

DISKS INCLUDED



Optimizing Windows NT™

The information
you need to
become an expert
on Windows NT!



Microsoft® WINDOWS NT RESOURCE KIT

For Windows NT Workstation and Windows NT Server Version 3.5

Microsoft Press

Optimizing
Windows NT™

Microsoft®
WINDOWS NT
RESOURCE KIT

*For Windows NT Workstation and
Windows NT Server Version 3.5*

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

Copyright © 1995 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data filed and pending.

Printed and bound in the United States of America.

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

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

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

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

Adaptec is a trademark of Adaptec, Inc. AppleTalk and Macintosh are registered trademarks of Apple Computer, Inc. DEC is a trademark of Digital Equipment Corporation. Olivetti is a registered trademark of Ing. C. Olivetti. Intel is a registered trademark and Pentium and i486 are trademarks of Intel Corporation. Microsoft, MS, MS-DOS, Win32, Windows, and XENIX are registered trademarks and Windows NT is a trademark of Microsoft Corporation in the U.S.A. and other countries. MIPS is a registered trademark of MIPS Computer Systems, Inc. NetWare and Novell are registered trademarks of Novell, Inc. UNIX is a registered trademark of Novell, Inc. in the U.S.A. and other countries, licensed exclusively through X/Open Company, Ltd. Unicode is a trademark of Unicode, Inc.

For my son Matthew: may his computers always be swift.

Contributors to this book include the following

Lead Editor:

Chris Dragich

Technical Editors:

Jeff Angus, Karin Carter, Alan Smith, and Sharon Tighe

Indexer

Jane Dow

Production Team:

Karye Cattrell, Yong Ok Chung, and Cathy Pfarr

Graphic Designer:

Sue Wyble

Graphic Artists:

Tina Anderson, Gwen Grey, Brenda Potts, Elizabeth Read, Steve Winard

Introduction

I encountered my first computer performance problem 25 years ago. A professor at my college wanted to build a tightrope for his children in the backyard. He needed to know how strong the rope would have to be, so he asked if I could help. Eager to brandish my new-found programming expertise, I readily volunteered. He told me the weight of his children and the length he wanted the tightrope to be, and I was off in a rush to the time-sharing terminal. I created the program, punched in the paper tape, and soon returned to him with the answer, gleeful with my success.

A few weeks later, another professor managing our link to the time-sharing computer stopped by and asked me if I knew what had caused the computer service to bill us an extra \$600 that month. I knew we were charged for connect time and for computation time, but the bill seemed incredible to me and I could think of no reason for it. A few days later it dawned on me. My little program to compute the tensile strength of the tightrope iterated repeatedly from 0 to degree x , where x was the angle of descent of the tightrope. Then x would be increased, but the program started back at 0 again. A very simple change to the program would have made it hundreds of times more efficient. \$600. Ouch.

So my first performance-measurement tool was a bill. Sometimes a bill is still the most effective tool, because poor computer performance costs money. And solving a performance problem by buying the wrong piece of hardware wastes money. But the biggest cost of poor performance is in the productivity of all the people who use our systems and our programs. If we could properly tune all the programs and computers in the country we could pay off the national debt. Well, make a dent anyway.

We want to help you to avoid that \$600 bill, which (given inflation) is likely to be somewhat larger today. This book describes the dynamic behavior of the Microsoft® Windows NT™ operating system and its applications, and how that behavior affects their performance.

This book will prove useful to a wide range of computer professionals, including:

- Corporate Information Systems and network administration personnel who evaluate, design, deploy, and administer Windows NT on both servers and desktop computers.
- Corporate departmental administrators who must maintain Windows NT departmental servers.
- Corporate Information Systems personnel who design, implement, and maintain mission-critical applications deployed on Windows NT.
- Independent software vendors who port 16-bit applications to Windows NT, or develop 32-bit applications for Windows NT.
- Windows NT users who make decisions concerning the addition of hardware to their computers to improve performance.

Because we figure that our readers include people with different backgrounds and with exposure to a wide variety of different systems, we have tried to be careful to define our terminology as we go along. If we're going to win this game, we need to start with a level playing field. Hopefully this will not prove too ponderous for those of you well-versed in computer science. Even if you are, you should stay on your toes: like all systems, Windows NT has its own terminology.

Once you have read this book, you will be able to:

- Determine hardware requirements for deploying Windows NT.
- Find bottlenecks in networks, servers, and desktop computers.
- Determine accurately what new hardware purchase will best enhance your productivity.
- Gain an understanding of how various activities affect the performance of your computer hardware.
- Perform capacity planning, to determine your future equipment needs for servers and desktop computers.
- Gain an understanding of the performance-critical features of Windows NT.
- Determine the effect of various design tradeoffs for optimally performing applications.
- Assess the hardware requirements for your applications.
- Determine the bottlenecks in your applications, and remove them if possible.

