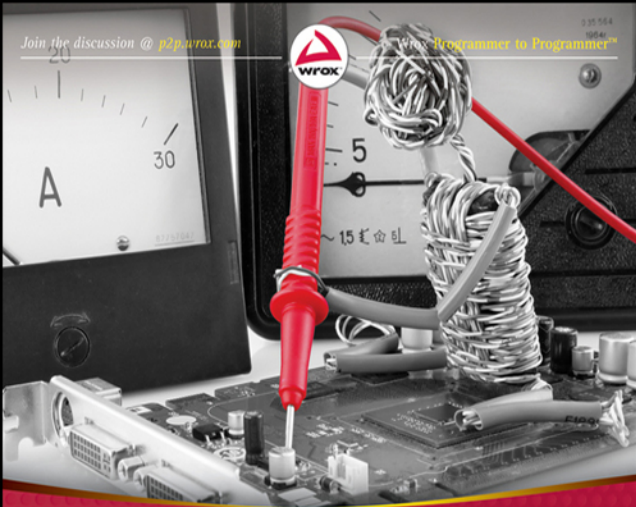


Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Professional Windows® Embedded Compact 7

Foreword by Mike Hall, *Principal Software Architect at Microsoft*

Samuel Phung, David Jones, Thierry Joubert

PROFESSIONAL WINDOWS® EMBEDDED COMPACT 7

FOREWORD	xxxv
INTRODUCTION.....	xxxvii
 ► PART I INTRODUCING EMBEDDED DEVELOPMENT	
CHAPTER 1 Embedded Development	3
CHAPTER 2 Windows Embedded Compact 7.....	11
CHAPTER 3 Development Station Preparation.....	19
CHAPTER 4 Development Process.....	29
CHAPTER 5 Development Environment and Tools	35
 ► PART II PLATFORM BUILDER AND OS DESIGN	
CHAPTER 6 BSP Introduction	53
CHAPTER 7 OS Design	69
CHAPTER 8 Target Device Connectivity and Download	87
CHAPTER 9 Debug and Remote Tools.....	97
CHAPTER 10 The Registry	125
CHAPTER 11 The Build System	137
CHAPTER 12 Remote Display Application.....	151
CHAPTER 13 Testing with Compact Test Kit.....	157
 ► PART III APPLICATION DEVELOPMENT	
CHAPTER 14 Application Development.....	173
CHAPTER 15 .NET Compact Framework.....	181
CHAPTER 16 CoreCon Connectivity.....	187
CHAPTER 17 Visual Studio Native Code Application Example	195
CHAPTER 18 Managed Code Application Example.....	207
CHAPTER 19 Platform Builder Native Code Application Example	219

Continues

CHAPTER 20	Developing Embedded Database Applications	233
CHAPTER 21	Silverlight for Windows Embedded	267
CHAPTER 22	Silverlight for Windows Embedded Application Examples	275
CHAPTER 23	Auto Launching Applications	289
CHAPTER 24	Application Deployment Options	297

► **PART IV DEPLOY WINDOWS EMBEDDED COMPACT 7 DEVICES**

CHAPTER 25	Deploy OS Run-time Images	305
CHAPTER 26	Bootloaders	313
CHAPTER 27	BIOSLoader	321
CHAPTER 28	The DiskPrep Power Toy	327

► **PART V DEVICE DRIVERS, BOOT LOADER, BSP, AND OAL DEVELOPMENT**

CHAPTER 29	An Overview of Device Drivers	339
CHAPTER 30	Device Driver Architectures	349
CHAPTER 31	Interrupts	367
CHAPTER 32	Stream Interface Drivers	393
CHAPTER 33	Developing a Stream Interface Driver	417
CHAPTER 34	Stream Driver API and Device Driver Testing	449
CHAPTER 35	The Target System	467

► **PART VI ADVANCED APPLICATION DEVELOPMENT**

CHAPTER 36	Introduction to Real-Time Applications	487
CHAPTER 37	A Simple Real-Time Application	497
CHAPTER 38	Extending Low-Level Access to Managed Code	509
CHAPTER 39	Extending Low-level Access to Managed Code with Messages	531
CHAPTER 40	A Web Server Application	545
CHAPTER 41	A USB Camera Application	563

► **PART VII SAMPLE PROJECTS**

CHAPTER 42 Develop a Windows Network Projector..... 577

CHAPTER 43 Phidgets Devices585

CHAPTER 44 FTDI Devices..... 601

CHAPTER 45 Integrating Managed Code Projects629

APPENDIX A Virtual PC Connectivity.....639

APPENDIX B Microsoft Resources645

APPENDIX C Community Resources649

APPENDIX D Embedded Hardware655

INDEX.....659

PROFESSIONAL

Windows[®] Embedded Compact 7

PROFESSIONAL

Windows[®] Embedded Compact 7

Samuel Phung
David Jones
Thierry Joubert



John Wiley & Sons, Inc.

