

"If you're interested in developing for this burgeoning platform, there is no one better able to get you up-to-speed."

—From the Foreword by **Rob Tiffany**,
mobility architect, Microsoft

Second Edition



Programming .NET Compact Framework 3.5



Paul Yao
David Durant

Contents

<i>Figures</i>	<i>xvii</i>
<i>Tables</i>	<i>xxi</i>
<i>Foreword</i>	<i>xxv</i>
<i>Preface</i>	<i>xxvii</i>
<i>Acknowledgments</i>	<i>xxxiii</i>
<i>About the Authors</i>	<i>xxxvii</i>

1 Mobile Phone Programming 1

1.1	Selecting an Application Programming Interface	4
1.1.1	<i>The Win32 API</i>	<i>8</i>
1.1.2	<i>The .NET Compact Framework</i>	<i>14</i>
1.1.3	<i>The Web Browser</i>	<i>17</i>
1.1.4	<i>Rich Internet Applications (RIAs)</i>	<i>18</i>
1.2	Memory Management	19
1.2.1	<i>Metadata Maps</i>	<i>19</i>
1.2.2	<i>The JITted Code Pool</i>	<i>20</i>
1.2.3	<i>Garbage Collector Pools</i>	<i>22</i>
1.2.4	<i>Garbage Collection and Data</i>	<i>23</i>
1.2.5	<i>Automatic Garbage Collection</i>	<i>25</i>
1.2.6	<i>Special Handling of Managed Data</i>	<i>27</i>
1.2.7	<i>Manual Memory Management of Native Data</i>	<i>36</i>
1.3	Conclusion	37

2	Extending Battery Life	39
2.1	What Is the Problem?	40
2.1.1	<i>Basic Questions to Ask</i>	41
2.1.2	<i>Factors Influencing Battery Life</i>	41
2.2	Measuring Battery Usage	43
2.2.1	<i>Talk Time and Standby Time</i>	44
2.2.2	<i>Power Measurement Method 1: Software-Only Approach</i>	45
2.2.3	<i>Power Measurement Method 2: Dedicated Hardware</i>	48
2.3	Device Power Study	51
2.3.1	<i>Standby Power Usage</i>	53
2.3.2	<i>Backlight Power Usage</i>	54
2.3.3	<i>Power Usage for Communicating</i>	57
2.3.4	<i>Multimedia Power Usage</i>	57
2.4	Conclusion	58
3	Platform Invoke	59
3.1	Overview of P/Invoke	60
3.1.1	<i>When to Use P/Invoke</i>	61
3.1.2	<i>Why We Prefer .NET Compact Framework Classes over Win32 Functions</i>	63
3.1.3	<i>Porting Native Code to Managed Code</i>	64
3.1.4	<i>Component Object Model (COM) Support</i>	65
3.2	Creating P/Invoke Declarations	66
3.2.1	<i>A Simple Function: MessageBox</i>	66
3.2.2	<i>Native Function Details</i>	68
3.2.3	<i>Function Return Values</i>	72
3.2.4	<i>Getting Started: C-Style Function Declarations</i>	72
3.3	Supported P/Invoke Function Parameters	75
3.3.1	<i>The Limits of Parameter Passing</i>	76
3.3.2	<i>Simple Data Types</i>	76
3.3.3	<i>Passing Parameters by Value versus by Reference</i>	79
3.3.4	<i>Passing String Parameters by Value</i>	82
3.3.5	<i>Structures</i>	82
3.3.6	<i>Type of Types versus Type of Parameters</i>	88
3.3.7	<i>Arrays</i>	90
3.4	A Sample Program: CallWin32	93
3.5	Writing Win32 Dynamic Link Libraries	96
3.5.1	<i>Declaring C++ Functions in a DLL</i>	97

3.6	Manual P/Invoke Parameter Passing	100
3.6.1	<i>The Marshal Class</i>	101
3.6.2	<i>Copying to Native Memory</i>	102
3.6.3	<i>Fine-Tuning Structures with the MarshalAs Attribute</i>	105
3.7	Communicating between Native and Managed Code	108
3.7.1	<i>The MessageWindow Class</i>	108
3.7.2	<i>Other Ways to Communicate between Native and Managed Code</i>	111
3.8	Comparing P/Invoke Support	112
3.8.1	<i>Windows CE-Specific Differences</i>	112
3.8.2	<i>.NET Compact Framework Implementation Details</i>	113
3.9	Conclusion	113
4	Data Binding to Controls	115
4.1	Data Binding	115
4.1.1	<i>Data-Bindable Controls</i>	117
4.1.2	<i>Data-Bindable Objects</i>	119
4.2	Complex Data Binding	124
4.2.1	<i>Using Complex Data Binding with ComboBox Controls</i>	125
4.3	Simple Data Binding	126
4.3.1	<i>The BindingsCollection Property</i>	126
4.3.2	<i>Formatting and Parsing</i>	129
4.4	The DataGridView Control	132
4.4.1	<i>Using Complex Data Binding with the DataGridView Control</i>	136
4.4.2	<i>Styling the Display of Data in a DataGridView Control</i>	137
4.4.3	<i>Creating Table and Column Styles</i>	139
4.4.4	<i>Creating Styles at Runtime</i>	142
4.4.5	<i>Responding to User Input</i>	144
4.4.6	<i>Using Simple Data Binding with the DataGridView Control</i>	149
4.4.7	<i>Accessing DataGridView Information</i>	149
4.4.8	<i>Providing Drill-Down Capability</i>	155
4.4.9	<i>Providing In-Place Editing Capability</i>	161
4.4.10	<i>Providing Automated In-Place Editing Capability</i>	167
4.5	Conclusion	177
5	Storage	179
5.1	Smart-Device Data Storage	179
5.1.1	<i>Installable File Systems</i>	181