This book is extremely topical, which is both an asset and a liability. It is tied to the initial release of Windows NT 3.5, and reveals a host of details that are extremely important for you to understand if you are to manage the performance of Windows NT. But such details may change in subsequent releases. If you are working with a later release of Windows NT, you may find that some algorithms have been refined and some counter names have changed, and some of the hints we mention here may no longer apply. We've tried to make sure that you'll have all the tools in your toolkit to characterize any changes clearly for yourself as they emerge in new releases.

As you will discover, optimizing Windows NT is not an issue of tweaking many magic system parameters. Instead you will learn a new set of powerful tools for analyzing system performance. Each tool will lead you to the next until you have decimated the bottleneck demon.

On the floppy disk (or CD-ROM) provided with this book, you will find a synthetic load generator along with quite a few other useful tools. The synthetic load generator, which helps you perform controlled performance experiments on your system, is described in Appendix C of this book. The other utilities have online documentation. We discuss most of these tools in the text as we cover related topics, but you should browse the floppy disk for tools which might be useful in your situation. Give them a try. They don't weigh much, so you might as well carry them around.

There are a few topics we have not tried to cover at all in this book. For example, we have not tried to help you to minimize the disk space required for Windows NT on your computer. As already mentioned, we have not tried to provide an exhaustive treatise on the effects of changing the parameters listed in Appendix B, mainly because we think you'll never need to change them unless your situation is quite unusual. And we have not tried to do an exhaustive comparison of the performance of Windows NT on this or that hardware platform. Instead, we have enabled you to do that comparison in your own environment.

A book like this is the result of an enormous team effort. It is a book built upon a great foundation of software. At the base of that foundation is the dedicated team that built Windows NT (led by Dave Cutler), that tested it (led by Moshe Dunie and Ken Gregg), and that documented it (led by Chris Brown and Peggy Etchevers). I designed and helped implement Performance Monitor, but the bulk of the code was written by Hon-Wah Chan and Mike Moskowitz, with a key contribution by Bob Watson, and help from Christos Tsollis and all the NT development team members who added counters to Windows NT. Windows NT also supports a strong set of application performance tuning tools provided with this book and in the Windows NT Software Development Kit and Device Driver Kit. These were implemented by Reza Baghai (lead programmer), with help from Paramesh Vaidyanathan, Lee Smith, Tom Zakrajsek, Mark Leaming, Mark Lucovsky, Lou Perazzoli, Mark Enstrom, and Phillip Marino. And I want to give special thanks to the dedicated editing team for this book. If this book is readable at all, it is due to their many tireless hours, and if it's not it's because I mangled the text hopelessly in the first place.

All these dedicated individuals have toiled long and hard to arm you with the most advanced tools for bottleneck detection and capacity planning. So go forth and slay those bottlenecks. It's a tough job, but somebody has to do it!

Russ Blake
Fall 1994

Contents

Introduction	xxi
Chapter 1 How to Optimize Windows NT	1
Windows NT Is Always in Tune	2
Detecting Bottlenecks	3
Capacity Planning	4
Optimizing Applications	4
Performance Monitor and Other Cool Tools to Use	5
Performance Monitor Is a New Breed of Application	6
Chapter 2 Zen and the Art of Performance Monitoring	7
Computer Architecture 101	8
Bottleneck Defined	12
What a Counter Counts	14
Why You Can't Always Get Easy Answers About Performance	15
How Performance Monitor Sees a Computer	16
Performance Monitor Overview	17
How Performance Counters Are Structured	21
Selecting Computers	22
Selecting Objects	26
Selecting Counters	28
Selecting Instances	31
Custom Displays	32
The Four Performance Monitor Views	35
Chart View	36
Report View	43
Alert View	45
Log View	48
Loading and Viewing Log Files	53
Saving Settings	60
Exporting Performance Data	62
Performance Monitoring Service	63

Performance Monitor Limitations	64
Why You Don't See Any Disk Data	64
Why the Processor Queue Is Always Empty	64
Ways to Print Performance Data	65
The GUI Batch Processor	65
TCP, SNMP, and Thee	66
Crucial Hot Keys	66
Chapter 3 Detecting Processor Bottlenecks	67
Bottlenecks Are Moving Targets	68
Getting Started: Making an Overview Settings File	70
Charting the Response Surface	75
Analyzing Processor Performance	75
Why Performance Monitoring Is Free (Not!)	79
Processor Scheduling on Windows NT	80
The Mystery of the Sawtooth Queue Length	83
User Mode and Privileged Mode	86
What Multiple Processes and Threads Look Like	91
Bottlenecks at Lower Utilization	98
How the Graphics System Uses the Processor	100
Processor Usage by 16-bit Windows Applications	103
Processor Usage by 16-bit MS-DOS Applications	109
Who Started All These Processes?	112
Getting Rid of a Processor Bottleneck	114
Monitoring Multiple Processors	116
Chapter 4 Detecting Disk Bottlenecks	127
Making Sure Disk Performance Statistics Are Collected	128
Busy Disks Are Happy Disks	130
Uncovering High Disk Throughput	139
Uncovering Even Higher Disk Throughput	142
Getting Rid of a Disk Bottleneck	149
Looking at Redundant Arrays of Inexpensive Disks	150

