several additional interfaces in the Jet engine, allowing programming interoperability between Access-compatible software and other database systems, including:

- SQL Database
- Excel 3.0
- Excel 4.0
- Excel 5.0
- Excel 8.0
- Exchange 4.0
- HTML Import
- Lotus WK1
- Lotus WK3
- Lotus WK4
- Microsoft Access
- Microsoft Active Server Pages (ASP)
- Microsoft IIS (Internet Information Server)
- Microsoft Access Data Access Page
- Microsoft IIS
- Outlook 9.0
- Rich Text Format (RTF)
- Snapshot Format
- Text
- Word for Windows Merge

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Menu Add-Ins

Access may be expanded or enhanced using menu add-in programs. Some menu add-ins supported by Microsoft Access include:

- Add-In Manager
- Database Splitter
- Linked Table Manager
- Switchboard Manager

A registry subkey defines each add-in in the Access\Menu Add-Ins subkey.

Options

The states of certain Access options are stored in the Options subkey. These options can vary greatly from installation to installation. An example value entry is AttachIndexWarning.

Report Formats

The Report Formats subkey stores the formats that Access is able to write reports in. Formats typically supported by most installations include:

- HTML
- Microsoft Excel
- MS-DOS Text
- Rich Text Format (RTF)
- Snapshot Format

Speller

Like the other members of the Microsoft Office family, Access supports a spell-checking mode. Spell checking is important if you or other users are as fumble-fingered as I am. Without a spelling checker, this book would be unreadable, and the editor, who had to work hard enough anyway, would have probably done nasty things to me.

Settings for spell checking include:

- Custom Dictionary


Desktop Library
[Click to access!](#)

- Ignore All Caps
- Ignore Mixed Digits
- Language ID (information about the current language being used)
- Suggest Always
- Suggest Main Dictionary Only

Wizards

Access uses wizards to perform a number of the more complex setup and processing tasks. Wizards allow inexperienced users to quickly become proficient and to get the maximum amount of use from Access without spending a great deal of time learning the product.

Each wizard has a unique set of objects. The main categories of wizards are:

- Control wizards
- Data Access Page wizards
- Form wizards
- Preferences
- Property wizards
- Query wizards
- Report wizards
- Table wizards

The Common Entries

In Microsoft Office, many of the applications share parts or components. For example, that funny and entertaining Office Assistant (see Figure 14.1) allows a user to get help quickly and easily in any application. Entries for these components are found in the Common subkey. Other common information—such as Default Save, New, Templates, and so on—is stored in this section as well.



FIGURE 14.1 The Office Assistant is always there to provide help. A single click displays the Help balloon, as shown.

Microsoft Office shares the following items between multiple Office applications:

Assistant: This subkey manages the configuration and customization of the Microsoft Office Assistant. Me, I like that cute cat figure. Several settings for the assistant are available, such as whether the assistant works with wizards.

General: This subkey holds a number of miscellaneous objects. General configuration items include installation information and setup data.

HelpViewer: This subkey is where you define the help viewer defaults for small and large help panes and pane position.

Internet: This subkey stores Internet locations, such as where components are downloaded from.

LanguageResources: This subkey stores the language ID (U.S. English is 409).



NOTE Wait a minute... Isn't U.S. English 1033? What's this 409 number? OK, use the Windows calculator program, and convert 0x409 from hexadecimal to decimal. If you did it right, you got 1033. That's the secret; the language ID in this object is in hexadecimal!

Migration: This object holds information about migration from one version to a newer version. Applications in this subkey include Excel, Office, Outlook, and Word.

Open Find: This object contains subkeys for each product. Within each subkey are settings specific to the application.

Toolbars: This subkey stores settings such as adaptive menus, whether menus auto-expand, etc.

UserInfo: This subkey holds the user's name, company, ID, initials, and user information.

InstallRoot

InstallRoot contains the Microsoft Office installation directory.

LV

When used, LV describes the installation type, product ID, product name, etc.

The Excel Entries

The Excel key, for the Microsoft Excel spreadsheet program, contains a couple of subkeys: Options and Recent Files. The Options subkey contains various optional values for Excel. The Recent Files subkey contains information about files that were recently opened.

The Outlook Entries

The Outlook key contains the settings for the Outlook e-mail client, available as part of the Microsoft Office package. I'll admit a preference for Outlook 2000. It's a tool that I use every day for e-mail, calendar, and contact management tasks. It offers substantial improvements over earlier versions of Outlook. I recommend that if you are considering Outlook, you get the latest version from Microsoft.

The Outlook subkey contains information on the following Outlook 2000 functions:

Dataviz: This subkey holds a flag indicating whether public folders are hidden or not.

InstallRoot: This object holds the fully qualified name of the directory that Outlook 2000 has been installed in.

NameSpaces: A namespace represents a data source, such as the MAPI message store, or file and directory names. This subkey holds namespace information for Outlook.

OMI: This subkey holds the Outlook Internet e-mail system configuration.

Operations: This subkey contains configurations of various operations that Outlook will perform, such as file import, file export, data link export, VCard import, Accounts import, Calendar import, and Eudora import.

SchedulingInformation: This subkey contains options and configuration information for the Outlook Schedule functionality.

SearchTypes: This is the table of CLSID entries for each search type.

Setup: This subkey lists options chosen during setup.

UpgradePath: This is the path to be used for upgrade.

Outlook is one of the most complex components of Microsoft Office. It can be difficult to configure; but once it's set up, it provides flexibility and power, making it a valuable tool for any busy computer user.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)**The PowerPoint Entries**

Microsoft PowerPoint is a tool used to create presentations for display on media, such as video screens, printed handouts, and slides. PowerPoint is capable of such tricks as animation, sounds, and special effects, making it a good presentation and training tool. Many users are familiar with PowerPoint, but most do not use this program to its fullest. Look at the supported features discussed in this section for some ideas of what can be done with this versatile program.

The following list contains subkeys that are specific to PowerPoint:

Addins: Microsoft PowerPoint supports a number of add-in product functionalities. This object contains information about add-ins, including auto-content and PowerPoint tools.

Answer Wizard: This subkey contains settings used with the supplementary help system that assists users in searching for additional help on problems they are having.

AutoContent Wizard: This subkey assists users in designing a presentation and developing content.

DLL Addins: This subkey contains add-in .dll files used by PowerPoint.

Document Routing: This subkey contains a flag to tell Microsoft PowerPoint whether or not to track status.

Export Modules: This subkey contains the names of modules used to export presentations to other formats, such as for an offline publishing or printing system.

InstallRoot: This subkey holds the root directory that Microsoft PowerPoint has been installed in.

OLE Play Options: This object contains any OLE multimedia support that is included. Typically, items in this subkey include sound and video support.

PPCentral: This object consists of basic Microsoft PowerPoint options, many of which are configurable by the user.

Sound: This subkey is concerned with sound formats, such as WAV, that are supported by Microsoft PowerPoint.

Sound Effects: This object contains information about sound effects. The user can use these included sound effects (like Typewriter, Whoosh, Laser, Camera, and Drive By) in a presentation.

Translators: This subkey contains information about import and export support for other versions of Microsoft PowerPoint. The tools include Export to Microsoft PowerPoint 7, and Import to Microsoft PowerPoint 7 and Microsoft PowerPoint 4.

ValuPack: This object specifies the location of the Microsoft Office ValuPack directory. Run Valupk8.hlp to find out more about the ValuPack.

Viewer: This object specifies the location for the Microsoft PowerPoint viewer, a stand-alone Microsoft PowerPoint display program.

The Publisher Entries

Microsoft Publisher is a midrange page-layout program that is well integrated with other Microsoft Office products. Using Microsoft Publisher is easy; the application allows you to create professional documents that may be printed locally or sent to a printer/typesetter for duplication. The following list contains subkeys specific to Microsoft Publisher:

ColorSchemes: This subkey contains the definition of the Microsoft Publisher color schemes. Microsoft Publisher allows users to switch color schemes at any point using a four-color palette.

Envelopes: This subkey contains information on any envelopes defined.

HTML: This subkey contains the filter to process HTML documents. Microsoft Publisher will publish in HTML if desired, allowing you to create Web pages from other existing documents with a minimum of effort.

HTMLCharacterEncodings: This subkey holds character encoding for foreign languages.

Mail Merge: This subkey contains information on mail merge. Using the Microsoft Access Jet database engine, Microsoft Publisher is able to merge database information, creating custom documents as necessary.

Page Size: This subkey defines custom page sizes, such as those required for business cards.

Printing: This subkey contains information used by Microsoft Publisher to print user documents.

ProPrint: This subkey contains information used by Microsoft Publisher to print using a high-end image-processing system.

PubBackground: This subkey contains the directory path to the Microsoft Publisher backgrounds that can be used with publications.

PubClipart: This subkey contains the directory path to the Microsoft Publisher clip art that can be used with publications.

Recent File List: This subkey holds the Microsoft Publisher MRU file list.

Spelling: This subkey contains spell-checking options.

Version: This subkey is where version information is saved.

WizType: This subkey contains the Microsoft Publisher wizards that are used to create basic publications with a minimum of effort.

The Registration Entries

The Registration subkey contains the Microsoft Office 2000 product ID value.

The Word Entries

The Word subkey contains information about the Microsoft Word installation directory. All of the remaining Word options are set in the user's configuration.

Microsoft Office User Configuration Information

The best way to get to the Office user configuration registry entries is through HKEY_CURRENT_USER\Software\Microsoft\Office\9.0. Although they are also available in the HKEY_USERS hive, accessing them through HKEY_CURRENT_USER instead will ensure that the correct set of entries (those for the currently logged-on user) is always modified.

Some ways to use these keys include:

- Backing up the key and saving it for another user ID. This second user ID could be a different user ID for the same user. This allows a user to recover their entire configuration without changes.
- Modifying a specific user's entries. Maybe the organization name changes—you could rely on users to update their systems, or you could go in and make the necessary changes for them.
- Implementing a specific backup and restore for whatever reason.



NOTE Sometimes we just want to start over and redo our user settings from scratch. So, we uninstall Microsoft Office and do a new clean install. Bang! There are all our old settings back again—we can't seem to get rid of them. The reason for this is simple: Uninstalling doesn't remove these user configuration settings. Uninstalling only removes the system settings. It is actually not even necessary to uninstall to change the user configuration settings; simply delete the user's configuration.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

In the next sections, you'll find some of the common user configuration settings. These may be altered, following standard precautions about backing up, and will only affect the current user. Other users will not see any changes made to the HKEY_CURRENT_USER configuration.

Access User Configurations

Microsoft Access user configurations are stored in subkeys under the Access key. The items found in this key will vary greatly depending on the user's configuration and use of Access.

Binder User Configurations

Any user-specific Binder settings are saved in the Binder subkey.

What's Binder? It is surprising just how many experienced Microsoft Office users don't know what Binder is or how to use it. Microsoft Binder is a program that makes it possible to group all of your documents, spreadsheets, and presentations for a project into a single master document. A typical use for Binder is to create a project proposal and presentation.

Common User Configurations

Items in the Common subkey are shared between more than one of the Microsoft Office applications. For example, the Microsoft Office Assistant is used by all the Microsoft Office products.

Commonly, any user-installed items, such as those from Visual Basic for Applications, will be stored in the Common subkey as well. See "Using the Registry from Microsoft Office Applications," later in this chapter.

Most Common subkeys contain:

Assistant: This subkey contains the settings for the Microsoft Office Assistant, including who the assistant is and other Office Assistant configurations. To change these settings, context-click the Microsoft Office Assistant and select an item to change.

AutoCorrect: AutoCorrect is mostly used in Microsoft Word. It allows a word that is misspelled to be automatically corrected. Users of Excel, PowerPoint, and Access will also find use for this functionality.

Cursors: This subkey is where you can configure cursors displayed by Microsoft Office. I've been unable to find any way other than registry manipulation to change Office's cursor selections.

FileNew: This subkey contains the configurations for the File New dialog box.



Desktop Library
[Click to access!](#)

General: This subkey contains the general settings for all Microsoft Office applications.

Internet: This subkey contains the Internet settings for all Microsoft Office applications.

Open Find: This subkey contains the settings for the Open dialog box.

Toolbars: This subkey contains the toolbar configurations and settings.

UserInfo: This subkey contains information about the current user.

Draw User Configurations

Microsoft Draw is a helper application that is used to edit and draw within Office documents. Primarily an OLE server application, Draw is not designed or intended to be used as a stand-alone application.

Excel User Configurations

The user's entire Excel configuration is contained in the Excel subkey. Keys in this subkey include:

Init Commands: This subkey contains the commands used to initialize Excel.

Init Menus: This subkey contains information used to initialize Excel's menu.

Line Print: This subkey contains Lotus macro line-printing settings.

Microsoft Excel: This subkey contains basic configuration settings.

Recent File List: This subkey contains the list of the most recently used files.

Spell Checker: This subkey contains the options for configuration of the spelling checker.

WK? Settings: This subkey contains the settings for the Lotus open-and-save feature.

Graph User Configurations

Microsoft Graph is a helper application that is used to edit and include simple graphs and tables within Office documents. Primarily an OLE server application, Graph is not designed or intended to be used as a stand-alone application.

Outlook User Configurations

Outlook's configuration settings are contained in this subkey. Settings for the following areas are included:

Appointment: This subkey contains the appointment book configuration information.

AutoNameCheck: This subkey contains the setting that indicates whether to automatically check names in the Send and CC lines of messages.

Categories: This subkey contains the message categories.

Contact: This subkey contains the contact (names) list management configuration and options.

Dataviz: This subkey contains the interface with external data sources such as the PAB (Personal Address Book) and other data sources.

Item Toolbars: This subkey contains the toolbar configurations.

Journal: This subkey contains the Outlook Journal, used to track items.

Journal Entry: This subkey contains the Outlook Journal configuration.

Message: This subkey contains the message box configuration.

Note: This subkey contains the note configuration.

Office Explorer: This subkey contains the configuration of the Office Explorer.

Office Finder: This subkey contains the Office Finder configuration and settings.

OMI Account Manager: This subkey contains the Outlook Internet e-mail system configuration.

Options: This subkey contains various miscellaneous settings.

Printing: This subkey contains the printing options and configuration.

Report: This subkey contains the reporting options and configuration.

Scripting: This subkey contains the scripting driver's CLSID.

Security: This subkey contains the security settings.

Setup: This subkey contains the setup options and settings.

Task: This subkey contains the task options and settings.

Today: This subkey contains the Outlook Today settings.

WAB: This subkey contains the settings for the Windows Address Book.

Wizards: This subkey contains the wizard settings.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**PowerPoint User Configurations**

Microsoft PowerPoint user settings are saved in the PowerPoint subkey.

Publisher User Configurations

Microsoft Publisher user settings are saved in the Publisher subkey. They include:

Preferences: This subkey contains the user preferences and settings.

Tracking Data: This subkey contains the tracking items.

UserInfo: This subkey contains the information specific to the current user, including the following entries:

- OtherOrganization
- Personal
- PrimaryBusiness
- SecondaryBusiness

Query User Configurations

The Query subkey contains information about the Microsoft Query (Msqry32.exe) program, if installed and used. Microsoft Query is useful for peeking at various data sources, such as database files created by any ODBC-compliant application.

Word User Configurations

Microsoft Word user configurations are saved in this subkey. Items saved here include:

Custom Labels: This subkey holds information that is used when printing on labels. Users may create their own custom label to match their label stock.

Data: This subkey includes the Word MRU file list. This list is hidden in another object and is not editable.

Default Save: This subkey contains the default save format.

Help: This subkey contains the Help file information.

List Gallery Presets: This subkey contains binary information about presets for the list gallery.

Options: This subkey holds various Microsoft Word options and settings.

Stationery: This subkey contains information used primarily when Microsoft Word is the e-mail editor.

Text Converters: This subkey contains the filters used to convert documents saved as text files into Word. Entries include:

Import\MSPAB: Filter for importing from the Microsoft Personal Address Book

Import\OUTLOOK: Filter for importing from Microsoft Outlook

Import\SPLUS: Filter for importing from Microsoft Schedule Plus

Wizards: This subkey contains configurations for the various Word wizards.

WordMail: This subkey contains WordMail settings used when Word is used as the e-mail editor.

Using the Registry from Microsoft Office Applications

Okay, in the final section of this chapter, let's figure out how to manipulate the registry from a Microsoft Office application. That's right, if you wanted to, you could write an entire registry editor using Microsoft Word.

Most Microsoft Office users won't have a great need to save items in the registry. However, if you find that you need to save information that must be persistent between sessions, saving this information in the registry can be an excellent method.

The example I'll use here is a publisher's system that authors use to manage their books. This system allows the author to automatically name his or her files and insert pertinent information, such as the book's title and ISBN, and it allows defaults for a vast number of different options that authors and editors like to use.

This system works by having the author enter some information at the beginning of a new project. This information is entered using a simple dialog box that was created with Microsoft Office's Visual Basic for Applications. The information is then written into the registry, in the user's section, under the key HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Common\Publisher. You would use a different name than Publisher, of course.

Figure 14.2 shows the Publisher key and the data values saved in it. These values are modifiable from various places in Word, where the publisher has added functionality.

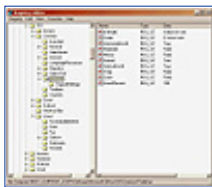


FIGURE 14.2 The Publisher subkey has a lot of information about the book.

The Publisher Options dialog box is used to modify the Publisher options, stored in the Publisher subkey. This dialog box (see Figure 14.3) is a multi-tab dialog box that allows easy data entry.

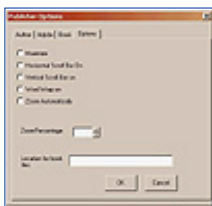


FIGURE 14.3 The Publisher Options dialog box sets and resets options saved in the registry.

I'm going to show some of the code used to create this dialog box here. Yours does not need to be this complex. A simple dialog box without tabs can be very effective in getting a user interface together.

First is the main function that displays the dialog box itself:

```
Sub DisplayPublisherOptionsDialogBox()
    frmPublisherOptions.Show
    frmPublisherOptions.mlpTabs.Value = 0
End Sub
```


This code shows the dialog box frmPublisherOptions, then exits. The dialog box takes care of actually initializing itself by reading the various registry entries and updating itself appropriately.

Next, the function ApplyViewPreferences does the actual initialization by reading the appropriate registry key and updating each of the controls located on the dialog box:

```
Sub ApplyViewPreferences()

    'Applies the user's preferences to the current view.

    Dim Publisher As String
    Publisher =
    "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Common\Publisher"
```

Which key is the right key? The key location is stored in a simple string variable. This saves a lot of typing and possible errors since spelling *Publisher* is so much easier than spelling that whole registry name.

```
'Work with the currently active window:
With ActiveWindow
```

Any line that begins with a leading single quote (') is a comment line. Comments are only there for our own information, and the system ignores them. ActiveWindow tells the system to work with the active window. An example of conditional processing follows:

```
    If System.PrivateProfileString("", Publisher, "Zoom") = "True" Then
        .View.Zoom.Percentage = System.PrivateProfileString("", Publisher,
    Ø "ZoomPercent")
    End If
```

An If statement allows what is called *conditional processing*. When the subject of the If statement is true, the statement after the Then is executed. Conditional processing is the cornerstone of computer programming. Without conditional processing, programs as we know them could not exist.

In the previous example, if the following function:

```
System.PrivateProfileString("", Publisher, "Zoom")
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

returns a value of "True", we fill in the View.Zoom.Percentage with the value contained in the entry ZoomPercent. The function System.PrivateProfileString() returns the value that the value entry contains if it exists. If the entry doesn't exist, System.PrivateProfileString() returns the default, an empty string.

The following code is very similar. We check to see if the data value "Wrap" exists, and what its value is. If the value is "True", we set the variable View.WrapToWindow to True. Otherwise, we set the variable View.WrapToWindow to False.

```
If System.PrivateProfileString("", Publisher, "Wrap") = "True" Then
    .View.WrapToWindow = True
Else
    .View.WrapToWindow = False
End If
```

This same process—of checking a registry entry for its value, setting a local variable to reflect the registry entry's value, and checking the next registry entry in the list—is performed by checking each of the relevant entries, as shown in the following code:

```
If System.PrivateProfileString("", Publisher, "VerticalScroll") = "True"
Then
    .DisplayVerticalScrollBar = True
Else
    .DisplayVerticalScrollBar = False
End If

If System.PrivateProfileString("", Publisher, "HorizontalScroll") = "True"
Then
    .DisplayHorizontalScrollBar = True
Else
    .DisplayHorizontalScrollBar = False
End If
```

After the user finishes with the dialog box and clicks the OK button, the registry is updated by the function All_OK(). If the user clicks the Cancel button instead of OK, Visual Basic for Applications discards everything without making any changes in the registry.

This function is virtually the opposite of the initialization code. Read the value of the dialog control and update the registry appropriately. First, a bit of housekeeping:

```
Sub All_OK()
Dim Publisher As String
```

```
Publisher = "HKEY_CURRENT_USER\Software\Microsoft\Office\9.0\Common\Publisher"
` Make the dialog box disappear!
  frmPublisherOptions.Hide
```

Start up the function and save the desired registry key as a string to avoid having to retype it. Next, hide the dialog box from the user by using `frmPublisherOptions.Hide`.

Simply check each dialog item's state or value, as appropriate, and update the registry. For example, Apply View Preferences is controlled by a pair of radio buttons. Radio buttons are mutually exclusive—only one may be selected at a time. Since there are only two radio buttons for View Preferences, we need only check one of them because the other will be of the opposite value.

```
`Publisher Options dialog box
```

```
If optAutomatically.Value = True Then
  System.PrivateProfileString("", Publisher, "ApplyView") = True
Else
  System.PrivateProfileString("", Publisher, "ApplyView") = False
End If
```

In the previous example, if the control `optAutomatically.Value` returns "True", we set the registry data value `ApplyView` to True. Otherwise, we know that `optManually` (the other control in View Preferences) will be True and we'd have to set `ApplyView` to False.

We continue in the same manner, checking the state or value of each of the dialog box controls, updating the registry as necessary.



TIP Need a quick way to start a Visual Basic for Applications function? Simply create a macro and edit the macro's code to include the functionality that you need.

Confused about System.PrivateProfileString?

The first example in this section retrieved a value from the registry. This value was used to initialize the dialog box controls. In the second example, we set a registry data value with the same function. How does it know the difference?

The magic is that the context of *how the call is made* tells Visual Basic for Applications *how to use it*. In the first instance, the call was within an If statement. In the second use, the call was part of an assignment statement. Visual Basic for Applications knows the difference.

Hints and Kinks from the Experts

Our experts don't have a lot of registry hints about Microsoft Office, because most serious problems with Microsoft Office require reinstallation of the product, usually due to problems with file corruption.

How to Disable That Leaky, Resource-Stealing FindFast

If you remove the FindFast shortcut from the StartUp group, the index files are not removed from your partition. The Microsoft Office apps still continue to use these old index files whenever you use the Open dialogue box, which can cause delays in finding documents. The proper way to remove FindFast follows:

1. Select Start Settings Control Panel Find Fast.
2. Select an entry in "Index for documents in and below" and click Delete Index from the Index menu. Click OK until the index is deleted.
3. Repeat step 2 until "Index for documents in and below" is empty.
4. On the Index menu, click Close and Stop.
5. Remove the FindFast shortcut from the StartUp group.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

PART IV

The Registry Reference

Reality is that which, when you stop believing in it, doesn't go away.

—Philip K. Dick (1928–1982), U.S. science fiction writer

CHAPTER 15

Introduction to HKEY_CLASSES_ROOT

FEATURING:

<i>GUIDs, UUIDs, and other funny numbers in Windows</i>	372
<i>HKEY_CLASSES_ROOT</i>	374
<i>Hints and kinks from the experts</i>	389

Many of the registry's entries deal with Windows 2000, the system. These entries comprise a substantial portion of a new installation's registry, although as more and more applications are installed, this percentage will drop.

Is there anything to fear in the registry's system components? Absolutely! A wrong entry in some system entries will make the system unstable, unbootable, or just plain dead.

We'll cover the registry, hive by hive, pointing out some of the more important entries, some values, and some cautions to consider. This chapter and the four subsequent chapters cover each of the major hives in the registry. This chapter covers HKEY_CLASSES_ROOT; Chapter 16 covers HKEY_CURRENT_USER and HKEY_USERS; Chapters 17, 18, and 19 cover HKEY_LOCAL_MACHINE; Chapter 19 covers HKEY_CURRENT_CONFIG.

The HKEY_CLASSES_ROOT branch contains information about both OLE and various file associations. The purpose of HKEY_CLASSES_ROOT is to provide compatibility with the existing Windows 3.x registry. The information contained in HKEY_CLASSES_ROOT is identical to information found in

HKEY_LOCAL_MACHINE\Software\Classes.

Before we talk too much about HKEY_CLASSES_ROOT, we'll delve into things like GUIDs, UUIDs, and funny registry numbers. Don't let this scare you—it is good (not absolutely necessary, just good) to understand what these numbers *really* are.

Knowing that you have backed up your registry *before* starting this chapter, let's dig in and see what's there.

GUIDs, UUIDs, and Other Funny Numbers in Windows

Windows 2000, Windows NT, and Windows 95/98 for that matter, are just chock full of strange, long numbers. One type of number is the GUID (globally unique ID), a.k.a. the UUID (universally unique ID). Regardless of which term is used, a GUID is *always* a unique number assigned to an application or component. Controls, applications, parts of Windows 2000, software and components, tools, compilers—everything today has one or more GUIDs. Used primarily with OLE (Object Linking and Embedding), GUIDs link components and the operating system. GUIDs are used as a linkage between applications, file types, embedding, OLE, objects, and the operating system.

For example, Microsoft Word has a GUID of {000209FF-0000-0000-C000-000000000046}. This is unique enough that we can be sure that a request for this GUID will always match Microsoft Word, and not some other application. How can we say that? After all, although a GUID is long (it's a number with 16 bytes, or 128 bits), what mechanisms are there to make sure that each programmer uses a unique GUID?

The process of obtaining a GUID is simple and, in most cases, doesn't even require any direct interaction with Microsoft. Does that make you rather nervous? Fear not, Microsoft provides a tool to generate a GUID, and that tool takes some rather interesting steps to attempt to make each GUID unique.

First, a bit of history (just what you wanted, a history lesson). All Ethernet network interface cards (NICs) have a unique identifying number built into them. That's right, the NIC in your computer is different from the NIC of the computer in the office next door. This means that each computer with a NIC actually has a form of a unique serial number.

The NIC's serial number is there to allow the hardware layer of the network to be able to distinguish between different computers on the network. An organization assigns part of this identifier to each NIC manufacturer, and the manufacturer assigns the second part of the identifier to each NIC at assembly time. Most NICs have their ID number written on a small sticker on the card, though in today's world, users and administrators have virtually no need for the NIC's ID.

The Microsoft GUID program takes the NIC's identifier number, which is unique, the current time and date information, hashed a bit, and a random number, and uses these to create the GUID. To have two identical GUIDs, it would be necessary to have two computers with the same NIC identifying numbers, at the same time (exactly, to the millisecond), and with the same random number.

In short, it is unlikely that two GUIDs would be the same. Even if a programmer were to get the command to run at *exactly* the same time on both computers, it is not reasonable that the random number would be the same on both runs. This is because the random number is not based on time or any other factor that a programmer might be able to influence. Hence, we can be reasonably sure that the GUID for each application will be unique.

There is actually one instance where a GUID might not be unique: That is when a programmer intentionally copies the GUID for one program into another program. This could be unintentional; but more likely, the program would do this by design. I can't think of any valid reason why a programmer might create two applications with the same GUID, but I'm sure that someone will write and tell me why this could, or would, happen.

A GUID consists of five groups of digits in hexadecimal. Hyphens separate each group. These groups display four bytes, two bytes, two bytes, two bytes, and six bytes—in that order—as the following GUID shows:

```
{000209FF-0000-0000-C000-000000000046}
```

It is common, although not specifically required, that braces enclose a GUID. However, whenever you encounter a number with the above arrangement of digits (8, 4, 4, 4, 12), you can generally assume that the

number is a GUID.

A Rose by Any Other Name

UUID and GUID are just different names for the same thing. Ditto for CLSID (class ID). CLSIDs, GUIDs, and UUIDs all identify a specific class of objects. Treat a CLSID the same as you would treat a GUID or a UUID, and all will be well.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

HKEY_CLASSES_ROOT

The HKEY_CLASSES_ROOT hive contains information about both OLE and various file associations.



WARNING A little later in this chapter we'll start fiddling with the registry. You are an intelligent person; therefore, you know that you should back up your registry before you start. Please, do not change the registry without having a good, easily restored backup.

HKEY_CLASSES_ROOT provides compatibility with the existing Windows 3.x registry; some applications and systems expect HKEY_CLASSES_ROOT to exist. The information contained in HKEY_CLASSES_ROOT is identical to information found in the subkey HKEY_LOCAL_MACHINE\Software\Classes. Actually, these two objects are physically the same. A change made in one will automatically modify the other. Think of HKEY_CLASSES_ROOT as a house on the corner of an intersection. The house might have two addresses, one on each street. Remember: HKEY_CLASSES_ROOT is HKEY_LOCAL_MACHINE\SOFTWARE\Classes and HKEY_LOCAL_MACHINE\SOFTWARE\Classes is HKEY_CLASSES_ROOT.

Managing File Types and File Extensions

The HKEY_CLASSES_ROOT hive consists of a list of all file extensions (file types) known to your installation of Windows 2000. Each time you install a new application, the application should add or modify one or more extensions. The application's setup program does this, and this process tells Windows 2000 that the application will handle (open, print, and so on) that type of file when users select it.

For example, the HKEY_CLASSES_ROOT key for an Excel spreadsheet file (any file that ends in .xls) is as follows:

```
HKEY_CLASSES_ROOT\.xls
HKEY_CLASSES_ROOT\.xls\Excel.Sheet.5
HKEY_CLASSES_ROOT\.xls\Excel.Sheet.5\ShellNew
HKEY_CLASSES_ROOT\.xls\Excel.Sheet.5\ShellNew\FileName = excel.xls

HKEY_CLASSES_ROOT\.xls\ExcelWorksheet
HKEY_CLASSES_ROOT\.xls\ExcelWorksheet\ShellNew
HKEY_CLASSES_ROOT\.xls\ExcelWorksheet\ShellNew\FileName = excel4.xls

HKEY_CLASSES_ROOT\.xls\ShellEx
```



Desktop Library
[Click to access!](#)

```
HKEY_CLASSES_ROOT\.xls\ShellEx\{00021500-0000-0000-C000-000000000046}
HKEY_CLASSES_ROOT\.xls\ShellEx\{00021500-0000-0000-C000-000000000046}
  \<NO NAME> = {83799FE0-1F5A-11d1-95C7-00609797EA4F}

HKEY_CLASSES_ROOT\.xls\ShellEx\{BB2E617C-0920-11d1-9A0B-00C04FC2D6C1}
HKEY_CLASSES_ROOT\.xls\ShellEx\{BB2E617C-0920-11d1-9A0B-00C04FC2D6C1}
  \<NO NAME> = {9DBD2C50-62AD-11d0-B806-00C04FD706EC}
```

This complex group of registry entries results from the complexity of the Microsoft Office product; after all, we pay a lot for those Office products.

Another example is for batch files with the extension of .bat. The entry we find for batch files is as follows:

```
HKEY_CLASSES_ROOT\.bat
HKEY_CLASSES_ROOT\.bat\<No Name> = batfile
```

We see an identifier, which has no name, with a data value of batfile. Looking a bit further down the line (or down HKEY_CLASSES_ROOT, so to speak), we find an entry called batfile. Coincidence? Luck? Secret conspiracy? Here are the facts for batfile:

```
HKEY_CLASSES_ROOT\batfile
HKEY_CLASSES_ROOT\batfile\<NO NAME> = MS-DOS Batch File
HKEY_CLASSES_ROOT\batfile\EditFlags = 0x00000430
```

```
HKEY_CLASSES_ROOT\batfile\DefaultIcon
HKEY_CLASSES_ROOT\batfile\DefaultIcon\<NO NAME> =
  %systemroot%\System32\shell32.dll,-153
```

```
HKEY_CLASSES_ROOT\batfile\shell
HKEY_CLASSES_ROOT\batfile\shell\edit
HKEY_CLASSES_ROOT\batfile\shell\edit\<NO NAME> = &Edit
```

```
HKEY_CLASSES_ROOT\batfile\shell\edit\command
HKEY_CLASSES_ROOT\batfile\shell\edit\command\<NO NAME> =
  %systemroot%\System32\notepad.exe %1
```

```
HKEY_CLASSES_ROOT\batfile\shell\open
HKEY_CLASSES_ROOT\batfile\shell\open\EditFlags = 0x00000000
HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\batfile\shell\open\command\<NO NAME> = "%1" %*
```

```
HKEY_CLASSES_ROOT\batfile\shell\print
HKEY_CLASSES_ROOT\batfile\shell\print\command
HKEY_CLASSES_ROOT\batfile\shell\print\command\<NO NAME> =
  %systemroot%\System32\notepad.exe /p %1
```

```
HKEY_CLASSES_ROOT\batfile\shellex\batfile\shellex\PropertySheetHandlers
HKEY_CLASSES_ROOT\batfile\shellex\PropertySheetHandlers\PifProps
HKEY_CLASSES_ROOT\batfile\shellex\PropertySheetHandlers\PifProps\<NO NAME>
  = {86F19A00-42A0-1069-A2E9-08002B30309D}
```

Now, the preceding set of entries tell us and Windows 2000 everything needed to handle a .bat file—the icon to display, how to edit it, how to open it, how to print it, and how to process (execute) it. Let's look at each section of this entry. We'll begin with the first section:

```
HKEY_CLASSES_ROOT\batfile
HKEY_CLASSES_ROOT\batfile\<NO NAME> = MS-DOS Batch File
HKEY_CLASSES_ROOT\batfile\EditFlags = 0x00000430
```

Initial handling for batch files includes (in an unnamed variable) the text string used both in Explorer for the file's properties dialog box and in the Type column of Explorer's detailed list view. Modifying this string

changes the behavior of Explorer and Windows for the properties displayed for a batch file.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

The EditFlags variable controls how Windows processes the command.

EditFlags and Bitmapped Variables

EditFlags is bitmapped, with a few apparent bits. Flags are combined by ANDing them together. For example, 00 08 00 00, ANDed with 00 01 00 00 would result in the value 00 09 00 00. Use the Windows calculator program and simply add the values; the result will be identical.

EditFlags seems to affect the way that Windows 2000 and its components, such as Internet Explorer, handle receiving certain files, as well as how files are processed.

Below is a list of the EditFlags values that I am aware of. Some flag bits seem to have two meanings.

01 00 00 00	For an actual file, removes the file type from the file types list.*
02 00 00 00	For objects, not files, adds the file type to the file types list.* Disables the MIME edit field.*
04 00 00 00	Used to signify that the type has no associated extension.*
08 00 00 00	Disables the Advanced button in the File Types tab.
10 00 00 00	Disables the Remove button in the File Types tab.*
20 00 00 00	Disables the Actions \emptyset New command in the File Types tab.*
40 00 00 00	Disables the Edit button in the Edit File Type dialog box.
80 00 00 00	Disables the Remove button in the Edit File Type dialog box.
00 01 00 00	Disables the Edit Name button in the Edit File Type dialog box.
00 02 00 00	Disables the Change Icon button in Edit File Type dialog box.
00 04 00 00	Disables the Set Default button in Edit File Type dialog box.*
00 08 00 00	Unknown.
00 10 00 00	Disables changing command actions.*
00 20 00 00	Disables changing DDE settings.*
00 40 00 00	Unknown.
00 80 00 00	Unknown.
00 00 01 00	Unknown.
00 00 02 00	Unknown.
00 00 04 00	Unknown.
00 00 08 00	Unknown.

00 00 10 00 Unknown.
* This setting is unconfirmed in Windows 2000.
00 00 20 00 Unknown.
00 00 40 00 Unknown.
00 00 80 00 Unknown.
00 00 00 01 Confirms open after download in Internet Explorer.
00 00 00 02 Unknown.
00 00 00 04 Unknown.
00 00 00 08 Unknown.
00 00 00 10 Unknown.
00 00 00 20 Unknown.
00 00 00 40 Unknown.
00 00 00 80 Unknown.

Some commonly used flag combination values are:

EditFlag 0xD2010000: For drives, directories, folders, etc. Disables Edit File Type, Remove, Description, and Edit Name in the Edit dialog box. Also adds the file type to the list if this isn't a real file.

EditFlag 0x30040000: For batch files, disables Set Default, Remove, and New.

EditFlag 0x38070000: For applications, disables Edit, Set Default, Change Icon, and Edit Description.

The next section for batch files is as follows:

```
HKEY_CLASSES_ROOT\batfile\DefaultIcon
HKEY_CLASSES_ROOT\batfile\DefaultIcon\<NO NAME> = %systemroot%\System32\
shell32.dll,-153
```

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The DefaultIcon entry specifies which icon Explorer displays in the Explorer program or on the Desktop, as appropriate. Notice that Explorer won't allow you to use the Explorer Properties dialog box to change the icon for a batch file. Here is where it is changed:

```
%systemroot%\System32\shell32.dll, -153
```

What does that magic line mean? The file named %systemroot%\System32\shell32.dll is a .dll file that has, in addition to other things, a whole bunch of icons. The second number is a bit of a mystery, right? First, it is negative; just how do you find a negative icon, anyway? Second, there doesn't seem to be any simple program or method to find which icon matches this magic number.

The negative number isn't so difficult. Icons are stored as resources in executable files; .exe and .dll files are both executable, but other extensions are also executable and can have icons in them. Each resource has a unique signed number from 0 to 65535 (a two-byte value) assigned. These *resources* (icons, dialog boxes, and strings) use this number to identify themselves to Windows. Programmers, and programmers' tools, ignore the fact that these resources are stored with signed numbers and simply ignore the signs. A programmer sets the icon's identifier to 65383; and Windows, to make things easy for all of us, displays it as -153. So icon number -153 is actually icon number 65383.

Some .dll files with lots of icons in them are pifmgr.dll, moricons.dll, and shell32.dll. There are other files containing icons, too.

Let's look at the next section:

```
HKEY_CLASSES_ROOT\batfile\shell
HKEY_CLASSES_ROOT\batfile\shell\edit
HKEY_CLASSES_ROOT\batfile\shell\edit\<NO NAME> = &Edit
```

```
HKEY_CLASSES_ROOT\batfile\shell\edit\command
HKEY_CLASSES_ROOT\batfile\shell\edit\command\<NO NAME> =
    %systemroot%\System32\notepad.exe %1
```

The shell\edit section describes how to edit the subject file. The name of the context-menu selection to edit is in the variable having no name. (Right-click the file in Explorer to see the context menu.) The default variable name for most programs is &Edit, which displays the word Edit. (The letter preceded with an ampersand is the accelerator key's letter and will be underscored.)

The section shell\edit\command contains a single, unnamed entry listing the editor to edit the file. In the case of



a batch file, the default editor is Notepad. If you have a favorite editor, you can plug it into this location to have it edit the file. Just remember that the editor must be able to open and save the file in the correct format. Fortunately for batch files, this is not difficult; they are plain text files with no special editing requirements. When the editor is called, the argument %1 will be substituted with the batch file's name, as shown here:

```
HKEY_CLASSES_ROOT\batfile\shell\open
HKEY_CLASSES_ROOT\batfile\shell\open\EditFlags = 0x00000000
HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\batfile\shell\open\command\<NO NAME> = "%1" %*
```

The shell\open section contains the code to execute the file. In the case of a batch file, the EditFlags value is 0x00000000. Notice the format of this command, especially the placement of the quotes: "%1" %*. This command string will have the initial (quoted) %1 substituted with the batch file's name and the second %* substituted with any parameters that the user passed to the command. If editing the data, be very careful not to place the quotes in the wrong place; don't quote the entire string, for example.

The next section handles printing requests:

```
HKEY_CLASSES_ROOT\batfile\shell\print
HKEY_CLASSES_ROOT\batfile\shell\print\command
HKEY_CLASSES_ROOT\batfile\shell\print\command\<NO NAME> = %systemroot%\
    System32\NOTEPAD.EXE /p %1
```

Printing, managed by the shell\print section, contains only one working entry under HKEY_CLASSES_ROOT\batfile\shell\print\command with a single, unnamed entry. This entry tells Explorer to print using Notepad, passing the filename and the /p option. The option /p is a relatively standard option telling the program to open the file, print it to the default printer, and then exit. Generally, Windows prints the entire file, although it is possible that some applications will provide options for the print process. (Notepad is not silent or hidden; you will see it open, see the print dialog, and see Notepad close.)

```
HKEY_CLASSES_ROOT\batfile\shellex
batfile\shellex\PropertySheetHandlers
HKEY_CLASSES_ROOT\batfile\shellex\PropertySheetHandlers\PifProps
HKEY_CLASSES_ROOT\batfile\shellex\PropertySheetHandlers\PifProps\
    <NO NAME> = {86F19A00-42A0-1069-A2E9-08002B30309D}
```

Mappings to programs for all CLSIDs are in the CLSID part of HKEY_CLASSES_ROOT. Looking up our magic CLSID, {86F19A00-42A0-1069-A2E9-08002B30309D}, we find it is registered for shell32.dll, along with a few other settings. This tells us that the PIF (Program Interface File) manager is actually part of shell32.dll, used to display the property sheet for batch files:

```
HKEY_CLASSES_ROOT\CLSID\{86F19A00-42A0-1069-A2E9-08002B30309D}
HKEY_CLASSES_ROOT\CLSID\{86F19A00-42A0-1069-A2E9-08002B30309D}\
    <NO NAME> = .PIF file property pages
HKEY_CLASSES_ROOT\CLSID\{86F19A00-42A0-1069-A2E9-08002B30309D}\
    InProcServer32
HKEY_CLASSES_ROOT\CLSID\{86F19A00-42A0-1069-A2E9-08002B30309D}\
    InProcServer32\<NO NAME> = shell32.dll
HKEY_CLASSES_ROOT\CLSID\{86F19A00-42A0-1069-A2E9-08002B30309D}\
    InProcServer32\ThreadingModel = Apartment
```

Several items in the CLSID section are worth noting. First, InProcServer32 is the name for a section dealing with in-process servers. In this case, we are working with a 32-bit in-process server, but that's not important right now.

We get the name of the server, shell32.dll, from the variable with no name; and we get the threading model, Apartment, from the ThreadingModel entry. These are important, since specifying the wrong ThreadingModel can cause data corruption. Other possible values for ThreadingModel are Single and Both, although it is unlikely that you will see Single specified.

One picture is worth a thousand words. Or so they say. Figure 15.1 shows the entries for a batch file (.bat) a bit more graphically.

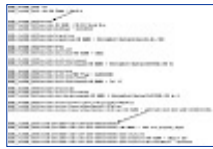


FIGURE 15.1 The entries for .bat type files in HKEY_CLASSES_ROOT, showing their relationships

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

As shown in Figure 15.1, batch files are a relatively more complex example of how a particular file type is processed. Some other types of files are simpler—for example, they may not support context menus—while some are much more complex. Each system will be different for optional components, although Windows 2000 components typically are similar regardless of the installation.

OK, what have we learned? First, for virtually any object that relates to a file (except My Computer, My Network Places, Recycle Bin, and so on), we can set the text description, change the icon, set an editor to edit, set a printer to print, and control how the object is executed or opened, as appropriate. In fact, we can add almost any functionality to the context menu we might want to. For instance, we can set a second editor for batch files; we'll use the command prompt's editor.

This example uses RegEdt32. I'm going to start right from the beginning since this is our first registry hack, I mean "fix."

1. Open RegEdt32. The current local registry will be displayed.
2. Make HKEY_CLASSES_ROOT the top window either by selecting it in the Window menu or by clicking it.
3. A batch file's extension is .bat, so find .bat in the list of extensions.
4. Open the .bat key, where you will see that there is one unnamed entry with a value of batfile. Figure 15.2 shows this entry. An additional subkey, called PersistentHandler, contains the GUID for the handler for a batch file.



FIGURE 15.2 Most extension entries have a single entry referring to a subsequent entry in HKEY_CLASSES_ROOT.

5. Find the entry batfile in HKEY_CLASSES_ROOT and expand the shell subkey. The original shell subkey contains three entries: edit, open, and print.
6. Create a new subkey under shell and call this new subkey NewEdit. (Sure, you can call this new subkey anything you want.)
7. In your NewEdit subkey, create an unnamed entry with a data type of REG_SZ. In this entry, put the text of the new command you are adding. In this example, we are adding the command-level editor (the editor displayed when you type edit at a command prompt), so I'm adding the string Edit with DOS edit. Take a gander at Figure 15.3 to see what we've done so far!

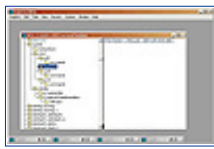


FIGURE 15.3 RegEdt32 with NewEdit open, showing the unnamed key with the command's menu text.

8. Under the NewEdit subkey, create a second subkey called command.
9. In the subkey command, create a new, unnamed value with a data type of REG_EXPAND_SZ. Make sure you use REG_EXPAND_SZ, and not REG_SZ, because this string will have an expansion variable embedded in it.
10. The string value for this new variable is the command itself. In our case, we are going to use the command editor, edit.com, which is located in %systemroot%\System32. Add this string like so:

```
%systemroot%\System32\edit.com %1
```

The %1 is a substitution variable, much like substitution variables in batch files, where Windows 2000 will substitute the name of the file to load into the editor. Figure 15.4 shows this change.

11. We are done adding a new context-menu selection. Close RegEdt32 and restart Windows.

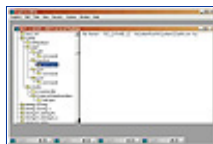


FIGURE 15.4 RegEdt32 with the new command added



NOTE Although not always necessary, I recommend that you restart Windows 2000 after each registry modification. Windows 2000 caches some parts of the registry, and changes won't become visible until after restarting. Actually, much of HKEY_CLASSES_ROOT is cached, so a reboot is an especially good idea here.

Figure 15.5 shows the new context menu in action. When the user clicks the menu selection Edit with DOS Edit, the command-prompt editor will open in its own window, and the selected file will be loaded, as shown in Figure 15.6.

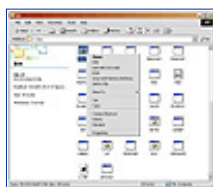


FIGURE 15.5 The new, modified context menu. Look—we now have a new editor to choose from.



FIGURE 15.6 Using a different editor may be just the trick for some users.

Editing with the command editor may be easier for some users, or maybe you have a favorite editor you would like to substitute.

You can modify all context menus in Explorer (which includes the Desktop) using this technique. You can add selections for different file types, new actions to take, new editors, and new print options, whatever.

Managing OLE and Embedding

The second function of HKEY_CLASSES_ROOT is to manage OLE. It is perhaps well beyond the scope of this book to really delve into the intricacies of OLE. But a quick introduction is in order.

OLE (Object Linking and Embedding) is a basic functionality that Microsoft has been working on for the last 8 to 10 years. The origins of OLE, or at least the concepts surrounding OLE, are vague. Some of these techniques and functions can be traced back to the beginnings of Windows and something called DDE (Dynamic Data Exchange). DDE was one hell of a difficult thing to work with, and Microsoft quickly expanded it to make it more flexible.

OLE consists of a whole slew of features, but the main one we'll worry about today is the concept of embedding. *Embedding* is the process of using one application inside another application. Many of Microsoft's applications rely heavily on embedding. Outlook is one example; it's the Microsoft Desktop information management system that many of us use for e-mail. Outlook can use Microsoft Word as the preferred e-mail editor by embedding Word into Outlook's e-mail editor window. When this is done, Word's menus, toolbars, and other functions are all available to the user.

Figure 15.7 shows Word as the e-mail editor, running and editing a message to my editor. An invisible Word window exists with this chapter open. Using Word to edit an e-mail message doesn't affect Word's ability to be a word processor, although I do save my work before using Outlook.

There is nothing that would prevent you from writing an application that allowed embedding Word. For that matter, embedding into a client application is possible for virtually all server applications. There are established mechanisms to determine the server's capabilities, what is needed to embed, and so on, although it is well beyond the scope of this book to get into that topic. They say there are only about two programmers who really understand embedding and OLE, and they both work for Microsoft.



FIGURE 15.7 Outlook's e-mail editor with Word embedded

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!
The Default Client—It's a Dirty Job but Someone Has to Do It

Windows 2000 offers what amounts to a default client application. It gives a server application that is incapable of running on its own a way to execute. This default client application is called RunDLL32.exe.

When RunDLL32 is executed, it is passed the name of the server, typically an ActiveX control (a.k.a. an OLE control), some actions, and the name of the subject object, typically a file of some sort.

For example, the entry for amovie.ocx, an ActiveX control used to display MPEG (video) files, is as follows:

```
%systemroot%\System32\RunDLL32.exe
%systemroot%\System32\amovie.ocx,RunDll /play /close %1
```

In this example, RunDLL32 will load amovie.ocx, passing these four parameters:

- RunDll—tells amovie.ocx who the client is.
- /play—tells amovie.ocx to play the specified object; in this case the object is a file.
- /close—tells amovie.ocx to close after playing the specified object.
- %1—tells amovie.ocx which file contains the object to be played.

Delving into the Unknown

Regardless of how many viewer controls, applications, and whatever else you install under Windows 2000, there are going to be unknown file types. Windows refers to these files as *unknown*. When Windows opens an unknown file, a dialog box is displayed that is named Open With. This dialog box, shown in Figure 15.8, allows you to select an application to open files whose file type is currently not defined.

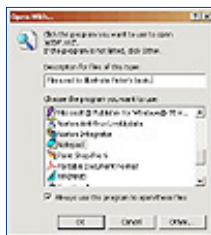


FIGURE 15.8 Open With allows both opening a specific file of an unknown type and setting a new default action for other files at the same time.

The Open With dialog box allows you to select the application used to open the file. Also, if the “Always use this program to open these files” box is checked, Open With will create a new default handler for the file type.

Default handlers include most applications installed under Windows 2000, that are properly registered in the registry—that is, installed correctly. In addition, several system components are used as handlers, including RunDLL32 (which allows running ActiveX controls), WinHlp32 (to open standard Windows help files), and Internet Explorer.

Whenever a selection does not appear within the list of programs in the Open With dialog box, clicking the Other button allows you to select any other executable program.

When the box labeled “Always use this program to open these files” is checked, new entries in HKEY_CLASSES_ROOT are created for this file type. These entries may then be edited or modified by the user, using the techniques previously shown, to change or enhance the behavior of the context menu.

Hints and Kinks from the Experts

In this chapter’s hints and kinks, we cover a couple of hints for the hive HKEY_CLASSES_ROOT.

Define a Default Open When You Right-Click in Explorer

When you right-click a file with a known extension (in Explorer), you get a choice of how to open the file. If the extension is unknown, you get an Open With choice.

You can define a default Open choice for those unknown extensions and still have the Open With by amending the registry as follows:

1. Open HKEY_CLASSES_ROOT\Unknown\Shell in the registry editor.
2. Double-click Shell.
3. Add a key called Open.
4. With Open selected, add a key called command.
5. With command selected, add a value with no name (leave it blank) and type **REG_EXPAND_SZ**.
6. Enter the full path to the executable, followed by a space and “%1”; for example, Drive:\Directory\Program.exe “%1”.

(Courtesy of Jerold Schulman.)

Add “open with xyz” to Every Explorer Right-Click

You can add a new option to every right-click by doing the following:

1. In the registry editor, edit the registry key HKEY_CLASSES_ROOT*.
2. Add a key called EditFlags with a type of REG_BINARY.
3. With the Hex button checked, enter a value for EditFlags of 02000000.
4. Open Explorer and select the File Types tab from the Options menu, displayed when selecting options in the View menu.
5. Highlight the * entry and click the Edit button.
6. Click the New button. In the Action box, type **open with xyz**. The xyz represents whatever program you want to run when the user selects the new menu selection.
7. In the Application box, browse to the program (xyz) you wish to use.
8. Click OK to close all dialog boxes.

Now, when you right-click in Explorer, you will always be able to “open with xyz.”

To restrict this functionality to unknown file types, perform the registry edit on HKEY_CLASSES_ROOT\Unknown, and modify the unknown selection (step 5) instead.



NOTE This procedure will prevent you from using the Office 95 toolbar.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 16**Introduction to HKEY_CURRENT_USER and HKEY_USERS****FEATURING:**

<i>HKEY_CURRENT_USER</i>	393
---------------------------------	------------

<i>HKEY_USERS</i>	430
--------------------------	------------

<i>Hints and kinks from the experts</i>	430
--	------------

HKEY_CURRENT_USER and HKEY_USERS are hives that deal with users and user profiles. When first installed, Windows 2000 systems have two profiles configured: the default user, used by the system when no user is logged on, and a profile for the currently logged-on user.

HKEY_CURRENT_USER is the profile for the currently logged-on user. HKEY_CURRENT_USER is actually just a link to the user's profile that is stored in HKEY_USERS. Changes made in HKEY_CURRENT_USER are also going to appear in HKEY_USERS and be saved when the user logs off (if the user doesn't have a mandatory profile).

HKEY_USERS contains the profile for the currently logged-on user and the default profile. All user profiles are stored as separate profile files and are loaded as needed. They are saved in the %systemroot%\Profiles\

For example, consider our fictional user, Pixel:

1. The user Pixel logs onto the system.
2. Windows 2000 validates the user ID and password with the Active Directory server or the local machine's security manager if the user is not logging onto a domain.
3. The logon checks the user's profile status.
4. If the profile is local, or if the user is not logging onto a domain, the profile is loaded from the local machine.
5. If the user's profile is not local, and the user is logging onto a domain and has a roaming or



mandatory profile, the correct profile is loaded from the appropriate network share. The user's NTUSER.DAT file will be loaded into the registry's HKEY_USERS hive with a subkey name equal to the user's SID.

6. The user's profile is read from the server and updated as necessary to reflect the user's preferences. The user uses the same profile when logging on at any computer in the domain.



NOTE Pixel, who's Pixel? Try Robert A. Heinlein's book, *The Cat Who Walked Through Walls* (ISBN 0-441-09499-6), to learn about the cat, Pixel.

In the process of loading the user's profile, the user's file NTUSER.DAT is loaded into HKEY_USERS. This hive contains the user's registry settings, everything that will later appear in the HKEY_CURRENT_USER hive. This becomes a hive with a name equal to the user's SID. Windows 2000 unloads and saves to the original location the profile of the previously logged-on user. This leaves HKEY_USERS with only two user profiles loaded at any given time, for the default user and the current user. (Actually, if no user logs on, then only one profile is present: the default profile.)



WARNING Careful—backing up a server's registry doesn't back up each user's profile. It is necessary to completely back up the server's Profiles directory, which contains information for each user who is defined on that machine and who has a profile stored there.

Users who don't have authority to modify their profile because they are using a mandatory profile may make changes. However, these changes will be lost when the user logs off.



NOTE What happens when the same user ID, with a roaming profile, is used concurrently on two different computers? This situation is not well defined. It is not an error; however, only one of these multiple logged sessions will actually save the user's profile. The session that is the last to log off will overwrite all other saves by other sessions.

The major user components in the registry are:

HKEY_CURRENT_USER: The HKEY_CURRENT_USER hive manages specific information about the currently logged-on user. Remember, Windows 2000 automatically reflects changes made in HKEY_CLASSES_ROOT to the user's information contained in HKEY_USERS.

HKEY_USERS: The HKEY_USERS hive contains information about the system when there is no logged-on user and about the user currently logged on.

Again, a backup is vital. I have to do one myself, so while I'm busy, why don't you do a registry backup, too.

HKEY_CURRENT_USER

The HKEY_CURRENT_USER branch manages specific information about the currently logged-on user. This hive contains a complete profile of how Windows will look and behave for the user.



NOTE If at any time it is desirable to modify Windows 2000's look and feel (the profile) when there is no logged-on user, modify the entries in the HKEY_USERS\DEFAULT subkey. There are parallel entries in this subkey for virtually every entry found in HKEY_CURRENT_USER. Realize that some changes won't be meaningful because they represent parts of the system that are inaccessible when there is no logged-on user.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

Major subkeys in HKEY_CURRENT_USER include the following:

AppEvents: This subkey includes information about labels for events, such as the default beep. Other information includes the sounds (such as the beeps, dings, and bongs) that Windows emits when things happen. It is not common to edit label entries, but it is possible. Use the Control Panel's Sounds applet to change event sounds, although some sounds must be changed directly in the registry.

Console: The colors, font, and other command-window metrics are stored in this subkey. These settings apply to console windows only; other windows have their metrics stored elsewhere.

Control Panel: This subkey holds settings for some of the Control Panel's applets. Examples of settings saved here include those for the Accessibility, Appearance, Mouse, and Keyboard applets.

Environment: The system environment strings are saved in this subkey.

EUDC: Information regarding End User Defined Characters is stored in this subkey.

Identities: This subkey holds settings for user-specific configurations of certain software.

Keyboard Layout: The keyboard layout can be modified from this subkey, typically when a special-purpose keyboard is used.

Network: All drives to which drive letters are mapped are managed in this subkey. Explorer primarily manages drive mapping of network shares.

Printers: All printers, local and remote, are managed in this subkey. Printer information is accessible in the system's Printer applet.

RemoteAccess: The wizard for dial-up networking services that allows connecting to remote computers and networks stores information in this subkey.

Software: Information about all installed software is stored in this subkey. The vendor typically arranges this information, although some applications may be in their own subkeys.

Unicode Program Groups: This subkey contains information used by Program Manager. (Is anyone using Program Manager anymore?) The subkeys found in Unicode Program Groups are in a binary format that is difficult to edit. (Have I ever seen entries in the Unicode Program Groups subkey? No, not yet.) Actually, users who upgraded from Windows NT 3.x may have entries in this subkey. However, Windows NT version 4 or later will not use this subkey.

Windows 3.1 Migrations Status: This subkey is contained only in systems that upgraded from Windows NT 3.x. It contains the keys used to show the status of conversion of the Program Manager Group files (.grp) and associated initialization files (.ini) that have been converted to Windows NT 4/Windows 2000 format. Deleting this value and not the Windows 3.1 Migration Status subkey causes Windows NT/2000 to attempt to convert the Program Manager files when Windows restarts. This re-conversion may change the Start menu substantially.

Volatile Environment: Typically, this subkey contains only one entry, a value called LOGONSERVER that contains a string with the logon server (the server the user is currently logged on to). For example, my logon server is \\DORA.

AppEvents

The AppEvents subkey contains all the information that Windows 2000 uses to play sounds whenever an event happens. There are event labels for a number of events, as Table 16.1 shows. This table lists sounds found in virtually all Windows 2000 systems right from the first installation. Additionally, the AppEvents subkey contains definitions of what sounds to play when an event occurs. Finally, AppEvents also contains sound schemes for both default sounds and no sounds. Users may create new schemes, as desired, using the Control Panel's Sounds applet—more on that later.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
TABLE 16.1: SOUNDS FOUND ON MOST WINDOWS SYSTEMS

Subkey	Default Text	Description
AppEvents\EventLabels\.Default	Default Beep	Default sound used when a sound is needed, but no specific sound has been defined
AppEvents\EventLabels\ActivatingDocument	Complete Navigation	Sound played when the navigation of an object is complete
AppEvents\EventLabels\AppGPFault	Program Error	Sound played when a program returns an error
AppEvents\EventLabels\CCSelect	Select	Sound played when an object is selected
AppEvents\EventLabels\Close	Close Program	Sound played when a program closes
AppEvents\EventLabels\EmptyRecycleBin	Empty Recycle Bin	Sound played when the Recycle Bin is emptied
AppEvents\EventLabels\MailBeep	New Mail Notification	Sound played when a new e-mail arrives
AppEvents\EventLabels\Maximize	Maximize	Sound played whenever a window is maximized
AppEvents\EventLabels\MenuCommand	Menu Command	Sound played whenever a menu item is selected
AppEvents\EventLabels\MenuPopup	Menu Pop-up	Sound played whenever a pop-up (context) menu item is selected
AppEvents\EventLabels\Minimize	Minimize	Sound played whenever a window is minimized
AppEvents\EventLabels\MoveMenuItem	Move Menu Item	Sound played whenever a menu item is moved
AppEvents\EventLabels\MSVC_HitBP	Breakpoint Hit	Sound played when a breakpoint in Microsoft Visual C++ has been reached (may not be present if Microsoft Visual C++ is not installed)


Desktop Library
 Click to access!

AppEvents\EventLabels\MSVC_OutputError	Error in Output	Sound played when an error in Microsoft Visual C++ has been detected (may not be present if Microsoft Visual C++ is not installed)
AppEvents\EventLabels\MSVC_OutputWarning	Warning in Output	Sound played when a warning in Microsoft Visual C++ has been detected (may not be present if Microsoft Visual C++ is not installed)
AppEvents\EventLabels\Navigating	Start Navigation	Sound played when navigation begins
AppEvents\EventLabels\Open	Open Program	Sound played when a program starts or opens
AppEvents\EventLabels\RestoreDown	Restore Down	Sound played when a window is restored from the maximized size to the normal size
AppEvents\EventLabels\RestoreUp	Restore Up	Sound played when a window is restored from the minimized size to the normal size
AppEvents\EventLabels\RingIn	Incoming Call	Sound played when an incoming telephony call is received
AppEvents\EventLabels\RingOut	Outgoing Call	Sound played when an outgoing telephony call is made
AppEvents\EventLabels\ShowBand	Show Toolbar Band	Sound played when the toolbar band is shown
AppEvents\EventLabels\SystemAsterisk	Asterisk	Sound played as the standard Windows asterisk sound
AppEvents\EventLabels\SystemExclamation	Exclamation	Sound played as the standard Windows exclamation sound
AppEvents\EventLabels\SystemExit	Exit Windows	Sound played when Windows is exited
AppEvents\EventLabels\SystemHand	Critical Stop	Sound played as the standard Windows critical stop sound
AppEvents\EventLabels\SystemQuestion	Question	Sound played as the standard Windows question sound
AppEvents\EventLabels\SystemStart	Start Windows	Sound played when Windows starts

Sounds based on events are set in the Control Panel's Sounds applet. Figure 16.1 shows this simple program. Each event can have one sound assigned, and users can create and save event sound schemes.

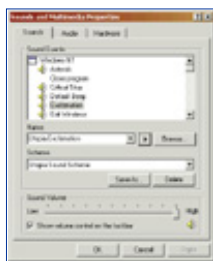


FIGURE 16.1 The Control Panel's Sounds applet sets sounds and sound schemes.

All sounds are rather meaningless unless the computer supports audio. Windows systems without sound compatibility will display these labels, and the user may set system sounds, but Windows cannot play these sounds. after all, how can Windows play a sound without a sound system? (Experiments at the Dilbert facility using Elbonionans to make the appropriate sounds did not succeed!)

Maybe, just maybe, there will be systems that don't have all of the above event labels. This is typically the case when a system administrator has substantially customized the installation and has deleted these objects. although there may seem to be good reasons to delete event labels, it rarely is a good idea—more likely, it is a case of someone trying to generate work for themselves.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

Once a system has more software applications and perhaps hardware too, these products may add events. These events will require labels and (probably, although not necessarily) sounds.

Microsoft Office, for example, adds about 40 event labels; that's more than the default version of Windows. Microsoft's Developer Studio consists of a myriad of development tools, including Visual C/C++, Visual BASIC, Visual FoxPro, and others. It will also add many new events. Events and event labels can become overwhelming if lots of applications are installed.

We previously discussed labels for events. Next, we need a table of sounds to "play" when the event happens. These events are located in HKEY_CURRENT_USER\AppEvents\Schemes\Apps\DEFAULT. This subkey contains entries to match each entry in the EventLabels subkey (listed in Table 16.1). Each subkey has at least two subkeys, including .Current and .DEFAULT. For example, HKEY_CURRENT_USER\AppEvents\Schemes\Apps\DEFAULT\Default\ contains the following:

.Current: This subkey contains one unnamed entry with the value of ding.wav, unless the user has changed the sound to be played. That is, when a default event (an event that doesn't have its own sound defined) occurs, Windows will play the ding.wav file.

.DEFAULT: This subkey contains one unnamed value with the value of windows\Media\ding.wav. Windows 2000 will actually hard-code the directory to the Windows 2000 installation directory. If at some time the user selects the default sound in the Control Panel's Sounds applet, this one is used.

Additionally, if the user has defined one or more schemes, there will be an entry for each user-defined scheme. The name that is used is a system-generated hash of the user's scheme name. For example, I created a scheme called Peter's Scheme, and Windows 2000 named the relevant subkeys Peter'0.

Scheme names are contained in the HKEY_CURRENT_USER\AppEvents\Schemes\Names subkey. There will be one subkey for each scheme created by users, plus the two default ones: .Default and .None. The .Default subkey is the scheme used to restore the sounds to their default values. The .None subkey is a scheme used to turn off all sounds, which in some situations may be a really good move. There have been times when I wanted to use a really big hammer on someone's speakers. Oh, and by the way, each of these scheme subkeys contains the username for the scheme.

Already we see the possibility to modify the default sounds so that there could be a standard set of sounds for an organization. After all, a company with specialized sounds (for example, any company in the entertainment business) might really want their sounds to be the default sounds.

In the Control Panel's Sounds applet, you can select default sounds in the Schemes section (see Figure 16.1). Select the Windows Default scheme to restore the defaults. In the Sounds applet, you can also select No Sounds to remove all sounds from events.



NOTE Before selecting a scheme and making massive changes, it may be a good idea to save the current settings in a new scheme so that you can back out of an undesired change with only a little work. You can delete schemes that you no longer need in the Sounds applet using the Delete button. Better safe than sorry.

Console

The Console subkey contains information used to configure the default sessions. Each entry sets parameters used for character-based applications; those with their own PIF files will use the PIF file settings rather than the settings in this subkey.

In Table 16.2, we are actually dealing with two-digit values (four of them in each ColorTable entry). a two-digit hex value can represent a value between 0 and 255 (that's 0 and 0xFF in hex). Table 16.2 shows each value entry, a typical value, and what the value entry means.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
TABLE 16.2: CONSOLE SETTINGS FOUND ON MOST WINDOWS SYSTEMS

Value Entry	Typical Value	Description
ColorTable00	0x00000000	An RGB color value that is black as night. RGB is additive, getting lighter as the values increase.
ColorTable01	0x00800000	Dark red.
ColorTable02	0x00008000	Dark green.
ColorTable03	0x00808000	Pea-green color (or a dark yellow, you decide).
ColorTable04	0x00000080	Dark blue.
ColorTable05	0x00800080	Violet.
ColorTable06	0x00008080	Dark cyan.
ColorTable07	0x00c0c0c0	Light gray.
ColorTable08	0x00808080	Darker gray.
ColorTable09	0x00ff0000	Bright red.
ColorTable10	0x0000ff00	Bright green.
ColorTable11	0x00ffff00	Yellow—or a really, really bright pea green.
ColorTable12	0x000000ff	Bright blue.
ColorTable13	0x00ff00ff	Bright violet.
ColorTable14	0x0000ffff	Cyan.
ColorTable15	00x0ffffff	White.
CurrentPage	0x00000000	Page zero is the current page.
CursorSize	0x00000019	The cursor is 25 percent of the character cell in size.
FaceName	(None)	Defines the name of the console font if defined. a default font is selected if none is defined.
FontFamily	0x00000000	Defines the console font family if defined. The default family for the selected font is used if none is defined; typical values include TrueType and Raster.
FontSize	0x00000000	Defines font size; the low word contains the character width; the high word contains the character height. For example, a font 8 × 16 would be 0x00080010.
FontWeight	0x00000000	Defines the weight (bolding) of the font; larger numbers are bolder.

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

FullScreen	0x00000000	A value of 0x00000001 is set if this window is full screen; a value of 0x00000000 is set if the window is not full screen.
HistoryBufferSize	0x00000032	The size of the history buffer in commands; the hex value of 32 is 50 commands in decimal.
InsertMode	0x00000000	A value of 0x00000001 is to use insert mode; 0x00000000 is to use overwrite mode.
NumberOfHistory-Buffers	0x00000004	Describes the number of history buffers used for this command session.
PopupColors	0x000000f5	Describes the color used for a pop-up window if displayed. The first four bits (f in the typical value) are the characters; the next four bits (5 in the typical value) are the foreground color. These values are indexes to the color values defined in this table.
QuickEdit	0x00000000	Setting the value to 0x00000001 enables QuickEdit; setting it to 0x00000000 disables QuickEdit. QuickEdit allows quick cut and paste to the Clipboard.
ScreenBufferSize	0x00190050	Describes the screen buffer size. In the example, 0x0019 equals 25 in decimal and 0x0050 equals 80 in decimal; therefore, the default screen buffer size is 25 × 80 in size. Other common sizes are 50 × 80 (0x00320050), or 43 × 80 (0x002b0050).
ScreenColors	0x00000007	Describes the index to colors for the screen. The next-to-last digit is the index for characters, and the last digit is the index for the background.
WindowSize	0x00190050	Describes the window size. In the example, 0x0019 equals 25 in decimal and 0x0050 equals 80 in decimal; therefore, the default screen buffer size is 25 × 80 in size. Other common sizes are 50 × 80 (0x00320050), or 43 × 80 (0x002b0050).

[Previous](#)
[Table of Contents](#)
[Next](#)

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)

[Products](#) |
 [Contact Us](#) |
 [About Us](#) |
 [Privacy](#) |
 [Ad Info](#) |
 [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

Click to access!



TIP Lazy and don't want to convert between hex and decimal using your fingers and toes—or just can't take off your shoes? The Windows Calculator program will convert between hex and decimal with ease. Just start Calculator and select View ∅ Scientific.

Hexadecimal and Colors

What the heck is hex? *Hexadecimal* numbers, usually just called *hex* for short, are expressed in base 16. To show a hex number, we use the numeric digits 0 through 9 and the letters A through F.

Computers are binary. They know only two number values: either 1 (on) or 0 (off). a single datum of computer data is called a *bit*, which represents either 0 or 1, and no other value in between.

In computers, numbers are stored in bytes, each comprising 8 bits. a byte's value may range from 0 to 255. In hex, that value range is 0x00 to 0xFF. The prefix 0x precedes hexadecimal numbers, and the letters may be either uppercase or lowercase (case doesn't matter).

Two bytes together (16 bits) form what is called a "WORD" (usually, but not always, written in uppercase). a WORD, if unsigned, may represent a value from 0 to 65535. a signed WORD value represents a value of -32767 to 32767.

Four bytes together (32 bits) form what called a "DWORD," short for double word. A double WORD, if unsigned, may represent a value from 0 to 4294967295. a signed double WORD represents a value from 2147483647 to -2147483647.

Oftentimes programmers try to fit as much information as possible into a WORD or DWORD value. History has shown how this can backfire, but for some data, this technique works well. Color values are a case where three sets of values (one each for red, green, and blue) fit within the DWORD's four bytes.

Oh, and one more bit of confusion: Half a byte, 4 bits, is called a *nibble*. A nibble can hold a value between 0 and 15. Several registry entries use 4-bit nibble values.

Control Panel

The Control Panel subkey in the registry is where many of the Control Panel applets store settings and defaults. The number of sections may vary depending on installed components. Things that affect the number of sections include special mouse support, screen savers, and installed optional components. There may be some differences between a Windows 2000 Server and a Windows 2000 Professional installation. This subkey also includes data stored in the Win.ini and System.ini files on Windows 3.x and earlier.

The sections that show up in many registries include:

Accessibility: Contains Windows 2000's features for users who require special support due to physical limitations—items such as a special keyboard, mouse, or sounds, and general support.

Appearance: Holds settings for Windows 2000's appearance and the schemes used for display configuration.

Cache: Unknown—don't you just hate it when there is a component that is both undocumented and apparently unused? I checked every system I could, and the Cache subkey was empty on all systems—both Server and Professional.

Colors: Contains the colors for buttons, text—just about everything displayed to the user.

Current: Contains the currently loaded color scheme.

Custom Colors: Contains any user-defined color schemes.

Desktop: Holds the Desktop configuration, colors, spacing metrics—everything about what the screen displays.

Don't load: Holds the names of any .cpl files that are not to be loaded if found on the system. These .cpl files are usually not loaded as they are incompatible with either Windows 2000 or another component.

Input Method: Contains information about the user's hotkey definitions.

International: Contains items dealing with the computer's location (country), including sorting orders.

IOProcs: Holds the Media View File System control.

Keyboard: Contains configurations for the keyboard, such as initial state of the toggle keys for Caps Lock, Num Lock, and Scroll Lock; and delay and repeat rates.

Microsoft Input Devices: Contains information about the interaction between input devices (such as the system mouse) and Windows 2000 components.

MMCPL: Contains Multimedia Control Panel settings.

Mouse: Contains mouse settings, such as speed, tracking, and other settings.

Patterns: Contains Windows 2000's patterns used to create backgrounds, such as Boxes, Critters, Diamonds, and so on.

Powercfg: Information about the power configuration settings, including definitions of the various power configuration policies.

Screen Saver.3DflyingObj: Contains configurations for this screen saver.

Screen Saver.3DPipes: Contains configurations for this screen saver.

Screen Saver.Bezier: Contains configurations for this screen saver.

Screen Saver.Marquee: Contains configurations for this screen saver.

Screen Saver.Mystify: Contains configurations for this screen saver.

Screen Saver.Stars: Contains configurations for this screen saver.

Sound: Contains information about sounds.

Sounds: Contains one entry for something called SystemDefault.

In the remainder of this section, we'll look at some of some entries that either seem interesting or can set data that cannot be set elsewhere. Most entries can be set using the Control Panel.

Accessibility

The concept of allowing Windows to be accessible to users who have special needs is relatively new. Windows 95 was the first version of Windows to offer accessibility configurations. Windows NT 4, released after Windows 95, followed suit.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The Accessibility key is subdivided into eight subkeys:

Keyboard Response: Entries include the following:

AutoRepeatDelay: This entry has a default value of 1000, or one second. Increasing this value increases the wait time before the keyboard auto-repeat kicks in.

AutoRepeatRate: This entry has a default value of 500. Increasing this value increases the repeat rate.

BounceTime: This entry has a default value of 0. This value specifies the amount of time to ignore a keystroke after pressing and releasing a key. This helps eliminate false double keystrokes.

DelayBeforeAcceptance: This entry has a default value of 1000. Increasing it increases the amount of time that a key must be pressed before it registers as being pressed. Changing the default here is useful if a user has a tendency to hit keys by mistake.

Flags: This is a character field containing a default value of 82. It enables or disables the previously discussed flags.

MouseKeys: Entries include the following:

MaximumSpeed: This entry has a default value of 80. It limits the maximum speed, in pixels per second, that the mouse cursor will move when a mouse-movement key is pressed.

TimeToMaximumSpeed: This entry has a default value of 3000, three seconds, and is used to determine the amount of time required for the mouse pointer to reach full speed (specified in MaximumSpeed) when a mouse-movement key is held down.

Flags: This is a character field containing a default value of 18. It is used to disable and enable the previously discussed flags.

SerialKeys: This subkey holds settings for a special input device connected to a serial port that emulates the keyboard and mouse on the computer. Typically, people who are unable to use standard keyboards take advantage of these devices. Each device has a unique configuration, and registry entries will be specific to the device. For systems that do not have serial keyboard/mouse emulation devices configured, SerialKeys will have no entries. Otherwise, it has the following objects:

ActivePort: This entry sets the COM port used and has a default of COM1.

Baud: This entry sets the serial speed, in baud, displayed as a hexadecimal number by default.

Port: This entry defines the COM port supported.

Flags: This entry contains a default value of 3, indicating that the feature is supported.

ShowSounds: This subkey holds the settings for the Sound tab of the Accessibility Options

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

applet.ShowSounds has a single registry entry and no other options:

On: A value of 0 is off, and a value of 1 is on.

SoundSentry: Inside the SoundSentry subkey, there are three or more value entries. Four are listed here:

FSTextEffect: This is full-screen text window (command- or character-based application) support. a value of 0 is off; non-zero is on.

TextEffect: This value is probably unused. The Accessibility Options applet in Control Panel will reset TextEffect to 0 if it is non-zero.

WindowsEffect: A value of 0 indicates no Windows effects; 1 indicates flash active caption bar; 2 indicates flash active window; 3 indicates flash Desktop.

Flags: A value of 2 is off; a value of 3 indicates that SoundSentry (a visual indication that a sound is being made) is on.

StickyKeys: With the StickyKeys feature, a key (such as Ctrl, Shift, or Alt) that normally is pressed at the same time as another key to modify the second key's meaning can be set to "stick" on until the next key is pressed. This avoids having to press two keys at the same time, a process that is difficult for some users. There is a single entry in StickyKeys called Flags, a REG_SZ variable that contains a decimal number that represents the Flags value. Table 16.3 shows the known bits.

How's That Number Again?

The value stored in the Windows 2000 registry is a text string containing a number. This number is in decimal. The default value, 510, indicates that the flags at 0x1FE are turned on. Looking at Table 16.3, and the default value of 510 (0x000001FE), we see that the set flags are:

- 2 (0x00000002)—StickyKeys is available (cannot be changed from the StickyKeys dialog box).
- 4 (0x00000004)—the StickyKeys hotkey is available.
- 8 (0x00000008)—displays a confirmation dialog when activating StickyKeys using the hotkey (cannot be changed from the StickyKeys dialog box).
- 16 (0x00000010)—Windows plays a siren sound when the hotkey turns on or off the StickyKeys feature.
- 32 (0x00000020)—enabling StickyKeys displays an indicator.
- 64 (0x00000040)—Windows plays a sound when a StickyKeys modifier is pressed.
- 128 (0x00000080)—pressing the modifier key twice locks it; a third press unlocks it (normally Windows automatically releases the modifier key after one use).
- 256 (0x00000100)—Windows turns off StickyKeys when a modifier key and another key are pressed simultaneously (cannot be changed from the StickyKeys dialog box).

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

TABLE 16.3: STICKYKEYS FLAGS BITS DEFINED

Flags	Bits	Flags (in decimal)	Windows NT 4 Compatible	Windows 2000 Compatible	Can Be Set?	Description
0x00000001	1		ü	ü	û	StickyKeys is turned on.
0x00000002	2		ü	ü	û	StickyKeys is available. (This setting cannot be changed from the StickyKeys dialog box.)
0x00000004	4		ü	ü	û	The StickyKeys hotkey is available.
0x00000008	8			ü	û	Displays a confirmation dialog box when StickyKeys is activated using the hotkey. (This setting cannot be changed from the StickyKeys dialog box.)
0x00000010	16		ü	ü	û	Tells Windows to play a siren sound when the hotkey is used to turn on or off the StickyKeys feature.
0x00000020	32			ü	û	Enabling StickyKeys displays an indicator.
0x00000040	64		ü	ü	û	Tells Windows to play a sound when a StickyKeys modifier is pressed.



Desktop Library

Click to access!

0x00000080 128	ü	ü	û	Allows locking the modifier key by pressing twice. Normally the modifier key is automatically released after use. A third press unlocks the modifier key.
0x00000100 256	ü	ü	û	Whenever a modifier key is pressed simultaneously with a another key, StickyKeys is turned off. (This setting cannot be changed from the StickyKeys dialog box.)
0x00000200 512		ü	û	Unknown.
0x00000400 1024		ü	û	Unknown.
0x00000800 2048		ü	û	Unknown.
0x00001000 4096		ü	û	Unknown.
0x00002000 8192		ü	û	Unknown.
0x00004000 16384		ü	û	Unknown.
0x00008000 32768		ü	û	Unknown.
0x00010000 65536		ü		Left Shift key is currently locked.
0x00020000 131072		ü		Right Shift key is currently locked.
0x00040000 262144		ü		Left Ctrl key is currently locked.
0x00080000 524288		ü		Right Ctrl key is currently locked.
0x00100000 1048576		ü		Left Alt key is currently locked.
0x00200000 20971520		ü		Right Alt key is currently locked.
0x00400000 4194304		ü		Left Windows key is currently locked.
0x00800000 8388608		ü		Right Windows key is currently locked.
0x01000000 16777216		ü		Left Shift key is currently latched.
0x02000000 33554432		ü		Right Shift key is currently latched.
0x04000000 67108864		ü		Left Ctrl key is currently latched.
0x08000000 34217728		ü		Right Ctrl key is currently latched.
0x10000000 68435456		ü		Left Alt key is currently latched.
0x20000000 536870912		ü		Right Alt key is currently latched.
0x40000000 1073741824		ü		Left Windows key is currently latched.
0x80000000 2147483648		ü		Right Windows key is currently latched.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)


NOTE Windows 2000 adds a number of new values for StickyKeys. (Windows 98 supports these values, too.) A total of seven values are currently unknown, and probably not used.

TimeOut: There are two entries in TimeOut that control when the accessibility options are turned off. They are based on non-use for a certain period of time.

Flags: A value of 1 is on; a value of 3 indicates that Windows plays a siren sound when the time-out period expires.

TimeToWait: The value, in milliseconds, that the computer has been idle before accessibility options are turned off. Five minutes is 300000.

ToggleKeys: There is a single value in ToggleKeys called Flags. ToggleKeys is a feature that works like StickyKeys (described above). Unlike with StickyKeys, toggled keys must be manually reset, as they do not reset after the next keystroke. A bitmapped value is used that has six bits defined (see Table 16.4).

TABLE 16.4: TOGGLEKEYS FLAGS BITS DEFINED**Flags bits Windows NT 4 Compatible Windows 2000 Compatible Description**

Flags bits	Windows NT 4 Compatible	Windows 2000 Compatible	Description
0x00000001	ü	ü	If set, then the ToggleKeys feature is turned on.
0x00000002	ü	ü	If set, then the ToggleKeys feature is available.
0x00000004	ü	ü	If set, the user is able to turn on and off the ToggleKeys feature by pressing the Num Lock key for eight seconds.
0x00000008		ü	Windows displays a dialog box to confirm activation using the hotkey.
0x00000010	ü	ü	Windows plays a siren sound when ToggleKeys turns on or off.
0x00000020			Not supported; this feature would provide a visual indicator of the ToggleKeys state.



Appearance

What Windows 2000 looks like is contained in the Windows 2000 schemes used for display configuration. Under the Appearance subkey, there is a single subkey called Schemes. In Schemes, there are keys, all REG_BINARY, containing definitions of the Windows 2000 standard color schemes, such as Lilac, Maple, Wheat, Windows Standard, and so on.

Each scheme in this subkey is loaded in the Control Panel's Display applet, which is also accessible from the Desktop's properties menu. Looking in the Appearance tab, there is a drop-down list to select a scheme from.

It is quite possible to hack a scheme from the registry, although many of the parts of the scheme may be modified more easily in the Display Properties dialog box. Once modified, a new scheme may be saved for later reloading as needed.

Cache

The Cache subkey seems to control how Control Panel displays its icons. Many Windows 2000 users do not have any entries in this subkey, while others do. An example of the Cache subkey is shown in Microsoft's Knowledge Base article Q150541, which you'll find at <http://support.microsoft.com/support/kb/articles/q150/5/41.asp>.

Colors

The Colors subkey contains the colors for buttons, text, and just about everything displayed to the user. Keep in mind that more colors may be defined as more applications and components are installed on Windows 2000.

Each entry listed in this subkey has a string containing three numbers representing the red, green, and blue color levels, as shown below. As the color value increases, the color becomes lighter, so that a value of 127 0 0 is a dark red, and a value of 255 0 0 is a bright red.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

ActiveBorder	92	192	192
ActiveTitle	0	0	128
AppWorkSpace 255	255	255	
Background	255	255	255
ButtonFace	192	192	192
ButtonHighlight	255	255	255
ButtonShadow	128	128	128
ButtonText	0	0	0
GrayText	128	128	128
Hilight	0	0	128
HilightText	255	255	255
InactiveBorder	192	192	192
InactiveTitle	192	192	192
InactiveTitleText	0	0	0
InfoText	0	0	0
InfoWindow	255	255	255
Menu 255	255	255	
MenuText	0	0	0
Scrollbar	192	192	192
TitleText 255	255	255	
Window	255	255	255
WindowFrame	0	0	0
WindowText	0	0	0

Current

The subkey Current contains the currently loaded color scheme. One key, called Color Schemes, contains the name of the color scheme. also check HKEY_CURRENT_USER\ControlPanel\Appearance\Schemes for a list of schemes installed on the computer.

Custom Colors

The Windows 2000 common dialog box called Colors allows you to define and save up to 16 custom color definitions. These custom colors are stored in the subkey Custom Colors, in entries named ColorA through ColorP. Each entry contains a six-digit string, in hexadecimal, nominally in RGB, for each custom color. The default value for each color entry is FFFFFFFF, or white.



Desktop

The configuration of the user's Desktop is contained in the Desktop subkey. This key contains between 25 and 50 different entries. Many of these items (see Table 16.5) can be adjusted in the various properties dialog boxes, but some must be changed directly from the registry.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
TABLE 16.5: DESKTOP SETTINGS FOUND ON MOST WINDOWS SYSTEMS

Entry	Typical Value	Description
ActiveWndTrkTimeout	dword:00000000	New to Windows 2000, this value is currently undefined.
AutoEndTasks	0	Sets the automatic task-ending mode that controls whether the system automatically ends a timed-out task without displaying a warning or prompt dialog box.
CaretWidth	dword:00000001	New to Windows 2000, this value is currently undefined.
CoolSwitch	1	The fast task-switching mode; set to 0 to disable. In Windows NT 4, the feature is always enabled.
CoolSwitchColumns	7	Sets the number of columns of icons in the Alt+Tab dialog box.
CoolSwitchRows	3	Sets the number of rows of icons in the Alt+Tab dialog box.
CursorBlinkRate	530	Sets the time between blinks of the cursor, in milliseconds; the default value is 530 milliseconds.
DragFullWindows	0	The drag mode in Windows 2000 that supports either full-window dragging or outline dragging; a value of 1 indicates that full-window dragging is enabled.
DragHeight	2	Sets the vertical size of the dragging box required before the mouse detects a drag operation.
DragWidth	2	Sets the horizontal size of the dragging box required before the mouse detects a drag operation.
FontSmoothing	2	Font smoothing makes certain fonts easier to read on high-resolution color adapters. It is set in the Plus tab of the Display Properties dialog box, under Smooth Edges of Screen Fonts.
ForegroundFlashCount	dword:00000003	New to Windows 2000, this value is currently undefined.
ForegroundLockTimeout	dword:00030d40	New to Windows 2000, this value is currently undefined.
GridGranularity	0	A grid that helps align objects on the Desktop may be enabled by setting this to any non-zero value.
HungAppTimeout	5000	Sets the time, in milliseconds, before a hung application (one that does not respond) will cause Windows 2000 to display a dialog box to prompt the user to either wait or kill the application.
IconSpacing	75	Sets the icon spacing granularity for the Desktop.

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

IconTitleFaceName	MS Sans Serif	Sets the icon font name.
IconTitleSize	9	Sets the size of icon titles.
IconTitleStyle	0	Sets the icon title style.
IconTitleWrap	1	Sets the entry that controls whether icon titles will wrap or be displayed on only one line.
LowPowerActive	0	New to Windows 2000, this value is currently undefined.
LowPowerTimeOut	0	New to Windows 2000, this value is currently undefined.
MenuShowDelay	400	The delay time set before showing a cascading menu; typical values are from 0 to 400, although values can be higher.
PaintDesktopVersion	dword:00000001	New to Windows 2000, this value is currently undefined.
Pattern	(None)	The pattern used under icon labels or exposed areas of the Desktop that the Desktop wallpaper doesn't cover; set in the Background tab of the Display Properties dialog box.
PowerOffActive	0	New to Windows 2000, this value is currently undefined.
PowerOffTimeOut	0	New to Windows 2000, this value is currently undefined.
ScreenSaveActive	1	If this value is set at 1, the screen saver will be displayed when the system has been inactive for a longer amount of time than is specified in ScreenSaveTimeOut.
ScreenSaverIsSecure	0	If this value is set at 1, the screen saver will prompt for a password.
ScreenSaveTimeOut	1500	Sets the amount of time the computer is inactive, in seconds, before displaying the screen saver.
SCRNSAVE.EXE	(NONE)	The name of the current screen saver.
TileWallpaper	0	The wallpapering mode. If the value is set at 0, the wallpaper is centered using only a single copy. If the value is 1, the wallpaper is tiled starting in the upper left corner.
UserPreferencesMask	hex:9e,3e,00,80	New to Windows 2000, this value is currently undefined.
WaitToKillAppTimeout	20000	Sets the amount of time that elapses, in milliseconds, before notifying users of any applications that are not responding properly when a logoff or shut-down command is received.
Wallpaper	(None)	Sets the name of the wallpaper file; a bitmap file.
WallpaperStyle	0	New to Windows 2000, this value is currently undefined.
WheelScrollLines	3	The number of lines that the Microsoft wheel mouse will scroll when the wheel is turned; the default value of 3 may be too much for some applications.

[Previous](#)
[Table of Contents](#)
[Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) |
 [Contact Us](#) |
 [About Us](#) |
 [Privacy](#) |
 [Ad Info](#) |
 [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

As the installation of Windows 2000 ages and more optional components are added, the number of entries in the Desktop subkey will increase. Many of the possible entries are self-explanatory. Generally, modifying a value won't cause a computer to crash, although the results may be unpleasant.

A subkey under Desktop, named WindowMetrics, contains entries that define the physical attributes for a number of the components that make up the Desktop:

Entry	Typical Contents	Description
BorderWidth	1	Sets the width of a resizable window's border
CaptionFont	A logical font structure defined a font to be used	Sets the font used for captions
CaptionHeight	-270	Sets the height of the characters
CaptionWidth	-270	Sets the width of the characters
IconFont	Sets the font used for icons	
IconSpacing	75	Sets the space between each icon
IconTitleWrap	1	Sets whether an icon's title wraps (non-zero) or not
IconVerticalspacing	-1125	Sets the spacing between rows of icons
MenuFont	A logical font structure defining a font to be used	Sets the font used for menus
MenuHeight	-270	Sets the height of the characters
MenuWidth	-270	Sets the width of the characters
MessageFont	A logical font structure defining a font to be used	Sets the font used for messages
ScrollHeight	-240	Sets the height of the characters
ScrollWidth	-240	Sets the width of the characters
Shell Icon BPP	16	Sets the number of bit planes for icons
ShellIconSize	32	Sets the size of icons in the shell
SmCaptionFont	A logical font structure defining a font to be used	Sets the font used for small captions
SmCaptionHeight	-180	Sets the height of the characters
SmCaptionWidth	-180	Sets the width of the characters


Desktop Library
[Click to access!](#)

StatusFont A logical font structure defining a font Defines the status bar font to be used

International

The International subkey stores items dealing with the computer's location (country), including sorting orders. Most of these entries are set in the Control Panel using the Regional Settings Properties applet.

Generally, there is little need to manually set anything in International. The Regional Settings Properties dialog box covers each entry fully and includes error checking.

IOProcs

The IOProcs subkey contains a reference to a single file, Mvfs32.dll, which is not found on any system that I have checked. Mvfs32.dll is the Media View File System .dll used by some applications to view media files. There is a strong probability that this file system and IOProcs are not used by more recent applications. There are two Media View File System .dlls supplied with Windows 2000: Mvfs13n.dll and Mvfs14n.dll. Like the Cache subkey mentioned earlier in this chapter, this is almost a mystery key. One Microsoft Knowledge Base entry does document a fix for a problem with Encarta 95 and Windows NT 3.5 that requires an entry in IOProcs for M12 = mvfs1232.dll; this is the only information available.

Keyboard

The Keyboard subkey stores configurations for the keyboard, such as the initial state of the toggle keys—Caps Lock, Num Lock, and Scroll Lock—and the delay and repeat rates.

A typical system will have these three entries:

InitialKeyboardIndicators: This is automatically set by Windows 2000 when users log off or when the system is shut down. It preserves the previous state of the Num Lock key. 0 turns off Num Lock when the user logs on, and 2 turns on Num Lock when the user logs on.

KeyboardDelay: This is the delay, when a key is held down, before the key auto-repeats. Values between 0 and 3 are accepted, with 0 being a delay of 250 milliseconds, and 3 being a delay of 1 second. (These times are approximate.)

KeyboardSpeed: This is the speed at which a key auto-repeats. Choose a value between 0, which repeats at two characters per second, and 31, which repeats at 30 characters per second.

MMCPL

Some ODBC (Open Database Connectivity) and Multimedia Control Panel settings are stored in the MMCPL subkey. Many computers do not have any entries in MMCPL. Typical entries might include the following:

```
mlcfg32.cpl=G:\PROGRA~1\COMMON~1\System\MAPI\1033\nt\mlcfg32.cpl
mlcfq32.cpl=G:\PROGRA~1\COMMON~1\System\MAPI\1033\nt\mlcfq32.cpl
NumApps=20
H=230
W=442
X=88
Y=84
```

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

It is possible to have Control Panel multimedia settings in other directories, with the exception of %systemroot%\System32, by specifying their names and paths in the MMCPL subkey.

Mouse

Mouse settings, such as speed and tracking, are set in the Mouse subkey. Typical settings include those shown in Table 16.6.

TABLE 16.6: MOUSE SETTINGS FOUND ON MOST WINDOWS SYSTEMS

Entry	Typical Value	Description
SwapMouseButton	0	If the value is 1, the right and left mouse buttons are swapped.
DoubleClickSpeed	500	Sets the amount of time between consecutive clicks of the mouse button for it to be considered a double-click.
DoubleClickHeight	4	Sets the amount of movement allowed (vertical) for a double-click to be valid.
DoubleClickWidth	4	Sets the amount of movement allowed (horizontal) for a double-click to be valid.
MouseThreshold1	6	Sets the motion factor that, when factored with MouseSpeed, controls the motion of the mouse.
MouseThreshold2	10	Sets the motion factor that, when factored with MouseSpeed, controls the motion of the mouse.
MouseSpeed	1	Sets the speed of the mouse pointer relative to the movement of the mouse.
SnapToDefaultButton	0	When this value is set to 1, the mouse will snap to the default button in dialog boxes.
ActiveWindowTracking	0x00000000	When this value is set to 0x00000001, the active window will always be the one the mouse is positioned on.

Patterns

The Patterns subkey is where patterns used to create backgrounds, such as Boxes, Critters, and Diamonds, are set. For Windows 2000, a pattern is expressed as an 8 × 8 box of color, black for each 1 bit and the



background color for each 0 bit. The first number represents the first, topmost, line in the pattern; the second number represents the second line in the pattern, and so forth.

Each line is a binary representation; for example, the Boxes pattern is:

127, 65, 65, 65, 65, 65, 127, 0

These values are binary numbers, stored in decimal format, as shown below:

Decimal Binary

127	0111 1111
65	0100 0001
65	0100 0001
65	0100 0001
65	0100 0001
65	0100 0001
127	0111 1111
0	0000 0000

You can compare these binary numbers with the Boxes pattern. To do so, use the Edit Pattern button (in the Pattern dialog box) to view the pattern in the pattern editor. This fully shows the relationship between the bits and the pattern.

Be creative; you can cook up new patterns using the pattern editor. Enter a new pattern name, click the Add button, and voila, there is your new pattern. Just remember: it can be hard to be creative using an 8 × 8 cell.

PowerCfg



Windows 2000 introduces power configuration and power savings. Power configurations are set in the Control Panel's Power Options applet. Each of the five supplied power schemes has a default name, and each scheme can be modified and saved by the user.

A default set of power policies is stored in GlobalPowerPolicy, a subkey in the key PowerCfg. another subkey in PowerCfg, PowerPolicies, contains subkeys for each configuration, both Microsoft and user created. The Microsoft power policies are:

Name	Description
Home/Office Desk	A desktop scheme, this is useful when the computer is connected to the power source permanently.
Portable/Laptop	This scheme is useful for notebook and laptop computers for which maximum battery life is important.
Presentation	When performing presentations, this scheme will not allow blanking (turning off) the monitor.
Always On	Typically for network servers, this scheme keeps everything running, without standby. This allows network access to the computer, regardless of whether it is currently being used or not.
Minimal Power Management	Another scheme designed for servers, this one keeps the computer on and at the highest performance level.
Max Battery	For notebooks and laptop computers, using Max Battery will maximize the battery's life.

Screen Saver.3DflyingObj

The Screen Saver.3DflyingObj key holds configurations for the 3D Flying Objects (OpenGL) screen saver. actually, you are able to access these settings from the Screen Saver tab in the Display Properties dialog box. Select the 3D Flying Objects (OpenGL) screen saver and click the Settings button to configure these settings.

Screen Saver.3Dpipes

The Screen Saver.3Dpipes key holds configurations for the 3Dpipes (OpenGL) screen saver.

Screen Saver.Bezier

The Screen Saver.Bezier key holds configurations for the Bezier (OpenGL) screen saver.

Screen Saver.Marquee

The Screen Saver.Marquee key holds configurations for the Marquee (OpenGL) screen saver.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Screen Saver.Mystify

TheScreen Saver.Mystify key holds configurations for the Mystify (OpenGL) screen saver.

Screen Saver.Stars

TheScreen Saver.Stars key holds configurations for the Stars (OpenGL) screen saver.

Sound

Information about basic sounds is contained in the Sound subkey. Two entries that I've found, Beep=yes and ExtendedSounds=yes, both seem to be present on all systems. Beep=yes is used to indicate whether Windows 2000 will make a warning beep when the user attempts to do something that is not allowed.

Sounds

The Sounds subkey contains one value entry called SystemDefault that typically has a value of , (that is just a comma, nothing else). Other entries you might find in the Sounds subkey include the following:

- Enable=1
- SystemAsterisk=chord.wav,Asterisk
- SystemDefault=ding.wav,Default Beep
- SystemExclamation=chord.wav,Exclamation
- SystemExit=chimes.wav,Windows Logoff
- SystemHand=chord.wav,Critical Stop
- SystemQuestion=chord.wav,Question
- SystemStart=tada.wav,Windows Logon

I assume that the Sounds key contains information used by legacy systems. Windows defines each of these sounds for use elsewhere in the registry.

Environment

The Control Panel's System Properties applet contains a tab called Advanced that is subdivided into three sections (Performance, Environment Variables, and Startup and Recovery). Clicking the Environment Variables button displays the Environment Variables dialog box. This dialog box is subdivided into System



Variables and User Variables for <user>, where <user> is the currently logged-on user (see Figure 16.2). any environment variable defined in System Variables will be available to all users, while environment variables defined in User Variables for <user> will only be available to the currently logged-on user.



FIGURE 16.2 The environment for each user is contained in HKEY_CURRENT_USER\Environment.

Notice in Figure 16.2 that the current user is Administrator. When the next user logs on, they will get a different environment.

There is little need to modify the HKEY_CURRENT_USER\Environment section directly. The Control Panel's System Properties applet does a better job of modifying entries in Environment, and using System Properties is much safer than manually editing the registry.



WARNING Avoid the urge to modify existing system variables unless you understand the ramifications of making such a change. For instance, changing the entry NUMBER_OF_PROCESSORS from 1 to 2 won't give you an extra CPU.

EUDC

EUDC, for End User Defined Characters, contains information about special characters users have created for their specific needs. This feature is supported in the Japanese, Chinese, Hangeul, and Johab character sets.

Identities

Configuration for some software, specific to the user, is stored in the Identities subkey. This subkey contains, on a typical system, information for Outlook Express (stationary lists, username, etc.).

Keyboard Layout

The Keyboard Layout subkey allows users to change keyboard configurations. Since different languages may have different layouts (usually for special symbols, such as currency), Windows 2000 allows users to change the keyboard layout using the Control Panel's Keyboard and Regional Options applets.

Configuring Windows 2000 for multiple languages is easy. Open the Control Panel's Regional Options applet, and use the General tab to change language settings.



NOTE You can have more than two input locales, although it is unusual to have many different locales defined.

You can define a hotkey to switch locales (in the Input Locales tab). The default hotkey is Left Alt+Shift, although you also can use Ctrl+Shift as an alternative hotkey. If you do not desire a hotkey, uncheck Enable Key Sequence. If you select the "no hotkey" option, it would be a good idea to check the Enable Indicator on Taskbar option so that you can use the Taskbar to switch input locales.

Now, back to our currently scheduled programming ...

In HKEY_CURRENT_USER\Keyboard Layout, there are the following three subkeys:

Preload: This subkey contains the keyboard layouts preloaded by Windows. Use the Control Panel's Regional Options applet, a hotkey, or the Taskbar to select the layout.

Substitutes: Any key substitutes will be defined in the Substitutes subkey. Key substitutes typically use the Dvorak keyboard layout. In Substitutes, a key named with the original locale will be substituted with the value of the substituting layout. For instance, the United States English locale is 409. In addition, 00000409 = 00000809 substitutes British English on the United States English locale.

Toggle: This subkey contains a single key whose data value will be 0 if no hotkey is defined, 1 if Left

Alt+Shift is defined as the hotkey, and 2 if Ctrl+Shift is defined as the hotkey.



TIP Tired of QWERTY? A different keyboard layout can be most useful if QWERTY is not your thing. One type of keyboard layout is the Dvorak layout. Said to improve typing proficiency by a great deal, Dvorak has a slowly growing band of supporters. To select the Dvorak layout, select an input locale in Regional Options, and click on Properties to modify the layout.

Generally, all modifications to HKEY_CURRENT_USER\Keyboard Layout should be done using the Input Locales tab of either the Regional Options applet or the Keyboard applet.

Network

HKEY_CURRENT_USER\Network contains configuration information for each network drive that the user has permanently mapped. Under Network you will find a subkey for each mapped drive letter.

For instance, if a user on My Network Places selects a server, selects a share, right-clicks the share, and selects Map Network Drive in the pop-up context menu, the Map Network Drive dialog box (see Figure 16.3) will appear. This allows the user to select which drive letter is mapped to the network share. This subkey also contains the following:

- The unmodifiable path
- A selection named Different User Name, allowing access to the drive as another user
- A check box called Reconnect at Logon, important because unchecking it will mean that the share is available only for the current session



FIGURE 16.3 The Map Network Drive dialog box



NOTE Any drives mapped without the Reconnect at Logon attribute will not be loaded into the HKEY_CURRENT_USER\Network subkey.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

You may map any unused drive letter to a network drive, using either the current user ID or another user ID.

Once you select a drive (H: in our next example) and click the OK button, the registry's HKEY_CURRENT_USER\Network subkey will have F as the new subkey. The F subkey contains five value variables:

ConnectionType: The value of 0x1 is for drives; 0x2 is for printers.

ProviderName: The network provider, typically Microsoft Windows Network (for Microsoft networking).

ProviderType: The network provider's type is 0x20000 for Microsoft Windows networking.

RemotePath: The UNC path to the network share; \\Dora\G is the share named G on the server whose name is Dora.

UserName: The name of the user for whom the share is established; it is either the current user's name or the name the user specifies in the Map Network Drive dialog box (see Figure 16.3). The username syntax consists of *domain\user*, where *domain* is the user's domain and *user* is the user ID.

Again, as with many other HKEY_CURRENT_USER entries, it is easy to manipulate the entries in Network without using the registry editor. If you have problems deleting a connection, one fix would be to delete the subkey from the HKEY_CURRENT_USER\Registry subkey.

Shares can be established quickly from a command-line prompt by using the command-line syntax:

```
net use H: \\Dora\G
```

Here, H: is the drive letter to be mapped and \\Dora\G is the share name. To get the full syntax of the net use command, type **net use /?** at a command-line prompt.

Printers

The HKEY_CURRENT_USER\Printers subkey contains information about printers—both local and remote. Locally attached printers typically use four subkeys in Printers:

Connections: This subkey contains subkeys for each remotely connected printer. Each subkey is named for the printer and contains two objects: Provider, a variable that contains the name of the driving .dll file; and Server, a variable that contains the name of the server that the printer is attached to.

DevModesPerUser: The user configurations for the printer, regardless of whether the printer is local or remotely attached, are stored in this subkey. Each printer has a key entry, named for the printer, with a binary data value following it.



Desktop Library

Click to access!

DevModes2: The configurations for the printer, regardless of whether the printer is local or remotely attached, are stored in this subkey. Each printer has a value entry, named for the printer, with a binary data value following it.

Settings: This subkey contains binary objects for each attached printer.

RemoteAccess

The RemoteAccess subkey contains connectoids for RAS (Remote Access Service). *Connectoids?* What the heck is a connectoid? A connectoid consists of all the information needed to implement a connection, typically to a remote computer.

The RemoteAccess subkey may include one value variable, called InternetProfile. Typically, the InternetProfile entry contains a null (empty) string.

Software

Information about all installed software is stored in the Software subkey. This information is typically arranged by vendor, although some applications may be in their own subkeys.

In a typical installation of Windows 2000, the Software key has at least the following subkey:

Microsoft: Information about many of the components that are part of Windows 2000 is found in this subkey. On a typical installation, we see about 30 entries. A better-equipped installation could have twice the number of entries. Below is a list of objects found in a typical system:

- Active Setup
- Advanced INF Setup
- Clock
- Command Processor
- Conferencing
- DataEnvironment Designer
- DevStudio
- Disk Administrator
- File Manager
- Full-Text-System
- IEAK
- Internet Explorer
- Java VM
- Microsoft Setup (ACME)
- Multimedia
- NetDDE
- Notepad
- NTBackup
- Outlook Express
- Protected Storage System Provider
- RegEdt32
- Schedule+
- SystemCertificates
- User Location Service
- Visual Basic
- WAB
- WebPost
- Windows
- Windows Help

- Windows NT
- ODBC
- Policies
- VDO

On more mature installations with more installed software, the Software key expands to cover different product lines. Notice that products are arranged by the company that has produced the product or the product's functionality, such as ODBC, rather than by specific product. For example, if there were two Adobe products installed on the computer, the Adobe key would have information about both products. Here is a list of subkeys found on a mature installation:

- Adobe
- Canon
- Dragon Systems
- Federal Express
- Forte
- Inetstp
- Microsoft
- Netscape
- ODBC
- Policies
- Qualcomm
- VDO
- Wang

Unicode Program Groups

The Unicode Program Groups subkey contains information used by Program Manager. The question is, of course, who uses Program Manager anymore?

The subkeys found in Unicode Program Groups are in a binary format that is difficult to edit. Have I ever seen entries in the Unicode Program Groups subkey? No, not yet. Wait a minute, I've never actually run Program Manager with Windows 2000, either. Maybe I'll give it a try...

After running Program Manager and creating a couple of personal groups and a few common groups, I now have entries in the Unicode Program Groups key. as mentioned before, these entries are in binary format and are complex structures. as Microsoft recommends, it is best to edit these entries using Program Manager.



NOTE Actually, users who upgraded from Windows NT 3.x may have entries in the Unicode Program Groups subkey; however, this subkey is unused by Windows NT version 4 or later unless the user has configured or run Program Manager (ProgMan.exe).

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)
Windows 3.1 Migrations Status

The Windows 3.1 Migration Status subkey is contained only in systems upgraded from Windows NT 3.x. (There may be a few of these installations remaining.) The subkey contains keys that are used to show the status of conversion of the Program Manager Group (.grp) files and associated initialization (.ini) files that have been converted to Windows 2000 version 4 format. Deleting this value, and not the Windows 3.1 Migration Status subkey, causes Windows 2000 to attempt to convert the Program Manager Group files when Windows 2000 restarts. This re-conversion may change the Start menu substantially, but should not cause serious damage.

Volatile Environment

Typically, the Volatile Environment subkey contains only one entry, a value called LOGONSERVER. It contains a string with the logon server (the server the user is currently logged on to). For example, my logon server is \\DORA.

HKEY_USERS

The HKEY_USERS hive contains information about each active user who has a user profile. There are a minimum of two keys in the HKEY_USERS hive: .DEFAULT and the ID for the currently logged-on user.

Hints and Kinks from the Experts

In this chapter's hints and kinks, we cover a few hints on using the two registry editors.

How Can I Audit Changes to the Registry?

With RegEdt32, it is possible to set auditing on certain parts of the registry. I should caution you that any type of auditing is a very sensitive subject lately, and you may want to add some sort of warning to let people know that their changes are being audited.

1. Start the registry editor (Regedt32.exe).
2. Select the key you wish to audit—for example, HKEY_LOCAL_MACHINE\Software.
3. Select Security Auditing.
4. Check the Audit Permission on Existing Subkeys option if you want to audit subkeys as well.
5. Click the Add button and select the users you want to be audited. Click Add, and then click OK.
6. Once there are names in the Names box, you can select which events to audit and whether to audit

success or failure.

7. After filling in all the information, click OK.

You will need to make sure you enable auditing for file and object access. To do so, select User Manager Polices Audit.

To view the information, use the Event Viewer program and look at the Security information.

(Courtesy of John Savill.)

Activate Screen Saver If No One Logs On

To activate a screen saver when there is no logged-on user, follow these steps:

1. Use RegEdit to edit the subkey HKEY_USERS\DEFAULT\Control Panel\Desktop.
2. Double-click ScreenSaveActive and set it to 1.
3. Double-click SCRNSAVE.EXE and enter the full path to the screen saver you want to use, such as scrsave.scr or sstars.scr.
4. Double-click ScreenSaveTimeOut and enter the number of seconds of inactivity required before activation of the screen saver begins.

You will need to reboot for this to become effective.



WARNING Never use anything other than the blank screen saver (scrsave.scr) on a server because it will steal needed cycles.

(Courtesy of Jerold Schulman.)

Define Initial Settings for New Users

In this chapter, you have seen hacks that may be applied to the HKEY_CURRENT_USER hive. any hack that you can make to HKEY_CURRENT_USER can be made to the default user hive, as well.

To modify the default user hive, follow these steps:

1. Start RegEdt32.
2. Highlight the HKEY_USERS window.
3. Select Registry Load Hive.
4. Select the NTUSER.DAT file, usually from the %windir%\Profiles\Default User directory.
5. Type **NTUSER** in the Key Name dialog box.
6. Now you can add or modify any key or value within this hive. When you finish, highlight NTUSER and select Security Permissions. add read permission to the Everyone group. Check the Replace Permission on Existing Subkeys check box and click OK.
7. Select Registry Unload Hive and exit RegEdt32.
8. Copy the profile to the NetLogon share on the PDC, which is usually at C:\%windir%\System32\Repl\Export\Scripts.

When a new user logs on, they will receive the default profile.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CHAPTER 17

Introduction to HKEY_LOCAL_MACHINE

JUMP TO TOPIC

FEATURING:

<i>HKEY_LOCAL_MACHINE</i>	434
<i>Hints and kinks from the experts</i>	461

The HKEY_LOCAL_MACHINE hive contains information about the system as it is currently running. This hive also holds information about users, groups, security, and installed software. HKEY_LOCAL_MACHINE is divided into five subkeys:

Hardware: Information about currently used hardware is contained in the Hardware subkey.

SAM: The Security Accounts Manager (SAM) defines all users who may use the computer. If the computer is a PDC/BDC, then the domain information is contained in the SAM subkey also.

Security: The Security subkey contains information about cached logons, policy, special accounts, and RXACT (registry transaction package).

Software: Software information, a superset of the information contained in HKEY_CURRENT_USER, is located in the Software subkey.

System: The system definitions, control sets, information about removable media (CD-ROMs), and information about Windows 2000 setup are contained in the System subkey.

So, you've backed up your registry before starting this chapter, right? Let's dig in and see what's there.



WARNING If you change *anything* in the HKEY_LOCAL_MACHINE hive, there's a good chance you'll trash your system registry. Really: HKEY_LOCAL_MACHINE is probably the most critical part of the registry, so *back up before you touch it*—before you even browse it!

HKEY_LOCAL_MACHINE

As I mentioned above, HKEY_LOCAL_MACHINE has five sections. Each section is separate, and we'll deal with each separately. Remember that SAM (the Security Accounts Manager) and Security actually cover



Desktop Library
Click to access!

different aspects of security, so don't confuse 'em.

Hardware

The Hardware subkey describes the system's hardware. Most everything in the Hardware subkey is set during boot-up by the NTDETECT.COM program or by the ARC (Advanced RISC Computer) database for users running Windows 2000 on RISC computers.

The Hardware subkey contains three subkeys:

Description: This subkey contains information about the processor, math coprocessor, and multifunction adapters (devices such as the PCI bus, Plug-and-Play BIOS, and ISA bus).

DeviceMap: This subkey holds subkeys for most devices, such as the keyboard, mouse, serial ports, and video.

ResourceMap: This subkey contains items such as HAL (Hardware Abstraction Layer), keyboard and pointer port(s), serial and parallel ports, SCSI devices (including IDE drives, too), and video information.



NOTE Windows NT 4 has some problems dealing with the PCI bus, as PCI is designed to work integrally with Plug and Play, which Windows NT 4 does not support. Windows 2000 is much better behaved in supporting PnP.

We'll cover these subkeys fully a little later. If you are having problems with your hardware (no, nothing here will help fix a broken keyboard!) interfacing with Windows 2000, it is possible that something in the Hardware key will help you fix the problem.

Before diddling with registry values in HKEY_LOCAL_MACHINE\Hardware directly, first try Windows 2000's Computer Management tool (shown in Figure 17.1), which manages devices, storage, services, and applications.



NOTE Many times you will see subkeys with numbers for names, starting at 0. Often there is only one of these subkeys, but if there are multiple objects of the type described by a key, there will be multiple subkeys numbered from 0 to n , where n is the number of objects of the type described by a key, minus 1. For example, a computer with five drives on a SCSI bus would have subkeys named 0, 1, 2, 3, and 4 in the key

HKEY_LOCAL_MACHINE\Hardware\Description\System\MultifunctionAdapter\3\DiskController\0\DiskPeripheral\.

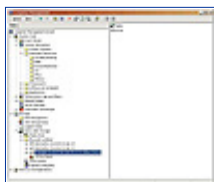


FIGURE 17.1 The Computer Management tool, showing physical locations for CD-ROM drives

Hardware Profiles, or How Can I Make Windows 2000 More Hardware-Friendly?

Windows 2000 is capable of supporting multiple hardware configurations. This functionality is most useful for notebooks and other portable computers. For instance, in the Control Panel's System Properties applet, the Hardware Profiles tab allows a user to create multiple hardware profiles. To create a new configuration, select an existing configuration and click the Copy button. Enter a new name for the configuration (I used the name New Configuration in my example), then click OK. Select the new configuration you just created, then click the Properties button to display the dialog box shown in Figure 17.2; this dialog box's title will vary depending on what you named your new configuration.



NOTE Don't confuse hardware profiles with user profiles. They are very different animals (goodness—is he referring to users as animals?), dealing with completely different areas.

For each configuration, it is possible to define the computer as portable, which is useful when there is docking support. When the option This Is a Portable Computer is checked, the docking state can be set as one of the following:

- The docking state is unknown.
- The computer is docked.
- The computer is undocked.

Windows 2000 also allows making the profile an option when starting the OS. This is useful when you change hardware configurations frequently.



FIGURE 17.2 The properties dialog box for the new hardware configuration

When creating a hardware profile, simply name it to reflect the configuration. For example, if you create a profile for the computer when docked in the docking station, name the profile Docked and make sure you select both This Is a Portable Computer and The Computer Is Docked. When creating a profile for the computer when not docked, name this profile UN-Docked and select This Is a Portable Computer and The Computer Is Undocked.

Sometimes it is necessary to create a profile for when the portable computer is either docked or undocked. Two scenarios can be envisioned: one where the docking state is not important and another where the docking state is important. When the docking state is not important, you can create a single profile with The Docking State Is Unknown option checked. In configurations where the docking state is important, simply create two profiles, one for each state.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

Description

Okay, first, the standard warning: be very careful about making any changes in the Description subkey. Everything in this subkey is set during boot-up by the NTDETECT.COM program or by the ARC database. As this information is volatile (it is regenerated each time Windows 2000 starts by the boot-up process), it is neither practical nor meaningful to modify the data. You can use a program called System Information to view information in these subkeys. Start System Information by selecting Start ⌘ Programs ⌘ Accessories ⌘ System Tools ⌘ System Information.



NOTE In Windows 2000, the System Information program replaces the WinMSD program of Windows NT. System Information also replaces the MSInfo32 program. If you attempt to execute either of these utilities in Windows 2000, System Information will start.

There is a single subkey in Description called System. The subkeys inside System are described in the following sections.

CentralProcessor The CentralProcessor subkey contains subkeys describing each processor found. For uniprocessor systems, a single subkey named 0 contains information about the processor. Multiprocessor computers will have subkeys named from 0 to n , where n is the number of processors minus 1. The following list shows the configuration of an Intel Pentium system. Users with other processors or processors from other suppliers will see some different information, although the keys will be identical.

Component Information: This data value identifies information about the processor.

Configuration Data: This is a REG_FULL_RESOURCE_DESCRIPTOR data object containing data about the processor.

Feature Set: This is a REG_DWORD value describing the computer's feature set.

~MHz: A DWORD value containing the CPU speed, this field may not be present or contain a value for some RISC processors.

Update Status: This is a DWORD value containing a flag indicating the upstate status for this system.

VendorIdentifier: This is the name of the company that manufactured the CPU, as a text string; for example, an Intel Pentium will have "GenuineIntel" as the vendor identifier.

Identifier: This is the CPU model and other CPU-specific identifying information; for example, an Intel Pentium might have the string "x86 Family 5 Model 4 Stepping 3" as the identifier. This string can be used to selectively apply patches to correct known flaws in certain CPUs, for example.

FloatingPointProcessor The FloatingPointProcessor key holds subkeys describing each floating-point math

coprocessor found. For uniprocessor systems, a single subkey named 0 contains information about the coprocessor. Multiprocessor computers will have subkeys named from 0 to n , where n is the number of processors minus 1. The following list shows an Intel Pentium system. Users with other processors or processors from other suppliers will see some different information, although the keys will be identical:

Component Information: This data value identifies information about the processor.

Configuration Data: This is a REG_FULL_RESOURCE_DESCRIPTOR data object containing data about the processor.

Identifier: This is the CPU model and other CPU-specific identifying information; for example, an Intel Pentium might have the string “x86 Family 5 Model 4 Stepping 3” as the identifier. This string can be used to selectively apply patches to correct known flaws in certain CPUs, for example.

MultifunctionAdapter As with the entries in the keys CentralProcessor and FloatingPointProcessor, the entries in MultifunctionAdapter are created either by the hardware recognizer (NTDETECT.COM) for Intel-based systems or by the ARC database found on RISC computers. Inside the MultifunctionAdapter key are subkeys describing the internal structure of the computer, bus structure, PnP (Plug and Play), BIOS (if PnP is installed), and devices installed on these buses.



NOTE Instead of the MultifunctionAdapter key used with ISA, MCA (Micro-Channel Architecture), and PCI bus machines, you may find EisaAdapter if your computer uses the EISA bus, or TcAdapter if your computer uses the TURBOChannel bus architecture. Entries for both EisaAdapter and TcAdapter are similar to those in MultifunctionAdapter; they vary based on what components are installed rather than on bus type.

It would not be practical to describe all subkeys found for every different type of computer in the MultifunctionAdapter subkey. Instead, let's look at a typical system: an Intel motherboard, a PCI bus, an IDE hard drive, and typical peripherals.

The MultifunctionAdapter key contains one or more subkeys. There is one subkey for each bus controller (PnP is counted as a bus and is included, though no devices are assigned to PnP). A typical PCI bus computer (virtually all PCI-based computers also have ISA bus support to allow using legacy interface cards) has three subkeys:

0: The PCI bus, with keys for Component Information, Configuration Data, and Identifier.

1: The PnP BIOS doesn't have a physical bus as such. PnP works with the ISA bus and the PCI bus in the computer. There are keys for Component Information, Configuration Data, and Identifier.

2: The ISA bus, with keys for Component Information, Configuration Data, and Identifier. The ISA bus key contains subkeys for other devices such as disk controllers, keyboards, and printer and serial ports.

Rather than describing all possible entries for the MultifunctionAdapter key, I'm going to suggest that you use RegEdt32 (in read-only mode, please!) and peruse this key. Figure 17.3 shows most of the MultifunctionAdapter key expanded on a typical PCI bus computer.



FIGURE 17.3 RegEdt32 shows the contents of the Multifunction-Adapter subkey.

DeviceMap

The DeviceMap subkey contains subkeys for devices such as the keyboard, mouse (or pointer), serial ports, SCSI, and video. As with other parts of the HKEY_LOCAL_MACHINE\Hardware key, DeviceMap is generated at boot time, making modifications ill advised. We can look, however, at several parts of DeviceMap that are typical in a Windows 2000 installation.

One subkey always found in DeviceMap is KeyboardClass. This subkey has a value entry called \Device\KeyboardClass0 (yes, this entry's name does contain backslashes). The data in this entry is \REGISTRY\Machine\System\ControlSet001\Services\Kbdclass. This is a reference to HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Kbdclass, where the current keyboard configuration and settings are.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Another DeviceMap subkey, Scsi, holds information pertaining to SCSI hard drives and IDE (ATAPI) hard drives. Windows 2000 blurs the line between IDE drives and SCSI drives by listing both under the same registry objects.

The keys for a Windows 2000 system with one IDE drive are as follows:

```
Scsi
Scsi\Scsi Port 3
Scsi\Scsi Port 3\Scsi Bus 0\
Scsi\Scsi Port 3\Scsi Bus 0\Target Id 0
Scsi\Scsi Port 3\Scsi Bus 0\Target Id 0\Logical Unit Id 0
  Identifier = "QUANTUM FIREBALL_TM2110S300X"
  Type = "DiskPeripheral"
```

These keys identify a QUANTUM FIREBALL 2GB SCSI hard drive.

Now let's look at another system, a small Windows 2000 server working as a PDC and as a file server (it serves four hard drives and three CD-ROM drives). For a system with two IDE CD-ROM drives, an IDE hard disk drive, a SCSI bus with four SCSI hard drives, and one SCSI CD-ROM drive, the subkeys look like this:

```
Scsi

Scsi\Scsi Port 0
  "FirstBusScanTimeInMs"=dword:000009ef
  "DMAEnabled"=dword:00000001
  "Driver"="atapi"

Scsi\Scsi Port 0\Scsi Bus 0

Scsi\Scsi Port 0\Scsi Bus 0\Initiator Id 255

Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0

  Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0
    "Identifier"="Maxtor 91728D8"
```




```
"Type"="DiskPeripheral"
```

```
Scsi\Scsi Port 1
```

```
"FirstBusScanTimeInMs"=dword:00000014
```

```
"DMAEnabled"=dword:00000000
```

```
"Driver"="atapi"
```

```
Scsi\Scsi Port 1\Scsi Bus 0
```

```
Scsi\Scsi Port 1\Scsi Bus 0\Initiator Id 255
```

```
Scsi\Scsi Port 1\Scsi Bus 0\Target Id 0
```

```
Scsi\Scsi Port 1\Scsi Bus 0\Target Id 0\Logical Unit Id 0
```

```
"Identifier"="MATSHITA CD-ROM CR-581-M"
```

```
"Type"="CdRomPeripheral"
```

```
"DeviceName"="CdRom0"
```

```
Scsi\Scsi Port 1\Scsi Bus 0\Target Id 1
```

```
Scsi\Scsi Port 1\Scsi Bus 0\Target Id 1\Logical Unit Id 0
```

```
"Identifier"="MATSHITA CD-ROM CR-581-M"
```

```
"Type"="CdRomPeripheral"
```

```
"DeviceName"="CdRom1"
```

```
Scsi\Scsi Port 2
```

```
"Interrupt"=dword:0000000a
```

```
"IOAddress"=dword:00006300
```

```
"Driver"="aic78xx"
```

```
Scsi\Scsi Port 2\Scsi Bus 0
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Initiator Id 7
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 0
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 0\Logical Unit Id 0
```

```
"Identifier"="QUANTUM FIREBALL_TM2110S300X"
```

```
"Type"="DiskPeripheral"
```

```
"InquiryData"=hex:00,00,02. . .
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 2
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 2\Logical Unit Id 0
```

```
"Identifier"="MICROP 2112-15MZ1001905HQ30"
```

```
"Type"="DiskPeripheral"
```

```
"InquiryData"=hex:00,00,02. . .
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 4
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 4\Logical Unit Id 0
```

```
"Identifier"="TOSHIBA CD-ROM XM-3301TA2342"
```

```
"Type"="CdRomPeripheral"
```

```
"InquiryData"=hex:05,80,02. . .
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 5
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 5\Logical Unit Id 0
```

```
"Identifier"="QUANTUM XP34301 1071"
```

```
"Type"="DiskPeripheral"
```

```
"InquiryData"=hex:00,00,02. . .
```



```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 6
```

```
Scsi\Scsi Port 2\Scsi Bus 0\Target Id 6\Logical Unit Id 0
  "Identifier"="TOSHIBA MK537FB      6262"
  "Type"="DiskPeripheral"
  "InquiryData"=hex:00,00,02. . .
```

In this example, there is one IDE hard drive and two IDE CD-ROM drives. Both CD-ROM drives are identical MATSHITA CR-581s. There is also one SCSI CD-ROM drive, a Toshiba XM-3301. And there are four SCSI hard drives:

- Identifier = "QUANTUM FIREBALL_TM2110S300X"
- Identifier = "MICROP 2112-15MZ1001905HQ30"
- Identifier = "QUANTUM XP34301 1071"
- Identifier = "TOSHIBA MK537FB 6262"

In addition to the two IDE CD-ROM drives, there is a single 18 GB IDE hard drive:

- Identifier = "Maxtor 91728D8"

If all drives are listed as SCSI, how do we tell the difference between different drive types? Well, actually, that's easy. Take a look at the different keys defined in the subkey Scsi Port *n*, such as in this example:

```
Scsi\Scsi Port 0
  "FirstBusScanTimeInMs"=dword:000009ef
  "DMAEnabled"=dword:00000001
  "Driver"="atapi"
```

In this example, the driver, ATAPI, tells us that the drive is an IDE drive. (ATAPI is short for AT Attachment Peripheral Interface; IDE is short for Integrated Drive Electronics.)

Drives and Buses!

Windows 2000 changes the way IDE devices are handled by the operating system. Under Windows NT version 4, the ATAPI.SYS file handles I/O for all PCI-connected IDE devices. When the device connects to an ISA bus controller, a different driver, ATDISK.SYS, manages the I/O.

ResourceMap

ResourceMap includes items such as HAL, keyboard and pointer port(s), serial and parallel ports, SCSI devices (which include IDE drives), and video information. This subkey also includes data about I/O channels, I/O ports and addresses, IRQs, and DMA channels. Everything in ResourceMap is generated at boot time, so changes are transient at best.

ResourceMap entries are based on class, then device, as Figure 17.4 shows. In this figure, notice that the System Resources subkey has been opened to show two entries: Physical Memory and Reserved. The Physical Memory entry describes the memory that exists in the system, while Reserved describes the parts of memory reserved for special uses. In the system in this example, there are 64MB of RAM, with some sections reserved for special use (check out Figure 17.5). Notice that both RAM and ROM (typically where the computer's BIOS is stored) are listed.

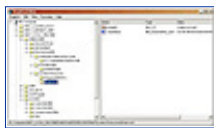


FIGURE 17.4 The ResourceMap key with the System Resources subkey opened

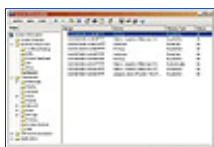


FIGURE 17.5 Reserved memory on this computer is made up of nine blocks of memory, each block having

its own attributes.

Each subkey in ResourceMap consists of one or more subkeys, typically containing two entries: .Raw and .Translated. These entries hold information about the device resources in a special variable type called REG_RESOURCE_LIST. To edit or view device resources, open the object (use RegEdt32 or RegEdit) to display the Resource Lists edit box (shown in Figure 17.6). In the Resource Lists edit box, select a resource and click the Display button to display the Resources dialog box (see Figure 17.7). In this dialog box, you can see a myriad of information, including DMA channel, interrupt (IRQ), memory used (commonly with video cards and some network cards), port used, and device-specific data.

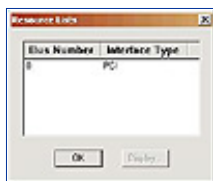


FIGURE 17.6 In the Resource Lists box, you can select a resource and click the Display button. This resource is reserved memory in system resources.



FIGURE 17.7 The Resources dialog box displays information about the reserved memory resource.

SAM

Generally, a Windows 2000 system configured as a domain controller will use Active Directory to manage users. Earlier versions of Windows NT, and Windows 2000 servers that are not part of a domain, use SAM, the Security Accounts Manager, to manage user accounts on the computer. Although Windows 2000 domain controllers use Active Directory, the SAM sections of the registry still exist and are updated by Windows 2000 installations that are not part of a domain.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

Normally, Windows protects SAM against any viewing or tampering by users. This is good; after all, who wants the typical user going in and monkeying about with the user security database? In Windows 2000 domain controllers, the Active Directory tools maintain the user information. In Windows NT 4, changes to SAM are made with either the User Manager or User Manager for Domains tools found in the Administrative Tools section of the Start menu.



WARNING Again, standard warnings: any playing with the SAM section may prevent one, more than one, or all, users from being able to log on. The ultimate result could be that the system may need to be restored from backup or reinstalled. Be most cautious in making any changes in SAM.

SAM consists of a single subkey, called (strangely enough) SAM. Inside the SAM subkey are two subkeys.

The first subkey, called Domains, contains several objects (we'll cover these in a minute, don't panic). The second subkey, called RXACT, contains, as far as I can determine, absolutely nothing useful to the average registry hacker. RXACT is the registry transaction package, used by Windows to manage the registry on servers that are not domain controllers. (However, my experience with Microsoft has indicated that when one thinks that some component of the registry is empty, one is not looking at it correctly.)

OK, back to the Domains subkey. Inside Domains are two subkeys, Account and Builtin.



NOTE Much of the data in the SAM keys is stored in a format that cannot be displayed or edited by the registry editor(s). Therefore, there is little possibility that you can edit these fields.

Domains\Account

In Windows 2000, the Account subkey contains virtually everything regarding the users and groups. Three subkeys, Aliases, Groups, and Users, hold information about aliases, groups, and users.



NOTE For information on user IDs, see Chapter 3. The section entitled "HKEY_USERS, Settings for Users" contains a full reference to SIDs (security identifiers), present throughout the SAM key of the registry. The descriptions below apply to Windows NT 4 and Windows 2000 servers that aren't domain controllers.

Aliases The Aliases subkey contains information on local groups defined in the registry by the system administrator. Local groups defined by the system are maintained in the Builtin subkey.



NOTE Windows 2000 and Active Directory substantially alter the configuration of SAM and Security in the registry. Do not expect to see all the items found in Windows NT 4 domain controllers in Windows 2000 domain controllers. All of the below illustrations are for Windows 2000 servers that are not configured as domain controllers and for Windows NT 4 (and earlier) servers.

Under Aliases, there are subkeys for each local group. (The example in Figure 17.8 has one local account.) Aliases also contains a subkey called Members that lists the user IDs of each member of this group. Each user is identified by a DWORD hexadecimal number (see the next section). In our example, users 000001F4 and 000001F5 are both members of the local Users group.

Groups The Groups subkey has a number of objects, all pertaining to user groups. For instance, on a Windows 2000 non-domain server, or a Windows NT 4 PDC/BDC, these three subkeys are found:

00000200: This subkey, when expressed as decimal (512), relates to the Domain Admins group.

00000201: This subkey, when expressed as decimal (513), relates to the Domain Users group.

00000202: This subkey, when expressed as decimal (514), relates to the Domain Guests group.

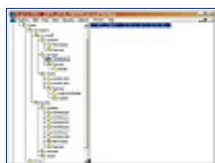


FIGURE 17.8 Local groups in the Aliases subkey

It's not hard to figure out that these first three entries are the default global or domain groups. If you have created any additional global groups, they are found typically with identifiers, such as the following (remember—this is an example):

000003F9: This subkey, when expressed as decimal (1017), relates to the global group called Domain Workers. I created Domain Workers to cover a number of users who needed to access the entire domain to do certain work.

000003FA: This subkey, when expressed as decimal (1018), relates to the global group called Programmers. I created Programmers to cover users working in the R&D division, who needed to access the entire domain to do their work.

Table 17.1 shows common SID values; pay particular attention to the ending digits. Notice that the final two groups, Domain Users and Domain Guests, are not present in the Account\Groups subkey in our example.

TABLE 17.1: COMMON SID VALUES FOR DOMAIN (GLOBAL) GROUPS USED BY WINDOWS 2000

Subkey Name	Global (Domain) Group	SID
00000200	Domain Admins	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-500
00000201	Domain Guests	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-501
00000202	Domain Users	S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx-502



NOTE Note that Windows works typically in unit increments when creating groups and users, so the next group or user created would be 3FA (1018). Notice that the same numbering scheme is used for both groups and users and that these numbers start at 3E8, which is 1000 in decimal. Another interesting tidbit: users and groups created by the system administrator start with 1000 and are incremented sequentially.

Groups also contains the Names subkey. Inside Names is the final piece of this puzzle—subkeys matching those previously described, with the group names as the subkey names. Each of these default subkeys contains a single, unnamed value containing data that neither registry editor is able to display or edit:

- Domain Admins
- Domain Guests
- Domain Users

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

Also within Names are the two groups added to this domain:

- Domain Workers
- Programmers



NOTE Careful—don't go jumpin' to no conclusions here. Do not assume that names listed in the Names subkey have a one-to-one correspondence with the entries above them. In our example, that relationship exists, but that is only because I managed to create groups in alphabetical order, the order in which objects in the Names subkey are stored.

Taking a quick look at Figure 17.9, we see the relationships among the objects contained in the Groups subkey. Five subkeys exist, one for each of the global groups we have (three default, supplied with Windows, and two added later by the system administrator). We also see the Names key, which has five subkeys, each with the name of a global or domain group. The first three are the three default global groups, and the last two are groups created by the system administrator.

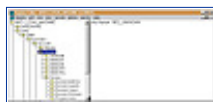


FIGURE 17.9 The relationship between the five numbered subkeys and the subkeys in Names

Users The final subkey within Domains\Account is Users. This subkey contains one entry for each user defined in the SAM. For PDC or BDC computers, the number of users could be large. It is not uncommon for a large network to have hundreds, if not thousands, of users defined. Fortunately, this author has a small network, so there are only a few users defined, making things a bit less cluttered.

Like with groups, system-created users have numbers less than 3F8 (that's 1016) and users created by the system administrator will have numbers greater than or equal to 3F8 (1016).



NOTE There will probably be gaps in the numbers. This is normal—once a user ID has been deleted, Windows will not reuse the number. After all, there are lots of numbers to use, millions and millions of them to be exact.

Rather than list every user ID and name on my system (you can see that I have 17 users defined: 2 defined by the system, and 15 defined by the system administrator), let's just talk about a few of them. First, directly under Users (see Figure 17.10) are some numbered subkeys. The first two subkeys are predefined by the system:

000001F4: The system administrator account has a SID ending in 500 (which is 0x1F4; funny how that

worked out).

000001F5: The system guest account has a SID ending in 501 (which is 0x1F5; just as funny how that worked out).

Looking back at Table 17.1, note the ending digits in the SIDs in the table and the above keys. This is much too scary to be coincidence.

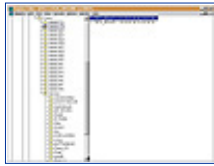


FIGURE 17.10 Users and their attributes are contained within the Users and Users\Names subkeys.

The next subkeys (starting with 000003E8, which is 1000 in decimal, and ending with 000003F8, or 1016) have been created for each administrator-created user ID. Each of these subkeys contains two named keys, called F and V. Feel free to guess as to the meanings of these names and what the data contained within each is. Some assumptions can be made; for instance, there must be an index between the subkeys and their counterparts found in the Names subkey. Information such as group membership, privileges, passwords, and all other data specified by the administrator must be contained in these variables, too.

In the Names subkey (refer again to Figure 17.10), you can see a number of user names found on my system. For example:

Administrator: The system-created system administrator account.

ANONYMOUS: A user ID used by IIS's FTP to allow users to FTP files anonymously.

darkstar-ph: A user ID used in-house to manage e-mail.

DECIEVER: A user ID used by a portable user.

Dialin: Any user may dial in without a password using this user ID.

DORA\$: The computer name for the computer DORA (this server, actually).

The remaining entries (Gay, Guest, and so forth) are similar. There is nothing unusual about them.

Anything ending in a \$ is a machine account, not a user ID. There is always confusion about what constitutes a machine account and what constitutes a user ID. To connect a computer to a server, the server must have both the computer's account and the user's account. Got that? Both must exist! Generally, when you create a new computer configuration, you create the computer name on that computer. The dialog box used to create the computer name has a section called "Create a computer account in the domain," which when checked will create the computer account. Sometimes, however, it is necessary to create a computer account in a domain, and all documentation says to use the User Manager for Domains program without telling you about this neat trick. No one mentions the ending \$, and computer accounts aren't visible in the User Manager for Domains.

Holy mackerel! Batman, what's a user to do?

Computer Accounts for Everyday Administrators

There are, it seems, two ways to create a computer account.

First, in the User Manager for Domains, you can create a user account with a trailing \$. This seems to create a valid computer account, although this is undocumented. Computer accounts created this way remain visible to the User Manager for Domains program.

A second method (a more approved method, I might add!) is to use the command-prompt NET command. Use NET COMPUTER \\<name> /ADD, where <name> is the name of the computer to be added. Use NET COMPUTER \\<name> /DEL, where <name> is the name of the computer to be deleted.

Though Windows 2000/NT computers require a computer account on the server, Windows 95/98 computers do not seem to need one. Other computers, including Windows CE machines, have been known to require a computer account in addition to the user account. Try working without a computer account, and if the user is unable to log on, add a computer account to the user database and try again.

Well, we've covered global users and groups that are created by default by the system. Next, we'll cover

local users and groups.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

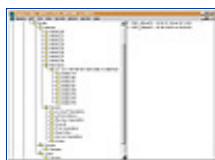
PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Domains\Builtin

Under SAM\Domains\Builtin, there are three subkeys. These subkeys perform a similar task to those in Domains\Account.

Aliases In Aliases, we find the group numbers for each of the local groups. For example, the local group 00000220, when viewed as a decimal number, is 544, which is the local Administrators group. Figure 17.11 shows the Builtin subkey, fully expanded.

Each of these local groups is used to maintain and use the local machine. There are domain groups to perform remote maintenance. Table 17.2 shows the default groups found in Builtin.

**FIGURE 17.11** Expanding Aliases shows eight built-in groups.**TABLE 17.2: THE BUILTIN LOCAL GROUPS**

Subkey Name	Builtin Local Groups	SID
00000220	BUILTIN\ADMINISTRATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-544
00000221	BUILTIN\USERS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-545
00000222	BUILTIN\GUESTS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-546
00000223	BUILTIN\POWER USERS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-547
00000224	BUILTIN\ACCOUNT OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-548
00000225	BUILTIN\SERVER OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-549
00000226	BUILTIN\PRINT OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-550
00000227	BUILTIN\BACKUP OPERATORS	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-551
00000228	BUILTIN\REPLICATOR	S-1-2-32-xxxxxxxx-xxxxxxxx-xxxxxxxx-552

Also present in Aliases is a subkey called Members. In this subkey, there are users and global groups that are members of local groups, identified by their SID suffixes.


Desktop Library
 Click to access!



NOTE Remember: Global groups may be members of local groups, but local groups may not be members of global groups. Oh, and yes, local users may be members of global groups, too. Confused? Good, then I am not alone! Basically, the only member relationship not allowed is local groups as members of global groups; other than that, anything goes.

Groups In Groups, we find a single subkey called Names. There appears to be no information stored in these subkeys.

Users In Users, we find a single subkey called Names. There appears to be no information stored in these subkeys.

RXACT

The key HKEY_LOCAL_MACHINE\SAM\SAM\RXACT is undocumented except that it is listed as belonging to the registry transaction package. The RXACT subkey contains a single, unnamed, REG_NONE type variable. This variable contains a set of three DWORD values, one of which seems to change between installations, the other two of which don't seem to change. The function of these values is unknown, although a guess is that the password-encryption algorithm uses them. Maybe and maybe not.

Security

Security? What is security? First, it is not something to depend on in your old age, that's assured!

In the HKEY_LOCAL_MACHINE\Security key, we find another key that normally only the system has access to. Like with SAM, discussed previously, you can change the access rights to the Security subkey to allow you (the administrator) to browse and (if you are daring, stupid, or both) modify items.

To change the access rights to Security, select it in RegEdt32 and click the Security menu's Permissions selection. In the Registry Key Permissions dialog box, select your user ID. In the Type of Access drop-down list, select Full Control, then click OK. It is imperative to be careful: changing something improperly can lead to disaster.

Cache

Windows 2000 is able to cache from 0 to 50 previous successful logons locally. This is typically done on systems where a domain controller is used to validate logons and security. Sometimes (it happens to all of us) the PDC is not available, and then it is still possible to provisionally log on the user, using locally stored logon credentials.



NOTE The number of cached logons defaults to 10; however, it may be set to any number between 0 and 50.

When the domain controller is unavailable and the user can be logged on using the cache, the following message appears:

A domain controller for your domain could not be contacted. You have been logged on using cached account information. Changes to your profile since you last logged on may not be available.

If caching has been disabled or a user's logon information is not in the cache, this message is displayed:

The system cannot log you on now because the domain <name of domain> is not available.

The Cache subkey holds 11 cache entries (or more, or fewer). One of these entries is NL\$Control, which contains the cached entry of the currently logged-on user.



NOTE RegEdit can display REG_BINARY values better than RegEdt32. With RegEdit, you are able to see both the hexadecimal values and an ANSI character representation, which is readable even with UNICODE characters.

The other entries in the Cache key are named NL\$1 through NL\$10. Each entry contains logon information for one of the previous ten people who logged on to the computer.



NOTE The ten previous logged-on users are unique users. If a user logs on twice, there will be only one entry in the cache—each entry in the cache is for a unique user account.

Policy

Psst, hey buddy, you want to buy some insurance?

No, not that type of policy! The Policy subkey contains security settings for users, groups, and other components.

There are a number of subkeys located under the Policy key. In Windows 2000, these subkeys include the following:

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Accounts	DefQuota
Domains	KerLogoff
KerMaxR	KerMaxT
KerMinT	KerOpts
KerProxy	PolAcDmN
PolAcDmS	PolAdtEv
PolAdtFL	PolAdtLg
PolDnDDN	PolDnDmG
PolDnTrN	PolEfDat
PolMod	PolPrDmN
PolPrDmS	PolRevision
PolSecretEncryptionKey	PolState
QuAbsMax	QuAbsMin
SecDesc	Secrets

Each subkey (excluding Accounts, Domains, and Secrets) is constructed in virtually the same manner: a single, unnamed data variable of type REG_NONE. This data variable will contain a binary value, the length of which depends on the entry's purpose.

The Accounts subkey will contain information on perhaps six or more different SIDs. Most of these SIDs are listed in Table 17.3, along with their descriptions. Note the changes to security and the introduction of the SID S-1-5-11 with Windows NT 4 Service Pack 3.

TABLE 17.3: SOME USERS LISTED IN THE ACCOUNTS SUBKEY

Subkey	Present in Windows 2000	Present in Windows NT 4	Description
S-1-1-0	X	X	(Everyone)
S-1-5-11	X	(Later versions)	Authenticated users; only found in Windows 2000 and Windows NT 4 with Windows NT Service Pack 3 and later
S-1-2-32-544	X	X	BUILTIN\ADMINISTRATORS
S-1-2-32-545		X	BUILTIN\USERS
S-1-2-32-546		X	BUILTIN\GUESTS
S-1-2-32-547		X	BUILTIN\POWER USERS



S-1-2-32-548	X	X	BUILTIN\ACCOUNT OPERATORS
S-1-2-32-549	X	X	BUILTIN\SERVER OPERATORS
S-1-2-32-550	X	X	BUILTIN\PRINT OPERATORS
S-1-2-32-551	X	X	BUILTIN\BACKUP OPERATORS
S-1-2-32-552		X	BUILTIN\REPLICATOR

The information contained in each subkey described in Table 17.3 is:

ActSysAc: A DWORD value, stored as binary (REG_BINARY) data; values range from 0x00000001 to 0x00000003

Privilgs: A variable-length binary value, not found in all Policy\Accounts subkeys

SecDesc: A variable-length binary value

Sid: A binary representation of the SID value for the subkey; the SID value is also used as the subkey name

The Domain subkey will contain information on each domain, typically only one. The Domain key will have a subkey for the domain that the computer belongs to, named with the domain server's SID.

This subkey typically contains four subkeys:

SecDesc: A binary value, probably variable length

Sid: A binary representation of the SID value for the subkey; the SID value is also used as the subkey name

TrDmName: A binary value containing both binary data and the name of the domain

TrDmPxOf: A DWORD value



NOTE A computer that is the domain server (PDC) will not have an entry in the Domain subkey.

The Secrets subkey will contain secret information. (Quiet, someone may be listening to this.) There are a number of subkeys in Secrets. Big users of the Secrets subkey include Windows and IIS.

Due to the nature of the data (no, not that it is secret, just that it is meaningless except to the application or system that is using it), we won't cover it.



NOTE Secret data is specific to the application that has stored the data there, and generally it is not meaningful to users.

RXACT

The RXACT subkey seems to contain an uninitialized value of RXACT that is stored in SAM.

SAM

SECURITY\SAM is an identical copy of the SAM\SAM subkey, which we covered previously.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)[KEYWORD SEARCH](#)[Search Tips](#)[Advanced Search](#)[PUBLICATION LOOKUP](#)[JUMP TO TOPIC](#)**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Previous](#)[Table of Contents](#)[Next](#)**Software**

The HKEY_LOCAL_MACHINE\Software key contains a collection of subkeys for various installed components and applications. Any application can create its own subkey (most do so when they install and store items such as file pointers, user initialization, etc.), although most often subkeys are based on the organization producing the software, with further subkeys for different applications. For example, Microsoft's subkeys might look like this (actually they are much more complex):

```
HKEY_LOCAL_MACHINE\Software\Microsoft
HKEY_LOCAL_MACHINE\Software\Microsoft\DrWatson
HKEY_LOCAL_MACHINE\Software\Microsoft\Exchange
HKEY_LOCAL_MACHINE\Software\Microsoft\IE4
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Audio
```

Each of these subkeys will have one or more entries. (DrWatson has twelve entries with data values for different user settings and filenames, set using DRWTSN32.EXE.)

Figure 17.12 shows the HKEY_LOCAL_MACHINE\Software key on a computer that has been running Windows 2000 for about nine months, and has a number of software packages installed.

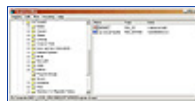


FIGURE 17.12 The HKEY_LOCAL_MACHINE\Software key can become large if there are many installed applications and system components.

Now, you may think that I should say more about the HKEY_LOCAL_MACHINE\Software key here in this chapter; but no, I won't. There is just too much really good stuff in HKEY_LOCAL_MACHINE\Software to not devote a complete chapter to it, so I've dedicated the next chapter (Chapter 18) to this key.

System

HKEY_LOCAL_MACHINE\System contains the system configurations. Subkeys include:

ControlSet001: A copy of the current control set.

ControlSet002: The Last Known Good control set used to boot from if there is a problem booting from the current control set.

CurrentControlSet: The control set used to boot from.

DISK: Contains information about drive letters, volume sets, RAID (mirrored, stripe, and stripe with parity), and CD-ROM and drive mapping.

MountedDevices: Contains information about currently available drives, including CD-ROM, hard, and floppy disks.

Select: Contains information about which control set is used for what purpose.

Setup: Contains information about Windows 2000's installation.

As with HKEY_LOCAL_MACHINE\Software, HKEY_LOCAL_MACHINE\System is a very important key, and therefore I'll cover it fully in Chapter 19. No sense in overdoing it here.

Hints and Kinks from the Experts

In this chapter's hints and kinks, we cover a few hints on using the two registry editors.

How Do You Find Out Who Changed the @!#* Administrator's Password?

To determine the username that changed the Administrator password, perform the following on the PDC:

1. Enable Success and Failure audits for File and Object Access using User Manager for Domains Policies Audit.
2. Using RegEdt32, select the SAM key in HKEY_LOCAL_MACHINE and use Security Permissions to set Full Control for the Administrators local group. Check Change Permissions on Existing Subkeys.
3. Navigate to HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F4. Select Security Audit Permissions and add the Administrators local group to the list. Select this group and enable Success and Failure auditing for Set Value events on this and all subkeys.

When a change is made to the Administrator account, the following event will indicate the username:

```
ID: 560
Source: Security
Type: Success Audit
Category: Object Access
```

(Courtesy of Jerold Schulman.)

How Do You Identify Registry Keys in the Registry?

Permanent keys, those not created at boot, are identified in the key at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\hivelist. The one exception is HKEY_CURRENT_USER, which is located at %systemroot%\Profiles\UserName.

The value entries identify the registry keys. All are type REG_SZ. Table 17.4 shows the permanent HKEY_LOCAL_MACHINE keys.

TABLE 17.4: PERMANENT HKEY_LOCAL_MACHINE KEYS

Permanent Key	Typical Default Value	Comment
\REGISTRY\MACHINE\HARDWARE (None)		The HKEY_LOCAL_MACHINE\Hardware key, recreated upon boot
\REGISTRY\MACHINE\SAM	\Device\Harddisk 0\Partition1\WINNT\System32\Config\SAM	HKEY_LOCAL_MACHINE\SAM
\REGISTRY\MACHINE\SECURITY	\Device\Harddisk 0\Partition1\WINNT\System32\Config\Security	HKEY_LOCAL_MACHINE\Security
\REGISTRY\MACHINE\SOFTWARE	\Device\Harddisk 0\Partition1\WINNT\System32\Config\Software	HKEY_LOCAL_MACHINE\Software
\REGISTRY\MACHINE\SYSTEM	\Device\Harddisk 0\Partition1\WINNT\System32\Config\System	HKEY_LOCAL_MACHINE\System
\REGISTRY\USER\DEFAULT	\Device\Harddisk 0\Partition1\WINNT\System32\Config\Default	HKEY_USERS\DEFAULT
\REGISTRY\USER\Security ID (SID)	\Device\Harddisk 0\Partition1\WINNT\Profiles\Username\ntuser.dat	The current user profile; if services are running under user accounts, their entries are also located here
\REGISTRY\USER\Security ID (SID)_Classes	\Device\HarddiskVolume1\WINNT\Profiles\Username\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat	The current user's classes definition (see Chapter 15, on HKEY_CLASSES_ROOT, for more information)

(Courtesy of Peter D. Hipson.)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 18**Introduction to HKEY_LOCAL_MACHINE\Software****FEATURING:**

<i>Classes</i>	468
<i>Clients</i>	468
<i>Microsoft</i>	470
<i>ODBC</i>	518
<i>Policies</i>	519
<i>Program Groups</i>	519
<i>Secure</i>	519
<i>Voice</i>	519
<i>Hints and kinks from the experts</i>	520

The HKEY_LOCAL_MACHINE\Software key contains information about installed software on your system. It also contains information about Windows 2000, although the HKEY_LOCAL_MACHINE\System key contains Windows 2000 information as well.

In a typical installation, HKEY_LOCAL_MACHINE\Software has at least eight subkeys:

- Classes
- Clients
- Microsoft
- ODBC





- Policies
- Program Groups
- Secure
- Voice
- Windows 3.1 Migration Status

For many installations, there will be at least one more additional subkey under HKEY_LOCAL_MACHINE\Software. For example:

- 3Com, or the supplier of the computer's NIC
- Adobe, for users of Adobe's PageMaker software
- Symantec, for anyone who uses Symantec antivirus software
- INTEL, for software such as the Intel 3D Scalability Toolkit
- Intuit, if you use their accounting or tax software
- Qualcomm, if you use one of their e-mail or communications products

Of course, only the number and types of software packages installed on the target computer limit the number of subkeys. Go hog wild, install tons of stuff, and you'll have a big HKEY_LOCAL_MACHINE\Software subkey.



WARNING You *have* backed up your registry before starting this chapter, right?

In Case of Disaster

Be cautious! Blow a subkey or value, and you probably will have to reinstall the affected product. If you're really unlucky, you might also have to reinstall Windows 2000, if the application's install program doesn't properly repair the registry. If you find you have to reinstall because you didn't have a good registry backup, follow these steps:

1. Reinstall the product without uninstalling the original installation. If this works, you may be able to recover your user settings, profiles, and such. If this doesn't work, try step 2.
2. Uninstall the product and then reinstall. This probably will cause you to lose the user settings, profiles, and such, but that's life. If this doesn't work, try step 3.
3. Install a second copy of Windows 2000, and install the application on the second copy of Windows 2000 into the product's original directory. If the product works on the second copy of Windows 2000, try the first copy again. If the product still doesn't work on the first copy but does work on the second copy, you'll have to restore everything from backups or reinstall everything from scratch. Either way, you are in for a long, long night.

As I mentioned above, HKEY_LOCAL_MACHINE\Software has at least eight sections. Each of the sections varies in size from small (Secure usually has nothing in it at all!) to huge (Microsoft has settings for every Microsoft application installed, and for some components of Windows 2000 also.)

In this chapter, I have tried to cover as many keys, values, and objects as possible. To cover *everything* would make this chapter totally unmanageable.



NOTE In Windows NT 4, a typical user's HKEY_LOCAL_MACHINE\Software key may contain only the Microsoft, Policies, and VDO subkeys.

Classes

The HKEY_LOCAL_MACHINE\Software\Classes subkey is a mapping of the HKEY_CLASSES_ROOT registry hive. This ensures that both have the same entries.

The HKEY_CLASSES_ROOT hive is described in Chapter 3 and in Chapter 15 (which is devoted entirely to this hive). The information in these chapters applies to all entries in HKEY_LOCAL_MACHINE\Software\Classes.

Clients

The HKEY_LOCAL_MACHINE\Software\Clients subkey contains information used by Windows 2000 for e-mail and related services. Users who have not installed any additional e-mail services will find the Microsoft Exchange client defined as their mail provider—Exchange is a default component of Windows 2000.

A Microsoft Internet Explorer installation includes features such as:

Contacts: An integral component of Outlook Express and Outlook, Contacts provides a powerful tool for managing contacts and names. Contacts manages names, addresses (including e-mail addresses), phone numbers, etc.

Mail: Internet Explorer 5 installs a product called Outlook Express, which is a scaled-down version of Microsoft's e-mail client, Outlook.

News: The News client is a component in Outlook Express. Newsgroups are public (and private) forums on the Internet where users are able to speak their minds on various topics.

EnvelopeHost: Installed by Office 2000, EnvelopeHost is a utility to create envelopes from each of the Office 2000 products.

A full installation of Microsoft's Outlook would include even more components, including:

Calendar: Outlook's calendar serves as an integrated time management tool and a powerful scheduling tool.

Internet Call: NetMeeting is a tool used to hold online, interactive meetings.

Typical subkeys found in HKEY_LOCAL_MACHINE\Software\Clients might include those shown in Figure 18.1, which shows a computer that has the full version of Outlook, Outlook Express, Forte Agent, and Eudora installed.

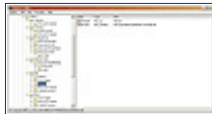


FIGURE 18.1 Clients can include products from more than one software vendor—check out my five e-mail clients, all usable at the same time!

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)**Why More Than One E-Mail Client?**

OK, I can hear the questions now: Why the heck does this guy need so many e-mail clients? Yes, there is a good reason (at least I think so!):

1. I use Microsoft Outlook as my primary e-mail client. However—
2. I use Eudora to get messages from a mailing list that I belong to, because most of the people on this list are not using Windows-compatible systems, and thus they don't receive Outlook-generated messages well. (Outlook creates two copies of the message, one in plain text and one in formatted text.) Also, as the mail list is archived, the Outlook-generated messages are too large for the archive.
3. I use Forte Agent to retrieve messages from news servers. I could use Microsoft's newsreader program, but I just like Agent. Because Agent can also send e-mail, it is an e-mail program, too.
4. Hotmail was installed with Windows 2000 by default.
5. Outlook Express installs with Internet Explorer. I have not used this program because I use Outlook.

All of these e-mail programs are an example of:

- How registry entries multiply (like rabbits!)
- How the entries get convoluted (like Exchange being deleted by Outlook, but entries for Exchange remaining in the registry, probably until hell freezes over)
- How products get installed without the user realizing that they have been installed (Outlook Express)

Microsoft

The Microsoft subkey in HKEY_LOCAL_MACHINE\Software is the largest of the subkeys in the Software key. One typical installation of a Windows 2000 Server, with a number of applications from Microsoft installed, has the following subkeys under HKEY_LOCAL_MACHINE\Software\Microsoft:

Access Runtime: Contains information about Microsoft Access 2000.

ACS: Found only in Windows 2000, this object contains information about SNMP's ACS functionality.

Active Accessibility: Contains support for Microsoft Active Accessibility (MSAA). MSAA improves the access of command bars.

Active Setup: Currently used with Internet Explorer as well as many other Windows 2000 components,

Active Setup is the installation process for applications distributed by the Internet.

ADs: Found only in Windows 2000, this object contains information about the Active Directory server functions.

Advanced INF Setup: Contains support for setups based on .inf files, used with Internet Explorer and some Internet Explorer components, such as Java.

AsyncMac: Also known as AsyMAC, this object contains an NDIS 3.0 MAC (Media Access Control) driver used with serial communications (RAS).

AudioCompressionManager: Contains support for the Microsoft Audio Compression Manager system.

BOOTPMibAgent: Found only in Windows 2000, this object contains information about support for the BootP MIB (Management Information Base) Agent.

Browser: Contains the configuration for the Computer Browser Service.

ClipArt Gallery: Contains information about clip art objects installed. Typically, but not always, installed with products such as Microsoft Office.

Code Store Database: Used with objects such as Java.

COM3: Found only in Windows 2000, this object contains information about COM (Common Object Model) version 3.

Command Processor: Found only in Windows 2000, this object contains initialization parameters for all command-prompt sessions.

Conferencing: Supports Microsoft NetMeeting, a virtual conferencing/meeting system.

Connection Manager Administration Kit: Found only in Windows 2000, this object contains information about the Connection Manager Administration Kit (CMAK), used to manage RAS connections.

CRS: Contains the configuration for the Content Replication System.

Cryptography: Found only in Windows 2000, this object contains information about each of the installed cryptography services. It contains the management for the Microsoft CryptoAPI (Cryptographic Application Program Interface).

DataAccess: Found only in Windows 2000, this object contains information about data access providers for HTTP, HTTPS, IIS, LDAP and others.

DataFactory: Found only in Windows 2000, this object contains information about OLE data factory handlers.

DeviceManager: Found only in Windows 2000, this object supports information about bus types (not fully implemented in Windows 2000) and troubleshooters.

Dfrg: Found only in Windows 2000, this object contains information about the disk defragmenter, a standard Windows 2000 feature.

DhcpMibAgent: Found only in Windows 2000, this object supports DHCP MIB.

DHCP Server: Contains the support and configuration for DHCP (Dynamic Host Configuration Protocol), which, in a nutshell, allows dynamic (automatic) allocation of IP addresses in a TCP/IP network.

Direct3D: Contains Microsoft's high-performance 3-D drawing API.

DirectDraw: Found only in Windows 2000, this object contains information about the high-speed Direct3D support.

DirectInput: Found only in Windows 2000, this object contains information about high-speed input interfaces.

DirectMusic: Found only in Windows 2000, this object contains information about the high-performance music player.

DirectPlay: Contains Microsoft's high-performance engine that provides a way for multiplayer games to communicate using networks.

DirectX: Contains Microsoft's I/O system.

DownloadManager: Contains a system that allows files being transferred by Internet Explorer to be downloaded in background mode, to be suspended and resumed as desired.

Driver Signing: Found only in Windows 2000, this object contains information about drivers and driver authentication.

DrWatson: Contains a system for providing information on application and system faults.

EnterpriseCertificates: Found only in Windows 2000, this object contains information about various enterprise certificate authorities.

ESENT: Found only in Windows 2000, this object contains information for the ESENT system.

EventSystem: Found only in Windows 2000, this object contains information for the event management system.

Exchange: Contains Microsoft's default e-mail client.

Fax: Found only in Windows 2000, this object contains information about the built-in Windows 2000 fax system.

FrontPage: Contains Microsoft's application to develop and manage Internet Web pages.

Home Publishing: Found only in Windows 2000, this object contains information about Microsoft Home Publishing.

HostMIB: Found only in Windows 2000, this object contains information for the Host MIB.

HTML Help Collections: Found only in Windows 2000, this object contains information about HTML help.

IASAgent: Found only in Windows 2000, this object contains information about the Internet Authentication Services agent.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

IE Setup: Found only in Windows 2000, this object contains information about setup for Internet Explorer.

IE4: Contains Microsoft's Internet Explorer 4.

IGMPMibAgent: Found only in Windows 2000, this object contains information about the Internet Group Management Protocol.

INetMgr: The Internet Service Manager, used to manage Microsoft IIS (Internet Information Server).

Inetsrv: Contains Microsoft Internet Information Server.

INetStp: Contains configuration information for Microsoft IIS.

InfoViewer: Contains Microsoft Information Viewer, a data and information retrieval system, typically used with Microsoft TechNet and MSDN.

IntelliPoint: Contains support for the Microsoft IntelliMouse, an enhanced pointing device.

InteractiveMusic: No, not karaoke. Microsoft Interactive Music is a system used to deliver music over the Internet.

Internet Account Manager: Used to manage e-mail accounts.

Internet Audio: Audio may be sent to clients on the Internet using a number of different compression techniques, such as CCITT, Lernout & Hauspie, and Microsoft's encoding.

Internet Connection Wizard: The Internet Connection Wizard automates the steps used to connect a new user to the Internet.

Internet Domains: Found only in Windows 2000, this object contains information about special domains. A typical default entry in this object consists of a number of data items for the Hotmail e-mail system.

Internet Explorer: Contains Microsoft Internet Explorer, currently version 5.

Internet Mail and News: Contains Internet Mail and News settings.

Internet Shopper: Contains Microsoft's client Internet commerce system.

IpInIp: Found only in Windows 2000, this object contains information about IpInIp.

IPMulticastMibAgent: Found only in Windows 2000, this object contains information about the IP Multicast MIB agent.

IPXMibAgent: Found only in Windows 2000, this object contains information about the IPX MIB agent.

Java VM: Contains the Java virtual machine configuration.

Jet: Contains the Microsoft Access database access engine, used by Microsoft Office and other



Desktop Library

Click to access!

applications.

KeyRing: Contains a small, usually metal, often lost, object used to hold keys. Or contains IIS's Key Manager program.

Languages: Used with Microsoft's Internet browser and server support, used to define file types (such as HTM, HTML, STM, STML, and ALX).

LANManagerMIB2Agent: Found only in Windows 2000, this object contains information about the LAN Manager MIB agent.

LanmanServer: Manages server support for the SMB (Server Message Block) protocol, the core of Microsoft networking.

LanmanWorkstation: Manages client support for the SMB protocol.

Machine Debug Manager: Works to help define which processes should be debugged and which should be ignored.

MediaPlayer: Found only in Windows 2000, this object contains information about Microsoft Media Player.

Microsoft Chat Character Editor: Contains an add-on graphics editor for Microsoft Chat characters.

Microsoft Comic Chat: Contains Internet Chat, an interactive conferencing system with a graphic interface (using comic characters).

Microsoft Expedia: Contains Pocket Streets 98, Microsoft's road atlas program.

Microsoft FrontPage: Contains Microsoft's Internet Web page publishing utility.

Microsoft Image Composer: Contains Microsoft's graphic- and image-editing application.

Microsoft Reference: Contains a component of Microsoft office that provides a complete reference section.

MMC: Found only in Windows 2000, this object contains information about the Microsoft Management Console.

MMCtIsForIE: Contains multimedia controls for Internet Explorer.

Mobile: Found only in Windows 2000, this object contains information about mobile or portable operations.

MOS: Contains configurations for Microsoft Office's Outlook and MSNAudio.

MosTrace: Found only in Windows 2000, this object contains support for gathering debug information.

MS Design Tools: Found only in Windows 2000, this object contains information for the Microsoft Design Tools feature.

MSDTC: Found only in Windows 2000, this object contains information about Microsoft DTC.

MSE: Found only in Windows 2000, this object contains information about the Microsoft Script Editor.

MSFTPSVC: Found only in Windows 2000, this object contains information about the Microsoft FTP service.

MSMQ: Found only in Windows 2000, this object contains information about Microsoft Message Queue Server.

MSTTS: Found only in Windows 2000, this object contains information about the Microsoft speech engine (TTS is short for text-to-speech).

Multimedia: Contains the configurations for Active Movie and DirectX components.

Nepa: Contains the configuration information for the Network Control Panel applet.

NdisWan: Contains the Network Device Interface Standard, used with a WAN (wide area network), such as the Internet.

NetBIOS: Contains the Network Basic Input/Output System, used to control basic network communications, including the software interface and naming conventions.

NetBT: Contains the configuration for NetBIOS when implemented over TCP/IP.

NetDDE: Contains the configuration for Network Dynamic Data Exchange.

NetSh: Found only in Windows 2000, this object contains information about performance managing for the networking scheduler.

NetShow: Found only in Windows 2000, this object contains information about the Microsoft Net Show feature.

Non-Driver Signing: Found only in Windows 2000, this object contains information about signing of non-driver files.

NTDebuggers: Found only in Windows 2000, this object contains information about the Windows NT debuggers.

NTDS: Found only in Windows 2000, this object contains information for the Windows NT directory service.

NwlnkIpx: Contains support for the Novell NetWare IPX protocol.

NwlnkNb: Contains the configuration for the NetWare network browser.

NwlnkSpx: Contains the configuration for the NetWare SPX protocol.

Office: Contains Microsoft Office, whichever version is installed, typically Office 97 or Office 2000.

Ole: Contains basic configuration information for Object Linking and Embedding.

OS/2 Subsystem for NT: Contains basic support for OS/2 standards.

OSPFMibAgent: Found only in Windows 2000, this object contains information about the OSPF MIB agent.

Outlook Express: Contains Microsoft's basic e-mail system, installed by default with Internet Explorer.

Pegasus: Contains winged horse of Greek mythology, or support for Windows CE 2.x.

Protected Storage System Provider: Contains an inaccessible subkey used to protect user security.

RasAuto: Contains configurations for the Remote Access Service AutoDial facility, used to automatically connect to a remote network.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

RAS: Contains configurations for the Remote Access Service, the dial-in component of Windows 2000 Server.

RasMan: Contains the Remote Access Service manager program.

ReferenceTitles: Contains Microsoft Bookshelf (part of Microsoft Office).

RemoteAccess: Contains some settings for RAS (Remote Access Service).

ResKit: The basic setup settings for the Windows 2000 Resource Kit (either for server or workstation).

Resource Kit: Contains the component settings for the Windows 2000 Resource Kit.

RFC1156Agent: Found only in Windows 2000, this object contains information about the MIB for use with network management protocols in TCP/IP-based networks.

RIPMibAgent: Found only in Windows 2000, this object contains information about the Routing Information Protocol MIB agent.

Router: Found only in Windows 2000, this object contains information about IP routing.

Rpc: Contains the configuration for Remote Procedure Calls.

RPCLOCATOR: Used to enable RPC applications to perform procedures on multiple remote computers.

Schedule+: Contains settings for Schedule+ or a substitute, such as Outlook or Exchange.

SchedulingAgent: Found only in Windows 2000, this object contains information about Windows 2000 scheduling.

ScriptDbg: Contains Microsoft Office's script debugger settings.

Secure: Found only in Windows 2000, this object contains information about security.

Shared Tools Location: Found only in Windows 2000, this object contains information about the location of tools that are shared between more than one application.

Shared Tools: Lists and describes relationships with various Microsoft tools that may be "shared" using OLE.

SNMP_EVENTS: Found only in Windows 2000, this object contains information about SNMP (Simple Network Management Protocol) events.

SNMPMIB: Found only in Windows 2000, this object contains information about the SNMP MIB.

SpeechAPI: Found only in Windows 2000, this object contains information about the built-in speech API.

SystemCertificates: Contains information about security certificates (used primarily with Internet Explorer).



Tcpip: Stores system configurations for TCP/IP. Computer-specific configuration information is stored elsewhere.

TelnetServer: Found only in Windows 2000, this object contains information about the Windows 2000 Telnet server.

Tracing: Found only in Windows 2000, this object contains information about event tracing.

Transaction Server: Found only in Windows 2000, this object contains information about Transaction Services.

TShoot: Found only in Windows 2000, this object contains information about troubleshooting.

User information: This object contains information for the Office 2000 user.

VBA: This is where Microsoft Visual Basic for Applications, used with a number of Microsoft products, is configured.

Visual Basic: This object contains information about Visual Basic.

VisualScriptEditor: This object contains information about the Visual Basic script editor.

VisualStudio: Contains the configuration for Microsoft's Visual Studio, the development platform.

W3SVC: Found only in Windows 2000, this object contains information about IIS's Web server.

WAB: Not "Windows Always Breaks"; this is where the WAB (Windows Address Book), used to manage addresses (different from Outlook's Contacts functionality), is configured.

WBEM: Found only in Windows 2000, this object contains information about Web-Based Enterprise Management.

Windows: This is where a number of Windows configuration parameters are set. See the second part of this section for more information on this subkey.

Windows Messaging Subsystem: Contains configurations for e-mail.

Windows NT: This is where a number of Windows configuration parameters are set. The second part of this section documents this subkey.

Windows Scripting Host: Found only in Windows 2000, this object contains information about scripting (JavaScript, or Visual Basic Script).

Wins: The Windows Internet Name Service, or WINS, saves configuration information in this subkey.

Looking at the previous list, you should realize that there are many more possible subkeys under Software—so many possibilities, in fact, that no single source could hope to document them all. Each installed application or component can and often does create a subkey in the HKEY_LOCAL_MACHINE\Software key.

Even more interesting and unexplainable is the fact that there are both Windows *and* Windows NT sections in HKEY_LOCAL_MACHINE\Software. Microsoft came up with Windows, then later developed Windows NT, and chose to group items whichever way they wanted. Were I to tell you that old stuff was in Windows and new stuff was in Windows NT, I'd be accused of making it all up (and rightly, I might add). There is little rhyme or reason to the organization and contents of these two subkeys.

Now, let's take a detailed peek into a few of the subkeys found on virtually every Windows 2000 system.



NOTE Some items below are specific to server installations, others are specific to workstation installations, and most are applicable to both servers and workstations. If known, I've indicated which ones are specific to which type of installation.

Windows

First, I know the questions you are asking right now: Why is configuration for parts of Windows 2000 included in HKEY_LOCAL_MACHINE\Software, and parts in other sections of the registry? Why isn't it all consolidated? Why spread it out? Why, oh why, is this so hard to understand?

Well, I can't answer the last question, but I may be able to shed a bit of light on a few of the others. Many of the components included in HKEY_LOCAL_MACHINE\Software are components that are or were separate from Windows 2000. For example, Internet Explorer, Font support, and even Explorer are separate from the base operating system. Yes, dear friends, you can run Windows 2000 without using Explorer—Program Manager is still part of Windows 2000. (No, I'm not going to comment on Windows 2000's support for this antiquated user-interface component!)

CurrentVersion

Under HKEY_LOCAL_MACHINE\Software\Microsoft\Windows is a subkey called CurrentVersion. In this subkey, there are a number of keys defining information about the current installation of Windows 2000. It also contains a large number of subkeys (which I've documented below, in separate sections) for various components of Windows 2000.



NOTE Many of the settings that were found in CurrentVersion in Windows NT 4 are now found in the subkey HKEY_CURRENT_USER\<sid>\Software\Microsoft\Windows\CurrentVersion. This move allows customization on a user-by-user basis.

CurrentVersion\AdminDebug

Found only in Windows 2000, the AdminDebug subkey contains information that Windows 2000 uses to manage debugging.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CurrentVersion\App Management

Found only in Windows 2000, the App Management subkey contains information that Windows 2000 uses to manage various applications. You'll find subkeys for many of the installed programs on the computer here.

CurrentVersion\App Paths

In a typical installation of Windows 2000, you might find as many as 30 or 40 objects in the App Paths subkey. A clean installation of Windows 2000 might have fewer than 10 objects; for example:

- DIALER.EXE
- HYPERTRM.EXE
- PINBALL.EXE
- WORDPAD.EXE

Even this "minimum" list could be smaller if Hyperterm or Pinball were not installed on the computer.

Each subkey in App Paths contains one required entry:

<No Name>: An unnamed value with a data value containing a string with the fully qualified path of the application, including the application's name and extension, typically .exe.

Each subkey may contain one additional entry:

Path: An entry with a data value containing a string with the fully qualified path of the application, typically used to locate supporting files, if necessary. Not all App Paths subkeys have the Path value.

If you must move an application component, check App Paths to see if the application is listed. If it is, when moving the component, make sure that the App Paths entries are updated to reflect the application's new location.

CurrentVersion\Applets

Found only in Windows 2000, the Applets subkey contains information that Windows 2000 uses to manage installed applets.

CurrentVersion\Control Panel

Found only in Windows 2000, the Control Panel subkey contains information that Windows 2000 uses to



manage the system Control Panel.

CurrentVersion\Controls Folder

In the Controls Folder subkey, there is a single binary value. Within this value, there is information used by the Control Panel to configure the display of Control Panel applets, including title information.

A number of Control Panel applets may also include subkeys in Controls Folder. For instance, when applets use special handlers, typically done with OLE, a mapping of tabs in the Control Panel applet to OLE server is found in Controls Folder.

Here is an example of the subkey for the Display Control Panel applet:

Display: Might contain an entry to manage optional components in the Display applet's main window.

In addition to the above example, other Control Panel applets, such as Desk and Device, can and do use the Controls Folder subkey.

CurrentVersion\CSCSettings

Found only in Windows 2000, the CSCSettings subkey contains information that Windows 2000 uses to manage CSC (Client Side Caching) settings.

CurrentVersion\Dynamic Directory

Found only in Windows 2000, the Dynamic Directory subkey contains information that Windows 2000 uses to manage the Dynamic Directory Service.

CurrentVersion\Explorer

Microsoft Explorer, which functions as the user interface for Windows 2000, has a number of configuration options. Some options are set with various configuration dialog boxes; others must be set using one of the registry editors.

AlwaysUnloadDLL: Contains a single, unnamed string value, with a value of either 1 or 0.

AutoComplete: Contains a single string value, named UseAutoComplete. The default value is Yes.

BrowseNewProcess: Contains a single string value, named BrowseNewProcess. The default value is Yes.

CSSFilters: Contains a number of entries primarily for Internet Explorer. These entries are for OLE controls used for visual effects, such as Blur, Invert, Glow, and Shadow.

Desktop: Typically contains three entries, for Inbox, Recycle Bin, and The Internet. These are default items on the Desktop.

FileTypesPropertySheetHook: Used by Internet Explorer, the entries in this subkey are used to display files, often containing MIME-encoded objects.

FindExtensions: Used by Internet Explorer, Outlook, and the Windows Address Book to manage their find functionality.

MyComputer: Used with the Start menu (and elsewhere). Other entries found on some computers include dial-up networking and mobile devices.

Namespace\Controls: Contains the Control Panel.

Namespace\Printers: Contains the Printers Panel.

NewShortcutHandlers: Used to manage items, such as property sheets.

RemoteComputer: Contains a subkey called Namespace, which includes information on remote printers.

Shell Folders: Used by the system to configure part of a user's profile. User profiles consist of two parts: the user's private items and a second common profile called All Users. The Shell Folders subkey contains four keys:

Common Desktop: Contains a path name to the profiles directory. On many systems this will be C:\Documents and Settings\All Users\Desktop.

Common Programs: Contains a path name to the common programs directory. On many systems, this will be C:\Documents and Settings\All Users\Start Menu\Programs.

Common Start Menu: Contains a path name to the Start menu. On many systems, this will be

C:\Documents and Settings\All Users\Start Menu.

Common Startup: Contains a path name to the Start menu\Programs directory. On many systems, this will be C:\Documents and Settings\All Users\Start Menu\Programs\Startup.

ShellExecuteHooks: Used by Internet Explorer to manage the execution of shell extensions.

SmallIcons: On the Plus tab of the Display Properties dialog box, the Use Large Icons checkbox state. String values allowed are YES and NO.

Streams: Could be small rivers, seasonally may be dry. More likely, contains the Taskbar and toolbar and only one subkey:

Desktop: Two entries, Default Taskbar and Default Toolbars, are found in this subkey.

Thumbnail View: Contains one entry, called AutoExtract. The value will be either 0x1 or 0x0.

Tips: Contains money or value given to a person who serves you, or words of advice. OK, really. Windows displays tips on a dialog box when a user logs on, although most users turn off the tips as their second or third action after installing Windows. Fifty tips exist in Windows 2000 by default, but you could add more. And a tip for you: If you add more, Windows won't know about them; Windows expects 50, and uses 50.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

User Shell Folders: Contains the folders used for users. Four keys exist in this subkey:

Common Desktop: Contains the path %systemroot%\Profiles\All Users\Desktop. This provides a path to the common desktop for users.

Common Programs: Contains the path %systemroot%\Profiles\All Users\Start Menu\Programs. This provides a path to the common Start Menu\Programs directory for users.

Common Start Menu: Contains the path %systemroot%\Profiles\All Users\Start Menu. This provides a path to the common Start menu directory for users.

Common Startup: Contains the path %systemroot%\Profiles\All Users\Start Menu\Programs\Startup. This provides a path to the common Start Menu\Programs\Startup directory for users.

User Shell Folders\New: Usually empty; contains a location for new common objects for users.

VolumeCaches: Empty, except for the following subkeys:

Active Setup Temp Folders: The description reads: "These files should no longer be needed. They were originally created by a setup program that is no longer running."

Downloaded Program Files: The description reads: "Downloaded Program Files are ActiveX controls and Java applets downloaded automatically from the Internet when you view certain pages. They are temporarily stored in the Downloaded Program Files folder on your hard disk."

Internet Cache Files: The description reads: "The Temporary Internet Files folder contains Web pages stored on your hard disk for quick viewing. Your personalized settings for Web pages will be left intact."

WindowsUpdate: Contains a single entry called UpdateURL, which contains a reference to the Windows Internet Connection Wizard.

CurrentVersion\Extensions

The Extensions subkey contains keys that define what program opens a specific file type. Similar to the Classes subkeys found elsewhere in the registry, Extensions is only for added-on, non-Microsoft applications. The Extensions subkey shows, in File Manager, the application that a user prefers to open a certain file with—for example, "Open .rtf files with Word for Windows." Explorer, Windows NT 4, and Windows 2000 do not appear to use this subkey.



NOTE Why have this in the registry if it is not used? Simple: Many legacy (older) applications will attempt to update the subkey even though it is not used. Also, since it is possible to use Program Manager with Windows NT 4, there actually is a potential use for these entries.

CurrentVersion\ExtShellViews

Found only in Windows 2000, the ExtShellViews subkey contains information that Windows 2000 uses to manage how a view is presented to the user: as either a Web view or a thumbnail view.

CurrentVersion\Group Policy

Found only in Windows 2000, the Group Policy subkey contains information that Windows 2000 uses to manage groups. Some of the Windows 2000 Active Directory information about groups is stored in this location.

CurrentVersion\H323TSP

Found only in Windows 2000, the H323TSP subkey contains information that Windows 2000 uses to manage H323 teleconferencing.

CurrentVersion\Installer

Found only in Windows 2000, the Installer subkey contains information that Windows 2000 uses to manage the Office 2000 installer program.

CurrentVersion\Internet Settings

The Internet Settings subkey consists of settings used with the Internet, primarily with Internet Explorer. Two keys present in this key include ActiveXCache, which points to a directory where Internet Explorer may cache ActiveX (OLE) controls; and CodeBaseSearchPath, which points to a Microsoft site where common code is downloadable.

In addition to these two keys, there are a number of subkeys contained in Internet Settings:

Accepted Documents: Some documents are accepted as safe. These include Word, Excel, and PowerPoint documents. (No, there is no need to tell me about all those nasty Word viruses. I know.) Also considered safe are the GIF, bitmap, and JPEG image types.

ActiveX Cache: A second set of ActiveX control cache directories. Two entries in this subkey provide two locations to store ActiveX controls installed on the user's system.

Cache: Contains Internet Explorer's cache parameters: cleanup factor, interval and time, a debug flag, freshness, and persistence. These factors are set in Internet Explorer's Settings dialog box, on the General tab.

Cache\Paths: Internet Explorer stores Web pages and objects in a series of cache directories. The default is to have four cache directories, though the number of cache directories can be modified if necessary. For each cache directory, a subkey named *path_n* (where *n* is the number of cache directories) is created. Each of the path subkeys contains a path name and a size limit.

Cache\Special Paths: Two special directories are used by Internet Explorer. These are for cookies (small files stored on the computer by a Web site), and for the Internet Explorer history list.

Cookies: Contains keys to limit the size of the cookies directory, the cache prefix (cookie:), and the directory path. Cookies track a user's usage of a particular site, monitor favorite selections, establish user-based defaults, and sometimes hold information about the user. Virtually all cookie use is benign, intended to optimize the Web site's presentation to the user or to cache user-specific information for reuse at a later time.



NOTE Don't like cookies? Don't accept them, and clean out your cookies directory. Nothing evil will happen—a few Internet sites will deny access to clients who don't allow cookies, but this is rare. Much of the paranoia about cookies is unfounded; cookies cannot dig into your system and gather information about you.

History: Internet Explorer keeps a (limited) history of sites visited by the user. This list is in the directory named by the unnamed value in this subkey.

Last Update: This subkey contains information about the last version of Internet Explorer components installed. Information may consist of a product's date or a version number.

Cryptography\AUTH2UPD: Contains the version number of the currently installed cryptography component.

IEXPLOREV2: Contains the product date for Internet Explorer 2.

IEXPLOREV3: Contains the product date for Internet Explorer 3.

IEXPLOREV4: Contains the product date for Internet Explorer 4.



NOTE With the Internet Explorer Last Update information, it is not possible to determine which, if any, versions are installed. It is safe to assume, however, that the latest version is probably the currently installed version. Of course, a user might have gone back to an earlier version—if so, it would not be possible to determine this from the Last Update subkey.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

SO: SO is short for security options. To get to these options, go to the Security tab of the Internet Options dialog box in Internet Explorer, then click the Settings button to display the Security Settings dialog box. Here you will find security-based options, subkeys for which include:

ACTIVE_CONTENT\ACTIVEX: Runs ActiveX controls and plug-ins.

ACTIVE_CONTENT\ENABLE: Downloads signed ActiveX objects.

ACTIVE_CONTENT\SAFETY: Initializes and scripts any ActiveX controls that have not been marked as safe.

ACTIVE_CONTENT\SCRIPTSAFE: Scripts any ActiveX controls that have been marked as safe.

ACTIVE_CONTENT\UNSIGNEDACTIVEX: Downloads unsigned ActiveX controls.

AUTH\LOGON: Sets how to handle logon credentials.

DOWNLOAD\FILEDOWNLOAD: Sets whether to download files or not.

DOWNLOAD\FONTDOWNLOAD: Sets whether to download and install fonts or not.

JAVAPER\JAVA: Sets Java permissions.

MISC\DRAGDROP: Sets whether to allow drag-and-drop or cut-and-paste of files.

MISC\FORMDATA: Sets submission of unencrypted form data.

MISC\INSTALLDIT: Sets whether to allow installation of Desktop items.

MISC\LAUNCHING: Sets whether to allow launching a file or application in an <IFRAME>.

MISC\SOFTDIST: Sets software channel permissions.

SCRIPTING\SCRIPT: Sets whether to allow active scripting.

SCRIPTING\SCRIPTJAVA: Sets whether to allow Java scripting.

SOIEAK: Contains security options for IEAK (Internet Explorer Administration Kit). With IEAK, you can customize the setup of Internet Explorer, presetting preferences and options to suit a particular set of circumstances. As with the SO options, previously listed, these settings will appear in the Security Settings dialog box in Internet Explorer 5.



NOTE These are not the only options or settings that may be configured with IEAK. IEAK allows customization of the installation for an ISP, for example, where the user's default home page will be the ISP's page.

ACTIVE_CONTENT\ACTIVEX: Runs ActiveX controls and plug-ins.

ACTIVE_CONTENT\ENABLE: Downloads signed ActiveX objects.

ACTIVE_CONTENT\SAFETY: Initializes and scripts any ActiveX controls that have not been marked

as safe.

ACTIVE_CONTENT\SCRIPTSAFE: Scripts any ActiveX controls that have been marked as safe.

ACTIVE_CONTENT\UNSIGNEDACTIVEX: Downloads unsigned ActiveX controls.

AUTH\LOGON: Sets how to handle logon credentials.

DOWNLOAD\FILEDOWNLOAD: Sets whether or not to download files.

DOWNLOAD\FONTDOWNLOAD: Sets whether or not to download and install fonts.

JAVAPER\JAVA: Sets Java permissions.

MISC\DRAGDROP: Sets whether or not to allow drag-and-drop or cut-and-paste of files.

MISC\FORMDATA: Sets submission of unencrypted form data.

MISC\INSTALLDT: Sets whether or not to allow installation of Desktop items.

MISC\LAUNCHING: Sets whether or not to allow launching a file or application in an <IFRAME>.

MISC\SOFTDIST: Sets software channel permissions.

SCRIPTING\SCRIPT: Sets whether or not to allow active scripting.

SCRIPTING\SCRIPTJAVA: Sets whether or not to allow Java scripting.

Subscription Folder: Holds certain subscribed objects.

TemplatePolicies: These settings initialize (and reset) the SO (security options) for Internet Explorer. The original factory default is Medium, which provides a reasonable medium between excessive safety and minimal safety.

High: Typically these settings will keep your system as safe as possible.

Low: These settings offer little safety to your system.

Medium: These settings offer a compromise between safety and ease of use.

Url History: Four entries in Url History manage the history list, including the cache limit (number of entries in the history list), the number of days to keep the cache (20 days is the default), and the directory where the history cache is kept.

User Agent: Contains a subkey used to manage MSN entries.

UA Tokens: Two entries for MSN (Microsoft Network) exist in this subkey, one for each version (2.0 and 2.5) of MSN.

ZoneMap: Four predefined zones which are groupings of Internet sites based on security issues, are contained in Internet Explorer 4. The user is able to set zone attributes (see SO, above) for each zone, and assign sites to a specific zone as desired. ZoneMap contains subkeys that define which sites fit within a specific zone (local sites not in other zones, sites that bypass the proxy server, and all UNC paths).

Domains: Typically contains an empty subkey.

ProtocolDefaults: Contains the various protocols allowed, such as file, ftp, and http.

Ranges: Contains a place where the buffalo roam, and also entries (if any) for zone ranges.

Zones: Contains definitions of the four default zones, plus the local computer (included but not a zone, as such).

0: The first zone is not a zone at all. Just your computer.

1: The local intranet zone is for sites within your own organization. Generally, all the local intranet sites can be trusted.

2: The trusted sites zone is for intranet and Internet sites that you trust to have safe content (for your computer, but not necessarily for you).

3: The everyone else zone is for sites that you have not placed in any other zone. This is the default zone.

4: The “I really don’t trust this site” zone is for sites that have content that is not tested, not known, or otherwise considered to be unsafe for your computer. Maybe call this the *Twilight Zone*?

CurrentVersion\IPConfMSP

Found only in Windows 2000, the IPConfMSP subkey contains information that Windows 2000 uses to manage Media Stream Providers.

CurrentVersion\IPConfTSP

Found only in Windows 2000, the IPConfTSP subkey contains information that Windows 2000 uses to manage Telephony Service Providers.

CurrentVersion\MCD

MCD is the OpenGL mini-client driver. In this model, the driver is responsible for hardware-accelerated features and handler software for all other features. The MCD subkey typically contains about six settings for MCD functionality. Most users only use OpenGL for screen savers. The Pipes screen saver is an example of an OpenGL program.

CurrentVersion\ModuleUsage

The ModuleUsage subkey contains a listing of modules, typically ActiveX controls and UUIDs. In the subkeys within ModuleUsage, there is information such as the module's owner (if known).

CurrentVersion\MS-DOS Emulation

If you're opening an MS-DOS application, and the application does not have its own PIF file, settings for the application's display are found in the MS-DOS Emulation subkey. In MS-DOS Emulation, a single subkey named Font controls the display's attributes:

Font: Contains the name of the font used for MS-DOS applications. The default is Lucida Console.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CurrentVersion\netcache

Found only in Windows 2000, the netcache subkey contains information that Windows 2000 uses to manage whether to enable the network cache, the size of the network cache, and whether the entire network cache is encrypted.

CurrentVersion\Nls

NLS (National Language Support) provides the support to manage and display characters using the Unicode character sets. With Unicode, it is possible to display characters from multiple languages at one time. The Nls subkey contains the following:

LocaleMapIDs: Contains a table of lookup values for NLS languages.

CurrentVersion\Policies

The Policies subkey manages RASC (Recreational Software Advisory Council) ratings. The Internet, Internet Explorer, and some games use these settings. This key contains a subkey called Explorer as well as other subkeys:

Explorer: Contains one entry by default. FileName0 contains the name of the RSAC ratings definition file; this file is text and is editable with Notepad. The second entry, called Key, contains a binary value.

ShowSuperHidden: A flag that tells Explorer whether it should show files and directories that have the super-hidden attributes applied to them. Super-hidden files/folders have both the system and hidden attributes set. These files are not visible through the GUI by default. You can still get to them through the command prompt or by disabling this in Folder Options, View tab, by unchecking Hide Protected Operating System Files. (They will not be visible unless you have the Show Hidden Files and Folders option selected also.)

NonEnum: Information about non-enumerated objects. One value is present by default:

{BDEADF00-C265-11D0-BCED-00A0C90AB50F}: The handler for Web folders.

Ratings: May contain two value entries and two subkeys. The value entry FileName0 contains the name of the RSAC ratings definition file. The second entry, Key, contains a binary value. The subkeys that may exist in Ratings are:

.Default: Contains three ratings-oriented keys: Allow_Unknowns, Enabled, and PleaseMom. Each is a binary value.

.Default\http://www.rsac.org/ratingsv01.html: Contains four DWORD values: 1, n, s, and v.



System: Contains a number of useful keys:

disablecad: Disabling the key combination Ctrl+Alt+Del is useful for some environments.

dontdisplaylastusername: Controls whether the logon screen will display the user ID of the last user to logon.

legalnoticecaption: The legal notice is a message that is displayed before a user is allowed to logon. The user must click OK to dismiss this message, allowing management to enforce rules or policy.

legalnoticetext: The text in the legal notice message box.

shutdownwithoutlogon: Controls whether Windows 2000 can be shut down without logging on first. The default is true for non-server installations, and false for servers. I often turn this option on when setting up servers, because if a user wanted to shut down the server, they could simply use the power switch.

DisableNT4Policy: Windows NT 4 policy can be enforced on Windows 2000 users, if desired.

CurrentVersion\RenameFiles

In Windows 2000, sometimes it is necessary for an application, when it is being installed, to remove a file for some reason. (The reason would be specific to the application.) Rather than deleting these files, which the user might need later should the application need to be removed, a common technique is to rename the files. Then, if necessary, they can be renamed back to their original names.

A few of the applications that rename files are:

- Sa
- Win
- WinMail
- WinNews
- WordPadAttribSet

CurrentVersion\Run

Here is one of those areas in the registry that you want to find, but never seem to be able to. The Run subkey contains the name of executables that will be run each time the system is started.

In one system that I have, the following are included in the Run subkey:

BrowserWebCheck: Contains Internet Explorer's application that uses pull technology to check the currency of subscribed Web pages.

H/PC Connection Agent: Contains a program that checks for an HPC (handheld PC) to be connected. If the program detects the HPC, it will automatically initiate logon for the HPC.

POINTER: Contains an enhanced mouse system, part of the Microsoft IntelliPoint program.

SystemTray: Contains the system tray.

TIPS: Contains the Mouse Tips program.

Most Windows 2000 systems only have an entry for SystemTray in Run. This subkey is much like the Start Menu\Programs\Startup directory—anything there will be run when a user logs on.



NOTE By putting items in CurrentVersion\Run, then protecting the registry key from modification, you can force users to open or run certain applications. They will be unable to change this behavior.

CurrentVersion\RunOnce

Once? When?

The RunOnce subkey allows executing a program the first time a user (any user) logs on, and does not allow the user to continue until they have exited the program(s). Once the program has completed execution, Windows 2000 will delete it from the RunOnce key.

To run a program one time, in RunOnce enter an arbitrary name as a value (the program's common name will

work fine here.); the string data for the program should be the program's fully qualified file name. For example:

```
JobRun = C:\Jobs\JobRun.exe
```

The value's data type should be REG_STRING.

The application runs after the next user logs on. It will not be necessary to restart the computer.

CurrentVersion\RunOnceEx

The RunOnceEx subkey is used by system components and Internet Explorer to run setup and configuration components. Works much like the RunOnce subkey.

CurrentVersion\Setup

The Setup subkey contains information including the boot directory (typically C:\), the installation source directory (often the drive letter of your CD-ROM drive), and the source path (often the same as the source directory).

After installing Windows, you may find that you want to change the CD-ROM's drive letter. (I use drive letters after S: for CD-ROM drives, for example.) If you don't tell Windows 2000 (in Setup) every time you attempt to change the installation (for instance, installing a new component or option), the Windows 2000 setup program will prompt you to insert the disk in the wrong drive, making the installation process more complicated. A simple change to the entries in this section will make the process much easier.

BaseWinOptions: May contain a number of subcomponents, all controlled by .inf files.

OC Manager: The master list of installed options and accessories.

OptionalComponents: Provides the status for each optional component installable with Windows 2000. This subkey contains a list of optional components and a set of corresponding subkeys, one for each optional component.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CurrentVersion\SharedDlls

The SharedDlls subkey contains .dll files shared between multiple applications. Windows maintains a list of all shared .dll files and a count of the number of applications using each shared file.

When removing an application using a shared .dll file, the uninstall program decreases the count by one. If the count becomes zero, Windows 2000 will prompt you to remove the shared .dll file.



NOTE Although this section implies that it is for .dll files only, actually any shared file may be included in the list.

CurrentVersion\Shell Extensions

Shell extensions are used to extend and expand the Windows user interface and capabilities. The Shell Extensions key contains a subkey, named Approved, where all shell extensions are stored.



NOTE For more information on shell extensions, check out Jeff Prosis's March 1995 *Microsoft Systems Journal* article titled "Writing Windows 95 Shell Extensions."

CurrentVersion\ShellScrap

The ShellScrap subkey, on most systems, holds one subkey, PriorityCacheFormats. In PriorityCacheFormats is a single value entry, named #3, that contains an empty string.

CurrentVersion\ShellServiceObjectDelayLoad

The ShellServiceObjectDelayLoad subkey loads objects subject to a delay. The delay allows the operating system to finish initializing, establish connections, etc. Most systems with Internet Explorer 5 installed load WebCheck. WebCheck is responsible for subscription maintenance. Other items in this subkey are Network.ConnectionTray and SysTray.

CurrentVersion\StillImage

Found only in Windows 2000, the StillImage subkey contains information that Windows 2000 uses to manage the Kodak imaging system.



Desktop Library
Click to access!

CurrentVersion\Syncmgr

Found only in Windows 2000, the Syncmgr subkey contains information that Windows 2000 uses to manage the synchronization of folders.

CurrentVersion\Telephony

Windows 2000 works with telecommunications. Modems and telephones establish remote connections (and voice calls, at times). Within the Telephony key, there are a number of subkeys:

Country List: This subkey contains about 240 subkeys, one for each country defined. A typical country code is one to four digits and matches the telephone company's country code. For example, the country code for Thailand is 66. (To make a long distance telephone call to Thailand, I'd dial 001-66, where the 001 is the overseas access code and 66 is the country code.) Information in each country subkey includes:

CountryCode: Contains a DWORD value that should be equal to the country code. (Remember that this value is displayed in hexadecimal format.) This code would have to be changed if a country's country code were to change, although this is unlikely.

Name: Contains a string with the country's name, such as Thailand.

InternationalRule: Contains the rules used to dial numbers in this country. (See the next sidebar for more on rules.)

LongDistanceRule: Contains the rules used to dial long distance in this country. (See the box below for more on rules.)

SameAreaRule: Contains the rules used to dial local numbers in this country. (See the box below for more on rules.)



TIP Need a list of all the countries in the world? Here they are, along with the applicable telephone country codes. Export this subkey of the registry to a text file, and use an editor to clean up the list!

Rules, Rules, and More Rules

Look in the InternationalRule, LongDistanceRule, and SameAreaRule subkeys given above; you'll see a jumble of letters and numbers. Each has meaning. For example:

- 0-9 Indicates a number that is to be dialed as entered.
- ABCD Indicates touch-tone characters to be dialed, only usable on tone dial systems. (This produces the special tones named A, B, C, and D.)
- E Dial the country code.
- F Dial the area code or city code.
- G Dial the local number.
- H Dial the card number.
- * Dial a * tone.
- # Dial a # tone.
- T Indicates subsequent numbers dialed as tone dial.
- P Indicates subsequent numbers dialed as pulse dial.
- , Pause for a fixed period of time (typically 1 second).
- ! Flash the hook (1/2 second on-hook, 1/2 second off-hook).
- W Wait for second dial tone (outside line dial tone).
- @ Wait for silent answer (ringback followed by silence for 5 seconds).
- \$ Wait for calling-card prompt tone.
- ? Pause for user input.

Using Thailand as our example:

InternationalRule = 001EFG

(Dial 001, the country code, the city code, the local number.)

LongDistanceRule = 0FG

(Dial 0, the city code, the local number.)

SameAreaRule = G

(Dial the local number.)

That's all folks, an easy set of rules! With these rules it's easy to add new countries (they pop up all the time, right?) if necessary. What with the sometimes major changes to area codes, which are equivalent to city codes in other countries, it is sometimes necessary to modify the United States entry. You can set rules in the Change Calling Card dialog box by clicking the Rules button.

Locations: Each user may have zero, one, or more locations defined. (Actually, each user should have one location: the user's current or home location.) Each location defined is stored in the Locations subkey, as Location0, Location1, etc.

Providers: Providers are the connections between Windows 2000 and the modem or other telecommunications device. The most common provider is the Unimodem driver, though there are also other drivers, including the TAPI interface.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library

[Click to access!](#)

CurrentVersion\Unimodem

The Unimodem driver is a universal modem driver (see, now Unimodem makes sense) used to control virtually all industry-compatible AT-command modems, also known as Hayes-compatible modems. Most standard modems must be connected to a POTS (plain old telephone service) line. In other words, lines that are not digital are controlled by the Unimodem driver.

The Unimodem driver also controls direct connections between two computers connected via a serial cable. Though good speed performance is impossible, serial cable connections are used when connecting some notebooks and most PDAs (personal digital assistants) and HPCs (handheld PCs). Note that some systems use an IR (infrared) link for these devices, too.



NOTE Please note that Windows 2000/NT 4 and Windows CE version 1 are not compatible. It will be necessary to upgrade to Windows CE version 2.x or later to connect an HPC to your Windows 2000/NT 4 system.

DeviceSpecific: Contains subkeys for each connection. For example, a typical system will have a subkey under DeviceSpecific for each modem type installed, and one for direct serial cable connections if installed. Each entry contains information that the device, modem, or connection might send to the host computer.

CurrentVersion\Uninstall

In the Control Panel's Add/Remove Programs applet, there is a list of applications to remove automatically. Using this feature, the removal will be smooth and will not cause problems with system stability.



WARNING This assumes that the applications designer did a credible job of creating his remove system. If the application does not have a good uninstaller, you may still have problems. No one, other than the supplier of an application, can assure you that the uninstall will go smoothly. Before uninstalling anything, make sure you have a backup of the system, the application (all of it), and the registry. With good backups, it is possible (although nothing is guaranteed) that you may be able to recover from an uninstall gone awry.

The Uninstall subkey contains a subkey for each component that is automatically uninstallable. For example, in the Uninstall\IntelliPoint subkey, you'll find the following:

DisplayName: Contains a REG_SZ value that holds the string Microsoft IntelliPoint.

UninstallString: Contains a value that holds the string C:\progra-1\MICROS-2\Mouse\UNINSTALL.EXE.

When you select Microsoft IntelliPoint in the Add/Remove Programs applet, the program or object in the UninstallString entry is executed, performing the uninstallation. Typically, for a system component such as the IntelliPoint mouse driver being uninstalled, it must reinstall the original component.



TIP Ever manually uninstalled a program and then realized that the Add/Remove Programs list had an uninstall for the program? Easy fix: delete the applicable subkey from the CurrentVersion\Uninstall key. Careful, don't remove the wrong one.

CurrentVersion\URL

Used with Internet Explorer, the URL subkey provides a default prefix for a URL when the user does not enter one. For example, I'm in the habit of accessing my Web page by typing in the following:

```
<www.mv.net/ipusers/darkstar>
```

When, in fact, the full URL is:

<http://www.mv.net/ipusers/darkstar>

Internet Explorer, using information stored in CurrentVersion\URL, determines that the default prefix should actually be http://.

DefaultPrefix: Contains the default prefix (usually http://) used when the user does not enter a prefix and the initial characters of the URL do not tell Internet Explorer what prefix from the prefixes list (below) to use. The default prefix can be changed if the user is primarily using FTP or Gopher, for example.

Prefixes: Contains a list of all valid prefixes, based on the initial part of the URL. For example, if the URL starts with www, or www., the prefix would be http://. If the URL starts with ftp, or ftp., then the prefix would be ftp://. The prefixes defined by default (you may add more if you wish) are:

Beginning of URL

ftp
ftp.
gopher
gopher.
home
home.
mosaic
mosaic.
www
www.

Prefix

ftp://
ftp://
gopher://
gopher://
http://
http://
http://
http://
http://
http://

CurrentVersion\WebCheck

Found only in Windows 2000, the WebCheck subkey contains information that Windows 2000 uses to manage Internet Explorer's customization on a per-user basis.

CurrentVersion>Welcome

Found only in Windows 2000, the Welcome subkey contains information that Windows 2000 uses to manage the Internet Connection Wizard. The single subkey in this key is called ICW; it contains a flag indicating whether the Internet Connection Wizard has been run or not.

CurrentVersion\WindowsUpdate

Found only in Windows 2000, the WindowsUpdate subkey contains information that Windows 2000 uses to manage OEM installations.

CurrentVersion\WinLogon

Found only in Windows 2000, the WinLogon subkey contains information that Windows 2000 uses to manage logon options. One object, DisableLockWorkstation, has a REG_DWORD value. With this object, the

LockWorkstation feature can be enabled and disabled as needed.

Help

The Help subkey contains a list of help files and their locations. These are used when, inside an application, the user either presses F1 (for help), or selects the What's This button and clicks on a control or object in the application's user interface.

It is possible to remove entries from this section, if desired, when you know for sure that the help file is either no longer used or has been removed.



TIP If you find that pressing F1 or selecting What's This brings up a WinHelp error, indicating that WinHelp cannot find the help file, search for the file; if you can find it, WinHelp will update this subkey to indicate this file's location.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

ITStorage

The ITStorage subkey is used with the Microsoft HTML Help control (an ActiveX control) to display help for HTML documents in Internet Explorer.

Finders: For each type of HTML help file, an entry is created. Each entry has a name equal to the extension of the help file. For instance, the CHM HTML help files are listed as being serviced by a specific control, identified by a UUID.

Windows Messaging Subsystem

Another subkey you'll find under HKEY_LOCAL_MACHINE\Software\Microsoft on most Windows 2000 systems is Windows Messaging Subsystem. MAPI uses this subkey, which contains a list of all MAPI-enabled applications.

Windows NT

Under HKEY_LOCAL_MACHINE\Software\Microsoft, you'll also find a Windows NT subkey. Much like the Windows subkey (described earlier), the Windows NT subkey sets a number of Windows NT operating parameters. Microsoft did not rename this (preexisting) key when Windows NT 5 was renamed "Windows 2000."

There is only one subkey in the Windows NT key. This subkey, CurrentVersion, contains about 30 subkeys and perhaps 15 value entries in a typical installation. Unlike the Windows subkey, the number of entries in Windows NT is relatively constant between different installations.

CurrentVersion

CurrentVersion contains a number of value entries. These entries hold information about the installation:

CSDVersion: Contains the level of the system. By level, I mean which service packs have been installed (if any). Remember, service packs are cumulative—installing Service Pack 3 automatically installs both Service Pack 1 and Service Pack 2. A system for which there is no installed service pack may not have a CSDVersion object.

CurrentBuild: Contains an obsolete data value containing old version and build information. Do not use this value; use CurrentBuildNumber to determine the build of Windows that is running.

CurrentBuildNumber: Contains a number that indicates which build of Windows NT is running. A higher number indicates a later operating system build. During the development process, build



numbers are incremented each time the developers create a complete operating system, sometimes daily.

CurrentType: Contains information on whether the installation is uniprocessor or multi-processor.

CurrentVersion: Contains the Windows NT version number, such as 4.0. Microsoft sometimes uses sub-version numbers, such as 3.11 or 3.51. The Windows 2000 version of NT is 5.0.

DigitalProductID: The Windows product ID and other binary information are stored here.

HWID: Contains a value of not used.

InstallDate: Contains information on the Windows 2000 installation date. This value is the number of seconds since January 1, 1970, and these dates remain valid until early 2038—not much of a Y2K problem there.

PathName: Contains information on the Windows 2000 installation path.

ProductID: Contains the Windows 2000 product ID. If Windows is installed from something other than OEM media, the product ID will consist of a total of twenty digits: five lead digits, the first three digits of the user's CD key, the last seven digits of the user's CD key, and five trailing digits. The leading and trailing digit numbers will vary from installation to installation. For OEM media installations, the product ID will be equal to the OEM CD key. In both cases, the CD key is written on a small yellow sticker on the back of the CD jewel case.

ProductName: The actual name of the operating system—for example, Microsoft Windows 2000.

RegDone: Tells if the user has registered the copy of Windows 2000.

RegisteredOrganization: Contains the name of your company or organization, as you entered it during setup. If your company or organization name changes, you can edit this value.

RegisteredOwner: Contains the name as you entered it during setup. If your name changes (maybe you inherited the computer from your predecessor?), you can edit this value.

Software Type: Contains the string SYSTEM.

SourcePath: Contains the source path you used to install Windows 2000. If you reassign CD-ROM drive letters (I do, to keep all CD-ROM drives at the end of the alphabet, using letters S through Z), you can edit this value to change the installation source path. This path could be a network path, if the installation is from a shared resource.

SystemRoot: Contains information used to create the `%systemroot%` environment variable, the base directory that Windows 2000 is installed in. Be cautious about changing this value and realize that Windows 2000, when booting, will update this registry entry anyway. There may be other locations where the Windows 2000 directory is coded without using the `%systemroot%` variable. To change the directory without reinstalling Windows 2000, see the “Hints and Kinks from the Experts” section at the end of this chapter.

CurrentVersion\AeDebug

Windows 2000 will launch a debugger when there is an application or system failure. A debugger is a program that will either save information about the failure or allow interactive debugging. Most users who are not developers will simply use Dr. Watson as their debugger. Dr. Watson is a simple program that saves vital information about what failed and why there was a failure to a debugging file.

For Dr. Watson users, the typical entries in AeDebug are as follows:

Auto: Contains a string value of 1 if automatic debugging is to be done, or 0 if no automatic debugging is to be done.

Debugger: Contains the name of the default debugger. If you have a debugger installed other than Dr. Watson, your debugger is listed here.

UserDebuggingHotKey: Allows a user to launch the debugger using a keystroke combination. Useful for developers, but the average user will find little use for this functionality.

Dr. Watson's Options

Dr. Watson, DRWTSN32.EXE, takes a number of command-line options when launched:

- Use the `-i` option to (re)install Dr. Watson as the default debugger. Use this option if a different debugger was installed in the past and you want to use Dr. Watson again.
- The `-g` option is ignored, but no error is generated. This option maintains compatibility with 16-bit (Windows 95 and Windows 3.x) versions of Dr. Watson.
- The `-p <pid>` option tells Dr. Watson to debug the process ID specified.
- The `-e <event>` option tells Dr. Watson to debug the event specified.
- Use `-?` to display a simple help screen of options.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CurrentVersion\Compatibility CurrentVersion\Compatibility2 CurrentVersion\Compatibility32

Within the three Compatibility objects are value entries for a number of legacy (older, preexisting) applications that are not very compatible with Windows 2000. A flag value (a hexadecimal number, expressed as a string) tells Windows about the incompatibility and allows Windows to modify the operating system's behavior to compensate for the application's incompatibility.

What does Compatibility do? During beta testing of the operating system, testers inform Microsoft of applications that do not perform correctly. Microsoft may contact the application's supplier, and work with them to make the program work correctly. For some applications, especially for applications where there is a large installed base of users, Microsoft will make patches to the operating system to allow that application to function correctly. Usually these patches consist of doing things that make the new version of the operating system look like the original version for that application. These patches are turned on and off with a set of binary switches—when the application is loaded, Compatibility is checked, and the necessary patches are turned on for that application.



NOTE Realize that these patches will be only visible to the offending application and not to any others.

CurrentVersion\Console

Found only in Windows 2000, the Console subkey contains information that Windows 2000 uses to manage the appearance of console applications.

CurrentVersion\Drivers

Some drivers use the CurrentVersion\Drivers section of the registry. In certain Windows 2000 installations, two drivers—timer.driv and mmdrv.dll—are installed. The timer.driv driver creates certain timer functions on PC-compatible systems, and mmdrv.dll is the low-level wave, MIDI, and AUX support driver.

CurrentVersion\drivers.desc

The drivers.desc subkey contains descriptions of certain drivers installed under Windows 2000. The descriptions are text, intended to be people readable.



Desktop Library
Click to access!

CurrentVersion\Drivers32

Driver mapping for certain virtual devices, such as multimedia, is done in the Drivers32 subkey. For instance, the value entry named midi contains the value mmdrv.dll.

CurrentVersion\EFS

Found only in Windows 2000, the EFS subkey contains information used to manage the encrypted file system.

CurrentVersion\Embedding

Embeddable applications (such as PaintBrush and Sound Recorder) are listed in value entries in the Embedding subkey.

CurrentVersion\File Manager

The File Manager subkey contains one subkey:

AddOns: Contains a subkey containing information on add-on software products for File Manager. WinZip is an add-on software product that fits into this category.

CurrentVersion\Font Drivers

The Font Drivers subkey contains any needed drivers used to display fonts. The increased usage of TrueType fonts has minimized the use of this subkey.

CurrentVersion\FontCache

In Windows 2000, management of fonts is critical to system performance. Using a cache allows much better performance when displaying frequently used fonts. Windows creates bitmaps of the TrueType fonts, and then caches these bitmaps so that they do not have to be re-created.

The FontCache subkey contains three value entries:

MaxSize: The maximum size of the font cache.

MinIncrSize: The minimum increment size for the font cache.

MinInitSize: The minimum initial size for the font cache.

CurrentVersion\FontCache>LastFontSweep: This subkey contains one variable:

LastSweepTime: A binary value indicating the last time the font cache was cleaned.

CurrentVersion\Font Drivers

Found only in Windows 2000, the Font Drivers subkey contains information that Windows 2000 uses to manage non-TrueType fonts, such as Adobe type fonts.

CurrentVersion\FontDPI

Found only in Windows 2000, the FontDPI subkey contains information that Windows 2000 uses to manage the sizing of fonts, based on pixels.

CurrentVersion\FontMapper

Font mapping is an internal component of Windows 2000 that compares the attributes for a requested but not available font, and then matches these attributes with available physical fonts.

In FontMapper, attribute modifiers are supplied for the font mapper in Windows 2000.

CurrentVersion\Fonts

The Fonts subkey contains a list of currently installed fonts. The list is made up of keys in the form:

Font display name = fontfile

where *Font display name* is the display name, such as Arial (TrueType), and *fontfile* is the actual font file—arial.ttf, for Arial (TrueType).

The Font applet in Control Panel and other applications (indirectly, through the operating system) use the information in Fonts. It is possible to manually manipulate the font information; however, using the Fonts applet will make the process easier.

CurrentVersion\FontSubstitutes

Some fonts that are commonly called for by applications are not supplied with Windows 2000. These fonts are older, bitmapped fonts commonly used with early versions of Windows and Windows NT but no longer supplied or directly supported. These fonts are simply mapped to newer TrueType fonts.

Font substitutions are:

Old font	New font
Helv	MS Sans Serif
Helvetica	Arial
Microsoft Shell Dlg	MS Sans Serif
Times	Times New Roman
Tms Rmn	MS Serif

CurrentVersion\GRE_Initialize

The subkey `GRE_Initialize` contains objects used by the GRE (Graphics Rendering Engine), which displays a few fonts that Windows 2000 supports. These fonts are bitmapped fonts (not TrueType). Fonts handled or remapped by GRE are:

FIXEDFON.FON	vgafix.fon
FONT.SFON	vgasys.fon
OENFONT.FON	vgaem.fon

CurrentVersion\HotFix

New to Windows 2000, the `HotFix` subkey contains information that Windows 2000 uses to manage whether any hotfixes have been applied. Each applied hotfix will have its own subkey, named with the fix's Q number.

CurrentVersion\ICM

Found only in Windows 2000, the `ICM` subkey contains information used to manage image color matching.

CurrentVersion\Image File Execution Options

Used for debugging objects such as services or DCOM, the `Image File Execution Options` subkey specifies what debugger to use for a specific service or DCOM object.



NOTE Notice that the term *image file* refers to an executable image file, not a graphics file.

Your Image File Name Here without a path: In this example subkey, value entries show how to configure the debugger. More information on image file debugging is available from NuMega Lab's Web site at <http://www.numega.com/newsletters/apr96.htm>.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

This site is brought to you by
EarthWeb



HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH



Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)

[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC



Desktop Library
[Click to access!](#)

CurrentVersion\IniFileMapping

The IniFileMapping subkey maps .ini files (as they were used with early versions of Windows) to registry keys. In all cases, the entries in IniFileMapping point to other registry entries.

CurrentVersion\LanguagePack

Found only in Windows 2000, the LanguagePack subkey contains information that Windows 2000 uses to manage installed second languages.

CurrentVersion\LastFontSweep

Found only in Windows 2000, the LastFontSweep subkey contains information that Windows 2000 uses to record the last time that the font cache was cleaned.

CurrentVersion\MCI

The MCI subkey contains the MCI (Media Control Interface) drivers. Most systems with an audio card will have four entries:

AVIVideo: Contains the AVI (video files) driver, mciavi.driv.

CDAudio: Contains the CD audio (music) player driver, mcicda.driv.

Sequencer: Contains the MIDI (sequencer) driver, mciseq.driv.

WaveAudio: Contains the wave file (audio files) driver, mciwave.driv.

CurrentVersion\MCI Extensions

The subkey MCI Extensions holds multimedia file extensions and the driver used to handle these objects. For example, the following entry:

mpeg = MPEGVideo

denotes that Windows 2000 should use the MPEGVideo driver to process MPEG files.

CurrentVersion\MCI32

The MCI32 subkey contains 32-bit MCI (Media Control Interface) drivers. Most systems with an audio card will have five entries:

AVIVideo: Contains the AVI (video files) driver, mciavi32.driv.

CDAudio: Contains the CD audio (music) player driver, mcicda.driv.

MPEGVideo: Contains the MPEG (video) driver, mciqtz32.dll.

Sequencer: Contains the MIDI (sequencer) driver, mciseq.driv.

WaveAudio: Contains the wave file (audio files) driver, mciwave.driv.



NOTE In MCI32, some drivers are common with CurrentVersion\MCI.

CurrentVersion\Midimap

MIDI (Musical Instrument Digital Interface) configuration is saved in the Midimap subkey. MIDI creates music using sound (instrument musical note) definitions, combined with the music's score. The score (in a special format) tells the computer how to "play" each instrument. As might be expected, the computer does not make many mistakes, assuming the score has been properly entered into the MIDI file.

Better-quality sound systems use actual recordings of instruments playing specific notes to create a very high quality sound.

CurrentVersion\ModuleCompatibility

In the ModuleCompatibility subkey, you will find entries much like those in CurrentVersion\Compatibility. A flag value (a hexadecimal number, expressed as a string) tells Windows 2000 about the incompatibility and allows Windows 2000 to modify the operating system's behavior to compensate for the application's incompatibility.

Each entry lists a module and a compatibility flag. For example:

MYST = 0x8000

CurrentVersion\Network

In the Network subkey (only in HKEY_LOCAL_MACHINE), there are four subkeys. There is some disagreement between what Microsoft documents should be in each subkey and what experience shows is actually there.

Shared Parameters: Documented to hold the single value entry Slow Mode, this object lists which servers and domains are accessed over a slow (typically dial-up or modem) connection. Using additional caching on these connections compensates for slow connections.

SMAddOns: Contains a pointer to Server Manager extension .dlls used to augment RAS.

UMAddOns: Contains a pointer to User Manager extension .dlls used to augment RAS.

World Full Access Shared Parameters: Documented to hold the value entry ExpandLogonDomain, this contains a value (yes or no) that defines whether Windows expands the Shared Directories list in the Connect Network Drive dialog box. Experience shows that the value entry named Slow Mode, used to list which servers and domains will be accessed over a slow connection (typically dial-up or modem), is also present in this subkey, as is the entry RAS Mode.

CurrentVersion\NetworkCard CurrentVersion\NetworkCards

The NetworkCard and NetworkCards objects are usually identical in content. For each network card installed (remember, servers can have multiple cards) and for remote access (RAS and/or DUN), there will be one subkey in NetworkCards. Subkeys are named with numbers, beginning with 1. In each is a subkey called NetRules. An example, using a 3Com 3C-590 PCI Ethernet card, is shown here:

1: Contains six entries, plus the subkey NetRules. The entries are:

```
Description : REG_SZ : 3Com Etherlink III Bus-Master Adapter (3C590)
InstallDate : REG_DWORD : <a date, expressed as the number of seconds since January
Manufacturer : REG_SZ : 3Com
ProductName : REG_SZ : E159X
ServiceName : REG_SZ : E159x1
Title : REG_SZ : [1] 3Com Etherlink III PCI Buss-Master Adapter (3C590)
```

1\NetRules: Contains the following entries:

```
bindform : REG_SZ : "Ei59x1" yes yes container
class : REG_MULTI_SZ : Ei59xAdapter basic
InfName : REG_SZ : oemnad0.inf
InfOption : REG_SZ : 3C590
type : REG_SZ : ei59x ei59xAdapter
```

CurrentVersion\NTVersionOfLastBackup

Found only in Windows 2000, the NTVersionOfLastBackup subkey contains information that Windows 2000 uses to manage information on the operating system version.

CurrentVersion\OpenGLDrivers

Found only in Windows 2000, the OpenGLDrivers subkey contains information that Windows 2000 uses to manage OpenGL drivers.

CurrentVersion\Perflib

Monitoring system performance is a critical part of managing a Windows 2000 server. Performance monitoring allows the graphing of between 500 and 800 different parameters. The number of parameters, which may be monitored, varies depending on system components, packages, and configurations. There will be one or more subkeys in the Perflib key, one for each installed language. In this example, 009 is the subkey for U.S. English, the language that is installed on my computer:

CurrentVersion\Perflib\009: Contains the performance item names and descriptions. Each is listed in the Performance monitor's Add Counters dialog box. A REG_MULTI_SZ string contains the item name, and a second REG_MULTI_SZ string contains the item description.

Running the Performance monitor can be very instructional, especially for Windows servers. With the Performance monitor, it is possible to see which applications are "hogging" resources, making pigs of themselves, etc. Also, the Performance monitor is able to show usage for optional components such as Exchange Server, SQL Server, and IIS, to name a few.

CurrentVersion\Ports

Ports (serial, printer, file, and network ports) are configured in the Ports subkey. For most ports, no entries are needed. For serial ports, the default settings (typically 9600, n, 8, 1 as set by the Control Panel's Ports applet) for some options are stored here.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

Search Tips

Advanced Search

PUBLICATION LOOKUP

JUMP TO TOPIC


Desktop Library
[Click to access!](#)


Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Previous](#)[Table of Contents](#)[Next](#)

CurrentVersion\Print

Found only in Windows 2000, the Print subkey contains information that Windows 2000 uses to manage all printer resources. Each printer will have a subkey.

CurrentVersion\ProfileGuid

A list of user GUIDs, mapped to the user's SID.

CurrentVersion\ProfileList

User profiles for all users who use the computer are listed in the ProfileList subkey. A subkey is created for each user, named with the user's SID. Inside each of these subkeys are five variables:

CentralProfile: Contains the location of the user's central profile, if the profile is not stored on the local machine. This location will be specified as a UNC path name.

Flags: Contains a DWORD value, typically 0x2.

ProfileImagePath: Contains the location of the user's local profile. For users with a central profile, a local copy is kept in case the central profile is unavailable.

Sid: Contains the user's SID, as a binary object.

State: Contains a DWORD value indicating the user's current state.

CurrentVersion\related.desc

The related.desc subkey contains descriptions (if any) for items such as wave1 and wave2.

CurrentVersion\SeCEdit

Found only in Windows 2000, the SeCEdit subkey contains information that Windows 2000 uses to manage the Security Configuration Editor.

CurrentVersion\SrvWiz

Found only in Windows 2000, the SrvWiz subkey contains information that Windows 2000 uses to manage the Server Wizard program. Information in this object includes the name of the server and the NetBIOS name for the server.

CurrentVersion\Svchost

Found only in Windows 2000, the Svchost subkey contains information that Windows 2000 uses to manage the Services Host features.

CurrentVersion\Terminal Server

Found only in Windows 2000, the Terminal Server subkey contains information that Windows 2000 uses to configure the Terminal Services features of Windows 2000.

CurrentVersion\Time Zones

Windows 2000 is able to compensate for various time zones, and for DST (daylight saving time) in those areas where there is support for DST. Though technically there can only be 24 time zones (if we assumed even hours), actually there are several time zones where the time difference is only 30 minutes, and some time zones have different names depending on the country. Windows 2000 supports about 47 different time zones, spanning the entire world. The Control Panel's Date/Time applet uses these settings, and they are passed to other applications as data.

In the Time Zones key are subkeys for each possible time zone. Each time zone has information that includes the following:

Display: Contains a string describing the time zone, such as 'Eastern Time (US & Canada)'.

Dlt: Contains a string describing the daylight time, such as 'Eastern Daylight Time'.

MapID: Contains a string containing coordinates for the world map displayed by the Control Panel's Date/Time applet. Allows scrolling of the map, although unlike some versions of Windows 95, individual time zones are not highlighted.

TZI: Contains time zone information, a structure documented in KB article Q115231.

CurrentVersion\Tracing

Found only in Windows 2000, the Tracing subkey contains information that Windows 2000 uses to manage IIS tracing of certain events.

CurrentVersion\Type 1 Installer

Adobe Illustrator Type1 fonts may be used with Windows 2000 by converting these fonts to TrueType fonts using the Control Panel's Fonts applet. The Type 1 Installer key contains four subkeys:

Copyrights: Contains encoded copyright information for Type1 fonts.

LastType1Sweep: Contains the time of the last Type1 font sweep, if there was one.

Type 1 Fonts: Lists any Type1 fonts installed.

Upgraded Type1: Lists any upgraded Type1 fonts installed.

CurrentVersion\Userinstallable.drivers

Any user-installed drivers are listed in the Userinstallable.drivers subkey. An example of a user-installed driver might be the Sound Blaster driver. This driver is not installed automatically by Windows 2000.

In Windows 2000,



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CurrentVersion\Winlogon

Ah, we've come to an important part of the registry. 'Bout time, you say? WinLogon contains the configuration for the logon portion of Windows 2000. Many logon defaults are stored in this subkey. Each important entry is covered in detail below. The first list shows those entries present on all Windows 2000 installations. A list of optional components comes next.

AutoRestartShell: A value of 0x1 indicates that if the shell (usually Explorer) crashes, then Windows 2000 will automatically restart it. A value of 0x0 tells Windows 2000 to not restart the shell (the user will have to log off and log back on to restart the shell).

CachedLogonsCount: Contains the number of cached logons. If Windows 2000 is unable to find an authenticating PDC, Windows authenticates the user's logon using the information cache. The default value is 10 cached entries.

CachePrimaryDomain: Contains the name of the current domain. If no domain is established, the value will be NEWDOMAIN.

DCacheUpdate: Listed by some sources as not used by Windows NT 4, this entry does have a value, which may be a date/time variable.

DebugServerCommand: Used with the internal Microsoft debug tool used to debug CSRSS.EXE, a Windows 2000 Executive subsystem used to display graphics for text-mode applications. The default value of this string is no.

DefaultDomainName: Contains the default domain name, usually the domain the user last logged on to. The default value is NEWDOMAIN.

DefaultUserName: Contains the name of the last user who logged on successfully. Displayed if DontDisplayLastUserName has a value of 0.

DontDisplayLastUserName: If this REG_STRING value is 0, the name of the last user to successfully log on will automatically be displayed in the system logon dialog box. Setting this value to 1 will force users to enter both a username and a password to log on. If using automatic logon, make sure this value is set to 0.

LegalNoticeCaption: An optional dialog box may be displayed prior to logging on a user. This value contains the dialog box's title. Typical usage of this dialog box is to advise users of organizational policy (such as a policy that a user may not install software without management approval). It is used with the value LegalNoticeText.

LegalNoticeText: A dialog box may optionally be displayed prior to logging on a user. This value contains the dialog box's text. Typical usage of this dialog box is to advise users of organizational


Desktop Library
 Click to access!

policy (such as a policy that a user may not install software without management approval). Used with LegalNoticeTitle.

PowerdownAfterShutdown: For computers that support automatic power-down, Windows 2000 is able to perform power-down. Some computers (such as those with the ATX-style motherboards and many notebooks) support automatic power-down. Set this string value to 1 to enable automatic power-down.

ReportBootOk: Used to enable or disable automatic start-up acceptance. This happens after the first successful logon. Use a value of 0 when using alternative settings in BootVerification or BootVerificationProgram.

Shell: Sets the shell or user interface displayed by Windows 2000 once a user has successfully logged on. The default value is Explorer.exe, though for users who insist, Program Manager, File Manager, or another shell program can be substituted. For users not using Explorer, entries in Shell might be: taskman, progman, wowexec. If the shell cannot be executed, then Windows 2000 will execute the programs found in the shell directory.

ShutdownWithoutLogon: The Windows 2000 logon dialog box has a button to shut down the system. For Windows 2000 Professional users, this button is enabled, and for Windows 2000 Server users, this button is disabled. When ShutdownWithoutLogon is equal to 1, the button is enabled. Changing this button for a server allows a user who's not logged on to shut down the server—but then so does the power switch.

System: The default entry is lsass.exe, the Local Security Authority system. The lsass.exe program is the one that displays the logon dialog box (displayed when the user presses Ctrl+Alt+Del), and it uses many of the entries in this subkey.

Userinit: Specifies the executable(s) run when the user logs on. Typically, userinit.exe starts the shell program (see Shell, previously discussed), and nddeagnt.exe starts NetDDE (Network DDE).

VmApplet: Runs the Control Panel's System Properties applet.

There are a number of entries that don't exist by default in WinLogon. These entries may be added to modify the logon behavior of the system. The list below shows those optional WinLogon entries that I am aware of:

AllocateCDRoms: This value entry is used to restrict access to the CDs in the CD-ROM drives to the currently logged-on user only. Otherwise, if not restricted, CD-ROM contents and drives are accessible to all processes on the system.

AllocateFloppies: This value entry is used to restrict access to the floppy disks in the floppy drives to the currently logged-on user only. Otherwise, if not restricted, floppy contents and drives are accessible to all processes on the system.

AutoAdminLogon: When used with DefaultPassword and DefaultUserName, and when DontDisplayLastUserName is false (0), AutoAdminLogon logs on a user automatically without displaying the logon dialog box.

CacheLastUpdate: Used internally by WinLogon and should not be modified.

CacheValid: Used internally by WinLogon and should not be modified. The typical value is 1.

DcacheMinInterval: Contains a value, in seconds, that specifies the minimum time period before the domain list cache is refreshed. Since refreshing the domain list cache may be a lengthy process and because the cache is refreshed when a workstation is unlocked, it may be wise to change this value to a longer period of time. The range of this value is from 120 seconds to 86,400 seconds (that's one day).

DefaultPassword: Used with AutoAdminLogon to provide password information for an automatic logon.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)


WARNING Be careful of both `DefaultPassword` and `AutoAdminLogon` because they can create security problems if misused. Do not automatically log on a user with special privileges, and resist the urge to automatically log on the system administrator for servers. The password stored in `DefaultPassword` is not encrypted, and `AutoAdminLogon` doesn't know or care who is sitting in front of the machine when it starts up and logs on the user.

DeleteRoamingCache: To conserve disk space, locally cached profiles may be deleted when the user logs off using this value. Set `DeleteRoamingCache` to 1, and when the user logs off, their cached profile will be deleted. Computers used by many users who have roaming profiles can create cached profiles that consume a substantial amount of disk space.

KeepRasConnections: Normally when a user logs off, all RAS sessions are canceled. By setting `KeepRasConnections` to 1, the system will keep these RAS sessions active through logons and logoffs. This is useful when there is a permanent connection to a WAN (such as the Internet) that must be maintained.

LogonPrompt: Placing a string (up to 255 characters) in this value allows displaying an additional message to users when they log on. This value is similar to the `LegalNoticeText` value in that it provides a method to advise all users who log on of something.

PasswordExpiryWarning: Provides a warning, in days, to users when their password is going to expire. The default is 14 days, though a shorter period—typically 5 days—is often used.

ProfileDlgTimeOut: Contains the amount of time, in seconds, in which a user must respond to the choice of using a local or a roaming (remote) profile. The default time-out period is 30 seconds.

RASForce: Used to force checking of the Logon Using Dial-up Networking check box in the logon dialog box. If `RASForce` is set to 1, then it is checked; if 0, it is unchecked. This is meaningful only if RAS is installed and the computer is a member of a domain.

RunLogonScriptSync: Windows 2000 is able to run both the logon script (if there is one) and the initialization of the Program Manager shell at the same time. If `RunLogonScriptSync` is set to 1, the logon script will finish before Windows 2000 starts to run Program Manager.

SlowLinkDetectEnabled: Determines if slow link detection is enabled. Used with roaming (remote) profiles to help minimize the amount of time a user might have to wait before a local profile is used.

SlowLinkTimeOut: Sets the amount of time (in milliseconds) that the system will wait for a slow connection when loading a user's profile.

Taskman: When the name of an alternative task manager is specified, Windows will use the specified program. The default task manager is `taskmgr.exe`.

Welcome: Allows you to specify the text displayed in the title of the logon and lock/unlocked screens.


Desktop Library
 Click to access!

Include a leading space in this text to separate your text from the default title, which is retained.

You can use the MMC with the Policy Editor snap-in to modify many of these settings.

CurrentVersion\WOW

WOW, or Win16 on Win32, is a system where legacy 16-bit Windows applications may be run on newer 32-bit Windows NT/2000 systems. WOW emulates Windows 3.1 in standard (not enhanced) mode.

The WOW key contains eight subkeys:

boot: Contains drivers (communications, display, mouse, keyboard, etc.) used to emulate the Windows 3.1 mode.

boot.description: Contains a description of the computer system (hardware) such as display, keyboard, and mouse. This subkey also includes the language support requirement—for instance, English (American).

Compatibility: The concept of compatibility and applying minor patches to the operating system to allow legacy applications that are not directly compatible with the newer version is an old one. In this case, compatibility is maintained between the 3.1 emulation and earlier versions of Windows.

keyboard: Holds the keyboard driver .dll file and the keyboard type and subtype.

NonWindowsApp: Could contain two entries, ScreenLines and SwapDisk. Generally, this section is not used in WOW unless these lines existed in the previous installation of Windows 3.x.

SetupPrograms: Contains a list of commonly known installation and setup programs.

standard: Contains entries from the standard-mode settings of System.Ini. If Windows 2000 upgraded a Windows 3.x installation, and System.Ini had modifications affecting standard mode (the mode that WOW runs in), these entries are moved to this subkey.

WowFax: Contains only the subkey SupportedFaxDrivers.

WowFax\SupportedFaxDrivers: Contains the name of the supported fax drivers. The only entries, by default, are for WinFax, E-FAX, MAXFAXP, Quick Link II Fax, Quick Link Gold, and ProComm Plus.

ODBC

ODBC (Open Database Connectivity) is a system for Windows (both Windows 2000/NT and Windows 95/98) used by applications to share data stored in databases. With ODBC, an application is able to open a database written by another application and read (and sometimes update) data in the database using a set of common API calls.

ODBC, having been around for a while, originally worked using a setup file called ODBCINST.INI. Today that file's contents have been moved to the registry as a subkey under ODBC, called (guess!) ODBCINST.INI. In the ODBCINST.INI subkey, there will be information about each installed driver. Drivers commonly installed include Access, Oracle, SQL Server, FoxPro, dBASE, and text files.



NOTE To learn more about ODBC, I recommend one of my programming books for database programmers, such as *Database Developer's Guide with Visual C++ 4* (Que, ISBN 0-672-30913-0). Though this book is out of print, copies are still available from some sources and libraries.

Policies



The Policies subkey contains settings used for network conferencing and system certificates. Most systems will have only a few data values within Policies.

[Previous](#) [Table of Contents](#) [Next](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Program Groups

Program Groups contains Program Manager's program groups. If a user runs Program Manager and creates any groups, then these groups will appear in the Program Groups key.

The Program Groups key also contains is a single value:

ConvertedToLinks: This value indicates that program groups were converted to Explorer links. If this value is equal to 0x0 or does not exist, Windows 2000 will attempt to convert program groups to links.

Secure

There's no documentation on the HKEY_LOCAL_MACHINE\Software\Secure subkey. No entries seem to exist in this key.

Voice



Information about the Windows 2000 text-to-voice engine is contained in the Voice subkey.

Windows 3.1 Migration Status

The HKEY_LOCAL_MACHINE\Software\Windows 3.1 Migration Status subkey is used to tell Windows 2000 that the system has migrated the existing Windows 3.x .ini and Reg.dat files to Windows 2000.

Deleting the Windows 3.1 Migration Status key causes Windows 2000, on the next boot, to prompt the user to migrate. Afterwards, Windows 2000 will re-create the value and subkeys as needed.

Hints and Kinks from the Experts

In this chapter's hints and kinks, we cover a few hints on using the two registry editors.

How Do You Change the Default Windows 2000 Install Path?

If you want to change where Windows expects to find the installation CD, edit HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Sourcepath and



HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup\Sourcepath.

If your CD drive is your D drive: and you are working with an Intel-based machine, the values should be D:\I386 and D:\ respectively.

(Courtesy of Jerold Schulman.)

Why Does Windows 2000 Run an Unknown Job at Login?

If you can't find it in the start-up groups, do the following:

1. Check HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows.
2. Load and/or run keys.
3. Remove the offending program.

Here are other places where a program can be loaded at start-up in Windows 2000:

- In the Startup folder for the current user and all users.
- In the registry, in the following places:
 - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
 - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
 - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
 - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
 - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
 - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
 - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce

(Courtesy of Peter D. Hipson.)

How Do You Uninstall Apps Without Add/Remove or an Uninstall Program?

If you want to uninstall an application that has no uninstall program and is not listed in the Add/Remove Programs applet of Control Panel (or the uninstall doesn't work), just delete the directory files. Drill down in HKEY_LOCAL_MACHINE\Software\ and HKEY_CURRENT_USER\Software. Locate the application's entries and delete them.

Use Explorer to remove the entries from the Start menu in %windir%\Profiles\All Users\Start Menu\Programs\ and/or %windir%\Profiles\YourId\Start Menu\Programs\.

If there is an entry in the Add/Remove list, edit

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall. Locate the entry and delete it.

If the app has a service, edit HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services. Scroll down until you locate the service, then delete it.

If this app starts automatically and there is no entry in the Startup folder(s), use RegEdt32 to edit HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows. Load keys and run keys. Next, remove the offending value and reboot.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
CHAPTER 19**Introduction to HKEY_LOCAL_MACHINE\System and HKEY_CURRENT_CONFIG****FEATURING:**

<i>The Control Set</i>	525
<i>DISK</i>	575
<i>Mounted Devices</i>	576
<i>Select</i>	576
<i>Setup</i>	576
<i>HKEY_CURRENT_CONFIG</i>	577
<i>Hints and kinks from the experts</i>	581

The HKEY_LOCAL_MACHINE\System key contains information about the system and system configuration. In a typical installation, seven subkeys exist in HKEY_LOCAL_MACHINE\System:

CurrentControlSet: Windows 2000 has booted from this control set. It is typically a mapping of ControlSet001 or ControlSet002.

ControlSet001: This is the primary control set, used by default to boot Windows 2000.

ControlSet002: This is the backup control set, used to boot in the event that ControlSet001 fails.

DISK: This object contains parameters used by the Disk Administrator program. Under Windows NT 4, it includes CD-ROM mappings and other binary information. Windows 2000 uses this object differently, in that the Disk Administrator functionality is now part of MMC (Microsoft Management Console).

MountedDevices: This key shows disk drives that are available to the system.

Select: This small subkey contains information about which control set is used to boot the computer.



Setup: This is a small subkey with information about the initial setup (installation) of Windows 2000.

Some systems will have slightly different names for the two numbered control sets. Some computers won't have a ControlSet002. For example, your computer might have the following subkeys:

- ControlSet001
- ControlSet003

It is also possible, but unlikely, that there may be more than two numbered control sets.

Each control set key contains four objects: Control, Enum, Hardware Profiles, and Services. The Enum and Hardware Profiles subkeys are new to Windows 2000 and were added to support Plug and Play.

The hive HKEY_CURRENT_CONFIG is a mapping of HKEY_LOCAL_MACHINE\System\CurrentControlSet and information from HKEY_LOCAL_MACHINE\Software.

What Are Mapped Registry Subkeys?

Sometimes more than one name refers to a single registry subkey (control sets in particular do this, as does HKEY_CLASSES_ROOT). The process is simple. Consider the mythical Fizbin Company, the proud maker of Fizzits. (You do use Fizzits, don't you?) Fizbin found that with a high-tech product such as Fizzits, it was necessary to have a high-tech company. They also wanted to make it seem as if they were more international than they really were.

However, Fizbin has a number of stodgy stockholders, most of whom have never seen or used a Fizzit and have only a vague idea of what a Fizzit is or does. These stockholders were dead set against renaming the company for any reason.

As a compromise, the company would still be called the Fizbin Company. However, when doing business, they would use the name International Fizbin. Regardless of the name used, it's the same company. A letter written to the president of International Fizbin still goes to the president of Fizbin Company—one company, two names. Therefore, when the president of Fizbin Company hires a new marketing manager, she automatically becomes the marketing manager of International Fizbin, too.

CurrentControlSet

The current control set is the control set used to boot the computer. It is copied from ControlSet001, or from one of the other numbered control sets if ControlSet001 failed to boot, and is the main control set. Except for the contents of keys that may be different, ControlSet002 (or ControlSet003, if that is what your computer has) has a structure identical to CurrentControlSet.

CurrentControlSet consists of four subkeys:

Control: Consists of information used to control how Windows 2000 operates. This information controls everything from boot-up to networking parameters to Windows to WOW (Windows on Windows).

Enum: Contains information about hardware, the hardware state, legacy devices, and so on.

Hardware Profiles: You use Hardware Profiles to configure Windows 2000 for hardware platforms that change frequently. This is common when dealing with notebook computers, for example, as they may be either docked or undocked. An installation of Windows will have one or more hardware profiles. The use of hardware profiles is most helpful when running Windows 2000 on portable computers, particularly those with docking stations.

Services: Manages services, such as support for hardware. Change Services using the Control Panel's Services applet.

Control

The Control subkey has a number of data values used for booting and system initialization. Control also contains about 30 subkeys.

Control's value entries are:

CurrentUser: The name of the currently logged-on user. Actually, this entry always has the default value USERNAME because Windows 2000 does not update it.

RegistrySizeLimit: If you change the registry size limit from the default value of 8MB, RegistrySizeLimit will contain the maximum registry size in bytes. Though users are only able to set the registry size limit in MB, Windows will store the value as a DWORD containing the maximum registry size in bytes.

SystemStartOptions: This entry contains start-up options passed from firm-ware or the start-up contained in Boot.ini. Options could include debugging information (such as a debugging port and the debugging port parameters) and perhaps information on the system root directory.

WaitToKillService: This entry contains the time, in milliseconds, to wait before killing a service when Windows 2000 is shutting down. If this value is too small, Windows 2000 may kill a service before it has finished writing its data; if too large, a hung service will delay shutdown. It is best to leave the WaitToKillService value at its default value of 20000 unless you know you are having a problem.

ApmActive

The ApmActive object holds information about APM (Advanced Power Management), new to Windows 2000. One value entry, named Active, has a REG_DWORD value of either 1 (APM is active) or 0 (APM is not active). Use of APM requires that the hardware support this function. Most newer desktop and notebook computers support APM. If your computer doesn't support APM, or you are not using APM, then this subkey may be missing from your registry.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.


Datamation
Profit and Value from Information Technology
[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

 This site is brought to you by
 EarthWeb

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)
[KEYWORD SEARCH](#)


Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:
[Search Tips](#)
[Advanced Search](#)
[PUBLICATION LOOKUP](#)
[Previous](#) [Table of Contents](#) [Next](#)
[JUMP TO TOPIC](#)

Desktop Library
[Click to access!](#)

ApmLegalHal

New to Windows 2000, the ApmLegalHal object holds information about whether the hardware actually supports APM. Generally, Windows 2000 queries the BIOS to determine APM support. If your BIOS does not have support for APM, then this subkey may be missing from your computer.

Arbiters

New to Windows 2000, the Arbiters object holds information about bus arbitration. This is part of the support for Plug and Play that has been added to Windows 2000.

BackupRestore

New to Windows 2000, the BackupRestore object holds settable configuration information for the backup program that comes with Windows 2000. Included in this object are DllPaths, FilesNotToBackup, and KeysNotToRestore.

Biosinfo

New to Windows 2000, the Biosinfo object supports Plug and Play. Entries in this object include date codes for the BIOS and FullDecodeChipsetOverride, a value that indicates support for extended address decoding.

BootVerificationProgram

One entry in BootVerificationProgram, ImagePath, is a data value with a string variable that the boot verification program uses. This value will contain the filename of the boot verification program. Enter an empty string, or delete this value if no boot verification program is used.

To enable boot verification, it is also necessary to set ReportBootOk to 1 in HKEY_LOCAL_MACHINE\Software\Windows NT\CurrentVersion\WinLogon. If ReportBootOk is 0, automatic (default) start-up acceptance is disabled. This happens after the first logon that is successful. (ReportBootOK is defined in Chapter 18.)

Class

The Class subkey contains a number of GUIDs, one for each of the following. Notice that though Windows 2000 renames some items slightly (making them plural), the functionality of the object is the same despite

renaming.

- Batteries (unique to Windows 2000)
- Computer (unique to Windows 2000)
- Disk drives (unique to Windows 2000)
- Display adapters
- DVD/CD-ROM drives (new to Windows 2000)
- Floppy disk controllers (new to Windows 2000)
- Floppy disk drives (new to Windows 2000)
- Human Interface Devices (new to Windows 2000)
- IBM Digital Signal Processors (new to Windows 2000)
- IDE ATA/ATAPI controllers (new to Windows 2000)
- IEEE 1394 Bus host controllers (new to Windows 2000)
- Imaging devices (new to Windows 2000)
- Infrared devices (new to Windows 2000)
- Keyboard (new to Windows NT 4)
- Keyboards (new to Windows 2000)
- Medium Changers (new to Windows 2000)
- Memory technology driver (new to Windows 2000)
- Mice and other pointing devices (new to Windows 2000)
- Modem (new to Windows NT 4)
- Modems (new to Windows 2000)
- Monitors (new to Windows 2000)
- Mouse (new to Windows NT 4)
- Multifunction adapters (new to Windows 2000)
- Multiport serial adapters (new to Windows 2000)
- Network adapters
- Network Client (new to Windows 2000)
- Network Protocol (new to Windows 2000)
- Network Service (new to Windows 2000)
- Non-Plug-and-Play drivers (new to Windows 2000)
- NT Apm/Legacy Support (new to Windows 2000)
- Other devices
- PCMCIA adapters (new to Windows 2000)
- Ports (COM & LPT)
- Printer (new to Windows NT 4)
- Printers (new to Windows 2000)
- SCSI and RAID controllers (new to Windows 2000)
- SCSI controllers (new to Windows NT 4)
- Smart card readers (new to Windows 2000)
- Sound, video, and game controllers
- Storage volumes (new to Windows 2000)
- System devices (new to Windows 2000)
- Tape drives
- Universal Serial Bus controllers (new to Windows 2000)

Each of these subkeys contains one or more of the following value entries:

(Default): The default name as a string; for example, Mouse or Mice and other pointing devices. When using RegEdt32, the name (Default) displays as <No Name>.

Class: The device's class as a single word with no embedded spaces. It is a string that is similar to the

default entry. For mice and other pointing devices, the value is Mouse.

Default Service: A string defining the default service, usually the same as the Class entry.

Icon: An index to the object's icon.

Installer32: A string pointing to the program or system to install this type of device. The Control Panel, or the SysSetup.dll, installs many devices.

LegacyInfoOption: A string with information about legacy support. It is usually a string name for the device, similar to the Class entry.

NoDisplayClass: A flag, 1 or 0, indicating whether to display the class.

NoInstallClass: Contains information as to the installability of the device.

TroubleShooter-0: Contains information used for the interactive troubleshooter application in Windows 2000.

UpperFilters: A filter designed for the specific device.

Subkeys do not have all possible entries. The entries (Default) and Class are universal to all subkeys, while Icon, Installer32, and others exist in many (but not all) subkeys.

CoDeviceInstaller

New to Windows 2000, the CoDeviceInstaller object holds the CLSIDs of handlers for installations from removable media such as CD-ROM drives.

Com Name Arbiter

New to Windows 2000, Com Name Arbiter manages ComDB services.

ComputerName

ComputerName contains two subkeys and no value entries:

ActiveComputerName: This subkey includes a single value entry, ComputerName, with a string containing the computer name.

ComputerName: This subkey contains a single value entry, ComputerName, with a string containing the computer name.

Yes, both subkeys contain exactly the same thing.

ContentIndex

New to Windows 2000, the ContentIndex object works with the management of content indexing. It contains a number of subkeys for both catalogues and languages.

ContentIndexCommon

New to Windows 2000, ContentIndexCommon has no contents.

CrashControl

CrashControl brings to mind all kinds of marvelous things. However, the subkey CrashControl is actually a basic function for Windows 2000 when it fails at the system level. CrashControl options are generally set using the System applet's Startup and Recovery dialog box (see Figure 19.1). The Startup and Recovery dialog box is activated from the Advanced tab of the Control Panel's System Properties applet.



FIGURE 19.1 The Startup and Recovery system properties

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

CrashControl options are set in the Recovery section at the bottom of the dialog box. There are six value entries in CrashControl:

AutoReboot: The Automatically Reboot control state is set to zero if there is no automatic reboot after a STOP error.

CrashDumpEnabled: This is the Write Debugging Information To control state. It is set to zero if no dump is required after a STOP error.

DumpFile: This is the text control under the Write Debugging Information To control. It contains the name of the debugging file. This file will be as large as (or slightly larger than) the physical memory installed in the computer. Make sure the device to receive this file is large enough to hold the file.

KernelDumpOnly: Found only on Windows 2000, this value indicates whether to save all memory or only the Windows 2000 kernel in the dump file.

LogEvent: This is the "Write an event to the system log" control state. When set to zero, Windows will not make an event log entry after a STOP error.

Overwrite: This is the Overwrite Any Existing File control state. It is set to zero if there is no automatic reboot after a STOP error. If this value is set, Windows 2000 will create a new debugging file with a new name.

SendAlert: This is the Send an Administrative Alert control state. When set to zero, Windows will not send an administrative alert after a STOP error.

All entries in CrashControl are REG_DWORD except for DumpFile, which is REG_EXPAND_SZ. Some references indicate that these values are REG_SZ; this information is incorrect.

CriticalDeviceDatabase

New to Windows 2000, the CriticalDeviceDatabase object contains the list of devices critical to the operation of Windows. This list is PnP-based.

DeviceClasses

New to Windows 2000, the DeviceClasses object contains CLSID listings of all devices installed on the system.

FileSystem

Entries in FileSystem vary based on the installed file system(s). There are three value entries in FileSystem:



Win31FileSystem: When this value is 1, LFNs (long filenames) are disabled. This maintains compatibility with older operating systems, such as Windows 3.1. However, using this option may create compatibility issues with Windows NT 4 or Windows 95/98, and it should be set only if absolutely necessary. Also, do not set this option except immediately after installing Windows 2000. If you do, it may cause existing, installed applications to fail.

Win95TruncatedExtensions: The following behavior will take place depending on the setting of this option: Say you have two files, smith.htm and jones.html. If Win95TruncatedExtensions is equal to 1, the command DEL *.htm will delete both files. The command DEL *.html will delete only jones.html. When Win95TruncatedExtensions is equal to 0, the command DEL *.htm will delete only smith.htm. The command DEL *.html will delete only jones.html.

NtfsDisable8dot3NameCreation: If set to 1, Windows 2000 will not automatically generate standard 8.3 filenames. Without 8.3 filenames, any legacy DOS or Windows 3.x applications lacking LFN support will fail. They will not be able to open or otherwise use any file that has an LFN unless the file is renamed to a valid 8.3 filename.

NtfsEncryptionService: This value holds the name of the service used to encrypt files under the NTFS file system.



NOTE Some applications, including Microsoft Office, will not even install if NtfsDisable8dot3NameCreation is set.



WARNING Be careful about changing these options. Once installed, some systems do not expect that the state of the FileSystem entries will change, or that support for LFNs will change, either. This is especially true when changing from allowing LFNs to not allowing them.

GraphicsDrivers

By default, the GraphicsDrivers subkey contains one subkey called DCI (Display Control Interface). This standard subkey contains a value entry named Timeout, which has a default value of 7.



NOTE Microsoft dropped DCI support for Windows NT 4, and yet DCI remains in Windows 2000. Go figure.

One optional entry can be found in GraphicsDrivers: DisableUSWC. With certain higher-performance video cards, Uncached Speculative Write Combining (USWC) memory is uncached. In addition, certain computers have a memory conflict with USWC that may cause the user interface to fail to respond after certain drag-and-drop operations. The DisableUSWC entry does not have a value; its presence in the registry is sufficient to turn off USWC. This type of error is rare.

GroupOrderList

Each service in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services is listed in the GroupOrderList subkey along with a binary value indicating the order in which the group is to be loaded at system start-up time. Systems typically have the following groups:

- Base
- Boot Bus Extender (supported in Windows 2000 only)
- Extended base
- ExtendedBase (supported in Windows 2000 only)
- Filter
- Keyboard Class
- Keyboard Port
- Ndis
- NetBIOSGroup (supported in Windows 2000 only)
- Network (supported in Windows 2000 only)
- Parallel arbitrator

- Pnp Filter (supported in Windows 2000 only)
- PNP_TDI (supported in Windows 2000 only)
- Pointer Class
- Pointer Port
- Primary Disk
- SCSI CDROM Class
- SCSI Class
- SCSI miniport
- SpoolerGroup
- Streams Drivers (supported in Windows 2000 only)
- System Bus Extender
- Video
- Video Init
- Video Save

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

Hivelist

Hivelist contains the following value entries listing registry keys and their source files.

\REGISTRY\MACHINE\HARDWARE does not have a source file because Windows creates this key dynamically at boot time. Specific user information (those items that include a SID) will vary based on the currently logged-on user.

- \REGISTRY\MACHINE\HARDWARE =
- \REGISTRY\MACHINE\SECURITY = \Device\Harddisk0\Partition1\WINNTWS\System32\Config\SECURITY
- \REGISTRY\MACHINE\SOFTWARE = \Device\Harddisk0\Partition1\WINNTWS\System32\Config\SOFTWARE
- \REGISTRY\MACHINE\SYSTEM = \Device\Harddisk0\Partition1\WINNTWS\System32\Config\SYSTEM
- \REGISTRY\MACHINE\SAM = \Device\Harddisk0\Partition1\WINNTWS\System32\Config\SAM
- \REGISTRY\USER\DEFAULT = \Device\Harddisk0\Partition1\WINNTWS\System32\Config\DEFAULT
- \REGISTRY\USER\S-1-5-21-45749729-16073390-2133884337-500 = \Device\Harddisk0\Partition1\WINNTWS\Profiles\Administrator.000\ntuser.dat

Here is some further information about the preceding names:

\REGISTRY: The name for the registry itself

\MACHINE: HKEY_LOCAL_MACHINE

\USER: HKEY_USERS

Windows dynamically creates HKEY_CURRENT_CONFIG at boot time. Windows creates HKEY_CLASSES_ROOT from other registry entries at boot time. Windows creates HKEY_CURRENT_USER when a user logs on.

IDConfigDB

The IDConfigDB subkey is the identification for the current configuration. This key contains four value entries:

CurrentConfig: Indicates which control set is being used.

IsPortable: A value of 1 indicates that this is a portable computer.

PropertyProviders: Specifies the name of the .dll file used to display property sheets under Windows 2000. The default is profext.dll.

UserWaitInterval: Specifies the period of time a user waits. In the boot-up Hardware Profile/Configuration Recovery menu, Windows takes the default choice after the user waits the time

specified in UserWaitInterval. The value is in seconds; the default value is 30 seconds.

IDConfigDB also contains the hardware profiles in the Hardware Profiles subkey.

The Windows 2000 installation process will create one configuration for the user. This default configuration is Original Configuration. Any additional configurations that users create will also appear in the Hardware Profiles subkey.



NOTE See the Hardware Profiles tab in the System applet of the Control Panel to get more information about setting up multiple hardware profiles.

IDConfigDB also contains the Alias and CurrentDockInfo subkeys, in which Windows 2000 saves the docking status.

Keyboard Layout

There are two subkeys in Keyboard Layout, each dealing with supporting MS-DOS applications to use languages other than U.S. English:

DosKeybCodes: Contains a list of keyboard layouts and a two-letter (MS-DOS-compatible) country code.

DosKeybIDs: Contains a list of keyboard layouts and keyboard ID values.



NOTE See the Knowledge Base article Q117850, titled “MS-DOS 6.22 COUNTRY.TXT File,” for more information about support for MS-DOS applications.

Keyboard Layouts

Keyboard Layouts contains a subkey for each keyboard layout that Windows 2000 supports for Windows applications. Each layout subkey contains two entries:

Layout File: The name of the .dll file that manages the keyboard using that character set; for example, the Icelandic keyboard layout .dll file is named kbic.dll.

Layout Text: A string identifying the keyboard layout; for example, for Iceland, the string is Icelandic.

It is possible, though difficult, to create custom keyboard layouts.

Support is available to Windows 2000 users who wish to use the Dvorak keyboard layouts. Use the Regional Settings applet in the Control Panel to select either the Dvorak right- or left-hand layout. No special hardware is required, though the markings on the standard keyboard will be incorrect. This is because the Dvorak keyboard has a different keyboard layout with letters arranged based on how often they are used.

LSA

LSA, the Local Security Authority, locally validates security for user rights, secret objects, and trusted domain objects. LSA uses the msv1_0.dll file to do the actual validation of security. Within the LSA key is a subkey called msv1_0 that contains items for msv1_0



WARNING Microsoft strongly recommends that you do not touch anything in the LSA subkey. An incorrect entry or change could send the system into a state where no users are able to log on, and the system would have to be completely restored.

MediaCategories

MediaCategories is a key used primarily to describe multimedia controls and objects. Each item is a CLSID-named key containing two value entries. The first entry is Display, a REG_DWORD that indicates to Windows to display this object. The second entry is Name, a REG_SZ containing the name of the object.

MediaInterfaces

MediaInterfaces is a key used to describe multimedia interfaces.

MediaProperties

MediaProperties is a key used primarily to describe MIDI and other device properties. This key contains subkeys to describe any MIDI schemes (custom configurations in the Control Panel's Multimedia Properties applet) that a user has created. The Media Properties subkey called PrivateProperties contains information about objects like the system joystick.

MediaResources

MediaResources is a key used to describe the resources available for multimedia (specifically for MIDI) on the computer.

Two to four subkeys exist in this section:

DirectSound: Contains information about Windows 2000's support for DirectSound.

MCI: Contains information about MCI (Media Control Interface) devices.

MIDI: Contains subkeys for each installed physical and virtual device. The device subkeys contain definitions for instruments that the user has defined.

NonGeneralMIDIDriverList: Contains resource definitions (including instrumentation) for users with non-general MIDI hardware.

MediaSets

MediaSets is a key used to describe resources available for multimedia on the computer.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

Network

The Windows 2000 Network key houses much of Windows 2000's network configuration information. This information includes Connections, NcQueue, RefNames, and several CLSID-named objects.

NetworkProvider

The NetworkProvider key contains one object, which indicates the specific order of network providers. There is one value entry in Order, ProviderOrder, a REG_SZ string containing a comma-separated list of providers. The default value is LanmanWorkstation.

If you're using RAS (Remote Access Service), there is an option to disable automatic (ghosted) restoration of network connections at logon. Most users prefer to have connections restored automatically at logon. If RAS is not installed, you can enable ghosted connections by putting the RestoreConnection value entry in the NetworkProvider key. This REG_DWORD entry may contain a value of either 0x1 or 0x0. If the value is 0x0, Windows 2000 will ghost connections.

NLS

The NLS subkey holds Windows's National Language Support functionality. There are three or four possible subkeys in NLS. However, the OEMLocale subkey is not present in installations of Windows 2000 unless an OEM has customized Windows 2000 for a specific locale.

CodePage CodePage contains a series of value entries. Each value's name matches a code page ID. The value will contain a REG_SZ value equal to the filename for that code page. For code pages with no supporting file, the value entry's value will be an empty string.

CodePage also includes the following additional value entries and subkeys:

ACP: This value is the active (or default) code page used by Windows 2000.

MACCP: This value is the Macintosh active code page.

OEMCP: This value is the OEM code page to translate ANSI characters.

OEMHAL: This value is the OEM display of extended characters at a command prompt.

EUDCCodeRange: This subkey holds information for fonts classified as End User Defined Characters. The information is in value entries that indicate ranges of the double-byte character set that are usable for EUDC.

Language The Language subkey contains entries used to identify files that support different languages. Each value entry's name matches the ID for a language. The entry contains a REG_SZ data value equal to the name of the file to support that language. If there is no support for the language (the support files have not been installed), the value entry's contents will be an empty string.

Locale The Locale subkey contains entries used to identify files that support different locations. Each value entry's name matches the ID for a location. The entry contains a REG_SZ data value equal to a numeric 1 if there is support for the locale. The value has an empty string if there is no support for the locale.

OEMLocale Not normally present on systems, OEMs add the OEMLocale subkey to support systems in their specific locale. Entries in OEMLocale are similar to those in the CodePage subkey.

OEMLocale contains a series of value entries. Each value's name matches a code page ID. Each entry contains a REG_SZ value equal to the filename for that code page. For code pages with no supporting file, the entry's value will be an empty string.



NOTE Windows will only check the OEMLocale subkey if a specific locale ID is not found in the default locale file (LOCALE.NLS).

NTMS

New to Windows 2000, the NTMS object supports the Windows NT Media Services. Support includes OMID (On-Media Identifiers).

Pnp

New to Windows 2000, the Pnp object manages the Windows 2000 Plug-and-Play capabilities. Much of this information is specific to certain computers and devices. This object includes the following two subkeys:

Pci: Contains PCI device information.

PciIrqRouting: Contains information about the actual handling of the PCI bus's IRQ routing. Remember—PCI supports sharing IRQs.

Print

The Print key contains all the information accessed by Windows 2000 when a printer is being used. In addition to the subkeys documented next, there are also a few value entries in the Print key:

BeepEnabled: Enables or disables a beep when a printer error is detected. Set to 0x1 to turn on beeps.

MajorVersion: The version number's high digit(s); for a product with a version number of 4.3, the major version is 4.

MinorVersion: The version number's low digit(s); for a product with a version number of 4.3, the minor version is 3.

NoRemotePrinterDrivers: Contains the name of print drivers incompatible with remote connections. This entry typically has Windows NT Fax Driver as a value.

PortThreadPriority: The Windows 2000 kernel priority for the printer driver. A value of 0 indicates normal priority; -1 indicates lower than normal priority; and 1 indicates a higher than normal priority.

PriorityClass: Used to set the priority class for the print spooler. A value of 0 (or no value) indicates a default priority class will be used. The default priority class is 7 for Windows 2000 Professional and 9 for Windows 2000 servers. Coding any other value will be translated to the priority class for servers, 9.

RouterCacheSize: The size of the router cache, in pages.

SchedulerThreadPriority: Sets the priority for the scheduler. Setting SchedulerThreadPriority to 1 sets the priority to above normal; 0 sets the priority to normal; and -1 (0xFFFFFFFF) sets the priority to below normal.

Upgrade: A flag to indicate the upgrade status.

The Print key contains five or six subkeys, discussed next.

Environments The Environments key contains subkeys for each possible platform. Each platform key contains two subkeys: Drivers and Print Processors. The platform subkeys are:

Windows 4.0: For Windows 95 drivers.

Windows NT Alpha_AXP: Provides support for Digital Alpha systems.

Windows NT PowerPC: Provides support for PowerPC systems.

Windows NT R4000: Provides support for the MIPS R4000 systems.

Windows NT x86: Provides support for Intel-based systems.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Each platform subkey contains:

Drivers: Contains the driver information for each installed print driver. Each installed printer has a separate subkey named for the printer. Printer driver configuration subkeys are contained in a subkey named Version-0 for Windows NT 3.1, Version-1 for Windows NT 3.5, Version-2 for Windows NT 4, or Version-3 for Windows 2000. Each printer driver subkey contains the following value entries:

Configuration file: The .dll file that holds the printer configuration.

Data file: The .ppd or .dll file containing printer data.

Datatype: The data type, such as RAW; most printers leave this field blank.

Dependent files: Any files that the printer driver is dependent on.

Driver: The name of the printer driver .dll file.

Help File: The name of the printer driver's help file.

Monitor: If there is a print monitor, the print processor will direct its output to the print monitor.

Version: Holds the printer's driver version number. It is a value of 0 for Windows NT 3.1, 1 for Windows NT 3.5, 2 for Windows NT 4, or 3 for Windows 2000.


Print Processors: There are one or more print processors with Windows 2000. The default processor is WinPrint (winprint.dll).

Forms By default, forms used when printing documents are defined in the Forms subkey. However, a user may also specify a custom form. In most installations of Windows 2000, the following forms will exist:

- A2 420.0 x 594.0 mm
- B3 364.0 x 515.0 mm
- Foolscap 13.50 x 17.00in
- #10 Env. 9.50 x 4.12 in
- DL Env. 220.0 x 110.0 mm
- Fanfold 9.50 x 11.00 in
- Fanfold 12.00 x 8.50 in
- Fanfold 14.50 x 11.00 in
- Letter+ 9.00 x 13.30 in
- A4+ 223.5 x 355.6 mm

Create custom forms using the Printers applet in the Control Panel. To do this, follow these steps:



1. Select File  Server Properties. The Forms tab includes a section to create a new form.
2. Click Create a New Form.
3. Change the name in Form Description to a name that describes the new form.
4. Change the sizes and margins to match your form.
5. Click Save Form.



NOTE The Metric/English Units control is for display—you may display forms in either metric or English units.

Monitors No, not that big thing on your desk. A monitor for printing is a program that receives messages from printers and displays information about print jobs to the user. The printer may be either locally connected to a computer or connected directly to the network with a network card installed inside the printer. When working with a network-connected printer, the management of the printer is a bit more difficult. In this case, a printer monitor program receives messages from the printer and then process these messages for the user.

Each type of networked printer is different, and there are different monitors designed to work with each printer. Monitors exist for Hewlett Packard, Lexmark, Digital, and other network printers. There are also monitors for locally connected printers that are more generic in nature—they work with any printer connected to the printer port. Default monitors include PJI (Printer Job Language) and Local Port.

PendingUpgrades Any pending upgrades are located in the PendingUpgrades subkey.

Printers Each printer installed has a subkey under the Printers key. The printer's key name is the same as the printer's description; by default, this is also the system name for the printer.

A printer's subkey is made up of one subkey and the following value entries:

Attributes: The printer attributes—for example, those set in the Scheduling tab of the printer's property page.

ChangeID: A funny number that is not documented anywhere. Every time you change something on the printer, ChangeID changes, too. It is used to track changes.

Datatype: The type of data passed to the printer, such as RAW.

Default DevMode: The printer's default DevMode structure.

Default Priority: The default priority.

Description: A printer description provided by the user.

DnsTimeout: The amount of time to wait before a DNS timeout, in milliseconds; the default is 15 seconds.

Location: User-supplied text describing the printer's location.

Name: The name of the printer.

Parameters: Any printer parameters.

Port: The port the printer connects to.

Print Processor: The print processor; WinPrint is the default.

Printer Driver: The .dll file used to drive the printer.

Priority: The printer's priority.

Security: Security attributes.

Separator File: The name of the job separator file.

Share Name: The name used to share the printer if shared.

SpoolDirectory: The directory used to spool, if not the default spool directory.

StartTime: Sets the earliest time the printer is available (see UntilTime). If StartTime = 0 and UntilTime = 0, the printer is always available.

Status: The current printer status.

TotalBytes: The total number of bytes in the print queue.

TotalJobs: The total number of jobs in the print queue.

TotalPages: The total pages to be printed.

txTimeout: The amount of time to wait before the printer times out, in milliseconds; the default is 45 seconds.

UntilTime: Sets the latest time the printer is available (see StartTime); this is set in the Scheduling tab of the printer's property page.

The only subkey, PrinterDriverData, has information about the printer's paper sources, permissions, and more. Information in PrinterDriverData is specific to each driver.

Providers The default provider for printing in Windows 2000 networking is LanMan Print Services. Another provider is Microsoft Windows Network. However, specifying the Microsoft Windows Network provider may cause problems in some installations.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

There are at least three value entries in the Providers key:

EventLog: A DWORD value that specifies the event log status.

NetPopup: A DWORD value that specifies the NetPopup service status. It is set to 1 to display a pop-up message for remote print jobs.

Order: A REG_MULTI_SZ multiple string that specifies the order of providers. Generally, Windows 2000 networks should list LanMan Print Services first in the Order value. If you find that you are unable to browse printers, check the Order value and ensure that it contains only LanMan Print Services and not Microsoft Windows Network.

Under the Providers key, you'll also find a subkey for each provider; I'll use the LanMan Print Services as an example. The following entries may be present for each provider:

DisplayName: A string that contains the name of the provider. For our example, it is LanMan Print Services.

Name: A string that contains the name of the driver .dll. For our example, it is win32spl.dll.

LoadTrustedDrivers: An optional data value that is a DWORD value. If it is set to 1, drivers will not be installed from a remote print server, but may only be taken from the path specified in TrustedDriverPath.

TrustedDriverPath: An optional data value that is a string containing the path to load trusted print drivers. Both this data value and LoadTrustedDrivers must be set to 1 to restrict the loading of drivers.

The provider key may also contain a subkey named Servers. This subkey may contain an entry called AddPrinterDrivers, a DWORD value that specifies who is allowed to add printer drivers using the Printer applet in the Control Panel. When set to 1, only administrators and print operators (if on a server) or power users (if on a workstation) may add printer drivers.

PriorityControl

The PriorityControl key contains a single value entry: Win32PrioritySeparation. It is a DWORD value containing a value between 0 and 32. (Note that Windows NT 4 uses values between 0 and 2.) The Advanced tab of the System applet in the Control Panel includes a section called Performance. Click the Performance Options button to display the Performance Options dialog box, where you'll find a set of buttons that allows you to optimize performance for applications or background services. When set to Applications, Win32PrioritySeparation is 18; otherwise it is 26.



WARNING Microsoft cautions that the only way to successfully set the value Win32PrioritySeparation is to use the System applet in the Control Panel. Do not attempt to change this value manually.

ProductOptions

ProductOptions has a single value entry that describes the type of product installed; Windows 2000 has several different versions.

ProductType: Contains a string with one of the following values:

LANMANNT: A Windows NT Advanced Server (3.1), a Windows NT 4 PDC or BDC configuration, or a Windows 2000 Server is running with Active Directory mode.

SERVERNT: Windows NT Server 3.5 or later is running in stand-alone (not a domain) mode.

WINNT: Windows NT Workstation or Windows 2000 Professional is running.

SecurePipeServers

Pipes—long, hollow objects used to transport fluid materials. Or a virtual connection between two computers using a network. A secure pipe is a virtual pipe that has encryption and other security features to enhance the security of data being moved in the pipe.

In most Windows 2000 systems, there's an object entry in the SecurePipeServers subkey: winreg, for the remote Windows registry editing facility. The winreg subkey contains one value entry and one subkey. The value entry, named Description, has the value Registry Server.

The winreg\AllowedPaths subkey contains a single value entry, Machine, a REG_MULTI_SZ string containing the registry keys that may be edited remotely using one of the registry editors. Modify this string to add or remove keys that you wish to edit remotely. The default values in the Machine entry are:

- System\CurrentControlSet\Control\ProductOptions
- System\CurrentControlSet\Control\Print\Printers
- System\CurrentControlSet\Services\Eventlog
- Software\Microsoft\Windows NT\CurrentVersion
- System\CurrentControlSet\Services\Replicator

SecurityProviders

Security and privacy are important buzzwords in cyberspace today. Governments are working hard at limiting privacy and, essentially, security. It is also the keen intent of users to keep what is private to them private from the prying eyes of their governments. This all makes security a hot, hot topic.

Windows 2000 includes support for security in the SecurityProviders subkey. SecurityProviders contains five subkeys, discussed next.



NOTE The abbreviation CA in the entries in SecurityProviders doesn't stand for California, it stands for Certificate Authority.

CertificationAuthorities The CertificationAuthorities subkey contains the names of a number of different organizations that issue certificates and their products. A typical installation might contain the following values:

- AT&T Certificate Services
- AT&T Directory Services
- AT&T Prototype Research CA
- GTE Cybertrust ROOT
- internetMCI Mall
- Keywitness Canada Inc.
- Thawte Premium Server CA
- Thawte Server CA
- Verisign Class 1 Public Primary CA
- Verisign Class 2 Public Primary CA
- Verisign Class 3 Public Primary CA

- Verisign Class 4 Public Primary CA
- Verisign/RSA Commercial
- Verisign/RSA Secure Server

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

Ciphers A *cipher* is a code or key used to encrypt or encode an object. Generally, the term *ciphers* includes the methodology in addition to the actual key. The Ciphers subkey contains information relating to a number of cipher technologies. Some of these technologies are more secure than others, although all are satisfactory for most routine work. Ciphers supported in Windows 2000 include the following:

- DES 40/56
- DES 56/56
- NULL
- RC2 128/128
- RC2 40/128
- RC4 128/128
- RC4 40/128
- RC4 64/128
- Skipjack
- Triple DES 168/168

Hashes A *hash* is a form of cipher. Typically thought of as weak encryption, hashes can serve well when small amounts of data are being transmitted; some hash algorithms are quite secure. Windows 2000 includes the ability to support the following hashes:

- MD5
- SHA

KeyExchangeAlgorithms *Key exchange* is the process in which users are able to pass keys among themselves. An encryption algorithm, called a *public-key algorithm*, is used to send the key using plain text. This is possible because the key used to encrypt the message is not the same key used to decrypt it. The encryption key, called the *public key*, is given to everyone who is to send you encrypted messages. You keep the secure decryption key to read your encrypted messages.



WARNING Improperly designed public-key encryption schemes have a great potential for back-door type flaws. A *back door* is a way to decrypt a message without actually having the decryption (or private) key. Many governments argue that they should have the ability to decrypt messages to promote law and order. However, that policy has yet to be shown as valid.

Key exchange algorithms supported by Windows 2000 include the following:



Desktop Library
Click to access!

- Diffie-Hellman
- Fortezza
- PKCS

Protocols *Protocols* are the methodologies used to transmit information. Five security protocols are supported in Windows 2000. The most common protocol that computer users are aware of is SSL (Secure Sockets Layer), which is used to transmit information over TCP/IP networks such as the Internet.

Secure protocols that Windows 2000 supports include:

- Protocols\Multi-Protocol Unified Hello
- Protocols\PCT 1.0
- Protocols\SSL 2.0
- Protocols\SSL 3.0
- Protocols\TLS 1.0

ServiceGroupOrder

The ServiceGroupOrder subkey has a single value entry named List. The List entry includes a REG_MULTI_SZ string containing the names, in load order, for the services.

When Windows 2000 starts the services, it will start them in the order given in ServiceGroupOrder/List. Services within each group then start in accordance with the values contained in the CurrentControlSet\Control\GroupOrderList key.

Drivers are loaded into memory in the order specified in ServiceGroupOrder/List; the default for most servers is the following:

- System Bus Extender
- SCSI miniport
- port
- Primary disk
- SCSI class
- SCSI CDROM class
- filter
- boot file system
- Base
- Pointer Port
- Keyboard Port
- Pointer Class
- Keyboard Class
- Video Init
- Video
- Video Save
- file system
- Event log
- Streams Drivers
- PNP_TDI
- NDIS
- NDISWAN
- TDI
- NetBIOSGroup
- SpoolerGroup
- NetDDEGroup
- Parallel arbitrator

- extended base
- RemoteValidation
- PCI Configuration

Notice that service groups may be different in different computers. Don't expect your system to have entries in the same order, or even to always have the same entries.



NOTE Generally, it would not be prudent to change the load order for services. Some services expect that other services be already loaded.

ServiceProvider

The ServiceProvider subkey works with the Winsock RNR (Resolution and Registration) Service APIs. ServiceProvider contains the subkeys Order and ServiceTypes. The entries in this subkey are pointers to other registry keys and entries.



NOTE Microsoft recommends that you do not manually change these entries.

The subkey ServiceProvider\Order contains two value entries:

ExcludedProviders: A REG_MULTI_SZ string consisting of numbers indicating service providers. Most Windows 2000 systems do not have any entries in this list. To add an excluded provider, enter the service provider's identifier from Table 19.1.

TABLE 19.1: SERVICE PROVIDERS AND THEIR IDENTIFIERS

Service Provider	Identifier
NS_SAP	1
NS_NDS	2
NS_TCPIP_LOCAL	10
NS_TCPIP_HOSTS	11
NS_DNS	12
NS_NETBT	13
NS_WINS	14
NS_NBP	20
NS_MS	30
NS_STDA	31
NS_CAIRO	32
NS_X500	40
NS_NIS	41

ProviderOrder: A REG_MULTI_SZ value entry containing zero, one, or more values. The number of values varies with the number of installed protocols. Systems might have Tcpip, NwlnkIpx, or other values in this entry. These values correspond to CurrentControlSet\Services values.

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 [EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

A second subkey under ServiceProvider is ServiceTypes. IIS uses this subkey. ServiceTypes typically contains four subkeys (there may be fewer or more, depending on what IIS components you have installed):

GOPHERSVC: Contains the Gopher service configuration. Information includes the GUID for the handler for Gopher requests and the TCP/IP port number (70 by default).

Microsoft Internet Information Server: Microsoft's IIS is capable of serving remotely as a service. That is, you can remotely administer IIS.

MSFTPCVC: The FTP service configuration is in this subkey. The information contained here includes the GUID for the handler for FTP requests and the TCP/IP port number (21 by default).

W3SVC: The Web (WWW) service configuration is in this subkey. The information contained here includes the GUID for the handler for Web requests and the TCP/IP port number (80 by default).

Users may modify the TCP ports for these services; see the Windows 2000 Server Resource Kit for more information. When modifying ports, use a port number greater than 1023 to avoid conflict with any existing assigned ports.

Session Manager

Session Manager is a complex subkey used to manage the user's session and basic Windows 2000 start-up. Session Manager contains a number of value entries and subkeys.

The value entries in Session Manager are relatively constant between different installations of Windows 2000:

BootExecute: Specifies programs started when the system boots. The default is autocheck autochk * and dfsInit. Autochk is the auto-check utility that is included with Windows 2000. DfsInit is the distributed file system initializer.

CriticalSectionTimeout: Specifies the time, in seconds, to wait for critical sections to time out. Since Windows 2000 (retail product) does not wait for critical sections to time out, the default value is about 30 days. Anyone care to wait that long? Not me!

EnableMCA: MCA (Machine Check Architecture) is used in some systems; some Pentium Pro processors support MCA. The default value is enabled (1).

EnableMCE: MCE (Machine Check Exception) is supported by some Pentium processors. By default, support for MCE is disabled.

ExcludeFromKnownDlls: Windows NT will use entries in the KnownDLLs key to search for .dlls when loading them. ExcludeFromKnownDlls is used to exclude a .dll from the KnownDLLs search.

GlobalFlag: Controls various Windows NT internal operations using a bitmapped flag. Table 19.2



shows some common GlobalFlag bit values. GFLAG.EXE (see Figure 19.2) is a useful tool to set GlobalFlag. It is a component of the Windows NT Server Resource Kit Supplement 2. There are indications that Windows 2000 has no support for GlobalFlag.

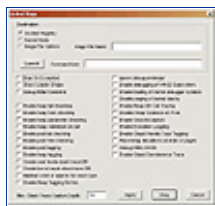


FIGURE 19.2 The GFLAGS program makes it easy to set the GlobalFlag value. Just click and select a value to be set.

TABLE 19.2: GLOBALFLAG BIT VALUES

Value (Bit)	Description
0x00000001	Stop when there is an exception.
0x00000002	Show loader snaps.
0x00000004	Debug initial command.
0x00000008	Stop on hung GUI.
0x00000010	Enable heap tail check.
0x00000020	Enable heap free check.
0x00000040	Check heap validate parameters.
0x00000080	Validate all heap allocations.
0x00000100	Enable pool tail check.
0x00000200	Enable pool free check.
0x00000400	Set up memory tagging.
0x00000800	Enable heap tagging.
0x00001000	Create user mode stack trace DB.
0x00002000	Create kernel mode stack trace DB.
0x00004000	Maintain a list of objects for each type.
0x00008000	Enable heap tags by DLL.
0x00010000	Ignore debug privilege.
0x00020000	Enable csrdebug.
0x00040000	Enable kernel debug symbol loading.
0x00080000	Disable page kernel stacks.
0x00100000	Enable heap call tracing.
0x00200000	Enable heap coalescing.

HeapDeCommitFreeBlockThreshold: Has a default of zero.

HeapDeCommitTotalFreeThreshold: Has a default of zero.

HeapSegmentCommit: Has a default of zero.

HeapSegmentReserve: Has a default of zero.

LicensedProcessors: Specifies the maximum number of processors that are allowed. The standard retail version of Windows 2000 allows a maximum of four processors in a multiprocessor server environment and two processors in a workstation environment.

ObjectDirectories: Contains a list of object directories to create during start-up.

ProcessorControl: An undocumented DWORD variable. The default value is 0x2.

ProtectionMode: When this value is set to 1, security is increased on shared base objects. The default of 0 reflects a weaker security level.



NOTE For more information on making Windows 2000 secure, see http://spider.osfl.disa.mil/dii/aog_twg/twg/ntsig/ntdoc_diicoe_dec5_DISA.htm.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Search Tips](#)[Advanced Search](#)

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

RegisteredProcessors: Specifies the number of processors allowed. The standard retail version of Windows 2000 allows a maximum of four processors in a multiprocessor server environment and two processors in a workstation environment.

ResourceTimeoutCount: Specifies the number of four-second ticks allowed before a resource will time out. Windows 2000 does not normally time out on resources, and the default value is 30 days.

There are also a number of subkeys in Session Manager, discussed next.

AppCompatibility The AppCompatibility subkey contains information about the compatibility of a number of applications.

AppPatches The AppPatches subkey contains patches for a number of applications. Typical installations include patches for the following:

- CWD
- Myst
- PalEd40
- USA
- VB
- VB 4

CheckBadApps and CheckBadApps400 These two subkeys are interrelated:

CheckBadApps: Contains applications that may be incompatible with earlier versions of Windows NT, such as Windows NT 3.51. There are only a few applications listed in this section.

CheckBadApps400: Contains applications that may be incompatible with Windows NT 4 and Windows 2000. This section has more applications listed than CheckBadApps.

When executing a listed application in one of these two subkeys, Windows displays a message for the user. This message tells the user about the possible problems encountered. The system does not prevent the user from running the application after displaying the warning message.

DOS Devices The DOS Devices subkey contains symbolic names and their corresponding logical names. Most systems have the following default entries in DOS Devices:

- advapi32 = advapi32.dll
- comdlg32 = comdlg32.dll
- crt.dll = crt.dll.dll
- DllDirectory = %SystemRoot%\System32



Desktop Library
[Click to access!](#)

- gdi32 = gdi32.dll
- kernel32 = kernel32.dll
- lz32 = lz32.dll
- olecli32 = olecli32.dll
- olesvr32 = olesvr32.dll
- rpctr4 = rpctr4.dll
- shell32 = shell32.dll
- user32 = user32.dll
- version = version.dll

Environment The Environment subkey holds the Windows 2000 system environment variables.

Executive The Executive subkey holds the Windows 2000 system executive configuration.

FileRenameOperations The FileRenameOperations subkey holds the list of files that must be renamed, but cannot be renamed at the time.

KnownDLLs The KnownDLLs subkey holds the .dll files that Windows 2000 knows about and searches first during a system start-up. This improves the search time in finding a specified .dll file object.

Memory Management Memory Management controls the system's virtual memory, paging files, and so on. To define the paging file parameters, use the System applet in the Control Panel and choose the Advanced tab's Performance Options. Other entries include the following:

ClearPageFileAtShutdown: If set to a value of 1, the paging file's contents (not the file, just its contents) will be cleared at shutdown. This option is useful for Windows 2000 installations that require a high degree of security. The default of 0 causes the paging file's contents to be left on the disk.

DisablePagingExecutive: Setting this value to 1 disables Windows 2000's automatic paging system. The default of 0 allows the paging executive to run normally. Do not change this option unless you understand exactly what the effects of disabling paging are.

IoPageLockLimit: Specifies the number of lockable bytes available for an I/O operation. The default is 0, which is equal to 512K.

LargeSystemCache: Specifies that the system will favor the system cache working set rather than the processes working set. Server installations typically set this to 1, while workstations will set it to 0.

NonPagedPoolQuota: The maximum space allocated by one process in a non-paged pool.

NonPagedPoolSize: The non-paged pool size. The default value of -0 indicates a default size based on the system's physical memory size. The maximum value allowed is 80 percent of the physical memory size.

PagedPoolQuota: The maximum space allocated by one process in a paged pool.

PagedPoolSize: The paged pool size. The default value of 0 specifies that the value will be 32MB. This value affects the maximum registry size.

PagingFiles: The name, initial size, and maximum size for the system paging file(s). Set this information using the Change button in the Performance tab of the System applet in the Control Panel.

SecondLevelDataCache: Specifies the size of the second-level data cache.

SystemPages: Specifies the number of page table entries. The default value of 0 denotes that the default number of entries is to be used.

Power The power management policies for both battery and non-battery operation are defined in the Power subkey.

SubSystems The SubSystems subkey contains subsystem settings established at start-up time. Most systems have the following entries:

Debug: The debug path, if used; most installations do not have Debug set.

Kmode: The path to the Win32 Driver; the default is win32k.sys.

Optional: Defines optional components that are only loaded when the user runs an application that requires them. Typical values include Os2 and Posix.

Os2: The path and filename of the optional Windows NT OS/2 1.x emulator.

Posix: The path and filename of the optional POSIX subsystem. This is the only POSIX entry in the registry.

Required: The default entry, Debug Windows, is required.

Windows: The path and name of the executable used to start the Win32 subsystem. The default value is:

```
%SystemRoot%\System32\csrss.exe ObjectDirectory=\Windows SharedSection=
1024,3072 Windows=On SubSystemType=Windows ServerDll=basesrv,1
ServerDll= winsrv:UserServerDllInitialization,3
ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off
MaxRequestThreads=16
```

SessionManager

Found only in Windows NT 4, SessionManager is different from Session Manager with a space, discussed earlier in this chapter. SessionManager contains lists of applications that may not run correctly with Windows NT 4.

Setup

The Setup key contains three value entries:

keyboard: The default value, STANDARD (indicating a standard keyboard), is found in virtually all systems.

pointer: The value mserr indicates that the standard Microsoft serial mouse is the default choice for setup. The value msp2 indicates that a Microsoft-compatible PS/2 mouse is connected.

video: The default value, VGA, indicates that VGA is the default video choice for setup.

StillImage

The StillImage subkey holds information about the still-image monitoring process.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

SystemResources

The SystemResources subkey holds information about various system resources, generally related to the computer's bus architecture. SystemResources contains three subkeys, discussed next.

AssignmentOrdering The AssignmentOrdering subkey contains entries for each possible bus type. Each entry specifies either an entry (the default is PCFlat) or a REG_RESOURCE_REQUIREMENTS_LIST object. Value entries in AssignmentOrdering include the following:

Eisa: Contains the string PCFlat.

Isa: Contains the string PCFlat.

MCA: Contains the string PCFlat.

PCFlat: Contains a REG_RESOURCE_REQUIREMENTS_LIST object.

PCI: Contains a REG_RESOURCE_REQUIREMENTS_LIST object.

PCMCIA: Contains the string PCFlat.

BusValues Value entries in BusValues order each bus structure using a REG_BINARY object. This object also contains a second field, the use for which is unknown. Table 19.3 shows each entry and the two values stored in each.

TABLE 19.3: BUSVALUE ENTRIES

Bus Type	Order Number	Unknown Number
CBus	9	0
Eisa	2	1
Internal	0	0
Isa	1	0
MCA	3	1
MPI	10	0
MPSA	11	0
NuBus	7	0
PCI	5	1
PCMCIA	8	1
TurboChannel	4	0
VME	6	0



NOTE Did you realize that there were that many different buses available for microcomputers? Actually, we do not use many of these buses anymore, or they are rather uncommon.

ReservedResources The ReservedResources subkey contains two entries: Isa and Eisa. The Isa entry contains a REG_RESOURCE_LIST object that lists the ISA bus's reserved resource as being bus number 0. The Eisa entry contains a REG_SZ object with an empty string.

Terminal Server

The Windows 2000 Terminal Server enhances accessibility to Windows 2000. There are about 10 entries in the Terminal Server subkey:

- DeleteTempDirsOnExit:** Holds a REG_DWORD value, with a default of 1.
- FirstCountMsgQPeeksSleepBadApp:** Holds a REG_DWORD value, with a default of 15.
- IdleWinStationPoolCount:** Holds a REG_DWORD value, with a default of 0.
- Modems With Bad DSR:** Holds a REG_MULTI_STRING value.
- MsgQBadAppSleepTimeInMillisec:** Holds a REG_DWORD value, with a default of 1.
- NthCountMsgQPeeksSleepBadApp:** Holds a REG_DWORD value, with a default of 5.
- PerSessionTempDir:** Holds a REG_DWORD value, with a default of 1.
- ProductVersion:** Holds a REG_SZ string, with a typical value of 5.0.
- TSAppCompat:** Holds a REG_DWORD value, with a default of 0.
- TSEnabled:** Holds a REG_DWORD value, with a default of 0.

One of the subkeys found in Terminal Server is AddIns, which contains the keys Clip Redirector and Terminal Server Redirector. In Clip Redirector, you'll find these values entries:

- Name:** Holds a REG_SZ string, with a typical value of RDPCLIP.
- Type:** Holds a REG_DWORD value, with a default of 3.

In the subkey Terminal Server Redirector, you'll find these entries:

- Name:** Holds a REG_SZ string, with a typical value of \\Device\\RdpDr.
- Type:** Holds a REG_DWORD value, with a default of 0.

The Terminal Server subkey AuthorizedApplications contains the following items:

- <default>:** This default value entry holds a value of a blank string.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

The Terminal Server subkey DefaultUserConfiguration contains the following value entries:

CallbackNumber: Holds a REG_SZ string, with a typical value of an empty string.

Callback: Holds a REG_DWORD value, with a default of 0.

Domain: Holds a REG_SZ string, with a typical value of an empty string.

fInheritAutoLogon: Holds a REG_DWORD value, with a default of 1.

fInheritCallbackNumber: Holds a REG_DWORD value, with a default of 0.

fInheritCallback: Holds a REG_DWORD value, with a default of 0.

fInheritInitialProgram: Holds a REG_DWORD value, with a default of 1.

fInheritMaxDisconnectionTime: Holds a REG_DWORD value, with a default of 0.

fInheritMaxIdleTime: Holds a REG_DWORD value, with a default of 0.

fInheritMaxSessionTime: Holds a REG_DWORD value, with a default of 0.

fInheritReconnectSame: Holds a REG_DWORD value, with a default of 0.

fInheritResetBroken: Holds a REG_DWORD value, with a default of 0.

fInheritShadow: Holds a REG_DWORD value, with a default of 0.

fLogonDisabled: Holds a REG_DWORD value, with a default of 0.

fPromptForPassword: Holds a REG_DWORD value, with a default of 0.

fReconnectSame: Holds a REG_DWORD value, with a default of 0.

fResetBroken: Holds a REG_DWORD value, with a default of 0.

InitialProgram: Holds a REG_SZ string, with a typical value of an empty string.

KbdIdleBusymsAllowed: Holds a REG_DWORD value, with a default of 60.

KbdIdleDetectAbsolute: Holds a REG_DWORD value, with a default of 1.

KbdIdleDetectProbationCount: Holds a REG_DWORD value, with a default of 80.

KbdIdleInProbationCount: Holds a REG_DWORD value, with a default of 35.

KbdIdlemsAllowed: Holds a REG_DWORD value, with a default of 0.

KbdIdlemsGoodProbationEnd: Holds a REG_DWORD value, with a default of 2500.

KbdIdlemsProbationTrial: Holds a REG_DWORD value, with a default of 2500.

KbdIdlemsSleep: Holds a REG_DWORD value, with a default of 100.

KeyboardLayout: Holds a REG_DWORD value, with a default of 0.

MaxConnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxDisconnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxIdleTime: Holds a REG_DWORD value, with a default of 0.

NWLogonServer: Holds a REG_SZ string, with a typical value of an empty string.

Password: Holds a REG_SZ string, with a typical value of an empty string.

Shadow: Holds a REG_DWORD value, with a default of 1.

The Terminal Server subkey Dos contains these value entries:

UserName: Holds a REG_SZ string, with a typical value of an empty string.

WorkDirectory: Holds a REG_SZ string, with a typical value of an empty string.

The Terminal Server subkey KeyboardType Mapping contains two subkeys. The first, JPN, is for Japanese keyboard mapping:

000000000017: Holds a REG_SZ string, with a typical value of kbdlk41a.dll.

00000000: Holds a REG_SZ string, with a typical value of kbd101.dll.

00000001: Holds a REG_SZ string, with a typical value of kbdax2.dll.

000000020015: Holds a REG_SZ string, with a typical value of kbdnecAT.dll.

000000020017: Holds a REG_SZ string, with a typical value of kbdlk41j.dll.

00000002: Holds a REG_SZ string, with a typical value of kbd106.dll.

00000003: Holds a REG_SZ string, with a typical value of kbdibm02.dll.

00000D01: Holds a REG_SZ string, with a typical value of kbdnecNT.dll.

00000D04: Holds a REG_SZ string, with a typical value of kbdnecNT.dll.

00010002: Holds a REG_SZ string, with a typical value of kbd106n.dll.

00010D01: Holds a REG_SZ string, with a typical value of kbdnec95.dll.

00010D04: Holds a REG_SZ string, with a typical value of kbdnec95.dll.

00020002: Holds a REG_SZ string, with a typical value of f3ahvoas.dll.

00020D01: Holds a REG_SZ string, with a typical value of kbdnecAT.dll.

00020D04: Holds a REG_SZ string, with a typical value of kbdnecAT.dll.

The other subkey under KeyboardType Mapping is KOR, for Korean keyboard mapping:

00000003: Holds a REG_SZ string, with a typical value of kbd101a.dll.

00000004: Holds a REG_SZ string, with a typical value of kbd101b.dll.

00000005: Holds a REG_SZ string, with a typical value of kbd101c.dll.

00000006: Holds a REG_SZ string, with a typical value of kbd103.dll.

The Terminal Server subkey Utilities has three subkeys, change, query, and reset. The reset subkey contains these value entries:

logon: Holds a REG_MULTI_STRING value.

port: Holds a REG_MULTI_STRING value.

user: Holds a REG_MULTI_STRING value.

winsta: Holds a REG_MULTI_STRING value.

The second Utilities subkey, query, contains:

appserver: Holds a REG_MULTI_STRING value.

process: Holds a REG_MULTI_STRING value.

session: Holds a REG_MULTI_STRING value.

user: Holds a REG_MULTI_STRING value.

winsta: Holds a REG_MULTI_STRING value.

The final Utilities subkey, reset, contains:

session: Holds a REG_MULTI_STRING value.

winsta: Holds a REG_MULTI_STRING value.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The Terminal Server subkey VIDEO contains a single subkey, rdpdd. In this subkey, you'll find the following value entries:

\\Device\\Video0: Holds a REG_SZ string, with a typical value of \\REGISTRY\\Machine\\System\\ControlSet001\\Services\\RDPDD\\Device0.

VgaCompatible: Holds a REG_SZ string, with a typical value of \\Device\\Video0.

The Terminal Server key Wds contains the subkey rdpwd, which contains these value entries:

BaudRate: Holds a REG_DWORD value, with a default of 57600.

ByteSize: Holds a REG_DWORD value, with a default of 8.

CfgDll: Holds a REG_SZ string, with a typical value of RDPCFGEX.dll.

ConnectType: Holds a REG_DWORD value, with a default of 1.

DeviceName: Holds a REG_SZ string, with a typical value of an empty string.

fAutoClientDrives: Holds a REG_DWORD value, with a default of 0.

fAutoClientLpts: Holds a REG_DWORD value, with a default of 1.

fDisableCam: Holds a REG_DWORD value, with a default of 1.

fDisableCcm: Holds a REG_DWORD value, with a default of 1.

fDisableCdm: Holds a REG_DWORD value, with a default of 1.

fDisableClip: Holds a REG_DWORD value, with a default of 0.

fDisableCpm: Holds a REG_DWORD value, with a default of 0.

fDisableEncryption: Holds a REG_DWORD value, with a default of 1.

fDisableLPT: Holds a REG_DWORD value, with a default of 0.

fEnableBreakDisconnect: Holds a REG_DWORD value, with a default of 0.

fEnableDsrSensitivity: Holds a REG_DWORD value, with a default of 0.

fEnabledTR: Holds a REG_DWORD value, with a default of 1.

fEnableRTS: Holds a REG_DWORD value, with a default of 1.

fFlowSoftwareRx: Holds a REG_DWORD value, with a default of 1.

fFlowSoftwareTx: Holds a REG_DWORD value, with a default of 1.

fForceClientLptDef: Holds a REG_DWORD value, with a default of 1.

fInheritAutoClient: Holds a REG_DWORD value, with a default of 1.

FlowHardwareRx: Holds a REG_DWORD value, with a default of 1.



Desktop Library

Click to access!

FlowHardwareTx: Holds a REG_DWORD value, with a default of 1.

FlowType: Holds a REG_DWORD value, with a default of 1.

The subkey rdpwd contains the subkey Tds. This subkey contains the subkey tcp:

InputBufferLength: Holds a REG_DWORD value, with a default of 2048.

InteractiveDelay: Holds a REG_DWORD value, with a default of 10.

MinEncryptionLevel: Holds a REG_DWORD value, with a default of 1.

OutBufCount: Holds a REG_DWORD value, with a default of 6.

OutBufDelay: Holds a REG_DWORD value, with a default of 100.

OutBufLength: Holds a REG_DWORD value, with a default of 530.

Parity: Holds a REG_DWORD value, with a default of 0.

PdClass: Holds a REG_DWORD value, with a default of 2.

PdDLL: Holds a REG_SZ string, with a typical value of tdtcp.

PdFlag: Holds a REG_DWORD value, with a default of 78.

PdName: Holds a REG_SZ string, with a typical value of tcp.

PortNumber: Holds a REG_DWORD value, with a default of 3389.

ServiceName: Holds a REG_SZ string, with a typical value of tcpip.

StartupPrograms: Holds a REG_SZ string, with a typical value of rdpcip.

StopBits: Holds a REG_DWORD value, with a default of 0.

WdDLL: Holds a REG_SZ string, with a typical value of rdpwd.

WdFlag: Holds a REG_DWORD value, with a default of 54.

WdName: Holds a REG_SZ string, with a typical value of Microsoft RDP 5.0.

WdPrefix: Holds a REG_SZ string, with a typical value of RDP.

WsxDLL: Holds a REG_SZ string, with a typical value of rdpwsx.

XoffChar: Holds a REG_DWORD value, with a default of 19.

XonChar: Holds a REG_DWORD value, with a default of 17.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

The Terminal Server key contains the subkey WinStations. WinStations contains a number of data values and nested keys. The data values contained in WinStations include:

Anonymous: A REG_BINARY value.

AppServer: A REG_BINARY value.

CallbackNumber: Holds a REG_SZ string, with a typical value of an empty string.

Callback: Holds a REG_DWORD value, with a default of 0.

Comment: Holds a REG_SZ string, with a typical value of System Console.

DefaultSecurity: A REG_BINARY value.

Domain: Holds a REG_SZ string, with a typical value of an empty string.

fEnableWinStation: Holds a REG_DWORD value, with a default of 1.

fInheritAutoLogon: Holds a REG_DWORD value, with a default of 0.

fInheritCallbackNumber: Holds a REG_DWORD value, with a default of 0.

fInheritCallback: Holds a REG_DWORD value, with a default of 0.

fInheritInitialProgram: Holds a REG_DWORD value, with a default of 0.

fInheritMaxDisconnectionTime: Holds a REG_DWORD value, with a default of 0.

fInheritMaxIdleTime: Holds a REG_DWORD value, with a default of 0.

fInheritMaxSessionTime: Holds a REG_DWORD value, with a default of 0.

fInheritReconnectSame: Holds a REG_DWORD value, with a default of 0.

fInheritResetBroken: Holds a REG_DWORD value, with a default of 0.

fInheritShadow: Holds a REG_DWORD value, with a default of 0.

fLogonDisabled: Holds a REG_DWORD value, with a default of 0.

fPromptForPassword: Holds a REG_DWORD value, with a default of 0.

fReconnectSame: Holds a REG_DWORD value, with a default of 0.

fResetBroken: Holds a REG_DWORD value, with a default of 0.

fUseDefaultGina: Holds a REG_DWORD value, with a default of 0.

InitialProgram: Holds a REG_SZ string, with a typical value of an empty string.

InputBufferLength: Holds a REG_DWORD value, with a default of 0.

KeyboardLayout: Holds a REG_DWORD value, with a default of 0.

KeyboardName: Holds a REG_SZ string, with a typical value of \\REGISTRY\\

JUMP TO TOPIC



Desktop Library

Click to access!

Machine\System\CurrentControlSet\Services\Kbdclass.

MaxConnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxDisconnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxIdleTime: Holds a REG_DWORD value, with a default of 0.

MouseName: Holds a REG_SZ string, with a typical value of \\REGISTRY\Machine\System\CurrentControlSet\Services\Mouclass.

OutBufCount: Holds a REG_DWORD value, with a default of 0.

OutBufDelay: Holds a REG_DWORD value, with a default of 0.

OutBufLength: Holds a REG_DWORD value, with a default of 0.

Password: Holds a REG_SZ string, with a typical value of an empty string.

PdClass: Holds a REG_DWORD value, with a default of 1.

PdDll: Holds a REG_SZ string, with a typical value of an empty string.

PdFlag: Holds a REG_DWORD value, with a default of 30.

PdName: Holds a REG_SZ string, with a typical value of console.

RemoteAdmin: A REG_BINARY value.

Shadow: Holds a REG_DWORD value, with a default of 0.

The WinStations key also contains the subkey Console:

UserName: Holds a REG_SZ string, with a typical value of an empty string.

WdDll: Holds a REG_SZ string, with a typical value of wdcon.

WdFlag: Holds a REG_DWORD value, with a default of 0.

WdName: Holds a REG_SZ string, with a typical value of Console.

WorkDirectory: Holds a REG_SZ string, with a typical value of an empty string.

Also found in WinStations is the subkey RDP-Tcp. This subkey contains these values:

CallbackNumber: Holds a REG_SZ string, with a typical value of an empty string.

Callback: Holds a REG_DWORD value, with a default of 0.

CdClass: Holds a REG_DWORD value, with a default of 0.

CdDLL: Holds a REG_SZ string, with a typical value of an empty string.

CdFlag: Holds a REG_DWORD value, with a default of 0.

CdName: Holds a REG_SZ string, with a typical value of an empty string.

CfgDll: Holds a REG_SZ string, with a typical value of RDP CFGEX.dll.

Comment: Holds a REG_SZ string, with a typical value of an empty string.

Domain: Holds a REG_SZ string, with a typical value of an empty string.

fAutoClientDrives: Holds a REG_DWORD value, with a default of 0.

fAutoClientLpts: Holds a REG_DWORD value, with a default of 1.

fDisableCam: Holds a REG_DWORD value, with a default of 1.

fDisableCcm: Holds a REG_DWORD value, with a default of 1.

fDisableCdm: Holds a REG_DWORD value, with a default of 1.

fDisableClip: Holds a REG_DWORD value, with a default of 0.

fDisableCpm: Holds a REG_DWORD value, with a default of 0.

fDisableEncryption: Holds a REG_DWORD value, with a default of 1.

fDisableExe: Holds a REG_DWORD value, with a default of 0.

fDisableLPT: Holds a REG_DWORD value, with a default of 0.

fEnableWinStation: Holds a REG_DWORD value, with a default of 1.

fForceClientLptDef: Holds a REG_DWORD value, with a default of 1.

fHomeDirectoryMapRoot: Holds a REG_DWORD value, with a default of 0.

[HOME](#)

[SUBSCRIBE](#)

[SEARCH](#)

[FAQ](#)

[SITEMAP](#)

[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

fInheritAutoClient: Holds a REG_DWORD value, with a default of 1.

fInheritAutoLogon: Holds a REG_DWORD value, with a default of 1.

fInheritCallbackNumber: Holds a REG_DWORD value, with a default of 1.

fInheritCallback: Holds a REG_DWORD value, with a default of 0.

fInheritInitialProgram: Holds a REG_DWORD value, with a default of 1.

fInheritMaxDisconnectionTime: Holds a REG_DWORD value, with a default of 1.

fInheritMaxIdleTime: Holds a REG_DWORD value, with a default of 1.

fInheritMaxSessionTime: Holds a REG_DWORD value, with a default of 1.

fInheritReconnectSame: Holds a REG_DWORD value, with a default of 1.

fInheritResetBroken: Holds a REG_DWORD value, with a default of 1.

fInheritSecurity: Holds a REG_DWORD value, with a default of 0.

fInheritShadow: Holds a REG_DWORD value, with a default of 1.

fLogonDisabled: Holds a REG_DWORD value, with a default of 0.

fPromptForPassword: Holds a REG_DWORD value, with a default of 0.

fReconnectSame: Holds a REG_DWORD value, with a default of 0.

fResetBroken: Holds a REG_DWORD value, with a default of 0.

fUseDefaultGina: Holds a REG_DWORD value, with a default of 0.

InitialProgram: Holds a REG_SZ string, with a typical value of an empty string.

InputBufferLength: Holds a REG_DWORD value, with a default of 2048.

InteractiveDelay: Holds a REG_DWORD value, with a default of 50.

KeepAliveTimeout: Holds a REG_DWORD value, with a default of 0.

KeyboardLayout: Holds a REG_DWORD value, with a default of 0.

LanAdapter: Holds a REG_DWORD value, with a default of 0.

MaxConnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxDisconnectionTime: Holds a REG_DWORD value, with a default of 0.

MaxIdleTime: Holds a REG_DWORD value, with a default of 0.

MaxInstanceCount: Holds a REG_DWORD value, with a default of -1.

MinEncryptionLevel: Holds a REG_DWORD value, with a default of 1.

NWLogonServer: Holds a REG_SZ string, with a typical value of an empty string.

OutBufCount: Holds a REG_DWORD value, with a default of 6.

OutBufDelay: Holds a REG_DWORD value, with a default of 100.

OutBufLength: Holds a REG_DWORD value, with a default of 530.

Password: Holds a REG_SZ string, with a typical value of an empty string.

PdClass: Holds a REG_DWORD value, with a default of 2.

PdDLL: Holds a REG_SZ string, with a typical value of `tdtcp`.

PdFlag: Holds a REG_DWORD value, with a default of 78.

PdName: Holds a REG_SZ string, with a typical value of `tcp`.

PortNumber: Holds a REG_DWORD value, with a default of 3389.

Shadow: Holds a REG_DWORD value, with a default of 1.

TraceClass: Holds a REG_DWORD value, with a default of 0.

TraceDebugger: Holds a REG_DWORD value, with a default of 0.

TraceEnable: Holds a REG_DWORD value, with a default of 0.

Username: Holds a REG_SZ string, with a typical value of an empty string.

WdDLL: Holds a REG_SZ string, with a typical value of `rdpwd`.

WdFlag: Holds a REG_DWORD value, with a default of 54.

WdName: Holds a REG_SZ string, with a typical value of `Microsoft RDP 5.0`.

WdPrefix: Holds a REG_SZ string, with a typical value of `RDP`.

WFProfilePath: Holds a REG_SZ string, with a typical value of an empty string.

WorkDirectory: Holds a REG_SZ string, with a typical value of an empty string.

WsxDLL: Holds a REG_SZ string, with a typical value of `rdpwsx`.

The RDP-Tcp subkey also contains the subkey `UserOverride`. This subkey contains the subkey `Control Panel`, which contains the subkey `Desktop`, which contains one value entry:

Wallpaper: Holds a REG_SZ string, with a typical value of an empty string.

TimeZoneInformation

The `TimeZoneInformation` subkey contains information used to manage time, time zones, and daylight time. Each entry is filled in from the time zone table contained in the subkey `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Time Zones`.

Value entries contained in this subkey include:

ActiveTimeBias: Specifies the number of minutes that local time is currently offset from GMT (UTC) time. This includes DST (daylight saving time). Divide this value by 60 to convert hours.

Bias: Specifies the number of minutes that local time is nominally offset from GMT (UTC) time, ignoring DST. Divide this value by 60 to convert to hours.

DaylightBias: Specifies the amount to change Bias to achieve ActiveTimeBias when DST is in effect.

DaylightName: Specifies the name of the time zone when DST is active; for example, eastern daylight time.

DaylightStart: A SYSTEMTIME structure indicating the start date for DST.

StandardBias: Specifies the amount to change Bias to achieve ActiveTimeBias when DST is not in effect. This value is typically 0.

StandardName: Specifies the name of the time zone when DST is not active; for example, eastern standard time.

StandardStart: A SYSTEMTIME structure indicating the end date for DST.

Update

Update contains information about how to update policies, which are set using the Active Directory system and MMC. Policies update the file `Config.Pol`, and this file's path is known. When a user logs on, the user's computer policies are automatically updated.

The Update subkey contains up to four value entries:

NetworkPath: Contains an empty string if UpdateMode is 1, or the network path to the location of the update files if UpdateMode is 2.

UpdateMode: Contains a DWORD value indicating the update mode. There are three values allowed in UpdateMode:

0: Do not use policies for updates.

1: Automatic policy mode is in effect after validating the user on the domain.

2: Manual policy mode is in effect. The NetworkPath variable is required when using this mode.

Verbose: Allows the system to display error messages if Verbose = 1. The default does not display error messages. The Verbose data value is not set by default.

LoadBalance: Allows the system to balance loads if LoadBalance = 1. The default does not display error messages. The LoadBalance data value is not set by default.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

VirtualDeviceDrivers

Windows 2000 does not support VDDs (virtual device drivers). The subkey VirtualDeviceDrivers contains any VDDs that are loaded in the VDM (Virtual DOS Machine) when initialized. This subkey is for IHVs (independent hardware vendors) who find it necessary to supply drivers for their hardware products.

There is a single value entry in VirtualDeviceDrivers: VDD, which contains a REG_MULTI_SZ string. This string contains the names of any VDDs used by the VDM. By default, this value is empty because there are no VDDs for Windows 2000.



NOTE Windows 2000 does not support any 16-bit virtual device drivers. Applications that rely on 16-bit virtual device drivers will fail.

Windows

The Windows subkey contains some configuration information for Windows 2000. Value entries included in this subkey are:

CSDVersion: The CSD (Microsoft's nomenclature for their service packs) status can be determined from this subkey. In earlier versions of Windows NT (other than Windows NT 3.1 Advanced Server), CSDVersion was a string. However, Windows NT 4 used a DWORD value. Windows 2000 changed back from REG_DWORD to REG_SZ.

Directory: Includes a REG_EXPAND_SZ string containing the value %SystemRoot%.

ErrorMode: May contain a value between 0 and 2:

0: The default mode that serializes errors and waits for a user response.

1: Non-system errors are considered normal and are not reported. The event log logs the system errors. The user is given no error message.

2: Errors are logged to the event log, and no error message is given to the user.

NoInteractiveServices: If set to 1, no interactive services are allowed.

ShutdownTime: Specifies the time of the last shutdown.

SystemDirectory: Includes a REG_EXPAND_SZ string containing the value %SystemRoot%\System32.

WMI



Desktop Library
Click to access!

WMI (Windows Management Instrumentation) provides kernel-level debugging instrumentation to Windows 2000. A single subkey exists in WMI. This subkey, Security, contains a REG_BINARY value with a name that is a CLSID.

WOW

WOW, or Windows on Windows, is an emulator that allows 16-bit applications to run on Windows 2000. It uses a simple emulation of the Windows 3.x standard mode. There are eight value entries in the WOW subkey:

cmdline: Contains the command line used to start the WOW system.

DefaultSeparateVDM: Specifies whether WOW is to allocate a default separate VDM (Virtual DOS Machine).

KnownDLLs: Contains a list of DLLs that the WOW VDM will load to provide compatibility for non-Win32 applications.

LPT_timeout: Specifies the time-out period for the printer port.

SharedWowTimeout: Contains the time-out value of 3600.

size: Contains the memory size allocated. A value of 0 means that the system used the default size.

Wowcmdline: Contains the command line used to start the WOW system, including any parameters.

wowsize: Specifies the amount of memory supplied to WOW applications. Released versions of Windows 2000 automate this value, and changes should not be necessary. The default value is 16.

Enum

The Enum subkey represents the beginning of the hardware tree. Through the Enum key, any subkey named Root (regardless of case) will represent enumerated devices.

Subkeys in this key include those discussed next.

DISPLAY

The DISPLAY subkey represents the display device, a.k.a the monitor, attached to the system. Generally, a generic monitor setting is all that is required.

FDC

The FDC subkey represents the floppy disk drive installed on the system. As with monitors, floppy disk drives are rather generic.

HTREE

In Windows NT 4, HTREE was a complex object. In Windows 2000, HTREE contains no data values and only a single subkey: ROOT. Within the ROOT key, again there are no data values, only a single subkey named 0. This subkey contains one data value:

ConfigFlags: Holds a REG_DWORD value of 32 (0x00000020).

IDE

New to Windows 2000, the IDE object contains information about all IDE devices. For example, on the machine I am using, there are two IDE CD-ROM drives and a single 18GB IDE hard drive. There is information in the IDE key for each of these three devices. Information is arranged with a key for each basic device and a subkey for each specific device within the basic device.

For example, my two Matsushita CD-ROM drives are attached to one of my IDE channels. Under IDE, there is a subkey named CdRomMATSHITA_CD-ROM_CR-581-M_____1.05____. Quite a name, isn't it?

Under this subkey, there are two additional subkeys, one for each drive:

4&13b4afd&0&0 . 0 . 0

4&13b4afd&0&0 . 1 . 0

Notice that the only difference between those two subkeys is a single character.

ISAPNP

New to Windows 2000, the ISAPNP object manages the ISA bus's PnP functionality. ISA systems have a single key named ReadDataPort, used to receive information from the system about PnP.

PCI

New to Windows 2000, PCI is a subkey with information on each of the PCI bus adapters. For example, one computer here has built-in PCI disk controllers, PCI-to-ISA and PCI-to-USB bridges, a PCI SCSI adapter, a PCI network interface card, and a PCI video adapter. Each PCI device has a subkey containing information about the device.

PCI_HAL

New to Windows 2000, the PCI_HAL object contains information about the actual PCI bus. A computer could have more than one PCI bus; however, most will have one, documented in this key.

PCIIDE

New to Windows 2000, the PCIIDE object contains information about all IDE controllers. Most modern computers have two IDE controllers (a primary and a secondary). Although many times these devices are listed as one being attached to the ISA bus and one to the PCI bus, oftentimes both are routed through the PCI bus (this allows for higher performance).

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#)[SUBSCRIBE](#)[SEARCH](#)[FAQ](#)[SITEMAP](#)[CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.


Datamation
Profit and Value from Information Technology
[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

 This site is brought to you by
 EarthWeb

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)
[KEYWORD SEARCH](#)

Mastering Windows 2000 Registry

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 **Pub Date:** 01/01/00

Search this book:
[Search Tips](#)
[Advanced Search](#)
[PUBLICATION LOOKUP](#)
[Previous](#) [Table of Contents](#) [Next](#)
[JUMP TO TOPIC](#)

Desktop Library
[Click to access!](#)

Root

The Root subkeys represent enumerators that Windows 2000 uses to hold information about the device(s). Each device listed in AttachedComponents receives a subkey under the Root key.

On one of my Windows 2000 systems, this object contains the following entries. These entries vary depending on the installed hardware.

- Root*PNP0000
- Root*PNP0100
- Root*PNP0200
- Root*PNP030b
- Root*PNP0400
- Root*PNP0501
- Root*PNP0700
- Root*PNP0800
- Root*PNP0B00
- Root*PNP0C01
- Root*PNP0C02
- Root*PNP0C04
- Root*PNP0F03
- Root*PNPB003
- Root*PNPB02F
- Root\dmio
- Root\ftdisk
- Root\LEGACY_CDFS
- Root\LEGACY_DMBOOT
- Root\LEGACY_DMLOAD
- Root\LEGACY_FASTFAT
- Root\LEGACY_MOUNTMGR
- Root\LEGACY_NTFS

- Root\LEGACY_PARTMGR
- Root\LEGACY_REMOTEREGISTRY
- Root\LEGACY_SNMP
- Root\LEGACY_SNMPTRAP
- Root\LEGACY_SYSMONLOG
- Root\LEGACY_VGA
- Root\MS_NDISWANBH
- Root\PCI_HAL

Notice that not all devices are really hardware. Items such as virtual drivers are included in the list.

These subkeys include information about each device. Devices that receive support from Windows NT have additional information in the form of an extra subkey. An example, the NIC (network interface card) in the computer is a 3Com 3C900 series, which is a PCI device (actually, Plug-and-Play) that Windows NT is able to support.

Under Windows 2000, the subkey for the 3C900 is Enum\Root\LEGACY_EL90X. Where'd the EL90X come from? EL is short for EtherLink (3Com's terminology for their Ethernet cards). The 90X is the designator for the 3C900 series, which contains a number of different devices with varying speeds (10Mbps and 100Mbps) and connection form factors.

The LEGACY_EL90X subkey contains the following entries:

- HKEY_CURRENT_CONFIG
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\0001

That's a mouthful, but it really says that there are three names for the same piece of information.



NOTE Changes made in the Current subkey change the corresponding subkey in the currently used configuration and vice versa.

SCSI

New to Windows 2000, SCSI is a subkey with information on each of the SCSI devices attached to the system. Some systems (many perhaps) do not have any SCSI devices. There is a subkey for each attached SCSI device.

STORAGE

New to Windows 2000, storage volumes allow Windows 2000 to very efficiently manage disk resources. Each writeable storage device (hard disk drive) will have a subkey in the STORAGE key. Within a device's key, there are two subkeys: Control and LogConf.

SW

New to Windows 2000, SW is used with support for streaming protocols. Items in this subkey could include:

- Microsoft Streaming Clock Proxy
- Microsoft Streaming File System I/O
- File System Reader
- File System Writer
- Microsoft Streaming Quality Manager Proxy
- Microsoft Streaming RIFF Wave File Parser
- RIFF Wave File Parser
- Microsoft Streaming Service Proxy
- Microsoft Streaming Tee/Sink-to-Sink Converter
- Tee/Sink-to-Sink Converter

- Microsoft Streaming Network Raw Channel Access
- Raw Channel Access Capture/Render
- WDM Streaming IOverlay Property Set Interface Handler
- RAS Async Adapter

USB

New in Windows 2000, USB (Universal Serial Bus) allows connecting, daisy-chain fashion, various devices. Though acceptance of USB has been slow, it appears that the next few years will bring a proliferation of USB devices, including keyboard, pointer, joystick, and output devices.

Hardware Profiles

In Windows 2000, the Hardware Profiles key contains information about the computer. This information is used in HKEY_CURRENT_CONFIG. In the Hardware Profiles key, there may be one or more subkeys, each named with a number; there is also a subkey named Current. For a fuller view of both this key and HKEY_CURRENT_CONFIG, see the sections on HKEY_CURRENT_CONFIG later in this chapter.

Windows maps the Current key to the currently used key, typically 0001. The HKEY_CURRENT_CONFIG hive maps to the Current key. Changes made in Current are reflected in the key it is mapped to and to HKEY_CURRENT_CONFIG. The converse is also true: changing HKEY_CURRENT_CONFIG will change both Current and the hive Current maps to.



NOTE We can assume that Microsoft has other plans for this key in the future. The naming of the key is odd, but then again, we are dealing with Microsoft on this one.

A decidedly non-hardware object is also contained in the Hardware Profiles key: Software. A single entry resides in the Software subkey: Internet Settings. The Internet Settings key contains two value entries for controlling how the system connects to the Internet:

EnableAutodial: Contains a DWORD value. If the value is 0x0, the system will not attempt to auto-dial to connect to the Internet (or other remote host). If the value is 0x1, the system will attempt to auto-dial when this is necessary to connect to the remote network.

ProxyEnable: Contains a DWORD value. If the value is 0x0, the system *will not* use a proxy server to connect to the Internet. If the value is 0x1, the system *will* use a proxy server to connect to the Internet.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)
Services

The Services subkey contains information about the Windows 2000 services. The Services applet in Administrative Tools manages the Services subkey.

A service is any Windows service, such as a device driver, the file system drivers, and so on. Services can be started in several ways:

Automatic: The service starts automatically when Windows starts.

Manual: The service starts manually and not when Windows starts.

Disabled: The service is disabled. The service cannot be started.

The Services subkey also contains devices listed in the Devices applet of the Control Panel. Similar to a service, a device may have a number of different start-up states:

Boot: The device starts when the system boots, before any other devices.

System: The device starts when the system boots, after the boot devices.

Automatic: The device starts when the system boots, after the boot and system devices.

Manual: The device starts manually; the system will not attempt to start the device automatically.

Disabled: A user cannot start the device.

Boot: The device starts when the system boots.

When a service runs, it often must log on as a user. Choices for a service include logging on as the system account or as a specific account. When logging on as a specific account, the service is configured with the account name and password information.



WARNING Be careful not to compromise system security by allowing a service to log on as a privileged account and interact with the Desktop.

ControlSet001

ControlSet001 is the control set used to boot the computer during normal operations. If it fails to boot for some reason, then ControlSet002 (or ControlSet003) will be used instead.



ControlSet002

Though object content may vary, ControlSet002 (or Control-Set003, if that is what your computer has) is identical in structure to ControlSet001.

ControlSet002 is the backup control set and the Last Known Good control set.

DISK

The DISK key contains information about specific types of drives, such as CD-ROM drive letter mappings. In Windows 2000, the DISK key contains information about disk configurations, such as fault-tolerant configurations (consisting of mirroring, stripe sets, stripe sets with parity, and so on). DISK contains a single value entry, named Information, which contains an undocumented REG_BINARY value.

Disk Administrator manages the information in the DISK key. In fact, this key doesn't even exist until the first time a user runs Disk Administrator. Disk Administrator also makes backups of this key. Start Disk Administrator and select Partition \emptyset Configuration Save. Disk Administrator will then write the DISK key information to a floppy disk. It will not write to any other device, such as a hard drive.



WARNING Disk Administrator writes the entire HKEY_LOCAL_MACHINE\System key to a registry export file. However, do not try reloading this file using anything other than Disk Administrator, because another program may restore keys and values that are not up to date.

Mounted Devices

The Mounted Devices key contains information about drives, such as CD-ROM drive letter mappings. In Windows 2000, the Mounted Devices key contains information about volume configurations and identifiers, as well as DOS configurations.

Select

The Select key contains information about which control set has been loaded by the system. The following four value entries reside in this key:

Current: Defines the currently used control set.

Default: Defines the currently used control set, which is typically also the current control set.

Failed: Lists a control set that has failed when the system was attempting to start.

LastKnownGood: Specifies the control set that is accessed when a user requests the Last Known Good control set from Windows 2000 at boot time.



WARNING When Windows 2000 shuts down, Windows copies the current control set to the Last Known Good control set. Be careful when attempting to boot the system that you do not inadvertently overwrite your Last Known Good copy of the control set with a copy that does not work correctly.

Setup

Setup contains information used by the system during the setup stage. This information is contained in a number of value entries:

CmdLine: Contains the command string to set up Windows 2000. Typically, this command is setup -newsetup.

NetcardDlls: Contains the names for the drivers for the NIC.

OsLoaderPath: Contains the path for the OS loader.

SetupType: Has a value of 0, 1, or 4. These values indicate the following:

0: Setup has completed.

1: Windows is doing a new full install.

4: Windows is doing an upgrade.

SystemPartition: A pointer to the system installation device. Typically, for SCSI systems, this string

will be \\Device\\Harddisk0\\Partition1.

SystemPrefix: Used to determine the system type.

SystemSetupInProgress: If the setup has not completed, the value in this entry is 0x1. Once setup has completed, it contains the value 0x0.

uniqueid: A unique directory name used during setup.

UpgradeInProgress: This value is 0 unless an upgrade is in progress.

HKEY_CURRENT_CONFIG

The HKEY_CURRENT_CONFIG hive is nothing more than an alias (or pointer) to the HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current key. HKEY_CURRENT_CONFIG stores only items changed from the standard configuration contained in HKEY_LOCAL_MACHINE\System\CurrentControlSet. The most common entries found in HKEY_CURRENT_CONFIG are the entries under Services for the video display. Windows 95 and Windows NT 4 introduced this object.

HKEY_CURRENT_CONFIG contains two subkeys, Software and System, discussed in the next two sections.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**[Click to access!](#)

HKEY_CURRENT_CONFIG\Software

The HKEY_CURRENT_CONFIG\Software key contains some settings that you may want to configure. Notice that there is no built-in methodology to edit or modify items in the Software key—each application or system must manage these entries and provide the method for the user to modify entries.

Under the Software key, Microsoft applications include a subkey named Microsoft. This subkey contains only one entry on most systems: Windows. The Windows subkey includes a subkey named CurrentVersion. Get the drift here? This structure (HKEY_CURRENT_CONFIG\Software\Microsoft\Windows\CurrentVersion) is exactly the same structure you find in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current\Software\Microsoft\Windows\CurrentVersion. Both subkeys have Microsoft\Windows\CurrentVersion. However, it is possible that other applications will have subkeys under this key, as well. Do not count on the Microsoft subkey being the only one present in HKEY_CURRENT_CONFIG\Software; there may be others some day.

The Software key also contains a subkey called Internet Settings. There are two value entries in this key:

EnableAutodial: Contains a DWORD value. If the value is 0x0, then the system will not attempt to auto-dial to connect to the Internet or any other remote host. If the value is 0x1, then the system will attempt to auto-dial if necessary to connect to the remote network.

ProxyEnable: Contains a DWORD value. If the value is 0x0, then the system will not use a proxy server to connect to the Internet. If the value is 0x1, then the system will use a proxy server to connect to the Internet.

HKEY_CURRENT_CONFIG\System

The HKEY_CURRENT_CONFIG\System key contains objects that temporarily modify the current control set. Microsoft chose to implement multiple hardware configurations this way, rather than allowing users to modify the current control set on the fly, for reliability reasons. Regardless of what happens to the system, Windows 2000 can be sure that current control set is representative for all users and configurations, and that if changes must be implemented for a specific configuration, these changes will be pointed to by HKEY_CURRENT_CONFIG.

CurrentControlSet

The System key contains a subkey called CurrentControlSet. This subkey matches the HKEY_LOCAL_MACHINE\System\CurrentControlSet key in function. Remember: only modifiers are present in

HKEY_CURRENT_CONFIG; therefore, if nothing in the HKEY_LOCAL_MACHINE\System\CurrentControlSet subkey needs modification, the HKEY_CURRENT_CONFIG\System\CurrentControlSet classes will be empty.

The HKEY_CURRENT_CONFIG\System\CurrentControlSet key contains three subkeys: Control, Enum, and Services.

Control

In HKEY_LOCAL_MACHINE\System\CurrentControlSet, the Control subkey has a number of value entries used for booting and system initialization, as well as about 30 subkeys. In HKEY_CURRENT_CONFIG, the Control subkey is typically empty, unless it is necessary to modify

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control based on a particular configuration.

In HKEY_CURRENT_CONFIG, the Control key may contain the following value entries:

RegistrySizeLimit: If the user changes the registry size limit from the default value of 8MB, RegistrySizeLimit will contain the maximum registry size, in bytes. Though users are only able to set the registry size limit in megabytes, Windows will store the value as a DWORD containing the maximum registry size in bytes.

SystemStartOptions: This value contains options used during start-up, passed from firmware or the start-up process (contained in Boot.ini). Options might include debugging information (such as a debugging port and the debugging port parameters) and perhaps information on the system root directory.

WaitToKillService: This value specifies the amount of time, in milliseconds, to wait before killing a service when Windows 2000 is shutting down. If this value is too small, Windows 2000 may kill the service before it has finished writing its data; if this value is too large, a hung service will delay shutdown. It is best to leave the WaitToKillService value at its default value of 20000, unless you know you are having a problem.

Control could also contain any of the subkeys found in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control.

Enum

The Enum subkeys represent the beginning of the hardware tree. Through the Enum key, any subkey named Root (regardless of case) will represent devices enumerated using non-PnP services.

Though HKEY_CURRENT_CONFIG\System\CurrentControlSet\Enum is typically empty, it could contain the two subkeys HTREE and Root.

HTREE The HTREE subkey represents the hardware devices. There is a subkey under HTREE called ROOT, and within ROOT, there is a subkey called 0. HTREE\ROOT\0 includes any devices that may be transient, such as a device contained within a docking station.

Similar to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Enum, notice that not all devices are really hardware. The list includes items such as virtual drivers and so on.

Root The Root subkeys represent enumerators used by Windows 2000 to hold information about the device(s). Each device listed in AttachedComponents receives a subkey under the Root key.

These subkeys contain information about each device. Devices that receive support from Windows 2000 have extra information in the form of an additional subkey. For example, the NIC (network interface card) in the computer is a 3Com 3C900, which is a PCI device (actually, Plug-and-Play) that Windows 2000 is able to support.

Again, because HKEY_CURRENT_CONFIG is used to modify HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\Root, only items that must be changed on a temporary basis are included in HKEY_CURRENT_CONFIG.

Services Services contains subkeys for each device changed from the default configuration. Every system has at least one entry in this subkey for the video card. For example, consider one computer that is running Windows 2000 Professional with an ARK chipset video adapter. This device's parameters are stored in a subkey called ark, at HKEY_CURRENT_CONFIG\System\CurrentControlSet\Services\ark.

Your computer will have a similar subkey for its video card with virtually identical entries. The name will be

different. For example, if you have a Matrox Millennium video card, the subkey's name will be mga_mil, not ark.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

For video (specifically VGA) subkeys, typical entries include:

DefaultSettings.BitsPerPel: Indicates the number of bits per pel (pixel); this will be a number between 1 (indicating a monochrome system) and 32 (true-color systems).

DefaultSettings.Xresolution: The resolution in the X plane (horizontal). Settings range from 640 to 1280 or more for very high resolution systems.

DefaultSettings.Yresolution: The resolution in the Y plane (vertical). Settings range from 480 to 1024 or more for very high resolution systems.

DefaultSettings.Vrefresh: The vertical refresh rate, which usually has a value between 20 and 100, with a typical value of about 70. This reflects the monitor's refresh rate, in Hz. If you change this value, make sure the video adapter at the specified resolution supports the value chosen. Oh, also realize that the Display Properties dialog box will probably change it back to whatever it wants.

DefaultSettings.Flags: This entry controls the specification of any device flags, as necessary.

DefaultSettings.Xpanning: If the device supports hardware panning, this specifies the default horizontal panning value.

DefaultSettings.Ypanning: If the device supports hardware panning, this is the default vertical panning value.

Hints and Kinks from the Experts

In this chapter's hints and kinks, we cover a few hints on using the two registry editors.

Disabling Autodisconnect

Windows 2000 uses two different auto-disconnect parameters—one for disconnecting Remote Access Service (RAS) connections and another for disconnecting LAN connections. The RAS autodisconnect parameter is well documented in the Windows NT Server Remote Access Service manual on page 82, but the LAN version is undocumented.

You can find the LAN autodisconnect parameter in the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters. Its purpose is to disconnect idle sessions after a set number of minutes. The number of minutes can be set at a command prompt using the Net Config Server command. For example, this is how you would set the autodisconnect value to 30 minutes:

```
Net Config Server /autodisconnect:30
```



The valid value range of this REG_DWORD value is from -1 to 65535 minutes at the command line. To disable autodisconnect, set it to -1.

Setting autodisconnect to 0 does not disable autodisconnect, and results in very fast disconnects within a few seconds of idle time. However, unlike with the LAN autodisconnect parameter, the RAS autodisconnect parameter is turned off if you set it to a value of 0.



TIP It is preferable to modify the LAN auto-disconnect directly in the registry. If you modify it at the command line, Windows 2000 may turn off its auto-tuning functions.

The valid value range, if you edit the LAN autodisconnect parameter in the registry, is from 0 to 4294967295 (0xffffffff). If you configure the autodisconnect option to -1 at the command prompt, autodisconnect is set to the upper value in the registry. This is approximately 8,171 years (not tested), which should be long enough to be the equivalent of turning auto-disconnect off.

(Courtesy of Jerold Schulman.)

Does Your CD-ROM Changer Cycle Excessively?

If your CD-ROM changer cycles excessively, try these three simple steps:

1. Set HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Cdrom\Autorun to 0.
2. In the Devices applet of the Control Panel, set Start Up for CD Audio to Manual. Click the Stop button. (This may not be required on your installation.)
3. Create a shortcut to Explorer (set to minimized) and place it in your Startup folder. Leave it minimized. It will share the CD information with all other copies of Explorer that you open and with all properly written applications.

(Courtesy of John Savill.)

How Do I Get Remote Access to the Registry?

In Windows 2000, only members of the Administrators group have access to the registry. You can alter this default by editing the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg. To change the default value, perform the following steps:

1. If the SecurePipeServers value does not exist, add it with a class of REG_SZ. If the winreg value does not exist, add it with a class of REG_SZ.
2. Add a value of Description as type REG_SZ and set the string to Registry Server.
3. Select the winreg value and choose Security Permissions from the RegEdt32 menu.
4. Grant the users and groups the access you want them to have. I would grant full control to administrators.

It is possible to bypass these access permissions. Some services, such as Directory Replicator and Spooler, require remote access to the registry. You can grant access to the account that runs these services. Or you can edit the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg by following these steps:

1. Add a value of AllowedPaths with an empty class.
2. Select HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg\All.
3. Add a value name of Machine as type REG_MULTI_SZ with the following string values, one per line:

```
System\CurrentControlSet\Control\ProductOptions
System\CurrentControlSet\Control\Print\Printers
System\CurrentControlSet\Services\Eventlog
Software\Microsoft\Windows NT\CurrentVersion
System\CurrentControlSet\Services\Replicator
```

If you wish, you can grant users access to listed locations in the registry by adding a value name of Users as type REG_MULTI_SZ and listing the registry locations, one per line.

(Courtesy of John Savill.)

I Make Changes to HKEY_LOCAL_MACHINE, but They Are Lost on Reboot

This happens because some keys in HKEY_LOCAL_MACHINE are re-created by the system at boot time. This means that any settings, such as ACLs, are lost.

(Courtesy of John Savill.)

VgaSave

What is VgaSave? And why do I have it?

VgaSave is a place where the settings for the default VGA adapter are stored when the user installs a new driver and a new VGA card. Windows 2000 copies the original settings to VgaSave and creates a new key for the new video driver.

How Do I Set RAS to Answer after More Than One Ring?

The only way I have been able to get RAS to answer after x rings is to edit the registry at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Class. Follow these steps:

1. Locate the modem's entry, which is usually the fourth entry.
2. Double-click to expand the entry.
3. Locate the modem in question, where 0000 is the first modem.
4. Double-click and highlight the Monitor value.
5. Double-click the first entry, which should be ATSO=, and change the number of rings.

You will need to stop and restart the RAS service, and a reboot may be necessary.



NOTE This technique could probably be done in HKEY_CURRENT_CONFIG by adding the necessary entries under the HKEY_CURRENT_CONFIG\System\CurrentControlSet\Control\Class subkey.

(Courtesy of Jerold Schulman.)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.


Datamation
Profit and Value from Information Technology
[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)

 This site is brought to you by
 EarthWeb

[HOME](#)
[SUBSCRIBE](#)
[SEARCH](#)
[FAQ](#)
[SITEMAP](#)
[CONTACT US](#)
[KEYWORD SEARCH](#)

Mastering Windows 2000 Registry
by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 **Pub Date:** 01/01/00

Search this book:
[Search Tips](#)
[Advanced Search](#)
[PUBLICATION LOOKUP](#)
[Previous](#) [Table of Contents](#) [Next](#)

PART V Appendices

APPENDIX A Common Hives and Keys

In virtually all registries, there are a number of common entries. These entries, mostly for basic system components, usually have either the same values or predictable values. Table A.1 lists some common registry hives and keys.

TABLE A.1: COMMON REGISTRY HIVES AND KEYS

Hive/Key	Subkey	Description
HKEY_LOCAL_MACHINE\	All keys	The main system description hive. This hive is critical to the execution of Windows 2000.



HKEY_LOCAL_MACHINE\Hardware\Description\	All keys	Contains information on installed hardware. This key is created at boot time, though some entries may be retained from previous executions.
HKEY_LOCAL_MACHINE\Hardware\Description\	System	Contains system device information, excluding NIC (network interface card) and video devices.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\CentralProcessor	Contains CPU information, such as make, model, and version.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\FloatingPointProcessor	Contains floating point processor data, such as make, model, and version.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter\2\DiskController\0\DiskPeripheral	Contains installed disk controller information. Systems may have one, two, or three controllers in a typical configuration: primary IDE, secondary IDE, and SCSI.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter\2\KeyboardController	Contains keyboard controller information at the hardware level.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter\2\ParallelController	Contains printer (parallel) port information.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter\2\PointerController	Contains mouse port information.

HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter\2\SerialController	Contains information on installed serial ports.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\MultifunctionAdapter	Contains information on device classes, other than network and disk.
HKEY_LOCAL_MACHINE\Hardware\Description\	System\PCMCIA PCCARDS	Contains information on installed PCMCIA (PC Card) devices.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	All keys	Contains basic device-mapping and control information.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	KeyboardClass	Contains keyboard device-mapping information.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	KeyboardPort	Contains keyboard port configuration information.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	PARALLEL PORTS	Contains printer (parallel) port configuration information.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	PointerClass	Contains mouse information.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	PointerPort	Contains information on the port (mouse port, PS/2 mouse port, serial port, and so on) the mouse (pointer) connects to.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	Scsi	Contains general disk interface information on IDE and SCSI devices.

HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	Scsi\Scsi Port 0	Contains information on the first disk drive interface adapter (although labeled as SCSI, this may be an IDE device).
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	Scsi\Scsi Port 1	Contains information on the second disk drive interface adapter (although labeled as SCSI, this may be an IDE device).
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	SERIALCOMM	Contains information on serial communications device configurations.
HKEY_LOCAL_MACHINE\Hardware\DeviceMap\	VIDEO	Contains video configuration information.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	All keys	Contains information on (hardware) system mapping.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	Hardware Abstraction Layer\PC Compatible Eisa\Isa HAL	Describes the system configuration to Windows 2000. HALs exist for generic systems and for computers that have special hardware configurations, such as multiple processors or special bus configurations.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	KeyboardPort\PointerPort	Contains general keyboard/mouse interface information.

HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	KeyboardPort\PointerPort\msi8042prt	Contains mouse/keyboard interface information.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	LOADED PARALLEL DRIVER RESOURCES	A description of currently loaded printer (parallel) port driver configurations.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	LOADED SERIAL DRIVER RESOURCES	A description of currently loaded serial port driver configurations.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	OtherDrivers	Contains general information on devices not otherwise classified.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	OtherDrivers\<NIC>	A description of the NIC.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	ScsiAdapter	Contains information about SCSI and IDE adapters.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	ScsiAdapter\atapi	A description of the installed IDE (ATAPI) disk interface.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	System Resources	Contains general information on system resources.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	System Resources\Reserved	Contains reserved system resources information.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	System Resources\Physical Memory	Contains system memory resources information.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	VIDEO	Contains information on video configurations supported by the system.

HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	VIDEO\chips	Contains information on the installed VGA adapter for the Chips & Technology VGA system.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	VIDEO\VgaSave	Contains information on the originally installed VGA video system, generally a generic VGA system.
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\	VIDEO\VgaStart	Contains information on the VGA driver used to start the system.
HKEY_LOCAL_MACHINE\SAM\	SAM	The SAM subkey. Usually protected from user browsing and modification. (Yes, the key is named SAM\SAM.) In a Windows 2000 domain using Active Directory, SAM is not used.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Aliases	Contains SAM alias information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Aliases\Members	Contains member alias information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Aliases\Names	Contains domain name alias information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Groups	Contains Groups information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Groups\Names	Contains group name information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Users	Contains specific user information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Users\Names	Contains user name information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Users\Names\Administrator	Contains user administrator information.

HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Account\Users\Names\Guest	Contains user guest information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx	Contains information on built-in users: Administrator and Guest.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx\00001F4	Contains information on the built-in user: Administrator.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx\00001F5	Contains built-in user information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx	Contains Domain Groups information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx\0000200	Contains Domain Admins group information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members\S-1-5-21-xxxxxxxx-xxxxxxxx-xxxxxxxx\0000201	Contains Domain Users group information.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Members	Contains member alias information for user groups.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Administrators	Contains member alias information for Administrators.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Backup Operators	Contains member alias information for Backup Operators (users who perform system backups).
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Guestsm	Contains member alias information for Domain Guests.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Power Users	Contains member alias information for Power Users.
HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Replicator	Contains member alias information for Replicator accounts.

HKEY_LOCAL_MACHINE\SAM\	SAM\Domains\Builtin\Aliases\Names\Users	Contains member alias information for Domain Users.
HKEY_LOCAL_MACHINE\SAM\	SAM\RXACT	The SAM RXACT key. Used by the registry transaction package, there are a number of RXACT keys located in the registry. Typically these keys contain nothing.
HKEY_LOCAL_MACHINE\Security\	All keys	The protected Windows 2000 security key.
HKEY_LOCAL_MACHINE\Software\	All keys	Contains information about installed user and system software
HKEY_LOCAL_MACHINE\Software\	Classes	Contains information about extensions and the usage of file types.
HKEY_LOCAL_MACHINE\Software\	Classes*	Contains information about files in general—that is, files that are not otherwise classified.
HKEY_LOCAL_MACHINE\Software\	Classes\CLSID	Contains information about CLSID (class ID) assignments. Almost all applications, and those that support OLE, have a CLSID.

HKEY_LOCAL_MACHINE\Software\	Classes\Interface	Contains information about OLE interface assignments. Almost all applications that support OLE have an OLE interface.
HKEY_LOCAL_MACHINE\Software\	Description	Contains information about RPC objects and configurations.
HKEY_LOCAL_MACHINE\Software\	Windows NT\CurrentVersion	Contains information on the currently installed version of Windows 2000.
HKEY_LOCAL_MACHINE\Software\	Program Groups	Contains information on program groups as used by Program Manager.
HKEY_LOCAL_MACHINE\Software\	Secure	Contains security information.
HKEY_LOCAL_MACHINE\Software\	Windows 3.1 Migration Status	Contains information on migration from Windows NT 3.x to Windows NT 4/2000.
HKEY_LOCAL_MACHINE\System\ControlSet001\	All keys	The control set used to manage system resources.
HKEY_LOCAL_MACHINE\System\ControlSet002\	All keys	Backup control sets are numbered 002, 003, 004, and so on. Typically, there will only be two control sets.
HKEY_LOCAL_MACHINE\System\ControlSet003\	All keys	Backup control sets are numbered 002, 003, 004, and so on. Typically, there will only be two control sets.

HKEY_LOCAL_MACHINE\System\ControlSet004\	All keys	Backup control sets are numbered 002, 003, 004, and so on. Typically, there will only be two control sets.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\	All keys	The current control set is mapped to the control set used for starting the computer.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\BootVerificationProgram		That program used to verify that the system booted correctly.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\Class		Contains information about CLSIDs (OLE).
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\ComputerName\ActiveComputerName		Holds the computer's current name.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\ComputerName\ComputerName		Holds the computer's name.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\CrashControl		Determines events when/if the system fails.
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Control\FileSystem		A description of the system file system (FAT or NTFS).
HKEY_LOCAL_MACHINE\System\Disk\	All keys	A description of the system disk.
HKEY_LOCAL_MACHINE\System\Select\	All keys	A description of the control set used.
HKEY_LOCAL_MACHINE\System\Setup\	All keys	A description of the system setup state.
HKEY_LOCAL_MACHINE\System\	All keys	A description of the system.
HKEY_USERS\	All keys	Contains general user information.

HKEY_USERS\DEFAULT\	All keys	The default user active when no other user is logged on. All information in .DEFAULT would also be found for specific users.
HKEY_USERS\DEFAULT\	AppEvents\EventLabels	Event labels are used to notify users (with sound) when events happen.
HKEY_USERS\DEFAULT\	AppEvents\Schemes	Schemes are used to apply which sounds are used for events.
HKEY_USERS\DEFAULT\	AppEvents	Contains application events, such as Startup, Document Open, and so on.
HKEY_USERS\DEFAULT\	Console	The system's command prompt for window(s) configuration.
HKEY_USERS\DEFAULT\	Control Panel	The System Control Panel used to configure Windows 2000.
HKEY_USERS\DEFAULT\	Control Panel\Accessibility	The Control Panel's Accessibility applet.
HKEY_USERS\DEFAULT\	Control Panel\Appearance	The Control Panel's Appearance applet.
HKEY_USERS\DEFAULT\	Control Panel\Colors	The Control Panel's Colors applet.
HKEY_USERS\DEFAULT\	Control Panel\Current	The Control Panel's Current applet.
HKEY_USERS\DEFAULT\	Control Panel\Custom Colors	The Control Panel's Custom Colors applet.

HKEY_USERS\DEFAULT\	Control Panel\Desktop	The Control Panel's Desktop applet.
HKEY_USERS\DEFAULT\	Control Panel\International	The Control Panel's International applet.
HKEY_USERS\DEFAULT\	Control Panel\IOProc	The Control Panel's IOProc applet.
HKEY_USERS\DEFAULT\	Control Panel\Keyboard	The Control Panel's Keyboard applet.
HKEY_USERS\DEFAULT\	Control Panel\MMCPL	The Control Panel's MMCPL applet.
HKEY_USERS\DEFAULT\	Control Panel\Mouse	The Control Panel's Mouse applet.
HKEY_USERS\DEFAULT\	Control Panel\Patterns	The Control Panel's Patterns applet.
HKEY_USERS\DEFAULT\	Control Panel\Screen Saver .3DFlyingObj	The Control Panel's Screen Saver.3DFlyingObj saved configuration.
HKEY_USERS\DEFAULT\	Control Panel\Screen Saver .3DPipes	The Control Panel's Screen Saver.3DPipes saved configuration.
HKEY_USERS\DEFAULT\	Control Panel\Screen Saver .Bezier	The Control Panel's Screen Saver.Bezier saved configuration.
HKEY_USERS\DEFAULT\	Control Panel\Screen Saver.Marquee	The Control Panel's Screen Saver.Marquee saved configuration.
HKEY_USERS\DEFAULT\	Control Panel\Screen Saver.Mystify	The Control Panel's Screen Saver.Mystify saved configuration.

HKEY_USERS\DEFAULT\	Control Panel\Screen Saver.Stars	The Control Panel's Screen Saver.Stars saved configuration.
HKEY_USERS\DEFAULT\	Control Panel\Sound	The Control Panel's Sound applet.
HKEY_USERS\DEFAULT\	Environment	Contains definitions of environment variables, used with both Windows and command prompts.
HKEY_USERS\DEFAULT\	Keyboard Layout	The keyboard layouts for NLS (National Language Support).
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows Help	Contains configurations for the Windows help system.
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows NT\CurrentVersion	Contains the Windows current software configurations.
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows NT\CurrentVersion\Devices	Contains configurations of software drivers for hardware.
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows NT	Contains Windows configuration items.
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows\CurrentVersion	Contains Windows configuration items.
HKEY_USERS\DEFAULT\	Software\Microsoft\Windows	Contains general information about Windows.
HKEY_USERS\DEFAULT\	Software\Microsoft	Contains information about Microsoft components and software

HKEY_USERS\DEFAULT\	Software	Contains software configurations (as compared to hardware configurations).
HKEY_USERS\DEFAULT\	UNICODE Program Groups	Unused on most systems.
HKEY_USERS\ <sid>\</sid>	All keys	Contains information for specific users as identified by <SID>.
HKEY_USERS\	<SID>\AppEvents\EventLabels\.Default	Contains information regarding application event labels, as in .DEFAULT.
HKEY_USERS\ <sid>_Classes\</sid>	<None>	New to Windows 2000, this key contains no usable information.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
[Click to access!](#)

APPENDIX B

Registry Data Types

A value entry in the registry can contain data in different formats. All registry data is stored in binary format, along with a value indicating the data's actual type. There are potentially hundreds of types of data that can be stored in the registry; however, Windows only uses fewer than 20 of these types. These types are classed as:

Common data types: These are supported and edited by RegEdit, RegEdt32, and most other registry tools.

Windows 2000-specific data types: These are supported and edited by RegEdt32 and some other registry tools.

Special and component/application-specific data types: These are both supported and unsupported by registry tools, but cannot usually be edited by users, except as binary data.

Keep in mind that registry editors actually allow the editing of unsupported data types, including data types that display as REG_UNKNOWN. However, editing is done in binary mode, requiring the user to have intimate knowledge of the data object.

The Data Types

If you find it necessary to modify the registry, it's important that you understand each data type, how data is stored for each data type, and so on. The registry data types are described in Table B.1.

TABLE B.1: KNOWN REGISTRY DATA TYPES

Type	Data Type Index (If Known)	Size	Description
REG_BINARY	3	0 or more bytes	A binary object that may contain any data
REG_COLOR_RGB	*	4 bytes	A color description
REG_DWORD	4	4 bytes	A DWORD value

REG_DWORD_BIG_ENDIAN	5	4 bytes	A value stored in reverse order of a DWORD value
REG_DWORD_LITTLE_ENDIAN	4	4 bytes	A DWORD value
REG_EXPAND_SZ	2	0 or more bytes	A string with an environment substitution placeholder
REG_FILE_NAME	*	0 or more bytes	A filename
REG_FILE_TIME	*	Unknown	A file time
REG_FULL_RESOURCE_DESCRIPTOR	9	Varies with contents	A list of hardware resources
REG_LINK	6	0 or more bytes	A Unicode string naming a symbolic link
REG_MULTI_SZ	7	0 or more bytes	A collection of strings, each separated by a null, with the final string terminated with two nulls
REG_NONE	0	Unknown	A data object with a defined type of REG_NONE for data that needn't be otherwise classified; different from REG_UNKNOWN
REG_QWORD		8 bytes	Twice the size of a REG_DWORD; a 64-bit integer variable
REG_QWORD_LITTLE_ENDIAN		8 bytes	The same as a REG_QWORD in size and format
REG_RESOURCE_LIST	8	Varies with contents	A list of resources used for a device
REG_RESOURCE_REQUIREMENTS_LIST	10	Varies with contents	A list of resources required by a driver
REG_SZ	1	0 or more bytes	A string terminated with a null
REG_UNKNOWN	(Undefined)	Unknown	An object whose type cannot be determined because the data type index is not valid

*At this time, these object types appear to be unsupported by Windows 2000.

Now, to determine what type a particular object is, the registry stores a data type index value that indicates the data type of the object. For instance, a REG_SZ object has a data type index of 1. For those objects in Table B.1 without a data type index value, Windows will list them as REG_UNKNOWN.

Oh, there is always the possibility that some additional data types will be added as Windows 2000 matures.



TIP Look in a RegEdit .reg file, and you will see many instances of hex(*n*) where *n* is a number, typically between 0 and 10. This number is the object's data type.

The following sections describe the compatibility of each data type.

REG_BINARY

REG_BINARY holds binary data. It is compatible with the following registry tool(s):

- RegEdit
- RegEdt32 Create

- RegEdt32 Edit
- RegEdt32 Display

REG_BINARY is the most basic data type used in the registry. Windows is able to express every registry data type in REG_BINARY form, although this can be very inconvenient.

Windows saves a binary object in the registry as a length and a series of bytes. When a REG_BINARY object is stored in the registry, the length parameter is preserved. However, the user is unable to change the length parameter except by changing the object's actual size.

REG_COLOR_RGB

REG_COLOR_RGB holds color definition. It is compatible with the following registry tool(s):

- RegEdt32 Display

REG_COLOR_RGB holds an RGB color index, which may be displayed by RegEdt32. None of the registry tools is able to create a REG_COLOR_RGB data object, and it cannot be determined whether Windows 2000 or any component supports this object type.

Future versions of Windows may support this object type. However, there is currently no support for the REG_COLOR_RGB object.

REG_DWORD

REG_DWORD holds a 32-bit number (that's 4 bytes folks, no more, no fewer) expressed in either decimal or hexadecimal. REG_DWORD is compatible with the following registry tool(s):

- RegEdit
- RegEdt32 Create
- RegEdt32 Edit
- RegEdt32 Display

Like REG_BINARY, the REG_DWORD type is a basic data type for registry value entries.

REG_DWORD_BIG_ENDIAN

REG_DWORD_BIG_ENDIAN holds a DWORD value. REG_DWORD_BIG_ENDIAN is compatible with custom-written registry tools. The original Windows NT 4 Resource Kit's RegChg program worked with this registry data type.

Different computers store numbers in memory in different orders. Two orders are used: big endian and little endian. The Intel processor stores numbers in little endian format.

In the big endian format, the 4 bytes of a DWORD value are stored with the highest-order byte at the highest address, and the lowest-order byte at the lowest address. See Table B.2 for an example of how the value 0x12345678 (in decimal that's 305419896) would be stored at memory address 0x5.

TABLE B.2: A DWORD VALUE STORED IN BIG ENDIAN FORMAT

Address	Value
0x5	12
0x6	34
0x7	56
0x8	78

REG_DWORD_LITTLE_ENDIAN

REG_DWORD_LITTLE_ENDIAN holds a DWORD value. REG_DWORD_LITTLE_ENDIAN is compatible with the following registry tool(s). (Note that this data type is typically treated as

REG_DWORD.)

- RegEdit
- RegEdt32 Create
- RegEdt32 Edit
- RegEdt32 Display

In the little endian format, the 4 bytes of a DWORD value are stored with the highest-order byte at the lowest address, and the lowest-order byte at the highest address. Table B.3 shows an example of the value 0x12345678 (in decimal that's 305419896) stored at memory address 0x5.

TABLE B.3: A DWORD VALUE STORED IN LITTLE ENDIAN FORMAT

Address	Value
0x5	78
0x6	56
0x7	34
0x8	12

Both Windows 2000/NT and Windows 95/98 run on computers that store numbers in little endian format. Some operating systems are designed for big endian computers. When an operating system is designed for a different endian than what the hardware supports, the operating system will convert as necessary.



NOTE It is generally unnecessary for users to consider endian issues. The only time that endian format becomes important is when transferring data between two dissimilar computer systems using raw binary transfer methods. Since virtually all transfer methods (including data transfers over the Internet) do not use raw binary transfers, it's rarely an issue.

In all cases, the Windows 2000 registry treats REG_DWORD_LITTLE_ENDIAN as a REG_DWORD type. There is no real difference between these two types in the Windows 2000 registry.

REG_EXPAND_SZ

The REG_EXPAND_SZ data object contains a single string terminated with a null; a null is a character whose value is zero. The REG_EXPAND_SZ data object is compatible with the following registry tool(s):

- RegEdt32 Create
- RegEdt32 Edit
- RegEdt32 Display

This string may contain one or more unexpanded environment variables.

There are many examples of REG_EXPAND_SZ in the registry, most of which are references to files accessed from the environment variables %systemroot%, %systemdrive%, and %path%.

Windows substitutes environment variables when percent (%) signs surround the name. In a command-prompt window, the command SET displays a list of all the current environment variables. These environment variables (with surrounding percent signs) are also used from command prompts or in batch files. In the next example, lines that I typed are in bold; the lines starting with REM are comments:

```
Windows 2000 18:05:31 C:\TEMP
REM-display the contents of the environment variable SystemRoot
Windows 2000 18:05:32 C:\TEMP
set systemroot
SystemRoot=C:\WINNT40
```

```
Windows 2000 18:05:34 C:\TEMP
REM-Use the environment variable SystemRoot
Windows 2000 18:05:35 C:\TEMP
```



```

dir %SystemRoot%\*.bat
Volume in drive C is (c) - Boot drive
Volume Serial Number is CC56-5631

Directory of C:\WINNT
02/10/98 03:55p          46          PB.BAT
    1 File(s)          46 bytes
    152,050,688 bytes free

```

Windows 2000 18:05:40 C:\TEMP

REG_FILE_NAME

The REG_FILE_NAME object type is not currently used (or supported) other than in RegEdt32 display mode. The REG_FILE_NAME object type is compatible with the following registry tool(s):

- RegEdt32 Display

None of the registry tools is able to create a REG_FILE_NAME data object, and it's unclear whether either Windows 2000 or any component supports this object type.

It is possible that future versions of Windows will support this object type. However, there is currently no support for the REG_FILE_NAME object.

REG_FILE_TIME

The REG_FILE_TIME object type is not currently used (or supported) other than in RegEdt32 display mode. The REG_FILE_TIME object type is compatible with the following registry tool(s):

- RegEdt32 Display

None of the registry tools is able to create a REG_FILE_TIME data object, and it's unclear whether either Windows or any component supports this object type.

It is possible that future versions of Windows will support this object type. However, there is currently no support for the REG_FILE_TIME object.

REG_FULL_RESOURCE_DESCRIPTOR

The REG_FULL_RESOURCE_DESCRIPTOR object contains a list of hardware resources that a physical device is using. The REG_FULL_RESOURCE_DESCRIPTOR object is compatible with the following registry tool(s):

- RegEdt32 Display

This information is detected and written into the HKEY_LOCAL_MACHINE\Hardware\Description key by the system at boot-up time.

Generally, it would be very unwise to edit a REG_FULL_RESOURCE_DESCRIPTOR object. RegChg allows the creation of a REG_FULL_RESOURCE_DESCRIPTOR object if an edit is necessary.

Table B.4 shows the objects found in a REG_FULL_RESOURCE_DESCRIPTOR.

TABLE B.4: OBJECTS IN A REG_FULL_RESOURCE_DESCRIPTOR

Object	Size	Details
--------	------	---------

Interface Type	4 bytes	<p>One of the following values:</p> <ul style="list-style-type: none"> -1 = Interface Type Undefined 0 = Internal 1 = ISA 2 = EISA 3 = Micro Channel 4 = Turbo Channel 5 = PCI Bus 6 = VME Bus 7 = NuBus 8 = PCMCIA Bus 9 = Cbus 10 = MPI Bus 11 = MPSA Bus 12 = Processor Internal 13 = Internal Power Bus 14 = PNP ISA Bus 15 = Maximum Interface Type
Bus Number	4 bytes	Bus number for this resource
Type*	1 bytes	<p>Uses CM_RESOURCE_TYPE:</p> <ul style="list-style-type: none"> 0 = Null 1 = Port 2 = Interrupt 3 = Memory 4 = DMA 5 = Device Specific 6 = Maximum
Share Disposition*	1 bytes	<p>Uses CM_SHARE_DISPOSITION:</p> <ul style="list-style-type: none"> 0 = Undetermined 1 = Device Exclusive 2 = DriverExclusive2 3 = Shared
Flags*	2 bytes	<p>The resource flag values depend on the resource type (see Type):</p> <p>For an Interrupt resource type:</p> <ul style="list-style-type: none"> 0 = INTERRUPT_LEVEL_SENSITIVE 1 = INTERRUPT_LATCHED <p>For a Memory resource type:</p> <ul style="list-style-type: none"> 0x0000 = READ_WRITE 0x0001 = READ_ONLY 0x0002 = WRITE_ONLY 0x0004 = PREFETCHABLE 0x0008 = COMBINEDWRITE 0x0010 = 24 <p>For a Port resource type:</p> <ul style="list-style-type: none"> 0 = PORT_MEMORY 1 = PORT_IO <p>For a DMA resource type:</p> <ul style="list-style-type: none"> 0x0000 = DMA_8 0x0001 = DMA_16 0x0002 = DMA_32
Start*	8 bytes	64 bit: 32 bits low, 32 bits high port address
Length*	4 bytes	Length of port address
Level*	4 bytes	Interrupt

Vector*	4 bytes	Interrupt
Affinity*	4 bytes	Interrupt
Start*	8 bytes	64-bit physical memory addresses
Length*	4 bytes	Length of memory
Channel*	4 bytes	DMA channel number
Port*	4 bytes	DMA port number
Reserved 1*	4 bytes	Reserved for future DMA use
Data Size*	4 bytes	Device-specific data size
Reserved 1*	4 bytes	Reserved for future use
Reserved 2*	4 bytes	Reserved for future use

*These objects, as a group, may be repeated as needed.

REG_LINK

REG_LINK is a Unicode string naming a symbolic link. REG_LINK is compatible with none of the registry tool(s).

This type is irrelevant to device and intermediate drivers and should be of no interest to anyone except programmers.

REG_MULTI_SZ

The REG_MULTI_SZ object consists of one or more strings. The REG_MULTI_SZ object is compatible with the following registry tool(s):

- RegEdt32 Create
- RegEdt32 Edit
- RegEdt32 Display

Windows separates each string from the next using a null. The final string terminates with two nulls. For strings that are Unicode (all strings in the Windows 2000 registry are Unicode), the null shall be the same width as a Unicode character. For example (\0 indicates a null character):