5.1.2	<i>The Windows CE File System</i>	182
5.1.3	<i>ROM-Based Files</i>	182
5.2	File I/O	183
5.2.1	<i>The File and Directory Classes</i>	184
5.2.2	<i>Byte-Level I/O</i>	185
5.2.3	<i>Higher-Level I/O</i>	187
5.2.4	<i>Encoding and Decoding Data</i>	190
5.2.5	<i>Using the I/O Classes</i>	191
5.2.6	<i>Text File I/O</i>	193
5.2.7	<i>Binary File I/O</i>	196
5.2.8	<i>Writing Binary Data</i>	197
5.2.9	<i>Reading Binary Data</i>	199
5.2.10	<i>Binary I/O and Structures</i>	200
5.2.11	<i>XML File I/O</i>	203
5.2.12	<i>Easier XML Serialization</i>	210
5.3	Registry Access	213
5.3.1	<i>Opening and Creating Registry Keys</i>	214
5.3.2	<i>Reading and Writing Registry Values</i>	215
5.3.3	<i>Updating the Storage Sample Application to Use the Registry</i>	217
5.4	Conclusion	221
6	ADO.NET Programming	223
6.1	Examining ADO.NET	226
6.1.1	<i>A Layered Approach</i>	227
6.1.2	<i>The ADO.NET Classes</i>	229
6.1.3	<i>ADO.NET Error Handling</i>	234
6.2	Working with Data Sets	236
6.2.1	<i>Creating and Accessing DataSet, DataTable, and DataView Objects</i>	237
6.2.2	<i>Data Binding</i>	241
6.2.3	<i>Reading and Writing a Data Set as XML</i>	252
6.3	Microsoft SQL Server CE	254
6.3.1	<i>SQL Server CE Files</i>	255
6.3.2	<i>SQL Server CE Syntax</i>	256
6.3.3	<i>SQL Server CE Query Analyzer</i>	259
6.3.4	<i>Creating a SQL Server CE Database</i>	261
6.3.5	<i>Populating a SQL Server CE Database</i>	263
6.3.6	<i>Retrieving and Displaying Data</i>	266
6.3.7	<i>Updating a SQL Server CE Database</i>	275

6.3.8	<i>The SqlConnection Class</i>	276
6.3.9	<i>Querying Schema Information</i>	282
6.4	Microsoft SQL Server	287
6.4.1	<i>Connecting to SQL Server</i>	289
6.4.2	<i>Creating Command Objects</i>	297
6.4.3	<i>Using SQL Server Stored Procedures</i>	297
6.4.4	<i>Using Stored Procedures with DataSet Objects</i>	303
6.4.5	<i>DataSet Objects and Concurrency</i>	307
6.4.6	<i>Using Multi SELECT Stored Procedures</i>	307
6.4.7	<i>Working with Typed Data Sets</i>	310
6.5	Web Services	318
6.5.1	<i>XML, XSD, and SOAP</i>	319
6.5.2	<i>A Web Services Application</i>	321
6.5.3	<i>A Web Services Client Application</i>	331
6.6	Conclusion	335
7	LINQ	337
7.1	Overview	337
7.1.1	<i>Set Classes in .NET</i>	340
7.1.2	<i>LINQ in the Compact Framework</i>	341
7.1.3	<i>Deferred Execution</i>	343
7.2	The Sample Application	344
7.2.1	<i>LINQ to Datasets</i>	348
7.2.2	<i>LINQ to Objects</i>	355
7.2.3	<i>Business Object Properties</i>	366
7.2.4	<i>The Hybrid Version</i>	374
7.3	LINQ to XML	378
7.4	Conclusion	383
8	Synchronizing Mobile Data	385
8.1	Understanding SQL Server CE Synchronization	387
8.1.1	<i>Three Synchronization Mechanisms</i>	388
8.1.2	<i>IIS Connectivity</i>	389
8.1.3	<i>Database Connectivity</i>	390
8.2	Installing Remote Data Connectivity	392
8.2.1	<i>Creating the Virtual Directory</i>	392
8.2.2	<i>Configuring Additional Components</i>	396

