

Pro Visual C++/CLI and the .NET 3.5 Platform



Stephen R. G. Fraser

Pro Visual C++/CLI and the .NET 3.5 Platform

Copyright © 2009 by Stephen R. G. Fraser

All rights reserved. No part of this work 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 the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1053-5

ISBN-13 (electronic): 978-1-4302-1054-2

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Matthew Moodie

Technical Reviewer: Don Reamy

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie, Duncan Parkes, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Liz Welch

Associate Production Director: Kari Brooks-Copony

Production Editor: Laura Cheu

Compositor: Susan Glinert Stevens

Proofreader: Linda Seifert, Lisa Hamilton

Indexer: John Collin

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>.

To my daughter Shaina Shoshana, who makes everything worthwhile

Contents at a Glance

About the Author	xxi
About the Technical Reviewer	xxiii
Introduction	xxv

PART 1 ■■■ The C++/CLI Language

■ CHAPTER 1	Overview of the .NET Framework	3
■ CHAPTER 2	C++/CLI Basics	27
■ CHAPTER 3	Object-Oriented C++/CLI	85
■ CHAPTER 4	Advanced C++/CLI	141

PART 2 ■■■ .NET Framework Development in C++/CLI

■ CHAPTER 5	The .NET Framework Class Library	193
■ CHAPTER 6	Integrated XML Documentation	217
■ CHAPTER 7	Collections	243
■ CHAPTER 8	Input, Output, and Serialization	321
■ CHAPTER 9	.NET Configuration File Programming	351
■ CHAPTER 10	Basic Windows Forms Applications	373

■ CHAPTER 11	Advanced Windows Forms Applications	441
■ CHAPTER 12	Graphics Using GDI+	509
■ CHAPTER 13	ADO.NET and Database Development	577
■ CHAPTER 14	XML	623
■ CHAPTER 15	Windows Services	671
■ CHAPTER 16	Web Applications	701
■ CHAPTER 17	Web Services	753
■ CHAPTER 18	Multithreaded Programming	779
■ CHAPTER 19	Network Programming	813
■ CHAPTER 20	Assembly Programming	847
■ CHAPTER 21	Security	895

PART 3 ■■■ Unsafe/Unmanaged C++/CLI

■ CHAPTER 22	Unsafe C++ .NET Programming	925
■ CHAPTER 23	Advanced Unsafe or Unmanaged C++ .NET Programming	945
■ CHAPTER 24	The C++ Support Library	967
■ INDEX	985

Contents

About the Author	xxi
About the Technical Reviewer	xxiii
Introduction	xxv

PART 1 ■■■ The C++/CLI Language

■ CHAPTER 1	Overview of the .NET Framework	3
	What Is .NET?	3
	What Is the .NET Framework?	4
	.NET Programming Advantages	5
	A Closer Look at the .NET Framework	6
	Assemblies	7
	Common Language Runtime	11
	Common Type System	17
	Common Language Specification	20
	.NET Application Development Realms	21
	.NET Framework Class Library	23
	A Sad Note About C++/CLI Support of 3.0 and 3.5	
	Application Development Technologies	25
	Summary	26
■ CHAPTER 2	C++/CLI Basics	27
	The Obligatory “Hello World!” Program	27
	Statements	29
	Variables and C++/CLI Data Types	29
	Declaring Variables	29
	Variable Name Restrictions	31
	Predefined Data Types	32
	User-Defined Data Types	42
	Boxing and Unboxing	51
	Type Modifiers and Qualifiers	52
	Type Conversions	53
	Variable Scope	54
	Namespaces	55

Literals	55
Numeric Literals	56
Boolean Literals	57
Character Literals	58
String Literals	59
Comments	60
Operators	61
Arithmetic Operators	61
Comparisons and Logical Operators.....	62
Bitwise Operators	63
Conditional Operator.....	64
Comma Operator.....	65
Assignment Operators	65
Address of, Reference, and Indirection Operators	66
Operator Precedence	68
Flow Control Constructs	69
if Statement.....	69
switch Statement	70
Looping Constructs	71
while Loop	72
do-while Loop	72
for Loop	73
for each Loop	74
Skipping Loop Iterations	75
Breaking Out of a Loop	75
Functions	76
Passing Arguments to a Function	76
Returning Values from a Function.....	78
Prototypes	80
Function Overloading	80
Passing Arguments to the main() Function	81
Summary	83