```
String one\0
String two\0
String three\0
Last string\0\0
```



NOTE What the heck is a null? A *null character* has a numeric value of zero, irrespective of the character's width.

REG_NONE

REG_NONE is data with no particular type. REG_NONE is compatible with none of the registry tool(s).

Notice that this type is different from REG_UNKNOWN. REG_NONE is data that is stored in binary format, with 0 or more bytes of information. Generally, REG_NONE objects are created by default, although some components and applications may create a REG_NONE object intentionally.

REG_QWORD

REG_QWORD holds a 64-bit number expressed in either decimal or hexadecimal. REG_QWORD is not compatible with any of the registry tool(s).

Like REG_DWORD, the REG_QWORD type is a basic data type, containing a numeric value. This registry type will become more commonly used with future 64-bit versions of Windows.

REG_QWORD_LITTLE_ENDIAN

REG_QWORD_LITTLE_ENDIAN holds a 64-bit number expressed in either decimal or hexadecimal. This type is identical to the REG_QWORD type. REG_QWORD_LITTLE_ENDIAN is not compatible with any of the registry tool(s).

Like REG_DWORD, the REG_QWORD_LITTLE_ENDIAN type is a basic data type, containing a numeric value. This registry type will become more commonly used with future 64-bit versions of Windows.

REG_RESOURCE_LIST

A REG_RESOURCE_LIST object describes information about resources used by a specific device. A REG_RESOURCE_LIST object is compatible with the following registry tool(s):

- RegEdt32 Display

Found primarily in HKEY_LOCAL_MACHINE\Hardware\ResourceMap, there are entries for each device installed in the system. REG_RESOURCE_LIST information is organized by bus structure:

- Cbus
- EISA
- Internal
- ISA
- Micro Channel
- MPI Bus
- MPSA Bus
- NuBus
- PCI Bus
- PCMCIA Bus
- Turbo Channel
- VME Bus

Information stored in REG_RESOURCE_LIST may include:

- Device Specific Data Size
- Device Specific Data Reserved1
- Device Specific Data Reserved2
- DMA Channel
- DMA Port
- DMA Reserved1
- Interrupt Affinity
- Interrupt Level
- Interrupt Vector
- Memory Length
- Memory Physical Address
- Memory Start
- Port Length
- Port Physical Address
- Port Start

Table B.5 shows the arrangement of information that is stored in the REG_RESOURCE_LIST registry data type.

TABLE B.5: ARRANGEMENT OF ACTUAL VALUES IN THE REG_RESOURCE_LIST TYPE

Object	Size	Details
Version	2 bytes	Version number

Revision	2 bytes	Revision number
Count	4 bytes	Object count
Option	1 bytes	Undocumented
Type*	1 bytes	Uses CM_RESOURCE_TYPE: 0 = Null 1 = Port 2 = Interrupt 3 = Memory 4 = DMA 5 = Device Specific 6 = Maximum
Share Disposition*	1 bytes	Uses CM_SHARE_DISPOSITION: 0 = Undetermined 1 = Device Exclusive 2 = Driver Exclusive 3 = Shared
Spare1*	1 bytes	Unused; it is present to force proper alignment.
Flags*	2 bytes	The resource flag values depend on the resource type (see Type near the beginning of this table): For an Interrupt resource type: 0 = INTERRUPT_LEVEL_SENSITIVE 1 = INTERRUPT_LATCHED For a Memory resource type: 0x0000 = READ_WRITE 0x0001 = READ_ONLY 0x0002 = WRITE_ONLY 0x0004 = PREFETCHABLE 0x0008 = COMBINEDWRITE 0x0010 = 24 For a Port resource type: 0 = PORT_MEMORY 1 = PORT_IO For a DMA resource type: 0x0000 = DMA_8 0x0001 = DMA_16 0x0002 = DMA_32
Spare2*	2 bytes	Align the remainder of the objects.
Length*	4 bytes	Only Port type resources use this variable.
Alignment*	4 bytes	Only Port type resources use this variable.
Minimum Address*	8 bytes	Only Port type resources use this variable.
Maximum Address*	8 bytes	Only Port type resources use this variable.
Length*	4 bytes	Only Memory type resources use this variable.
Alignment*	4 bytes	Only Memory type resources use this variable.
Minimum Address*	8 bytes	Only Memory type resources use this variable.
Maximum Address*	8 bytes	Only Memory type resources use this variable.
Minimum Vector*	4 bytes	Only Interrupt type resources use this variable.
Maximum Vector*	4 bytes	Only Interrupt type resources use this variable.
Minimum Channel*	4 bytes	Only DMA type resources use this variable.
Maximum Channel*	4 bytes	Only DMA type resources use this variable.

*These objects, as a group, may be repeated as needed.

REG_RESOURCE_REQUIREMENTS_LIST

REG_RESOURCE_REQUIREMENTS_LIST objects contain lists of hardware resources that a device driver would require. REG_RESOURCE_REQUIREMENTS_LIST objects are compatible with the following registry tool(s):

- RegEdt32 Display

This list of hardware resources is used to update the HKEY_LOCAL_MACHINE\Hardware\ResourceMap subkey.

Table B.6 shows the arrangement of objects in the REG_RESOURCE_REQUIREMENTS_LIST object.

TABLE B.6: OBJECTS IN A REG_RESOURCE_REQUIREMENTS_LIST

Object	Size	Details
List Size	4 bytes	
Interface Type	4 bytes	One of the following values: -1 = Interface Type Undefined 0 = Internal 1 = ISA 2 = EISA 3 = Micro Channel 4 = Turbo Channel 5 = PCI Bus 6 = VME Bus 7 = NuBus 8 = PCMCIA Bus 9 = Cbus 10 = MPI Bus 11 = MPSA Bus 12 = Processor Internal 13 = Internal Power Bus 14 = PNP ISA Bus 15 = Maximum Interface Type
Bus Number	4 bytes	
Slot Number	4 bytes	
Reserved[3]	12 bytes	
Alternative Lists	4 bytes	
Version*	2 bytes	
Revision*	2 bytes	
Count*	4 bytes	
Option*	1 byte	
Type*	1 bytes	Uses CM_RESOURCE_TYPE: 0 = Null 1 = Port 2 = Interrupt 3 = Memory 4 = DMA 5 = Device Specific 6 = Maximum
Share Disposition*	1 bytes	Uses CM_SHARE_DISPOSITION: 0 = Undetermined 1 = Device Exclusive 2 = DriverExclusive2 3 = Shared
Spare1*	1 bytes	Unused (typically used to force proper alignment).

Flags*	2 bytes	The resource flag values depend on the resource type (see Type near the beginning of this table): For an Interrupt resource type: 0 = INTERRUPT_LEVEL_SENSITIVE 1 = INTERRUPT_LATCHED For a Memory resource type: 0x0000 = READ_WRITE 0x0001 = READ_ONLY 0x0002 = WRITE_ONLY 0x0004 = PREFETCHABLE 0x0008 = COMBINEDWRITE 0x0010 = 24 For a Port resource type: 0 = PORT_MEMORY 1 = PORT_IO For a DMA resource type: 0x0000 = DMA_8 0x0001 = DMA_16 0x0002 = DMA_32
Spare2*	2 bytes	Unused.
Length*	4 bytes	Only Port type resources use this variable.
Alignment*	4 bytes	Only Port type resources use this variable.
Minimum Address*	8 bytes	Only Port type resources use this variable.
Maximum Address*	8 bytes	Only Port type resources use this variable.
Length*	4 bytes	Only Memory type resources use this variable.
Alignment*	4 bytes	Only Memory type resources use this variable.
Minimum Address*	8 bytes	Only Memory type resources use this variable.
Maximum Address*	8 bytes	Only Memory type resources use this variable.
Minimum Vector*	4 bytes	Only Interrupt type resources use this variable.
Maximum Vector*	4 bytes	Only Interrupt type resources use this variable.
Minimum Channel*	4 bytes	Only DMA type resources use this variable.
Maximum Channel*	4 bytes	Only DMA type resources use this variable.

*These objects, as a group, may be repeated as needed.

REG_SZ

REG_SZ contains a single string terminated with a null. REG_SZ is compatible with the following registry tool(s):

- RegEdit
- RegEdt32 Create
- RegEdt32 Edit
- RegEdt32 Display

The REG_SZ string will be in Unicode for Windows 2000 installations.

REG_UNKNOWN

If Windows encounters an undefined registry data type, the REG_UNKNOWN object type is used. The REG_UNKNOWN object type is compatible with the following registry tool(s):

- RegEdit

- RegEdt32 Display

Windows permits system components and system applications to write their own types of data into the registry. Moreover, none of the registry tools, such as the registry editors, will know about these data types. Whenever a registry tool encounters a data type that it does not know about, the tool will display the data in binary format and flag it as REG_UNKNOWN.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

Search Tips

Advanced Search

PUBLICATION LOOKUP

JUMP TO TOPIC


Desktop Library
 Click to access!
**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

[Previous](#) [Table of Contents](#) [Next](#)

APPENDIX C

Where Can I Get More Help?

Maybe this book didn't have enough information for you, or more likely, you need some very specialized help. There are many sources of assistance. Many are free, and some are costly; but generally, we get what we pay for in life. As a rule, personalized, hand-holding assistance can be expensive—consultants typically cost between \$100 and \$350 an hour, and usually charge for travel time and other expenses.

User groups and other self-help sources are cheaper, but you may not profit from assistance that is quickly shot out to you in email. You must determine if such help will work for you.

The Web

There are many sites on the World Wide Web that offer assistance to Windows users. No site is specific to the registry, though many of these sites offer either a section for the registry or a general area that may be helpful to someone who has a registry problem. The site at <http://www.jsiinc.com/reghack.htm> is run by Jerold Schulman (JSI, Inc.). It is jam-packed with helpful tips and ideas. The registry section that Jerold has set up includes almost 500 different registry-related tips and techniques.

The site at <http://www.ntfaq.com> is run by John Savill (SavillTech Ltd.). It has a number of very interesting features including answers to over 1600 common questions, a Java and CGI search engine with a single-file version that can be downloaded, and several help file versions that can be used locally.

The site at <http://www.bhs.com> is run by Beverly Hills Software. It is one of the premier sites for users of Windows NT. A close alliance with Microsoft has allowed BHS to remain in the forefront of Windows NT support and technology.

The site at <http://www.sysinternals.com/> is run by Mark Russinovich and Bryce Cogswell. These two experts in the Windows NT field offer freeware utilities that they have written. (Freeware is software that is essentially free, though freeware authors may ask for a donation.)

At <http://www.wugnet.com/wininfo/win2000/>, you'll find Paul Thurrott's Super Site for Windows 2000, a site dedicated to Windows 2000. Paul is an expert who is well known for his expertise with Windows NT. Paul's main site for Windows is <http://www.wugnet.com/wininfo/>, a site well worth browsing.

The site <http://www.betaos.com/> has lots of information about Windows 2000. This site emphasizes the version of Windows that is in beta, but also has coverage of the current product.

When looking for hardware drivers, the site <http://www.windrivers.com/> is a good place to start. Go there and do a search, and you will receive all relevant information about the driver.

The Interior Alaska Windows NT Users Group hosts the site <http://www.iawntug.org>. This site has a number of useful features and links.

<http://gamble.risjak.net/html/windows1.htm> is non-English in places, but this site can still be easily navigated by those who don't speak Serbian.

The URL <http://www.ping.be/bios/> will take you to all the information about the PC BIOS. Included in this site are tips on how to update a BIOS, BIOS numbers, and a FAQ.

At <http://support.microsoft.com/support/NTServer/content/faq/default.asp>, you'll find Microsoft's online FAQ list for Windows NT Server. This site, along with other Microsoft sites, form Microsoft's online interactive support services.

The site at <http://backoffice.microsoft.com> is Microsoft's site for information about Windows 2000 and the other members of the Microsoft BackOffice family. BackOffice is Microsoft's business-solutions offering for advanced users.

You can contact Microsoft's technical people by using Microsoft's sites at <http://www.microsoft.com/support/> and <http://backoffice.microsoft.com/bbs/newsgroups.asp>. The second URL allows access to the newsgroups described in the "Newsgroups" section later in this appendix.

Companies and Magazines

Many companies offer support, training, and other services. These companies are usually a combination of consultants and resellers. To find larger support companies, peruse magazines such as *PC Magazine*, *InfoWorld*, and *PC Week*.

Consultants

In virtually all locations, you will find independent consultants. They are sometimes difficult to find, and most consultants don't advertise. Consultants are often one- and two-person outfits, and they are typically expensive.

Similar to the procedure with stores (discussed next), always check references if possible. Check the consultant's areas of expertise. A consultant who is respected in the field and writes articles and books is probably qualified. Ask the consultant if they are using tools, such as Microsoft TechNet and MSDN. If not, the consultant is probably going to spend a lot of time reinventing the wheel by trying to solve problems that were solved and documented long ago.

Consultants' rates vary, with most requiring either a minimum charge of one to two hours, and/or a charge for travel time. Most consultants do not have a storefront, so walk-in business is not the norm. Hourly rates vary from a low of about \$100 to a high of almost \$350. Generally, the higher-priced service people are going to solve the problem more quickly than the lower-priced ones.

Computer Stores

Many computer stores offer services to users of Microsoft Windows 2000. Start with local, non-chain computer stores. Most chain stores do not specialize in offering service, while non-chain stores usually excel in service. Ask for references.

Consider bringing in the computer if possible, to save on the cost of repair. Hourly rates vary from a low of about \$50 to a high of almost \$200.



NOTE No one has solved the problem of paying for incompetence. You hire someone to do a job and may actually end up paying for his or her learning experience. To avoid paying for work that is not solving the problem, be sure to check references.

User Groups

User groups form a two-way street: if you get assistance from the group, you must support them as well. Most user groups are either not funded or are very limited financially—don't ask them to call you back long distance.

No user group list is ever up to date. That's life—things are always changing, especially with users' groups. Table C.1 is a listing of some Windows NT users' groups. For any group you might be interested in, please either check the group's Web page or email the contact at the address given for more information about such points as the group's meeting place, date, and time. Some user groups don't hold regular meetings, but there may be other assistance they can provide.

Sizes of groups range from small (perhaps fewer than 20 members), to medium (about 20 to 100 members), to large (generally more than 100 members).

Is there no user group in your area? Start one! Most newspapers will provide free publicity. Usually meetings can be at local libraries, schools, or at your business. You will meet new people, make friends, and become very popular. Oh, and if you didn't guess, starting a user group will take some time and effort—but it's worth it!

TABLE C.1: WINDOWS NT USER GROUPS AROUND THE WORLD

User Group	Web Page Address	City, State, Country	Group Size	Contact Name	Contact E-Mail Address
------------	------------------	----------------------	------------	--------------	------------------------

Mastering Windows 2000 Registry: Where Can I Get More Help?

Birmingham Windows NT User Group	http://www.bwntug.org/	Birmingham, Alabama	Small	Mike Chilson	mchilson@bwntug.org
Huntsville NT Users Group—HUNTUG	http://www.mbsinc.com/huntug	Huntsville, Alabama	Medium	Dan Wygant	huntug@mbsinc.com
Anchorage Windows NT Users Group	http://www.rmm.com/awntug/	Anchorage, Alaska	Unknown	Jon Dawson	jdawson@anc.ak.net
Interior Alaska Windows NT Users Group	http://www.iawntug.org/	Fairbanks, Alaska	Small	Roger M Marty	marty@rmm.com
Windows NT Group (Argentina)	http://mspro.smart.com.ar	Capital Federal, Argentina	Medium	Ricardo Fig	RFig@smart.com.ar
NT Users Group	None	San Miguel, Argentina	Unknown	Angulo	aANGULO@TELEFONICA.COM.AR
Phoenix PCUG NT SIG	None	Phoenix, Arizona	Unknown	Ray Moore	nt-sig@phoenixpcug.org
Brisbane NT Users Group	http://www.ozemail.com.au/~dkowald/bntug.htm	Brisbane, Australia	Unknown	Derek Kowald	dkowald@ozemail.com.au
Queensland NT Users Group	http://www.qntug.asn.au/	Brisbane, Australia	Medium	David Steadson	davids@ambience.com
Microsoft Systems User Group	http://www.adfa.oz.au/~rjn/msug/	Canberra, Australia	Medium	Neil Pinkerton	neil.pinkerton@cao.mts.dec.com
Melbourne NT User Group	http://www.co.rmit.edu.au/ausnt	Melbourne, Australia	Small	Mark A Gregory	m.gregory@rmit.edu.au
Perth NT Users Group	None	Perth, Australia	Small	Kevin Merritt	kevinm@acslink.aone.net.au
Suntug NT Users Group	None	Sunshine Coast, Australia	Unknown	Peter Williams	peter@beachaccess.com.au
Sydney NT Users Group	None	Sydney, Australia	Unknown	J. Noiles	killara@hotmail.com
NT Users in Vienna	None	Vienna, Austria	Unknown	Gerhard Wenk	wenk@vienna.at
Windows NT user group of Austria	http://www.wug.or.at	Vienna, Austria	Medium	Sepp Reichholf	josef.reichholf@reichholf.co.at
WUG-Oesterreich	http://www.wug.or.at	Vienna, Austria	Medium	Windows User Group Oesterreich	josef.reichholf@reichholf.co.at
MBS	None	Belgium	Small	Gonzalez Jose	jgon@usa.net
BeNTUG—Belgian NT User Group	http://www.bentug.org	Brussels, Belgium	Medium	Unknown	info@bentug.org
NTX—Belgian Windows NT Corporate Account Group	http://www.econ.kuleuven.ac.be/ntx	Leuven, Belgium	Medium	Wim Van Holder	Wim.VanHolder@econ.kuleuven.ac.be
DIN NT User Group	None	Curitiba, Brazil	Small	Wallace A.B.S Macedo	wallace@furukawa.com.br
Exchange	None	Rio de Janeiro, Brazil	Unknown	Roberto Boclin	boclin@unikey.com.br
NT User Group Brazil—NT.br	None	Rio de Janeiro, Brazil	Unknown	Paul Smith	psmith@centroin.com.br
NT Services	None	Sao Paulo, Brazil	Unknown	Ana	analucia@opus.com.br
Fresno PC Users Group	None	Fresno, California	Large	Susy Ball or George Simpson	susyball@aol.com georgesi@cybergate.com
jk	None	Los Angeles, California	Medium	LA User's Group	bla@aol.comUser

Los Angeles Windows NT User Group	http://www.lantug.org	Los Angeles, California	Large	Jerry Boshear	info@lantug.org
Riverside NT User Group	None	Riverside, California	Unknown	Chris Navigato	chris@navigato.com
Sacramento NT Users Group	None	Sacramento, California	Unknown	Steven W. Linthicum	slinthe@ns.net
San Diego County Windows NT User Group	http://www.sdwntug.org	San Diego, California	Unknown	Chris Dickey	webmaster@sdwntug.org
Northern California Windows NT Users Group	http://www.actioninc.com/winntug.htm	San Francisco, California	Unknown	Check with the group's Web page	None
NT Engineering Assoc. (NTEA)	http://www.wntea.org	San Jose, California	Large	Michael Masterson	mmasterson@taos.com
Southern California Exchange Users Group	None	Santa Monica, California	Medium	Ivan K. Nikkhoo	ivann@vertexsystems.com
Edmonton Windows NT/BackOffice User Group	http://www.tnc.com/mug	Edmonton, Alberta, Canada	Medium	Leigh Anne Chisholm	lachisho@tnc.com
Quebec Windows NT Users Group	None	Montreal, Quebec, Canada	Unknown	Maxime Bombardier	maxime@4dm.com
NT Networking and MCSE Forum	http://move.to/mcse_mtl	Montreal, Quebec, Canada	Small	Florian Hehlen	fh@binex.com
Vancouver Island NT User Group	http://www.vintug.bc.ca	Victoria British Columbia, Canada	Guy Gondor	webmaster@vintug.bc.ca	
Vancouver NT Users Group (VANTUG)	http://www.vantug.com	Vancouver British Columbia, Canada	Large	Nancy Pearce	nancy@steeves.bc.ca
Windows NT Group, La Serena, Chile	None	La Serena, Chile	Small	Pablo Peña	win-nt@mercury.andesnet.cl
Download Center	None	Nj, China	Unknown	Ghost	Heaven_ghost@163.net
NT-Colombia-Cali	None	Cali, Colombia	Small	Guillermo Matiz	gmatizo@col2.telecom.com.co
GUNTCOL	None	Cartagena de Indias, Santafé de Bogotá, Colombia	Unknown	Tomas Mac Master	ima@axisgate.com
Rocky Mountain Windows NT User Group	http://www.rmwnntug.org	Denver, Colorado	Unknown	Dennis Martin	76314.1441@compuserve.com
Northern Colorado Windows NT Users Group	None	Fort Collins, Colorado	Unknown	Eric Leftwich	ncwntug@ataman.com
Connecticut Area NT User Group	None	Farmington, Connecticut	Medium	Art Alexander	Arthur_Alexander@msn.com
Windows NT Group Consults	None	Costa Rica	Unknown	Alejandro Esquivel Rodriguez	aesquiv@irazu.una.ac.cr
Zagreb Windows NT User Group	http://winnt.zv.hr	Zagreb, Croatia (Hrvatska)	Medium	Alejandro Andrej Skendrovic	askendrovic@zv.hr

Mastering Windows 2000 Registry: Where Can I Get More Help?

Association of Windows NT Systems Professionals (NT*Pro)		http://www.ntpro.org	Washington, DC	Large	Charles Kelly	ckelly@msn.com
S.Florida NT Group	None		Ft. Lauderdale, Florida	Unknown	Gabriel B.H. Polmar	mongo@netrox.net
Jacksonville Back Office User Group		http://www.jbug.com	Jacksonville, Florida	Unknown	Kevin Haynes	jbug@tech-point.com
Jacksonville BackOffice Users Group	None		Jacksonville, Florida	Small	James Farhat	ajbug@tech-point.com
NW Florida NT Users Group	None		Panama City, Florida	Small	Jeff Bankston	jeff@mail.bciassoc.com
French Windows NT Users Group (FWNTUG)		http://www.fwntug.org/	France	Large	Anthony Moillic	AMoillic@fwntug.org
Atlanta Back Office User Group	None		Atlanta, Georgia	Large	Lisa Thomassie	mind_share@msn.com
Looking for One	None		Atlanta, Georgia	Unknown	Donameche Miller	Donameche@Tsi.Tsifax.Com
German NT User Group	None		Germany	Unknown	Paulette Feller	pfeler@skd.de
NT Guatemala Users Group	None		Guatemala City, Guatemala	Small	Jose A. Chajon	viacomp@guate.net
Manu	None		Essen, Germany	Unknown	Lars Hasshoff	lhasshof@manu.com
Big Island NT		http://bintug.org	Kamuela, Hawaii	Small	Matthew Pearce	sysop@bintug.org
Boise NT User Group (BNTUG)		http://www.bntug.com (under development)	Boise, Idaho	Small	Kelly Householder	kellyh@primenet.com
Windows NT Users Group of the Chicago Computer Society		http://billslater.com/ntsig.htm	Chicago, Illinois	Small	Bill Slater	slater@xsite.net
infosys nt group	None		Bangalore, India	Unknown	Krishnakumar	krishnakumar@inf.com
NT Developers Group, INDIA	None		Bangalore, India	Small	Nilendu Pal	nilendu@wipinfo.soft.net
Trigent-systems	None		Bangalore, India	Small	Saravana Bhavan	saravana_bk@trigent.com
Migration from Novell Netware to Windows NT	None		Bombay, India	Medium	Manoj Rakheja	manoj.rakheja@owenscorning.com
KPMF NT User Group	None		Chennai, India	Small	Solomon Sagayaraj	solomonj@hotmail.com
NTGROUP	None		Chennai, India	Small	M.Subramonian	MSM24@HOTMAIL.COM
Welcome Group	None		Chennai, India	Unknown	Santhosh	santhosh@pentafor.com
Mavericks	None		Madras, India	Small	Ramesh Venkatraman	tnvramesh@hotmail.com
HAWKS	None		New Delhi, India	Small	Himanshu Sharma	himanshu@rsysi.stpn.soft.net
Aroostook NT User Group		http://www.aroostook.org	Huntington, Indiana	Unknown	Jerry Curtis	winnt_users@aroostook.org
Windows NT Users Group of Indianapolis		http://www.wintugi.org	Indianapolis, Indiana	Medium	Peter Vanvleet	vanvleetp@juno.com
gba c-250	None		Bandung, Indonesia	Unknown	Ipoel	ikocel@rocketmail.com

NT Blom Indonesia	None	Kemang Raya 24 Jakarta 12730, Indonesia	Unknown	Muhammad Abduh	ma@blom.co.id
IRELAND	None	Ireland	Unknown	John Doherty	doherty_mj@hotmail.com
Windows NT Italian User Group WNTIUG	None	Potenza, Italy	Medium	Enrico Fasulo	webmaster@powernet.it
WIN NT-BHS Gruppo Italia-Discussione	None	Rome, Italy	Unknown	Davide Rossi	rossi@ancitel.it
Windows NT Users Group of Indianapolis	http://www.wintugi.org	Indianapolis, Indiana	Medium	Peter Vanvleet	vanvleetp@juno.com
ClieNT & Servers Iowa NT User Group	None http://nt-resources.idesignworks.com	Ames, Iowa Cedar Rapids, Iowa	Unknown Medium	Rick Gammon Don Howard	bulldog@netins.net DonHoward@earthlink.net
IOWA Windows NT User Group	None	Iowa City, Iowa	Unknown	Alex Postnikov	apostnik@blue.weeg.uiowa.edu
MS Israel	None	Tel Aviv, Israel	Medium	Cohen Gal	root@widecom.sys.co.il
Japan Windows NT Users Group	http://www.jwntug.or.jp/	Tokyo, Japan	Large	Ryoji Kaneko	rkaneko@jwntug.or.jp
Tokyo English NT Users Group	http://www.tentug.org	Tokyo, Japan	Unknown	Administrator	admin@tentug.org
Kansas City Windows NT Users Group	None	Kansas City, Kansas	Unknown	Steve Rodgers	srodgers@kumc.edu
Northeast Kansas Windows NT Users Group	None	Unknown, somewhere in Kansas	Unknown	Lad	lad@tinman.dot.state.ks.us
JPA Communications	None	Elizabethtown, Kentucky	Small	June Mizoguchi	jpa@ekx.infi.net
NT Bluegrass Users Group (NTBUG)	http://www.rmm.com/ntbug	Lexington, Kentucky	Unknown	Richard K. Marshall	rkm@mis.net
NT-ISP-Support	None	Lexington, Kentucky	Unknown	Lee Murphy	web@chapel1.com
LitNT	None	Kaunas, Lithuania	Small	Ricardas Baltaduonis	baltad@soften.ktu.lt
Louisiana WinNT Users Group	None	Monroe, Louisiana	Small	Richard Driggers	richard@lawinntug.org
NT User Group	None	Unknown, somewhere in Louisiana	Unknown	Laird Goolsby	laird@addtech.com
NT4U	http://www.info-trek.com/nt4uhome.htm	Kuala Lumpur, Malaysia	Unknown	Noel Teng	aaant4u@lycosemail.com
PowerNT	None	Kuala Lumpur, Malaysia	Large	Ahmad Ridzuan Mohd Noor	ridzuan.mn@feldaprodata.com.my
Baltimore Windows NT Users Group (BUGME)	http://www.techarchitects.com/bugme	Baltimore, Maryland	Small	Jack Bauer	jackb@techarchitects.com
New England Computer Society Windows 95/NT User Group	None	Springfield, Massachusetts	Small	Rene M. Laviolette	rene@costimator.com

Mastering Windows 2000 Registry: Where Can I Get More Help?

NENTUG	http://www.nentug.org		Waltham, Massachusetts	Medium	Mike Eisenberg	mikeei@microsoft.com
BWUG—Boston Windows User Group	http://www.bwug.org		Waltham, Massachusetts	Medium	Steve Allen	sallen@world.std.com
WinTech Group	http://www.webss.com		Boston/Waltham, Massachusetts	Small	Len Segal	lsegal@fcl.net
GUNT—Grupo de Usuarios de NT	None	Celaya, Mexico	Unknown	German Rodriguez	soporte@mail.mindvox.ciateg.mx	
West Michigan NT User Group (WMNTUG)	http://www.wmntug.org		Grand Rapids, Michigan	Medium	Brett R. Mello	bmello@perrigo.com
Great Lakes Windows NT Users Group	http://www.ntexperts.com/greatlakes		Southfield, Michigan	Large	Tim Rothe	GreatLakes@NTug.com
Michigan Windows NT User Group	None		Southfield, Michigan	Unknown	Donald Barry	dbarry@wisne.com
Minneapolis Windows NT Users Group	http://mnnt1.hep.umn.edu/ntug/ntug.cgi		Minneapolis, Minnesota	Small	Henley Quadling	bleimeyer.paul@mayo.edu
Minnesota Windows NT/Back-Office User's Group	http://www.mnwinnt.org		Twin Cities, Minnesota	Large	Paul Bleimeyer	paulb@mayo.edu
Windows NT Users Group of St. Louis (WNTUG-STL)	http://www.directpt.com/wntug-stl		St. Louis, Missouri	Large	Bryan Muehlberger	bryan@directpoint.net
CSP	None		Kishibeu, Moldova	Small	Timur Fbucashvili	tt97702@usm.md
Windows NT The Netherlands (TADIS)	None		Leusden, Netherlands	Small	A.P. Meeuwsen	tadis@Compuserve.com
Las Vegas Windows NT Users Group	http://www.lv.vitrex.com/lvwntug/default.htm		Las Vegas, Nevada	Medium	Kenneth Pribble	ken.pribble@vitrex.com
NT Support	None		Las Vegas, Nevada	Unknown	Nigel Sampson	Nigel@therio.net
New Jersey Windows NT User Group	http://www.njwindows.org		Edison, New Jersey	Medium	David J. Straley	djs@sandyhook.com
The Mad Scientist Club	None		Warren, New Jersey	Small	Frank A. Del Buono	fdelbuono@hiserv-na.com
New Jersey PC User Group—NT-SIG	http://www.njpcug.org		Wyckoff, New Jersey	Medium	Terry P. Gustafson	terryg@warwick.net
Developer SIG	None		Unknown, somewhere in New Jersey	Unknown	Unknown	djs@cnj.digex.net
Monkeys	None		Albuquerque, New Mexico	Small	Pedro Morales	Pmorales@bernco.gov
Albuquerque Windows NT User Group	http://mack.rt66.com/jheald/ntuser.htm		Albuquerque, New Mexico	Small	Jerry Heald	jheald@RT66.com
Capital District NT/Backoffice Users Group	None		Albany, New York	Small	Cindy Hermann	hermann@taconic.net
Westchester County NT User Group	None		Elmsford, New York	Unknown	Dinesh Ganesh	sherrydin@yahoo.com

New York Developers SIG Only		http://budman.cmdl.noaa.gov/rmwntug/user_groups/nyntsig3.htm	New York, New York	Unknown	Check with the group's Web page	Check the group's Web page
New York Windows NT User Group	None		New York, New York	Large	Bill Zack	wzack@compuserve.com
New York City Developers SIG	None		New York, New York	Unknown	Unknown	lee_t@access.digex.net
New York City Windows NT Users Group	None		New York, New York	Large	John Rhodes	JRhodes@bbn.com
Rochester/Monroe County Library User Group	None		Rochester, New York	Small	David Deaugustine	davidd@mcls.rochester.lib.ny.us
SI NT Admin	None		Staten Island, New York	Unknown	Craig Caggiano	crc61@bigfoot.com
Long Island NT Users Group	None		Uniondale, New York	Unknown	Gerald Shamberger	Gerald.Shamberger@uscoopers.com
NZNTUG	None		New Zealand	Unknown	Nathan Mercer	nathan@MCS.co.nz
Taranaki NT Users Group	None		New Plymouth, New Zealand	Unknown	Chris Sharpe	chris.sharpe@computerland.co.nz
NCNT	None		Charlotte, North Carolina	Unknown	Stephen Carabetta	vianet@ix.netcom.com
Piedmont Triad NT Users Group		http://www.ptntug.org	Greensboro, North Carolina	Medium	Dr. Bill Bailey	baileyb@ptntug.org
TNTUG—Triangle NT Users Group	None		Raleigh, North Carolina	Large	John McMains	tntug@networks.com
Red River Windows NT Users Group	None		Grand Forks, North Dakota	Medium	Roy Beard	rbeard@plains.nodak.edu
MANUS—Microsoft Advanced Network Users Society	None		Norway	Large	Kay Seljeseth	Kay.Seljeseth@Cinet.No
BACUP—Backoffice Atlantic Canada Users Platform		http://www.bacup.com	Nova Scotia	Medium	John Gauthier	gauthier.john@mit-it.com
COUNT, Central Ohio Users NT	None		Columbus, Ohio	Medium	Ed Zirkle	Zirkle.5@osu.edu
Dayton NT Users Group (DAYNTUG)	None		Dayton, Ohio	Medium	Chris T Haaker	chris.haaker@stdreg.com
NT in Manufacturing Automation Users	None		Findlay, Ohio	Unknown	Bill Wagner	BillW@FESTech.com
Philadelphia Back-Office Users Group (PBOG)	None		Philadelphia, Pennsylvania	Medium	Bill Wolff, Mike Ward	bill@wolffdata.com, Mike_Ward@onlc.com
Pittsburgh Area Windows NT User Group (PAWNTUG)		http://www.pghnt.org	Pittsburgh, Pennsylvania	Medium	Robert Smith Jr., Avram Cheaney	info@pghnt.org
Print & Publishing in Poland		http://arra.com.pl	Kalisz, Poland	Medium	Roman Lewicki	Webmaster@arra.com.pl
Media Print & Publishing in Poland—Inwestycje, Marketin, Media		http://www.arra.com.pl	Kalisz, Poland	Small	Roman Lewicki	Webmaster@arra.com.pl

Mastering Windows 2000 Registry: Where Can I Get More Help?

Polish Users of Windows NT 4.0 Server	None	Poznan, Poland	Medium	Tomasz Waslowicz	tomaszwa@free.polbox.pl
Windows NT Group, Portugal (WNTGP)	None	Portugal	Unknown	Luis Centeio	luisc@poboxes.com
PRUNT (Puerto Rico NT User Group)	None	Toa Alta, Puerto Rico	Unknown	Richard Arroyo	rarroyo@prtc.net
Rhode Island NT Users Group	None	Coventry, Rhode Island	Unknown	Ernie Quaglieri	celebty@concentric.net
SneNUG Southern New England Network Users Group	None	East Providence, Rhode Island	Large	Bill Dwyer	billd@loa.com
Russian Windows NT User Group	None	Moscow, Russian Federation	Unknown	Konstantin Gusev	gusevk@quarta.com
Cape Microsoft User Group	None	Cape Town, South Africa	Unknown	Gordon Thelander	enigmax@iafrica.com
grupo de usuarios de nt espaïoles	None	Valencia, Spain	Unknown	Isaac Jaramillo	ijs@mx3.redestb.es
NT Anvndargruppen	None	Stockholm, Sweden	Unknown	Friberg	iir@iir.telegate.se
Swiss NT Users Group	None	Switzerland	Unknown	Deffer@Eunet.Ch	adeffer@eunet.ch
CERN Windows NT Users Group	None	Geneva, Switzerland	Unknown	Alberto AIMAR	alberto.aimar@cern.ch
NT Syria by CompuCrest	None	Damascus, Syria	Small	aA. Aziz	maziz@usa.net
Chattanooga-River Valley NT Users Group	None	Chattanooga, Tennessee	Unknown	James	james@press.southern.edu
NT4_Users Group	None	Martin, Tennessee	Large	J. Garner	jgarner@utm.edu
Microsoft Networking SIG	http://budman.cmdl.noaa.gov/rmwntug/user_groups/dlsugan.htm	Tennessee	Unknown	Deborah	deborah@microsoft.com
Capital Area NT User Group	None	Austin, Texas	Unknown	Perry Stokes	stokes@jump.net
DFW NT Users Group	None	Dallas/Ft. Worth, Texas	Unknown	Ralph Shumway	rshumway@swbell.net
Dallas BackOffice Users Group (DeBUG)	http://www.debug.org/	Dallas, Texas	Medium	Mark Saum	info@debug.org
North Texas NT Users Group (NTsquared)	Dallas, Texas	Large	Charles Reiss	charlier@unicomp.net	
The El Paso & Las Cruces Windows NT User Group	None	El Paso, Texas	Medium	Shane A. Weddle	sweddle@worldnet.att.net
Fort Worth NT User Group	None	Fort Worth, Texas	Unknown	Paul Knox	paulk@netaci.com
Houston Area NT Users Group	None	Houston, Texas	Unknown	Arthur Kettelhut	atkette@ix.netcom.com

Houston Microsoft User Group	None	Houston, Texas	Medium	Alisa Wanger	alisaw@infotecweb.com
NT TECH PARTY	None	Houston, Texas	Unknown	Hiphop	hiphopluva@hotmail.com
aPlano BackOffice User Group (PBUG)	None	Plano, Texas	Medium	Marcia Loughry	mloughry@cyberramp.net
Alamo PC Organization	None	San Antonio, Texas	Medium	Larry Lentz	Larry@LentzComputer.com
Windows NT SIG					
Rio Grande Valley Area IT Group	None	Weslaco, Texas	Unknown	Cindy Barber	cabarber@usa.net
ThaiNTprimer	None	Bangkok, Thailand	Unknown	Siramet	siramet@yth.co.th
TURKEY NT USER GROUP	None	Istanbul, Turkey	Small	Ozan Zkara	OZAN.OZKARA@USA.NET
Wyndows NT GRUBU	None	Istanbul, Turkey	Small	Harun Aksoz	harunaksoz@hotmail.com.tr
TURKWNT	http://www.turkwnt.org	Izmir, Turkey	Medium	E.Onder Kokturk	kokturk@bornova.ege.edu.tr
Emirates NT User Group	None	Dubai, United Arab Emirates	Small	Rizwan Ahmed Khan	rizwan@emirates.net.ae
Microsoft Windows NT & BackOffice Forum	None	London, United Kingdom	Large	Tony Larks	tony_larks@researchgroup.co.uk
The Microsoft Windows (NT) & BackOffice Forum (UK)	http://www.researchgroup.co.uk	London, United Kingdom	Large	Simon Moores	smoores@softech.co.uk
Wasatch Windows NT User Group	http://www.wwntug.org	Salt Lake, Utah	Medium	Steve Adams	sadams@acs.utah.edu
Vermont NT Users Group (VTNTUG)	None		Medium	Michael Gambler	mgambler@sover.net
Central Virginia NT Users Group	http://www.harbour.net/ntgroup/	Richmond, Virginia	Small	Anthony S. Harbour, Med	aharbour@harbour.net
BackOffice Professionals Association	None	Seattle, Washington	Large	Marjorie James	backoffice@ariscorp.com
Pacific NorthWest NT User/Small Business Group	http://www.ntug.org	Tacoma, Washington	Small	Chris Melton	info@ntug.org
Wisconsin NT Users Group	None	Brookfield, Wisconsin	Medium	Bob Escher	bescher@wintug.org
NT User Group	None	Belgrade, Yugoslavia	Large	Milan Zivkovic	zivkovic@internetplus.ch
BackOffice User Group	http://www.elcoma.ch/bousgroup/	Belgrade, Yugoslavia	Large	Milan Zivkovic	mzivkovic@elcoma.ch

Virtual Support Groups: ClubWin

There are a number of organizations dedicated to assisting users of Microsoft products. ClubWin is an important one of these. This group is part of Microsoft's marketing arm and is made up of volunteer, non-Microsoft users who have demonstrated a great deal of skill in their fields.

It is a self-organized group of computer professionals and advanced users from the U.S., Canada, and Europe, who have hundreds of thousands of hours of experience with Microsoft Windows 95 and Windows 98. The group is made up of hardware, software, applications, networking, and support professionals who have a commitment to providing information and support for others using Windows 95/98. ClubWin has joined forces to cover major online services (such as MSN, CompuServe, America Online, Prodigy, and others) and the World Wide

Web. The ClubWin designation on a Web page, or in a tagline or signature, designates a person who has made a commitment to the organization.

I am a member of ClubWin. The group is an excellent source of help and information. Their Web page is located at <http://www.clubwin.com>.

Training

Several companies offer training on registry-related maintenance. Contact Data-Tech Institute at <http://www.datatech.com> for more information on their courses.

Another source of training is your local college or university. For example, Boston University offers a number of excellent training courses for Windows professionals, mostly in the Boston, Massachusetts area.

Generally, conferences offer some useful training opportunities. Most conferences advertise using direct mail.

Other Internet Information Sources

The Internet offers many areas of support and assistance, including newsgroups and chat (IRC) groups. Both of these resources can be unexpected sources of valuable advice and information.

Newsgroups

One source of information on the Internet is newsgroups. The snews.microsoft.com site has all of the Microsoft-supported newsgroups. There are over 700 different groups on this news server, and between 40 and 50 groups are dedicated to various versions of Windows NT. In addition, there are a number of programming groups oriented toward Windows NT platforms from which valuable information may be gleaned.

If you are unfamiliar with using newsgroups, you need to have a newsgroup reader. There are many free or low-cost newsreaders. For example, Microsoft's Outlook Express has a built-in newsreader. In addition, FortZ offers a product called FreeAgent; see their licensing agreement to know if you qualify for a free version. You can find FortZ on the Web at <http://www.forteinc.com/agent/>.

There are well-established protocols for posting in a newsgroup. It is generally best if you simply read messages (lurk) for a few days to get the feel of things. *Lurking* is the term for a person who reads, but does not post, in a newsgroup. Then you will have a feel for how to ask questions. Follow these simple guidelines when posting:

- Don't post "off topic" messages. If you are in the Windows NT Setup group, don't post a message about Exchange.
- Never, ever post a "get rich quick" message.
- Make the title of your message or topic informative and catchy. You want the right people to read your question. Don't title your message "Help Me!" Say something like, "System Hangs with BSOD after Installing ASDF App."
- Though your message will contain your email address (and you should include your name and email in the message), don't expect anyone to email you a solution. They will post the reply in the newsgroup, so check there. It's bad form to ask for email replies.
- To avoid getting spam (unsolicited, mass email marketing messages), it is acceptable to subtly change your email address, both in the program and in the message signature. Many users add something to their email address that somewhat obviously doesn't belong there; for example, newsgroup members would recognize that to send e-mail to me, they'd have to alter phipson@nospam.acm.org by deleting the `nospam.` part of the address.
- Last, but certainly not least, check the newsgroup's previous messages before posting your problem. It is quite possible that someone else has had the same problem recently and you can benefit from responses to that person. People get tired of answering the same question repeatedly.

Chat

Chat (IRC) is a well-used part of the Internet. Chat sessions can sometimes be useful, especially if you get onto the right chat server. The best chat server for Windows 2000 users is accessed infrequently, but it can be helpful. Figure C.1 shows the chats that were available to users early one Saturday morning. There were only 11 users online with this server, guaranteeing good performance.

OK, this is secret, very hidden, never discussed information: The URL for Microsoft's BackOffice chat server is chat.backoffice.microsoft.com. Got that? Virtually no one knows about this chat server.



TIP Microsoft is in the habit of changing their sites on an almost daily basis. Now, this has good effects, but sometimes we find that things move or simply disappear! If you can't find the server, search for it, or browse the Microsoft BackOffice site.

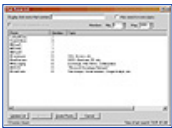


FIGURE C.1 This chat room list shows how little used the Microsoft BackOffice chat server really is.

One interesting thing about chat servers is that if there is not a chat covering the topic you are interested in, you can start your own chat session. Microsoft also offers an ActiveX control to access chat groups. It is available at the Microsoft BackOffice Web site. The best chat client is Microsoft's Comic Chat. Go to Microsoft's Web site at <http://www.microsoft.com/windows/ie/chat/> and download the latest version of Comic Chat.



NOTE Like those classy Comic Chat characters? If so, you can actually create a personalized character for your identity. Use the Comic Chat character editor, which is available at the same Web address as Comic Chat.

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

APPENDIX D

Performance Counters

Table D.1 shows the performance monitor counters for a typical Windows 2000 installation. Each counter is stored in two registry keys—one for the counter name, the other for the help description. Actually, Windows 2000 (and all versions of Windows NT after NT 3.51) stores all of the performance counters in two files, and the system accesses these files as if they were part of the registry. These two files are:

perfc009.dat: Contains all the counter names. Items in this file display either as counters or as performance objects.

perfh009.dat: Contains the corresponding help text (displayed when the user requests help) for each item in the list of counters and performance objects.

Figure D.1 shows the Add Counters dialog box with the System performance object selected and one of its counters highlighted. Help text for the highlighted counter is displayed in the bottom of the dialog box.



FIGURE D.1 The Add Counters dialog box allows the user to specify performance objects and counters.

Counter names are stored in the REG_MULTI_SZ data object at HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib\009\Counter, which through the magic of the registry is actually stored in the file perfc009.dat.



NOTE I've noticed that RegEdit has some difficulty with the Counter and Help subkeys (they are REG_MULTI_SZ, which RegEdit doesn't handle well). I suggest using RegEdt32 to edit and view these items.

A help description has an index number that is one greater than the number for the counter, or object, name. For example, for the % Processor Time counter (at index 6), the help text is at index 7. (Even though the two lists are separate, Microsoft uses different numbers for the counter name and the help text.) The counter

descriptions are stored in the REG_MULTI_SZ data object at

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib\009\Help, which is actually the file perfh009.dat.



NOTE Why a subkey named 009? Why do the performance monitor support files (perfc009.dat and perfh009.dat) have 009 in them? This is because the performance counters' names and descriptions are language specific. If your installation is not U.S. English (language ID 009), it is likely that your registry will have different subkey containing counters and descriptions in your language.

Table D.1 lists the main performance monitoring objects and the counters that these objects use. (Notice that some counters seem to be reused. Actually, it's only the descriptive text that is reused; each actual counter is stored in a unique location.)

In Table D.1, the Object column shows the index number for the performance monitoring object. For example, the first object, System, is at index 2. In the System object, there are an additional 17 counters, starting with index number 12. It is important to realize that counters are not stored consecutively. The Name column shows the text that the MMC Performance Monitor snap-in uses for counters and objects.

The Instance column shows whether the item is an object or a single- or multiple-instance counter. Some counters (such as the counter that monitors disk performance, which would have an instance for each drive) have multiple instances, while other counters (such as Processes) have only a single instance.



NOTE See Chapter 11 for more on using performance objects and counters, and for information on adding custom counters for your applications and systems. All of the object and counter index numbers mean little at this stage. The (hidden) registry hive HKEY_PERFORMANCE_DATA contains the necessary information to allow an application to successfully interact with and display performance data.

Table D.1: Performance Counters on a Windows 2000 Server

Object	Counter	Name	Instance
2		System	Object
	12	File Write Operations/sec	Single
	14	File Control Operations/sec	Single
	16	File Read Bytes/sec	Single
	18	File Write Bytes/sec	Single
	20	File Control Bytes/sec	Single
	44	Processor Queue Length	Single
	146	Context Switches/sec	Single
	150	System Calls/sec	Single
	248	Processes	Single
	250	Threads	Single
	406	File Data Operations/sec	Single
	674	System Up Time	Single
	686	Alignment Fixups/sec	Single
	688	Exception Dispatches/sec	Single
	690	Floating Emulations/sec	Single
	1350	% Registry Quota In Use	Single
	1350	% Registry Quota In Use	Single
4		Memory	Object
	24	Available Bytes	Single
	26	Committed Bytes	Single
	28	Page Faults/sec	Single
	30	Commit Limit	Single
	32	Write Copies/sec	Single
	34	Transition Faults/sec	Single

	36	Cache Faults/sec	Single
	38	Demand Zero Faults/sec	Single
	40	Pages/sec	Single
	42	Page Reads/sec	Single
	48	Pages Output/sec	Single
	50	Page Writes/sec	Single
	56	Pool Paged Bytes	Single
	58	Pool Non-paged Bytes	Single
	60	Pool Paged Allocations	Single
	64	Pool Non-paged Allocations	Single
	66	Pool Paged Resident Bytes	Single
	68	System Code Total Bytes	Single
	70	System Code Resident Bytes	Single
	72	System Driver Total Bytes	Single
	74	System Driver Resident Bytes	Single
	76	System Cache Resident Bytes	Single
	678	Free System Page Table Entries	Single
	818	Cache Bytes	Single
	820	Cache Bytes Peak	Single
	822	Pages Input/sec	Single
	1406	% Committed Bytes In Use	Single
	1406	% Committed Bytes In Use	Single
52		Browser	Object
	54	Announcements Server/sec	Single
	78	Announcements Domain/sec	Single
	80	Election Packets/sec	Single
	82	Mailslot Writes/sec	Single
	84	Server List Requests/sec	Single
	156	Enumerations Server/sec	Single
	158	Enumerations Domain/sec	Single
	160	Enumerations Other/sec	Single
	162	Missed Server Announcements	Single
	164	Missed Mailslot Datagrams	Single
	166	Missed Server List Requests	Single
	168	Server Announce Allocations Failed/sec	Single
	170	Mailslot Allocations Failed	Single
	188	Announcements Total/sec	Single
	190	Enumerations Total/sec	Single
	806	Mailslot Receives Failed	Single
	808	Mailslot Writes Failed	Single
	810	Mailslot Opens Failed/sec	Single
	812	Duplicate Master Announcements	Single
	814	Illegal Datagrams/sec	Single
86		Cache	Object
	88	Data Maps/sec	Single
	90	Sync Data Maps/sec	Single
	92	Asynchronous Data Maps/sec	Single
	94	Data Map Hits %	Single
	94	Data Map Hits %	Single
	96	Data Map Pins/sec	Single

	96	Data Map Pins/sec	Single
	98	Pin Reads/sec	Single
	100	Sync Pin Reads/sec	Single
	102	Asynchronous Pin Reads/sec	Single
	104	Pin Read Hits %	Single
	104	Pin Read Hits %	Single
	106	Copy Reads/sec	Single
	108	Sync Copy Reads/sec	Single
	110	Asynchronous Copy Reads/sec	Single
	112	Copy Read Hits %	Single
	112	Copy Read Hits %	Single
	114	MDL Reads/sec	Single
	116	Sync MDL Reads/sec	Single
	118	Asynchronous MDL Reads/sec	Single
	120	MDL Read Hits %	Single
	120	MDL Read Hits %	Single
	122	Read Aheads/sec	Single
	124	Fast Reads/sec	Single
	126	Sync Fast Reads/sec	Single
	128	Asynchronous Fast Reads/sec	Single
	130	Fast Read Resource Misses/sec	Single
	132	Fast Read Not Possible/sec	Single
	134	Lazy Write Flushes/sec	Single
	136	Lazy Write Pages/sec	Single
	138	Data Flushes/sec	Single
	140	Data Flush Pages/sec	Single
230		Process	Object
	6	% Processor Time	Multiple
	28	Page Faults/sec	Multiple
	56	Pool Paged Bytes	Multiple
	58	Pool Non-paged Bytes	Multiple
	142	% User Time	Multiple
	144	% Privileged Time	Multiple
	172	Virtual Bytes Peak	Multiple
	174	Virtual Bytes	Multiple
	178	Working Set Peak	Multiple
	180	Working Set	Multiple
	182	Page File Bytes Peak	Multiple
	184	Page File Bytes	Multiple
	186	Private Bytes	Multiple
	680	Thread Count	Multiple
	682	Priority Base	Multiple
	684	Elapsed Time	Multiple
	784	ID Process	Multiple
	952	Handle Count	Multiple
	1410	Creating Process ID	Multiple
	1412	IO Read Operations/sec	Multiple
	1414	IO Write Operations/sec	Multiple
	1416	IO Data Operations/sec	Multiple
	1418	IO Other Operations/sec	Multiple

	1420	IO Read Bytes/sec	Multiple
	1422	IO Write Bytes/sec	Multiple
	1424	IO Data Bytes/sec	Multiple
	1426	IO Other Bytes/sec	Multiple
232		Thread	Object
	6	% Processor Time	Multiple
	46	Thread State	Multiple
	142	% User Time	Multiple
	144	% Privileged Time	Multiple
	146	Context Switches/sec	Multiple
	336	Thread Wait Reason	Multiple
	682	Priority Base	Multiple
	684	Elapsed Time	Multiple
	694	Priority Current	Multiple
	706	Start Address	Multiple
	784	ID Process	Multiple
	804	ID Thread	Multiple
234		Physical Disk	Object
	198	Current Disk Queue Length	Multiple
	200	% Disk Time	Multiple
	200	% Disk Time	Multiple
	202	% Disk Read Time	Multiple
	202	% Disk Read Time	Multiple
	204	% Disk Write Time	Multiple
	204	% Disk Write Time	Multiple
	206	Average. Disk sec/Transfer	Multiple
	206	Average. Disk sec/Transfer	Multiple
	208	Average. Disk sec/Read	Multiple
	208	Average. Disk sec/Read	Multiple
	210	Average. Disk sec/Write	Multiple
	210	Average. Disk sec/Write	Multiple
	212	Disk Transfers/sec	Multiple
	214	Disk Reads/sec	Multiple
	216	Disk Writes/sec	Multiple
	218	Disk Bytes/sec	Multiple
	220	Disk Read Bytes/sec	Multiple
	222	Disk Write Bytes/sec	Multiple
	224	Average. Disk Bytes/Transfer	Multiple
	224	Average. Disk Bytes/Transfer	Multiple
	226	Average. Disk Bytes/Read	Multiple
	226	Average. Disk Bytes/Read	Multiple
	228	Average. Disk Bytes/Write	Multiple
	228	Average. Disk Bytes/Write	Multiple
	1400	Average. Disk Queue Length	Multiple
	1402	Average. Disk Read Queue Length	Multiple
	1404	Average. Disk Write Queue Length	Multiple
	1482	% Idle Time	Multiple
	1482	% Idle Time	Multiple
	1484	Split IO/Sec	Multiple
238		Processor	Object

	6	% Processor Time	Multiple
	142	% User Time	Multiple
	144	% Privileged Time	Multiple
	148	Interrupts/sec	Multiple
	696	% DPC Time	Multiple
	698	% Interrupt Time	Multiple
	1334	DPCs Queued/sec	Multiple
	1336	DPC Rate	Multiple
	1338	DPC Bypasses/sec	Multiple
	1340	APC Bypasses/sec	Multiple
260		Objects	Object
	248	Processes	Single
	250	Threads	Single
	252	Events	Single
	254	Semaphores	Single
	256	Mutexes	Single
	258	Sections	Single
262		Re-Director	Object
	10	File Read Operations/sec	Single
	12	File Write Operations/sec	Single
	264	Bytes Received/sec	Single
	266	Packets Received/sec	Single
	268	Read Bytes Paging/sec	Single
	270	Read Bytes Non-Paging/sec	Single
	272	Read Bytes Cache/sec	Single
	274	Read Bytes Network/sec	Single
	276	Bytes Transmitted/sec	Single
	278	Packets Transmitted/sec	Single
	280	Write Bytes Paging/sec	Single
	282	Write Bytes Non-Paging/sec	Single
	284	Write Bytes Cache/sec	Single
	286	Write Bytes Network/sec	Single
	290	Read Operations Random/sec	Single
	292	Read Packets/sec	Single
	294	Reads Large/sec	Single
	296	Read Packets Small/sec	Single
	300	Write Operations Random/sec	Single
	302	Write Packets/sec	Single
	304	Writes Large/sec	Single
	306	Write Packets Small/sec	Single
	308	Reads Denied/sec	Single
	310	Writes Denied/sec	Single
	312	Network Errors/sec	Single
	314	Server Sessions	Single
	316	Server Reconnects	Single
	318	Connects Core	Single
	320	Connects LAN Manager 2.0	Single
	322	Connects LAN Manager 2.1	Single
	324	Connects Windows NT	Single
	326	Server Disconnects	Single

	328	Server Sessions Hung	Single
	388	Bytes Total/sec	Single
	392	Current Commands	Single
	400	Packets/sec	Single
	406	File Data Operations/sec	Single
330		Server	Object
	56	Pool Paged Bytes	Single
	58	Pool Non-paged Bytes	Single
	264	Bytes Received/sec	Single
	276	Bytes Transmitted/sec	Single
	314	Server Sessions	Single
	340	Sessions Timed Out	Single
	342	Sessions Errored Out	Single
	344	Sessions Logged Off	Single
	346	Sessions Forced Off	Single
	348	Errors Logon	Single
	350	Errors Access Permissions	Single
	352	Errors Granted Access	Single
	354	Errors System	Single
	356	Blocking Requests Rejected	Single
	358	Work Item Shortages	Single
	360	Files Opened Total	Single
	362	Files Open	Single
	366	File Directory Searches	Single
	370	Pool Non-paged Failures	Single
	372	Pool Non-paged Peak	Single
	376	Pool Paged Failures	Single
	378	Pool Paged Peak	Single
	388	Bytes Total/sec	Single
	404	Context Blocks Queued/sec	Single
	692	Logon/sec	Single
	1260	Logon Total	Single
502		NTB Connection	Object
	264	Bytes Received/sec	Multiple
	388	Bytes Total/sec	Multiple
	506	Bytes Sent/sec	Multiple
510		Network Interface	Object
	264	Bytes Received/sec	Multiple
	266	Packets Received/sec	Multiple
	388	Bytes Total/sec	Multiple
	400	Packets/sec	Multiple
	452	Packets Sent/sec	Multiple
	506	Bytes Sent/sec	Multiple
	520	Current Bandwidth	Multiple
	524	Packets Received Unicast/sec	Multiple
	526	Packets Received Non-Unicast/sec	Multiple
	528	Packets Received Discarded	Multiple
	530	Packets Received Errors	Multiple
	532	Packets Received Unknown	Multiple
	536	Packets Sent Unicast/sec	Multiple

	538	Packets Sent Non-Unicast/sec	Multiple
	540	Packets Outbound Discarded	Multiple
	542	Packets Outbound Errors	Multiple
	544	Output Queue Length	Multiple
546		IP	Object
	438	Datagrams/sec	Single
	442	Datagrams Sent/sec	Single
	446	Datagrams Received/sec	Single
	552	Datagrams Received Header Errors	Single
	554	Datagrams Received Address Errors	Single
	556	Datagrams Forwarded/sec	Single
	558	Datagrams Received Unknown Protocol	Single
	560	Datagrams Received Discarded	Single
	562	Datagrams Received Delivered/sec	Single
	566	Datagrams Outbound Discarded	Single
	568	Datagrams Outbound No Route	Single
	570	Fragments Received/sec	Single
	572	Fragments Re-assembled/sec	Single
	574	Fragment Re-assembly Failures	Single
	576	Fragmented Datagrams/sec	Single
	578	Fragmentation Failures	Single
	580	Fragments Created/sec	Single
582		ICMP	Object
	584	Messages/sec	Single
	586	Messages Received/sec	Single
	588	Messages Received Errors	Single
	590	Received Destination Unreachable	Single
	592	Received Time Exceeded	Single
	594	Received Parameter Problem	Single
	596	Received Source Quench	Single
	598	Received Redirect/sec	Single
	600	Received Echo/sec	Single
	602	Received Echo Reply/sec	Single
	604	Received Timestamp/sec	Single
	606	Received Timestamp Reply/sec	Single
	608	Received Address Mask	Single
	610	Received Address Mask Reply	Single
	612	Messages Sent/sec	Single
	614	Messages Outbound Errors	Single
	616	Sent Destination Unreachable	Single
	618	Sent Time Exceeded	Single
	620	Sent Parameter Problem	Single
	622	Sent Source Quench	Single
	624	Sent Redirect/sec	Single
	626	Sent Echo/sec	Single
	628	Sent Echo Reply/sec	Single
	630	Sent Timestamp/sec	Single
	632	Sent Timestamp Reply/sec	Single
	634	Sent Address Mask	Single
	636	Sent Address Mask Reply	Single

638	TCP	Object
640	Segments/sec	Single
642	Connections Established	Single
644	Connections Active	Single
646	Connections Passive	Single
648	Connection Failures	Single
650	Connections Reset	Single
652	Segments Received/sec	Single
654	Segments Sent/sec	Single
656	Segments Retransmitted/sec	Single
658	UDP	Object
438	Datagrams/sec	Single
442	Datagrams Sent/sec	Single
446	Datagrams Received/sec	Single
664	Datagrams No Port/sec	Single
666	Datagrams Received Errors	Single
700	Paging File	Object
702	% Usage	Multiple
702	% Usage	Multiple
704	% Usage Peak	Multiple
704	% Usage Peak	Multiple
870	RAS Port	Object
872	Bytes Transmitted	Multiple
874	Bytes Received	Multiple
876	Frames Transmitted	Multiple
878	Frames Received.	Multiple
880	Percent Compression Out	Multiple
882	Percent Compression In	Multiple
884	CRC Errors	Multiple
886	Timeout Errors	Multiple
888	Serial Overrun Errors	Multiple
890	Alignment Errors	Multiple
892	Buffer Overrun Errors	Multiple
894	Total Errors	Multiple
896	Bytes Transmitted/Sec	Multiple
898	Bytes Received/Sec	Multiple
900	Frames Transmitted/Sec	Multiple
902	Frames Received/Sec	Multiple
904	Total Errors/Sec	Multiple
906	RAS Total	Object
872	Bytes Transmitted	Single
874	Bytes Received	Single
876	Frames Transmitted	Single
878	Frames Received.	Single
880	Percent Compression Out	Single
882	Percent Compression In	Single
884	CRC Errors	Single
886	Timeout Errors	Single
888	Serial Overrun Errors	Single
890	Alignment Errors	Single

	892	Buffer Overrun Errors	Single
	894	Total Errors	Single
	896	Bytes Transmitted/Sec	Single
	898	Bytes Received/Sec	Single
	900	Frames Transmitted/Sec	Single
	902	Frames Received/Sec	Single
	904	Total Errors/Sec	Single
	908	Total Connections	Single
920		WINS Server	Object
	922	Unique Registrations/sec	Single
	924	Group Registrations/sec	Single
	926	Total Number of Registrations/sec	Single
	928	Unique Renewals/sec	Single
	930	Group Renewals/sec	Single
	932	Total Number of Renewals/sec	Single
	934	Releases/sec	Single
	936	Queries/sec	Single
	938	Unique Conflicts/sec	Single
	940	Group Conflicts/sec	Single
	942	Total Number of Conflicts/sec	Single
	944	Successful Releases/sec	Single
	946	Failed Releases/sec	Single
	948	Successful Queries/sec	Single
	950	Failed Queries/sec	Single
1150		Telephony	Object
	1152	Lines	Single
	1154	Telephone Devices	Single
	1156	Active Lines	Single
	1158	Active Telephones	Single
	1160	Outgoing Calls/sec	Single
	1162	Incoming Calls/sec	Single
	1164	Client Apps	Single
	1166	Current Outgoing Calls	Single
	1168	Current Incoming Calls	Single
1300		Server Work Queues	Object
	264	Bytes Received/sec	Multiple
	288	Read Operations/sec	Multiple
	298	Write Operations/sec	Multiple
	404	Context Blocks Queued/sec	Multiple
	506	Bytes Sent/sec	Multiple
	508	Total Bytes/sec	Multiple
	1302	Queue Length	Multiple
	1304	Active Threads	Multiple
	1306	Available Threads	Multiple
	1308	Available Work Items	Multiple
	1310	Borrowed Work Items	Multiple
	1312	Work Item Shortages	Multiple
	1314	Current Clients	Multiple
	1320	Bytes Transferred/sec	Multiple
	1324	Read Bytes/sec	Multiple

	1328	Write Bytes/sec	Multiple
	1332	Total Operations/sec	Multiple
1450		Print Queue	Object
	1452	Total Jobs Printed	Multiple
	1454	Bytes Printed/sec	Multiple
	1456	Total Pages Printed	Multiple
	1458	Jobs	Multiple
	1460	References	Multiple
	1462	Max References	Multiple
	1464	Jobs Spooling	Multiple
	1466	Max Jobs Spooling	Multiple
	1468	Out of Paper Errors	Multiple
	1470	Not Ready Errors	Multiple
	1472	Job Errors	Multiple
	1474	Enumerate Network Printer Calls	Multiple
	1476	Add Network Printer Calls	Multiple
1500		Job Object	Object
	1502	Current % Processor Time	Single
	1504	Current % User Mode Time	Single
	1506	Current % Kernel Mode Time	Single
	1508	This Period mSec - Processor	Single
	1510	This Period mSec - User Mode	Single
	1512	This Period mSec - Kernel Mode	Single
	1514	Pages/Sec	Single
	1516	Process Count - Total	Single
	1518	Process Count - Active	Single
	1520	Process Count - Terminated	Single
	1522	Total mSec - Processor	Single
	1524	Total mSec - User Mode	Single
	1526	Total mSec - Kernel Mode	Single
1548		Job Object Details	Object
	6	% Processor Time	Single
	28	Page Faults/sec	Single
	56	Pool Paged Bytes	Single
	58	Pool Non-paged Bytes	Single
	142	% User Time	Single
	144	% Privileged Time	Single
	172	Virtual Bytes Peak	Single
	174	Virtual Bytes	Single
	178	Working Set Peak	Single
	180	Working Set	Single
	182	Page File Bytes Peak	Single
	184	Page File Bytes	Single
	186	Private Bytes	Single
	680	Thread Count	Single
	682	Priority Base	Single
	684	Elapsed Time	Single
	784	ID Process	Single
	952	Handle Count	Single
	1410	Creating Process ID	Single

	1412	IO Read Operations/sec	Single
	1414	IO Write Operations/sec	Single
	1416	IO Data Operations/sec	Single
	1418	IO Other Operations/sec	Single
	1420	IO Read Bytes/sec	Single
	1422	IO Write Bytes/sec	Single
	1424	IO Data Bytes/sec	Single
	1426	IO Other Bytes/sec	Single
1848		ACS/RSVP Service	Object
	1850	Interfaces	Multiple
	1852	Network sockets	Multiple
	1854	Timers	Multiple
	1856	RSVP message buffers in use	Multiple
	1858	GQOS sessions	Multiple
	1860	API sockets	Multiple
	1862	PATH from API	Multiple
	1864	RESV from API	Multiple
	1866	Failed API requests	Multiple
	1868	Failed API sends	Multiple
	1870	API notifications	Multiple
2398		Messages	Object
	2400	Messages Delivered Total	Multiple
	2402	Messages Delivered/sec	Multiple
	2404	Message Delivery Retries	Multiple
	2406	Average Retries/message Delivered	Multiple
	2408	Base Average Retries/message Delivered	Multiple
	2410	Pickup Directory Messages Retrieved Total	Multiple
	2412	Pickup Directory Messages Retrieved/sec	Multiple
	2414	NDRs Generated	Multiple
	2416	Local Queue Length	Multiple
	2418	Local Retry Queue Length	Multiple
	2420	Number of MailFiles Open	Multiple
	2422	Number of QueueFiles Open	Multiple
	2424	Categorizer Queue Length	Multiple
	2426	Messages Sent Total	Multiple
	2428	Messages Sent/sec	Multiple
	2430	Message Send Retries	Multiple
	2432	Average Retries/message Sent	Multiple
	2434	Base Average Retries/message Sent	Multiple
	2436	Average Recipients/message Sent	Multiple
	2438	Base Average Recipients/message Sent	Multiple
	2440	Remote Queue Length	Multiple
	2442	DNS Queries Total	Multiple
	2444	DNS Queries/sec	Multiple
	2446	Remote Retry Queue Length	Multiple
	2450	Inbound Connections Total	Multiple
	2452	Inbound Connections Current	Multiple
	2454	Outbound Connections Total	Multiple
	2456	Outbound Connections Current	Multiple
	2458	Outbound Connections Refused	Multiple

2460	Total Connection Errors	Multiple
2462	Connection Errors/sec	Multiple
2464	Directory Drops Total	Multiple
2466	Directory Drops/sec	Multiple
2468	Routing Table Lookups Total	Multiple
2470	Routing Table Lookups/sec	Multiple
2472	ETRN Messages Total	Multiple
2474	ETRN Messages/sec	Multiple
2476	Badmailed Messages (No Recipients)	Multiple
2478	Badmailed Messages (Hop Count Exceeded)	Multiple
2480	Badmailed Messages (General Failure)	Multiple
2482	Badmailed Messages (Bad Pickup File)	Multiple
2484	Badmailed Messages (Triggered via Event)	Multiple
2486	Badmailed Messages (NDR of DSN)	Multiple
2488	Messages Pending Routing	Multiple
2490	Messages Currently Undeliverable	Multiple
2492	Total messages submitted	Multiple
2496	Current Messages in Local Delivery	Multiple
2498	Cat: Messages submitted	Multiple
2500	Cat: Messages submitted/sec	Multiple
2502	Cat: Categorizations completed	Multiple
2504	Cat: Categorizations completed/sec	Multiple
2506	Cat: Categorizations in progress	Multiple
2508	Cat: Categorizations completed successfully	Multiple
2510	Cat: Categorizations failed (non-retryable error)	Multiple
2512	Cat: Categorizations failed (retryable error)	Multiple
2514	Cat: Categorizations failed (Out Of Memory)	Multiple
2516	Cat: Categorizations failed (DS logon failure)	Multiple
2518	Cat: Categorizations failed (DS connection failure)	Multiple
2520	Cat: Categorizations failed (sink retryable error)	Multiple
2522	Cat: Messages Categorized	Multiple
2524	Cat: Messages bifurcated	Multiple
2526	Cat: Messages aborted	Multiple
2528	Cat: Recipients before categorization	Multiple
2530	Cat: Recipients after categorization	Multiple
2532	Cat: Recipients NDRd by categorizer	Multiple
2534	Cat: Recipients NDRd (unresolved)	Multiple
2536	Cat: Recipients NDRd (ambiguous address)	Multiple
2538	Cat: Recipients NDRd (illegal address)	Multiple
2540	Cat: Recipients NDRd (forwarding loop)	Multiple
2542	Cat: Recipients NDRd (sink recipient errors)	Multiple
2544	Cat: Recipients in categorization	Multiple
2546	Cat: Senders unresolved	Multiple
2548	Cat: Senders with ambiguous addresses	Multiple
2550	Cat: Address lookups	Multiple
2552	Cat: Address lookups/sec	Multiple
2554	Cat: Address lookup completions	Multiple
2556	Cat: Address lookup completions/sec	Multiple
2558	Cat: Address lookups not found	Multiple
2560	Cat: mailmessage duplicate collisions	Multiple

	2562	Cat: LDAP connections	Multiple
	2564	Cat: LDAP connection failures	Multiple
	2566	Cat: LDAP connections currently open	Multiple
	2568	Cat: LDAP binds	Multiple
	2570	Cat: LDAP bind failures	Multiple
	2572	Cat: LDAP searches	Multiple
	2574	Cat: LDAP searches/sec	Multiple
	2576	Cat: LDAP paged searches	Multiple
	2578	Cat: LDAP search failures	Multiple
	2580	Cat: LDAP paged search failures	Multiple
	2582	Cat: LDAP searches completed	Multiple
	2584	Cat: LDAP searches completed/sec	Multiple
	2586	Cat: LDAP paged searches completed	Multiple
	2588	Cat: LDAP search completion failures	Multiple
	2590	Cat: LDAP paged search completion failures	Multiple
	2592	Cat: LDAP general completion failures	Multiple
	2594	Cat: LDAP searches abandoned	Multiple
	2596	Cat: LDAP searches pending completion	Multiple
	2598	SMTP NTFS Store Driver	Multiple
	2600	Messages in the queue directory	Multiple
	2602	Messages allocated	Multiple
	2604	Messages deleted	Multiple
	2606	Messages enumerated	Multiple
	2608	Open message bodies	Multiple
	2610	Open message streams	Multiple
3264		DNS	Object
	3270	Total Query Received	Single
	3272	Total Query Received/sec	Single
	3274	UDP Query Received	Single
	3276	UDP Query Received/sec	Single
	3278	TCP Query Received	Single
	3280	TCP Query Received/sec	Single
	3282	Total Response Sent	Single
	3284	Total Response Sent/sec	Single
	3286	UDP Response Sent	Single
	3288	UDP Response Sent/sec	Single
	3290	TCP Response Sent	Single
	3292	TCP Response Sent/sec	Single
	3294	Recursive Queries	Single
	3296	Recursive Queries/sec	Single
	3298	Recursive Send TimeOuts	Single
	3300	Recursive TimeOut/sec	Single
	3302	Recursive Query Failure	Single
	3304	Recursive Query Failure/sec	Single
	3306	Notify Sent	Single
	3308	Zone Transfer Request Received	Single
	3310	Zone Transfer Success	Single
	3312	Zone Transfer Failure	Single
	3314	AXFR Request Received	Single
	3316	AXFR Success Sent	Single

3318	IXFR Request Received	Single
3320	IXFR Success Sent	Single
3322	Notify Received	Single
3324	Zone Transfer SOA Request Sent	Single
3326	AXFR Request Sent	Single
3328	AXFR Response Received	Single
3330	AXFR Success Received	Single
3332	IXFR Request Sent	Single
3334	IXFR Response Received	Single
3336	IXFR Success Received	Single
3338	IXFR UDP Success Received	Single
3340	IXFR TCP Success Received	Single
3342	WINS Lookup Received	Single
3344	WINS Lookup Received/sec	Single
3346	WINS Response Sent	Single
3348	WINS Response Sent/sec	Single
3350	WINS Reverse Lookup Received	Single
3352	WINS Reverse Lookup Received/sec	Single
3354	WINS Reverse Response Sent	Single
3356	WINS Reverse Response Sent/sec	Single
3358	Dynamic Update Received	Single
3360	Dynamic Update Received/sec	Single
3362	Dynamic Update NoOperation	Single
3364	Dynamic Update NoOperation/sec	Single
3366	Dynamic Update Written to Database	Single
3368	Dynamic Update Written to Database/sec	Single
3370	Dynamic Update Rejected	Single
3372	Dynamic Update TimeOuts	Single
3374	Dynamic Update Queued	Single
3376	Secure Update Received	Single
3378	Secure Update Received/sec	Single
3380	Secure Update Failure	Single
3382	Database Node Memory	Single
3384	Record Flow Memory	Single
3386	Caching Memory	Single
3388	UDP Message Memory	Single
3390	TCP Message Memory	Single
3392	Nbstat Memory	Single
3448	DHCP Server	Single
3450	Packets Received/sec	Single
3452	Duplicates Dropped/sec	Single
3454	Packets Expired/sec	Single
3456	Milliseconds per packet (Average).	Single
3458	Active Queue Length	Single
3460	Conflict Check Queue Length	Single
3462	Discovers/sec	Single
3464	Offers/sec	Single
3466	Requests/sec	Single
3468	Informs/sec	Single
3470	Acks/sec	Single

3472	Nacks/sec	Single
3474	Declines/sec	Single
3476	Releases/sec	Single

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), [Copyright © 1996-2000 EarthWeb Inc.](#) All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC


Desktop Library
 Click to access!

APPENDIX E

Plug-and-Play Identifiers

Microsoft, when developing the specifications for Plug and Play (PnP), worked with an existing standard call EISA (Extended Industry Standard Architecture). EISA includes a methodology for identifying devices, assigning each device a three-character manufacturer identifier and a four-digit device identifier (expressed in hexadecimal).

The three-character manufacturer identifier “PNP” was acquired by Microsoft for use with devices that did not normally have identifiers. This allows Windows to use identifiers with devices such as hardware on the system board, I/O ports, keyboards, mice, etc.

In the Windows registry (mostly in HKEY_LOCAL_MACHINE), there are five main classes of PnP numbers, as shown in Table E.1.

TABLE E.1: CLASSES OF PNP DEVICE IDENTIFIERS

Class	ID range
System devices, I/O, timers, input, etc.	PNP0xxx
Network adapters	PNP8xxx
SCSI, proprietary CD adapters	PNPAxxx
Multimedia hardware	PNPBxxx
Modems	PNPCxxx–PNPDxxx

System Devices

System devices use identifiers from PNP0000 to PNP0FFF. Microsoft has subdivided the system class of devices into a number of subcategories (see Table E.2).

TABLE E.2: CATEGORIES OF SYSTEM DEVICES

Type of System Device	Device ID Range
-----------------------	-----------------

Interrupt controllers	PNP0000–PNP00FF
Timers	PNP0100–PNP01FF
DMA	PNP0200–PNP02FF
Keyboards	PNP0300–PNP03FF
Parallel devices	PNP0400–PNP04FF
Serial devices	PNP0500–PNP05FF
Disk controllers	PNP0600–PNP06FF
Compatibility	PNP0802–PNP08FF
Display adapters	PNP0900–PNP09FF
Peripheral buses	PNP0A00–PNP0AFF
Real-time clock	PNP0B00–PNP0BFF
BIOS and system motherboard devices	PNP0C00–PNP0CFF
PCMCIA controller	PNP0E00–PNP0EFF
Mice	PNP0F00–PNP0FFF

Table E.3 shows device IDs for some specific devices in the system class. There may be other devices, though the list does not change frequently.

TABLE E.3: THE SYSTEM DEVICES CLASS

Device ID	Description
Interrupt Controllers	
PNP0000	AT interrupt controller
PNP0001	EISA interrupt controller
PNP0002	MCA interrupt controller
PNP0003	APIC
PNP0004	Cyrix SLiC MP interrupt controller
Timers	
PNP0100	AT timer
PNP0101	EISA timer
PNP0102	MCA timer
DMA	
PNP0200	AT DMA controller
PNP0201	EISA DMA controller
PNP0202	MCA DMA controller
Keyboards	
PNP0300	IBM PC/XT keyboard controller (83-key)
PNP0301	IBM PC/AT keyboard controller (86-key)
PNP0302	IBM PC/XT keyboard controller (84-key)
PNP0303	IBM Enhanced (101/102-key, PS/2 mouse support)
PNP0304	Olivetti keyboard (83-key)
PNP0305	Olivetti keyboard (102-key)
PNP0306	Olivetti keyboard (86-key)
PNP0307	Microsoft Windows keyboard
PNP0308	General Input Device Emulation Interface (GIDEI) legacy
PNP0309	Olivetti keyboard (101/102 key)
PNP030A	AT&T 302 keyboard
PNP030B	Reserved by Microsoft
PNP0320	Japanese 101-key keyboard
PNP0321	Japanese AX keyboard
PNP0322	Japanese 106-key keyboard A01

PNP0323	Japanese 106-key keyboard 002/003
PNP0324	Japanese 106-key keyboard 001
PNP0325	Japanese Toshiba Desktop keyboard
PNP0326	Japanese Toshiba Laptop keyboard
PNP0327	Japanese Toshiba Notebook keyboard
PNP0340	Korean 84-key keyboard
PNP0341	Korean 86-key keyboard
PNP0342	Korean Enhanced keyboard
PNP0343	Korean Enhanced keyboard 101b
PNP0343	Korean Enhanced keyboard 101c
PNP0344	Korean Enhanced keyboard 103
Parallel Devices	
PNP0400	Standard LPT printer port
PNP0401	ECP printer port
Serial Devices	
PNP0500	Standard PC COM port
PNP0501	16550A-compatible COM port
PNP0510	IrDA-compatible device
Disk Controllers	
PNP0600	Generic ESDI-, IDE-, and ATA-compatible hard disk controllers
PNP0601	Plus Hardcard II
PNP0602	Plus Hardcard IIXL/EZ
PNP0603	Generic IDE that supports the Microsoft Device Bay specification
PNP0700	Standard PC floppy disk controller
PNP0701	Standard PC floppy disk controller that supports the Microsoft Device Bay specification
Compatibility	
PNP0800	AT-style speaker sound
PNP0802	Microsoft Sound System-compatible devices (obsolete; instead use PNPB0xx, multimedia devices)
Display Adapters	
PNP0900	VGA compatible
PNP0901	Video Seven VRAM/VRAM II/1024i
PNP0902	8514/A compatible
PNP0903	Trident VGA
PNP0904	Cirrus Logic Laptop VGA
PNP0905	Cirrus Logic VGA
PNP0906	Tseng ET4000
PNP0907	Western Digital VGA
PNP0908	Western Digital Laptop VGA
PNP0909	S3 Inc. 911/924
PNP090A	ATI Ultra Pro/Plus (Mach 32)
PNP090B	ATI Ultra (Mach 8)
Display Adapters	
PNP090C	XGA compatible
PNP090D	ATI VGA Wonder
PNP090E	Weitek P9000 graphics adapter
PNP090F	Oak Technology VGA
PNP0910	Compaq QVision
PNP0911	XGA/2

PNP0912	Tseng Labs W32/W32i/W32p
PNP0913	S3 Inc. 801/928/964
PNP0914	Cirrus Logic 5429/5434 (memory mapped)
PNP0915	Compaq Advanced VGA (AVGA)
PNP0916	ATI Ultra Pro Turbo (Mach64)
PNP0917	Reserved by Microsoft
PNP0918	Matrox MGA
PNP0919	Compaq QVision 2000
PNP091A	Tseng W128
PNP0930	Chips & Technologies Super VGA
PNP0931	Chips & Technologies Accelerator
PNP0940	NCR 77c22e Super VGA
PNP0941	NCR 77c32blt
PNP09FF	Plug-and-Play monitors (VESA DDC)

Peripheral Buses

PNP0A00	ISA bus
PNP0A01	EISA bus
PNP0A02	MCA bus
PNP0A03	PCI bus
PNP0A04	VESA/VL bus
PNP0A05	Generic ACPI bus
PNP0A06	Generic ACPI Extended-IO bus (EIO bus)

Real-Time Clock

PNP0B00	AT real-time clock
---------	--------------------

BIOS and System Motherboard Devices

PNP0C00	Plug-and-Play BIOS (only created by the root enumerator)
PNP0C01	System board
PNP0C02	General ID for reserving resources required by Plug-and-Play motherboard registers (not specific to a particular device)
PNP0C03	Plug-and-Play BIOS event notification interrupt
PNP0C04	Math coprocessor
PNP0C05	APM (Advanced Power Management) BIOS (version independent)
PNP0C06	Reserved for identification of early Plug-and-Play BIOS implementations
PNP0C07	Reserved for identification of early Plug-and-Play BIOS implementations
PNP0C08	ACPI (Advanced Configuration and Power Interface) system board hardware
PNP0C09	ACPI embedded controller
PNP0C0A	ACPI control method battery
PNP0C0B	ACPI fan
PNP0C0C	ACPI power-button device
PNP0C0D	ACPI lid device
PNP0C0E	ACPI sleep-button device
PNP0C0F	PCI interrupt link device
PNP0C10	ACPI system indicator device
PNP0C11	ACPI thermal zone
PNP0C12	Device Bay Controller

PCMCIA Controller

PNP0E00	Intel 82365-compatible PCMCIA controller
PNP0E01	Cirrus Logic CL-PD6720 PCMCIA controller
PNP0E02	VLSI VL82C146 PCMCIA controller
PNP0E03	Intel 82365-compatible CardBus controller

Mice

PNP0F00	Microsoft bus mouse
PNP0F01	Microsoft serial mouse
