Inside Windows Debugging

Windows NT/2000 Native API Reference
Windows Developer Power Tools
Subclassing and Hooking with Visual Basic
Accelerated Windows Debugging 3
Inside Windows Debugging
Inside Windows NT
Advanced Windows Debugging
Troubleshooting with the Windows Sysinternals Tools
Debugging
Windows Internals
Debugging Applications for Microsoft .NET and Microsoft Windows
Windows Internals
Windows Debugging
Perl Debugged
The Old New Thing
Gray Hat Python
Windows® via C/C++
The Art of Debugging with GDB, DDD, and Eclipse
Security Warrior
The Windows 2000 Device Driver Book
Windows Debugging Notebook
Practical Foundations of Windows Debugging, Disassembling, Reversing
Windows 10 Inside Out (includes Current Book Service)
Advanced .NET Debugging
Windows Graphics Programming
Accelerated Windows Debugging 3
Debugging Windows Programs
Advanced R
Windows Kernel Programming
Windows Debugging Notebook
Practical Debugging for . NET Developers
X64 Windows Debugging
The Rootkit Arsenal: Escape and Evasion
Windows Internals, Part 2
Windows Internals
Introducing Windows 10 for IT Professionals
Windows 2000 Kernel Debugging
Inside Windows Debugging
Inside the Microsoft Build Engine
Undocumented Windows 2000 Secrets

BOWERS JOHNS

Windows NT/2000 Native API Reference
"O'Reilly Media, Inc."
"Raymond Chen is the original raconteur of Windows." --Scott Hanselman, ComputerZen.com "Raymond has been at Microsoft for many years and has seen many nuances of Windows that others could only ever hope to get a glimpse of. With this book, Raymond shares his knowledge, experience, and anecdotal stories, allowing all of us to get a better understanding of the operating system that affects millions of people every day. This book has something for everyone, is a casual read, and I highly recommend it!" --Jeffrey Richter, Author/Consultant, Cofounder of Wintellect "Very interesting read. Raymond tells the inside story of why Windows is the way it is." --Eric Gunnerson, Program Manager, Microsoft Corporation "Absolutely essential reading for understanding the history of Windows, its intricacies and quirks, and why they came about." --Matt Pietrek, MSDN Magazine's Under the Hood Columnist "Raymond Chen has become something of a legend in the software industry, and in this book you'll discover why. From his high-level reminiscences on the design of the Windows Start button to his low-level discussions of GlobalAlloc that only your inner-geek could love, The Old New Thing is a captivating collection of anecdotes that will help you to truly appreciate the difficulty inherent in designing and writing quality software." --Stephen Toub, Technical Editor, MSDN Magazine Why does Windows work the way it does? Why is Shut Down on the Start menu? (And why is there a Start button, anyway?) How can I tap into the dialog loop? Why does the GetWindowText function behave so strangely? Why are registry files called "hives"? Many of Windows' quirks have perfectly logical explanations, rooted in history. Understand them, and you'll be more productive and a lot less frustrated. Raymond Chen--who's spent more than a decade on Microsoft's Windows development team--reveals the "hidden Windows" you need to know. Chen's engaging style, deep insight, and thoughtful humor have made him one of the world's premier technology bloggers. Here he brings together behind-the-scenes explanations, invaluable technical advice, and illuminating anecdotes that bring Windows to life--and help you make the most of it. A few of the things you'll find

inside: What vending machines can teach you about effective user interfaces A deeper understanding of window and dialog management Why performance optimization can be so counterintuitive A peek at the underbelly of COM objects and the Visual C++ compiler Key details about backwards compatibility--what Windows does and why Windows program security holes most developers don't know about How to make your program a better Windows citizen