Professional Windows® Embedded Compact 7

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2011 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-05046-0
ISBN: 978-1-118-16750-2
ISBN: 978-1-118-16748-9
ISBN: 978-1-118-16747-2

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Not all content that is available in standard print versions of this book may appear or be packaged in all book formats. If you have purchased a version of this book that did not include media that is referenced by or accompanies a standard print version, you may request this media by visiting <http://booksupport.wiley.com>. For more information about Wiley products, visit us at www.wiley.com.

Library of Congress Control Number: 2011934627

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

ABOUT THE AUTHORS



SAMUEL PHUNG has worked in the technology field for more than 20 years. In the early 1990s, he led a financial database software development team, developing software for the banking industry. Later he led a software team developing Windows-Based telephony applications for a venture capital-funded startup. He started to work in the embedded computing field in the late 1990s and engaged with the Windows Embedded product team, starting with Windows NT 4.0 Embedded. He has been working with Windows Embedded Compact since version 2.12 was introduced.

As the VP of sales and marketing for ICOP Technology, a hardware manufacturer headquartered in Taiwan with a branch office in the United States and a manufacturing facility in China, Samuel is responsible for strategic business development for ICOP in the North America region. In 2003, he created the Vortex86 branding and started an initiative focused on developing business around Windows Embedded technology for ICOP.

In 2009, he wrote *Professional Microsoft Windows Embedded CE 6.0*.

Samuel enjoys working with technology, actively engages with the academic community, and received the Windows Embedded MVP recognition from Microsoft since 2005. As part of his involvement in the academic community, Samuel actively works with university teaching professionals in the United States, China, and Taiwan and other regions to adopt Windows Embedded technology as part of their teaching curriculum.

As part of his Windows Embedded community activities, Samuel maintains a personal website: www.embeddedpc.net, to provide information resources related to Windows Embedded. In 2010, he initiated the Embedded101 Windows Embedded community portal, www.embedded101.com.



DAVID JONES has a Master of Engineering degree from RMIT University and BSc(Hon) from Melbourne University. David has been actively engaged in Embedded Systems and Computing Technologies for more than twenty years. From 1990 to 2006, he was a University Lecturer in Computer Engineering at RMIT University in Melbourne, Australia. After leaving his university teaching role in late 2006, he joined the Victorian Partnership for Advanced Computing (www.vpac.org) to provide embedded system training, consulting, and development services with a focus on modern Embedded-system technologies. VPAC is a non-profit research agency established in 2000 by a consortium of Victorian Universities to provide advanced computing expertise, training, and support to academia, industry, and government.

While teaching at RMIT, he mentored student teams participating in the Windows Embedded Student Challenge competition sponsored by Microsoft. One of the student teams he mentored won first place during the 2005 worldwide final competition. David actively engages in the Windows Embedded community. He has delivered presentations on behalf of Microsoft in the Asia Pacific

region, covering Windows Embedded and .NET technologies. In 2010, he initiated the effort to develop a Device-Driver Wizard and a Component Wizard, both for Windows Embedded CE 6.0 and Windows Embedded Compact 7. He released community versions for both. David is a certified Windows Embedded trainer.



THIERRY JOUBERT is the CTO and co-founder for THEORIS, a technology company in France that provides project management, software consulting, outsourcing, and training services with focus on modern embedded technology. He graduated from the Ecole Centrale de Nantes in France with an engineering degree in computer science. Thierry has been actively engaged in Embedded-system design and real-time application development for over 25 years.

