

Advanced Windows Jeffrey Richter

Windowsreg; 95 and Windows NT & allow software developers to use the powerful programming technique of multithreading: dividing a single application into multiple "threads " that execute separately and get their own CPU time. This can result in significant performance gains, but also in programming headaches. Multithreading is difficult to do well, and previous coverage of the subject in Windows has been incomplete. In this book programmers will get hands-on experience in when and how to use multithreading, together with expert advice and working examples in C++ and MFC. The CD-ROM includes the code and sample applications from the book, including code that works with Internet Winsock.

This is the eBook version of the print title, Framework Design Guidelines, Second Edition . Access to all the samples, applications, and content on the DVD is available through the product catalog page www.informit.com/title/9780321545619 Navigate to the "Downloads" tab and click on the "DVD Contents" links - see instructions in back pages of your eBook. Framework Design Guidelines, Second Edition, teaches developers the best practices for designing reusable libraries for the Microsoft .NET Framework. Expanded and updated for .NET 3.5, this new edition focuses on the design issues that directly affect the programmability of a class library, specifically its publicly accessible APIs. This book can improve the work of any .NET developer producing code that other developers will use. It includes copious annotations to the guidelines by thirty-five prominent architects and practitioners of the .NET Framework, providing a lively discussion of the reasons for the guidelines as well as examples of when to break those guidelines. Microsoft architects Krzysztof Cwalina and Brad Abrams teach framework design from the top down. From their significant combined experience and deep insight, you will learn The general philosophy and fundamental principles of framework design Naming guidelines for the various parts of a framework Guidelines for the design and extending of types and members of types Issues affecting-and guidelines for ensuring-extensibility How (and how not) to design exceptions Guidelines for-and examples of-common framework design patterns Guidelines in this book are presented in four major forms: Do, Consider, Avoid, and Do not. These directives help focus attention on practices that should always be used, those that should generally be used, those that should rarely be used, and those that should never be used. Every guideline includes a discussion of its applicability, and most include a code example to help illuminate the dialogue. Framework Design Guidelines, Second Edition, is the only definitive source of best practices for managed

code API development, direct from the architects themselves. A companion DVD includes the Designing .NET Class Libraries video series, instructional presentations by the authors on design guidelines for developing classes and components that extend the .NET Framework. A sample API specification and other useful resources and tools are also included.

The Classic Guide to ATL-Now Updated for ATL 8 and Visual Studio 2005 Four leading Windows programming experts systematically reveal ATL's inner workings, explaining not just how ATL works, but why it works the way it does. Client-side developers will master ATL's resources for windowing, COM control, MFC integration, web service proxy generation, and more. Server-side programmers will discover ATL's full COM server and object services, and its extensive support for high-throughput, high-concurrency web applications, and services. Every Windows developer will learn powerful ways to increase flexibility, reduce overhead, and maximize transparency and control.

- Discover ATL's internals through diagrams, example code, and internal ATL implementation code
- Walk through wizards that simplify ATL usage in common applications
- Master string handling in C++, COM, and ATL
- Leverage ATL smart types, including CComPtr, CComQIPtr, CComBSTR, and CComVariant
- Understand and choose the right options for implementing IUnknown
- Create glue code that exposes COM objects from COM servers
- Use canned interface implementations to support object persistence, COM collections, enumerators, and connection points
- Build standalone applications and UI components with ATL window classes and controls
- Use ATL Server to develop web applications that run on Microsoft IIS

Master Windows 8.1/Windows Runtime Programming Through 80 Expert Projects This is the most complete, hands-on, solutions-focused guide to programming modern Windows applications with the Windows Runtime. Leading Windows development consultants Jeremy Likness and John Garland present easy-to-adapt C# and XAML example code for more than 80 projects. Their real-world application examples help you apply Windows 8.1's best improvements, including large tiles, the new search control, flyouts, command bars, native WinRT networking, and new deployment and sideloading options. Drawing on their pioneering experience, they illuminate key areas of the Windows Runtime API, offering uniquely detailed coverage of encryption, cloud connectivity, devices, printers, and media integration. You'll find cutting-edge tips and tricks available in no other book. This is an indispensable resource for all intermediate-to-advanced Windows developers, and for any architect building desktop, tablet, or mobile solutions with Microsoft technologies. Its focus on both C# and XAML will make it valuable to millions of Windows developers already familiar with