Windows Developer Power Tools
Addison-Wesley Professional
Optimize Windows system reliability and performance with Sysinternals IT pros and power users consider the free Windows Sysinternals tools indispensable for diagnosing, troubleshooting, and deeply understanding the Windows platform. In this extensively updated guide, Sysinternals creator Mark Russinovich and Windows expert Aaron Margosis help you use these powerful tools to optimize any Windows system's reliability, efficiency, performance, and security. The authors first explain Sysinternals' capabilities and help you get started fast. Next, they offer in-depth coverage of each major tool, from Process Explorer and Process Monitor to Sysinternals' security and file utilities. Then, building on this knowledge, they show the tools being used to solve real-world cases involving error messages, hangs, sluggishness, malware infections, and much more. Windows Sysinternals creator Mark Russinovich and Aaron Margosis show you how to: Use Process Explorer to display detailed process and system information Use Process Monitor to capture low-level system events, and quickly filter the output to narrow down root causes List, categorize, and manage software that starts when you start or sign in to your computer, or when you run Microsoft Office or Internet Explorer Verify digital signatures of files, of running programs, and of the modules loaded in those programs Use Autoruns, Process Explorer, Sigcheck, and Process Monitor features that can identify and clean malware infestations Inspect permissions on files, keys, services, shares, and other objects Use Sysmon to monitor security-relevant events across your network Generate memory dumps when a process meets specified criteria Execute processes remotely, and close files that were opened remotely Manage Active Directory objects and trace LDAP API calls Capture detailed data about processors, memory, and clocks Troubleshoot unbootable devices, file-in-use errors, unexplained communication, and many other problems Understand Windows core concepts that aren't well-documented elsewhere
Subclassing and Hooking with Visual Basic
Pearson Education
An Essential Reference for Intermediate and Advanced R Programmers Advanced R presents useful tools and techniques for attacking many types of R programming problems, helping you avoid mistakes and dead ends. With more than ten years of experience programming in R, the author illustrates the elegance, beauty, and flexibility at the heart of R. The book develops the necessary skills to produce quality code that can be used in a variety of circumstances. You will learn: The fundamentals of R, including standard data types and functions Functional programming as a useful framework for solving wide classes of problems The positives and negatives of metaprogramming How to write fast, memory-efficient code This book not only helps current R users become R programmers but also shows existing programmers what's special about R. Intermediate R programmers can dive deeper into R and learn new strategies for solving diverse problems while programmers from other languages can learn the details of R and understand why R works the way it does.
Accelerated Windows Debugging 3
"O'Reilly Media, Inc."
Python is fast becoming the programming language of choice for hackers, reverse engineers, and software testers because it's easy to write quickly, and it has the low-level support and libraries that make hackers happy. But until now, there has been no real manual on how to use Python for a variety of hacking tasks. You had to dig through forum posts and man pages, endlessly tweaking your own code to get everything working. Not anymore. Gray Hat Python explains the concepts behind hacking tools and techniques like debuggers, trojans, fuzzers, and emulators. But author Justin Seitz goes beyond theory, showing you how to harness existing Python-based security tools—and how to build your own when the pre-built ones won't cut it. You'll learn how to: –Automate tedious reversing and security tasks –Design and program your own debugger –Learn how to fuzz Windows drivers and create powerful fuzzers from scratch –Have fun with code and library injection, soft and hard hooking techniques, and other software trickery –Sniff secure traffic out of an encrypted web browser session –Use PyDBG, Immunity Debugger, Sulley, IDAPython, PyEMU, and more The world's best hackers are using Python to do their handiwork. Shouldn't you?

Inside Windows Debugging Addison-Wesley Professional
Microsoft Windows NT is the foundation of the new 32-bit operating system designed to support the most powerful workstation and server systems. The initial developer support for Windows NT has been phenomenal--developers have demonstrated more than 50 Windows NT applications only months after receiving the pre-release version of the software. This authoritative text--by a member of the Windows NT development group--is a a richly detailed technical overview of the design goals and architecture of Windows NT. (Operating Systems)
Inside Windows NT Pearson Education
This resource helps technical support, escalation engineers, and Windows software testers master necessary prerequisites to understand and start debugging and crash dump analysis on Windows platforms.
Advanced Windows Debugging No Starch Press
The definitive guide–fully updated for Windows 10 and Windows Server 2016 Delve inside Windows architecture and internals, and see how core components work behind the scenes. Led by a team of internals experts, this classic guide has been fully updated for Windows 10 and Windows Server 2016. Whether you are a developer or an IT professional, you'll get critical, insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand–knowledge you can apply to improve application design, debugging, system performance, and support. This book will help you: · Understand the Window system architecture and its most important entities, such as processes and threads · Examine how processes manage resources and threads scheduled for execution inside processes · Observe how Windows manages virtual and physical memory · Dig into the Windows I/O system and see how device drivers work and integrate with the rest of the system · Go inside the Windows security model to see how it manages access, auditing, and authorization, and learn about the new mechanisms in Windows 10 and Server 2016
Troubleshooting with the Windows Sysinternals Tools "O'Reilly Media, Inc."
Delve inside Windows architecture and internals—and see how core components work behind the scenes. Led by three renowned internals experts, this classic guide is fully updated for Windows 7 and Windows Server 2008 R2—and now presents its coverage in two volumes. As

always, you get critical insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand—knowledge you can apply to improve application design, debugging, system performance, and support. In Part 2, you'll examine: Core subsystems for I/O, storage, memory management, cache manager, and file systems Startup and shutdown processes Crash-dump analysis, including troubleshooting tools and techniques