In addition to his responsibility working on commercial projects, Thierry is actively involved in the academic community, delivering Windows Embedded trainings and technical seminars for engineering schools and universities. In 2004, Thierry developed a case study on Windows CE for Microsoft's MSDN Academic Alliance curriculum, and published multiple technical papers to help teach Windows Embedded technology on the Microsoft Faculty Resource site. To recognize Thierry's effort and contribution to the Windows Embedded developer community, Microsoft has awarded the Windows Embedded MVP status to Thierry since 2007.

ABOUT THE TECHNICAL EDITOR



DOUG LOYD first learned to write code on his parents' Commodore 64, drawing inspiration from the pages of BYTE magazine. He earned his degree in Computer and Information Sciences from the University of Delaware and has spent the last 10 years working on Windows CE devices. He lives in rural Maryland with his wife and daughter. You can contact Doug at douglas.loyd@gmail.com.

CREDITS

ACQUISITIONS EDITOR

Paul Reese

PROJECT EDITOR

Ed Connor

TECHNICAL EDITOR

Doug Loyd

PRODUCTION EDITOR

Daniel Scribner

COPY EDITOR

San Dee Phillips

EDITORIAL MANAGER

Mary Beth Wakefield

FREELANCER EDITORIAL MANAGER

Rosemarie Graham

ASSOCIATE DIRECTOR OF MARKETING

David Mayhew

MARKETING MANAGER

Ashley Zurcher

BUSINESS MANAGER

Amy Knies

PRODUCTION MANAGER

Tim Tate

VICE PRESIDENT AND EXECUTIVE GROUP**PUBLISHER**

Richard Swadley

VICE PRESIDENT AND EXECUTIVE PUBLISHER

Neil Edde

ASSOCIATE PUBLISHER

Jim Minatel

PROJECT COORDINATOR, COVER

Katie Crocker

PROOFREADER

Jen Larsen, Word One

INDEXER

Robert Swanson

COVER DESIGNER

LeAndra Young

COVER IMAGE

©Aleksandr Volkov/iStockPhoto

ACKNOWLEDGMENTS

FIRST, I WANT TO RECOGNIZE the Windows Embedded Compact development team's effort. Without their hard work, the Windows Embedded Compact product would not be where it is today.

As I went through the process to learn Windows Embedded Compact, I found many information resources on the news group and forum, which helped me learn and resolved problems. I want to thank the developers in the community who helped answer questions on the news group, shared their knowledge, and posted valuable application notes online to help others.

Thanks to David and Thierry for participating in this book project and helping to expand the contents. Throughout the book project, I gained valuable knowledge from David and Thierry.

I want to recognize the following individuals for their helpfulness:

- Michael Fosmire with the MVP team. He is always accommodating and willing to listen. I want to thank Michael for the resources he provided to help the Windows Embedded community.
- Olivier Bloch with the Windows Embedded team. I could count on Olivier to be responsive and help provide answers to the questions we had throughout the book project.
- D'Arcy Salzmänn with the Windows Embedded team. D'Arcy initiated the dialog about the book project and motivated us to move forward with this book.
- James Y. Wilson, one of the authors of the *Building Powerful Platforms with Windows CE (version 3.0)* book. James helped me to get over the initial hurdle to learn and engage in Windows Embedded Compact development. For the more than 7 years that I have known James, he has provided valuable resources and contributions to both the professional and academic developer communities.

As an amateur writer, with English as my second language, writing is not an easy task. I want to thank Ed Connor and San Dee Phillips, editors for the book project, for reviewing my writing, correcting many mistakes that I made, and providing valuable input.

Most of all, I want to thank my wife, Ann, and my children, Aaron, Narissa and Nathan for their understanding and patience while I took time away from the family to work on the book.

—SAM PHUNG

I WOULD LIKE TO ACKNOWLEDGE the support and assistance I have previously received from Microsoft staff in past, particularly when I was an academic. People in Australia such as Nigel Watson, John Warren, Don Kerr, Tim Schroeder and others have assisted me in many ways. At

Redmond I'd also like to thank Mike Hall, Stewart Tansley, Lindsay Kane, and Sondra Weber. Thanks also to Nelson Lin for your assistance and friendship.

I would like to thank the many students who have worked on Windows Embedded projects with me; especially those who competed in Microsoft Windows Embedded Student Challenges. It has been great to act a facilitator of those projects. I am always amazed at the way students can take nebulous ideas and turn them into something substantial and useful.