■ CHAPTER 3 **Object-Oriented C++/CLI** 85

Object-Oriented Concepts	85
Encapsulation	86
Inheritance.....	86
Polymorphism	87
Applying Objects to Software Development.....	87

ref class/struct Basics	90
Declaring ref classes and structs	90
Using the ref class	97
Member Variables	100
Member Methods	100
Member Properties	120
Nested ref classes	130
Type Casting Between Classes	133
Abstract ref classes	135
Interfaces	137
Summary	140

CHAPTER 4 Advanced C++/CLI 141

Preprocessor Directives	141
Defining Directives	142
Conditional Directives	144
Include Directive	145
Using Directive	146
Multifile Libraries	147
Header Files	148
Source Files	148
Namespaces	149
Building Assemblies from Multifile Libraries	150
Assembly Referencing	155
Templates	157
Function Templates	157
Class Templates	158
Template Specialization and Partial Specialization	159
Template Parameters	160
Generics	164
typedef	166
Exceptions	167
Basics of Exception Handling	167
.NET Framework Base Class: Exception Classes	169
Throwing ApplicationExceptions	171
Rethrowing Exceptions and Nested try Blocks	173
Catching Multiple Exceptions	174
Catching All Previously Uncaught Exceptions	176
Executing Code Regardless of an Exception	178

Delegates and Events	180
Delegates.....	180
Events	185
Summary	190

PART 2 ■■■ .NET Framework Development in C++/CLI

CHAPTER 5	The .NET Framework Class Library	193
	Library Organizational Structure	193
	Library Namespaces	194
	System.....	194
	System::Collections	195
	System::Configuration	197
	System::Data.....	197
	System::Deployment	199
	System::Diagnostics.....	199
	System::DirectoryServices.....	200
	System::Drawing.....	201
	System::EnterpriseServices.....	202
	System::Globalization.....	202
	System::IO	203
	System::IO::Ports	204
	System::Management	204
	System::Net.....	205
	System::Reflection	206
	System::Resources.....	207
	System::Runtime::InteropServices	208
	System::Runtime::Remoting	209
	System::Runtime::Serialization	211
	System::Security.....	211
	System::Threading	212
	System::Web	213
	System::Windows::Forms	214
	System::Xml	215
	Microsoft::Win32.....	216
	Summary	216

CHAPTER 6	Integrated XML Documentation	217
	The Basics	217
	The Triple Slash Comment	218
	Adding Triple Slash Comment to Your Code	219
	Generating XML Documentation Files	220
	Viewing Integrated XML Documentation in IntelliSense	223
	Documentation Tags	224
	Functionality Tags	224
	Formatting Tags	230
	Reference Tags	235
	Documentation Example	239
	Summary	242
CHAPTER 7	Collections	243
	IEnumerable, IEnumerator, and for each	246
	Standard Collections	247
	ArrayList	247
	BitArray	251
	Hashtable and SortedList	254
	Queue and Stack	258
	Specialized Collections	260
	ListDictionary	260
	StringCollection	262
	StringDictionary	263
	NameValueCollection	264
	Generic Collections	267
	List<T>	268
	LinkedList<T>	272
	Queue<T> and Stack<T>	274
	Dictionary<K,V>, SortedDictionary<K,V>	276
	Collection<T> and ReadOnlyCollection<T>	280
	STL/CLR	281
	Containers	282
	Algorithms	314
	Summary	319

CHAPTER 8	Input, Output, and Serialization	321
	File System Input and Output	321
	Managing the File System	322
	Opening Files	329
	The Open Methods	330
	I/O Manipulation	332
	Serialization of Managed Objects	344
	Setting Up Classes for Serialization	344
	BinaryFormatter vs. SoapFormatter	346
	Serialization Using BinaryFormatter	346
	Serialization Using SoapFormatter	348
	Summary	350
CHAPTER 9	.NET Configuration File Programming	351
	What Is a Configuration File?	351
	Configuration File Inheritance	352
	Adding an app.config to a C++/CLI Project	353
	System::Configuration Namespace	354
	Reading from a .config File	356
	Modifying a .config File	359
	Updating a Value for a Key	359
	Adding and Removing Key/Value Pairs	360
	Dynamically Creating a .config File	361
	Boolean appSettings	361
	Arrays in .config Files	361
	Creating Custom Configuration Sections	364
	The Custom Configuration Section	365
	Reading and Updating Custom Configuration Sections	366
	Encrypting or Decrypting a .config File	368
	Encrypting	368
	Decrypting	371
	Summary	372
CHAPTER 10	Basic Windows Forms Applications	373
	Win Forms Are Not MFC	373
	“Hello World!” Win Form Style	374
	Customizing the Form Class	378
	Handling Win Form Delegates and Events	382