Debugging Pearson Education
This training course is a combined, reformatted, improved, and modernized version of the two previous books (x64) Windows Debugging: Practical Foundations, that drew inspiration from the original lectures we developed almost 18 years ago to train support and escalation engineers in debugging and crash dump analysis of memory dumps from Windows applications, services, and systems. At that time, when thinking about what material to deliver, we realized that a solid understanding of fundamentals like pointers is needed to analyze stack traces beyond a few WinDbg commands. Therefore, this book is not about bugs or debugging techniques but about the background knowledge everyone needs to start experimenting with WinDbg and learn from practical experience and read other advanced debugging books. This body of knowledge is what the author of this book possessed before starting memory dump analysis using WinDbg 18 years ago, which resulted in the number one debugging bestseller: multi-volume Memory Dump Analysis Anthology. Now, in retrospection, we see these practical foundations as relevant and necessary to acquire for beginners as they were 18 years ago because operating systems internals, assembly language, and compiler architecture haven't changed much in those years. The book contains two separate sets of chapters and corresponding illustrations. They are named Chapter x86.NN and Chapter x64.NN respectively. The new format makes switching between and comparing x86 and x64 versions easy. Both sets of chapters can be read independently. We included x86 chapters because many 3rd-party Windows applications are still 32-bit and executed in 32-bit compatibility mode on x64 Windows systems. Almost 5 years have passed since the first edition of the combined training course that used the earlier version of Windows 10. Since then, we have also published "Practical Foundations of Linux Debugging, Disassembling, Reversing" and "Practical Foundations of ARM64 Linux Debugging, Disassembling, Reversing" books. At that time, we thought about revising our Windows course. Since then, Windows 11 appeared, and we also added Docker support for most of our Windows memory dump analysis courses. While working on the "Accelerated Windows Debugging 4D "course, we decided to make the second edition of Practical Foundations of Windows Debugging based on WinDbg from Windows 11 SDK and Visual Studio 2022 build tools and an optional Docker support for the exercise environment. We also changed the " =" operator to "" in pseudo-code for x64 AT&T disassembly syntax flavor and " The book is useful for: - Software technical support and escalation engineers; - Software engineers coming from managed code or JVM background; - Software testers; - Engineers coming from non-Wintel environments; - Windows C/C++ software engineers without assembly language background; - Security researchers without x86/x64 assembly language background; - Beginners learning Windows software reverse engineering techniques; This introductory training course can complement the more advanced course Accelerated Disassembly, Reconstruction and Reversing, Revised Edition. It may also help with advanced exercises in Accelerated Windows Memory Dump Analysis books. This book can also be used as an Intel assembly language and Windows debugging supplement for relevant undergraduate-level courses.

Windows Internals HarperChristian + ORM
For professional software developers, debugging is a way of life. This book is the definitive guide to Windows debugging, providing developers with the strategies and techniques they need to fulfill one of their most important responsibilities efficiently and effectively. Debugging Windows Programs shows readers how to prevent bugs by taking full advantage of the Visual C++ development tools and writing code in a way that makes certain types of bugs impossible. They also will learn how to reveal bugs with debugging statements that force bugs to expose themselves when the program is executed, and how to make the most of debugging tools and features available in Windows, Visual C++, MFC, and ATL. The authors provide specific solutions to the most common debugging problems, including memory corruption, resource leaks, stack problems, release build problems, finding crash locations, and multithreading problems. These essential topics are covered: The debugging process Writing C++ code for debugging Strategically using assertions, trace statements, and exceptions Windows postmortem debugging using Dr. Watson and MAP files Using the Visual C++ debugger Debugging memory Debugging multithreaded programs Debugging COM Each chapter provides developers with exactly what they need to master the subject and improve development productivity and software quality. Comprehensive, current, and practical, Debugging Windows Programs helps developers understand the debugging process and make the most of the Visual C++ debugging tools. 020170238XB04062001

