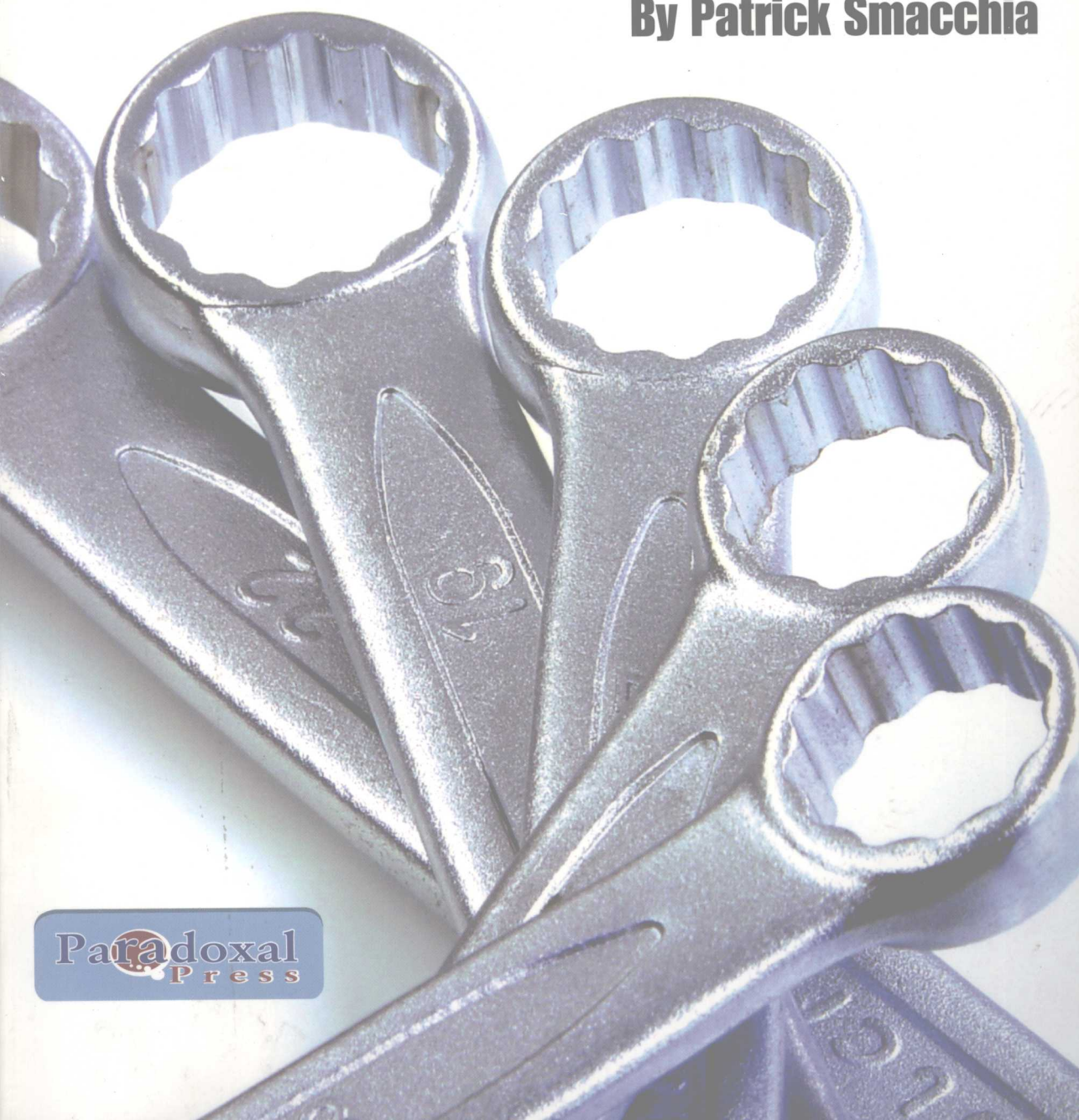


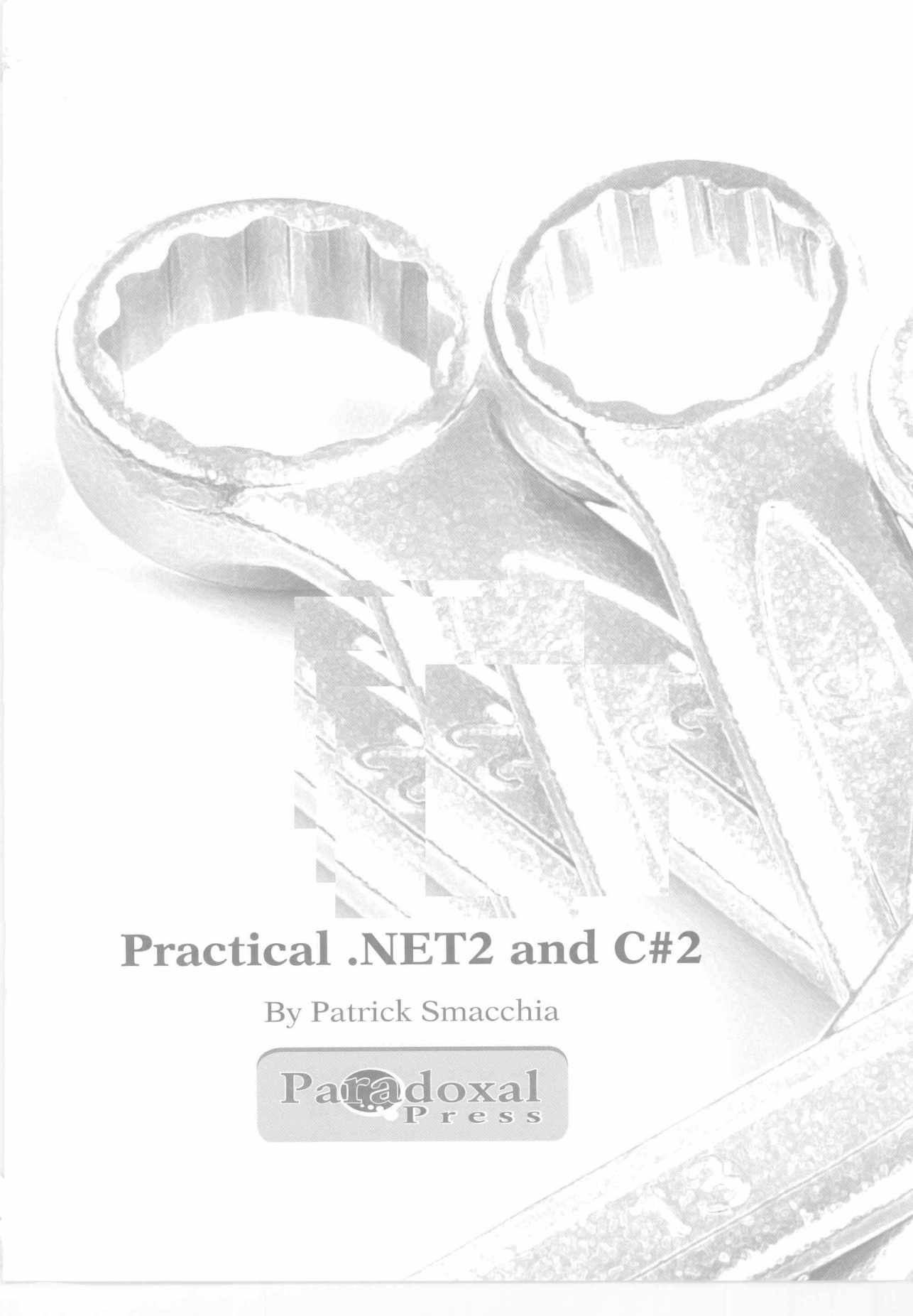
Practical _NET2 and C#2

Harness the Platform the Language the Framework

By Patrick Smacchia



**Paradoxal
Press**



Practical .NET2 and C#2

By Patrick Smacchia

Paradoxal
Press

PUBLISHED BY

Paradoxal Press

9981 Avondale Rd. NE

Redmond, WA 98052

Unites States of America

<http://www.ParadoxalPress.com>

info@ParadoxalPress.com

Copyright ©2005-2006 by Patrick Smachia.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system without written permission from Paradoxal Press, except for the inclusion of brief quotations in a review.

