



WINDOWS DEVELOPER POWER TOOLS

Turbocharge Windows Development with 170+ Free Tools

O'REILLY®

James Avery & Jim Holmes

WINDOWS DEVELOPER POWER TOOLS

Turbocharge Windows Development with 170+ Free Tools

“James Avery and Jim Holmes have done great work in locating and documenting a vast collection of useful tools here, saving you the work of hunting them down one at a time.”

—From the Foreword by Mike Gunderloy, Editor, Larkware.com

This book is an encyclopedic guide to more than 170 free and open source programming tools, components, and frameworks available to help you create, test, and build top-notch Windows software—from Windows Forms and ASP.NET applications to web services. Much more than a simple listing, *Windows Developer Power Tools* also helps you choose the right tools for solving both common and uncommon problems that you face each day.

Inside, you'll find:

- Tools for mastering 23 common Windows and .NET software development tasks
- A mini user's guide for each of the 170 tools covered, many written by the tools' authors
- “Quick pick” lists that help you find and choose the right tool for a particular task
- Links to additional detail online when you wish to delve more deeply into features and functionality
- Information about the online Power Tools book site maintained by the authors to keep you up-to-date on new tools as they appear and to tell you of new releases and features of tools covered in this book

As good as it is, there's more to Windows development than Visual Studio. This one-stop resource will forever change the way you tackle the tasks of planning, developing, testing, and rolling out great software.

“Tools aim to save you time—this book aims to save you even more, by finding those gems that you can't live without.”

—From the Foreword by Scott Hanselman, <http://www.hanselman.com/tools>

James Avery is a Microsoft MVP and ASPInsider. He's written articles for *MSDN Magazine* and *ASPToday*, and is the author of *Visual Studio Hacks* (O'Reilly). Jim Holmes is a Microsoft MVP with nearly 25 years experience in the IT industry. He has written extensively for VisualStudioHacks.com.

US \$59.99 CAN \$77.99
ISBN-10: 0-596-52754-3
ISBN-13: 978-0-596-52754-9



O'REILLY®
www.oreilly.com

Safari Includes
BOOKS ONLINE FREE 45-Day
ENABLED Online Edition

WINDOWS DEVELOPER POWER TOOLS



WINDOWS DEVELOPER POWER TOOLS



James Avery and Jim Holmes

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

Windows Developer Power Tools

by James Avery and Jim Holmes

Copyright © 2007 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor: John Osborn

Developmental Editor: Ralph Davis

Production Editor: Mary Brady

Copyeditor: Rachel Wheeler

Proofreader: Mary Brady

Indexer: Angela Howard

Cover Designer: Marcia Friedman

Interior Designer: David Futato

Illustrators: Robert Romano and Jessamyn Read

Printing History:

December 2006: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Windows Developer Power Tools*, the image of a circular saw, and related trademarks are trademarks of O'Reilly Media, Inc.

Microsoft, MSDN, the .NET logo, Visual Basic, Visual C++, Visual Studio, and Windows are registered trademarks of Microsoft Corporation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN-10: 0-596-52754-3

ISBN-13: 978-0-596-52754-9

[C]

Table of Contents

Foreword	xv
Credits	xix
Preface	xxix

Part I Writing Code

1. Building ASP.NET Applications	3
1.1 Building Sophisticated Ajax Applications with ASP.NET Atlas	5
1.2 Adding Ajax Functionality with Anthem.NET	16
1.3 Adding Ajax Functionality with Ajax.NET Professional	20
1.4 Generating User-Friendly URLs with UrlRewritingNet.UrlRewrite	23
1.5 Showing a “Please Wait” Dialog with BusyBoxDotNet	26
1.6 Creating Cleaner HTML Output with CSS Friendly Control Adapters	32
1.7 Adding Word-Processing Capabilities to Your Application with FreeTextBox	39
1.8 Creating Charts with the WebChart Control	42
1.9 Consuming and Publishing RSS Feeds with the RSS Toolkit	46
1.10 For More Information	51
2. Working with Windows Forms	53
2.1 Getting “That WinXP Look” with XP Common Controls	54
2.2 Creating Dockable Windows with DockPanel Suite	63
2.3 Creating a Professional User Interface with Krypton Toolkit	69
2.4 Learning More About Windows Controls with Control Spy	74
2.5 For More Information	78