The current Windows Embedded team at Microsoft have been timely and constructive with their support during this activity. Thanks to Olivier Bloch, D'Arcy Salzmänn, and others. They have been busy with the release of Compact 7 but found the time to support us.

I would also like to thank all of those Embedded MVPs and others who have contributed to my understanding of Windows Embedded through books, presentations, newsgroups, forums, and blogs. There are many of you. ("Standing on the shoulders of giants" — Isaac Newton.)

Thanks Sam and Thierry for input to this activity as co-authors. I have known Sam for a number of years through the Windows Embedded forum. His contribution to Windows Embedded through such things as student and Embedded Spark competitions is invaluable. Sam has been at the helm of this project and without his effort it would not have come to fruition. Thierry has been a great help on the technical side. He has much experience with commercial development with Windows Embedded. His feedback has been precise and constructive.

Thanks also to Ed Connor and San Dee Phillips for their reviews and feedback of my chapters. As a first-time author this has been a big learning curve for me. Their assistance is greatly appreciated.

I'd like to finish with a big thank you to my wife Wendy who has had to put up with my long hours working at this project. Thanks Wendy.

—DAVID JONES

I STARTED WORKING WITH Windows CE 3.0 when Microsoft released it in 2000 and the product has come a long way since then to reach Windows Embedded Compact 7. All these years the Windows Embedded development and marketing teams have made sustained efforts to improve their products. I thank Lorraine Bardeen, Myriam Semery, Sondra Webber, Kevin Dallas, Olivier Bloch, Mike Hall, and D'Arcy Salzmänn for their availability and openness when we make suggestions.

A special thanks to Samuel, who invited me as a co-author of this book, and to David who contributed to make the writing task enjoyable. I also thank my colleague Vincent Cruz, who gave me the image transformation code in Chapter 41, and our reviewers Ed Connor and San Dee Phillips.

Most of all I want to thank my family for their patience during this long period where I rarely left my desk.

—THIERRY JOUBERT

CONTENTS

FOREWORD

xxxv

INTRODUCTION

xxxvii

PART I: INTRODUCING EMBEDDED DEVELOPMENT

CHAPTER 1: EMBEDDED DEVELOPMENT 3

What Is an Embedded Device? 3

Similarity to Personal Computer 4

Difference from Personal Computer 4

Specialized Purpose Device 5

Example of Embedded Devices 5

What Is Embedded Software? 6

Programming Languages and Principles 6

Programming Discipline 6

Specialized Purpose Application 7

Development Considerations 7

Hardware 8

Operating Environment 9

User Environment 9

Feasibility 9

Summary 10

CHAPTER 2: WINDOWS EMBEDDED COMPACT 7 11

What Is Windows Embedded Compact? 11

Windows Embedded Compact 7 Features 12

Modular and Compact Operating System 14

Real-Time Operating System 14

Support Broad Range of Hardware 15

History 15

Targeted Market 16

Why Windows Embedded Compact? 17

Developer-Friendly Tools 17

Debug, Testing, and Quality Control 18

Summary 18

CHAPTER 3: DEVELOPMENT STATION PREPARATION	19
Development Computer Requirements	19
Hardware	20
Software	20
Windows Embedded Compact 7 Software	21
Recommended Installation Steps	21
Quick Fix Engineering Updates	23
Development Environment Setup	23
Target Device	23
Virtual PC as Target Device	24
LAN with DHCP	24
LAN without DHCP	24
Connectivity for Debugging and Testing	25
Ethernet	25
Serial	26
Typical Development Environment	26
Summary	27
CHAPTER 4: DEVELOPMENT PROCESS	29
Planning	29
Hardware Selection	30
Software Selection	31
Typical Development Processes	31
Board Support Package (BSP)	32
OS Design	33
Application Development	33
Debugging and Testing	33
Deployment	34
Post-Deployment Support and Updates	34
Summary	34
CHAPTER 5: DEVELOPMENT ENVIRONMENT AND TOOLS	35
Development Environment	35
Compact 7 Terminology	36
Platform Builder for Windows Embedded Compact 7	37
What's New in Compact 7	39
Environment Variables	40
Board Support Package (BSP)	43
BSP Cloning Wizard	43
OS Design Wizard	44