The Paradoxal Press logo and related trade dress are trademarks of Paradoxal Press and may not be used without written permission.

Educational facilities, companies, and organizations interested in multiple copies or licensing of this book should contact the publisher for quantity discount information. Training manuals, CD-ROMs, and portions of this book can be tailored for specific needs.

ISBN-10: 0-9766132-2-0

ISBN-13: 978-0-9766132-2-0

Library of Congress Control Number: 2005910166

Printed in the United States of America

President, Paradoxal Press: Sebastien St-Laurent

Copy Editor: Nicole St-Laurent

Interior Layout: Sebastien St-Laurent

Cover Designer: Sebastien St-Laurent

Indexer: Patrick Smacchia

Preface

About this book

The .NET **MSDN** documentation from *Microsoft* is vast and focuses on the details of each member of thousands of types. It also contains several articles in regards to the use of various parts of .NET. As a developer, I know how much the use of MSDN is fundamental when developing using *Microsoft* technologies. However, because of the volume of this documentation, it is difficult to get a global view of the features offered by .NET. From my own experience, new ideas and concepts are better acquired from a book. Yes, we could print the thousands of pages contained in MSDN but it would be hard to carry them around to read calmly in a garden or on your couch.

This book was conceived with the vision of being used conjointly with MSDN. The goal is not to enumerate the thousands of members within the thousands of .NET types but to rather explain and illustrate with concise and usable examples the multiple facets of the .NET platform, the C# language and the .NET framework. I hope that this book will give you a good insight into the motivations behind the technology and that it will carry you through the unbeaten paths to the discovery of modern software programming.

How the book is organized

Part I: The .NET platform

The first part of this book describes the underlying architecture of the .NET platform. It is in this part that you will find answers to questions such as:

What are the links between the execution of a .NET application and the underlying operating system?

What is the structure of the files produced by the compilation of my programs?

How are security and resource access managed?

How can I draw from all this to improve the quality and performance of my applications?

How can I take advantage of code already developed for *Windows* from my .NET applications?

Part II: The C#2 language and the C# 2/C++ comparison

The second part completely describes the C#2 language. This language is much closer to Java than C++. Hence, I made an effort to describe the similarities and differences between C++ and C# for each facet of the C# language. I hope that this approach will quickly answer most of the questions developers migrating from C++ might have.

Part III: The .NET framework

The third part describes the base classes of the .NET framework. The functionality of these classes can be separated into the following categories:

Collections.

The classic base classes for mathematical operations, dates and durations, folders and files, traces and debugging, regular expressions, console.

The management of I/O using data streams.

The development of graphical applications with *Windows Forms 2*.

The management of databases using *ADO.NET 2*.

The management of transactions.

The creation and manipulation of XML documents.

Distributed object applications using .NET Remoting.

The development of web applications using ASP.NET 2.

Web services.

Notes on the organization

This plan allows you to constantly have a good idea of where you are through your reading. However, it is obvious that such a vast technology has many facets which transcend this organization. For example, we have chosen to explain how to synchronize concurrent resource access in the first part since this topic is based on the underlying notions of threads and processes which are closed to platform. However, classes dedicated to synchronization are also part of the .NET framework and could have been also introduced in the third part. Finally, the C# language contains specialized keywords to simplify the use of some of these classes and therefore, this subject could also have been partially covered in the second part.

This book contains several internal references which I hope, should help you navigate through the different subjects discussed.

Who this book is for?