PNP0F02	Microsoft InPort mouse
PNP0F03	Microsoft PS/2-style mouse
PNP0F04	Mouse Systems mouse
PNP0F05	Mouse Systems three-button mouse (COM2)
PNP0F06	Genius mouse (COM1)
PNP0F07	Genius mouse (COM2)
PNP0F08	Logitech serial mouse
PNP0F09	Microsoft BallPoint serial mouse
PNP0F0A	Microsoft Plug-and-Play mouse
PNP0F0B	Microsoft Plug-and-Play BallPoint mouse
PNP0F0C	Microsoft-compatible serial mouse
PNP0F0D	Microsoft InPort-compatible mouse
PNP0F0E	Microsoft-compatible PS/2-style mouse
PNP0F0F	Microsoft BallPoint-compatible serial mouse
PNP0F10	Texas Instruments QuickPort mouse
PNP0F11	Microsoft-compatible bus mouse
PNP0F12	Logitech PS/2-style mouse
PNP0F13	PS/2 port for PS/2-style mice
PNP0F14	Microsoft Kids mouse
PNP0F15	Logitech bus mouse
PNP0F16	Logitech SWIFT device
PNP0F17	Logitech-compatible serial mouse
PNP0F18	Logitech-compatible bus mouse
PNP0F19	Logitech-compatible PS/2-style mouse
PNP0F1A	Logitech-compatible SWIFT device
PNP0F1B	HP Omnibook mouse
PNP0F1C	Compaq LTE trackball PS/2-style mouse
PNP0F1D	Compaq LTE trackball serial mouse
PNP0F1E	Microsoft Kids trackball mouse
PNP0F1F	Reserved by Microsoft Input Device Group
PNP0F20	Reserved by Microsoft Input Device Group
PNP0F21	Reserved by Microsoft Input Device Group
PNP0F22	Reserved by Microsoft Input Device Group
PNP0F23	Reserved by Microsoft Input Device Group
PNP0FFF	Reserved by Microsoft Systems

NIC Devices

The next class of PnP devices includes NIC (network interface card) devices. NICs are found on virtually all computers. Table E.4 lists the NIC device IDs.

TABLE E.4: THE NETWORK INTERFACE CARDS CLASS

Device ID	Description
PNP8001	Novell/Anthem NE3200
PNP8004	Compaq NE3200
PNP8006	Intel EtherExpress/32
PNP8008	HP EtherTwist EISA LAN Adapter/32 (HP27248A)

PNP8065	Ungermann-Bass NIUps or NIUps/EOTP
PNP8072	DEC (DE211) EtherWorks MC/TP
PNP8073	DEC (DE212) EtherWorks MC/TP_BNC
PNP8078	DCA 10 Mb MCA
PNP8074	HP MC LAN Adapter/16 TP (PC27246)
PNP80C9	IBM Token Ring
PNP80CA	IBM Token Ring II
PNP80CB	IBM Token Ring II/Short
PNP80CC	IBM Token Ring 4/16Mbps
PNP80D3	Novell/Anthem NE1000
PNP80D4	Novell/Anthem NE2000
PNP80D5	NE1000 compatible
PNP80D6	NE2000 compatible
PNP80D7	Novell/Anthem NE1500T
PNP80D8	Novell/Anthem NE2100
PNP80DD	SMC ARCNETPC
PNP80DE	SMC ARCNET PC100, PC200
PNP80DF	SMC ARCNET PC110, PC210, PC250
PNP80E0	SMC ARCNET PC130/E
PNP80E1	SMC ARCNET PC120, PC220, PC260
PNP80E2	SMC ARCNET PC270/E
PNP80E5	SMC ARCNET PC600W, PC650W
PNP80E7	DEC DEPCA
PNP80E8	DEC (DE100) EtherWorks LC
PNP80E9	DEC (DE200) EtherWorks Turbo
PNP80EA	DEC (DE101) EtherWorks LC/TP
PNP80EB	DEC (DE201) EtherWorks Turbo/TP
PNP80EC	DEC (DE202) EtherWorks Turbo/TP_BNC
PNP80ED	DEC (DE102) EtherWorks LC/TP_BNC
PNP80EE	DEC EE101 (built in)
PNP80EF	DEC pc 433 WS (built in)
PNP80F1	3Com Etherlink Plus
PNP80F3	3Com Etherlink II or IITP (8- or 16-bit)
PNP80F4	3Com TokenLink
PNP80F6	3Com Etherlink 16
PNP80F7	3Com Etherlink III
PNP80F8	3Com generic Etherlink Plug-and-Play device
PNP80FB	Thomas Conrad TC6045
PNP80FC	Thomas Conrad TC6042
PNP80FD	Thomas Conrad TC6142
PNP80FE	Thomas Conrad TC6145
PNP80FF	Thomas Conrad TC6242
PNP8100	Thomas Conrad TC6245
PNP8105	DCA 10 MB
PNP8106	DCA 10 MB fiber-optic
PNP8107	DCA 10 MB twisted-pair
PNP8113	Racal NI6510
PNP811C	Ungermann-Bass NIUpc
PNP8120	Ungermann-Bass NIUpc/EOTP
PNP8123	SMC StarCard PLUS (WD/8003S)

PNP8124	SMC StarCard PLUS with onboard hub (WD/8003SH)
PNP8125	SMC EtherCard PLUS (WD/8003E)
PNP8126	SMC EtherCard PLUS with boot ROM socket (WD/8003EBT)
PNP8127	SMC EtherCard PLUS with boot ROM socket (WD/8003EB)
PNP8128	SMC EtherCard PLUS TP (WD/8003WT)
PNP812A	SMC EtherCard PLUS 16 with boot ROM socket (WD/8013EBT)
PNP812D	Intel EtherExpress 16 or 16TP
PNP812F	Intel TokenExpress 16/4
PNP8130	Intel TokenExpress MCA 16/4
PNP8132	Intel EtherExpress 16 (MCA)
PNP8137	Artisoft AE-1
PNP8138	Artisoft AE-2 or AE-3
PNP8141	Amplicard AC 210/XT
PNP8142	Amplicard AC 210/AT
PNP814B	Everex SpeedLink /PC16 (EV2027)
PNP8155	HP PC LAN Adapter/8 TP (HP27245)
PNP8156	HP PC LAN Adapter/16 TP (HP27247A)
PNP8157	HP PC LAN Adapter/8 TL (HP27250)
PNP8158	HP PC LAN Adapter/16 TP Plus (HP27247B)
PNP8159	HP PC LAN Adapter/16 TL Plus (HP27252)
PNP815F	National Semiconductor EtherNODE *16AT
PNP8160	National Semiconductor AT/LANTIC EtherNODE 16-AT3
PNP816A	NCR Token Ring 4Mbps ISA
PNP816D	NCR Token Ring 16/4Mbps ISA
PNP8191	Olicom 16/4 Token Ring adapter
PNP81C3	SMC EtherCard PLUS Elite (WD/8003EP)
PNP81C4	SMC EtherCard PLUS 10T (WD/8003W)
PNP81C5	SMC EtherCard PLUS Elite 16 (WD/8013EP)
PNP81C6	SMC EtherCard PLUS Elite 16T (WD/8013W)
PNP81C7	SMC EtherCard PLUS Elite 16 Combo (WD/8013EW or 8013EWC)
PNP81C8	SMC EtherElite Ultra 16
PNP81E4	Pure Data PDI9025-32 (Token Ring)
PNP81E6	Pure Data PDI508+ (ARCNET)
PNP81E7	Pure Data PDI516+ (ARCNET)
PNP81EB	Proteon Token Ring (P1390)
PNP81EC	Proteon Token Ring (P1392)
PNP81ED	Proteon ISA Token Ring (1340)
PNP81EE	Proteon ISA Token Ring (1342)
PNP81EF	Proteon ISA Token Ring (1346)
PNP81F0	Proteon ISA Token Ring (1347)
PNP81FF	Cabletron E2000 Series DNI
PNP8200	Cabletron E2100 Series DNI
PNP8209	Zenith Data Systems Z-Note
PNP820A	Zenith Data Systems NE2000 compatible
PNP8213	Xircom Pocket Ethernet II
PNP8214	Xircom Pocket Ethernet I
PNP821D	RadiSys EXM-10
PNP8227	SMC 3000 Series
PNP8228	SMC 91C2 controller
PNP8231	Advanced Micro Devices AM2100/AM1500T

PNP8263	Tulip NCC-16
PNP8277	Exos 105
PNP828A	Intel '595-based Ethernet
PNP828B	TI2000-style Token Ring
PNP828C	AMD PCNet Family cards
PNP828D	AMD PCNet32 (VL version)
PNP8294	IrDA infrared NDIS driver (Microsoft supplied)
PNP82BD	IBM PCMCIA-NIC
PNP82C2	Xircom CE10
PNP82C3	Xircom CEM2
PNP8321	DEC Ethernet (all types)
PNP8323	SMC EtherCard (all types except 8013/A)
PNP8324	ARCNET compatible
PNP8326	Thomas Conrad (all ARCNET Types)
PNP8327	IBM Token Ring (all types)
PNP8385	Remote network access driver
PNP8387	RNA Point-to-Point Protocol driver
PNP8388	Reserved for Microsoft Networking components
PNP8389	Peer IrLAN infrared driver (Microsoft supplied)

SCSI and CD-ROM Devices

THE NEXT CLASS OF PNP DEVICES INCLUDES THE SCSI AND CD-ROM devices. Most high-performance computers use SCSI to interface the disk storage with the system. In addition, virtually every computer today is equipped with at least one CD-ROM drive. Table E.5 lists the SCSI and CD-ROM device IDs.

TABLE E.5: THE SCSI AND CD-ROM CLASS

Device ID	Description
PNPA002	Future Domain 16-700 compatible controller
PNPA003	Panasonic proprietary CD-ROM adapter (Sound Blaster Pro/Sound Blaster 16)
PNPA01B	Trantor 128 SCSI controller
PNPA01D	Trantor T160 SCSI controller
PNPA01E	Trantor T338 Parallel SCSI controller
PNPA01F	Trantor T348 Parallel SCSI controller
PNPA020	Trantor Media Vision SCSI controller
PNPA022	Always IN-2000 SCSI controller
PNPA02B	Sony proprietary CD-ROM controller
PNPA02D	Trantor T13b 8-bit SCSI controller
PNPA02F	Trantor T358 Parallel SCSI controller
PNPA030	Mitsumi LU-005 Single Speed CD-ROM controller & drive
PNPA031	Mitsumi FX-001 Single Speed CD-ROM controller & drive
PNPA032	Mitsumi FX-001 Double Speed CD-ROM controller & drive

Multimedia Devices

The next class of PnP devices includes multimedia devices. Most computers have a sound card and many are video capable. Nearly all computers have a joystick port too, useful for games and other software. Table E.6 lists the multimedia device IDs.

TABLE E.6: THE MULTIMEDIA DEVICES CLASS

Device ID	Description
PNPB000	Sound Blaster 1.5-compatible sound device
PNPB001	Sound Blaster 2.0-compatible sound device
PNPB002	Sound Blaster Pro-compatible sound device
PNPB003	Sound Blaster 16-compatible sound device
PNPB004	Thunderboard-compatible sound device
PNPB005	Adlib-compatible FM synthesizer device
PNPB006	MPU401 compatible
PNPB007	Microsoft Windows Sound System-compatible sound device
PNPB008	Compaq Business Audio
PNPB009	Plug-and-Play Microsoft Windows Sound System device
PNPB00A	MediaVision Pro Audio Spectrum (Trantor SCSI enabled, Thunder Chip disabled)
PNPB00B	MediaVision Pro Audio 3D
PNPB00C	MusicQuest MQX-32M
PNPB00D	MediaVision Pro Audio Spectrum Basic (no Trantor SCSI, Thunder Chip enabled)
PNPB00E	MediaVision Pro Audio Spectrum (Trantor SCSI enabled, Thunder Chip disabled)
PNPB00F	MediaVision Jazz-16 chipset (OEM Versions)
PNPB010	Auravision VxP500 chipset - Orchid Videola
PNPB018	MediaVision Pro Audio Spectrum 8-bit
PNPB019	MediaVision Pro Audio Spectrum Basic (no Trantor SCSI, Thunder Chip enabled)
PNPB020	Yamaha OPL3-compatible FM synthesizer device
PNPB02F	Joystick/game port

Modem Devices

The final class of PnP devices includes modem devices, both internal and external. Most computers have a modem, used to connect to the Internet, other computers, or online services. Modem devices are assigned identifiers in the range PNPC000 to PNPC001. Table E.7 lists the modem device IDs.

TABLE E.7: THE MODEM DEVICES CLASS

Device ID	Description
PNPC000	Compaq 14400 modem (TBD)
PNPC001	Compaq 2400/9600 modem (TBD)

[Previous](#) [Table of Contents](#) [Next](#)

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

JUMP TO TOPIC

**Desktop Library**

Click to access!

APPENDIX F

Office 2000 CLSIDs

Office 2000 installs a number of CLSIDs in the registry. If you come across a CLSID that don't recognize, check this list to see if the CLSID is from Office 2000. Keep in mind that CLSIDs are critical to the proper operation of Windows, so do not delete them unless you know what you are doing.

The Office 2000 CLSIDs—including brief descriptions of their functionality, and file, type, and ordinal information—are listed in Table F.1; not every installation will install all of these CLSIDs.

TABLE F.1: OFFICE 2000 CLSIDS

CLSID	Description	File, Ordinal, Type
0000002F-0000-0000-C000-000000000046	CLSID Record Info	oldeaut32.REG, 10, CLSID
0000002F-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 20, CLSID
00020420-0000-0000-C000-000000000046	PS Dispatch	oldeaut32.REG, 40, CLSID
00020420-0000-0000-C000-000000000046	Inproc Server	oldeaut32.REG, 50StdPicture, CLSID
00020420-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 7, oldeaut32.REG, 8, CLSID
00020421-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 11, oldeaut32.REG, 12, CLSID
00020421-0000-0000-C000-000000000046	PS Enum Variant	oldeaut32.REG, 9, CLSID
00020421-0000-0000-C000-000000000046	Inproc Server ORAPI	JETERR40.CHM
00020422-0000-0000-C000-000000000046	PS Type Info	oldeaut32.REG, 13, CLSID
00020422-0000-0000-C000-000000000046	Inproc Server OLE2disp.DLL	oldeaut32.REG, 14, CLSID
00020422-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 15, oldeaut32.REG, 16, CLSID

00020423-0000-0000-C000-000000000046	PS Type Lib	oldeaut32.REG, 17, CLSID
00020423-0000-0000-C000-000000000046	Inproc Server	oldeaut32.REG, 18, CLSID
00020423-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 19, oldeaut32.REG, 2, CLSID
00020424-0000-0000-C000-000000000046	PS OA Interface	oldeaut32.REG, 21, CLSID
00020424-0000-0000-C000-000000000046	Inproc Server	oldeaut32.REG, 22, CLSID
00020424-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 23, oldeaut32.REG, 24, CLSID
00020425-0000-0000-C000-000000000046	PS Type Comp	oldeaut32.REG, 25, CLSID
00020424-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 23, oldeaut32.REG, 24, CLSID
00020425-0000-0000-C000-000000000046	PS Type Comp	oldeaut32.REG, 25, CLSID
00020425-0000-0000-C000-000000000046	Inproc Server	oldeaut32.REG, 26, CLSID
00020425-0000-0000-C000-000000000046	Inproc Server32	oldeaut32.REG, 27, oldeaut32.REG, 28, CLSID
00020430-0000-0000-C000-000000000046	OLE Automation Microsoft Office Web Components 9	0msrtdit 1.0 Type Library
0002E500-0000-0000-C000-000000000046	Implemented Categories	<Unknown>
0002E500-0000-0000-C000-000000000046	Microsoft Office Chart 9	<Unknown>
0002E500-0000-0000-C000-000000000046	Programmable	<Unknown>
0002E500-0000-0000-C000-000000000046	ToolboxBitmap32	<Unknown>
0002E500-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E500-0000-0000-C000-000000000046	TypeLib	<Unknown>
0002E500-0000-0000-C000-000000000046	Version 1.0	msowcd.REG, 130OWC.FieldList.9, CLSID
0002E500-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 140OWC.Record Navigation Control.9, CLSID
0002E500-0000-0000-C000-000000000046	Control	<Unknown>
0002E500-0000-0000-C000-000000000046	Implemented Categories	msowcd.REG, 30, CLSID
0002E500-0000-0000-C000-000000000046	Implemented Categories	<Unknown>
0002E500-0000-0000-C000-000000000046	Inproc Server32	msowcd.REG, 50, CLSID
0002E500-0000-0000-C000-000000000046	Miscellaneous Status	<Unknown>
0002E500-0000-0000-C000-000000000046	Miscellaneous Status 1	msowcd.REG, 80, CLSID
0002E500-0000-0000-C000-000000000046	ProgID	msowcd.REG, 90, CLSID
0002E510-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E510-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 16, CLSID
0002E510-0000-0000-C000-000000000046	Control	msowcd.REG, 17, CLSID
0002E510-0000-0000-C000-000000000046	Implemented Categories	msowcd.REG, 18, CLSID
0002E510-0000-0000-C000-000000000046	Implemented Categories	<Unknown>

0002E510-0000-0000-C000-000000000046	Inproc Server32	msowcd.REG, 2, CLSID
0002E510-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 22, CLSID
0002E510-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 23, CLSID
0002E510-0000-0000-C000-000000000046	ProgID	msowcd.REG, 24, CLSID
0002E510-0000-0000-C000-000000000046	Programmable	msowcd.REG, 25, CLSID
0002E510-0000-0000-C000-000000000046	ToolboxBitmap32	msowcd.REG, 26, CLSID
0002E510-0000-0000-C000-000000000046	TypeLib	msowcd.REG, 27, CLSID
0002E510-0000-0000-C000-000000000046	Version	msowcd.REG, 28, CLSID
0002E510-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 29, CLSID
0002E520-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E520-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 3, CLSID
0002E520-0000-0000-C000-000000000046	Control	msowcd.REG, 31, CLSID
0002E520-0000-0000-C000-000000000046	Implemented Categories	msowcd.REG, 32, CLSID
0002E520-0000-0000-C000-000000000046	Implemented Categories	<Unknown>
0002E520-0000-0000-C000-000000000046	InprocServer32	InprocServer32
0002E520-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 36, CLSID
0002E520-0000-0000-C000-000000000046	Miscellaneous Status 1	msowcd.REG, 37, CLSID
0002E520-0000-0000-C000-000000000046	ProgID	msowcd.REG, 38, CLSID
0002E520-0000-0000-C000-000000000046	Programmable	msowcd.REG, 39, CLSID
0002E520-0000-0000-C000-000000000046	ToolboxBitmap32	msowcd.REG, 4, CLSID
0002E520-0000-0000-C000-000000000046	TypeLib	msowcd.REG, 41, CLSID
0002E520-0000-0000-C000-000000000046	Version	msowcd.REG, 42, CLSID
0002E520-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 43, CLSID
0002E530-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E530-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 44, CLSID
0002E530-0000-0000-C000-000000000046	Control	msowcd.REG, 45, CLSID
0002E530-0000-0000-C000-000000000046	Inproc Server32	<Unknown>
0002E530-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 48, CLSID
0002E530-0000-0000-C000-000000000046	aMiscellaneous Status	msowcd.REG, 49, CLSID
0002E530-0000-0000-C000-000000000046	ProgID	msowcd.REG, 5, CLSID
0002E530-0000-0000-C000-000000000046	Programmable	msowcd.REG, 51, CLSID
0002E530-0000-0000-C000-000000000046	ToolboxBitmap32	msowcd.REG, 52, CLSID
0002E530-0000-0000-C000-000000000046	TypeLib	msowcd.REG, 53, CLSID

0002E530-0000-0000-C000-000000000046	Version	msowcd.REG, 54, CLSID
0002E530-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 55, CLSID
0002E531-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E531-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 56, CLSID
0002E531-0000-0000-C000-000000000046	Control	msowcd.REG, 57, CLSID
0002E531-0000-0000-C000-000000000046	Inproc Server32	<Unknown>
0002E531-0000-0000-C000-000000000046	Miscellaneous Status	<Unknown>
0002E531-0000-0000-C000-000000000046	Miscellaneous Status 1	msowcd.REG, 61, CLSID
0002E531-0000-0000-C000-000000000046	ProgID	msowcd.REG, 62, CLSID
0002E531-0000-0000-C000-000000000046	Programmable	msowcd.REG, 63, CLSID
0002E531-0000-0000-C000-000000000046	ToolboxBitmap32	msowcd.REG, 64, CLSID
0002E531-0000-0000-C000-000000000046	TypeLib	msowcd.REG, 65, CLSID
0002E531-0000-0000-C000-000000000046	Version	msowcd.REG, 66, CLSID
0002E531-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 67, CLSID
0002E532-0000-0000-C000-000000000046	<Unknown>	<Unknown>
0002E532-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 68, CLSID
0002E532-0000-0000-C000-000000000046	Control	msowcd.REG, 69, CLSID
0002E532-0000-0000-C000-000000000046	Inproc Server32	msowcd.REG, 7, CLSID
0002E532-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 72, CLSID
0002E532-0000-0000-C000-000000000046	Miscellaneous Status	msowcd.REG, 73, CLSID
0002E532-0000-0000-C000-000000000046	ProgID	msowcd.REG, 74, CLSID
0002E532-0000-0000-C000-000000000046	Programmable	msowcd.REG, 75, CLSID
0002E532-0000-0000-C000-000000000046	ToolboxBitmap32	msowcd.REG, 76, CLSID
0002E532-0000-0000-C000-000000000046	TypeLib	msowcd.REG, 77, CLSID
0002E532-0000-0000-C000-000000000046	Version	msowcd.REG, 78, CLSID
0002E532-0000-0000-C000-000000000046	Version-Independent ProgID	msowcd.REG, 79, CLSID
0002E540-0000-0000-C000-000000000046	<Unknown>	msowcd.REG, 120OWC.DataSourceControl.9, CLSID
002C9999-0000-0000-C000-000000000112	Global Jet Replication International Global Jet Replication VB	MSRCLR40.DLL msrclr40
0BE35203-8F91-11CE-9DE3-00AA004BB851	Standard Font	oldeaut32.REG, 29, CLSID
0BE35203-8F91-11CE-9DE3-00AA004BB851	CLSID	oldeaut32.REG, 3, oldeaut32.REG, 30, CLSID
0BE35203-8F91-11CE-9DE3-00AA004BB851	Inproc Server32	oldeaut32.REG, 31, oldeaut32.REG, 32, CLSID
0BE35203-8F91-11CE-9DE3-00AA004BB851	ProgID Standard Font	oldeaut32.REG, 33, CLSID

0BE35203-8F91-11CE-9DE3-00AA004BB851	<Unknown>	oldeaut32.REG, 49, oldeaut32.REG, 5, CLSID
0BE35204-8F91-11CE-9DE3-00AA004BB851	Standard Picture	oldeaut32.REG, 34, CLSID
0BE35204-8F91-11CE-9DE3-00AA004BB851	CLSID	oldeaut32.REG, 35, CLSID
0BE35204-8F91-11CE-9DE3-00AA004BB851	Inproc Server32	oldeaut32.REG, 36, oldeaut32.REG, 37, CLSID
0BE35204-8F91-11CE-9DE3-00AA004BB851	ProgID Standard Picture	oldeaut32.REG, 38, CLSID
0BE35204-8F91-11CE-9DE3-00AA004BB851	<Unknown>	oldeaut32.REG, 51, SOFTWARE Classes Interface
0D5BC421-274B-11D2-A185-00A0C90AB50F	Global Setup Install MSI	msxml.DLL
10048713-2C96-11D2-9A97-006097C4E452	Global System STANDARD OLE	OLEpro32.DLL
153E1F3A-DD55-11D1-9393-00A0C90F27F9	Global Data Access ADO Core	msador15.DLL
153E1F3C-DD55-11D1-9393-00A0C90F27F9	Global Data Access ADO International	msado15.DLL
153E1F3E-DD55-11D1-9393-00A0C90F27F9	Global Data Access Ado Extras	msader15.DLL
153E1F46-DD55-11D1-9393-00A0C90F27F9	SharedOLEDbFoldermsolap.DLL	<Unknown>
153E1F48-DD55-11D1-9393-00A0C90F27F9	Global Data Access DCube.SQL	
153E1F48-DD55-11D1-9393-00A0C90F27F9	0001Global Data Access Jet.acc IISAM	SQLwoa.DLL
153E1F4A-DD55-11D1-9393-00A0C90F27F9	Global Data Access SQL OLE DB	SQLsrv32.DLL
153E1F4C-DD55-11D1-9393-00A0C90F27F9	Global Data Access SQL Inst cat	dbnmpntw.DLL
153E1F50-DD55-11D1-9393-00A0C90F27F9	Installation Catalog SQL Global Data Access SQL ODBC Core	<Unknown>
153E1F52-DD55-11D1-9393-00A0C90F27F9	ODBCSQLDriverFolderODBCbcp.DLL	<Unknown>
153E1F54-DD55-11D1-9393-00A0C90F27F9	Global Data Access SQL Connectivity NT	SQL OLEdb.DLL
1C783E10-75C5-11D2-AAA5-00A0C90F57B0	Global Data Access RDS Win Dir	msdatsrc.tlb
20A57C61-3DE0-11D2-AA8B-00A0C90F57B0	Global Web Components XC1046 TARGET DIR	msowc.msi
2F7FC181-292B-11D2-A795-DFAA798E9148	<Unknown>	msrtdedit.REG, 14, CLSID
2F7FC18E-292B-11D2-A795-DFAA798E9148	ToolboxBitmap32	msrtdedit.REG, 11, CLSID
2F7FC18E-292B-11D2-A795-DFAA798E9148	Miscellaneous Status	msrtdedit.REG, 12, CLSID
2F7FC18E-292B-11D2-A795-DFAA798E9148	Miscellaneous Status 1	msrtdedit.REG, 13, CLSID
2F7FC18E-292B-11D2-A795-DFAA798E9148	TypeLib	<Unknown>
2F7FC18E-292B-11D2-A795-DFAA798E9148	Version	msrtdedit.REG, 9, CLSID
2F7FC18E-292B-11D2-A795-DFAA798E9148	Inproc Server32 ODBC Default DSN (ODBC Default DSNF Folder Value)	oldeaut32.REG, 1, CLSID
30A095E2-9A0C-11D2-93BB-00105A994D2C	Inproc Server32	msoeuro.REG, 1, CLSID
30A095E2-9A0C-11D2-93BB-00105A994D2C	ProgID	msoeuro.Converter.1 msoeuro.REG, 2, CLSID

30A095E2-9A0C-11D2-93BB-00105A994D2C	Version-Independent ProgID	msoeuro Converter msoeuro.REG, 3, msoeuro Converter Current version msoeuro.REG, 4, msoeuro.Converter.1 CLSID
30A095E2-9A0C-11D2-93BB-00105A994D2C	<Unknown>	msoeuro.REG, 5, msoeuro Converter CLSID
359C9A84-B774-11D1-A2CC-00E029143AAC	Office International Folder	msowcrdp.chm, Global User Assistance OCP Pivot List
37A53440-26DC-11D2-ACBB-0080C7FCBB84	Windows Folder	msdfmap.ini, Global Data Access SQL.Connectivity
46763EE0-CAB2-11CE-8C20-00AA0051E5D4	Obsolete Font	oldeaut32.REG, 39, CLSID
46763EE0-CAB2-11CE-8C20-00AA0051E5D4	<Unknown>	oldeaut32.REG, 4, CLSID
46763EE0-CAB2-11CE-8C20-00AA0051E5D4	Inproc Server32	oldeaut32.REG, 41, CLSID
46763EE0-CAB2-11CE-8C20-00AA0051E5D4	ProgID OldFont	oldeaut32.REG, 42, CLSID
46763EE0-CAB2-11CE-8C20-00AA0051E5D4	<Unknown>	oldeaut32.REG, 47, oldeaut32.REG, 48StdFont, CLSID
474F61F1-7342-11D2-A199-00A0C90AB50F	Global User Assistance OCP Access Data Pages	stdOLE2.tlb
4C85388F-1500-11D1-A0DF-00C04FC9E20F	<Unknown>	<Unknown>
4C85388F-1500-11D1-A0DF-00C04FC9E20F	<Unknown>	msowcd.REG, 8, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Implemented Categories	msowcd.REG, 81, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Implemented Categories	<Unknown>
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Inproc Server32	<Unknown>
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Miscellaneous Status	msowcd.REG, 85, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Miscellaneous Status 1	msowcd.REG, 86, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	ProgID	msowcd.REG, 87, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Programmable	msowcd.REG, 88, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	ToolboxBitmap32	msowcd.REG, 89, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	TypeLib	msowcd.REG, 9, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Version	msowcd.REG, 91, CLSID
4C85388F-1500-11D1-A0DF-00C04FC9E20F	Version-Independent ProgID	msowcd.REG, 92, CLSID
4EF6100A-AF88-11D0-9846-00C04FC29993	Ifont Events Display	oldeaut32.REG, 52 SOFTWARE Classes Interface
4EF6100A-AF88-11D0-9846-00C04FC29993	ProxyStubClsid32	<Unknown>
5D0A2606-B783-11D1-B901-00AA00585640	Microsoft Office Web Components Function Library	<Unknown>
76F6F3F5-9937-11D2-93BB-00105A994D2C	Microsoft Office Euro Converter Object Library	<Unknown>
7878A751-C051-11D1-9568-00A0C9082635	Global SQL Personal Server DCOM	MSI Folder ADMIN instmsi.exe
790EF0D4-DE15-11D1-9394-00A0C90F27F9	Global Data Access Jet IIS AM	msjet40.DLL
790EF0D6-DE15-11D1-9394-00A0C90F27F9	Global Data Access Jet International	msexch40.DLL
790EF0D8-DE15-11D1-9394-00A0C90F27F9	Global Data Access Jet System	msjint40.DLL

790EF0DA-DE15-11D1-9394-00A0C90F27F9	ODBC Desktop Driver Folder	msrd2x40.DLL, Global Data Access Jet Core
790EF0DC-DE15-11D1-9394-00A0C90F27F9	Global Data Access JRO	msjtes40.DLL
7BF80981-BF32-101A-8BBB-00AA00300CAB	IPicture Display	oldeaut32.REG, 54 SOFTWARE Classes Interface
7BF80981-BF32-101A-8BBB-00AA00300CAB	ProxyStubClsid32	oldeaut32.REG, 55 SOFTWARE Classes Interface
8488FAA3-8EA6-11D1-9ABC-006097C4E452	Fonts Folder Global Internet Explorer HTML Help	<Unknown>
902D8580-2026-11D2-93B1-00A0C90F27F9	ADO Verbs Including Global Data Access DCube Core	<Unknown>
902D8584-2026-11D2-93B1-00A0C90F27F9	Global Data Access RDS Client	msdaosp.DLL
902D8586-2026-11D2-93B1-00A0C90F27F9	Global Data Access ODBC Core	msjro.DLL
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	<Unknown>	<Unknown>
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	<Unknown>	msowcd.REG, 93, CLSID
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	Inproc Server32	<Unknown>
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	ProgID	msowcd.REG, 96, CLSID
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	Programmable	msowcd.REG, 97, CLSID
97D6D376-23BB-11D1-A0E1-00C04FC9E20F	Version-Independent ProgID	msowcd.REG, 98, CLSID
997FA962-E067-11D1-9396-00A0C90F27F9	Global System OLEPRO32	oldeaut32.DLL
9C0B38C0-9092-11D2-AAA7-00A0C90F57B0	Euro Folder Global Fonts Tahoma	<Unknown>
9FADACD8-37AC-11D2-93BC-00A0C90F27F9	Global Web Components Core International	ocp.DLL
A9A10010-338A-11d2-BE43-006008C3FEBE	<Unknown>	msowcd.REG, add02Component Categories
A9A10010-338A-11D2-BE43-006008C3FEBE	Create Design Time UI Activatable Control	msowcd.REG, 1, CLSID
B196B286-BAB4-101A-B69C-00AA00341D07	PS Factory Buffer	oldeaut32.REG, 43, CLSID
B196B286-BAB4-101A-B69C-00AA00341D07	Inproc Server32	oldeaut32.REG, 44 oldeaut32.REG, 45 oldeaut32.REG, 46OldFont, CLSID
B196B286-BAB4-101A-B69C-00AA00341D07	<Unknown>	oldeaut32.REG, 53 SOFTWARE Classes Interface
BEF6E001-A874-101A-8BBA-00AA00300CAB	Standard OLE Types Absent Path bytes GBKBMB	<Unknown>
BEF6E003-A874-101A-8BBA-00AA00300CAB	IFontDisp	oldeaut32.REG, 56 SOFTWARE Classes Interface
BEF6E003-A874-101A-8BBA-00AA00300CAB	ProxyStubClsid32	oldeaut32.REG, 6, CLSID
BF83B991-DF02-11D1-9395-00A0C90F27F9	Global Data Access ODBC International	ODBC32.DLL
BF83B996-DF02-11D1-9395-00A0C90F27F9	Global Data Access ODBC Desktop Core	ODBCint.DLL
BF83B99C-DF02-11D1-9395-00A0C90F27F9	Global Data Access ODBC Desktop International	oddbse32.DLL
BF83B9A0-DF02-11D1-9395-00A0C90F27F9	Global Data Access ODBC Direct Core	ODBCji32.DLL
BF83B9A4-DF02-11D1-9395-00A0C90F27F9	Global Data Access OLE DB Client	msrdo20.DLL

BF83B9A8-DF02-11D1-9395-00A0C90F27F9	Global Data Access OLE DB Server	OLEdb32.DLL
BF83B9AE-DF02-11D1-9395-00A0C90F27F9	Global Data Access OLE DB Server International	msdaenum.DLL
BF83B9B0-DF02-11D1-9395-00A0C90F27F9	Global Data Access Oracle ODBC Core	msdaSQLr.DLL
BF83B9B4-DF02-11D1-9395-00A0C90F27F9	ODBC Oracle Driver Folder, Global Data Access Oracle OLE DB Core	msorcl32.DLL
BF83B9B8-DF02-11D1-9395-00A0C90F27F9	Global Data Access Oracle Shared	msdaora.DLL
BF83B9BA-DF02-11D1-9395-00A0C90F27F9	Global Data Access OSP	mtxoci.DLL
BF83B9BC-DF02-11D1-9395-00A0C90F27F9	Shared RDS Folder Global Data Access RDS Client International	msadce.DLL
BF83B9BE-DF02-11D1-9395-00A0C90F27F9	Global Data Access RDS Server	msadcer.DLL
BF83B9C2-DF02-11D1-9395-00A0C90F27F9	Global Data Access RDS Server International	msadds.DLL
BF83B9C4-DF02-11D1-9395-00A0C90F27F9	Global Data Access RDS System	msaddsr.DLL
CC29E98B-7BC2-11D1-A921-00A0C91E2AA2	<Unknown>	<Unknown>
CC29EAA9-7BC2-11D1-A921-00A0C91E2AA2	<Unknown>	<Unknown>
CC29EAA9-7BC2-11D1-A921-00A0C91E2AA2	<Unknown>	<Unknown>
CC29EAF3-7BC2-11D1-A921-00A0C91E2AA2	Global Web Components Core II	Office Folder msowcf.DLL
CC29EAF9-7BC2-11D1-A921-00A0C91E2AA2	Global Web Components CZ1029	Oc pintl.DLL
CC29EC4F-7BC2-11D1-A921-00A0C91E2AA2	Global User Assistance OCP Web Calc	Oc pdpl.chm
CC29EC51-7BC2-11D1-A921-00A0C91E2AA2	Global VCCRT40	Oc pwc.chm
CC29EDBF-7BC2-11D1-A921-00A0C91E2AA2	<Unknown>	<Unknown>
D267E19A-0B97-11d2-BB1C-00C04FC9B532	<Unknown>	msoeuro.REG, 0, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	OC Function Class	msowcf.REG, 10, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	<Unknown>	msowcf.REG, 2, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	Inproc Server32	msowcf.REG, 20 OC Func CurVer msowcf.REG, 3 Both msowcf.REG, 4, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	ProgID	msowcf.REG, 5, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	Programmable	msowcf.REG, 6, CLSID
D7546AAE-A77A-11D1-B901-00AA00585640	Version-Independent ProgID	msowcf.REG, 7, CLSID
D7546ABD-A77A-11D1-B901-00AA00585640	ProgID OCA TP1	msowcf.REG, 11, CLSID
D7546ABD-A77A-11D1-B901-00AA00585640	Programmable	msowcf.REG, 12, CLSID
D7546ABD-A77A-11D1-B901-00AA00585640	Version-Independent ProgID OCA TP	msowcf.REG, 13OCA TP Class msowcf.REG, 14 msowcf.REG, 15 OC ATP1, CLSID
D7546ABD-A77A-11D1-B901-00AA00585640	<Unknown>	msowcf.REG, 16 OC
D7546ABD-A77A-11D1-B901-00AA00585640	<Unknown>	msowcf.REG, 8, CLSID

D7546ABD-A77A-11D1-B901-00AA00585640	Inproc Server32	msowcf.REG, 9, msrtdedit.REG, 10, CLSID
D83D7DB1-CF1F-11D1-9392-00A0C90F27F9	DCOMdir dcom95.exe Global System OLEAUT32	<Unknown>
DB6E8F48-FD3E-11D0-A0BC-00C04FC9E20F	Inproc Server32 Threading Model = Both	msowcd.REG, 101, CLSID
DB6E8F48-FD3E-11D0-A0BC-00C04FC9E20F	ProgID DP Data View DP DV Event 9	msowcd.REG, 102, CLSID
DB6E8F48-FD3E-11D0-A0BC-00C04FC9E20F	Version-Independent ProgID DP Data View DPDV Event	msowcd.REG, 103 Component Categories
DB6E8F48-FD3E-11D0-A0BC-00C04FC9E20F	<Unknown>	<Unknown>
DB6E8F48-FD3E-11D0-A0BC-00C04FC9E20F	<Unknown>	msowcd.REG, 99msowcf.REG, 1, CLSID
DECCE4F1-4845-11D2-AA8F-00A0C90F57B0	Global Internet Explorer HTML Help Exe	hhctrl.ocx
DECCE4F3-4845-11D2-AA8F-00A0C90F57B0	Global Internet Explorer MSXML	hh.exe
DECCE4F7-4845-11D2-AA8F-00A0C90F57B0	Office DEU Folder ADMIN OR DEU Global Web Components EL1032	<Unknown>
DECCE4F9-4845-11D2-AA8F-00A0C90F57B0	Office JPN Folder ADMIN OR JPN Global Web Components KO1042	<Unknown>
DECCE4FB-4845-11D2-AA8F-00A0C90F57B0	Office FRN Folder ADMIN OR FRN Global Web Components HBR1037	<Unknown>
DECCE4FD-4845-11D2-AA8F-00A0C90F57B0	Office SPN Folder ADMIN OR SPN Global Web Components SV1053	<Unknown>
DEE35070-506B-11CF-B1AA-00AA00B8DE95	Implemented Categories	<Unknown>
E01C96B1-4436-11D2-9AA4-006097C4E452	0001 Global Web Components AR1025	msvcrt40.DLL
EEE4D68C-20C7-11D2-A185-00A0C90AB50F	<Unknown>	<Unknown>
EEE4D68C-20C7-11D2-A185-00A0C90AB50F	Available Volume Cost, Difference Volume Cost, Required Volume Cost, Size Disk, Size Volume, Cost Volume	<Unknown>
F2BB56D1-DB07-11D1-AA6B-006097DB9539	Design Time UI Activatable Control	<Unknown>
F2BB56D1-DB07-11d1-AA6B-006097DB9539	<Unknown>	msowcd.REG, 19, CLSID
F2BB56D1-DB07-11d1-AA6B-006097DB9539	<Unknown>	msowcd.REG, 33, CLSID
F2BB56D1-DB07-11d1-AA6B-006097DB9539	<Unknown>	msowcd.REG, 40, CLSID
F2BB56D1-DB07-11d1-AA6B-006097DB9539	<Unknown>	msowcd.REG, 82, CLSID
F425DE81-4D88-11D2-A188-00A0C90AB50F	Global Data Access SQL Connectivity Win9x	dbmsadsn.DLL
F425DE83-4D88-11D2-A188-00A0C90AB50F	Global Euro Converter	Version9X db msshnm.DLL
F61F69E1-7359-11D2-AAA3-00A0C90F57B0	Office HBR Folder ADMIN OR HBR Global Web Components HU1038	<Unknown>

F61F69E9-7359-11D2-AAA3-00A0C90F57B0	Office CHS Folder ADMIN OR CHS Global Web Components CHT1028	<Unknown>
F61F69EB-7359-11D2-AAA3-00A0C90F57B0	Office CHT Folder ADMIN OR CHT Global Web Components Core	<Unknown>
F61F69F1-7359-11D2-AAA3-00A0C90F57B0	Office ARF Folder ADMIN OR ARA Global Web Components CHS2052	<Unknown>
F61F69F3-7359-11D2-AAA3-00A0C90F57B0	Office EUF older ADMIN OR BAS Global Web Components FA1035	<Unknown>
F61F69F5-7359-11D2-AAA3-00A0C90F57B0	Office XC Folder ADMIN OR BRZ Global Web Components YX1050	<Unknown>
F61F69F7-7359-11D2-AAA3-00A0C90F57B0	Office YX Folder ADMIN OR CRO Action Data Action Progress 95 Progress Bar Progress done Cancel Push Button	<Unknown>
F61F69F9-7359-11D2-AAA3-00A0C90F57B0	Office CZ Folder ADMIN OR CZH Global Web Components DA1030	<Unknown>
F61F69FB-7359-11D2-AAA3-00A0C90F57B0	Office DA Folder ADMIN OR DAN Global Web Components DEU1031	<Unknown>
F61F69FD-7359-11D2-AAA3-00A0C90F57B0	Office NL Folder ADMIN OR DUT Global Web Components NO1044	<Unknown>
F61F69FF-7359-11D2-AAA3-00A0C90F57B0	Office FA Folder ADMIN OR FIN Global Web Components FRN1036	<Unknown>
F61F6A01-7359-11D2-AAA3-00A0C90F57B0	Office EL Folder ADMIN OR GRK Global Web Components EU1069	<Unknown>
F61F6A03-7359-11D2-AAA3-00A0C90F57B0	Office HKEY_USERS Folder ADMIN OR HNG Global Web Components IT1040	<Unknown>
F61F6A05-7359-11D2-AAA3-00A0C90F57B0	Office IT Folder ADMIN OR ITN Global Web Components JPN1041	<Unknown>
F61F6A07-7359-11D2-AAA3-00A0C90F57B0	Office KO Folder ADMIN OR KOR Global Web Components NL1043	<Unknown>
F61F6A09-7359-11D2-AAA3-00A0C90F57B0	Office NO Folder ADMIN OR NOR Global Web Components PL1045	<Unknown>
F61F6A0B-7359-11D2-AAA3-00A0C90F57B0	Office PL Folder ADMIN OR POL Global Web Components PT2070	<Unknown>
F61F6A0D-7359-11D2-AAA3-00A0C90F57B0	Office PT Folder ADMIN OR POR Global Web Components RO1048	<Unknown>
F61F6A0F-7359-11D2-AAA3-00A0C90F57B0	Office RO Folder ADMIN OR ROM Global Web Components RU1049	<Unknown>

F61F6A11-7359-11D2-AAA3-00A0C90F57B0	Office RU Folder ADMIN OR RUS Global Web Components SK1051	<Unknown>
F61F6A13-7359-11D2-AAA3-00A0C90F57B0	Office SK Folder ADMIN OR SLK Global Web Components SL1060	<Unknown>
F61F6A15-7359-11D2-AAA3-00A0C90F57B0	Office SL Folder ADMIN OR SLV Global Web Components SPN3082	<Unknown>
F61F6A17-7359-11D2-AAA3-00A0C90F57B0	Office SV Folder ADMIN OR SWE Global Web Components TH1054	<Unknown>
F61F6A19-7359-11D2-AAA3-00A0C90F57B0	Office TH Folder ADMIN OR THI Global Web Components TR1055	<Unknown>
F61F6A1B-7359-11D2-AAA3-00A0C90F57B0	Office TR Folder ADMIN OR TUR Global Web Components Web Install Logic	<Unknown>

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.



Datamation

Profit and Value from Information Technology

[Ecommerce & Extranets](#) : [Client Systems](#) :
[Enterprise Applications](#) : [Application Development](#)
This site is brought to you by
EarthWeb

HOME

SUBSCRIBE

SEARCH

FAQ

SITEMAP

CONTACT US

KEYWORD SEARCH

**Mastering Windows 2000 Registry**

by Peter D. Hipson

Sybex, Inc.

ISBN: 0782126154 Pub Date: 01/01/00

Search this book:

Search Tips

Advanced Search

PUBLICATION LOOKUP

[Previous](#) [Table of Contents](#) [Next](#)

INDEX

Note to the Reader: Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

A

“A domain controller for your domain could not be contacted...” message, 456

abbreviations for hives, 6

Accepted Documents key, 485

access control, policies for, 140

Access Control Settings dialog box, **96–101**, 97, 99, 102

Access program configuration information

system, **350–353**

user, 359

Access Runtime key, 470

Accessibility key, **404**Keyboard Response key, **405**MouseKeys key, **405**SerialKeys key, **405–406**

ShowSounds key, 406

SoundSentry key, **406**StickyKeys key, **406–409**

TimeOut key, 409

ToggleKeys key, **410**Account key, **448**Aliases key, **448**, 449Groups key, **448–451**, 451Users key, **451–453**, 452**Desktop Library**[Click to access!](#)

- ACCOUNT OPERATORS group, 63
- Accounts key, 458
- ACP value entry, 538
- ACS key, 470
- Actions menu, Properties command, 131
- Active Accessibility key, 470
- ACTIVE_CONTENT key
 - in SO, 486
 - in SOIEAK, 487
- active directory
 - backing up, 23
 - in restoring, 31
- Active Directory Users and Computers tool, **134–135**
- Active Server Pages, **257–258**, 258
- Active Setup key, 470
- Active Setup Temp Folders key, 483
- Active value entry, 526
- ActiveComputerName key, 530
- ActivePort value entry, 405
- ActiveTimeBias value entry, 566
- ActiveWindowTracking value entry, 293, 418
- ActiveWndTrkTimeout value entry, 412
- ActiveX Cache key, 485
- ActiveX controls, **154–156**
- ACTIVEX key
 - in SO, 486
 - in SOIEAK, 487
- ActSysAc value entry, 459
- adapters
 - HKEY_CURRENT_CONFIG hive for, 66
 - initialization of, 9
 - in MultifunctionAdapter, **439–440**, 440
 - Plug-and-Play identifies for, 667–668
- Adaptive_ifs value, 339–340
- Add Counters dialog box, 257, 258, 642, 642
- add function, **108–109**
- Add-In Manager, 351
- add-ins for Access, **351**
- Add Key dialog box, 92, 92
- Add/Remove Programs applet, 220, **497–498**
- Add Value dialog box, 93, 93
- Addins key in PowerPoint, 356
- AddIns key in Terminal Server, **558**
- AddOns key, 504
- addresses
 - for IDE controllers, 334
 - in ResourceMap, 56, 444
- .adm files, **125–128**

- AdminDebug key, 479
- Administrative Templates, **125–129**, 127
- Administrator passwords, **461–462**
- ADMINISTRATORS group, 63
- Adoat15.reg file, 348
- Adobe Illustrator fonts, **512–513**
- Adobe key, 58
- Adofre15.reg file, 348
- ADs key, 470
- Advanced INF Setup key, 470
- Advanced Options Menu, 208–209, 208
- Advanced Power Management (APM)
 - in ApmActive, **526**
 - in ApmLegalHal, **527**
- AeDebug key, **502–503**
- Affinity object, 608
- Agent program, 469
- aliases, **6–7**
- Aliases key
 - in Account, **448**, 449
 - in Domain, **454–455**, 454
- Alignment object
 - in REG_RESOURCE_LIST, 612–613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- All_OK function, 367
- All Users profile, 482
- AllocateCDRoms value entry, 515
- AllocateFloppies value entry, 515
- Allow inheritable auditing entries from parent to propagate to this object option, 99
- Also backup the registry option, 106
- Alternative Lists object, 614
- Always On power policy, 420
- AlwaysUnloadDLL key, 481
- Anonymous value entry, 562
- Answer Wizard key, 356
- API functions, **231–246**
- APM (Advanced Power Management)
 - in ApmActive, **526**
 - in ApmLegalHal, **527**
- ApmActive key, **526**
- ApmLegalHal key, **527**
- App Management key, 480
- App Paths key, **480**
- AppCompatibility key, 553
- Appearance key, **410**
- AppEvent.Evt file, 19
- AppEvents key, 64, 394, **395–399**
- AppInit_DLLs value entry, 513

- Applets key, 480
- application specific data types, 600
- applications
 - client-server, **157–161**, *158–160*
 - data for, 11
 - data types for, 8
 - deleted, **221**
 - errors in, **222**
 - migrating, **251**
 - overwritten, **221–222**
 - in stabilizing system, **218**
 - uninstalling, **521**
- ApplyViewPreferences function, 365
- Appointment key, 360
- AppPatches key, 553
- Approved key, 494
- Apps key, 299
- appserver value entry in query, 560
- AppServer value entry in WinStations, 562
- Arbiters key, 527
- AssignmentOrdering key, **556**
- Assistant key
 - for system configuration, 353, 353
 - for user configuration, 359
- associations, keys for, 59, **374–386**
- AsyncMac key, 470
- ATAPI.SYS file, 444
- Atdisk key, 333
- attrib command, 40
- Attributes value entry, 542
- AudioCompressionManager key, 470
- Audit Permissions on Existing Subkeys option, 430
- auditing, **99–101**, 99, **430**
- Auditing tab, **99–101**, 99
- AUTH key
 - in SO, 486
 - in SOIEAK, 487
- AUTH2UPD value entry, 486
- AuthorizedApplications key, 558
- auto-disconnect services, **581**
- Auto Refresh options, **102–103**
- Auto value entry, 502
- AutoAdminLogon value entry, 515–516
- AutoComplete key, 481
- AutoContent Wizard key, 356
- AutoCorrect key, 359
- AutoEndTasks value entry, 412
- Autoexec.nt file, 17–18, 28

AutoExtract value entry, 482
 Automatic Policy mode, 567
 AutoNameCheck key, 361
 AutoReboot value entry, 531
 AutoRepeatDelay value entry, 405
 AutoRepeatRate value entry, 405
 AutoRestartShell value entry, 513–514
 AuxUserType key, 163–164
 AVIVideo value entry
 in MCI, 507
 in MCI32, 508

B

back doors, 546
 background, **280–283**
 for command prompt, **301–304**, 302–303
 patterns for, **285**, **419**
 TileWallpaper value entry for, **283–284**
 Wallpaper value entry for, **282–283**
 WallpaperOriginX value entry for, **284**
 WallpaperOriginY value entry for, **284–285**
 WallpaperStyle value entry for, **283**
 Backup Domain Controllers (BDCs), 68, 451
 backup function, **109–110**
 BACKUP OPERATORS group, 63
 Backup program, **104–105**
 for ERDs, **105–106**, 105
 saving repair directory in, **106**, 106
 BackupRestore key, 527
 backups
 Copy and Xcopy for, **24–28**
 number of, **21–22**
 RegEdit for, **28–29**
 restoring, **30–35**
 in stabilizing systems, **205–206**
 to tape, **23–24**
 base 16 numbers, **402–403**
 BaseAddress value entry, 334
 BaseWinOptions key, 493
 .bat extension, key for, **375–376**, **381–382**, 381
 batch command, 40
 batfile key, 375–376, **381–382**, 381
 Baud value entry, 405
 BaudRate value entry, 561
 BDCs (Backup Domain Controllers), 68, 451
 BeepEnabled value entry, 539
 Beverly Hills Software Web site, 618

- Bias value entry, 566
- big endian data type, **603**
- Binary Editor dialog box, 50, 50
- binary numbers, **402–403**
- binary values, data type for, **50, 602**
- Binders, 359
- bindform value entry, 510
- BIOS devices, Plug-and-Play identifiers for, 669
- BIOS in MultifunctionAdapter, 439
- Biosinfo key, 527
- bit-planes for icons, 416
- bitmaps
 - EditFlags variables, **376–378**
 - for wallpaper, 282–283
- bits, 402
- blink rate for cursor, **289**, 413
- bold fonts, 308–309
- boot.description key, 517
- boot files, backing up, 23
- boot.ini file
 - on boot floppies, 227
 - in boot process, 10
 - editing, **26–27**
 - Recovery Console modification of, 37
 - for Windows directory, 324–326
- boot key, 517
- boot menu, customizing, **26–27**
- boot process, **9–10**
 - crashes after, **214–216**
 - in disaster recovery, **207–214**, 207–208, 211–212
 - System hive for, **60–62**
- boot sector, 9, 33
- bootable disks, 25, **227**
- BootExecute value entry, 550
- BootPMibAgent key, 470
- BOOTSEC.DAT file, 39
- Bootsect.dos file, 227
- BootVerificationProgram key, **527**
- BorderWidth value entry, 415
- BounceTime value entry, 405
- Brockschmidt, Kraig, 148
- BrowseNewProcess key, 481
- Browser key, 471
- Browser Monitor key, 340
- BrowserWebCheck value entry, 492
- buffers. *See also* caches
 - command history, **311–312**, 401
 - for fonts, **505**

- for networks, **320–321**
- screen, 304–305
- BUILTIN groups, 63
- Builtin key
 - in Accounts, 458
 - in Domains, 448, **454–455**, 454
- Bus Number object
 - in REG_FULL_RESOURCE_DESCRIPTOR, 607
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- buses, 55, 440
 - Plug-and-Play identifiers for, 668
 - in SystemResources, **556–557**
- BusValues key, **556–557**
- bytes, 402
- ByteSize value entry, 561

C

- /C parameter, 83
- C programming, 230, **235**
- Cache key
 - in Control Panel, 411
 - in Internet Settings, 485
 - in Security, **456–457**
- CachedLogonsCount value entry, 514
- CacheLastUpdate value entry, 515
- CachePrimaryDomain value entry, 514
- caches. *See also* buffers
 - for fonts, **505**
 - for icons, 411
 - for Internet Explorer, 485
 - for logons, **456–457**, 514–516
- CacheValid value entry, 516
- Calculator program, 402
- calendars in Clients, 468
- Callback value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 564
 - in WinStations, 562
- CallbackNumber value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 564
 - in WinStations, 562
- Caps Lock settings, **290**, 417
- CaptionFont value entry, 415
- CaptionHeight value entry, 415
- CaptionWidth value entry, 415
- CaretWidth value entry, 412

- cascading menus, **289–290**
- Categories key, 361
- cd command, 40
- CD-ROM Changers, cycling in, **582**
- CD-ROM drives
 - in DISK, 61
 - drive letters for, **330**
 - in paths, 331
 - Plug-and-Play identifiers for, **675**
 - in Scsi, 441–444
- CDAudio value entry
 - in MCI, 507
 - in MCI32, 508
- CdClass value entry, 564
- CdDLL value entry, 564
- CdName value entry, 564
- Centered mode for wallpaper, 281, 284
- CentralProcessor key, **54–55, 438–439**
- CentralProfile value entry, 511
- CertificationAuthorities key, **545–546**
- CF_ Clipboard formats, **150–151**
- CfgDll value entry
 - in RDP-Tcp, 564
 - in rdpwd, 561
- change key, 560
- ChangeID value entry, 542
- changes
 - auditing, **430**
 - comparing registry for, 69
 - rebooting for, 35, 43, 316, 583
- Channel object, 608
- character-mode named pipes, 321
- chat, 474, **638–639, 638**
- chdir command, 40
- CheckBadApps key, 553
- CheckBadApps400 key, 553
- chkdsk program, 40, 223
- Ciphers key, **546**
- Class key, **527–529**
- class value entry, 510
- Classes key, 59, **468**
- Clean Up menu, Delete Entries command, 190
- CleanReg program, **194–199, 197**
- ClearPageFileAtShutdown value entry, 554
- client-server applications, **157–161, 158–160**
- clients, default, **387–388**
- Clients key, 59, **468–470, 469**
- Clip Redirector key, 558

- ClipArt Gallery, 346
- ClipArt Gallery key, 471
- Clipboard, **149–151**
 - for Access, **350**
 - copying key names to, 81
 - with DDE, 151–152
 - QuickEdit for, **313**
- Clipboard Formats key, **350**
- clnregui.cpp file, 195
- clock, Plug-and-Play identifiers for, 668
- Clone key, 60
- Close entry point for performance monitoring, 272
- closing registries, **85–86**
- cls command, 41
- CLSID key, 380–381
- CLSID view in RegMaid, **189–191**, 190
- CLSIDs, 374
 - for Desktop objects, 295–296
 - list of, **678–693**
 - mappings to programs for, 380–381
 - searching for applications by, 224–225
 - for Start menu entries, 298
- ClubWin support group, **636**
- cmdcons directory, **37–39**
- /cmdcons option, 36–37, 37
- CmdLine value entry in Setup, 576
- cmdline value entry in WOW, 568
- co-processors, 55, **439**
- Code Store Database key, 471
- CodePage key, **538**
- CoDeviceInstaller key, 529
- Cogswell, Bryce, 618
- [Collage Capture] section, 177
- Collect entry point for performance monitoring, 272
- color
 - for command prompt, **301–304**, 302–303
 - data type for, **602–603**
 - for display items, **411–412**
 - for Publisher, 357
 - RGB, 402
- Colors key, **411–412**
- Colors tab, 302, 303
- ColorSchemes key, 357
- ColorTable value entries, **301–304**, 302–303, **400**
- columns in CoolSwitch dialog box, **287**
- COM (Common Object Model) associations, 59
- COM+ Class registration database, 23
- Com Name Arbiter key, 529

- COM ports, **335–337**
- COM3 key, 471
- Comic Chat, 474, 639
- Comic Chat control, **161–168**, 162
- command key, 379, 383–384, 384
- command line, RegEdit editor from, **82–83**
- Command Processor key, 301, 471
- command prompt, **300–301**
 - colors for, **301–304**, 302–303
 - cursors for, **305–306**, 305
 - fonts for, **306–309**, 306
 - history buffer for, **311–312**
 - insert/overwrite modes in, **312–313**
 - memory for, **304–305**
 - window appearance, **309–311**, 309
- command prompt mode, 208, 211
- Comment value entry
 - in RDP-Tcp, 564
 - in WinStations, 562
- common.adm template, 136
- common data types, 600
- Common key
 - for system configuration, **353–354**
 - for user configuration, **359–360**
- Common Desktop key
 - in Shell Folders, 482
 - in User Shell Folders, 483
- Common Object Model (COM) associations, 59
- Common Programs key
 - in Shell Folders, 482
 - in User Shell Folders, 483
- Common Start Menu key
 - in Shell Folders, 482
 - in User Shell Folders, 483
- Common Startup key
 - in Shell Folders, 482
 - in User Shell Folders, 483
- companies for help, 619
- comparing registries, 69
- compatibility, Plug-and-Play identifiers for, 667
- Compatibility key
 - in CurrentVersion, **503–504**
 - in WOW, 517
- Compatibility2 key, **503–504**
- Compatibility32 key, **503–504**
- CompletionChar value entry, **313**
- Component Information value
 - in CentralProcessor, 438

- in FloatingPointProcessor, 439
 - in MultifunctionAdapter, 440
- compound data, **149**
- Computer Configuration policies, 124
- Computer Is Docked option, 437
- Computer Is Undocked option, 437
- Computer Management tool, 435, 436
- ComputerName key, 530
- ComputerName value entry, 530
- computers
 - accounts for, **453**
 - policies for, **135**
- conditional processing, 366
- Conferencing key, 471
- config directory, **18–21**
- Config.nt file, 17–18, 28
- ConfigFlags value entry, 569
- configuration
 - in HKEY_CURRENT_CONFIG, 50, **66, 577**
 - Software key, **577–578**
 - System key, **578–580**
 - machine. *See* HKEY_LOCAL_MACHINE hive
- Configuration Data value entry
 - in CentralProcessor, 438
 - in FloatingPointProcessor, 439
 - in MultifunctionAdapter, 440
- Configuration file value entry, 540
- Confirm on Delete option, **103–104**
- Confirmation value entry, 340
- Connection Manager Administration Kit key, 471
- connections
 - ghosted, **318–319**
 - LAN, 581
 - persistent, **318–319**
 - for printers, 426
 - remote registry, **77, 77**
- Connections key, 426
- ConnectionType value entry, 425
- connectoids, 426
- ConnectType value entry, 561
- console, **300–301**
 - colors for, **301–304, 302–303**
 - cursors for, **305–306, 305**
 - for default sessions, **400–402**
 - fonts for, **306–309, 306**
 - history buffer for, **311–312**
 - insert/overwrite modes in, **312–313**
 - for logon script window, **314**

- window appearance, **309–311**, 309
- Console key
 - in CurrentVersion, 504
 - in HKEY_CURRENT_USER, **300–301**, **400–402**
 - in HKEY_USERS, 64
 - in WinStations, **564**
- Console menu, Save As command, 133
- consultants, **619–620**
- Contact key, 361
- contacts
 - in Clients, 468
 - in Outlook, **355**, 361
- ContentIndex key, 530
- ContentIndexCommon key, 530
- context-clicks, 80
- context menus
 - modifying, **382–386**, 383–385
 - removing items from, **332**
- context switching in OLE, 159
- Control.ini file, 234
- Control key in CurrentControlSet, **526**
 - in HKEY_CURRENT_CONFIG, **578–579**
 - in HKEY_LOCAL_MACHINE
 - ApmActive key, **526**
 - ApmLegalHal key, **527**
 - Arbiters key, 527
 - BackupRestore key, 527
 - Biosinfo key, 527
 - BootVerificationProgram key, **527**
 - Class key, **527–529**
 - CoDeviceInstaller key, 529
 - Com Name Arbiter key, 529
 - ComputerName key, 530
 - ContentIndex key, 530
 - ContentIndexCommon key, 530
 - CrashControl key, **530–531**
 - CriticalDeviceDatabase key, 530
 - DeviceClasses key, 530
 - in Enum, 336
 - FileSystem key, **531–532**
 - GraphicsDrivers key, **532–533**
 - GroupOrderList key, **533–534**
 - Hivelist key, **534**
 - IDConfigDB key, **534–535**
 - Keyboard Layout key, **535**
 - Keyboard Layouts key, **535–536**
 - LSA key, **536**
 - MediaCategories key, 536

MediaInterfaces key, 536
 MediaProperties key, **536**
 MediaResources key, **537**
 MediaSets key, 537
 Network key, 537
 NetworkProvider key, **537**
 NLS key, **537–538**
 NTMS key, 538
 Pnp key, 539
 Print key, **539–543**
 PriorityControl key, **544**
 ProductOptions key, **544**
 SecurePipeServers key, **544–545**
 SecurityProviders key, **545–547**
 ServiceGroupOrder key, **547–548**
 ServiceProvider key, **548–550**
 Session Manager key, **550–555**
 SessionManager key, **556**
 Setup key, **556**
 StillImage key, 556
 SystemResources key, **556–557**
 Terminal Server key, **557–566**
 TimeZoneInformation key, **566–567**
 Update key, **567**
 VirtualDeviceDrivers key, **567–568**
 Windows key, **568**
 WOW key, **568–569**

Control Panel

for CrashControl, 530–531, 530
 for hardware profiles, 436
 for Keyboard Layout, 423
 for networking, 316–317
 policies for, 137
 for printers, 541
 for removing registry objects, 220, **497–498**
 for sounds, 397–398, 398
 special folder for, 298
 for user interface, 278
 for user profiles, 88

Control Panel key, 65, **403–404**

Accessibility key, **404–410**
 Appearance key, **410**
 Cache key, 411
 Colors key, **411–412**
 Current key, 412
 in CurrentVersion, **480**
 Custom Colors key, 412

- Desktop key, **412–416**
- International key, **416–417**
- IOProcs key, **417**
- Keyboard key, **417**
- MMCPL key, **417–418**
- Mouse key, **418**
- Patterns key, **419**
- PowerCfg key, **419–420**
- Screen Saver.3DflyingObj key, 420
- Screen Saver.3Dpipes key, 420
- Screen Saver.Bezier key, 420
- Screen Saver.Marquee key, 421Screen Saver.Mystify key, 421
- Screen Saver.Stars key, 421
- Sound key, 421
- Sounds key, **421**
 - in UserOverride, 566
- controllers, IDE, **332–334**
- Controls Folder key, **481**
- Controls key, 482
- ControlSet001 key, 60, 213, 575, 594
- ControlSet002 key, 60, 213, 575, 594
- ControlSet003 key, 61, 213, 594
- ControlSet004 key, 594
- ControlSets, **213, 524**
- ConvertedToLinks value entry, 519
- Cookies key, 485
- CoolSwitch value entry, **286**, 413
- CoolSwitchColumns value entry, **287**, 413
- CoolSwitchRows value entry, **288**, 413
- Copy command, 41
 - for backups, **24–28**
 - for restoring, **31–32**
- copy function, **110–111**
- copying keys with RegEdit, **81**
- copyrights
 - for themes, 280
 - for Type 1 fonts, 512
- Copyrights key, 512
- Count object
 - in REG_RESOURCE_LIST, 611
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- CounterOffset field, 269–270
- counters for performance monitoring, **256–259**, 258, 260, **642–662**, 642
- CounterSize field, 269–270
- CounterType field, **269–271**, 270
- Country List key, 494–495
- CountryCode key, 495
- CPUs, 54–55, **438–439**

- CrashControl key, **530–531**
- CrashDumpEnabled value entry, 531
- crashing systems, stabilizing, **214–217**
- Create Link access setting, 98
- Create Link events, 100
- Create Subkey access setting, 98
- Create Subkey events, 100
- CREATOR OWNER group, 64
- CriticalDeviceDatabase key, 530
- CriticalSectionTimeout value entry, 550
- CRS key, 471
- Cryptography key
 - in Last Update, 486
 - in Microsoft, 471
- CSCSettings key, 481
- CSDVersion value entry, 501, 568
- CSSFilters key, 481
- curly braces ({})
 - for CLSIDs, 298
 - for GUIDs, 374
- current configuration in HKEY_CURRENT_CONFIG, 50, **66**, **577**
 - Software key, **577–578**
 - System key, **578–580**
- .Current key for sounds, 399
- Current key in Control Panel, 412
- Current value entry, 576
- CurrentBuild value entry, 501
- CurrentBuildNumber value entry, 501
- CurrentConfig value entry, 534
- CurrentControlSet key in System
 - in HKEY_CURRENT_CONFIG, **578–580**
 - in HKEY_LOCAL_MACHINE, 7, 60–61, 213, **525–526**, **594**
 - Control key. *See* Control key in CurrentControlSet
 - Enum key, **569–573**
 - Hardware Profiles key, **573–574**
 - Services key, **574**
- CurrentPage value entry, 304
- CurrentType value entry, 501
- CurrentUser value entry, 526
- CurrentVersion key, 329–330
 - in Windows
 - AdminDebug key, 479
 - App Management key, 480
 - App Paths key, **480**
 - Applets key, 480
 - Control Panel key, **480**
 - Controls Folder key, **481**
 - CSCSettings key, 481

Dynamic Directory key, 481
Explorer key, **481–483**
Extensions key, **484**
ExtShellViews key, 484
Group Policy key, 484
H323TSP key, 484
Installer key, 484
Internet Settings key, **484–489**
IPConfMSP key, 489
IPConfTSP key, 489
MCD key, **489**
ModuleUsage key, 489
MS-DOS Emulation key, **489**
netcache key, 490
Nls key, 490
Policies key, **490–491**
RenameFiles key, **491**
Run key, **491–492**
RunOnce key, **492**
RunOnceEx key, 493
Setup key, **493**
SharedDlls key, **493**
Shell Extensions key, **494**
ShellScrap key, 494
ShellServiceObjectDelayLoad key, 494
StillImage key, 494
SyncMgr key, 494
Telephony key, **494–496**
Unimodem key, **497**
Uninstall key, **497–498**
URL key, **498–499**
WebCheck key, 499
Welcome key, 499
WindowsUpdate key, 499
WinLogon key, 499

in Windows NT, **501–502**

AeDebug key, **502–503**
Compatibility keys, **503–504**
Console key, 504
drivers.desc key, 504
Drivers key, **504**
Drivers32 key, 504
Embedding key, 504
File Manager key, 504
Font Drivers key, 505
FontCache key, **505**
FontMapper key, **505**
Fonts key, **505–506**

FontSubstitutes key, **506**
 GRE_Initialize key, **506**
 HotFix key, 506
 ICM key, 506
 Image File Execution Options key, **507**
 IniFileMapping key, 507
 MCI key, 507
 MCI Extensions key, 508
 MCI32 key, **508**
 Midimap key, **508**
 ModuleCompatibility key, **508**
 Network key, **509**
 NetworkCards key, **509–510**
 NTVersionOfLastBackup key, 510
 OpenGLDrivers key, 510
 Perflib key, **510**
 Ports key, 510
 Print key, 511
 ProfileGuid key, 511
 ProfileList key, **511**
 related.desc key, 511
 SeCEdit key, 511
 Srvhost key, 511
 SrvWiz key, 511
 Terminal Server key, 512
 Time Zones key, **512**
 Tracing key, 512
 Type 1 Installer key, **512–513**
 Userinstallable.drivers key, **513**
 Windows key, **513**
 Winlogon key, **513–517**
 WOW key, **517–518**

CurrentVersion value entry, 501
 CursorBlinkRate value entry, **289**, 413
 cursors
 blink rate for, **289**, 413
 for command-prompt window, **305–306**, 305
 for Office, 359
 Cursors key, 359
 CursorSize value entry, 305–306, 305
 Custom Colors key, 412
 Custom Labels key, 362
 custom user interface, policies for, 138
 CustomizableAlerts key, 350
 CWinApp class, **247–250**

D

- /D parameter, 83
- damaged software, 203
- Data file value entry, 540
- Data key, 362
- data sharing
 - Clipboard for, **149–151**
 - DDE for, **151–152**
 - OLE for. *See* OLE (Object Linking and Embedding)
- Data Size object in REG_FULL_RESOURCE_DESCRIPTOR, 608
- data types, 7–8, **50–53**, 50–53, **600–602**
 - with Clipboard, 149
 - with reg add, 109
 - REG_BINARY, **602**
 - REG_COLOR_RGB, **602–603**
 - REG_DWORD, **603**
 - REG_DWORD_BIG_ENDIAN, **603**
 - REG_DWORD_LITTLE_ENDIAN, **604–605**
 - REG_EXPAND_SZ, **605–606**
 - REG_FILE_NAME, **606**
 - REG_FILE_TIME, **606**
 - REG_FULL_RESOURCE_DESCRIPTOR, **606–608**
 - REG_LINK, **609**
 - REG_MULTI_SZ, **609**
 - REG_NONE, **609–610**
 - REG_QWORD, **610**
 - REG_QWORD_LITTLE_ENDIAN, **610**
 - REG_RESOURCE_LIST, **610–613**
 - REG_RESOURCE_REQUIREMENTS_LIST, **613–616**
 - REG_SZ, **616**
 - REG_UNKNOWN, **616**
 - with RegEdit, 74
 - with RegEdt32, 93
 - for value entries, 5
- data values. *See* value entries; values
- DATA1.msi file, 347
- DataAccess key, 471
- database program configuration information, **350–353**
- Database Splitter, 351
- DataFactor key, 471
- Datatype value entry
 - in Drivers, 540
 - in Printers, 542
- Dataviz key
 - in Outlook, 355
 - for user configuration, 361
- Daylight Saving Time (DST), 512
- DaylightBias value entry, 566
- DaylightName value entry, 566

- DaylightStart value entry, 567
- DcacheMinInterval value entry, 338, 516
- DCacheUpdate value entry, 514
- DCI key, 532
- DDE (Dynamic Data Exchange), 149, **151–152**
- Debug value entry, 555
- debugger, AeDebug key for, **502–503**
- Debugger value entry, 502
- debugging modes, booting in, **208–214**, 208
- DebugServerCommand value entry, 514
- DEF\$\$\$\$.DEL file, 19
- Default Beep value, 421
- Default DevMode value entry, 542
- default files, 18–19, 34
- .DEFAULT key
 - in HKEY_CURRENT_USERS, 399
 - in HKEY_USERS, 49–50, 62, **594–597**
- Default Priority value entry, 542
- Default.sav file, 19
- Default Save key, 362
- Default Service value entry, 529
- Default Taskbar value entry, 482
- Default Toolbars value entry, 482
- .DEFAULT user, 6
- Default User directory, 67
- default user profiles
 - editing, **67–69**
 - RegEdt32 for, **87–89**
- (Default) value entry in Class, 529
- .DEFAULT value entry in HKEY_USERS, 430
- .Default value entry in Ratings, 490
- Default value entry in Select, 576
- DefaultDomainName value entry, 514
- DefaultExtension key, 164
- DefaultIcon key, 164, 296–297, 378
- DefaultPassword value entry, 516
- DefaultPrefix value entry, 498
- defaults
 - clients, **387–388**
 - handlers, 389
 - install path, **520**
 - remote access, **582–583**
 - zones, 489
- DefaultSecurity value entry, 562
- DefaultSeparateVDM value entry, 569
- DefaultSettings.BitsPerPel value entry, 580
- DefaultSettings.Flags value entry, 580
- DefaultSettings.Vrefresh value entry, 580

- DefaultSettings.Xpanning value entry, 580
- DefaultSettings.XResolution value entry, 580
- DefaultSettings.Ypanning value entry, 580
- DefaultSettings.YResolution value entry, 580
- DefaultUserConfiguration key, **558–559**
- DefaultUserName value entry, 514
- DelayBeforeAcceptance value entry, 405
- delays
 - keyboard, 405, 417
 - menu display, **289–290**
 - networking, **341**
 - object loading, 494
- Delete access setting, 98
- delete command, 41
- Delete events, auditing, 100
- delete function, **111–112**
- delete option for RegEdt32 editor, **103–104**
- deleted objects and applications, **221**
- DeleteRoamingCache value entry, 516
- DeleteTempDirsOnExit value entry, 557
- deleting
 - keys, **94–95**
 - registry entries, **182**
 - CleanReg for, **194–199**, 197
 - context menu items, **332**
 - in disaster recovery, **220**, **224–225**
 - RegClean for, **183–186**, 184–185, **199**
 - RegMaid for, **187–194**, 190–191
 - software, problems from, 203
- DelRegTree function, 250
- Dependent files value entry, 540
- Description key, 49, **54–55**, 60, **438–440**, 440, **588–589**
- Description value entry
 - in NetworkCards, 509
 - in Printers, 542
- descriptions for drivers, 504
- Desktop, **279–280**
 - background and wallpaper, **280–285**
 - cursor, **289**
 - icons, **295–298**, 297
 - keyboard, **290–292**
 - menus, **289–290**
 - mouse, **292–294**
 - moving windows, **287–288**
 - policies for, 137
 - task switching, **285–287**
- Desktop key
 - in Control Panel, 280, **412–416**, 566

- in Explorer, 481
- in Streams, 482
- Device Parameters key, **336–337**
- DeviceClasses key, 530
- DeviceManager key, 471
- DeviceMap key, 55–56, 59, **441–444, 589–590**
- DeviceName value entry, 561
- DeviceNotSelectedTimeout value entry, 513
- devices, startup modes for, 574
- [Devices] section, 178
- DeviceSpecific key, 497
- DevModes2 key, 426
- DevModesPerUser key, 426
- Dfrg key, 471
- DHCP (Dynamic Host Configuration Protocol), **339**
- DHCP.prm file, 339
- DHCPInfo00 key, 339
- DhcpMibAgent key, 471
- DHCPServer key, 471
- diagnostic software, **206**
- Dial-Up Networking component
 - policies for, 141
 - special folder for, 298
- dialog boxes, ID numbers for, 379
- DigitalProductID value entry, 501
- dir command, 41
- Direct Memory Access (DMA), 56, 444, 446, 665
- direct switching, 286
- Direct3D key, 471
- DirectDraw key, 471
- DirectInput key, 471
- DirectMusic key, 472
- directories
 - backing up, 23
 - renaming, 214
 - Windows, moving, **321–328**
- Directory Services Restore Mode option, 31, 209
- Directory value entry, 568
- DirectPlay key, 472
- DirectX key, 472
- disable command, 41
- disablecad key, 491
- DisableLockWorkstation key, 499
- DisableNT4Policy key, 491
- DisablePagingExecutive value entry, 554
- DisableUSWC value entry, 533
- disaster prevention, **16–20**
 - making backups, **21–29**

- Recovery Console for, **36–43**, 37
- restoring backups, **30–35**
- disaster recovery, **202**
 - analysis in, **219–220**
 - application errors in, **222**
 - boot floppies for, **227**
 - deleted objects in, **221**
 - drive errors in, **223–224**
 - fixing registry items in, **220**
 - overwritten application files in, **221–222**
 - removing registry items in, **224–225**
 - repair vs. replace decisions in, **203–204**
 - repairing registry in, **226**
 - stabilizing system in, **204–219**
- Disconnect Network Drive command, **332**
- disk controllers, Plug-and-Play identifiers for, 667
- DISK key, 61, **575**, 594
- disk101 file, 39
- diskpart command, 41
- disks in paths, 331
- DiskSpaceThreshold value, 331
- display adapters, Plug-and-Play identifiers for, 667–668
- Display applet, 278, 410, 481
- display delay for menus, **289–290**
- DISPLAY key, 569
- Display Properties dialog box, 295, 300
- Display value entry, 512
- DisplayName value entry, 498, 543
- DisplayPublisherOptionsDialogBox function, 365
- DLL Addins key, 356
- DLL (dynamic link library) files
 - AppInit_DLLs value entry for, 513
 - icons for, **299**
 - SharedDlls key for, **493**
- Dlt value entry, 512
- DMA (Direct Memory Access), 56, 444, 446, 665
- Dns Timeout value entry, 542
- DnsEvent.Evt file, 19
- Docking State Is Unknown option, 438
- docking states for portable computers, 437–438
- DocObject technology, 164, 167
- Document Routing key, 356
- dollar signs (\$) in machine accounts, 453
- Domain Admins value, 450
- “domain controller for your domain could not be contacted...” message, 456
- domain controllers, 447
- domain group policy objects, 123
- domain groups in Account, 448–451

- Domain key, 459
- domain lists, refresh intervals for, **338–339**
- Domain Users value, 450
- Domain value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 564
 - in WinStations, 562
- Domains key in SAM, 447
 - Account key, **447–454**
 - Builtin key, **454–455**, 454
- Domains value entry, 488
- dontdisplaylastusername key, 491
- DontDisplayLastUserName value entry, 514
- DOS Devices key, **553–554**
- Dos key, 559
- DosKey program, 311
- DosKeybCodes key, 535
- DosKeybIDs key, 535
- DoubleClickHeight value entry, **293**, 418
- DoubleClickSpeed value entry, 293, 418
- DoubleClickWidth value entry, **293**, 418
- DOWNLOAD key
 - in SO, 486
 - in SOIEAK, 487
- Downloaded Program Files key, 483
- DownloadManager key, 472
- DRAGDROP key
 - in SO, 487
 - in SOIEAK, 487
- DragFullWindows value entry, **287–288**, 413
- dragging, **287–288**
- DragHeight value entry, **287–288**, 413
- DragWidth value entry, **287–288**, 413
- Draw program, 346, 360
- DriveControl value entry, 334
- Driver Signing key, 472
- Driver value entry, 540
- [driver32] section, 176
- drivers
 - font, 505
 - help for, 619
 - load order for, 547–548
 - modem, **497**
 - multimedia, 508
 - in OLE, failures in, 172
 - print, **540**
- Drivers key

- in CurrentVersion, **504**
- in Environments, **540**
- drivers.desc key, 504
- [drivers] section, 176
- Drivers32 key, 504
- drives and drive letters
 - for backups, 24
 - for CD-ROM drives, **330**
 - free space on, **331**
 - mapped, 424–426, 425
 - read errors in, **223–224**
 - restoring from, **30–31**
 - in Win.ini, 179
- DrWatson key, 472
- DrWatson settings, **503**
- DST (Daylight Saving Time), 512
- DumpFile value entry, 530
- duplication
 - in history buffer, **312**
 - in registry, 6–7
- Dvorak keyboard, 424, 536
- DWORD Editor window, 52, 52
- DWORDs, **52**, 402–403
 - big endian, **603**
 - little endian, **604–605**
- dynamic data, 11
- Dynamic Data Exchange (DDE), 149, **151–152**
- Dynamic Directory key, 481
- Dynamic Host Configuration Protocol (DHCP), **339**
- dynamic link library (DLL) files
 - AppInit_DLLs value entry for, 513
 - icons for, **299**
 - SharedDlls key for, **493**

E

- e-mail, **355**, **468–470**, 469
- /E parameter, 83
- Easter Eggs, **300**
- Edit File Type dialog box, 186
- edit key, 379
- Edit menu, 74
 - Add Key command, 92
 - Add Value command, 93
 - Binary command, 94
 - Copy Key Names command, 81
 - Delete command, 94
 - DWORD command, 94
 - Find command, 81–82

- Find Next command, 81–82
- Modify command, 80
- Multi String command, 94
- New menu, 80
- Rename command, 80
- String command, 94
- Edit verb, 166
- EditFlags variable, **376–378**
- editing
 - boot.ini file, **26–27**
 - default user profiles, **67–69**
 - registry. *See* RegEdit editor; RegEdt32 editor
- Editing Action for Type dialog box, 186
- editors for policies, **142–143**
- 8.0 key, 348
- Eisa value entry, 556–557
- EisaAdapter value entry, 439
- embedding
 - applications, 504
 - HKEY_CLASSES_ROOT for, **386, 387**
 - in OLE, **154, 170–171**
- Embedding key, 504
- Emergency Repair Disk option, 105–106
- emergency repair diskettes (ERDs)
 - contents of, **28**
 - creating, **105–106, 105**
 - limitations of, 17
 - vs. repair disks, 44
 - for restoring, **32–34**
 - scheduling, **44**
- Enable Boot Logging option, 208–209
- enable command, 41
- Enable Indicator on Taskbar option, 423
- ENABLE key
 - in SO, 486
 - in SOIEAK, 487
- Enable Key Sequence option, 423
- Enable VGA Mode option, 209
- EnableAutodial value entry, 574, 578
- EnableMCA value entry, 550
- EnableMCE value entry, 550
- encryption, **546–547**
- End User Defined Characters, 65, 423
- EnterpriseCertificates key, 472
- Enum key
 - in CurrentControlSet, **579**
 - DISPLAY key, 569

- FDC key, 569
- HTREE key, 569
- IDE key, **569–570**
- ISAPNP key, 570
- PCI key, 570
- PCI_HAL key, 570
- PCIIDE key, 570
- Root key, **570–572**
- SCSI key, 572
- STORAGE key, 572
- SW key, 572–573
- USB key, 573
- Enumerate Subkeys access setting, 98
- Enumerate Subkeys events, auditing, 100
- EnvelopeHost utility, 468
- Envelopes key, 357
- Environment key, 65, **422**, 422, 554
- environment variables, **18**
 - in Environment key, **422**, 422
 - in REG_EXPAND_SZ, **605–606**
- Environments key, **540**
- Equation Editor, 346
- ERDs. *See* emergency repair diskettes (ERDs)
- error handling, **245**
- ErrorMode value entry, 568
- ESDI controllers, 333
- ESENT key, 472
- Ethernet networks, **339–340**
- EUDC key, 65, 423
- Eudora program, 469
- Event Labels key, 395–397
- Event Viewer, failures in, 172
- Event Viewer key, 340
- EventLabels, 64
- EventLog value entry, 543
- events, **100–101**, **395–399**
- EventSystem key, 472
- EVERYONE group, 64
- Excel key, 354
- Excel program
 - DDE with, 151–152
 - user configuration information for, **360**
- Exchange key, 472
- ExcludedProviders value entry, 549
- ExcludeFromKnownDlls value entry, 550
- Executive key, 554
- exit command, 41
- expand command, 41, 330

- expandable strings, **51, 605–606**
- ExpandLogonDomain value entry, 509
- expiration messages for passwords, **338**, 516
- Explorer
 - context menu in, 332
 - OLE used by, 170
- Explorer key
 - in CurrentVersion, **481–483**
 - in Policies, 490
- Export Modules key, 356
- Export Registry File dialog box, 29
- exporting registry items, 69, **74–76**, 75
- ExtendedSounds value entry, 421
- extensions
 - keys for, 59, **374–386**, 484
 - in Win.ini, 177
- Extensions key, **484**
- [extensions] section, 177–178
- ExtShellViews key, 484

F

- F key
 - in Network, 425
 - in Users, 452
- FaceName value entry, **307**, 400
- Failed value entry, 576
- Failure events, auditing, 101
- FAT (File Allocation Table) system
 - backing up, **24–25**
 - boot floppies for, **227**
 - moving Windows directory in, **322–325**
 - restoring, **31–32**
 - types of, 23
- fAutoClientDrivers value entry
 - in RDP-Tcp, 564
 - in rdpwd, 561
- fAutoClientLpts value entry
 - in RDP-Tcp, 564
 - in rdpwd, 561
- Favorites list, **294**
- Favorites menu, 74
- Fax key, 472
- FDC key, 569
- fDisable value entries
 - in RDP-Tcp, 564
 - in rdpwd, 561
- Federal Express key, 58

- fEnable value entries
 - in rdpwd, 561
 - in WinStations, 562
- fEnableWinStation value entry, 564
- fFlowSoftwareRx value entry, 561
- fFlowSoftwareTx value entry, 561
- fForceClientLptDef value entry
 - in RDP-Tcp, 564
 - in rdpwd, 561
- fHomeDirectory value entry, 564
- File Allocation Table system. *See* FAT (File Allocation Table) system
- file extensions
 - keys for, 59, **374–386**, 484
 - in Win.ini, 177
- File Manager key, 504
- File menu
 - Compare Files command, 69
 - New command, 235
 - Server Properties command, 541
- FILE ports, 335
- File Rep.evt file, 19
- file systems, policies for, 140
- File Types tab, 186
- FILEDOWNLOAD key
 - in SO, 486
 - in SOIEAK, 487
- filenames, **195–199**, **532**
- FileNew key, 359
- FileRenameOperations key, 554
- files for hives, 5
- [files] section, 177
- FileSystem key, **531–532**
- filetype associations
 - keys for, 59, **374–386**, 484
 - in Win.ini, 177
- FileTypesPropertySheetHook key, 481
- Find dialog box
 - in RegEdit, 82, 82
 - in RegEdt32, 95, 95
- Finders key, 500
- FindExtensions key, 482
- FindFast shortcut, **368**
- finding. *See* searching
- fInherit value entries
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 564–565
 - in WinStations, 562–563
- fInheritAutoClient value entry, 561

- FirstCountMsgQPeeksSleepBadApp value entry, 557
- Fix Errors button, 185, 185, 199
- fixboot command, 42
- fixmbr command, 42
- Flags object
 - in REG_FULL_RESOURCE_DESCRIPTOR, 608
 - in REG_RESOURCE_LIST, 612
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Flags value entry
 - in Keyboard Response, 405
 - in MouseKeys, 405
 - in ProfileList, 511
 - in SerialKeys, 406
 - in SoundSentry, 406
 - in StickyKeys, 406–409
 - in TimeOut, 409
- FloatingPointProcessor key, **55**, **439**
- fLogonDisabled value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- FlowHardwareRx value entry, 561
- FlowHardwareTx value entry, 561
- FlowType value entry, 561
- folders, policies for, 138
- Font dialog box, 102, 103
- Font key, 489
- Font Drivers key, 505
- Font tab, 306, 306
- FontCache key, **505**
- FONTDOWNLOAD key
 - in SO, 486
 - in SOIEAK, 487
- FontFaceName value entry, 340
- FontFamily value entry, **307–308**, 401
- FontHeight value entry, 341
- FontItalic value entry, 341
- FontMapper key, **505**
- fonts
 - Adobe Illustrator, **512–513**
 - for command-prompt window, **306–309**, 306
 - in CurrentVersion, **505–506**
 - for default sessions, 400–401
 - for Desktop, 413
 - drivers for, 505
 - in GRE_Initialize, **506**
 - for MS-DOS applications, 489
 - for networking, 340–341

- for RegEdt32, **102**, 103
- in Win.ini, 177
- Fonts key, **505–506**
- [fonts] section, 177–178
- FontSize value entry, **308**, 401
- FontSmoothing value entry, 413
- FontSubstitutes key, **506**
- [fontSubstitutes] section, 178
- FontWeight value entry, **308–309**, 341, 401
- ForceFifoEnable value entry, 336
- foreground color, **301–304**, 302–303
- ForegroundFlashCount value entry, 413
- ForegroundLockTimeout value entry, 413
- format command, 42
- FormatMessage function, **245–246**
- formats
 - for Access reports, 351
 - for Clipboard, **150–151**, **350**
- FORMDATA key
 - in SO, 487
 - in SOIEAK, 488
- Forms key, **540–541**
- fPromptForPassword value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- FPU (floating point units), **55**, **439**
- fReconnectSame value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- free space on drives, **331**
- FreeAgent newsgroup reader, 637
- fResetBroken value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- frmPublisherOptions dialog box, 365
- FrontPage key, 472
- FSTextEffect value entry, 406
- FTP, MSFTPCVC key for, 550
- ftp prefix for URLs, 499
- Full Control access setting, 57
- FullScreen value entry, **309–310**, 401
- fUseDefaultGina value entry
 - in RDP-Tcp, 565
 - in WinStations, 563

G

General key

- for system configuration, 354
- for user configuration, 359

General tab, 423

GetAppRegistry function, 248, 250

GetPrivateProfileInt function, 234

GetPrivateProfileSection function, 234

GetPrivateProfileSectionNames function, 234

GetPrivateProfileString function, 234

GetPrivateProfileStruct function, 234

GetProfileBinary function, 249

GetProfileInt function, 234, 249

GetProfileSection function, 234

GetProfileString function, 234, 249

GetSectionKey function, 248, 250

GFLAG.EXE program, 550, 551

ghosted connections, **319–320**

global groups in Account, 449–451

GlobalFlag value entry, **550–552**

Globally Unique IDs (GUIDs), **372–374, 527–529**

GlobalPowerPolicy key, 420

gopher prefix for URLs, 499