Chapter 5 Detecting Memory Bottlenecks	157
How the Windows NT Virtual Memory System Works	158
Configuring Available Memory in Windows NT	160
Examples of Memory Activity and Paging	160
Paging with Lots of Processes	171
Monitoring the Nonpaged Pool	176
Lack of Memory Causes Disk to Suffer	177
What a Memory Hog Looks Like	178
Chapter 6 Detecting Cache Bottlenecks	181
File System Cache Overview	182
Basic Cache Experiments	184
Sequential Reading and Writing	190
Reading and Writing Randomly	197
Mapping Files Into Memory	203
Tuning the Cache	206
Chapter 7 Detecting Network Bottlenecks	207
A Profile of Network Throughput	208
A Simple Model of a Network Bottleneck	216
The Mystery of the Missing Time	217
Generalizing Network Bottleneck Detection	219
Using Role Reversal to Compare Platforms	223
Adding Clients to a Test Server	226
Server Disk Activity	233
Copying a Directory from Server to Client	237
Monitoring TCP/IP Performance	243
Monitoring NWLink Performance	249
Chapter 8 Capacity Planning	255
Performance Monitoring Service	256
Monitoring Multiple Servers	258
Archiving and Storing Performance Data	260
Analyzing Trends	261
Monitoring Desktop Computers	263

Chapter 9 Writing High-Performance Windows NT Applications	265
Managing Memory	266
Using the Kernel Wisely	267
Grappling with Graphics	269
Batch Processing for Graphics	271
High Performance Graphics	272
Managing the Device Context	273
Asynchronous Input and the Window Manager	274
Considerations for RISC Computers	274
Choosing Between API Sets	275
Chapter 10 Tuning Windows NT Applications	279
Run Performance Monitor First	280
The Windows API Profiler	283
Setting Up the Profiling Environment	286
Profiling an Application	286
Listing an Application's DLLs	286
Collecting WAP Data	287
Excluding Some APIs from Analysis with WAP	287
Running WAP on Solitaire	288
Ending WAP	290
The Call Attributed Profiler	290
Using CAP	292
Capview: a Visual Form of CAP	296
The FIOSAP Profiler	298
Using the I/O Profiler	301
The Win32 API Logger	304
Other Useful Tools	305
PView	306
PMon	308
WPerf	309
The Symbolic Debugger wt Command	309

Chapter 11 Tuning the Working Set of Your Application	311
How Working Set Tuning Can Help You	312
How the Working Set Tuner Works	314
Using the Working Set Tuner	316
Looking Inside Your Working Set	318
Chapter 12 Writing a Custom Windows NT Performance Monitor	323
Performance Monitor Source Code	324
Design Philosophy	324
Retrieving Performance Data	325
How the Performance Data Is Structured	327
Navigating Through the Performance Data Structures	330
Retrieving Counter Names and Explanations	330
Retrieving Selected Data	332
Performance Counter Definitions	333
Monitoring Within an Application	340
Chapter 13 Adding Application Performance Counters	341
Adding Performance Counters: the Big Picture	342
Object and Counter Design	343
Setting Up the Registry	343
Creating the Application's Performance Key	344
Adding Counter Names and Descriptions to the Registry	345
Other Registry Entries	351
Collecting Performance Data	352
Creating the Performance DLL	353
How the DLL Interfaces with a Performance Monitor Application	353
Error Handling in the DLL	358
Measuring Foreign Computers	358
Installing Your Application	359
Sample Code	360
Instrumenting the VGA Driver	360
Data Collection DLL	363

Appendixes

Appendix A Windows NT Performance Counters	395
AppleTalk Object	396
Browser Object	400
Cache Object	404
FTP Server Object	410
Gateway Service for NetWare Object	413
ICMP Object	418
Image Object	422
IP Object	424
LogicalDisk Object	428
MacFile Server Object	431
Memory Object	434
NBT Connection Object	439
NetBEUI Object	440
NetBEUI Resource Object	448
Network Interface Object	449
Network Segment Object	452
NWLink IPX Object	453
NWLink NetBIOS Object	460
Objects Object	467
Paging File Object	469
PhysicalDisk Object	469
Process Object	472
Process Address Space Object	477
Processor Object	486
RAS Port Object	487
RAS Total Object	490
Redirector Object	493
Server Object	501
System Object	506
TCP Object	510
Thread Object	512
Thread Details Object	515
UDP Object	515
WINS Server Object	516