3. Developing in .NET 3.0 (a.k.a. “WinFx”)	80
3.1 Creating Great Visuals for Your Applications with Expression Graphic Designer	82
3.2 Creating WinFx User Interfaces Without Code Using Interactive Designer	90
3.3 Writing XAML in XamlPad	97
3.4 Avoiding XML Configuration Files with the Service Configuration Editor	102
3.5 Deciphering WCF Logs with the Service Trace Viewer	121
3.6 Developing WinFx Applications in Visual Studio with Development Tools for WinFx	126
3.7 For More Information	132
4. Working with Code Libraries	133
4.1 Externalizing Business Rules with NxBRE	134
4.2 Diagnosing Without Downtime via log4net	143
4.3 Searching Your Data Using Lucene.Net	149
4.4 Finding Changes Between Assembly Versions with LibCheck	161
4.5 Comparing Assemblies with Reflector.Diff	166
4.6 Implementing Spell Checking in Your Windows and Web Apps with NetSpell	172
4.7 Creating Graphs and Plot Charts Quickly with NPlot	176
4.8 Sorting Algorithms in C# with NSort	185
4.9 Creating RSS Feeds with RSS.NET	188
4.10 Using Zip, GZip, Tar, and BZip2 Archives in Your Software with SharpZipLib	191
4.11 Generating Excel Files from Code Using ExcelXmlWriter (Without Having Excel!)	195
4.12 Creating PDFs with iTextSharp	202
4.13 For More Information	207
5. Generating Code	209
5.1 Creating Code Automatically with MyGeneration	211
5.2 Generating Your Data-Access Layer with Codus	222
5.3 Writing Your Web Services Contract First with WSCF	226
5.4 Generating XML Schemas and Strongly Typed DataSets with XSD.exe	234
5.5 Practicing Contract-First XML Development with XSXObjectGenerator	240
5.6 For More Information	249

6. Writing Code	250
6.1 Building .NET Applications with Visual Studio Express	251
6.2 Developing .NET Applications Using SharpDevelop	270
6.3 Writing Mono Applications on Linux and Mac OS X with MonoDevelop	284
6.4 Creating Standalone Snippets with Snippet Compiler	290
6.5 Handling Small Tasks with Notepad2	293
6.6 Mastering Regular Expressions with the Regulator	297
6.7 Creating Regular Expressions with ReguLazy	303
6.8 For More Information	309
7. Creating Documentation	310
7.1 Documenting Your Source Code with GhostDoc	312
7.2 Viewing Documentation in Visual Studio with CR_Documentor	320
7.3 Creating Professional Documentation with NDoc	322
7.4 Printing PDF Documents with PDFCreator	328
7.5 Building a FAQ with skmFAQs	333
7.6 For More Information	340
8. Enhancing Visual Studio	341
8.1 Boosting Productivity with CoolCommands	342
8.2 Improving Visual Studio's Designers with the PowerToy for Class and Distributed Systems Designers	346
8.3 Opening Files Quickly with VSFileFinder	350
8.4 Mastering Unmanaged APIs with PINVOKE.NET	354
8.5 Spell Checking ASP.NET and HTML with the HTML/ASP.NET Spell Checker Add-in	359
8.6 Switching Window Layouts Quickly with VSWindowManager	366
8.7 Blogging or Emailing Your Code in Style with the CopySourceAsHtml Visual Studio Add-in	368
8.8 Editing CSS in Visual Studio with the CSS Properties Window	371
8.9 Bringing Web Application Projects Back to Visual Studio 2005	374
8.10 Sharing Your Visual Studio Goodies with the Content Installer Power Toys	378
8.11 For More Information	382