OS Design Templates	44
Catalog Items	45
SDK Wizard	46
Remote Tools	47
Windows Embedded Silverlight Tool	47
Target Device Connectivity	48
Kernel Independent Transport Layer (KITL)	48
Core Connectivity	48
Application for Compact 7	49
Windows Embedded Compact Test Kit	49
Summary	49

PART II: PLATFORM BUILDER AND OS DESIGN

CHAPTER 6: BSP INTRODUCTION 53

BSP Provided by Platform Builder	54
BSP Components, Files, and Folders	54
Clone an Existing BSP	56
Customize the Cloned BSP	57
Add an ATAPI Driver Component to MyBSP	58
Add a Hive-Based Registry Component to MyBSP	60
Add a Display Configuration Component to MyBSP	61
Add Files to the BSP	63
Add a Component to Configure System Memory	65
Add Serial Port Driver Components	66
Add a Component to Enable Serial Debug Messages	67
Summary	68

CHAPTER 7: OS DESIGN 69

What Is an OS Design?	69
Develop an OS Design	70
OS Design Wizard	70
OS Design Project Files and Folders	74
Catalog Item View	74
Customize the OS Design	75
Compile and Generate OS Run-time Image	82
Generate SDK from the OS Design	83
Create a New SDK	83
Build and Generate an SDK MSI File	85
Summary	86

CHAPTER 8: TARGET DEVICE CONNECTIVITY AND DOWNLOAD	87
Target Device Connectivity	88
Connecting to the Target Device	88
Establish Connection to Download Compact 7 Image to Target Device	88
Download OS Run-time Image to Target Device	92
Target Device Connectivity Setting	92
Summary	95
CHAPTER 9: DEBUG AND REMOTE TOOLS	97
Debugging Environment	97
Kernel Independent Transport Layer (KITL)	98
Core Connectivity (CoreCon)	98
Serial Debug	99
Debugging the OS Design	100
Build Error	101
Remote Tools	103
Remote File Viewer	105
Remote Heap Walker	106
Remote Process Viewer	107
Remote Profiler	108
Remote Registry Editor	111
Remote Resource Consumer	112
Remote Resource Leak Detector	113
Remote System Information	116
Remote Zoom-In	117
Remote Timeline Viewer	118
Target Control	121
Display Target Device Running Processes	121
Launch Internet Explorer with Target Control	121
Terminate Running Process with Target Control	122
Other Target Control Commands	122
Summary	123
CHAPTER 10: THE REGISTRY	125
Windows Embedded Compact Registry	125
RAM-Based Registry	126
Hive-Based Registry	126
Registry for Windows Embedded Compact Component	129
Registry for the Serial Port	129
Useful Registry References	131

Registry for the FTP Server	131
Registry for the Telnet Server	132
Device Identification and Description	132
Registry to Launch Application during Startup	132
Device Name for USB Removable Storage	133
Internet Explorer Startup Page	133
Auto Flush and Save Registry	133
Disable Suspend Function on the Start Menu	133
Static IP Address	134
Windows Embedded Compact Registry Files	134
COMMON.REG	135
PLATFORM.REG	135
Catalog Item Registry	135
Subproject Registry	135
OSDESIGN.REG	135
REGINIT.INI	135
Accessing the Registry	136
Summary	136
CHAPTER 11: THE BUILD SYSTEM	137
The OS Design Build Process	137
Pre-Sysgen Phase — Build OS	139
Sysgen Phase	139
Build Phase	140
Build Release Phase	142
Make Image Phase	142
Build System Tools	142
Build from the Platform Builder IDE	143
Build from the Command Line	145
Best Practice to Save Time and Minimize Problems	146
Understand the Build Process	146
Project Documentation and Archive	148
Summary	149
CHAPTER 12: REMOTE DISPLAY APPLICATION	151
Access Compact 7 Desktop Remotely	151
Add Remote Display Application to an OS Design	151
Add Remote Display Application from the Catalog	152
Add Registry to Launch Remote Display Application	153
Generate OS Run-time Image with Remote Display Application	153
How-To: Use Remote Display Application	154

