微软编程圣典丛书（影印版）

Microsoft®

配光盘

基于服务器的应用程序内幕

（影印版）

Get Ready For Windows 2000

Features information to help you develop and deploy applications for Windows 2000

Inside Server-Based Applications

- 来自微软的第一手技术资料，深入专业的编程技术
- 应用最新的Winsock、ASP、ISAPI和SQL Server 7等工具
- 面向Windows 2000平台，开发服务器端应用程序
- 高级程序员必备

［美］Douglas J. Reilly 著

北京大学出版社
http://cbs.pku.edu.cn

微软编程圣典丛书(影印版)

基于服务器的应用程序内幕

Microsoft 公司 著

北京大学出版社

内容简介

本书是《微软编程圣典丛书（影印版）》之一，讲述开发基于服务器的解决方案。本书以详尽的内容，向 Windows 2000 用户讲述了如何将应用层转移到服务器上，如何支持瘦客户机和新型的客户机，以及如何将传统的应用程序转移到基于 Web 的传递系统上。本书首先对服务器端开发基础作了回顾，然后说明了 Windows 2000 如何支持它，详细论述了服务器端 API 和服务，并展示了如何在 Windows 2000 下将所有的服务器端片段集中在一起。为了增加本书的实用性，特以配套光盘的形式提供了丰富的程序实例以及本书的电子版。

本书由微软公司组织专家编写，具有相当的技术深度，是中、高级程序员必备的参考书。

著作权合同登记号：图字 01-2000-3032 号

书名：基于服务器的应用程序内幕（影印版）
责任著作者：Microsoft 公司 著
标准书号：ISBN 7-900629-36-X/TP・30
出版者：北京大学出版社
地址：北京市海淀区中关村北京大学校内 100871
网址：http://cbs.pku.edu.cn
电话：出版部 62752015 发行部 62765127 62754140 编辑室 62765127
电子邮箱：wdzh@mail.263.net.cn
印刷者：中国科学院印刷厂
发行者：北京大学出版社
经销者：新华书店
787 毫米×1092 毫米 16 开本 39.5 印张 1200 千字
2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷
定价：98.00 元

丛 书 序

世纪交替，IT 产业更加步履匆匆。

Microsoft 公司早已以其在编程方面的非凡成就闻名于世，并树立了在计算机软件领域和发展史上不可动摇的地位。毋庸置疑，该公司技术上的优势是其获得成功的重要因素之一。今天，它的技术不但已经变得非常强大，而且具有惊人的发展速度。尤其是 Windows 2000 技术的推出，更是展示了 Microsoft 的无穷魅力，它突然间提供了如此丰富的新特性，使我们仿佛在一瞬间便被淹没在 Windows 2000 浩瀚的技术海洋之中！

工欲善其事，必先利其器。作为 Windows 应用程序设计人员，必须紧密跟踪 Microsoft 公司的最新技术，深入 Microsoft Windows 编程的内幕，掌握关键的编程技术。这套《微软编程圣典丛书（影印版）》的推出，就是为了向有关的专业人员全面推介微软编程的核心技术，以便于他们设计高质量的 Windows 应用程序。

Microsoft 技术博大而精深，绝非某个人在短时间内所能掌握。为此，特按照技术上的逻辑关系组织成 9 个相对独立的部分，分别涉及基于服务器的应用程序、COM+基本服务、Windows 网络编程、国际化程序、MFC、Windows 编程、服务器端应用程序、Outlook 与 Exchange 编程、驱动程序模型等。每一部分的内容独立成册，集中讲述一组相关的编程技术。这套《微软编程圣典丛书（影印版）》共 9 本。特定编程领域的专业人员可以从中选取自己需要的一本或几本，使学习过程更加快速、省时、有效而直观。

这套丛书中的任何一本都涉及一门完整的编程技术，因此有着相当的深度，而且内容比较丰富。为了避免将其写成深奥而抽象的理论书，特在书中适当的位置穿插进许多贴切的程序实例。另外，每本书都有配套的 CD-ROM，内有书中的程序实例和本书的电子版。

本套丛书由 Microsoft 公司组织相关领域的专家编写。他们深谙 Microsoft 的编程技术内幕，具有丰富的程序开发经验，所以，这套丛书是他们智慧的结晶，是该领域极具权威性的著作，堪称独领风骚。

鉴于此，特向中、高级 Windows 应用程序设计人员郑重推荐这套佳作！

出版者

2000 年 9 月

Preface

Inside Server-Based Applications is designed to guide you from being a server-based applications novice to becoming a server-based applications expert. I start with some general background material about server-based applications and then move into the specifics of the APIs and technologies that allow you to create useful server-based applications for Microsoft Windows 2000.