This book addresses itself to you as soon as you have an interest in developing under .NET, whether you are a student, professional or amateur developer, teacher, architect or technical team leader.

Each chapter has been designed to be read in a linear way but this is by no means the same for the book in its totality. The first part, the .NET platform, is considered as the most difficult but also as the most fundamental part (and according to me the most interesting one). It is simply not possible to develop properly using .NET without having some good knowledge of the underlying platform.

The **beginner** can start by learning the C# language and the object development technologies, all while discovering the possibilities of .NET.

The **reader which is experienced in other technologies** should benefit from the explanations concerning the numerous innovative features offered by .NET.

The **reader experienced with .NET 1** can use Appendix B, which references all the new features brought by .NET 2 covered in this book.

Support

This book is supported on the following site: <http://www.PracticalDOT.Net/>

There, you can download the examples found in this book. We believe that quite often, a well written sample be more valuable than a long discussion on the details of a topic. This book contains 647 examples where 523 listings are in C# and a hundred are dedicated to ASP.NET 2. They are all available online at <http://www.PracticalDOT.Net/>.

You can also contact us with your comments and feedback on this title by writing to *Paradoxal Press*: 9981 Avondale Rd. NE, Redmond WA 98052

Email: info@ParadoxalPress.com

Acknowledgments

First of all, I wish to thank my friend, **Eli Ane** for her support which gave me so much during the authoring of this title. I have also greatly appreciated the support of **Francis, France, Michel, Christine, Mathieu, Julien, Andrée, Patrick, Marie-Laure** and **Philippe**.

A thank goes to **Sébastien St Laurent** of *Paradoxal Press* for his professionalism and his great help in translating and publishing this book. Thank you also to **Xavier Cazin** of *O'Reilly France* who has supported and helped me with this project since the beginnings of .NET.

My gratitude also goes to those who have proofread and to my friends for their valuable advice:

Alain Metge, 18 years of experience. Responsible for the southern France highway software architecture.

Dr. Bertrand Le Roy, 8 years of experience, participated for 3 years at the conception of ASP.NET technologies with *Microsoft Corporation*.

Bruno Boucard, 18 years of experience, architect/trainer at the *Société Générale* bank for 8 years. *Microsoft Informed Architect*.

Frédéric De Lène Mirouze (aka. Améthyste), web development specialist, 20 years of experience, collaborates with *ELF, Glaxo, Nortel, Usinor. MCAD.NET*.

Jean-Baptiste Evain, 3 years of experience, specialist in the Common Language Infrastructure, contributor to *Mono* and *AspectDNG*.

Laurent Desmons, 10 years of experience, architect and .NET consultant, collaborates with *Péchiney, Arcelor, Sollac. MCS.D.NET*.

Matthieu Guyonnet-Duluc, 4 years of experience, commercial developer with *France Télécom*.

Dr Michel Futtersack, Master of Computing Conference, *René Descartes University*, teaching for more than 10 years in OOP conception and programming.

Nicolas Frelat, .NET consultant with 4 years of experience. Early adopter of the .NET 2 platform.

Olivier Girard, 6 years of experience, EAI specialist, architect at *Banque de France*.

Patrick Philippot, freelancer, 30 years of experience (with 19 at *IBM*), .NET MVP www.mainsoft.fr.

Sami Jaber, 8 years of experience, trainer and senior consultant with *Valtech*, collaborates with Airbus and webmaster for www.dotnetguru.org.

Sébastien Ros, 7 years of experience, O/R mapping specialist, author of the *DTM* tool, CTO of *Evaluant* www.evaluant.com.

Sébastien Vaucouleur, 8 years of experience, language specialist, collaborates with *Bull*, *Fujitsu*, research assistant at *ETH university* (Zurich).

Thibaut Barrère, freelancer, J2EE/.NET/C++ platform specialist, programming since 1984 and has recently collaborated with *Calyon*, *PPR/Redoute* and *MCS*. Open source contributor on *CruiseControl.Net*, *NAnt* and *TestDriven.Net*.