Silverlight, WPF, and/or .NET. Coverage includes

- Creating robust app interfaces with the newest XAML controls, including flyouts and command bars
- Saving data in a persistent “roaming zone” for syncing across Windows 8.1 devices
- Using Visual State Manager (VSM) to build apps that adapt to various device resolutions and orientations
- Integrating virtually any form of data into your apps
- Connecting with web services, RSS, Atom feeds, and social networks
- Securing apps via authentication, encrypting, signing, and single sign-on with Microsoft Account, Facebook, Google, and more
- Leveraging Windows 8.1 media enhancements that improve battery life and app performance
- Networking more effectively with Windows 8.1’s revamped HTTP implementation and new location APIs
- Using Tiles and Toasts to keep apps alive and connected, even when they aren’t running
- Enabling users to send content between devices via NFC tap and send
- Ensuring accessibility and globalizing your apps
- Efficiently debugging, optimizing, packaging, and deploying your apps
- Building sideloadable apps that don’t have to be published in Windows Store

“This book doesn’t just focus on singular concepts, it also provides end-to-end perspective on building an app in WinRT. It is one of those essential tools for Windows developers that will help you complete your software goals sooner than without it!” —Tim Heuer, Principal Program Manager Lead, XAML Platform, Microsoft Corporation

Teach yourself Visual C# 2010-one step at a time. Ideal for developers with fundamental programming skills, this practical tutorial features learn-by-doing exercises that demonstrate how, when, and why to use the features of the C# rapid application development environment. You'll learn how to use Microsoft Visual Studio 2010 and Microsoft .NET Framework 4.0; develop a solid, fundamental understanding of C# language features; and then get to work creating actual components and working applications for the Windows operating system. You'll also delve into data management technologies and Web-based applications.

Understand .NET memory management internal workings, pitfalls, and techniques in order to effectively avoid a wide range of performance and scalability problems in your software. Despite automatic memory management in .NET, there are many advantages to be found in understanding how .NET memory works and how you can best write software that interacts with it efficiently and effectively. **Pro .NET Memory Management** is your comprehensive guide to writing better software by understanding and working with memory management in .NET. Thoroughly vetted by the .NET Team at Microsoft, this book contains 25 valuable troubleshooting scenarios designed to help diagnose challenging memory problems. Readers will also benefit from a multitude of

.NET memory management "rules" to live by that introduce methods for writing memory-aware code and the means for avoiding common, destructive pitfalls. What You'll Learn Understand the theoretical underpinnings of automatic memory management Take a deep dive into every aspect of .NET memory management, including detailed coverage of garbage collection (GC) implementation, that would otherwise take years of experience to acquire Get practical advice on how this knowledge can be applied in real-world software development Use practical knowledge of tools related to .NET memory management to diagnose various memory-related issues Explore various aspects of advanced memory management, including use of Span and Memory types Who This Book Is For .NET developers, solution architects, and performance engineers

An authoritative guide to programming with Active Template Library (ATL), complete with under-the-hood details and explanations. Visual C++ programmers will learn how to develop components easier and faster by mastering ATL. The CD-ROM supplies programmers with the book's sample code as well as abundant sample controls and components.

The world's most complete guide to Windows graphics programming! Win32 GDI and DirectDraw: Accurate, under the hood, and in depth Beyond the API: Internals, restrictions, performance, and real-life problems Complete: Pixel, lines, curves, filled area, bitmap, image processing, fonts, text, metafile, printing, and more Up to date: Windows 2000 and Windows 98 graphics enhancements CD-ROM: Exclusive and professional quality generic C++ classes, reusable functions, demonstration programs, kernel mode drivers, GDI exploration tools, and more! Hewlett-Packard Professional Books To deliver high-performance Windows applications, you need an in-depth understanding of the Win32 GDI and DirectDraw--but until now, it's been virtually impossible to discover what's going on "behind" Microsoft's API calls. This book rips away the veil, giving experienced Windows programmers all the information and techniques they need to maximize performance, efficiency, and reliability! You'll discover how to make the most of Microsoft's Windows graphics APIs--including the important new graphics capabilities built into Windows 2000. Coverage includes: Uncovering the Windows system architecture and graphics system internal data structure Building graphics API "spies" that show what's going on "under the hood" Detecting GDI resource leaks and other powerful troubleshooting techniques Expert techniques for working with the Win32 GDI and DirectDraw APIs Device context, coordinate space and transformation, pixels, lines, curves, and area fills Bitmaps, image processing, fonts, text, enhanced metafiles, printing, and more "Windows Graphics