GOPHERSVC key, 549

GPOs (group policy objects), 122

Graph program, 346, 360

Graphics Rendering Engine (GRE), **506**

GraphicsDrivers key, **532–533**

GRE_Initialize key, **506**

GridGranularity value entry, 413

Group Policy key, 484

group policy objects (GPOs), 122

Group Policy tab, 131, 132

Group Policy window, 126, 127, 131, 132

GroupOrderList key, **533–534**

groups

- built-in, 63
- in Policy, **457–459**
- special, 64

Groups key

- in Account, **448–451, 451**
- in Builtin, 455

GUESTS group, 63

H

H/PC Connection Agent value entry, 492

H323TSP key, 484

- HAL (Hardware Abstraction Layer)
 - in boot process, 10
 - in ResourceMap, 444
- Handsafe.reg file, 348
- Handunsf.reg file, 348
- hardware
 - data type for, **52, 606–608**
 - in stabilizing system, **218–219**
- Hardware Abstraction Layer (HAL)
 - in boot process, 10
 - in ResourceMap, 444
- Hardware key, 49, **54, 435**
 - Description key, **54–55, 438–440, 440**
 - DeviceMap key, **55–56, 441–444**
 - profiles in, **436–438, 437**
 - ResourceMap key, **56, 444–446**
 - summary of contents, **588–591**
- Hardware Profile/Configuration Recovery Menu, 211, *211*
- hardware profiles, **436–438, 437**
- Hardware Profiles dialog box, 212, *212*
- Hardware Profiles key, **573–574**
- Hardware Profiles tab, 436
- Hashes key, 546
- HeapDeCommitFreeBlockThreshold value entry, 552
- HeapDeCommitTotalFreeThreshold value entry, 552
- HeapSegmentCommit value entry, 552
- HeapSegmentReserve value entry, 552
- height
 - of drag rectangle, **287–288, 413**
 - of fonts, 341
- help
 - Assistant for, 353, 353
 - chat for, **638–639, 638**
 - for command prompt, 301
 - companies and magazines for, 619
 - consultants for, **619–620**
 - newsgroups for, **637**
 - stores for, **620**
 - training, **636**
 - user groups for, **620–635**
 - virtual support groups for, **636**
 - Web for, **618–619**
 - in Windows, **500**
 - in Word, 362
- help command, 42
- Help File value entry, 540
- Help key
 - in Windows, **500**

- in Word, 362
- HelpViewer key, 354
- hexadecimal numbers, **402–403**
- HEY_PERFORMANCE_DATA hive, **66**
- Hide Protected Operating Systems Files option, 490
- hiding logon script window, **314**
- Hipson, Peter D., hints by
 - application migration, **251**
 - hivelist, **462–463**
 - network delay, **341**
 - startup jobs, **520**
- history, command, **311–312**, 401
- History key, 486
- HistoryBufferSize value entry, **311**, 401
- HistoryNoDup value entry, **312**
- hivelist key, 462, **534**
- hives, **5–7**, **46–48**. *See also* keys; value entries
 - definition, **11–12**
 - exporting, **74–76**, 75
 - importing, **75–76**
 - restoring, with RegEdt32, **91**
 - saving, **89–90**, 90
- HK abbreviations, 6
- HKEY_CLASSES_ROOT hive, 48, **372**
 - for file extensions, **374–386**
 - for OLE, **386**, 387
 - for unknown files, **388–389**, 388
- HKEY_CURRENT_CONFIG hive, **50**, **66**, **577**
 - Software key, **577–578**
 - System key, **578–580**
- HKEY_CURRENT_USER hive, 48, **392–395**
 - as alias, 6–7
 - AppEvents key, **395–399**
 - Console key, **400–402**
 - Control Panel key, **403–421**
 - Environment key, **422**, 422
 - EUDC key, 423
 - Identities key, 423
 - Keyboard Layout key, **423–424**
 - Network key, **424–426**, 425
 - Printers key, **426**
 - RemoteAccess key, **426**
 - Software key, **427–428**
 - Unicode Program Group key, **429**
 - Volatile Environment key, 429
 - Windows 3.1 Migration Status key, **429**
- HKEY_DYN_DATA hive, 6, 66
- HKEY_LOCAL_MACHINE hive, **49**, **53–54**, **434–435**

- Hardware key, **54–57**, **435–446**, 437, 440, 446
- permanent value entries in, **462–463**
- SAM key, **57**, **447–456**, 449, 451–452, 454
- Security key, **58**, **456–460**
- Software key, **58–60**. *See also* Software key
- summary of contents, **588–594**
- System key, **60–62**, 317. *See also* System key
- HKEY_PERFORMANCE_DATA hive, **254–271**
- HKEY_USERS hive, **49–50**, **62–66**, **392–393**, 430
 - alias for current user in, 6–7
 - summary of contents, **595–597**
- Home/Office Desk power policy, 420
- home prefix for URLs, 499
- Home Publishing key, 472
- HostMIB key, 472
- HotFix key, 506
- hotkeys for locales, 423
- HTML key, 357
- HTML Help Collections key, 472
- HTML value entry, 350
- HTMLCharacterEncodings key, 357
- HTREE key, 569, **579**
- HungAppTimeout value entry, 413

I

- IASAgent key, 472
- ICM key, 506
- Icon value entry, 529
- IconFont value entry, 415
- icons
 - bit-planes for, 416
 - Cache key for, 411
 - in Class, 529
 - custom, **295–298**, 297
 - for Desktop, 413–414
 - for DLL files, **299**
 - dragging, 287
 - ID numbers for, 379
 - with wallpaper, 281–282
- IconSpacing value entry, 413, 415
- IconTitleFaceName value entry, 414
- IconTitleSize value entry, 414
- IconTitleStyle value entry, 414
- IconTitleWrap value entry, 414, 416
- IconVerticalSpacing value entry, 416
- ICW key, 499
- ID value entry, 430

- IDConfigDB key, **534–535**
- IDE drives
 - controllers for, **332–334**
 - in Scsi, 441–444
- IDE key, **569–570**
- Identifier value entry
 - in CentralProcessor, 439
 - in FloatingPointProcessor, 439
 - in MultifunctionAdapter, 440
- Identities key, 423
- IdleWinStationPoolCount value entry, 557
- IE Setup key, 472
- IE4 key, 472
- IEAK (Internet Explorer Administration Kit), **487–488**
- IEXPLOREV2 value entry, 486
- IEXPLOREV3 value entry, 486
- IEXPLOREV4 value entry, 486
- If statements, 366
- IGMPMibAgent key, 473
- IIS (Internet Information Server), 550
- Image File Execution Options key, **507**
- Import keys, 363
- importing registry items, **75–76**
- ImprocHandler32 key, 164
- in-process handlers, 164
- Inbox icons, 295
- Indexing Service, counters for, 259
- Indexing Service Filter, counters for, 259
- INetMgr key, 473
- Inetsrv key, 473
- INetStp key, 473
- InfName value entry, 510
- InfOption value entry, 510
- InfoViewer key, 473
- .ini files, 4, 230–231
 - API functions for, 233–234
 - mapping, 234, 507
 - System.ini file, 4, **174–176**
 - Win.ini file, 4, **174–179**
- IniFileMapping key, 175, 234, 507
- IniFiles key, 60
- Init Commands key, 360
- Init Menus key, 360
- initial settings for new users, **431**
- InitialKeyboardIndicators value entry, **290**, 417
- InitialProgram value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565

- in WinStations, 563
- InProcServer32 key, 381
- Input Locales tab, 423
- InputBufferLength value entry
 - in RDP-Tcp, 565
 - in Tds, 561
 - in WinStations, 563
- Insert menu, Object command, 158
- insert mode in command-prompt window, **312–313**
- Insert New Object dialog box, 164, 167
- Insert Object dialog box, 167
- InsertMode value entry, **312–313**, 401
- Inspect boot sector option, 33
- Inspect registry files option, 33
- Inspect start-up environment option, 33
- install path for Windows 2000, **520**
- Installation Sources value entry, 330
- InstallDate value entry
 - in CurrentVersion, 501
 - in NetworkCards, 509
- INSTALLDT key
 - in SO, 487
 - in SOIEAK, 488
- installed applications in stabilizing system, **218**
- Installer key, 484
- Installer32 value entry, 529
- installing
 - Recovery Console, **36–37**, 37
 - software, problems from, 203
- InstallRoot key
 - in Access, **350**
 - in Common, 354
 - in Outlook, 355
 - in PowerPoint, 356
- “Insufficient privilege to save the key” message, 90
- INTEL key, 58
- Intellimouse, **292–294**
- IntelliPoint key
 - in Microsoft, 473
 - in Uninstall, 498
- inter-application transfers
 - Clipboard for, **149–151**
 - DDE for, **151–152**
 - OLE for. *See* OLE (Object Linking and Embedding)
- INTERACTIVE group, 64
- InteractiveDelay value entry
 - in RDP-Tcp, 565
 - in Tds, 561

- InteractiveMusic key, 473
- interface, policies for, 138
- Interface Type object
 - in REG_FULL_RESOURCE_DESCRIPTOR, 607
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- Interface view in RegMaid program, **193**
- Interior Alaska Windows NT Users Group, 619
- International key, **416–417**
- InternationalRule key, 495–496
- Internet, icons for, 295
- Internet Account Manager key, 473
- Internet Audio key, 473
- Internet Cache Files key, 483
- Internet Call feature, 468
- Internet Connection Wizard key, 473
- Internet Domains key, 473
- Internet Explorer
 - Favorites list, **294**
 - icons for, 295
 - key for, 473
 - Last Update for, 486
- Internet Explorer Administration Kit (IEAK), **487–488**
- Internet Information Server (IIS), 550
- Internet key, 354, 360
- Internet Mail and News key, 473
- Internet News Server, 204–205
- Internet Options dialog box
 - SO key for, **486–487**
 - SOIEAK key for, **487–488**
- Internet Settings key, **484–489**
- Internet Shopper key, 473
- InternetProfile value entry, 426
- Interrupt Requests (IRQs) and interrupts
 - for IDE controllers, 334
 - Plug-and-Play identifiers for, 665
 - in ResourceMap, 56, 444, 446
- Interrupt value entry, 334
- intranet zones, 489
- Intuit key, 59
- I/O channels, 444
- I/O ports, 444
- IoPageLockLimit value entry, 554
- IOProcs key, **417**
- IPConfMSP key, 489
- IPConfTSP key, 489
- IpInIp key, 473
- IPMulticastMibAgent key, 473
- IPXMibAgent key, 473

IRC, **638–639**, 638
IRQs (Interrupt Requests)
 for IDE controllers, 334
 Plug-and-Play identifiers for, 665
 in ResourceMap, 56, 444, 446
ISA bus in MultifunctionAdapter, 439–440
Isa value entry, 556–557
ISAPNP key, 570
IsPortable value entry, 534
Item Toolbars key, 361
ITStorage key, **500**

J

Japanese keyboard mapping, **559–560**
JAVA key
 in SO, 486
 in SOIEAK, 487
Java VM key, 473
JAVAPER key
 in SO, 486
 in SOIEAK, 487
Jet key, **350–351**, 473
Journal key, 361
Journal Entry key, 361
JPN key, **559–560**

K

Kbdclass key, 441
KbdIdle value entries, 558
KeepAliveTimeout value entry, 565
KeepRasConnections value entry, 516
kernel, initialization of, 10
KernelDumpOnly value entry, 530
Key Name dialog box, 142
keyboard key
 in Control Panel, **417**
 in WOW, 518
Keyboard Layout key
 in Control, **535**
 in HKEY_CURRENT_USER, **423–424**
 in HKEY_USERS, 65
Keyboard Layout value entry, **291–292**
Keyboard Layouts key, **535–536**
Keyboard Response key, **405**
keyboard value entry, 556
KeyboardClass key, 55, 441
KeyboardDelay value entry, **290–291**, 417

- KeyboardLayout value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- KeyboardName value entry, 563
- KeyboardPort key, 56
- keyboards
 - accessibility options for, **405**
 - in Control Panel, **417**
 - delay settings for, **290–291**
 - in DeviceMap, 55–56, 441
 - language support for, 65, **423–424, 535–536**
 - layout of, **291–292**
 - at logon, **314**
 - Plug-and-Play identifiers for, 666
 - in ResourceMap, 444
 - settings for, **290–291, 417**
 - in WOW, 518
- KeyboardSpeed value entry, **291, 417**
- KeyboardType Mapping key, **559–560**
- KeyExchangeAlgorithms key, **546–547**
- KeyRing key, 474
- keys, 5, 12. *See also* hives; value entries
 - accessing information about, **242–243**
 - adding, **108–109**
 - backing up, **109–110**
 - copying, **81, 110–111**
 - creating, **79, 81, 92, 92**
 - deleting, **94–95, 111–112**
 - enumerating, **243**
 - exporting, **74–76, 75**
 - importing, **75–76**
 - loading, **87–89, 112–113**
 - opening, 241–242
 - restoring, with reg restore, **114–115**
 - saving, **115–116**
 - searching
 - with reg query, **113–114**
 - with RegEdt32, **95, 95**
 - in SecurityProviders, **546–547**
 - unloading
 - with reg update, **117**
 - with RegEdt32, 89
 - with unload, **116**
 - updating, **117**
- Kmode value entry, 555
- KnownDLLs key, 554
- KOR key, **560**

Korean keyboard mapping, **560**

L

/L parameter, 83

labels

for events, 395–397

for icons, 281–282

mailing, 362

LAN connections, 581

LanAdapter value entry, 565

Language key, 538

LanguageResources key, 354

languages

code pages for, 538

keyboard layout for, 65, **423–424, 535–536**

Languages key, 474

LANManagerMIB2Agent key, 474

LanmanServer key, 474

LanmanWorkstation key, 474

LargeSystemCache value entry, 554

Last Known Good boot process, **61–62**

Last Update key, 486

LastFontSweep key, 505

LastKnownGood configuration

in boot process, 209, **211–214, 211–212**

protecting, 205

LastKnownGood value entry, 576

LastSweepTime value entry, 505

LastType1Sweep key, 512

LAUNCHING key

in SO, 487

in SOIEAK, 488

layout, keyboard, **291–292, 423–424**

Layout File key, 535

layout programs, Publisher, **357**

Layout Text key, 536

leaky memory, **273–274**

lease information for DHCP, **339**

legacy applications, 174, **503**

LEGACY_EL90X key, 572

LegacyInfOption value entry, 529

legalnoticecaption key, 491

LegalNoticeCaption value entry, 514

legalnoticetext key, 491

LegalNoticeText value entry, 514

Length object

in REG_FULL_RESOURCE_DESCRIPTOR, 608

- in REG_RESOURCE_LIST, 612–613
- in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Level object, 608
- Library entry point for performance monitoring, 272
- LicensedProcessors value entry, 552
- licenses for themes, 280
- Line Print key, 360
- Linked Table Manager, 351
- links
 - data type for, **609**
 - in OLE, **154**. *See also* OLE (Object Linking and Embedding)
- List Gallery Presets key, 362
- List Size object, 614
- List value entry, 547
- listsvc command, 42
- little endian data type, **604–605**
- load function, **112–113**
- LoadBalance value entry, 567
- loading
 - keys, **87–89, 112–113**
 - objects, delays for, 494
 - .reg files, **34–35**
- LoadTrustedDrivers value entry, 543
- Local Computer Properties dialog box, 135–136, 136
- local groups
 - in Aliases, 448, 449
 - in Builtin, **454–455, 454**
- Local Security Authority, **536**
- Local User Properties dialog box, 136
- Locale key, 538
- LocaleMapIDs value entry, 490
- locales, keyboard layout for, **423–424**
- location
 - of printers, 542
 - of registry, **18–20**
 - of users, 496
- Location value entry, 542
- Location0 key, 496
- Locations key, 496
- LODCTR program, 272
- LODCTR.INI file, 272–273
- .log extension, 20
- log files
 - in boot process, **208–210**
 - for hives, 5
- LogConf key, 336
- LogEvent value entry, 530
- logon command, 42

- LOGON KEY, 486–487
- logon scripts, hiding window for, **314**
- logon value entry, 560
- LogonPrompt value entry, 516
- logons
 - in Cache, **456–457**
 - keyboard layout at, **314**
 - policies for, 140
- LOGONSERVER value entry, 429
- long filenames, **195–199**
- LongDistanceRule key, 495–496
- LPT ports, 336
- LPT_timeout value entry, 569
- LSA key, **536**
- lurking, 637
- LV key, 354

M

- MACCP value entry, 538
- machine accounts, **453**
- Machine Check Architecture (MCA), 550
- Machine Check Exception (MCE), 550
- machine configuration. *See* HKEY_LOCAL_MACHINE hive
- Machine Debug Manager key, 474
- magazines for help, 619
- mail, **355, 468–470, 469**
- Mail Merge key, 357
- [Mail] section, 177–178
- maintenance, Registry for, 4
- MajorVersion value entry, 539
- Manual Policy mode in Update, 567
- Manufacturer value entry, 509
- map command, 42
- Map Network Drive command, **332**
- Map Network Drive dialog box, 318, 424, 425
- MAPI32.DLL file, 164
- MapID value entry, 512
- mapping
 - drive letters, 424–426, 425
 - .ini files, 234, 507
 - keyboard, **559–560**
- Master Boot Record (MBR), 9
- math co-processors, 55, **439**
- Max Battery power policy, 420
- MaxCmds value entry, 321
- MaxCollectionCount value entry, 321
- MaxConnectionTime value entry

- in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- MaxDisconnectionTime value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- MaxIdleTime value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- Maximum Address object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Maximum Channel object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 616
- Maximum Vector object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 616
- MaximumSpeed value entry, 405
- MaxIncrSize value entry, 505
- MaxInitSize value entry, 505
- MaxInstanceCount value entry, 565
- MaxSize value entry, 505
- MaxThreads value entry, 321
- MBR (Master Boot Record), 9
- MCA (Machine Check Architecture), 550
- MCA value entry, 556
- MCD key, **489**
- MCE (Machine Check Exception), 550
- MCI key, 507
- MCI Extensions key, 508
- [mci] section, 176
- [mci extensions] section, 177–178
- MCI32 key, **508**
- md command, 42
- Media Control Interface (MCI), 507
- Media Player, 346
- Media View File System DLL, 417
- MediaCategories key, 536
- MediaInterfaces key, 536
- MediaPlayer key, 474
- MediaProperties key, **536**
- MediaResources key, **537**
- MediaSets key, 537
- Members key, 448
- memory

- for command-prompt windows, **304–305**
- leaky, **273–274**
- parity in, 16
- in ResourceMap, 56, 444, 445
- testing, 9
- Memory Management key, **554–555**
- Menu flags, 166
- MenuFont value entry, 416
- MenuHeight value entry, 416
- menus
 - add-ins for, **351**
 - context
 - modifying, **382–386**, 383–385
 - removing items from, **332**
 - display delay for, **289–290**, 414
 - pop-up, **295**
- MenuShowDelay value entry, **289–290**, 414
- MenuWidth value entry, 416
- Message key, 361
- MessageFont value entry, 416
- MF_ flags, 166
- MFC (Microsoft Foundation Classes), **247–250**
- Microsoft Chat Character Editor key, 474
- Microsoft Client Service for NetWare Networks, 141
- Microsoft Comic Chat key, 474
- Microsoft Excel key, 360
- Microsoft Expedia key, 474
- Microsoft Foundation Classes (MFC), **247–250**
- Microsoft FrontPage key, 474
- Microsoft Image Composer key, 474
- Microsoft Internet Information Server key, 549
- Microsoft key in Software
 - in HKEY_CURRENT_CONFIG, 577
 - in HKEY_CURRENT_USER, **427–428**
 - in HKEY_LOCAL_MACHINE, 59, 460, **470–479**, 500. *See also* CurrentVersion key
- Microsoft Management Console (MMC), 122, **130–133**, 131–133
- Microsoft Network, 295
- Microsoft Office. *See* Office
- Microsoft Paint program, **158–161**, 159–160
- Microsoft Reference key, 474
- Microsoft Web site, 619
- MIDI (Musical Instrument Digital Interface)
 - MediaProperties key for, **536**
 - MediaResources key for, 537
 - Midimap key for, **508**
- MIDI key, 537
- Midimap key, **508**
- migrate.inf file, 39

- migration, **251**
- Migration key, 354
- MinEncryptionLevel value entry
 - in RDP-Tcp, 565
 - in Tds, 561
- Minimal Power Management power policy, 420
- Minimum Address object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Minimum Channel object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 616
- Minimum Vector object
 - in REG_RESOURCE_LIST, 613
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- MinorVersion value entry, 539
- MISC key
 - in SO, 487
 - in SOIEAK, 487
- MiscStatus flag values, 165
- “Missing entry point RunDll” message, 157
- mkdir command, 42
- MMC (Microsoft Management Console), 122, **130–133**, 131–133
- MMC key, 474
- MMCPL key, **417–418**
- MMCTlsForIE key, 474
- mmdrv.dll driver, 176
- Mobile key, 474
- modem devices, Plug-and-Play identifiers for, **676**
- modem drivers, **497**
- Modems With Bad DSR value entry, 557
- ModuleCompatibility key, **508**
- ModuleUsage key, 489
- Monitor value entry, 540
- monitoring performance. *See* PerfMon1 program; performance and performance data
- Monitors key, **541**
- more command, 42
- MOS key, 474
- mosaic prefix for URLs, 499
- most recently used (MRU) lists, 198
- MosTrace key, 474
- motherboard, Plug-and-Play identifiers for, 669
- MountedDevices key, 61, 576
- mouse
 - in DeviceMap, 441
 - doubleclick settings for, **293**, 418
 - in HKEY_CURRENT_USER, **418**
 - Intellimouse, **292–294**

- miscellaneous setting for, **293–294**
- Plug-and-Play identifiers for, 669–670
- scroll lines setting for, **292**
- speed of, 405

Mouse key, **418**

MouseKeys key, **405**

MouseName value entry, 563

MouseSpeed value entry, 293, 418

MouseThreshold1 value entry, 293, 418

MouseThreshold2 value entry, 294, 418

moving

- windows, **287–288**
- Windows directory, **321–328**

MPEGVideo value entry, 508

MRU (most recently used) lists, 198

MS Design Tools key, 474

MS-DOS Emulation key, **489**

.msc extension, 133

MSDTC key, 475

MSE key, 475

MSFTPCVC key, 550

MSFTPSVC key, 475

MsgQBadAppSleepTimeInMillisec value entry, 558

MSPAB key, 363

MSTTS key, 475

multi-processor computers, **438–439**

Multi-String editor, 52, 52

MultifunctionAdapter key, **55, 439–440, 440**

multimedia

- configurations for, 475
- drivers for, 508
- Plug-and-Play identifiers for, **675–676**

Multimedia key, 475

multiple boot configurations, **25–26**

Musical Instrument Digital Interface (MIDI)

- MediaProperties key for, **536**
- MediaResources key for, 537
- Midimap key for, **508**

Mvfs32.DLL file, 417

My Computer, icons for, 295

My Network Places, icons for, 295

MyComputer key, 482

N

Name key, 495

Name value entry

- in Clip Redirector, 558

- in Printers, 542
 - in Providers, 543
 - in Terminal Server Redirector, 558
- named pipes, character-mode, 321
- names
 - for hardware profiles, 436–437
 - for keys, 79
 - in networking, **320**, 543
 - for printers, 542
 - in telecommunications, 495
 - for value entries, 5, 80
 - for Windows directory, **321–328**
- Names key
 - in Groups, 450, 451
 - in Users, 451, 452
- NameSpace key
 - in MyComputer, 482
 - for system configuration, 355
- National Language Support (NLS)
 - in CurrentVersion, 490
 - in NLS, **537–538**
- Ncpa key, 475
- NdisWan key, 475
- Net ports, 336
- nesting organizational units, 123
- NET COMPUTER command, 453
- net share command, 68
- NET USE command, 318
- NetBIOS key, 475
- NetBT key, 475
- netcache key, 490
- NetcardDlls value entry, 576
- NetDDE key, 475
- Netlogon.dnb file, 19
- Netlogon.dns file, 19
- Netlogon share, 68
- NetMeeting tool, 468
- NetPopup value entry, 543
- NetRules key, 509
- NetSh key, 475
- NetShow key, 475
- Network applet, 316–317
- Network Control Panel, 138
- NETWORK group, 64
- network interface cards (NICs), 373, **671–674**
- Network key
 - in CurrentControlSet, 537
 - in CurrentVersion, 317, **509**

- in HKEY_CURRENT_USER, **424–426**, 425
 - in HKEY_USERS, 65
- Network Neighborhood, icons for, 295
- Network Provider key, 320
- NetworkCard key, **509–510**
- NetworkCards key, **509–510**
- networking, **316**
 - common entries for, **340–341**
 - in CurrentVersion, 317, **509–510**
 - delays in, **341**ghosted connections, **319–320**
 - in HKEY_CURRENT_USER, **424–426**, 425
 - names in, **320**
 - performance of, **320–321**
 - persistent connections in, **318–319**
- NetworkPath value entry, 567
- NetworkProvider key, **537**
- New dialog box, 235
- New key, 483
- new users, initial settings for, **431**
- newsgroups, 204–205, 468, **637**
- NewShortcutHandlers key, 482
- nibbles, 403
- NICs (network interface cards), 373, **671–674**
- 9.0 key, 348, 358
- NL\$ value entries, 457
- NLS key in Control, **537–538**
- Nls key in CurrentVersion, 490
- <No Name> value entry, 480
- No Override attribute, 123
- NoDisplayClass value entry, 529
- NoInstallClass value entry, 529
- NoInteractiveServices value entry, 568
- Non-Driver Signing key, 475
- .None key, 399
- NonEnum key, 490
- NoNetConnectDisconnect value entry, 332
- NonGeneralMIDIList key, 537
- NonPagedPoolQuota value entry, 555
- NonPagedPoolSize value entry, 555
- NonWindowsApp key, 518
- NoRemotePrinterDrivers value entry, 539
- normal mode, booting in, **207–214**, 207–208, 211–212
- Note key, 361
- Notepad for .reg files, 35
- Notify access setting, 98
- Notify events, auditing, 100
- NT File System. *See* NTFS (NT File System)
- NTBackup program, 23

NTBootdd.sys file, 227
 ntbtdlog.txt file, **208–210**
 NTConfig.pol file, 122
 NTDebuggers key, 475
 NTDetect.com file on boot floppies, 10, 227
 ntdll.dll file, 39
 NTDS.Evt file, 19
 NTDS key, 475
 NtFrs.Evt file, 19
 NTFS (NT File System), 23

- backing up, **25–26**
- boot floppies for, **227**
- moving Windows directory in, **325–328**
- restoring, **31–32**

 NtfsDisable8dot3NameCreation value entry, **532**
 NtfsEncryptionService value entry, 532
 NthCountMsgQPeeksSleepBadApp value entry, 558
 NTLDR file, 9–10, 227
 NTMS key, 538
 NTUSER.DAT file, 34, **67–69**, **87–89**, 392–393
 NTVersionOfLastBackup key, 510
 null characters, 609
 Num Lock settings, **290**, 417
 NumberOfHistoryBuffers value entry, **312**, 401
 NwlnkIpx key, 475
 NwlnkNb key, 475
 NwlnkSpx key, 476
 NWLogonServer value entry

- in DefaultUserConfiguration, 558
- in RDP-Tcp, 565

O

OADIST.EXE file, 183
 Object dialog box, 158, *158*
 Object Linking and Embedding. *See* OLE (Object Linking and Embedding)
 object-oriented programming (OOP), 247
 ObjectDirectories value entry, 552
 OC Manager key, 493
 ODBC key, 59, **518**
 ODBCINST.INI file, 518
 OEMCP value entry, 538
 OEMHAL value entry, 538
 OEMLocale key, 538
 Office, **344–345**

- as client-server application, **157–161**, *158–160*
- with FindFast, **368**
- .reg files for, **347–348**

- registry manipulation for, **363–368**, 364
- Setup program for, **345–347**
- shared components in, **345–346**
- system configuration information for, **348–350**
 - Access, **350–353**
 - Common, **353–354**
 - Excel, 354
 - Outlook, **355**
 - PowerPoint, **355–356**
 - Publisher, **357**
 - Registration, 358
 - Word, 358
- user configuration information for, **358**
 - Access, 359
 - Binder, 359
 - Common, **359–360**
 - Draw, 360
 - Excel, **360**
 - Graph, 360
 - Outlook, **360–361**
 - PowerPoint, 361
 - Publisher, **362**
 - Query, 362
 - Word, **362–363**
- Office key, 348, 476
- Office Explorer key, 361
- Office Finder key, 361
- OLE (Object Linking and Embedding), **148–149**, **152–153**
 - ActiveX controls, **154–156**
 - client-server applications, **157–161**, 158–160
 - configuration information for, 476
 - disasters in, **171–172**
 - embedding in, **154**, **170–171**
 - GUIDs in, 372
 - HKEY_CLASSES_ROOT for, **386**, 387
 - linking in, **154**
 - operation of, **168–170**
 - and registry, **161–168**, 162
 - service and driver failures in, 172
- OLE automation, 153
- Ole key, 476
- OLE Play Options key, 356
- Ole32.DLL file, 164, 172
- OLEAUT32.DLL driver, 183–184
- OLEVERBATTRIB_ flags, 167
- OMI key, 355
- OMI Account Manager key, 361
- OOP (object-oriented programming), 247

- Open DataBase Connectivity (ODBC), 59, **518**
- Open entry point for performance monitoring, 272
- open key, 380
- Open Find key, 354, 360
- Open With dialog box, **388–390**, 388
- Open With option, 295
- OpenGL 3D Text screen saver, 300
- OpenGL mini-client driver, **489**
- OpenGLDrivers key, 510
- opening
 - keys, 241–242
 - registries, **85–86**
- operating systems for stabilizing systems, **205–206**
- Operations key, 355
- Option object
 - in REG_RESOURCE_LIST, 611
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- Optional value entry, 555
- OptionalComponents key, 493
- OptionalNames value entry, **320**
- Options dialog box
 - in Explorer, 186
 - in Word, 126, 127
- Options key
 - in Access, 351
 - in Excel, 354
 - in Outlook, 361
 - in Word, 362
- Options menu
 - Auto Refresh command, 102
 - Confirm on Delete command, 103
 - Font command, 102
 - Read-Only command, 103
 - Save on exit command, 104
- Options tab, 305, 305
- Order key
 - in NetworkProvider, 537
 - in ServiceProvider, **548–549**
- Order value entry, 543
- Organization Chart program, 346
- organizational unit group policy objects, 123
- origin for wallpaper, **284–285**
- OS/2 Subsystem for NT key, 476
- Os2 value entry, 555
- OsLoaderPath value entry, 576
- OSPFMibAgent key, 476
- OutBufCount value entry
 - in RDP-Tcp, 565

- in Tds, 561
 - in WinStations, 563
- OutBufDelay value entry
 - in RDP-Tcp, 565
 - in Tds, 562
 - in WinStations, 563
- OutBufLength value entry
 - in RDP-Tcp, 565
 - in Tds, 562
 - in WinStations, 563
- Outlook
 - for e-mail, 469
 - icons for, 295
 - OLE with, 153
 - system configuration for, **355**
 - user configuration information for, **360–361**
- Outlook Express key, 476
- Outlook key in Office, **355**
- OUTLOOK key in Word, 363
- overwrite mode in command-prompt window, **312–313**
- Overwrite value entry, 530
- overwritten applications, **221–222**
- Owner tab, 101, *102*
- ownership in security, **101**, *102*

P

- Page Size key, 357
- PagedPoolQuota value entry, 555
- PagedPoolSize value entry, 555
- PagingFiles value entry, 555
- Paint program, **158–161**, *159–160*
- PaintDesktopVersion value entry, 414
- panning settings, 580
- parallel devices, Plug-and-Play identifiers for, 666
- Parallel key, 337
- parallel ports
 - in DeviceMap, 55
 - in ResourceMap, 444
 - in Services, **337**
- PARALLEL PORTS key, 55
- parameters for OLE Controls, 158
- Parameters key
 - for free disk space, 331
 - for networking, 320–321, 339–340
- Parameters value entry, 542
- parity
 - memory, 16

- for serial ports, 336
- Parity value entry, 562
- Partition menu, Configuration Save command, 575
- Password value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 565
 - in WinStations, 563
- PasswordExpiryWarning value entry, 338–339, 516
- passwords
 - for Administrators, **461–462**
 - expiration messages for, **338–339**, 516
 - policies for, 138, 141
 - for Recovery Console, 38–39
- patches for compatibility, 503
- Path value entry
 - in App Paths, 480
 - in InstallRoot, 350
- PathName value entry, 501
- paths
 - for applications, 480
 - for cache directories, 485
 - disks in, 331
 - policies for, 141
 - for Windows NT installation, 501
- Paths key, 485
- Pattern value entry, **285**, 414
- Patterns key, **419**
- PCFlat value entry, 556
- PCI bus, 440
- PCI_HAL key, 570
- PCI key, 570
- PCI value entry, 556
- PCIIDE key, 570
- PCMCIA controller, Plug-and-Play identifiers for, 669
- PCMCIA value entry, 556
- PdClass value entry
 - in RDP-Tcp, 565
 - in Tds, 562
 - in WinStations, 563
- PDCs (Primary Domain Controllers)
 - in Cache, 456
 - in Users, 451
- PdDll value entry
 - in RDP-Tcp, 565
 - in WinStations, 563
- PdDLL value entry in Tds, 562
- PdFlag value entry
 - in RDP-Tcp, 566

- in Tds, 562
 - in WinStations, 563
- PdName value entry
 - in RDP-Tcp, 566
 - in Tds, 562
 - in WinStations, 563
- Pegasus key, 476
- percent signs (%) for environment variables, 605
- PERF_COUNTER_DEFINITION structure, **269–271**
- perfc009.dat file, 642
- perfh009.dat file, 642
- Perflib key, **510**
- PerfMon1 program, **254–255**
 - counters in, **256–259**, 258, 260
 - listing of, **260–271**
 - output from, **255–256**
- performance and performance data, 510
 - adding to registry, **271–273**
 - counters for, **642–662**, 642
 - HEY_PERFORMANCE_DATA for, **66**
 - leaky memory, **273–274**
 - network, **320–321**, **339–340**
 - read access to, **274**
 - Registry for, 4
 - in stabilizing system, 219
- Performance Monitor, **510**
- performance objects, 257
- peripheral buses, Plug-and-Play identifiers for, 668
- permissions
 - for Security, 456
 - setting, 57, 69, 90, **96–99**, 96–98
- Permissions For dialog box, 96–99, 96
- Permissions tab, 97–99, 97
- PerSessionTempDir value entry, 558
- persistent connections, **318–319**
- persistent shares, 178
- Personal key, 362
- Physical Memory value entry, 444
- Pietrek, Matt, 194–195
- PIF (Program Interface File) manager, 380
- pipes
 - character-mode, 321
 - secure, **544–545**
- Plug and Play (PnP) devices
 - HARDWARE hive for, 55
 - in MultifunctionAdapter, 440
 - with serial ports, 336
- Plug-and-Play identifiers

- for modem devices, **676**
- for multimedia devices, **675–676**
- for NIC devices, **671–674**
- for SCSI and CD-ROM devices, **675**
- for system devices, **664–670**
- PMON program, 273–274
- Pnp key, 539
- PnPManager key, 57
- pointer ports, 444
- pointer value entry
 - in Run, 492
 - in Setup, 556
- PointerClass key, 56
- PointerPort key, 56
- .pol extension, 135
- policies, **122–123**
 - Active Directory Users and Computers tool for, **134–135**
 - Administrative Templates for, **125–129**, 127
 - editors for, **142–143**
 - MMC for, **130–133**, 131–133
 - RASC, **490–491**
 - security, **457–459**
 - Software Settings for, **124**
 - System Policy Editor for, **129–130**, **135–141**, 136
 - for Windows NT 4, **135–141**, 136, 142
 - Windows Settings for, **124**
- Policies key
 - in CurrentVersion, **490–491**
 - in Software, 519
- Policy key, **457–459**
- PollingPeriod value entry, 336
- pop-up menus, **295**
- pop-up windows, color for, **303**, 401
- PopupColors value entry, **303**, 401
- Port object, 608
- Port value entry
 - in change, 560
 - in Printers, 542
 - in SerialKeys, 406
- portable computers, 437–438
- Portable/Laptop power policy, 420
- PortName value entry, 336
- PortNumber value entry
 - in RDP-Tcp, 566
 - in Tds, 562
- ports
 - configuring, 510
 - in DeviceMap, 55, 441

- parallel, **337**
 - for printers, 542
 - in ResourceMap, 56, 444
 - serial, **335–337**, 405–406
- Ports key, **335–337**, 510
- [Ports] section, 178
- PortThreadPriority value entry, 539
- position
 - of Comic Chat control, 162
 - of windows, **310–311**
- Posix value entry, 555
- power management
 - in ApmActive, **526**
 - in ApmLegalHal, **527**
 - options for, **419–420**
- Power On Self Tests (POSTs), 9
- Power Options applet, **419–420**
- POWER USERS group, 63
- PowerCfg key, **419–420**
- PowerdownAfterShutdown value entry, 514
- PowerOffActive value entry, 414
- PowerOffTimeout value entry, 414
- PowerPoint key, **355–356**
- PowerPoint program, **355–356**, 361
- PowerPolicies key, 420
- PPCentral key, 356
- Preferences key, 362
- prefixes for URLs, **498–499**
- Prefixes value entry, 498
- Preload key, 423
- Presentation power policy, 420
- presentation programs, PowerPoint, **355–356**, 361
- Primary Domain Controllers (PDCs)
 - in Cache, 456
 - in Users, 451
- primary IDE controllers, 333
- PrimaryBusiness key, 362
- Print dialog box, 78, 78
- Print key, **539**
 - in CurrentVersion, 511
 - Environments key, **540**
 - Forms key, **540–541**
 - Monitors key, **541**
 - Printers key, **542–543**
 - Providers key, **543**
- print key in shell, 380
- PRINT OPERATORS group, 63
- Print Processor value entry, 542

- Print Processors key, 540
- Printable key, 165
- Printer Driver value entry, 542
- [PrinterPorts] section, 178
- printers
 - drivers for, 542
 - parallel ports for, **337**
 - policies for, 138–139, 141
- Printers applet, 541
- Printers component, special folder for, 298
- Printers key
 - in HKEY_CURRENT_USER, **426**
 - in HKEY_USERS, 65
 - in NameSpace, 482
 - in Print, **542–543**
- Printing key
 - in Outlook, 361
 - in Publisher, 357
- printing registry
 - in RegEdit, **78, 78**
 - in RegEdt32, **91–92**
- Priority value entry, 542
- PriorityCacheFormats key, 494
- PriorityClass value entry, 539
- PriorityControl key, **544**
- PrivateProfileString function, 366–368
- PrivateProperties key, 536
- Privilgs value entry, 459
- process value entry, 560
- ProcessorControl value entry, 552
- processors
 - in CentralProcessor, 54–55, **438–439**
 - LicensedProcessors value entry for, 552
- ProductID value entry, 501
- ProductName value entry, 502, 509
- ProductOptions key, **544**
- ProductType value entry, 544
- ProductVersion value entry, 558
- ProfileDlgTimeOut value entry, 516
- ProfileGuid key, 511
- ProfileImagePath value entry, 511
- ProfileList key, **511**
- profiles
 - hardware, **436–438, 437**
 - HKEY_USERS for, **62–66**
 - NTUSER.DAT for, **67–69**
 - policies for, 140–141
 - in ProfileList, **511**

- in Shell Folders, 482
- in Win.ini, 178
- ProgId view in RegMaid program, **191–192**, 191
- Program Groups key, 59, **519**
- Program Interface File (PIF) manager, 380
- Program Manager
 - Unicode Program Group for, **429**
 - Win.ini settings for, 179
- Program name key, 340
- programming, **230–231**
 - for accessing registry, **235–246**, 247
 - API for, **231–234**
 - MFC for, **247–250**
 - and migration, **251**
- Projects tab, 235
- Properties dialog box
 - for command-prompt window, 302, 309, 309
 - for cursors, 305, 305
 - for Desktop, 280
 - for fonts, 306, 306
 - for history buffer, 311
- PropertyProviders value entry, 534
- ProPrint key, 357
- Protected Storage System Provider key, 476
- ProtectionMode value entry, 552
- Protocol.ini file, 174
- protocol key, 167
- ProtocolDefaults value entry, 488
- Protocols key, **547**
- ProviderName value entry, 425
- ProviderOrder value entry, 537, 549
- Providers key
 - in Print, **543**
 - in Telephony, 496
- ProviderType value entry, 425
- ProxyEnable value entry, 574, 578
- PubBackground key, 357
- PubClipart key, 357
- public-key algorithms, **546–547**
- Publisher
 - system configuration for, **357**
 - user configuration information for, **362**
- Publisher key, **357**, **363–368**, 364
- Publisher Options dialog box, 364

Q

- Qualcomm key, 59

- query function, **113–114**
- query key, 560
- Query program, 362
- Query Value access setting, 97
- Query Value events, auditing, 100
- querying the interface technique, 169
- QuickEdit mode, **313**
- QuickEdit value entry, **313**, 401
- quotes (“”) for filenames, 196

R

- /R parameter, 83
- Ranges value entry, 488
- RAS (Remote Access Service)
 - answer setting for, **583–584**
 - configuration for, 476
 - connectoids for, 426
 - in upgrading, 330
- RAS key, 476
- RasAuto key, 476
- RASForce value entry, 517
- RasMan key, 476
- raster fonts, 307
- Ratings key, 490
- .Raw value entry, 446
- rd command, 42
- RDisk utility, 22
- RDP-Tcp key, **564–566**
- rdpdd key, 560
- rdpwd key, **561–562**
- Read Control access setting, 98
- Read Control events, 101
- read errors, **223–224**
- read-only RegEdt32 option, **103**
- real-time clock, Plug-and-Play identifiers for, 668
- rebooting
 - for changes, 35, 43, 316, 583
 - in restoring, 31
- Recent File List key
 - deleting items in, 198
 - in Publisher, 357
 - for user configuration, 360
- Recent Files key, 354
- Reconnect at Logon option, 318, 424–425
- Recovery Console, **36**
 - commands and options for, **40–43**
 - components of, **37–39**
 - on installation CD-ROM, **40**

- installing, **36–37**, 37
- using, **39**
- Recreational Software Advisory Council (RSAC) ratings, **490–491**
- Recycle Bin
 - icons for, **297–298**, 297
 - special folder for, 298
- ReferenceTitles key, 476
- reformatting after read errors, 223
- refresh intervals for domain lists, **338–339**
- refresh options for RegEdt32, **102–103**
- refresh rate, video, 580
- reg add function, **108–109**
- reg backup function, **109–110**
- REG_BINARY data type, 4, **50**, **602**
- REG_COLOR_RGB data type, **602–603**
- reg copy function, **110–111**
- REG.DAT key, 60
- reg delete function, **111–112**
- REG_DWORD data type, **52**, **603**
- REG_DWORD_BIG_ENDIAN data type, **603**
- REG_DWORD_LITTLE_ENDIAN data type, **604–605**
- Reg.exe program, **107–108**
 - add function, **108–109**
 - backup function, **109–110**
 - copy function, **110–111**
 - delete function, **111–112**
 - load function, **112–113**
 - query function, **113–114**
 - restore function, **114–115**
 - save function, **115–116**
 - unload function, **116**
 - update function, **117**
- REG_EXPAND_SZ data type, **51**, **605–606**
- REG_FILE_NAME data type, **606**
- REG_FILE_TIME data type, **606**
- .reg files, **34–35**, 344, **347–348**
- REG_FULL_RESOURCE_DESCRIPTOR data type, **52**, **606–608**
- REG_LINK data type, **609**
- reg load function, **112–113**
- REG_MULTI_SZ data key, 7, **52**
- REG_MULTI_SZ value entry, **609**
- REG_NONE data type, 53, **609–610**
- REG program, 43
- reg query function, **113–114**
- REG_QWORD data type, 8, **610**
- REG_QWORD_LITTLE_ENDIAN data type, **610**
- REG_QWORD_LITTLE_INDIAN data type, 8

- REG_RESOURCE_LIST data type, 446, **610–613**
- REG_RESOURCE_REQUIREMENTS_LIST data type, **613–616**
- reg restore function, **114–115**
- reg save function, **115–116**
- REG_SZ data type, **51, 616**
- REG_UNKNOWN data type, 8, 53, **616**
- reg unload function, **116**
- reg update function, **117**
- Reg1.cpp program, **235–240, 247**
- RegClean program, **183–184, 199**
 - running, **184–185, 184–185**
 - undoing, **185–186, 186, 199**
- RegCloseKey function, 231
- RegConnectRegistry function, 231
- RegCreateKey function, 231
- RegCreateKeyEx function, 231
- RegDeleteKey function, 231
- RegDeleteValue function, 231
- RegDone value entry, 502
- RegEdit editor, **72–74, 74**
 - for backups, **28–29**
 - from command line, **82–83**
 - copying keys with, **81**
 - creating keys with, **79**
 - creating value entries with, **79–80, 81**
 - exporting hives and keys in, **74–76, 75**
 - importing hives and keys in, **75–76**
 - for NTUSER.DAT, **68–69**
 - printing registry with, **78, 78**
 - vs. RegEdt32, 118
 - with remote registries, **77, 77, 84–85**
 - searching with, **81–82, 82**
 - tips for, **83–84**
- RegEdt32 editor, **85**
 - creating keys in, **92, 92**
 - creating value entries in, **93–95, 93**
 - delete option for, **103–104**
 - deleting keys in, **94–95**
 - fonts for, **102, 103**
 - loading keys in, **87–89**
 - opening and closing registries with, **85–86**
 - printing registry in, **91–92**
 - read-only option for, **103**
 - refresh options for, **102–103**
 - vs. RegEdit, 118
 - with remote registries, **86, 86**
 - restoring hives in, **91**
 - restricting access to, **118–119**

- save option for, 104
- saving hives in, **89–90**, 90
- searching keys in, **95**, 95
- security in, **96–101**, 96–100, 102
- tips for, **104**
- RegEnumKey function, 231, 243
- RegEnumKeyEx function, 232
- RegEnumValue function, 232, 244
- RegFlushKey function, 232
- RegGetKeySecurity function, 232
- Regional Options applet, 423
- Regional Settings Properties dialog box, 536
- RegisteredOrganization value entry, 502
- RegisteredOwner value entry, 502
- RegisteredProcessors value entry, 552
- Registration key, 358
- Registry Key Permissions dialog box, 456
- Registry menu, 74
 - Close command, 85
 - Connect Network Registry command, 73
 - Export Registry File command, 29, 69, 75, 224
 - Import Registry File command, 35, 75
 - Load Hive command, 68, 88
 - Open Local command, 85–86
 - Print command, 78
 - Printer Setup command, 91
 - Restore command, 91
 - Save Key command, 90
 - Save Subtree As command, 89–90, 90, 104
 - Select Computer command, 86
 - Unload Hive command, 69
- Registry.pol files, **128–129**
- RegistrySizeLimit value entry
 - in Control, 526
 - in CurrentControlSet, 579
- RegLoadKey function, 232
- RegMaid program, **187–189**
 - CLSID view in, **189–191**, 190
 - Interface view in, **193**
 - ProgId view in, **191–192**, 191
 - recommendations for, **193–194**
 - TypeLib view in, **192**, 193
- RegNotifyChangeKeyValue function, 232
- RegOpenKey function, 232
- RegOpenKeyEx function, 232, 242
- RegOpenUserClassesRoots function, 233
- RegOverridePredefKey function, 233
- RegQueryInfoKey function, 232, **242–243**

- RegQueryMultipleValues function, 232
- RegQueryValue function, 232
- RegQueryValueEx function, 232
- RegReplaceKey function, 233
- RegRestoreKey function, 233
- RegSaveKey function, 233
- REGSERV service, 84
- RegSetKeySecurity function, 233
- RegSetValue function, 233
- RegSetValueEx function, 233
- RegUnloadKey function, 233
- related.desc key, 511
- remote access
 - defaults for, **582–583**
 - policies for, 140
- Remote Access Service (RAS)
 - answer setting for, **583–584**
 - configuration for, 476
 - connectoids for, 426
 - in upgrading, 330
- remote registries
 - RegEdit with, **77, 77, 84–85**
 - RegEdt32 with, **86, 86**
 - restricting access to, **118–119**
- RemoteAccess key
 - in HKEY_CURRENT_USER, **426**
 - in HKEY_USERS, 65
 - in Microsoft, 476
- RemoteAdmin value entry, 563
- RemoteComputer key, 482
- RemotePath value entry, 425
- removing. *See* deleting
- rename command, 42
- RenameFiles key, **491**
- renaming
 - directories, 214
 - files, **491**
 - Windows directory, **321–328**
- repair directory, 19, 21–22, **106, 106**
- repair disks vs. ERDs, 44
- repeat settings, keyboard, **291, 405**
- Replace Permission on Existing Subkeys option, 431
- REPLICATOR group, 63
- Report Formats key, 351
- Report key, 361
- ReportBootOk value entry, 514, 527
- Required value entry, 555
- Reserved 1 object, 608

- Reserved 2 object, 608
- reserved memory in ResourceMap, 56
- Reserved Memory value entry, 444, 445
- Reserved object, 614
- ReservedResources key, **557**
- reset key in Utilities, 560
- ResKit key, 476
- resolution, video, 580
- Resolution and Registration (RNR) Service APIs, **548–550**
- Resource Lists dialog box, 446, 446
- ResourceKit key, 476
- ResourceMap key, 49, **56, 444–446, 446, 590–591**
- resources
 - data types for, **606–608, 610–613**
 - ID numbers for, 379
- Resources dialog box, 52, 53, 446, 446
- ResourceTimeoutControl value entry, 553
- restore function, **114–115**
- RestoreConnection value entry, **319–320**
- restoring, **30**
 - from copies, **31–32**
 - emergency repair diskettes for, **32–34**
 - after read errors, 223
 - .reg files, **34–35**
 - with RegEdt32, **91**
 - from tape, **30**
 - Windows NT, **20–21**
- restrictions, policies for, 138–139
- Revision object
 - in REG_RESOURCE_LIST, 611
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- RFC1156Agent key, 476
- RGB colors, 402, **602–603**
- RIPMibAgent key, 476
- rmdir command, 42
- RNR (Resolution and Registration) Service APIs, **548–550**
- Root key in Enum, **570–572, 579–580**
- ROOT key in HTREE, 569
- Router key, 476
- RouterCacheSize value entry, 539
- rows in CoolSwitch dialog box, **288**
- Rpc key, 476
- RPCLOCATOR key, 477
- RSAC (Recreational Software Advisory Council) ratings, **490–491**
- Run key, **491–492**
- Run settings, policies for, 139
- RunDll32 file, 158–159, **387–388**
- RunLogonScriptSync value entry, 517

RunOnce key, **492**
 RunOnceEx key, 493
 Russinovich, Mark, 618
 RXACT key
 in SAM, 447, **456**
 in Security, 460
 RxFIFO value entry, 336

S

Safe Mode, booting in, **210–211**
 Safe Mode option, 208
 Safe Mode with Command Prompt option, 208, 211
 Safe Mode with Networking option, 208, 210–211
 SAFETY key
 in SO, 486
 in SOIEAK, 487
 SAM files, 18–19, **57**
 backing up, 29
 repairing, 34
 SAM key, **447**
 Domains key, **447–456**
 in HKEY_LOCAL_MACHINE, 49, **57, 591–593**
 RXACT key, **456**
 in Security, 460
 SAM sub-key, 447
 SameAreaRule key, 495–496
 Sane filenames option, 198
 .sav extension, 20
 save function, **115–116**
 SaveSettings value entry, 340
 Savill, John, hints by
 auditing, **430**
 CD-ROM Changer cycling, **582**
 change determination, 69
 OLE, 172
 registry cleanup, 199
 remote access, **582–583**
 restricting RegEdt32 editor access, **118**
 restricting remote registry access, **118–119**
 Web site for, 618
 saving
 hives, **89–90, 90**
 RegEdt32 options for, 104
 repair directory, **106, 106**
 share information, **334–335**
 Schedule+ key, 477
 ScheduleThreadPriority value entry, 539

- scheduling ERDs, **44**
- SchedulingAgent key, **477**
- SchedulingInformation key, **355**
- schemes
 - for display, **410**
 - for sounds, **398–399**
- Schemes key
 - in Appearance, **410**
 - for sounds, **399**
- Schulman, Jerold, hints by
 - Administrator passwords, **461–462**
 - auto-disconnect, **581**
 - boot floppies, **227**
 - DHCP lease information, **339**
 - Ethernet network performance, **339–340**
 - FindFast, **368**
 - initial settings, **431**
 - install path, **520**
 - keyboard layout, **314**
 - leaky memory, **273–274**
 - logon script window, **314**
 - Open With, **389–390**
 - performance access, **274**
 - policy editors, **142–143**
 - RAS answer setting, **583–584**
 - registry repair, **226**
 - screen saver, **430–431**
 - System.ini hive location, **179**
 - Unattended Setup, **179–180**
 - uninstalling, **521**
 - Web site for, **618**
- screen buffers, size of, **304–305**
- Screen Saver.3DflyingObj key, **420**
- Screen Saver.3Dpipes key, **420**
- Screen Saver.Bezier key, **420**
- Screen Saver.Marquee key, **421**
- Screen Saver.Mystify key, **421**
- Screen Saver.Stars key, **421**
- Screen Saver tab, **300**
- screen savers, **300, 420–421**
 - activating, **430–431**
 - in System.ini, **176**
- ScreenBufferSize value entry, **304–305, 401**
- ScreenColors value entry, **304, 401**
- ScreenLines value entry, **518**
- ScreenSaveActive value entry, **414**
- ScreenSaverIsSecure value entry, **414**
- ScreenSaveTimeout value entry, **414, 431**

- SCRIPT key
 - in SO, 487
 - in SOIEAK, 488
- SCRIPTING key
 - in SO, 487
 - in SOIEAK, 488
- Scripting key in Outlook, 361
- SCRIPTJAVA key
 - in SO, 487
 - in SOIEAK, 488
- SCRIPTSAFE key
 - in SO, 486
 - in SOIEAK, 487
- SCRNSAVE.EXE value entry, 414, 431
- scroll lines setting, **292**
- Scroll Lock settings, 290, 417
- ScrollHeight value entry, 416
- ScrollWidth value entry, 416
- ScrpDbg key, 477
- SCSI devices
 - Plug-and-Play identifiers for, **675**
 - in ResourceMap, 444
- Scsi key in DeviceMap, 56, **441–443**
- SCSI key in Enum, 572
- ScsiAdapter key, **55**
- searching
 - for applications, **224–225**
 - for keys and value entries
 - with reg query, **113–114**
 - with RegEdit, **81–82**, 82
 - with RegEdt32, **95**, 95
 - for startup jobs, **520**
- SearchTypes key, 355
- SecDC.inf file, 19
- SecDesc value entry
 - in Accounts, 459
 - in Domain, 459
- SeCEdit key, 511
- SecEvent.Evt file, 19
- secondary IDE controllers, 333
- SecondaryBusiness key, 362
- SecondLevelDataCache value entry, 555
- “Seconds until highlighted choice will be started automatically” message, 208
- Secrets key, 459
- SecSetup.inf file, 19
- sections, registry. *See* hives
- Secure key, 59, 477, 519
- secure protocols, **547**

- Secure Sockets Layer (SSL), 547
- SecurePipeServers key, **544–545**
- security, **22**
 - auditing for, **99–101**, 99
 - for backups, 24
 - ownership for, **101**, 102
 - permissions for, **96–99**, 96–98
 - in Recovery Console, 38
 - in RegEdt32, **96–101**, 96–100, 102
- Security Access Manager. *See* SAM files; SAM key
- Security files, 18
 - backing up, 29
 - repairing, 34
- Security Identifiers (SIDs), 6–7
 - in Account, 449–452
 - in Desktop, 280
 - in HKEY_USERS, 62–64
 - in Policy, 459
- Security key, 49, **58**, **456**, 593
 - Cache key, **456–457**
 - in Outlook, 361
 - Policy key, **457–459**
 - RXACT key, 460
 - SAM key, 460
- Security menu
 - Audit Permissions, 462
 - Auditing command, 430
 - Permissions command, 57, 69, 90, **96–99**, 96–98
- Security Settings dialog box
 - SO key for, **486–487**
 - SOIEAK key for, **487–488**
- Security tab
 - SO key for, **486–487**
 - SOIEAK key for, **487–488**
- Security value entry, 542
- SecurityProviders key, **545–547**
 - CertificationAuthorities key, **545–546**
 - Ciphers key, **546**
 - Hashes key, 546
 - KeyExchangeAlgorithms key, **546–547**
 - Protocols key, **547**
- Select Computer dialog box, 86, 86
- Select key, 61, **576**, 594
- Select Users, Computer, or Group dialog box, 99, 100
- SendAlert value entry, 530
- Separator File value entry, 542
- Sequencer value entry
 - in MCI, 507

- in MCI32, 508
- serial devices, Plug-and-Play identifiers for, 667
- serial ports, **335–337**, 405–406, 441
- SERIALCOMM key, 56
- SerialKeys key, **405–406**
- Server Manager key, 340
- SERVER OPERATORS group, 63
- service failures in OLE, 172
- Service Packs
 - in restoring, **20–21**, **203–204**
 - in upgrading, 329–330
- ServiceGroupOrder key, **547–548**
- ServiceName value entry
 - in CurrentVersion, 509
 - in Tds, 562
- ServiceProvider key, **548–550**
- Services key, **574**, **580**
- ServiceTypes key, **549–550**
- Session Manager key, **550–553**
 - AppCompatibility key, 553
 - AppPatches key, 553
 - CheckBadApps key, 553
 - CheckBadApps400 key, 553
 - DOS Devices key, **553–554**
 - Environment key, 554
 - Executive key, 554
 - FileRenameOperations key, 554
 - KnownDLLs key, 554
 - Memory Management key, **554–555**
 - SubSystems key, **555**
- session value entry
 - in query, 560
 - in reset, 560
- SessionManager key, **556**
- set command, 42
- Set Value access setting, 97
- Set Value events, 100
- SetRegistryKey function, 248, 250
- Settings key, 350, 426
- Setup key, 330
 - in Control, **556**
 - in CurrentVersion, **493**
 - in Outlook, 355, 361
 - in SYSTEM, 61, **576–577**, 594
- Setup.log file
 - in config, 19
 - on ERDs, 17, 28
 - for NTFS partitions, 327–328

- Setup program
 - for Office, **345–347**
 - for restoring, 33–34
- SetupPrograms key, 518
- setupreg.hiv file, 39
- SetupType value entry, 576
- Shadow value entry
 - in DefaultUserConfiguration, 558
 - in RDP-Tcp, 566
 - in WinStations, 563
- Share Disposition object
 - in REG_FULL_RESOURCE_DESCRIPTOR, 607
 - in REG_RESOURCE_LIST, 612
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- share information, saving, **334–335**
- Share Name value entry, 542
- Shared Parameters key, 509
- Shared Tools key, 477
- Shared Tools Location key, 477
- SharedDlls key, **493**
- Shares key, 335
- sharing data
 - Clipboard for, **149–151**
 - DDE for, **151–152**
 - OLE for. *See* OLE (Object Linking and Embedding)
 - policies for, 139, 141
- shell, policies for, 137–139
- shell key, 379–380
- Shell Extensions key, **494**
- Shell Folders key, 482
- Shell Icon BPP value entry, 416
- Shell value entry, 515
- shell32.dll file, 378, 380–381
- ShellExecuteHooks key, 482
- ShellIconSize value entry, 416
- ShellScrap key, 494
- ShellServiceObjectDelayLoad key, 494
- Shortcut Bar key, 349
- shortcuts as links, 154
- shotgun repairs, 202
- Show Hidden Files and Folders option, 490
- ShowLinkDetectEnabled value entry, 517
- ShowLinkTimeOut value entry, 517
- ShowSounds key, 406
- ShowSuperHidden flag, 490
- ShutdownTime value entry, 568
- shutdownwithoutlogon key, 491
- ShutdownWithoutLogon value entry, 515

- <SID> key in HKEY_USERS, 597
- Sid value entry
 - in Accounts, 459
 - in Domain, 459
 - in ProfileList, 511
- SIDs (Security Identifiers), 6–7
 - in Account, 449–452
 - in Desktop, 280
 - in HKEY_USERS, 62–64
 - in Policy, 459
- site group policy objects, 123
- size
 - of Comic Chat control, 162
 - of cursor, 305–306, 305
 - of fonts, **308**, 341
 - of history buffer, **311**
 - of objects, 4
 - of registry, 4
 - of screen buffers, 304–305
 - of windows, **310**
- size value entry in WOW, 569
- SizeReqBuf value entry, 341
- Slot Number object, 614
- SMAAddOns key, 509
- SmallIcons key, 482
- SmCaptionFont value entry, 416
- SmCaptionHeight value entry, 416
- smss.exe file, 39
- SnapMouseButtons value entry, 294
- SnapToDefaultButton value entry, 294, 418
- SNMP, policies for, 139, 141
- SNMP_EVENTS key, 477
- SNMPMIB key, 477
- SO key, **486–487**
- SOF\$\$\$\$.DEL file, 19
- SOFTDIST key
 - in SO, 487
 - in SOIEAK, 488
- software, 19, 34
 - damaged, problems from, 203
 - diagnostic, **206**
 - installing and deleting, problems from, 203
- Software key
 - in HKEY_CURRENT_CONFIG, 66, **577–578**
 - in HKEY_CURRENT_USER, **427–428**
 - in HKEY_LOCAL_MACHINE, 49, **58–60**, 460, 460, **466–467**
 - Classes key, **468**

- Clients key, **468–470**, 469
 - Microsoft key, 59, 460, **470–479**. *See also* CurrentVersion key
 - ODBC key, **518**
 - Policies key, 519
 - Program Groups key, **519**
 - searching in, 224
 - Secure key, 519
 - summary of contents, **593**
 - Voice key, 519
 - Windows 3.1 Migration Status key, **519**
 - in HKEY_USERS, 65
- Software.sav file, 19
- Software Settings for policies, **124**
- SoftwareType value entry, 502
- SOIEAK key, **487–488**
- SortOrder value entry, 341
- Sound key
 - in HKEY_CURRENT_USER, 421
 - for system configuration, 356
- Sound Effects key, 356
- sounds
 - adding, **299**
 - AppEvents for, **395–399**
 - in HKEY_CURRENT_USER, **421**
- Sounds applet, 299, 397–398, 398
- Sounds key, **421**
- SoundSentry key, **406**
- SourcePath value entry, 330, 502
- spam, 638
- Spare1 object
 - in REG_RESOURCE_LIST, 612
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Spare2 object
 - in REG_RESOURCE_LIST, 612
 - in REG_RESOURCE_REQUIREMENTS_LIST, 615
- Special Access option, 57
- special data types, 600
- special groups, 64
- Special Paths key, 485
- SpeechAPI key, 477
- speed
 - CPU, 438
 - keyboard, **291**, 417
 - mouse, 405
 - serial port, 336
- Spell Checker key, 360
- Speller key, 351
- Spelling key, 357

- SPLUS key, 363
- SpoolDirectory value entry, 542
- Spooler value entry, 513
- spreadsheets, 354
- .srg files, 347–348
- Srvhost key, 511
- SrvWiz key, 511
- SSL (Secure Sockets Layer), 547
- stabilization in disaster recovery, **204–215**
 - booting in normal mode for, **207–214**, 207–208, 211–212
 - of crashing systems, **214–217**
 - hardware concerns in, **218–219**
 - installed applications in, **218**
 - operating systems for, **205–206**
 - performance analysis in, 219
 - system components in, **217–218**
- standard key, 518
- StandardBias value entry, 567
- StandardName value entry, 567
- StandardStart value entry, 567
- Start menu, enhancing, **298**
- Start object, 608
- start-up information, System key for, **60–62**
- StartTime value entry, 542
- Startup and Recovery dialog box, 530, 530
- startup files
 - finding, **520**
 - in Run, **491–492**
 - in RunOnce, **492**
 - in stabilizing system, **214–215**
- startup modes for devices, 574
- StartupPrograms value entry, 562
- State value entry, 511
- static data, 11
- Stationery key, 362
- Status value entry, 542
- StatusFont value entry, 416
- stdafx.cpp file, 240
- stdafx.h file, 240–241
- StdFileEditing key, 168
- StickyKeys key, **406–409**
- StillImage key
 - in Control, 556
 - in CurrentVersion, 494
- stop bits for serial ports, 336
- StopBits value entry, 562
- STORAGE key, 572
- stores for help, **620**

- Streams key, 482
- Stretched mode for wallpaper, 281
- String Editor window, 51, 51, 93–94, 94
- strings
 - data types for, **51–52, 605–606, 609, 616**
 - ID numbers for, 379
- style for wallpaper, 283
- sub-hives, 12, 47–48
- subkeys, 5, 12
- Subscription Folder key, 488
- Substitutes key, 423–424
- SubSystems key, **555**
- Success events, auditing, 101
- Super Site for Windows 2000 site, 618
- support groups, **636**
- SupportedFaxDrivers key, 518
- SW key, 572–573
- SwapDisk value entry
 - in NonWindowsApp, 518
 - in Windows, 513
- SwapMouseButtons value entry, 418
- Switchboard Manager, 351
- SyncMgr key, 494
- SYSDLLS.DEL file, 19
- SysEvent.evt file, 19
- system, policies for, 137
- SYSTEM.ALT file, 20
- System applet
 - for CrashControl, 530, 530
 - for hardware profiles, 212
 - for user profiles, 88
- “system cannot log you on now...” message, 457
- system components
 - deleted, disasters from, **221**
 - in stabilizing system, **217–218**
- system configuration information for Office, **348–350**
 - Access, **350–353**
 - Common, **353–354**
 - Excel, 354
 - Outlook, **355**
 - PowerPoint, **355–356**
 - Publisher, **357**
 - Registration, 358
 - Word, 358
- System Control Panel, policies for, 138
- system data, nature of, 11
- system devices, Plug-and-Play identifiers for, **664–670**
- system entries, **317**

- system files, 19, 34
- SYSTEM group, 64
- System Information program, 438
- System.ini file, 4, **174–176**
 - hive for, 179
 - mapping functions for, 234
- System key
 - in Description, 54, **438–440**, 440
 - in HKEY_CURRENT_CONFIG, 66, **578–580**
 - in HKEY_LOCAL_MACHINE, 49, **60–62**, **461**, **524**
 - ControlSet001 key, 575
 - ControlSet002 key, 575
 - CurrentControlSet key. *See* Control key in CurrentControlSet; CurrentControlSet key in System
 - DISK key, **575**
 - Mounted Devices key, 576
 - Select key, **576**
 - Setup key, **576–577**
 - summary of contents, **594**
 - in HKEY_USERS, 65
 - in Policies, **491**
- System Policy Editor, **129–130**, **135–141**, 136
- System Properties applet, 436
- System Resources key, 444, 445
- System.sav file, 20
- System State option
 - for backups, 23
 - for restoring, 30–31
- System value entry, 515
- System Variables, 422, 422
- system32 directory, 39
- SystemCertificates key, 477
- SystemDefault value entry, 421
- SystemDirectory value entry, 568
- SystemPages value entry, 555
- SystemPartition value entry, 577
- SystemPrefix value entry, 577
- SystemResources key, **556–557**
- systemroot command, 43
- %SystemRoot%_system32_ntvdm.exe key, 300
- SystemRoot value entry, 502
- SystemSetupInProgress value entry, 329, 577
- SystemStartOptions value entry
 - in Control, 526
 - in CurrentControlSet, 579
- SystemTray value entry, 492
- SysVol directory, backing up, 23

T

tape

backing up to, **23–24**restoring from, **30**

Task key, 361

Task Manager, **216**

Task Switch dialog box, 285

task switching, **285**CoolSwitch value entry for, **286**CoolSwitchColumns value entry for, **287**CoolSwitchRows value entry for, **288**

Taskman value entry, 517

TcAdapter value entry, 439

TCP/IP network performance, **339–340**

Tcipip key

in Microsoft, 477

in Parameters, 339

TcpWindowSize value entry, 340

Tds key, **561–562**

TechNet program, 107

Telephony key, **494–496**

TelnetServer key, 477

TemplatePolicies key, 488

Terminal Server key, **557–558**AddIns key, **558**

in CurrentVersion, 512

DefaultUserConfiguration key, **558–559**

Dos key, 559

KeyboardType Mapping key, **559–560**Utilities key, **560**

VIDEO key, 560

Wds key, **561–562**WinStations key, **562–566**

Terminal Server Redirector key, 558

text carets. *See* cursors

Text Converters key, 363

TextEffect value entry, 406

The Computer Is Docked option, 437

The Computer Is Undocked option, 437

The Docking State Is Unknown option, 438

“The system cannot log you on now...” message, 457

themes application, 279–280

32-bit mode in boot-up, 10

[386Enh] section, 176

This Is a Portable Computer option, 437

ThreadingModel key, 381

#3 value entry, 494

Thumbnail View key, 482
Thurrott, Paul, 618
Tiled mode for wallpaper, 281, 284
TileWallpaper value entry, **283–284**, 414
time intervals for counters, 257
Time Zones key, **512**
TimeOut key, 409
Timeout value entry, 532
timer.driv driver, 176
timers, Plug-and-Play identifiers for, 665
TimeToMaximumSpeed value entry, 405
TimeToWait value entry, 409
TimeZoneInformation key, **566–567**
Tips key, 483
TIPS value entry, 492
Title value entry, 509
Today key, 361
Toggle key, 424
toggle key settings, **290, 410**, 417
ToggleKeys key, **410**
Toolbars key, 354, 360
Tools menu in Word, 126
TotalBytes value entry, 542
TotalJobs value entry, 542
TotalPages value entry, 542
TraceClass value entry, 566
TraceDebugger value entry, 566
TraceEnable value entry, 566
Tracing key, 477, 512
Tracking Data key, 362
training, **636**
Transaction Server key, 477
.Translated value entry, 446
Translators key, 356
TransmissionRetryTimeout value entry, 513
transportability of embedded documents, 170
TrDmName value entry, 459
TrDmPxOf value entry, 459
TroubleShooter value entry, 529
TrueType fonts, 307
trusted site zones, 489
TrustedDriverPath value entry, 543
TSAppCompat value entry, 558
TSEnabled value entry, 558
TShoot key, 477
TxFIFO value entry, 336
txTimeout value entry, 542
Type 1 Fonts key, 512

Type 1 Installer key, **512–513**
type command, 43
Type object
 in REG_FULL_RESOURCE_DESCRIPTOR, 607
 in REG_RESOURCE_LIST, 612
 in REG_RESOURCE_REQUIREMENTS_LIST, 614
Type value entry
 in Clip Redirector, 558
 in Terminal Server Redirector, 558
type value entry in NetRules, 510
TypeLib view in RegMaid program, **192**, 193
TZI value entry, 512

U

UA Tokens key, 488
UMAddOns key, 509
Unattended Answer file, **179–180**
Uncached Speculative Write Combining (USWC) memory, 533
undoing RegClean actions, **185–186**, 186, 199
Unicode character strings
 for backups, 29
 for value entry names, 5
Unicode Program Groups key, **429**
UNICODE Program Groups key, 65–66
Unimodem key, **497**
Uninstall key, **497–498**
uninstalling
 applications, 182, 220, **521**
 CurrentVersion key for, **497–498**
UninstallString value entry, 498
uniqueid value entry, 577
Universally Unique IDs (UUIDs), **372–374**
unknown files, Open With dialog box for, **388–389**, 388
unload function, **116**
unloading keys
 with reg unload, **116**
 with RegEdt32, 89
UNSIGNEDACTIVEX key
 in SO, 486
 in SOIEAK, 487
UntilTime value entry, 543
Update after delete option, 198
update function, **117**
Update key, **567**
UpdateMode value entry, 567
updating
 in DDE, 151–152

- keys with reg update, **117**
- policies, **567**
- Upgrade value entry, **539**
- Upgraded Type1 key, **512**
- UpgradeInProgress value entry, **329, 577**
- UpgradePath key, **355**
- upgrading Windows 2000
 - problems in, **329–330**
 - Unattended Answer file for, **179–180**
- UpperFilters value entry, **529**
- URL key, **498–499**
- UrlHistory key, **488**
- USB key, **573**
- UseAutoComplete value entry, **481**
- user configuration information for Office, **358**
 - Access, **359**
 - Binder, **359**
 - Common, **359–360**
 - Draw, **360**
 - Excel, **360**
 - Graph, **360**
 - Outlook, **360–361**
 - PowerPoint, **361**
 - Publisher, **362**
 - Query, **362**
 - Word, **362–363**
- User Configuration policies, **124**
- user groups
 - in Account, **448–455, 451**
 - for help, **620–635**
- User information key, **477**
- user interface, **278–279**
 - command prompt, **300–313, 302–303, 305–306, 309**
 - completion feature, **313**
 - data for, **11**
 - Desktop, **279–294**
 - DLL file icons, **299**
 - Easter Eggs, **300**
 - Favorites list, **294**
 - icons, **295–298, 297**
 - logon script window, **314**
 - policies for, **138**
 - pop-up menus, **295**
 - QuickEdit mode, **313**
 - sounds, **299**
 - Start menu, **298**
- User Manager, **447**
- User Manager key, **340**

- User Manager menu, Policies command, 430
- User Manager for Domains
 - for computer accounts, 453
 - for SAM, 447
- User Manager for Domains menu, Policies command, 461
- user profiles
 - HKEY_USERS for, **62–66**
 - NTUSER.DAT for, **67–69**
 - policies for, 140–141
 - in ProfileList, **511**
 - in Shell Folders, 482
 - in Win.ini, 178
- User Profiles tab, 88
- User Shell Folders key, 483
- user value entry
 - in change, 560
 - in query, 560
- User Variables, 422, 422
- UserAgent key, 488
- UserDebuggingHotKey value entry, 503
- Userdiff file, 20
- userids
 - in Aliases, 448
 - in Users, 453
- UserInfo key, 354, 360
- UserInit value entry, 515
- Userinstallable.drivers key, **513**
- UserName value entry
 - in Dos, 559
 - in F, 425
- UserOverride key, 566
- UserPreferencesMask value entry, 415
- users
 - initial settings for, **431**
 - policies for, **134**
 - in Policy, **457–459**
 - profiles for. *See* user profiles
- USERS group, 63
- Users key
 - in Builtin, 455
 - in Domains, **451–453**, 452
- UserWaitInterval value entry, 534
- USWC (Uncached Speculative Write Combining) memory, 533
- Utilities key, **560**
- UUIDs (Universally Unique IDs), **372–374**

V

- V key, 452
- value entries, 5, 12
 - accessing, **243–245**
 - adding, **108–109**
 - backing up, **109–110**
 - copying, **110–111**
 - creating, **79–80, 81, 93–95, 93**
 - deleting, **111–112**
 - exporting, **74–76, 75**
 - importing, **75–76**
 - saving, **115–116**
 - searching for, **113–114**
 - updating, **117**
- values, 5, **7–8**, 12, 48
 - accessing, **243–245**
 - default, 80
- ValuPack key, 356
- variables
 - environment, **18**
 - in Environment key, 422, 422
 - in REG_EXPAND_SZ, **605–606**
 - System and User, 422, 422
- VBA key, 477
- VBX controls, **155**
- VDD value entry, 567
- Vector object, 608
- VendorIdentifier value, 439
- verb key, 168
- Verbose value entry, 567
- verbs for objects, 165–166
- Verify Windows NT system files option, 33
- Version key, 357
- Version object
 - in REG_RESOURCE_LIST, 611
 - in REG_RESOURCE_REQUIREMENTS_LIST, 614
- Version value entry, 540
- VGA mode, 209
- VgaCompatible value entry, 560
- VgaSave value entry, 583
- video
 - in DeviceMap, 56, 441
 - Services key for, **580**
 - in System, 66
- VIDEO key
 - in DeviceMap, 56
 - in Terminal Server, 560
- video value entry, 556
- View menu

- Next Change command, 69
- Options command, 186
- Refresh Active command, 102–103
- Refresh All command, 102–103
- Scientific command, 402
- View tab, 490
- Viewer key, 356
- virtual drivers, 504
- virtual support groups, **636**
- VirtualDeviceDrivers key, **567–568**
- Visual Basic key, 478
- Visual Basic programming, 230
- VisualScriptEditor key, 478
- VisualStudio key, 478
- VmApplet value entry, 515
- Voice key, 519
- Volatile Environment key, 66, 429
- VolumeCaches key, 483

W

- w3ctrs.dll file, 272
- W3SVC key, 478, 550
- WAB key
 - in Microsoft, 478
 - in Outlook, 361
- WaitToKillAppTimeout value entry, 415
- WaitToKillService value entry
 - in Control, 526
 - in CurrentControlSet, 579
- wallpaper, **282–285**
 - Pattern value entry for, **285**
 - TileWallpaper value entry for, **283–284**
 - Wallpaper value entry for, **282–283**
 - WallpaperOriginX value entry for, **284**
 - WallpaperOriginY value entry for, **284–285**
 - WallpaperStyle value entry for, **283**
- Wallpaper value entry, **282–283**, 415, 566
- WallpaperOriginX value entry, **284**
- WallpaperOriginY value entry, **284–285**
- WallpaperStyle value entry, **283**, 415
- Warning dialog box, 103, *104*
- WaveAudio value entry
 - in MCI, 507
 - in MCI32, 508
- WBEM key, 478
- WdDLL value entry
 - in RDP-Tcp, 566

- in Tds, 562
- WdDll value entry in Console, 564
- WdFlag value entry
 - in Console, 564
 - in RDP-Tcp, 566
 - in Tds, 562
- WdName value entry
 - in Console, 564
 - in RDP-Tcp, 566
 - in Tds, 562
- WdPrefix value entry
 - in RDP-Tcp, 566
 - in Tds, 562
- Wds key, **561–562**
- Web
 - for help, **618–619**
 - W3SVC key for, 550
- WebCheck key, 499
- weight of fonts, **308–309**, 341
- Welcome key, 499
- Welcome value entry, 517
- WFProfilePath value entry, 566
- WheelScrollLines value entry, **292**, 415
- width
 - of drag rectangle, **287–288**, 413
 - of windows, 415
- Win.ini file, 4, **174–179**, 234
- Win31FileSystem value entry, **531–532**
- Win31Upgrade statement, 179
- Win32PrioritySeparation key, 544
- Win95TruncatedExtensions value entry, **532**
- windiff utility, 69
- Window PosX PosY SizX SizY value entry, 341
- WindowMetrics key, **415–416**
- WindowPosition value entry, **310–311**
- windows
 - command-prompt, **309–311**, 309
 - moving, **287–288**
 - width of, 415
- Windows 3.1 Migration Status key, 60, **429**, **519**
- Windows 4.0 key, 540
- Windows 95 and Windows 98
 - policies for, 138–141
 - remote registries with, **84–85**
- Windows 2000
 - default install path for, **520**
 - IDE controllers in, **332–334**
 - moving directory for, **321–328**

- upgrading, **329–330**
- Windows 2000 Resource Kit programs, **107**
 - add function, **108–109**
 - backup function, **109–110**
 - copy function, **110–111**
 - delete function, **111–112**
 - load function, **112–113**
 - query function, **113–114**
 - Reg.exe, **107–108**
 - restore function, **114–115**
 - save function, **115–116**
 - unload function, **116**
 - update function, **117**
- Windows 2000-specific data types, 600
- windows.adm template, 136
- Windows key, 478
 - in Control, **568**
 - in Microsoft, **479**
 - CurrentVersion key. *See* CurrentVersion key
 - Help key, **500**
 - ITStorage key, **500**
 - in Windows NT, **513**
- Windows Messaging Subsystem key, 478, 500
- Windows NT 4, policies for, **135–141**, 136, 142
- Windows NT Alpha_AXP key, 540
- Windows NT key, 478. *See also* CurrentVersion key
- Windows NT PowerPC key, 540
- Windows NT R4000 key, 540
- Windows NT x86 key, 540
- Windows on Windows, 179, **568–569**
- Windows Scripting Host key, 478
- Windows Settings for policies, **124**
- Windows value entry, 555
- WindowsEffect value entry, 406
- WindowSize value entry, **310**, 401
- WindowsUpdate key, 483, 499
- WinLogon key
 - in CurrentVersion, 499, **513–517**
 - user settings in, **338**
- WinLogon program, starting, 10
- WinMSD utility, 438
- winnt.adm template, 136
- winnt32.exe program, 36–37, 37
- winreg data key, 118
- winreg.h file, 6
- Wins key, 478
- winsta value entry
 - in change, 560

- in query, 560
- in reset, 560
- WinStations key, **562–563**
 - Console key, **564**
 - RDP-Tcp key, **564–566**
- [WinZip] section, 177–178
- Wizards key
 - in Access, **352–353**
 - in Outlook, 361
 - in Word, 363
- WizType key, 357
- WK? Settings key, 360
- Word key, 358
- Word program
 - .adm files in, **125–128**
 - as client-server applications, 158, *160*
 - with OLE, 153, 386, 387
 - user configuration information for, **362–363**
- WordArt program, 346
- WordMail key, 363
- WORDS, 402–403
- WorkDirectory value entry
 - in Console, 564
 - in Dos, 559
 - in RDP-Tcp, 566
- World Full Access Shared Parameters key, 509
- World Wide Web
 - for help, **618–619**
 - W3SVC key for, 550
- WOW key, 179
 - in Control, **568–569**
 - in CurrentVersion, **517–518**
- Wowcmdline value entry, 569
- WowFax key, 518
- wowsize value entry, 569
- Write DAC access setting, 98
- Write DAC events, 100
- Write Owner access setting, 98
- Write Owner events, auditing, 101
- WritePrivateProfileSection function, 234
- WritePrivateProfileString function, 234
- WritePrivateProfileStruct function, 234
- WriteProfileBinary function, 250
- WriteProfileInt function, 249
- WriteProfileSection function, 234
- WriteProfileString function, 234, 249
- WsxDLL value entry
 - in RDP-Tcp, 566

in Tds, 562
www prefix for URLs, 499

X

Xcopy command
for backups, **24–28**
with restoring, **31–32**
.xls extension, hive for, 375
XoffChar value entry, 562
XonChar value entry, 562

Y

Your Image File Name Here without a Path key, 507

Z

00000_ keys, 448–452
0000000 value entries, **559–560**
009 key, 510
ZoneMap key, **488**
Zones key, **489**

[Previous](#) [Table of Contents](#) [Next](#)

[HOME](#) [SUBSCRIBE](#) [SEARCH](#) [FAQ](#) [SITEMAP](#) [CONTACT US](#)

[Products](#) | [Contact Us](#) | [About Us](#) | [Privacy](#) | [Ad Info](#) | [Home](#)

Use of this site is subject to certain [Terms & Conditions](#), Copyright © 1996-2000 EarthWeb Inc. All rights reserved. Reproduction whole or in part in any form or medium without express written [permission](#) of EarthWeb is prohibited. Read EarthWeb's [privacy](#) statement.