Adding Controls	387
The Label Control	388
The Button Controls	390
The Text Controls	406
The Selection Controls	421
Timers	436
Summary	439
 CHAPTER 11 Advanced Windows Forms Applications	441
ImageList	441
Views	443
ListView	443
TreeView	450
Container Controls	458
TabControl	458
SplitContainer	461
Strips	466
ToolStripContainer and ToolStripPanel	466
ToolStripManager	467
ToolStrip	467
StatusStrip	473
MenuStrip and ContextMenuStrip	477
Bells and Whistles Controls	483
PictureBox	483
MonthCalendar	485
ErrorProvider	489
NotifyIcon	493
Dialog Boxes	497
Custom Dialog Boxes	497
Common .NET Framework–Provided Dialog Boxes	505
Summary	507
 CHAPTER 12 Graphics Using GDI+	509
What Is GDI+?	509
A Quick Look at the GDI+ Namespaces	510
“Hello World!” GDI+ Style	511
OnPaint vs. PaintEventHandler	513

The Graphics Class	517
Graphics Class Members	517
Disposing of Resources with Deterministic Cleanup	519
Rendering Outside of the Paint Event	519
The Invalidate Method	523
GDI+ Coordinate Systems	523
Common Utility Structures	526
Point and PointF	527
Size and SizeF	528
Rectangle and RectangleF	529
Region	533
Drawing Strings	535
Fonts	539
Colors	543
Custom Colors	544
Named Colors	544
Pens and Brushes	545
Pens	545
Brushes	549
Rendering Prebuilt Images	553
Drawing Your Own Lines and Shapes	555
Advanced GDI+	557
Scrollable Windows	557
Optimizing GDI+	561
Double Buffering	564
Printing	572
Summary	576

■ CHAPTER 13 ADO.NET and Database Development 577

What Is ADO.NET?	577
Building a Database with Visual Studio	579
Creating a New Database	580
Adding and Loading Tables and Views to a Database	581
Building Stored Procedures	588
Managed Providers	589
Connected ADO.NET	590
Using Simple Connected ADO.NET	590
Using Connected ADO.NET with Transactions	601

Disconnected ADO.NET	606
The Core Classes	606
Creating a Table Manually in Code	610
Developing with Disconnected ADO.NET	611
Summary	622
CHAPTER 14 XML	623
What Is XML?	623
The .NET Framework XML Implementations	624
Forward-Only Access	625
Reading from an XML File	626
Validating an XML File	633
Writing a New XML Stream	640
Updating an Existing XML File	644
Working with DOM Trees	647
Reading a DOM Tree	651
Updating a DOM Tree	654
Writing XmlNodes in a DOM Tree	656
Navigating with XPathNavigator	658
Basic XPathNavigator	660
XPathNavigator Using XPath Expressions	662
XML and ADO.NET	667
Summary	669
CHAPTER 15 Windows Services	671
What Are Windows Services?	671
Architecture of Windows Services	673
Service Application	673
Service Control Application	674
Service Configuration Application	674
The ServiceProcess Namespace	675
Creating Windows Services	675
Autogenerated Windows Service	676
Customizing the Windows Service	681
Installing and Uninstalling Windows Services	687
Managing Windows Services	691
Services Application	691
Custom Service Control Application	693

Debugging Windows Services	697
Attaching the Debugger to the Windows Service	697
A Special Main() Function	698
Summary	700

■ CHAPTER 16 Web Applications 701

C++/CLI Restriction for ASP.NET Support	701
Configuring Visual Studio for ASP.NET	702
“Hello World,” Web Form Style	702
Web Form Controls	709
Label	711
Image	714
TextBox	717
Buttons and Hyperlinks	720
Lists	726
Tables	731
User Controls	738
Creating a User Control	738
Statically Implementing a User Control	741
Dynamically Implementing a User Control	743
ASP.NET Master Pages	746
Master Page	746
Web Content Page	748
Summary	751

■ CHAPTER 17 Web Services 753

What Are Web Services?	753
Components of a Web Service	754
Communication Protocols	754
Description Service	755
Discovery Service	755
The Web Services Namespaces	755
A Simple Web Service	756
Accessing a Web Service Using HTTP POST	764
Accessing a Web Service Using SOAP	766
Debugging a Web Service	769