Part II Checking Code

9. Analyzing Your Code	387
9.1 Checking Complexity and Dependencies with Peli's Reflector Addins	391
9.2 Checking Your Source Code's Complexity with SourceMonitor	396
9.3 Analyzing Code Inside Visual Studio with CR_Metrics	404
9.4 Finding Your Code's Dependency Complexities with NDepend	409
9.5 Writing Better Code with FxCop	418
9.6 For More Information	427
10. Testing Your Software	429
10.1 Unit Testing Your Code with NUnit	433
10.2 Speeding Your Test Development with NUnit Code Snippets	450
10.3 Writing Advanced Unit Tests with MbUnit	452
10.4 Getting More from Your Tests with Zanebug	462
10.5 Bringing Your Customers into the Testing Process with FitNesse	468
10.6 Analyzing Unit-Test Coverage with NCover	480
10.7 Analyzing Code Coverage with NCoverExplorer	494
10.8 Integrating Unit Testing into Visual Studio with TestDriven.NET	504
10.9 Integrating Mock Objects into Your Testing with NMock 2.0	510
10.10 Creating Strongly Typed Mocks with Rhino.Mocks	522
10.11 Unit Testing Your GUI with NUnitForms	526
10.12 Creating Performance Benchmark Tests with NTime	532
10.13 Automating Web Application Testing with Selenium Core	537
10.14 Driving Your Web Applications Automatically with Watir	544
10.15 For More Information	549

Part III Running a Development Project

11. Working with Source-Control Systems	553
11.1 Setting Up Subversion Rapidly with SVN 1-Click Setup	556
11.2 Accessing Subversion and CVS with TortoiseCVS and TortoiseSVN	561
11.3 Using Subversion Inside Visual Studio with AnkhSVN	581
11.4 Removing Visual Source Safe Bindings with SourceSafe Binding Remover	587
11.5 Comparing Files with WinMerge	590
11.6 For More Information	594

12. Building, Using Continuous Integration on, and Deploying Your Applications	595
12.1 Creating Consistent Builds with NAnt	598
12.2 Automating Your Build Process with MSBuild	605
12.3 Building .NET 1.1 Assemblies in MSBuild with MSBee	610
12.4 Extending MSBuild Capabilities with MSBuild Community Tasks	614
12.5 Managing Your Build Process Visually with MSBuild Sidekick	619
12.6 Shortening the Development Cycle with CruiseControl.NET	626
12.7 Easing the Burden of Implementing a Continuous Integration Process with CI Factory	637
12.8 Simplifying Web Application Deployments with Unleash It	650
12.9 Easing Web Application Deployments with Web Deployment Projects	657
12.10 Creating a Setup Project with WiX	662
12.11 For More Information	665
13. Boosting Team Collaboration	667
13.1 Improving Team Communication and Collaboration with Basecamp	669
13.2 Building Online Communities with Community Server	677
13.3 Creating a Blog with Subtext	686
13.4 Collaborating Online with FlexWiki	689
13.5 Driving Another PC Remotely via XP's Remote Assistance	694
13.6 Talking to Colleagues with Skype	710
13.7 Chatting with Anyone Using GAIM	719
13.8 Administering Team Foundation Users with the TFS Administration Tool	723
13.9 For More Information	727
14. Tracking Bugs, Changes, and Other Issues	729
14.1 Implementing a Bare-Bones Defect Tracker with BugTracker.NET	731
14.2 Handling Mid-Sized Projects with CodeTrack	737
14.3 Running an Enterprise-Level Bug-Tracking System with Bugzilla	747
14.4 Managing a Project with Trac	755
14.5 For More Information	762

Part IV Troubleshooting Code and Applications

15. Troubleshooting and Debugging	765
15.1 Monitoring File Access in Your System with Filemon	767
15.2 Identifying TCP Port and Endpoint Issues with TCPView	771
15.3 Seeing What's Being Accessed in Your Registry with Regmon	774
15.4 Getting a Better View of Processes on Your System with Process Explorer	776
15.5 Digging into Your Program's CLR Interactions with Pinvoke and COM Using CLR SPY	782
15.6 Tracking Down Memory-Allocation Problems with CLR Profiler	793
15.7 Debugging Your Application After It's Been Deployed with ClrDump	798
15.8 Debugging Hangs and Monitoring Processes with Managed Stack Explorer	804
15.9 Examining HTTP Traffic with Fiddler	809
15.10 Identifying Network Problems with PingPlotter Freeware	815
15.11 Examining Logs with LogParser	818
15.12 Simplifying Debugging with Visual Studio 2005 Visualizers	825
15.13 Finding Bottlenecks with NProf	831
15.14 For More Information	835
16. Using Decompilers and Obfuscators	837
16.1 Examining Common Ground with ILDASM	839
16.2 Analyzing Assemblies with Reflector	850
16.3 Debugging .NET Assemblies Without Source Code Using Dotnet IL Editor	856
16.4 Preventing Reverse-Engineering of Your Code with Dotfuscator	868
16.5 For More Information	876