WHO SHOULD READ THIS BOOK?

I wrote this book for the developer who wants to understand the underlying Win32 programming elements that make server-based development possible. In order to get the most from this book, you should be familiar with C++, especially the Microsoft Visual C++ programming environment. It is not essential that you have extensive experience with the Win32 API, but it won't hurt.

The Active Server Pages (ASP) and Internet Server API (ISAPI) examples in this book assume that you have some familiarity with Internet Information Server (IIS). The ASP examples also assume some familiarity with VBScript or Microsoft Visual Basic. Familiarity is not essential, however, because most of the VBScript used in the examples simply accesses methods of objects, something any C++ programmer will likely be comfortable doing. One of the examples uses JScript created on the server to be passed to the client. You'll also need to know JScript to understand the details of the example, but understanding the broader implications of creating client-side scripting and sending it to the client does not require knowledge of JScript.

Some examples use Microsoft SQL Server, both as a simple data source and as a sever-side development environment. Using SQL Server as a server-side development environment does require some understanding of SQL.

ABOUT THE EXAMPLES AND THE COMPANION CD-ROM

One thing you will notice about the first few examples is that they are implemented as console mode applications. Console mode applications are similar to the MS-DOS text mode applications from the 16-bit world, but they have some important differences. First these are true 32-bit applications. Second these applications can call dynamic-link libraries (DLLs) and other parts of the Win32 API with few exceptions. Finally the Windows 2000 service applications that I discuss are at their heart special-purpose console mode applications.

I created all of the examples on the companion CD-ROM with Microsoft Visual Studio 6.0 and the July 1999 MSDN Library. All examples have been compiled and run on Windows 2000 using Release Candidate 1 (Build 2072). I created the ASP examples with Microsoft Visual InterDev. To run the ASP and ISAPI applications, you must have Internet Information Server (at least version 4 but preferably version 5, which comes with Windows 2000). Some of the examples use SQL Server. These examples will work just as well with the Microsoft Database Engine (which is available through *http://msdn.microsoft.com/vstudio/msde/download.asp*). Some examples use TCP/IP and thus require a functional network with TCP/IP installed and configured properly with a TCP/IP naming service of some kind (such as WINS or DNS) in place. (The readme file on the companion CD-ROM also contains a list of the software required to run the examples.)

OVERVIEW

Chapters 1 and 2 present an overview of the requirements for server-based applications. Then, chapter by chapter, I discuss the APIs that allow Windows 2000 server-based applications to do what they must do. Chapter 3 discusses the Service API that allows you to create a true server-based application—one that can run even when no one is logged in to the server. Here I introduce a C++ class to support server-based applications. Chapter 4 is an introduction to the impersonation APIs, including the security implications of the basic question, "Who are you?" Chapter 5 discusses event logging and integrates it into the C++ class introduced in Chapter 3.

Chapter 6 uses the Remote Access Server (RAS) API to create virtual networks using standard phone lines. Chapter 7 discusses the Telephony API (TAPI) and Messaging API (MAPI), with special emphasis on the potential pitfalls when integrating these technologies into a Windows 2000 service application. In Chapter 7, I expand the C++ service class introduced earlier to support integrating COM technologies, such as MAPI, or any feature that requires an application with a message pump.

Chapter 8 discusses Open Database Connectivity (ODBC) and creates a class to exploit ODBC-compliant data sources. While there are many spiffy new technologies for database access, ODBC has the advantage of speed and widespread availability. In Chapter 8, I also introduce several classes supporting ODBC. You'll use these classes throughout the balance of the book.