Programming" delivers extensive code, practical techniques, and unprecedented insight--plus an exclusive CD-ROM containing original system-level tools, kernel mode drivers, sample code, and generic C++ classes for Windows graphics programming without MFC. If you want to build Windows graphics applications that deliver breakthrough performance and reliability, you'll find this book indispensable.

[Advanced Windows 5](#)
[Windows 2000 Registry](#)
[Windows Internals](#)
[Advanced .NET Debugging](#)

[The Developer's Guide to the Win32 API for Windows NT 3.5 and Windows 95](#)
[CLR via C#](#)
[For Better Code, Performance, and Scalability](#)
[Windows Forms Programming in Visual Basic .NET](#)
[Advanced Windows Debugging](#)
[Inside COM+ Base Services](#)

Dependency Injection in .NET is a comprehensive guide that introduces DI and provides an in-depth look at applying DI practices to .NET apps. In it, you will also learn to integrate DI together with such technologies as Windows Communication Foundation, ASP.NET MVC, Windows Presentation Foundation and other core .NET components. Building on your existing knowledge of C# and the .NET platform, this book will be most beneficial for readers who have already built at least a few software solutions of intermediate complexity. Most examples are in plain C# without use of any particular DI framework. Later, the book introduces several well-known DI frameworks, such as StructureMap, Windsor and Spring.NET. For each framework, it presents examples of its particular usage, as well as examines how the framework relates to the common patterns presented earlier in the book.

"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 Practical explanations are given of Microsoft's networking APIs. This definitive reference covers the network programming interfaces available on the Windows 98, Windows NT/200, and Windows CE platforms. The CD-ROM features reusable code examples in Visual C++.

Delve inside the Windows Runtime - and learn best ways to design and build Windows Store apps. Guided by Jeffrey Richter, a recognized expert in Windows and .NET programming, along with principal Windows consultant Maarten van de Bospoort, you'll master essential concepts. And you'll gain practical insights and tips for how to architect, design, optimize, and debug your apps. With this book, you will: Learn how to consume Windows Runtime APIs from C# Understand the principles of architecting Windows Store apps See how to build, deploy, and secure app packages Understand how apps are activated and the process model controlling their execution Study the rich features available when working with files and folders Explore how to transfer, compress, and encrypt data via streams Design apps that give the illusion of running using live tiles, background transfers, and background tasks Share data between apps using the clipboard and the Share charm Get advice for monetizing your apps through the Windows Store About This Book Requires working knowledge of Microsoft .NET Framework, C#, and the Visual Studio IDE Targeted to programmers building Windows Store apps Some chapters also useful to those building desktop apps Technologies Covered Windows 8.1 Microsoft Visual Studio 2013

Master the new Windows Driver Model (WDM) common to Windows 98 and Windows 2000. You get theory, instruction and practice in driver development, installation and debugging. Addresses hardware and software interface issues, driver types, and a description of the new 'layer' model of WDM. ;

Kovach provides an ideal tutorial to the Direct3D APIs and shows how to use them in real gaming applications, with an emphasis on best performance practices. The author shows how to add 3D effects to any application's UI quickly, and demonstrates how to get great performance and impact from the investment.