Passing Data Using a Web Service	770
Creating the Web Service Class Definition.....	771
Returning a DataSet.....	772
Inserting, Updating, and Deleting Rows in a DataSet.....	772
Authors DataSet Processing Web Service Client.....	774
Summary	778

CHAPTER 18 Multithreaded Programming 779

What Is Multithreaded Programming?	779
Basic .NET Framework Class Library Threading	780
Thread State	781
Thread Priorities	783
Using Threads	784
Starting Threads.....	784
Getting a Thread to Sleep	787
Aborting Threads	789
Joining Threads	791
Interrupting, Suspending, and Resuming Threads	792
Using Thread Pools	795
Synchronization	797
The ThreadStatic Attribute.....	798
The Interlocked Class.....	800
The Monitor Class.....	802
The Mutex Class	805
The ReaderWriterLock Class	809
Summary	812

CHAPTER 19 Network Programming 813

The Network Namespaces	813
Connection-Oriented Sockets	814
The TCP Server	814
The TCP Client.....	821
Connectionless Sockets	824
UDP Server	824
UDP Client Example	828
Using Connect() with UDP	829

Socket Helper Classes and Methods	830
TcpListener	830
TcpClient	831
TCP Helper Class Example	832
UdpClient	835
Changing Socket Options	837
Asynchronous Sockets	839
Accepting Connections	839
Connecting to a Connection	841
Disconnecting from a Connection	841
Sending a Message	842
Receiving a Message	842
Asynchronous TCP Server	843
Summary	846

■ CHAPTER 20 Assembly Programming

Reflection	847
Examining Objects	848
Dynamically Invoking or Late-Binding Objects	854
Attributes	857
Creating a Custom Attribute	857
Implementing a Custom Attribute	860
Using a Custom Attribute	861
Shared Assemblies	865
The Global Assembly Cache	866
Adding Assemblies to the GAC	867
The Shared Assembly's Strong Name	867
Resigning an Assembly	868
Signcoded Digital Signature	868
Versioning	868
No DLL Hell Example	870
Application Configuration Files	874
Resources	875
Creating Resources	876
Embedding Resources	877
Accessing Resources	881

Globalization and Localization	883
The Globalization Tools	884
The Localization Tools	886
Building a Multicultural Windows Application	887
Building a Multicultural Console Application	890
Summary	893
CHAPTER 21 Security	895
The Security Namespaces	895
Role-Based Security	896
Identities	896
Principal	897
Working with Identities and Principals	898
Securing Your Code Using Roles	900
Code Access Security	903
Permissions	903
Policy Statement	905
Code Groups	906
Evidence	910
Securing Your Code Using CAS	915
Summary	922
 PART 3 ■■■ Unsafe/Unmanaged C++/CLI	
CHAPTER 22 Unsafe C++ .NET Programming	925
What Is Unsafe Code?	925
Why Do We Still Need Unsafe Code?	926
Creating Unsafe Code	927
The Managed and Unmanaged #pragma Directives	927
Unmanaged Arrays	930
Unmanaged Classes/Structs	931
Pointers	935
Placing Managed Classes in Unmanaged Classes	940
Summary	943

CHAPTER 23	Advanced Unsafe or Unmanaged C++ .NET Programming	945
	P/Invoke	945
	Calling DLLs Without P/Invoke	946
	Using P/Invoke	948
	Data Marshaling	953
	MarshalAsAttribute	953
	Marshaling Strings	954
	Marshaling Ref and Value Classes	955
	Accessing COM Components from .NET	957
	Interop Assembly	959
	Creating the Interop Assembly	959
	Invoking the Interop Assembly	961
	Handling COM Object Errors	964
	Late Binding a COM Object	964
	Summary	966
CHAPTER 24	The C++ Support Library	967
	The C++ Support Library Headers	967
	auto_handle	968
	gcroot and auto_gcroot	970
	com::ptr	973
	PtrToStringChars() Function	976
	marshal_as and marshal_context	977
	_safe_bool	979
	lock	981
	Summary	984
INDEX		985

About the Author



STEPHEN R. G. FRASER is a senior consultant at Allin Consulting, where he develops custom software solutions for top-tier companies in northern California. Stephen has over 20 years of IT experience working for a number of consulting companies, ranging from the large consulting firms of EDS and Andersen Consulting (Accenture) to start-up e-business and medical companies. His IT experience covers all aspects of application and Web development and management, ranging from initial concept all the way through to deployment.