Once I've discussed the basics of creating Windows 2000 server-based applications and the Win32 APIs that support them, I move into a discussion of several Web technologies that enable you to create server-based applications for the Internet age. Using ASP in Chapter 9 and ISAPI in Chapters 10 and 11, I show how you can create cutting-edge applications that take the best of traditional server-based applications and marry them to Internet technologies. Chapter 12 describes client communication using both named pipes and WinSock, implemented using Win32 events as well as I/O completion ports. Chapter 13 looks at database access again, but this time SQL Server specifically, with an eye toward fully exploiting SQL Server's server-side application support. Finally, in Chapters 14 and 15, I present an example of a network monitoring system that can function as a stand-alone service application or work in concert with Web clients, using ASP and ISAPI.

THE WAY I WORK

One comment about the way that I work: with some notable exceptions, I do not use frameworks such as MFC and ATL, but rather I write to the Win32 API. For traditional client applications, the frameworks are often appropriate because these tools are optimized for supporting client applications. In general these frameworks are not as useful for server-based applications. There are exceptions, of course. We can use ATL to create a server-side Microsoft ActiveX control for use with ASP. There is little reason not to use ATL for this task, as the code from the ATL Object Wizard is designed for server-side use. MFC is used extensively in the ISAPI examples, and again there is no good reason not to use the MFC ISAPI extensions. The MFC ISAPI classes can operate without taking the baggage of the rest of the MFC classes along for the ride. Once you know the underlying API or technology, you will be much more effective using the frameworks when it is appropriate to use the frameworks' wrappers around the APIs and technologies. I hope you agree.

I have enjoyed writing this book about the technologies I use almost daily to create server-based applications. I hope that you will find here the grounding you need to create the reliable, robust applications required in the increasingly server-based computing environment of the twenty-first century.

Acknowledgments

You might have noticed that this book is dedicated to my brother Bobby. He was the first person in the family to get involved with computers, and in many ways Bobby was responsible for my eventual career in computer programming. Bobby died from colon and liver cancer while I was writing this book. Shortly thereafter, during a routine screening for a minor health problem, an alert ultrasound technician noticed a spot on *my* liver.

While virtually everyone—including very good doctors in several hospitals—assured me that this was likely nothing to worry about, I worried. I also hassled one of the doctors I had been seeing for many years for the treatment of Gardner's Syndrome, the underlying condition that had killed both Bobby and my father, many years earlier. Dr. Hans Gerdes and his office assistant, Joanne Booth-Pezantez, were every bit as relentless as I was in trying to get to the bottom of this, and we did. Not long after Bobby died, I had successful liver surgery to remove two liver tumors, one malignant. Without the excellent care I received at Memorial Sloan-Kettering Cancer Center, writing this book would not have been possible.

Working with Microsoft Press has been a joy. In addition to Ben Ryan, whom I have known and worked with since before he came to Microsoft Press, the editors who worked on the book were uniformly helpful even when assaulted by my occasionally tortured prose. Kathleen Atkins, Mary Barnard, Molly Fulton, and Rebecca McKay, among others, have vastly improved and clarified my basic ideas. Jim Fuchs has been a truly amazing technical editor, not only clarifying technical points where needed but also adding insights and pushing me to improve in some areas. When I discovered a significant problem with some of the code late in the game, Jim worked to make the required changes, ensuring this would be the best book possible. Thanks to everyone at Microsoft Press!

Claire Horne of the Moore Literary Agency helped me clarify what it was I thought I had to say and presented it to several publishers until one of them agreed that there was a book in there somewhere. Thank you, Claire.

Tom Dignan of Bear Mountain Software came to my rescue as I was at sea in a pool of MAPI. When I needed insight into the inner workings of MAPI, I turned to Tom. The MAPI examples are largely drawn from code Tom kindly provided for me to use. Even the significant modifications to that code were the result of conversations with Tom. The final example in the book is a simple network monitor. For many of you, this simple example will serve as a starting point for your needs. For those of you with more complex needs, Tom's Bear Mountain Software puts out a great product named Topper that you should consider owning.