In Essential Windows Workflow Foundation, two WF lead architects--Dharma Shukla and Bob Schmidt--offer an under-the-hood look at the technology, explaining the why and not

just the how of WF's key concepts and architecture. Serious WF developers seeking details about how to effectively utilize and extend the framework by writing activities will find cogent explanations and answers here. With simple and illustrative examples, the authors demonstrate exactly how to leverage WF's extensible programming model to craft domain-specific programs. Drawing on their unique vantage point in designing and developing WF, Shukla and Schmidt deliver authoritative coverage of The core concepts and ideas that form the heart of WF's programming model The execution model for activities, with details of the activity automaton, bookmarking, scheduling, and the threading model of the WF runtime Advanced execution concepts, including activity execution contexts, transactions, persistence points, passivation, fault handling, cancellation, compensation, and synchronization Hosting the WF runtime in applications The activity component model, with details of validation, compilation, serialization, and visualization Databinding, XAML, dependency properties, and WF program metadata Declarative conditions and rules, activity designers, and designer hosting Custom control flow patterns ranging from simple sequencing and iteration to more complex graphs and state machines Dynamic editing of running WF program instances Essential Windows Workflow Foundation is the definitive resource for developers seeking an in-depth understanding of this novel technology.

Dig deep and master the intricacies of the common language runtime, C#, and .NET development. Led by programming expert Jeffrey Richter, a longtime consultant to the Microsoft .NET team - you'll gain pragmatic insights for building robust, reliable, and responsive apps and components. Fully updated for .NET Framework 4.5 and Visual Studio 2012 Delivers a thorough grounding in the .NET Framework architecture, runtime environment, and other key topics, including asynchronous programming and the new Windows Runtime Provides extensive code samples in Visual C# 2012 Features authoritative, pragmatic guidance on difficult development concepts such as generics and threading

[The Developer's Guide to the Win32 Application Programming Interface](#)

[Framework Design Guidelines](#)

[Microsoft Visual C# 2010 Step by Step](#)

[CLR Via C#](#)

[Advanced Windows NT](#)

[PROGRAMMING APPLICATIONS FOR MICROSOFT WINDOWS, 4TH EDITION](#)

[\(With CD\)](#)

[Advanced Windows](#)

[Pro .NET Memory Management](#)

[Network Programming for Microsoft Windows](#)

[Concurrent Programming on Windows](#)

[Windows via C/C++](#)

A First Course in Game Programming Most of today's commercial games are written in C++ and are created using a game engine. Addressing both of these key elements, Programming 2D Games provides a complete, up-to-date introduction to game programming. All of the code in the book was carefully crafted using C++. As game programming techniques are introduced, students learn how to

incorporate them into their own game engine and discover how to use the game engine to create a complete game. Enables Students to Create 2D Games The text covers sprites, animation, collision detection, sound, text display, game dashboards, special graphic effects, tiled games, and network programming. It systematically explains how to program DirectX applications and emphasizes proper software engineering techniques. Every topic is explained theoretically and with working code examples. The example programs for each chapter are available at www.programming2dgames.com.

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

Here is the perfect book for Windows developers who want to join the forces of Windows NT developers. Each chapter attacks a specific topic of Windows NT programming, explaining how it fits into the big picture and then detailing what programmers need to know to exploit the feature or mechanism in their program.

- The WinForms team at Microsoft praises Chris as a definitive authority; Microsoft has named Chris one of eight Software Legends
- The content and structure are based on years of experience both building apps with WinForms as well as teaching other developers about WinForms
- Alan Cooper, the 'father of Visual Basic', has provided the foreword for the book

An in-depth architectural overview of COM+ component technologies for enterprise developers, this book offers a

detailed look by providing implementation details and sample code. Content includes scalability, queued components and MSMQ, the in-memory database, and role-based security.

See how the core components of the Windows operating system work behind the scenes—guided by a team of internationally renowned internals experts. Fully updated for Windows Server(R) 2008 and Windows Vista(R), this classic guide delivers key architectural insights on system design, debugging, performance, and support—along with hands-on experiments to experience Windows internal behavior firsthand. Delve inside Windows architecture and internals: Understand how the core system and management mechanisms work—from the object manager to services to the registry Explore internal system data structures using tools like the kernel debugger Grasp the scheduler's priority and CPU placement algorithms Go inside the Windows security model to see how it authorizes access to data Understand how Windows manages physical and virtual memory Tour the Windows networking stack from top to bottom—including APIs, protocol drivers, and network adapter drivers Troubleshoot file-system access problems and system boot problems Learn how to analyze crashes

The First In-Depth, Real-World, Insider's Guide to Powerful Windows Debugging For Windows developers, few tasks are more challenging than debugging—or more crucial. Reliable and realistic information about Windows debugging has always been scarce. Now, with over 15 years of experience two of Microsoft's system-level developers present a thorough and practical guide to Windows debugging ever written. Mario Hewardt and Daniel Pravat cover debugging throughout the entire application lifecycle and show how to make the most of the tools currently available—including Microsoft's powerful native debuggers and third-party solutions. To help you find real solutions fast, this book is organized around real-world debugging scenarios. Hewardt and Pravat use detailed code examples to illuminate the complex debugging challenges professional developers actually face. From core Windows operating system concepts to security, Windows® Vista™ and 64-bit debugging, they address emerging topics head-on—and nothing is ever oversimplified or glossed over!