About the Technical Reviewer

■ **DON REAMEY** is a software development engineer at Microsoft where he works on Microsoft InfoPath Forms Service, which is a part of Microsoft SharePoint. Don has been with Microsoft for eight years, and he spent ten years developing software for several large financial institutions. Don holds a bachelor of science degree in information systems from Pfeiffer University in Charlotte, NC. Don also uses Pau Reed Smith guitars and Mesa Boogie amplification.

Introduction

In the first edition of this book, I said that .NET is the future. In the second edition, I said that C++/CLI is the future. In this third edition, I say, “Welcome to the future!” .NET has proven itself to be the future of software development, and C++/CLI has shown itself to be the leading language of bridging the gap between the past and the future.

Don’t get me wrong; C# and Visual Basic 2008 are great development languages, but neither has the flexibility or the pedal-to-the-metal power of C++/CLI. And they are not designed to link the code from other realms that needs .NET to make it shine.

With C++/CLI, you can practically mix and match .NET code and ANSI C++ code (or code from many other development languages) at will. Of course, doing so comes at a cost (we’ll get to that later in this book), but the benefits of being able to mix the code without having to rewrite a lot of it is often worth that cost. As a designer, architect, or developer, your task is to determine whether performing this mixing and matching *is* worth it.

But C++/CLI is not just a language for bridging the past and the future. It is the most powerful of the .NET languages for developing new code as well. Any functionality you want coded in .NET can be done in C++/CLI. And this book proves it.

Unfortunately, C++/CLI is frequently overshadowed by his younger sibling C#, who gets the majority of the limelight. Well, this book is designed to refocus the light in the correct direction—toward C++/CLI.

What Is This Book About?

This book is about writing .NET 3.5 applications using C++/CLI. You’ll cover a lot of ground in a short period of time. In the end, you’ll be proficient at developing .NET applications, be they console applications, Windows applications, Windows services, Web applications, or Web services.

While you’re learning the ins and outs of .NET application development, you’ll be learning the syntax of C++, both old and new to .NET 3.5. You will also gain a good understanding of the .NET architecture.

This book does not leave legacy developers out in the cold, as it also shows how to integrate your previously built C++ code and COM, DCOM, COM+, and ActiveX components with your new .NET 3.5 code. Note that this book does not show you how to build any of this legacy code (other than a very simple example code). Instead, it shows you how to code in the world of .NET 3.5 and how to access this legacy code only when it is needed.

Changes in This .NET 3.5 Edition

Microsoft has made several changes to C++/CLI between versions 2.0 and 3.5, fortunately none of them as large as those in the other .NET languages (it’s nice to have a little stability once in a while).

To reflect the changes made between versions, a number of small changes were required throughout this book. In addition to these small changes, there are four significant additions:

- A lengthy section in Chapter 7 on collections covering the STL/CLR
- Chapter 9 on programming with configuration files
- Chapter 16 on Web application development
- Chapter 24 on the C++ Support Library

Who Should Read This Book?

If you're new to the Visual C++ language, this book is for you. The software world is changing, and learning a new language is hard enough without getting unnecessarily bogged down with a complex set of old technologies before you learn about the new ones.

If you're an experienced Visual C++ programmer, this book is also for you. Microsoft is changing your world, and this book will show you these changes. You'll find many books on the market that try to teach you how to force your old world into this new one. This book isn't one of those. Instead, you'll learn the right way to develop .NET code, as the only focus here is the new world: .NET development.

This book is for Visual C++ programmers who don't care about COM, DCOM, COM+, or ActiveX components, either because they already know them or because they never had any reason to learn to code them. You'll use a pure .NET development environment. The only time you'll use components is when you access them—a necessary evil, as there are thousands of them out there that may never be converted to .NET.

This book is also for the (gasp!) non-Microsoft C++ developer who wants to dive into the .NET world without getting bogged down with all the things that he or she disliked about pre-.NET Windows development.

What Does This Book Cover?

This book addresses the topic of C++/CLI in three parts.

The first four chapters cover the basics and background information that make up the C++/CLI and .NET worlds. I recommend that you read these chapters first, as they provide information that you'll need to understand the remainder of this book. I also recommend that you read these chapters in sequential order, because they build on one another.

The main body of the book is the next 17 chapters, which stand alone and cover specific topics. Here, you can pick and choose the chapters that interest you the most (hopefully every chapter) and read them in any order.