While I was writing the book, I couldn't ignore my day job. Special thanks to the folks at Home Care Information Systems (HCIS) for their kindness not only during the book-writing process but also during my illness, which was plopped right in the middle of the process. Michael Zaccardi of HCIS has the distinction of being the first person to purchase the book on line—before I even completed it. Kathy Collins and Rich Wheatley of the St. Barnabas Health Care System have continued to allow me to work on some neat projects in spite of a much-too-limited time budget. Ed Colosi and Lenny Gray have been sounding boards for many of the topics covered in this book, even in subjects beyond their special areas of expertise. Thanks all.

Family life with two teenagers can be trying in the best of circumstances. As you can gather from my brief discussion of my family's recent health problems, my children, Tim and Erin, have not had the best of circumstances. In spite of that, I am pleased to report that as of this writing they are happily in college, studying (I hope) even as I write this. Kids, I love you very much and am proud to see the grown-ups you are becoming.

Finally, this book would not have been possible without one other person in my life—my wife, Jean. I am not sure she knew what she thought she was getting into when she signed on for this marriage 21 years ago, but I bet it was a bit different from the way it turned out. She has been with me and other family members literally up and down the east coast, getting the best care we could. Many other wives would have said "Enough!" some time ago. She keeps on trucking. Here's to 100 more years, with 100-year extensions. I love you.

Douglas Reilly
doug@accessmicrosystems.com

Contents

Preface xi

Acknowledgments xv

Part I Server Basics

Chapter 1 **Server-Based Development** 3

WHAT IS A SERVER? 3

THE GROWTH OF SERVER-BASED DEVELOPMENT 5

NetWare 6

Cracks in the Fat Client Model 6

The Internet: A Solution to TCO Problems 8

The Intranet: Bandwidth Nirvana 8

Windows 2000 Hardware Support 9

WHAT MAKES SERVER-BASED DEVELOPMENT DIFFERENT? 10

The Server Is Unattended 10

Resource Leaks Are Hidden Bugs 12

Multitasking Is Essential 13

SERVER-BASED APPLICATIONS 14

Traditional Server-Based Systems 14

The Web-Based Server 15

WIN32 SUPPORT FOR SERVER-BASED PROGRAMMING 17

The Console API 19

The Service API 20

The Impersonation API 20

The Event Logging API 21

RAS, MAPI, and TAPI 21

Database Connectivity 21

Active Server Pages 22

The Internet Server API 22

Communication APIs 23

A PEEK AT SQL SERVER 25

CONCLUSION 26

Chapter 2 **Server-Based Development 101** 27

MULTITASKING BASICS 27

Cooperative Multitasking 28

Preemptive Multitasking 29

MULTITHREADING 30

SYNCHRONIZATION OBJECTS 32

Critical Sections 33

Events 40

Mutexes 41

Semaphores 42

Waitable Timers 43

Wait Functions and Thread Synchronization 44

CREATING A NEW THREAD 46

Thread Creation Functions 47

A Multithreaded Example Program 48

Part II Windows 2000 as a Server Platform

Chapter 3 **The Service API** 53

WHAT IS THE SERVICE API? 53

WHO RUNS THE SERVICE? 55

HELLO, SERVICE API WORLD! 56

Installing the Service 63

Using *CreateService* to Install the Service 64

Using *DeleteService* to Uninstall the Service 68

Starting the Service 68

Inside *ServiceMain* 68

A C++ FRAMEWORK FOR WINDOWS 2000 SERVICES 71

Writing a C++ Class for Windows 2000 Services Is Harder Than It Looks 72

Implementing the C++ Service Class 74

One More Enhancement 94

CONCLUSION 95

Chapter 4 **Impersonation APIs** 97

WINDOWS 2000 SECURITY OVERVIEW 98

IMPERSONATION BASICS 99

CLASSES OF IMPERSONATION 100
Impersonation Based on a Connection 100
Impersonation Based on Information About the Client 101
Protecting Server-Specific Resources 105
Putting Impersonation and Custom Security Together 107
CONCLUSION 113