Thomas Gil, 8 years of experience, *Aspect Oriented Programing* specialist, chief of the *AspectDNG* project, consultant and independent trainer, co-webmaster of www.dotnetguru.org.

Vincent Canestrier, previous teacher at *Conservatoire National des Arts et Métiers*, previous technical director at *Cap Gemini Ernst & Young*.

I also wish to thank **Brian Grunkemeyer**, **Florin Lazar**, **Krzysztof Cwalina** and **Michael Maruchek**, all engineers at *Microsoft Corp*, for all their great answers to my questions.

Finally, I wish to thank you to have chosen this title. I sincerely hope that it will help you get all your tasks accomplished.

Contents at a Glance

PART I - The .NET2 Platform	9
1. Introduction to .NET	1
2. Assembly, module, IL language	11
3. Build, deploy and configure your .NET applications	39
4. The CLR (Common Language Runtime)	71
5. Processes, threads and synchronization	109
6. Security	153
7. Reflection, late binding, attributes	193
8. Interoperability between .NET and native code	219
PART II - The C#2 Language	255
9. Fundamental concepts of the language	257
10. The .NET 2 type system from a C#2 point of view	281
11. Classes and objects	327
12. Inheritance, polymorphism and abstraction	367
13. Generics	387
14. Unsafe code, exceptions, anonymous methods, iterators	417
PART III - The .NET2 Framework	469
15. Collections	469
16. Base Classes	497
17. Input/Output and Streams	523
18. Windows Forms Application	551
19. ADO.NET 2	585
20. Transactions	615
21. XML	631
22. .NET Remoting	653
23. ASP.NET 2	713
24. Introduction to web services development with .NET	819
Appendix	841
Index	858

Table Of Contents

Preface	xi
About this book	xi
How the book is organized	xi
Who this book is for?	xii
Support	xiii
Acknowledgments	xiii
1. Introduction to .NET	1
What is .NET?	1
History	2
.NET outside Microsoft and Windows	4
Links on .NET	5
 <i>PART I - The .NET2 Platform.</i>	 9
2. Assembly, module, IL language	11
Assemblies, modules and resource files	11
Dissecting a module	12
Analysis of an assembly with the ildasm.exe and Reflector tools	15
Assembly Attributes and Versioning	19
Strong named assemblies	22
Internationalization/localization and satellite assemblies	27
Introduction to the IL language	33
3. Build, deploy and configure your .NET applications	39
Building your applications with MSBuild	39
MSBuild: Target, Task, Property, Item and Condition	39
Advanced MSBuild	43
Configuration files	46
Assembly deployment: XCopy vs. GAC	51
Publisher policy assemblies	53
Introduction to .NET application deployment	55
Deploying an application with a cab file	57
Deploying an application with the MSI technology	59
Deploying an application with the ClickOnce technology	62
Deploying and application with the No Touch Deployment (NTD) technology ...	68
What if the .NET runtime is not installed on the target machine?	69

4. The CLR (Common Language Runtime)	71
Application Domains (AppDomain)	71
Loading the CLR inside a Windows process with the runtime host.	76
Profiling the execution of your .NET applications.	83
Locating and loading assemblies	83
Resolving types at runtime.	89
JIT compilation (Just In Time)	90
The garbage collector (GC) and the managed heap	95
Facilities to make your code more reliable	102
CLI and CLS	107
5. Processes, threads and synchronization	109
Introduction	109
Processes.	109
Threads.	112
Introduction to resource access synchronization.	117
Synchronization with volatile fields and the Interlocked class.	119
Synchronization with System.Threading.Monitor and the lock keyword	120
Synchronizing with win32 objects: mutex, events and semaphores	126
Synchronizing using the System.Threading.ReaderWriterLock class	130
Synchronizing using the SynchronizationAttribute attribute	132
The CLR's threadpool	137
Timers.	139
Calling a method asynchronously	141
Threads-resources affinities	145
Introduction to execution context	149
6. Security.	153
Introduction to Code Access Security (CAS).	153
CAS: Evidences and permissions	155
CAS: Granting permissions from evidences by applying security policies	160
CAS: The FullTrust permission.	164
CAS: Imperative permission check from the source code.	164
CAS: Declarative permissions check using attributes	168
CAS: Facilities to test and debug your mobile code.	169
CAS: The isolated storage permission.	170
.NET, Windows users and roles	170
.NET and access control to Windows resources	174
.NET and roles	179
.NET and cryptography: symmetric algorithms	181
.NET and cryptography: asymmetric algorithms (public/private keys)	183
The Data Protection API (DPAPI)	187
Authenticating your assemblies with the Authenticode and X.509 certificates	191