The final three chapters cover unsafe code and how to integrate it with C++/CLI. Like with the first four chapters, I recommend you read them in order, as they build on each other.

Chapter 1: Overview of the .NET Framework

In this chapter, you address the basics of the .NET architecture. You're bombarded with many new .NET terms such as assemblies, common language runtime (CLR), common language specification (CLS), common type system (CTS), just-in-time (JIT) compilation, Microsoft Intermediate Language (MSIL or IL), and manifests. This chapter tries to soften the blow of your first foray into the .NET world.

Chapter 2: C++/CLI Basics

This chapter should be a refresher course on the basics of C++. Be careful when you read it though, because there have been several changes related to C++/CLI, and some of them are subtle. This chapter covers the core syntax of C++/CLI. Old-time C++ programmers should pay attention to the reference handle.

Chapter 3: Object-Oriented C++/CLI

Now, with the basics covered, you delve into object-oriented programming (OOP). This chapter covers topics that old-time C++ programmers will take for granted, such as inheritance, encapsulation, polymorphism, classes, methods, and operator overloading. But be careful with this chapter too, as .NET makes some significant changes—in particular, properties, constructors, and two different destructors.

Chapter 4: Advanced C++/CLI

In this chapter, I start to discuss things that should make even seasoned C++ programmers sit up and take notice, because most of the topics I cover are new to C++. This chapter's topics include multifile programming, exception handling, and delegates.

Chapter 5: The .NET Framework Class Library

In this chapter, you start to work with .NET as you make your first strides into the .NET Framework class library. This chapter is just an overview and takes a cursory look at many of the framework's base classes. I focus on helping you learn how to find the classes that you need. In later chapters, I go into some of these base classes in much more detail.

Chapter 6: Integrated XML Documentation

In this chapter, you will learn how to add, generate, and finally view XML documentation that you will imbed in your C++/CLI code. This much-needed and welcome feature was added to C++/CLI in version 2.0 and closely maps to the documentation that has been available to the C# developer since the release of .NET.

Chapter 7: Collections

Working with collections should be nearly second nature to the average software developer. Because collections are so commonplace, most programmers expect powerful and feature-rich ways of handling them, and .NET doesn't disappoint. This chapter covers the four primary sets of collections available to the .NET Framework programmer, including the new addition to .NET 3.5 of STL/CLR.

Chapter 8: Input, Output, and Serialization

Many programs that you'll write in your career will involve moving, copying, deleting, renaming, reading, and/or writing files. More recently, with object-oriented programming, many of a file's I/O activities in a program involve serialization. With this in mind, you'll explore the `System::IO` and `System::Runtime::Serialization` namespaces.

Chapter 9: .NET Configuration File Programming

Since “one size fits all” does not always apply to software development, Microsoft added the configuration file as a dynamic common method of configuring .NET applications. You will cover how to read, modify, and create your own configuration file sections. A neat feature that I threw into this chapter is how to encrypt (and decrypt) sections in your configuration files.

Chapter 10: Basic Windows Forms Applications

Almost all Windows developers, at some time in their careers, will create a Windows application. This chapter shows you how to do it .NET style. You’ll explore how Visual Studio .NET simplifies your development experience. You’ll also cover the basic controls found in the `System::Windows::Forms` namespace in some detail.

Chapter 11: Advanced Windows Forms Applications

Having a handle on the basics is all well and good, but I’m sure that you, as a .NET developer, will want to add more elaborate controls to your Windows applications. This chapter takes what you learned in Chapter 10 and expands on it by exploring some of the more advanced controls available to you in the `System::Windows::Forms` namespace.

Chapter 12: Graphics Using GDI+

If you’re like me, you like a little pizzazz in the form of graphics to spice up a boring Windows application. This chapter shows you how .NET has made adding images and graphics a whole lot easier with the `System::Drawing` namespace.

Chapter 13: ADO.NET and Database Development

What is software development without databases? In most cases, the answer is “not much.” Microsoft is well aware of this and has gone to great lengths to make database programming easier. The solution is ADO.NET. In this chapter, you’ll explore the many features of ADO.NET that you can find in the `System::Data` namespace.

Chapter 14: XML

XML is the new world order when it comes to data storage, and Microsoft has embraced XML in a big way. This chapter shows the many ways that you can now access XML data in the .NET environment.

Chapter 15: Windows Services

The C++ language has long been a stronghold for Windows services development. This will not change with C++/CLI. In fact, I predict that some of the defection to C# in this area may return because of the power of C++/CLI. In this chapter, you will see just how easy it is to create Windows services using C++/CLI.