Using Remote Display Application on Headless Device	155
Summary	156
CHAPTER 13: TESTING WITH COMPACT TEST KIT	157
Compact Test Kit	157
Establishing Connectivity for CTK	158
Preparing an OS Run-time Image to Support CTK	158
Connecting CTK to the Target Device with KITL	158
Connecting CTK to a Target Device with CoreCon	160
Testing Compact 7 Device with CTK	162
CTK Test with Manual Interaction: Input Device	163
Unattended CTK Test: Display Driver	166
BSP Test	167
CTK Test Pass	168
Summary	170
PART III: APPLICATION DEVELOPMENT	
CHAPTER 14: APPLICATION DEVELOPMENT	173
Developing Compact 7 Applications	173
Differences When Developing Desktop Applications	174
Real-Time Application	175
Managed-Code Applications Using Visual Studio	176
Native Code Application for Compact 7	177
Connectivity to Deploy and Debug Application	179
CoreCon	179
Kernel Independent Transport Layer (KITL)	179
Summary	180
CHAPTER 15: .NET COMPACT FRAMEWORK	181
.NET Compact Framework Application	182
Required Compact 7 OS Components	182
Connectivity to Target Device	182
Steps to Develop, Deploy, Test, and Debug	182
.NET CF Application Considerations	183
Similarity to Windows Phone 7	183
.NET Compact Framework Garbage Collector	184
Platform Invoke (P/Invoke)	184
Separate Thread for Event Handler and GUI	184
Differences from the Full .NET Framework	184
Summary	185

CHAPTER 16: CORECON CONNECTIVITY	187
Implementing CoreCon for Application Development	187
Required CoreCon Files	188
Copy CoreCon Files to Compact 7 Device's File System	188
Edit OSDesign.BIB to Include CoreCon Files in the OS Image	189
Using a Third-Party CoreCon Component	190
Connecting to a Target Device with CoreCon	192
Summary	194
CHAPTER 17: VISUAL STUDIO NATIVE CODE APPLICATION EXAMPLE	195
Prerequisites and Preparation	195
Develop a Native Code Application for Compact 7	196
Creating a Win32 Smart Device Application Project	196
Add Code to a Win32 Smart Device Application	197
Connecting to a Target Device	198
Deploy an Application to a Target Device for Testing	200
Debugging a Win32 Smart Device Application	201
Summary	205
CHAPTER 18: MANAGED CODE APPLICATION EXAMPLE	207
Prerequisites and Preparation	207
Developing a Managed Code Application for Compact 7	208
Creating a Managed Code Smart Device Application Project	209
Add Code to a Managed Code Smart Device Application	210
Connecting to a Target Device	212
Deploying a Managed Code Smart Device Application to the Target Device	213
Debugging a Managed Code Smart Device Application	215
Summary	218
CHAPTER 19: PLATFORM BUILDER NATIVE CODE APPLICATION EXAMPLE	219
Prerequisites and Preparation	219
Developing a Virtual PC OS Design	220
Using a Virtual PC as the Target Device	221
Establish a Connection to a Virtual Machine	221
Create a Target Device Profile for a VM	222
Configure the Virtual Machine	222
Download an OS Run-time Image to a Virtual Machine	223

Developing a Platform Builder Native Code Application for Compact 7	224
Creating a Platform Builder Application Subproject	224
Including an Application in the OS Run-time Image	226
Downloading an OS Run-time Image to a Virtual PC	226
Debugging a Platform Builder Native Code Application	226
Building and Launching a Platform Builder Subproject Application	228
Rebuild and Relaunch a Platform Builder Subproject Application	229
Debug Messages	230
Summary	231
 CHAPTER 20: DEVELOPING EMBEDDED DATABASE APPLICATIONS	 233
Introducing Microsoft SQL Server Compact	233
Some Data to Store	234
Microsoft SQL Server Compact	234
What Is a Database?	235
Where Is the Database?	236
Windows Embedded Compact Database Engines	236
SQL Server Compact 3.5	236
Compact Database Requirements	238
Compact 7 Device Requirements	238
Managed Code Requirements	239
SqlCe Classes	240
Building a SQL Compact Database Application Using Visual Data Designers	241
Getting Started	241
Creating a New SQL Compact Server Database	242
Creating a Table in the Database	242
Populating the Table	243
Creating the Data Source	243
Displaying a SQL Compact Table in a DataGrid	243
Editing Data	244
Inserting, Updating, and Deleting Queries	244
Adding an Edit and a View Form	245
Enabling Record Deletes	246
A Media Playlist List Application	247
Playlist Functions	249
Text File Data and XML Serialization	250
Reading and Writing to a Text File	250
XML Serialization	252
Building the Managed Code Data Application (Text and XML)	254