7. Reflection, late binding, attributes193

Reflection	193
Late bindings	197
Attributes	206
Dynamically building an assembly and using it on the fly	212

8. Interoperability between .NET and native code219

P/Invoke	219
Introduction to interoperability with the C++/CLI language	225
.NET and win32 Handles	229
Using COM objects from .NET code	230
Wrapping a .NET object into a COM Callable Wrapper (CCW)	238
Introduction to COM+.	244
Overview of the COM+ enterprise services.	245
Harnessing COM+ services in .NET classes	247

PART II - The C#2 Language.....255

9. Fundamental concepts of the language.....257

Organizing your source code	257
Compilation Steps	259
The preprocessor	260
The csc.exe compiler	264
The aliases	266
Comments and automatic documentation.....	269
Identifiers	271
Control Structures	272
The Main() method.....	278

10. The .NET 2 type system from a C#2 point of view281

The storage of objects in memory	281
Reference types vs. Value types	283
The Common Type System (CTS)	285
The System.Object class	287
Comparing objects.....	288
Cloning an object.....	290
Boxing and Unboxing	292
Primitive types.....	295
Operations on primitive types	299
Structures	303
Enumerations.....	305
Strings.....	308
Delegate classes and delegate objects.....	313
Nullable types	319
Partial types	325

11. Classes and objects.	327
Introduction	327
Vocabulary	327
Class definition	328
Fields	329
Methods	330
Properties	337
Indexer	339
Events	340
Nested types	345
Encapsulation and visibility	345
The this keyword	347
Constructors	348
Object finalization and destruction	350
Static members	356
Operator overloading	358
12. Inheritance, polymorphism and abstraction	367
Objective: code reuse	367
Class inheritance	368
Virtual methods and polymorphism	371
Abstraction	375
Interfaces	377
Virtual and abstract properties, events and indexers	383
The is and as operator	384
Solutions for code reuse	386
13. Generics	387
A C#1 problem and how to solve it with .NET 2 generics	387
.NET 2 generics: the big picture	390
Type parameter constraints	392
Members of generic types	395
Operators and generics	399
Casting and generics	401
Inheritance and generics	403
Generic methods	404
Delegates, events and generics	407
Reflection, attributes, IL and generics	409
Generics in the .NET 2 framework	414

14. Unsafe code, exceptions, anonymous methods, iterators . .417

Pointers and unsafe code	417
Using pointers in C#	418
Handling errors with exceptions	423
Exception objects and defining custom exception classes	425
Catch and finally blocks	428
Exceptions thrown from a constructor or from a finalizer	430
Exception handling and the CLR	432
Exception handling and Visual Studio.	434
Guidelines on exception management	434
Anonymous methods.	436
The C#2 compiler and anonymous methods	440
Advanced uses of anonymous methods.	446
C#1 iterators	448
C#2 iterators.	451
The C#2 compiler and iterators	456
Advanced use of C#2 iterators	459

PART III - The .NET2 Framework469

15. Collections469

Iterating through the items of a collection with the 'foreach' and 'in' keywords. . .	469
Arrays	471
Sequences.	478
Dictionaries	484
Sorting items of a collection	489
Functors as a mean to work with collections	492
Correspondence between System.Collections.Generic and System.Collections . .	496