Debugging Applications for Microsoft .NET and Microsoft Windows Addison Wesley Longman
Subclassing & Hooking with Visual Basic offers developers a unique way to customize Windows behavior.Windows is a message-based system. Every action you request creates one or more messages to carry out the action. These messages are passed between objects and carry with them information that gives the recipient more detail on how to interpret and act upon the message.With Subclassing and the Windows hooking mechanism ("hooks"), you can manipulate, modify, or even discard messages bound for other objects within the operating system, in the process changing the way the system behaves. What kinds of results can you achieve using the power of subclassing and hooking? Here are just a few of the possibilities: Determine when a window is being activated or deactivated and respond to this change. Display descriptions of menu items as the mouse moves across them. Disallow a user to move or resize a window. Determine where the mouse cursor is and respond accordingly. Determine when the display resolution has been changed. Monitor the system for a low system resource condition. Modify or disallow keystrokes sent to a window or a control. Create an automated testing application. Determine when an application is idle. Along with this power comes responsibility; Windows is very unforgiving if subclassing and hooking are used incorrectly. Subclassing & Hooking with Visual Basic demonstrates the various techniques for intercepting messages bound for one or more windows or controls: the intercepted message can be left in its original state or modified; afterwards, the message can be sent to its original destination or discarded.For both VB 6 and VB.NET developers, Subclassing & Hooking with Visual Basic opens up a wealth of possibilities that ordinarily would

be completely unavailable, or at least not easy to implement.

Windows Internals Microsoft Press
The ability to solve difficult problems is what makes a good engineer great. This book teaches techniques and tools for developers to tackle even the most persistent bugs. You'll find that tough issues can be made simple with the right knowledge, tools, and practices. Practical Debugging for .NET Developers will transform you into the guy or gal who everyone turns to for help. Issues covered include .NET Core, C#, Memory Leaks, Performance Problems, ASP.NET, Performance Counters, ETW Events, Production Debugging, Memory Pressure, Visual Studio, Hangs, Profiling, Deadlocks, Crashes, Memory Dumps, and Azure. * Discover the best tools in the industry to diagnose and fix problems * Learn advanced debugging techniques with Visual Studio * Fix memory leaks and memory pressure issues * Detect, profile, and fix performance problems * Find the root cause of crashes and hangs * Debug production code and third-party code * Analyze ASP.NET applications for slow performance, failed requests, and hangs * Use dump files, Performance Counters, and ETW events to investigate what happens under the hood * Troubleshoot cloud environments, including Azure VMs and App Services * Code samples in C# * Covering .NET Core, .NET Framework, Windows, and Linux

Windows Debugging Addison-Wesley Professional
Written by the founder of DumpAnalysis.org, this resource can help technical support and escalation engineers and Windows software testers without the knowledge of assembly language master necessary prerequisites to understand and start debugging and crash dump analysis on X64 Windows platforms.

Perl Debugged Sams Publishing
The start-to-finish tutorial and reference for Windows 2000 kernel debugging! The expert guide to Windows 2000 kernel debugging and crash dump analysis Interpreting Windows 2000 stop screens--in depth! Making the most of WinDbg and KD Debugging hardware: ports, BIOS, PCI and SCSI buses, and chipsets Advanced coverage: remote debugging, Debugging Extensions, Driver Verifier, and more Step-by-step crash dump analysis and kernel debugging How to interpret every element of a Windows 2000 stop screen Using WinDbg: configuring options, symbol paths, DLLs, and more Debugging hardware: ports, BIOS, PCI and SCSI buses, chipsets, and more Configuring local and remote kernel debugging environments Includes extensive code samples This comprehensive guide to Windows 2000 kernel debugging will be invaluable to anyone who must analyze and prevent Windows 2000 system crashes--especially device driver authors and debuggers. Renowned kernel debugging expert Steven McDowell covers every aspect of kernel debugging and crash dump analysis--including advanced hardware debugging and other techniques barely addressed in Microsoft's documentation. Discover what Microsoft's WinDbg debugger can (and can't) do for you, and how to configure both local and remote kernel debugging environments. Learn to use Windows 2000's crash dump feature, step by step. Learn how to start and stop errant drivers, pause target systems, retrieve system and driver state, and step through source code using breakpoints and source-level debugging. McDowell demonstrates techniques for taking control of target systems,including finding "lost" memory blocks, setting process and thread contexts, and reviewing I/O system error logs. You'll learn how to use Microsoft's powerful Debugger Extensions to run virtually any command you choose, and master the new Driver Verifier, which can detect common mistakes in driver code with unprecedented speed and accuracy.