Building a Managed Code Remote Database Application	257
Preparation	257
Writing Fracks to a Remote SQL Server	257
Reading Tracks from a Remote SQL Server	259
Building a Managed Code Compact Database Application	261
Writing Tracks to a Compact Database File	261
Reading Tracks from a Compact Database File	263
Summary	264
CHAPTER 21: SILVERLIGHT FOR WINDOWS EMBEDDED	267
Silverlight: User Interface Development Framework	267
Silverlight for Windows Embedded	268
Development Environment and Tools	269
Required Software	269
XAML: Extensible Application Markup Language	270
Code-Behind	272
Contract Between Designer and Application Developer	273
Development Process	273
Summary	274
CHAPTER 22: SILVERLIGHT FOR WINDOWS EMBEDDED APPLICATION EXAMPLES	275
Prerequisites and Preparation	275
Develop a Compact 7 OS Design with Silverlight Support	276
Develop the SWE Application Project Using Expression Blend 3	277
Port a XAML Code Project to Native Code Using Windows Embedded Silverlight Tools	279
Add the SWE Application as a Subproject, Compile, and Launch	281
Add Application as Subproject	281
Build and Compile a Subproject Application	281
Launch the Subproject Application on the Target Device	282
Add Event Handler to Silverlight XAML Code Project	283
Update the SWE Application Subproject	285
Create a User Control	285
Update the SWE Application Subproject to Include Animation	287
Summary	288
CHAPTER 23: AUTO LAUNCHING APPLICATIONS	289
Configuring the Registry to Auto Launch Application	290
Understanding the Compact 7 Startup Registry	290
Configure Registry to Launch Application During Startup	291

Auto Launch Application from Startup Folder	292
Using the AutoLaunch Component	293
AutoLaunch Multiple Applications	294
Summary	295
CHAPTER 24: APPLICATION DEPLOYMENT OPTIONS	297
Deploying a Compact 7 Applications	297
Options	298
Deploying an Application: CAB Installable Package	298
Deploying an Application: Local Storage	299
Deploying an Application: Compiled as Part of the OS Image	300
Deploying an Application: Hybrid Between Compiled as Part of the OS Image and Local Storage	300
Summary	301
PART IV: DEPLOY WINDOWS EMBEDDED COMPACT 7 DEVICES	
CHAPTER 25: DEPLOY OS RUN-TIME IMAGES	305
Considerations	305
Network: PXE	306
Removable Storage: USB Flash Storage	307
Fixed Storage: Hard Drive, IDE Flash, and Compact Flash	307
Bootloader	308
Deploying an OS Run-time Image	308
Image Deployment on RISC Platforms	309
Summary	312
CHAPTER 26: BOOTLOADERS	313
Compact 7 Bootloader	313
Ethernet Bootloader (Eboot)	314
Serial Bootloader (Sboot)	315
Loadcepc	315
BIOSLoader	316
Compact 7 Bootloader Framework	317
Summary	319
CHAPTER 27: BIOSLOADER	321
BIOSLoader Startup Parameters	321
BIOSLoader Files and Utility	322
Using BIOSLoader	323

Boot Sector Image: BSECT.IMG	323
CESYS.EXE Utility	323
BIOSLoader Image: BLDR	324
Preparing a Storage Device with BIOSLoader	324
Startup Splash Screen	324
Windows CE Splash Generator	325
Summary	325
CHAPTER 28: THE DISKPREP POWER TOY	327
Prerequisites and Preparation	328
DiskPrep to Launch EBOOT.BIN	328
DiskPrep to Launch NK.BIN	328
Using DiskPrep Power Toy	328
Configuring Compact Flash with BIOSLoader for eBox-3310A	329
Configuring Virtual Hard Disk with BIOSLoader for Virtual PC	333
Summary	335
PART V: DEVICE DRIVERS, BOOT LOADER, BSP, AND OAL DEVELOPMENT	
CHAPTER 29: AN OVERVIEW OF DEVICE DRIVERS	339
What Is a Device Driver?	339
System Concepts	340
Operating System Structure	343
An Operating System as Interrupts	343
An Operating System as Drivers	343
Windows Embedded Compact Drivers	343
GWES and File System	344
Device and Services	344
Board Support Packages	344
Drivers in the Compact 7 Catalog	345
Custom Drivers	347
Custom Driver Development	347
Use a Third-Party Driver Binary	348
Summary	348
Note	348
CHAPTER 30: DEVICE DRIVER ARCHITECTURES	349
Introducing Device Driver Architectures	349
What Is a Hardware Device Driver?	349
What Then Is a Virtual Device Driver?	350
Driver Stack	351