16. Base Classes.497

Math	497
Time, date and duration	499
Drives, directories, files and paths.	503
Registry.	508
Debugging	510
Traces	513
Regular expressions.	517
Console	520

17. Input/Output and Streams523

Introduction to streams523

Reading and writing files525

Harnessing TCP/IP with sockets530

Getting information about network interfaces and status537

HTTP and FTP clients.....538

Coding an HTTP server with the HttpListener class over HTTP.SYS.....540

Support for mails protocols (SMTP and MIME)542

Buffering and compressing data streams543

Reading and writing data on the serial port.....545

Support for secure communication protocols: SSL, NTLM and Kerberos.....546

18. Windows Forms Application551

Windows user interfaces.....551

Introduction to Windows Forms development553

Facilities to develop Windows Forms applications558

Standard controls.....563

Creating custom controls565

Viewing and editing data570

Windows Forms and localization.....576

GDI+576

19. ADO.NET 2585

Introduction to databases.....585

Introduction to ADO.NET586

Connections and data providers590

Working in connected mode with DataReader597

Working in unconnected mode with DataSet600

Typed DataSet606

Bridges between the connected and the unconnected modes.....609

Bridges between objects and the relational data.....610

Functionalities specific to the SQL Server data provider612

20. Transactions615

Introduction to transactions615

System.Transactions619

Advanced usage of System.Transactions.....624

Facilities for implementing a custom RM.....626

21. XML631

Introduction	631
Introduction to XSD, XPath, XSLT and XQuery.	633
Approaches to traverse and edit an XML document	636
The cursor approach with the XmlReader and XmlWriter classes	636
The tree/DOM approach using the XmlDocument class.....	639
Traversing and editing an XML document using XPath	641
Transforming an XML document using a XSLT stylesheet	644
Bridges between relational data and XML documents	644
Bridges between objects and XML documents (XML serialization)	648
Visual Studio and XML	651

22. .NET Remoting653

Introduction	653
Marshaling By Reference (MBR)	655
Marshaling By Value (MBV) and binary serialization	657
The ObjectHandle class.	658
Object activation	660
Well-Known Object activation (WKO)	661
Client Activated Object (CAO)	664
The factory design pattern and the soapsuds.exe tool	667
Life cycle of Well-Know and Client Activated Objects	670
Configuring .NET Remoting	672
Deployment of a .NET Remoting server.....	677
Securing a .NET Remoting channel	679
Proxy and message	679
Channel.....	690
.NET context.....	700
Summary	711

23. ASP.NET 2713

Introduction.....	713
ASP.NET: The big picture	714
ASP.NET application source code.....	719
Compilation and deployment models.....	723
Web forms and controls	725
Page life cycle.....	734
ASP.NET application configuration (Web.Config files)	737
HTTP Pipeline	740
State and session management	745
The provider design pattern.....	750
Error handling	750
Trace, diagnostic and event management	752
Validation of input data	754
User controls	757
Caching.....	762
Data sources.....	771
Viewing and editing data	776
Master pages	786
ASP.NET 2 and localization.....	791
Site navigation.....	792
Security.....	794
Personalization and user profiles.....	802
Styles, Themes and Skins	805
WebParts.....	808

24. Introduction to web services development with .NET ...819

Introduction.....	819
Developing a simple web service	822
Testing and debugging a web service	824
Creating a .NET client of a web service	825
Asynchronous calls and Message Exchange Patterns	828
Using a web service from a .NET Remoting client	828
SOAP messages	829
Web services contracts and the WSDL language.....	832
Introduction to WSE and to WS-* specifications	836
WS-* specifications not yet supported by WSE	838
Introduction to WCF (Windows Communication Framework)	840

Appendix841
Appendix A: keywords of the C#2 language841
Appendix B: .NET 2 enhancements.....844
Appendix C: Introduction to design patterns.....854
Appendix D: Tools for the .NET 2 platform.....856
Index858