The Old New Thing Pearson Education
PLEASE PROVIDE DESCRIPTION

Gray Hat Python Jones & Bartlett Publishers
"Jocelyn Brooke is a great writer. . . . If you care enough for literature, seek out The Scapegoat."--Elizabeth Bowen "Brooke marked out his magical, personal kingdom, different from any other writer."--Anthony Powell

Windows® via C/C++ Pearson Education
Master the intricacies of application development with unmanaged C++ code—straight from the experts. Jeffrey Richter’s classic book is now fully revised for Windows XP, Windows Vista, and Windows Server 2008. You get in-depth, comprehensive guidance, advanced techniques, and extensive code samples to help you program Windows–based applications. Discover how to: Architect and implement your applications for both 32-bit and 64-bit Windows Create and manipulate processes and jobs Schedule, manage, synchronize and destroy threads Perform asynchronous and synchronous device I/O operations with the I/O completion port Allocate memory using various techniques including virtual memory, memory-mapped files, and heaps Manipulate the default committed physical storage of thread stacks Build DLLs for delay-loading, API hooking, and process injection Using structured exception handling, Windows Error Recovery, and Application Restart services

The Art of Debugging with GDB, DDD, and Eclipse No Starch Press
This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Conquer today’s Windows 10—from the inside out! Dive into Windows 10—and really put your Windows expertise to work. Focusing on the most powerful and innovative features of Windows 10, this supremely organized reference packs hundreds of timesaving solutions, tips, and workarounds—all fully reflecting the major Windows 10 Anniversary Update. From new Cortana and Microsoft Edge enhancements to the latest security and virtualization features, you’ll discover how experts tackle today’s essential tasks—and challenge yourself to new levels of mastery. Install, configure, and personalize the newest versions of Windows 10 Understand Microsoft’s revamped activation and upgrade processes Discover major Microsoft Edge enhancements, including new support for extensions Use today’s improved Cortana services to perform tasks, set reminders, and retrieve information Make the most of the improved ink, voice, touch, and gesture support in Windows 10 Help secure Windows 10 in business with Windows Hello and Azure AD Deploy, use, and manage new Universal Windows Platform (UWP) apps Take advantage of new entertainment options, including Groove Music Pass subscriptions and connections to your Xbox One console Manage files in the cloud with Microsoft OneDrive and OneDrive for Business Use the improved Windows 10 Mail and Calendar apps and the new Skype app Fine-tune performance and troubleshoot crashes Master high-efficiency tools for managing Windows 10 in the enterprise Leverage advanced Hyper-V features, including Secure Boot, TPMs, nested virtualization, and containers In addition, this book is part of the Current Book Service from Microsoft Press. Books in this program will receive periodic updates to address significant software changes for 12 to 18 months following the original publication date via a free Web Edition. Learn more at https://www.microsoftpressstore.com/cbs.

Security Warrior Prentice Hall Professional
There is nothing like the power of the kernel in Windows - but how do you write kernel drivers to take advantage of that power? This book will show you how. The

book describes software kernel drivers programming for Windows. These drivers don't deal with hardware, but rather with the system itself: processes, threads, modules, Registry, and more. Kernel code can be used for monitoring important events, preventing some from occurring if needed. Various filters can be written that can intercept calls that a driver may be interested in. The second edition expands on existing topics, and adds new topics, such as using the Windows Filtering Platform, and describing advanced programming techniques.

The Windows 2000 Device Driver Book

Independently Published

Delve inside Windows architecture and internals—and see how core components work behind the scenes. Led by three renowned internals experts, this classic guide is fully updated for Windows 7 and Windows Server 2008 R2—and now presents its coverage in two volumes. As always, you get critical insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand—knowledge you can apply to improve application design, debugging, system performance, and support. In Part 1, you will: Understand how core system and management mechanisms work—including the object manager, synchronization, Wow64, Hyper-V, and the registry Examine the data structures and activities behind processes, threads, and jobs Go inside the Windows security model to see how it manages access, auditing, and authorization Explore the Windows networking stack from top to bottom—including APIs, BranchCache, protocol and NDIS drivers, and layered services Dig into internals hands-on using the kernel debugger, performance monitor, and other tools