Chapter 5 **Event Logging** 115
THE EVENT LOGGING API 118
Writing Events to the Event Log 119
Adding Event Logging to *CPPService* 125
Reading Events from the Event Log 148
CONCLUSION 154

Chapter 6 **Remote Access Service API** 155
RAS API OVERVIEW 156
AN EXAMPLE USING *RASDIAL* 156
Controlling Multiple Threads 163
The *RasDialFunc2* Callback Function 165
OTHER USES FOR RAS 166
CONCLUSION 168

Chapter 7 **Telephony API and the Messaging API** 169
TAPI 169
TAPI Versions 170
TAPI Devices 170
TAPI Errors 170
Getting Feedback from TAPI 171
A TAPI Example 171
What Else Can TAPI Do? 188
MAPI 189
Flavors of MAPI 189
Selecting the Correct Client Interface 190
Simple MAPI? 190
Full MAPI 208
Putting It All Together: Using MAPI and TAPI in a Service 227
CONCLUSION 271

Chapter 8 **The ODBC API** 273

WHAT ODBC PROVIDES 274

ODBC ALTERNATIVES 275

ODBC COMPONENTS 277

The Application 277

The Driver Manager 278

The Drivers 278

The Data Sources 279

ODBC CLASSES 279

An Example: ODBC Classes in Action 280

The *ODBCDatabase* Class 284

The *CODBCCursor* Class 300

ODBC TRAPS AND PITFALLS 301

Chapter 9 **Active Server Pages** 303

WHAT IS AN ACTIVE SERVER PAGE? 304

It's like an HTML Page... 304

...But It's Not *Exactly* the Same as an HTML Page 305

ASP STRENGTHS 310

ASP WEAKNESSES 311

OBJECTS AVAILABLE IN ASP APPLICATIONS 312

The *Application* Object 313

The *Session* Object 316

The *Server* Object 317

The *Request* Object 319

The *Response* Object 323

The *ObjectContext* Object 325

ACTIVE SERVER PAGE EXAMPLES 326

A Simple Phone Directory 327

A More Complex ASP Example: Troubleshooter 332

CONCLUSION 354

Chapter 10 **Internet Server API (ISAPI)** 355

DLL REVIEW 357

ISAPI OVERVIEW 359

Benefits of ISAPI over CGI 360

Benefits of ISAPI over ASP 361

A Simple ISAPI Extension 361
MFC—A Better Way to Create an ISAPI Extension 369
Another Use for an ISAPI Extension 392
Conclusion 399

Chapter 11 **ISAPI Filters** 401
ISAPI Filter Overview 401
Functions Exported by ISAPI Filter DLLs 403
A Sample ISAPI Filter 408
MFC ISAPI Filters 414
Conclusion 416

Chapter 12 **Client/Server Communication APIs** 417
Can We Talk? How Clients and Servers Communicate 418
TCP/IP (WinSock) 418
Named Pipes 419
WinSock or Named Pipes? 420
Dispatching Communication Requests 421
Managing Client State 421
Managing Failures 422
Dispatching Requests 423
WinSock vs. Named Pipes 457
Events vs. I/O Completion Ports vs. *QueueUserWorkItem* 457
A Generalized Client Communication Class 458
Conclusion 464

Chapter 13 **Integrating with SQL Server** 465
Transact-SQL 466
Triggers 467
Stored Procedures 471
Extended Stored Procedures 472
Conclusion 488

Part III Building Windows 2000 Server Applications

Chapter 14 **A Traditional Service Application** **493**

A Network Monitor 493

What Is a Network Monitor? 494

The Network Monitor's Structure 497

The *CMonitorService* Class 498

Monitoring Task Functions 524

Conclusion 540

Chapter 15 **Administering a Service Application on the Web** **543**

Using ASP to Monitor MonitorService 544

Interacting with Alerts 547

Interacting with Tasks 555

Using ISAPI to Monitor MonitorService 567

CMonitorService Modifications 568

The ISAPIMonitor ISAPI Extension 573

Conclusion 578

Appendix A **Setting Up Virtual Directories in IIS** **581**

Conclusion 586

Appendix B **Reacting to Service Failures** **587**

Bibliography 591

Index 595

Part I

Server Basics