Appendix B Registry Value Entries	519
CurrentControlSet\Select Subkey	520
CurrentControlSet\Control Subkeys	521
FileSystem Control Entries	522
Session Manager Control Entries	523
WOW Startup Control Entries	525
CurrentControlSet\Services Subkeys	526
Serial Subkey Entries in the CurrentControlSet\Services Subkey	527
Mouse and Keyboard Driver Entries	528
SCSI Miniport Driver Entries	532
Video Device Driver Entries	533
Registry Entries for Network Services	535
AppleTalk and MacFile Service Entries for SFM	535
Browser Service Entries	537
DiskPerf Service Entries	539
DLC System Driver Entries	540
EventLog Service Entries	541
NBF (NetBEUI) Transport Entries	542
NetLogon Service Entries	548
NWLink Transport Entries (IPX/SPX)	552
Redirector (Rdr) Service Entries	563
Remote Access Service (RAS) Entries	564
Replicator Service Entries	570
Server Service Entries	572
TCP/IP Transport Entries	579
Workstation Service Entries	594
Registry Entries for Microsoft Mail	600
Microsoft Mail Entries	600
MMF Entries for Mail	602
Microsoft Schedule+ Entries	604
Registry Entries for User Preferences	605
Console Entries for Users	605
Cursors Entry Values for Users	607
Desktop Entry Values for Users	607
Keyboard and Keyboard Layout Entries for Users	608
Mouse Entries for Users	609
Network Entries for Users	610

- Program Manager Entries for Users 611
- Recovery Entries for Users 612
- Windows Entries for Users 613
- Registry Entries for Winlogon 615
- Registry Entries for Fonts 615
 - FontCache Entries 615
 - FontDPI Entries 616
- Registry Entries for Printing 616
- Registry Entries for Software Classes 619
- Registry Entries for Subsystems 619
 - Microsoft OS/2 Version 1.x Software Registration Entries 619
 - Windows Software Registration Entries 620
- Appendix C Using Response Probe 621**
- Why You Would Use Response Probe 622
- Response Probe Design 622
 - Normal Distribution 623
 - THINK State 624
 - FILE ACCESS State 625
 - COMPUTE State 625
- Response Probe Input Files 626
- Performing Response Probe Experiments 626
 - Script Files 627
- Output Format 632

- Index 635**

Figures and Tables

Figures

Figure 2.1	Block diagram of the original IBM personal computer	8
Figure 2.2	Block diagram of a current Intel 486-based computer.....	8
Figure 2.3	Memory bus organization of a current Intel 486-based computer ...	10
Figure 2.4	Block diagram of a RISC-based personal computer.....	11
Figure 2.5	Block diagram of a multiprocessor computer	12
Figure 2.6	Performance Monitor view of personal computer hardware.....	17
Figure 2.7	Performance Monitor Chart view.....	18
Figure 2.8	Performance Monitor Report view	18
Figure 2.9	Performance Monitor Alert view	19
Figure 2.10	Performance Monitor Log view	20
Figure 2.11	Excel chart of exported Performance Monitor data.....	20
Figure 2.12	Add To Chart dialog box	22
Figure 2.13	Computer Selection dialog box	23
Figure 2.14	Overhead of remote monitoring on the monitored computer.....	24
Figure 2.15	Overhead of remote monitoring on the monitoring machine.....	24
Figure 2.16	Selecting multiple objects for logging	28
Figure 2.17	Selecting a counter for measurement.....	28
Figure 2.18	Selecting multiple counters for measurement	29
Figure 2.19	Using the Explain text.....	30
Figure 2.20	Viewing all the counters for an object at one time	31
Figure 2.21	Performance Monitor display options	33
Figure 2.22	Minimal Performance Monitor chart and report arrangement	34
Figure 2.23	All chart display options in graph mode of current activity	37
Figure 2.24	Chart histogram mode: a view of many processes' % Processor Time	41
Figure 2.25	Report with counters from multiple computers.....	44
Figure 2.26	Some common alerts and their alert logs.....	46
Figure 2.27	Setting an alert on multiple instances	48
Figure 2.28	Creating a log file	49
Figure 2.29	Analyzing data from a log file.....	49
Figure 2.30	Add To Log dialog box	50
Figure 2.31	Log Options dialog box.....	50
Figure 2.32	Log view during data collection	51
Figure 2.33	Chart of a log file with fewer than 100 snapshots	54