Chapter 16: Web Applications

I added this chapter back after removing it from the previous edition, because it turns out that you can still create Web applications using C++/CLI, and this chapter proves it (okay, I had to create my own Visual C++ template to do it, but hey, the Web applications work). This chapter briefly covers HTML and ASP.NET. You then learn about the `System::Web` namespace in some detail.

Chapter 17: Web Services

The concept of Web services is not unique. In this chapter, you'll explore Web services within the .NET Framework. You'll examine how to design and create them by walking through the process yourself, creating a simple Web service and three different clients (console, Windows application, and Web application) to interact with the service.

Chapter 18: Multithreaded Programming

Being able to run multiple threads at the same time allows for better CPU usage and is a powerful feature. This chapter explores how the .NET Framework makes working concurrently with multiple threads a snap as you cover the .NET Framework's built-in multithreading capabilities.

Chapter 19: Network Programming

In this chapter, you'll examine the different methods of moving data over a network using .NET. Specifically, the chapter will examine socket coding in C++/CLI for both TCP and UDP in synchronous and asynchronous approaches.

Chapter 20: Assembly Programming

In traditional C++, application and library developers had few choices regarding what went into .exe and .dll files. With .NET assemblies, this limitation has changed, and you now have plenty of choices. This chapter explores those choices by looking at how you can augment your assemblies with resources, localization, attributes, and reflection.

Chapter 21: Security

.NET is touted as being an extremely secure software environment, and this is evident in the plethora of .NET Framework security features. In this chapter, we will look at how you can access many of them using C++/CLI.

Chapter 22: Unsafe C++ .NET Programming

This chapter takes a look at what is involved in mixing and matching unsafe C++, also known as unmanaged C++ or traditional C++, with C++/CLI. This chapter is designed to fill in the code areas not normally associated with C++/CLI. In fact, compiling any code from this chapter would require a special compiler option.

Chapter 23: Advanced Unsafe or Unmanaged C++ .NET Programming

Unlike other books that cover this topic, this book looks at advanced unsafe C++ from the eyes of someone who is coding in C++/CLI and wants to integrate some unsafe or unmanaged code into existing code. Usually, the approach is the opposite (i.e., a developer who is coding unsafe or unmanaged code is trying to force it into the C++/CLI environment). This chapter will regard the unsafe/unmanaged code as a black box that you will attach to your C++/CLI code in different fashions, depending on the type of unsafe/unmanaged code to which you are connecting.

Chapter 24: The C++ Support Library

Mixing managed and unmanaged code is quite often complex. Microsoft, trying to ease the process, has provided a library full of classes, functions, and templates to alleviate the complexity. This chapter walks you through the functionality provided by the C++ Support Library.

What You Need to Use This Book

The first thing you should probably do is download the code for this book from the Source Code section of the Apress Web site (<http://www.apress.com>) or from my Web site (<http://www.procppcli.net>). Most of the code in this book is listed in its entirety, but some of the larger programs (in particular, the Windows Forms applications) list only relevant code.

In addition to the source code, you should have a copy of Visual Studio 2008 in any of its flavors. Note that most, but not all, of the features mentioned in this book work with the free Visual C++ Express 2008 version.

As long as you have the .NET Framework version 3.5 and its associated C++ compiler, however, you should be able to build nearly everything in the book (though, in several areas, with a lot more effort if you don't have Visual Studio or Visual C++ Express 2008).

Caution This book contains material that isn't supported in Visual Studio .NET 2003 and the .NET Framework 1.1 or earlier.

This Book Is Not the End of the Story

A book is a pretty static thing, and once you finish reading it, you have to go elsewhere for more information. Fortunately, I have built a Web site devoted entirely to C++/CLI: <http://www.procppcli.net>.

On this site, you will not only find all the source code for this book but also further writings on C++/CLI by me and other authors. The Web site's goal is to promote further exploration of C++/CLI, thus the site will also contain news, a discussion area, an area to upload your code, and an area to download third-party code.

How to Reach Me

I would like to hear from you. Feel free to e-mail me at srgfraser@procppcli.net. If you have a question and you think others would benefit from the answer, ask it on the <http://www.procppcli.net> discussion board. I will respond to every e-mail and discussion entry that I can. Questions, comments, and suggestions are all welcome.

Oh, by the way, thank you for buying my book. Now, let's get started!