8.3	Using RDA	397
8.3.1	<i>RDA Capabilities and Overhead</i>	397
8.3.2	<i>Programming for RDA</i>	398
8.4	Using Merge Replication	409
8.4.1	<i>Using Good Design to Avoid Synchronization Failures</i>	410
8.4.2	<i>Configuring Merge Replication</i>	415
8.4.3	<i>Programming for Merge Replication</i>	416
8.5	Choosing between Merge Replication and RDA	424
8.6	Using Data Synchronization Services	424
8.6.1	<i>Understanding Data Synchronization Service Requirements</i>	425
8.6.2	<i>Building a Data Synchronization Service</i>	427
8.7	Conclusion	448
9	The Remote API	451
9.1	RAPI Fundamentals	451
9.1.1	<i>Available RAPI Functions</i>	452
9.1.2	<i>Building .NET ActiveSync Applications</i>	452
9.1.3	<i>RAPI Startup and Shutdown</i>	458
9.2	Accessing the Object Store	470
9.2.1	<i>Using RAPI to Access Device Files</i>	471
9.2.2	<i>Remote Access to Device Registry Entries</i>	486
9.2.3	<i>Remote Access to Device Property Databases</i>	496
9.3	Detecting Changes in Device Connection State	499
9.3.1	<i>The Auto-Start Approach</i>	500
9.3.2	<i>The Callback Approach</i>	501
9.4	Loading Programs and DLLs	507
9.4.1	<i>Running Device-Side Programs</i>	508
9.4.2	<i>Loading Device-Side DLLs</i>	509
9.5	Conclusion	514
10	Windows Communication Foundation	515
10.1	What Is WCF?	515
10.1.1	<i>WCF Terminology</i>	518
10.1.2	<i>WCF in the .NET Compact Framework</i>	522
10.2	Creating a WCF Service	523
10.2.1	<i>Generating the Code</i>	523
10.2.2	<i>Making a Windows Mobile-Compatible WCF Service</i>	525

10.2.3	<i>Setting the Host Address</i>	526
10.2.4	<i>SmartMeter: A Sample WCF Service</i>	530
10.3	Creating a WCF Client in Windows Mobile	538
10.3.1	<i>WCF Client Namespaces and Assemblies</i>	538
10.3.2	<i>Generating the WCF Client Proxy</i>	539
10.3.3	<i>Instantiating a WCF Client</i>	541
10.3.4	<i>Accessing the WCF Service</i>	542
10.3.5	<i>WCF Client Sample: ReadSmartMeter</i>	542
10.4	Conclusion	546
11	Creating Graphical Output	549
11.1	An Introduction to .NET Compact Framework Graphics	549
11.1.1	<i>Drawing Surfaces</i>	551
11.1.2	<i>Drawing Function Families</i>	553
11.1.3	<i>.NET Compact Framework Graphics</i>	555
11.2	Drawing on the Display Screen	561
11.2.1	<i>Accessing a Graphics Object</i>	562
11.2.2	<i>Drawing in Controls</i>	564
11.2.3	<i>The Paint Event</i>	566
11.2.4	<i>Non-Paint Event Drawing</i>	568
11.3	Raster Graphics	570
11.3.1	<i>Specifying Colors</i>	570
11.3.2	<i>Creating Brushes</i>	577
11.3.3	<i>Creating Bitmaps</i>	579
11.3.4	<i>Drawing Bitmaps</i>	591
11.3.5	<i>A Sample Program: ShowBitmap</i>	596
11.4	Vector Graphics	602
11.4.1	<i>Creating Pens</i>	604
11.4.2	<i>A Game: JaspersDots</i>	605
11.5	Conclusion	619
12	Text and Fonts	621
12.1	Drawing Text	621
12.1.1	<i>Text-Drawing Support in the .NET Compact Framework</i>	622
12.1.2	<i>The DrawString Method</i>	623
12.1.3	<i>A Sample Program: SimpleDrawString</i>	624
12.2	Font Selection	625
12.2.1	<i>The Font Property of Controls</i>	626
12.2.2	<i>Enumerating Fonts</i>	627

12.2.3	<i>A Sample Program: FontPicker</i>	627
12.2.4	<i>A Sample Program: RotateText</i>	628
12.3	Placing Text	632
12.3.1	<i>Text Size and the MeasureString Method</i>	633
12.3.2	<i>A Sample Program: MeasureString</i>	634
12.3.3	<i>Text Alignment</i>	635
12.3.4	<i>A Sample Program: TextAlign</i>	636
12.4	Conclusion	640
A	Hungarian Notation for .NET Programs	641
A.1	Goals and Objectives	642
A.2	Guidelines	642
A.3	.NET Naming Guidelines	644
A.4	Hungarian Notation	646
A.4.1	<i>Our Use of Hungarian Notation</i>	647
A.4.2	<i>The m_ Prefix for Private Data</i>	647
A.4.3	<i>Hungarian Prefixes for CTS Value Types</i>	649
B	Windows API Allocation and Cleanup Functions	659