Part V Code Tools

17. Tightening Up Your Security	881
17.1 Analyzing Threats Your Application Faces with the Threat Analysis & Modeling Tool	882
17.2 Protecting Your Communications with the Bouncy Castle Cryptography APIs	896
17.3 Mitigating XSS Vulnerabilities with the Anti-Cross Site Scripting Library	906

17.4	Highlighting Explorer Windows' Security Privileges Using PrivBar	909
17.5	Opening a Command Window with Admin Privileges Using MakeMeAdmin.cmd	912
17.6	For More Information	914
18.	Building Your Application on Frameworks	917
18.1	Reusing Professional Infrastructure with Enterprise Library	919
18.2	Creating Web Applications and Systems with DotNetNuke	928
18.3	Improving Smart Device Development with the Smart Device Framework	937
18.4	Building and Deploying Cross-Platform .NET Applications with Mono	943
18.5	Spelunking the Framework with Rotor	956
18.6	Simplifying Web Development with Castle MonoRail	963
18.7	Obtaining a Decoupled Architecture with the Castle Windsor Container	972
18.8	Improving Your Code's Design and Testability with PicoContainer.NET	980
18.9	For More Information	988
19.	Working with XML	990
19.1	Transforming XML Documents with nXslt2	992
19.2	Displaying XML on the Web with eXml	995
19.3	Building Composite XML Documents with XInclude.NET	1000
19.4	Referencing Part of an XML Document with XPointer.NET	1004
19.5	Extending XSLT Processing with EXSLT.NET	1009
19.6	Comparing and Unit Testing XML with XML Diff and Patch	1015
19.7	For More Information	1021

Part VI Working with Databases

20.	Interacting with Databases	1025
20.1	Finding the Proper Connection String with ConnectionStrings.com	1027
20.2	Creating SQL Connection Strings Quickly with SQL Server Connection String Builder	1030
20.3	Administering SQLite Databases with the SQLite Administrator	1032
20.4	Working with Oracle Inside Visual Studio with Oracle Developer Tools	1038

20.5	Managing Oracle Databases with Oracle SQL Developer	1045
20.6	For More Information	1050
21.	Exploring Object/Relational Mapping	1052
21.1	Mapping Objects to the Database with NHibernate (or How to Stop Writing All That Data-Access Code)	1054
21.2	Mapping Business Objects with NPersist	1062
21.3	Easing the Burden of Creating O/RM Files with ObjectMapper	1072
21.4	Simplifying Data Access with Castle ActiveRecord	1079
21.5	For More Information	1085

Part VII Miscellaneous

22.	Enhancing Web Development	1089
22.1	Avoiding Web Headaches with the Web Developer Extension for Firefox	1091
22.2	Debugging Web Pages with the Web Developer Toolbar for Internet Explorer	1098
22.3	Diagnosing JavaScript and Ajax with Web Development Helper	1102
22.4	Eliminating Memory Leaks in Internet Explorer with Drip	1109
22.5	Debugging Web Pages at Runtime with DOM Helper	1116
22.6	Checking Web Pages with the W3C Markup Validation Service	1123
22.7	Checking Your Web Site's Accessibility with WebXACT	1126
22.8	Cleaning Up Your HTML with HTML Tidy	1130
22.9	For More Information	1133
23.	Boosting Productivity with Windows Utilities	1134
23.1	Launching Tools and Web Sites Quickly with SlickRun	1136
23.2	Spotlighting Content in Your Presentations with ZoomIt	1142
23.3	Getting to the Right Detail with Magnifixer	1145
23.4	Extracting Files from Windows Installer Packages with Less MSIsérables	1149
23.5	Handling FTP More Efficiently with FileZilla	1152
23.6	Opening a Shell Where You Need It with Command Prompt Here	1157
23.7	Grabbing Screenshots with Cropper	1159
23.8	Grabbing Color Values with ColorMania	1163
23.9	Bringing .NET to the Command Line with PowerShell	1167
23.10	Getting the Most out of PowerShell with PowerShell IDE	1177
23.11	Managing and Switching Tasks with TaskSwitchXP	1182

23.12	Keeping Data in Sync with SyncToy	1186
23.13	Resolving Locking Problems with Unlocker	1191
23.14	Bringing POSIX Emulation to Windows with Cygwin	1194
23.15	For More Information	1207

**Part
VIII**

Appendix

Common Tasks in Visual Studio	1211
Index	1219

Foreword

“We shall not fail or falter; we shall not weaken or tire... Give us the tools and we will finish the job.”