PART 1



The C++/CLI Language



Overview of the .NET Framework

First off, let's get one thing straight. This book is about developing code within the confines of the Microsoft .NET Framework 3.5. Therefore, it only makes sense that you start by getting acquainted with the underlying architecture with which you will be developing your code: the .NET Framework.

I cover a lot of material in this chapter, mostly at the 30,000-foot level. The main goal here isn't to make you a .NET expert. This chapter is designed to provide you with a level playing field from which to start your C++/CLI code development while exploring this book.

I start with a brief description of .NET and the .NET Framework and why we programmers need it. Then, I briefly examine the assembly, which is the central building block for all .NET Framework application distribution and execution. Next, I move on to the core of the .NET Framework: the common language runtime (CLR), the common type system (CTS), and the common language specification (CLS). Finally, I discuss, at a very high level, the software components available to .NET Framework developers.

What Is .NET?

I guess getting the definition from the horse's mouth would be a good place to start. Microsoft describes .NET on their Web site (<http://www.microsoft.com/net/Overview.aspx>) in the following way:

The .NET Framework is a development and execution environment that allows different programming languages and libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.

Built on Web service standards, .NET enables both new and existing personal and business applications to connect with software and services across platforms, applications, and programming languages. These connections give users access to key information, whenever and wherever you need it.

Microsoft .NET-connected software makes the "real-time" enterprise real by enabling information to flow freely throughout the organization, accessible to business partners, and delivering value to customers. With .NET-connected software, users can increase the value of existing systems and seamlessly extend those systems to partners, suppliers, and customers.

Quite a mouthful, don't you think? So what does it mean?

The first thing many developers mistakenly assume is that .NET is strictly a network or Web architecture. You would think so with Microsoft's definition. Heck, even the name ".NET" suggests it. Well, truthfully, .NET sort of is and sort of isn't.

Within .NET are many features that enable a developer to create some truly awesome stand-alone applications—and very easily, I might add. But, according to Microsoft, as their definition suggests, developing stand-alone applications is not the goal of .NET.

That being said, what is .NET? Well, my definition is a little less verbose:

.NET is a set of technologies that allow entire software applications to be created rapidly and easily using an integrated network-centric architecture.

I have to admit that Microsoft's definition does sound much more impressive. But when you boil down Microsoft's marketing fluff, this is really all they are saying.

The key concept Microsoft is trying to push with .NET is interconnectivity between computer systems. True, interconnectivity is hardly new. A host of technologies, such as DCOM, COM+, and CORBA, have been doing this for quite a long time. What make .NET special is how nearly effortless it is to develop this interconnectivity within your applications.

When architecting, designing, and developing using .NET, you are not restricted to your single workstation, LAN, or even your company's WAN. With .NET, your application can use the entire Internet. In fact, not all the parts of your system have to be owned or maintained by your company. What this means is you can have part of your application running in your data center in India, another part in China owned by a third party, which prints out to a client in Russia, and it's all driven from a workstation in the United States. (Okay, lag might be an issue with all these distance places, but that is a hardware issue so it's not my concern... I'm joking... really.)

What is really cool is that .NET uses a technology called the Web service, which is based on XML and allows .NET to interconnect with systems on architectures not based on .NET. Thus, not only can your application be dispersed all over the globe, but the applications it can interconnect with can be Unix, Linux, Mac OS, or any other operating system that supports XML (off the top of my head, I can't think of any).

You might be asking why is this book so large then, if .NET is all about network interconnectivity? This is where the other key concept of my definition comes into play: "entire." True, you are developing network-centric applications, but you are also creating all parts of the application. This means with .NET you can create the presentation tier, business tier, database tier, and anything in between—and in fact you frequently do. To accomplish this, .NET provides a huge framework from which to do your development called the .NET Framework.

Note Wherever you read the word "Internet," you can assume "intranet" and "extranet" apply as well.

What Is the .NET Framework?

The .NET Framework comprises all the pieces needed to develop, deploy, and execute Web services, Web applications, Windows services, Windows applications, and console applications. (Well, almost all the pieces. IIS is needed for Web services and Web applications.) I discuss each of these in more detail later in the chapter. You can think of the .NET Framework as a three-level hierarchy consisting of the following:

- Application development technologies like ASP.NET, Windows Forms, ADO.NET, Windows Presentation Foundation, Windows Communication Foundation, Windows Workflow Foundation, Windows CardSpace, and LINQ.
- .NET Framework base class library
- CLR