This complete, hands-on Windows 2000 registry guide is

organized around the specific problems and solutions Windows professionals actually encounter. Readers get an introduction to the functions of the registry and learn everything else they need to know about using, maintaining, troubleshooting, and securing it.

[Programming the Windows Runtime by Example](#)

[Programming 2D Games](#)

[Conventions, Idioms, and Patterns for Reusable .NET Libraries](#)

[Windows 3.1](#)

[Win32 GDI and DirectDraw](#)

[The Complete Guide to Threads](#)

[A Developer's Guide](#)

[ATL Internals](#)

[Threads Primer](#)

[Programming the Microsoft Windows Driver Model](#)

[Programming Windows](#)

The Microsoft® Windows® driver model (WDM) supports Plug and Play, provides power management capabilities, and expands on the driver/minidriver approach. Written by long-time device-driver expert Walter Oney in cooperation with the Windows kernel team, this book provides extensive practical examples, illustrations, advice, and line-by-line analysis of code samples to clarify real-world driver-programming issues. And it's been updated with the latest details about the driver technologies in Windows XP and Windows 2000, plus more information about how to debug drivers. Topics covered include: Beginning a driver project and the structure of a WDM driver; NEW: Minidrivers and class drivers, driver taxonomy, the WDM development environment and tools, management checklist, driver selection and loading, approved API calls, and driver stacks Basic programming techniques; NEW: Safe string functions, memory limits, the Driver Verifier scheme and tags, the kernel handle flag, and the Windows 98 floating-point problem Synchronization; NEW: Details about the interrupt request level (IRQL) scheme, along with Windows 98 and Windows Me compatibility The I/O request packet (IRP) and I/O control operations; NEW: How to send control operations to other drivers, custom queue implementations, and how to handle and safely cancel IRPs Plug and Play for function drivers; NEW: Controller and multifunction devices, monitoring device removal in user mode, Human Interface Devices (HID), including joysticks and other game controllers, minidrivers for non-HID devices, and feature reports Reading and writing data, power management, and Windows Management Instrumentation (WMI) NEW: System wakeup, the WMI control for idle detection, and using WMIMOFCK Specialized topics and distributing drivers; NEW: USB 2.0, selective suspend, Windows Hardware Quality Lab (WHQL) certification, driver selection and loading, officially approved API calls, and driver stacks COVERS WINDOWS 98, WINDOWS ME,

WINDOWS 2000, AND WINDOWS XP! CD-ROM FEATURES: A fully searchable electronic copy of the book Sample code in Microsoft Visual C++® A Note Regarding the CD or DVD The print version of this book ships with a CD or DVD. For those customers purchasing one of the digital formats in which this book is available, we are pleased to offer the CD/DVD content as a free download via O'Reilly Media's Digital Distribution services. To download this content, please visit O'Reilly's web site, search for the title of this book to find its catalog page, and click on the link below the cover image (Examples, Companion Content, or Practice Files). Note that while we provide as much of the media content as we are able via free download, we are sometimes limited by licensing restrictions. Please direct any questions or concerns to booktech@oreilly.com.