Driver Loading and Use	352
Kernel and User Driver Modes	353
Native and Stream Drivers	356
Monolithic and Layered Driver Models	358
Monolithic Drivers	358
Layered Drivers	358
Monolithic Versus Layered Drivers	360
Stream, Block, Bus, and USB Drivers	360
How to Check if the Bluetooth Stack Is Loaded	362
Using the Compact 7 Bluetooth Components	365
Summary	366
CHAPTER 31: INTERRUPTS	367
Polling and Interrupts	367
Compact 7 Interrupt Architecture	370
Setup	370
Interrupt Sequence	371
Interrupt Service Routine (ISR)	371
Interrupt Service Thread (IST)	371
Latency	371
System Interrupts and Hardware Interrupts	372
Installable Interrupts	373
Watchdog Timer	374
Compact 7 Watchdog API	375
Using WatchDog Timers in System Development	376
The Vortex86 SoC Watchdog Timer	376
The Compact 7 Message Queue API	378
A Watchdog Timer Driver and Application	379
WDT_ISR	379
WDT_DRV	381
The Test Application	384
Using the WDT Test Application	387
Creating a Console Application with a Dynamic Link Library	388
The DLL Project	389
The Console Application	389
Testing the Projects	391
Summary	391
CHAPTER 32: STREAM INTERFACE DRIVERS	393
Loading a Driver	393
When Is a Driver Loaded?	393
Registry	394

Built-In Drivers	395
Installable Drivers	395
Bus Drivers	395
FileSys	396
Stream Drivers	396
Why Stream Drivers?	396
Stream Driver Architecture	396
Stream Driver Functions	397
Stream Driver Function Details	399
Stream Driver Configuration	401
Driver Naming Conventions	401
Registry Entries	402
BIB Entries	404
Kernel Mode Load	404
User Mode Load	404
Imaging a Developmental Driver	405
Driver Context	405
Device Context	405
Stream Context	407
Driver Classes	407
Application Streaming APIs	409
Asynchronous I/O	409
Power Management	410
Power Management Interface	411
Compact 7 Power States	411
Class Power Management Interfaces	413
Compact 7 Power Management Architecture	413
An Application to Test if a Stream is Loaded	415
Summary	415
 CHAPTER 33: DEVELOPING A STREAM INTERFACE DRIVER	 417
Stream Interface Driver Development Overview	418
The Stream Interface Functions	421
A Minimal Stream Driver	422
A Stream Driver with Open Context	422
Stream Power Functions	423
A Simple Stream Driver Project	423
A Compact 7 Stream Driver Project	424
The sources File	426
The Project Parameter Files	427
Stream Functionality	429
The Source Code File	430

Building a Stream Driver for Testing	433
Test the String Buffer Stream Driver	434
CEDriver Wizard	434
Platform Builder Scripts	435
Using CEDriverWiz	436
Implementing IOCTLs	440
Driver Context and Shared Memory	440
Registry Access from a Driver	441
Implementing Power Management	443
Summary	447
CHAPTER 34: STREAM DRIVER API AND DEVICE DRIVER TESTING	449
Debugging Overview	449
Build Configurations	450
First Some Simple Checks	451
Breakpoints	452
Debug Macros	453
Message Macros	454
Assert Macros	454
Debug Zones	454
Turning Debug Zones On and Off	456
Target Control	457
Using Remote Tools	457
Stream Driver API and Test Applications	458
The Stream Driver API	458
Loading the Stream Driver	458
Windows Embedded Test Kit (CTK)	461
Other Compact 7 Debugging Features	464
Target Control	464
CeDebugX	465
eXDI	465
Summary	466
CHAPTER 35: THE TARGET SYSTEM	467
BSP Overview	467
Some Compact 7 Target Boards	468
BSP Components	469
SDKs and BSPs	470
BSP Folders	470
Bootloader	470
XLDR	472