“ Mario Hewardt ’ s Advanced .NET Debugging is an excellent resource for both beginner and experienced developers working with .NET. The book is also packed with many debugging tips and discussions of CLR internals, which will benefit developers architecting software. ” –Jeffrey Richter, consultant, trainer, and author at Wintellect “ Mario has done it again. His Advanced Windows Debugging (coauthored with Daniel Pravat) is an invaluable resource for native code debugging, and Advanced .NET Debugging achieves the same quality, clarity, and breadth to make it just as invaluable for .NET debugging. ” –Mark Russinovich, Technical Fellow, Microsoft Corporation The Only Complete, Practical Guide to Fixing the Toughest .NET Bugs Advanced .NET Debugging is the first focused, pragmatic guide to tracking down today ’ s most complex and challenging .NET application bugs. It is the only book to focus entirely on using powerful native debugging tools, including WinDBG, NTSD, and CDB, to debug .NET applications. Using these tools, author Mario Hewardt explains how to identify the real root causes of problems—far more quickly than you ever could with other debuggers. Hewardt first introduces the key concepts needed to successfully use .NET ’ s native debuggers. Next, he turns to sophisticated debugging techniques, using real-world examples that demonstrate many common C# programming errors. This book enables you to Make practical use of postmortem debugging, including PowerDBG and other “ power tools ” Understand the debugging details and implications of the new .NET CLR 4.0 Master and successfully use Debugging Tools for Windows, as well as SOS, SOSEX, CLR Profiler, and other powerful tools Gain a deeper, more practical understanding of CLR internals, such as examining thread-specific data, managed heap and garbage collector, interoperability layer, and .NET exceptions Solve difficult synchronization problems, managed heap problems, interoperability problems, and much more Generate and successfully analyze crash dumps A companion web site (advanceddotnetdebugging.com) contains all sample code, examples, and bonus content.

* No other book covers all existing versions of SQL Server – this one does. * No other book contains as detailed explanations of the inner workings of the authentication and authorization processes. * This book is the result of nearly two

years of research, and every example has been tested exhaustively to ensure the book is technically accurate. * Lewis has been a contributing author to SQL Server Magazine since its inception and trained hundreds of students on SQL Server since 1995; therefore, he knows how to make complex topics understandable to a wide range of people. * Lewis has consulted with several Fortune 500 companies on various aspects of database administration, and the book draws on his experience to highlight the critical weaknesses commonly found in even large companies with well-trained administrators. * If properly applied, the recommendations in this book result in a safer, more secure database environment. For example, this book recommended configuring firewalls to block the traffic used by the "Slammer" virus long before the virus became news. Those who read this book and followed its advice slept soundly the weekend that "Slammer" was taking the Internet down.

The acknowledged standard for unlocking the power and versatility of Microsoft Visual C++, this resource has been updated to cover the latest features that support Internet development. An enclosed CD-ROM contains valuable sample source code and sample applications developed for the book. All of which makes this volume an indispensable tool that every professional should keep close at hand.

Advanced Windows
The Developer's Guide to the Win32 API for Windows NT 3.5 and Windows 95

A guide to the workings of the common language runtime, Microsoft .NET, and C#.

A guide to computer security for software developers demonstrates techniques for writing secure applications, covering cryptography, authentication, access control, and credentials.

Contains new examples that show the latest method of programming specifically for Windows NT 4 and Windows 95. Covers the new Windows NT 4 and Windows 95 programming style and techniques.

[Windows Graphics Programming](#)

[A Guide to Multithreaded Programming](#)

[Dependency Injection In .Net](#)

[Programming Windows Security](#)

[Practical Development Throughout the Evolution of Windows, The Old New Thing](#)

[Multithreading Applications in Win32](#)

[A Comprehensive Guide to WinRT with Examples in C# and XAML](#)

[Debugging Applications for Microsoft .NET and Microsoft Windows](#)

[Programming Microsoft Visual C++](#)

[\[Komponenten-Entwicklung für verteilte Anwendungen\]. Buch](#)

Presents an updated guide to the workings and latest feature of the common language runtime, Microsoft .NET, and C#.

A guide to 32-bit programming demonstrates its elegant and powerful capabilities

over conventional 16-bit applications and includes accompanying sample code and compiled applications. Original. (Intermediate).

“When you begin using multi-threading throughout an application, the importance of clean architecture and design is critical. . . . This places an emphasis on understanding not only the platform’s capabilities but also emerging best practices. Joe does a great job interspersing best practices alongside theory throughout his book.” – From the Foreword by Craig Mundie, Chief Research and Strategy Officer, Microsoft Corporation Author Joe Duffy has risen to the challenge of explaining how to write software that takes full advantage of concurrency and hardware parallelism. In *Concurrent Programming on Windows*, he explains how to design, implement, and maintain large-scale concurrent programs, primarily using C# and C++ for Windows. Duffy aims to give application, system, and library developers the tools and techniques needed to write efficient, safe code for multicore processors. This is important not only for the kinds of problems where concurrency is inherent and easily exploitable—such as server applications, compute-intensive image manipulation, financial analysis, simulations, and AI algorithms—but also for problems that can be speeded up using parallelism but require more effort—such as math libraries, sort routines, report generation, XML manipulation, and stream processing algorithms. *Concurrent Programming on Windows* has four major sections: The first introduces concurrency at a high level, followed by a section that focuses on the fundamental platform features, inner workings, and API details. Next, there is a section that describes common patterns, best practices, algorithms, and data structures that emerge while writing concurrent software. The final section covers many of the common system-wide architectural and process concerns of concurrent programming. This is the only book you’ll need in order to learn the best practices and common patterns for programming with concurrency on Windows and .NET.

“Look it up in Petzold” remains the decisive last word in answering questions about Windows development. And in *PROGRAMMING WINDOWS, FIFTH EDITION*, the esteemed Windows Pioneer Award winner revises his classic text with authoritative coverage of the latest versions of the Windows operating system—once again drilling down to the essential API heart of Win32 programming. Topics include: The basics—input, output, dialog boxes An introduction to Unicode Graphics—drawing, text and fonts, bitmaps and metafiles The kernel and the printer Sound and music Dynamic-link libraries Multitasking and multithreading The Multiple-Document Interface Programming for the Internet and intranets Packed as always with definitive examples, this newest Petzold delivers the ultimate sourcebook and tutorial for Windows programmers at all levels working with Microsoft Windows 95, Windows 98, or Microsoft Windows NT. No aspiring or experienced developer can afford to be without it. An electronic version of this book is available on the companion CD. For customers who purchase an ebook version of this title, instructions for

downloading the CD files can be found in the ebook.

本書では、Windowsのさまざまな機能と、C/C++プログラミング言語を使ってそれらにアクセスする方法を紹介しています。Windows XP、Windows Vista、Windows Server 2008で追加されている170以上の新しい関数とWindowsの基本機能を取り上げ、システムがそれらのAPI関数や基本機能をどのように使用し、アプリケーションがそれらを最大限に利用するには、どのようにすればよいかについてOSの仕組みから詳細に解説しています。下巻では、メモリ管理、ダイナミックリンクライブラリ、構造化例外処理について取り上げています。OSの機能を熟知し、本書に掲載したプログラミング技法を豊富なサンプルコードで習得すれば、安定した質の高い高度なソフトウェア開発が可能になります。

About The Book: An update to a bestselling, practical Windows programming guide, this title is a comprehensive inside look at the Windows 2000 and 64-bit Windows environments. It provides detailed system information that's unavailable elsewhere, including architectural and implementation details and sample code. This book is an essential guide to those who want information on user-mode development above and beyond the essential course from Programming Windows. This book covers many topics. In today's world of developers, one can never be fully sure that the documentation one encounters for a technology is valid. The only way to confirm that documentation is to invest an extra thirty dollars into a reference that will teach you information you will find invaluable in your Windows development career. Programming Applications is the best place to start.

Providing an overview of the Solaris and POSIX multithreading architectures, this book explains threads at a level that is completely accessible to programmers and system architects with no previous knowledge of threads. It covers the business and technical benefits of threaded programs, along with discussions of third party software that is threaded, pointing out the benefits. It also describes the design of the Solaris MT API, with references to distinctions in POSIX, contains a set of example programs which illustrate the usage of the Solaris and POSIX APIs, and explains the use of programming tools: Thread Analyzer, LockLint, LoopTool and Debugger.

Offers application debugging techniques for Microsoft .NET Framework and Windows, covering topics such as exception monitoring, crash handlers, and multithreaded deadlocks.

[Writing Windows WDM Device Drivers](#)

[Working with ATL 8](#)

[Inside ATL \[Medienkombination\]](#) -

[SQL Server Security Distilled](#)

[Windows Runtime via C#](#)

[Essential Windows Workflow Foundation](#)

[Windows NT Win32 API SuperBible](#)

[Inside Direct3D](#)