—Sir Winston Churchill (1874–1965)
BBC radio broadcast, Feb. 9, 1941

I love my toolbox. I hate wasted time, wasted keystrokes, wasted potential. I’ve worked as a professional developer using Microsoft operating systems of one kind or another for nearly 15 years. As a developer, I’ve always tried to get the most out of my hardware and software, and as I’ve learned new techniques or discovered faster ways to work, I’ve gathered tools for my ever-growing collection. My toolbox is my collection of not only cool utilities, but also best practices as a user and a developer.

I remember the joy of squeezing another 2K out of upper memory on my 386SX. The tool that made it possible? QEMM from Quarterdeck. Didn’t have enough space on my massive 20 MB drive? Stacker gave it new life, if only for a few days. Then Windows 3.1 came along, and with it new tools like Norton Desktop and PKZIP. Windows 95 opened a world of Power Toys and Kernel Toys and WindowBlinds to manipulate my reality. The advent of Visual Basic 6 brought with it tools garnished with lime-green backgrounds and purple buttons, but tools they were and use them I did.

When the .NET Framework came along, I continued to add to my tool belt with gems from the increasingly prolific .NET open source community. Since 2004, I’ve maintained a pretty long list of tools at <http://www.hanselman.com/tools/> that I revise each year. My list of tools has breadth but is short on depth. This book fills in the gaps so many simple tool lists contain—it gives us deep context, a better understanding of *why* to choose certain tools, and information on how to use those tools effectively.

James and Jim have done a massive amount of work for us, applying their years of development expertise and enthusiasm for great tools into the creation of the book you’re holding. James’s enthusiasm for hacks and tools has even overflowed onto the pages of his web site, <http://www.visualstudiohacks.com>.

Why buy a book filled with lists of tools? Tools aim to save you time, and this book aims to save you even more, by tracking down for you those gems you won't be able to live without. Far from being just a simple list, this compilation is filled with applets and applications along with deep analysis of and commentary on their relevance to your life as a developer. Some of these tools you'll recognize; some you won't. All can be incorporated into your own Windows development process and have been carefully selected to help *you* finish the job.

—Scott Hanselman

<http://www.computerzen.com> (my blog)

<http://www.hanselman.com/tools/> (the tools list)

The Scottish essayist Carlyle famously defined man as the tool-using animal. Somewhere in the deep past, our monkey ancestors started banging together stones and bones and brought forth hand axes, and the idea caught on. These days, the urge to build tools seems to run particularly strong among software developers. Give a developer a choice between writing some code and writing a general-purpose code generator to write the code for him, and you'll find him staying up late cranking out a wonderfully complex framework that will make life easier down the road.

Microsoft long ago recognized this bit of monkey behavior among some of their customers and started figuring out ways to take advantage of it. You can trace a pattern through Windows, Office, and Visual Studio of documenting APIs, providing sample code, and implementing extensibility points, with the end result being that it has become easier and easier for external developers to layer their own tools on top of Microsoft's offerings. If you're faced with a repetitive task in Windows, you can automate it (the new PowerShell command processor brings new levels of flexibility to this arena). If you're solving business productivity problems, you can inject your own code into Microsoft Office in a variety of ways. If you're a developer writing tools for developers, you can tap the dozens of extensibility points and thousands of APIs in .NET and Visual Studio to create almost any tool imaginable.

This last ecosystem has been one of the most fertile in the Microsoft universe, and it's the one that the authors of the book that you hold in your hands have chosen to explore. Several factors come together to explain the explosion of Windows developer power tools in the last few years: the incredible richness of the .NET Framework, the sheer volume of the available documentation, and the rise of the Internet as a home for developer-to-developer interaction have all contributed to a rich environment for tool-building. But there's another factor, too: developers are inclined to scratch their own itches.

Developers see the problems in their mainline development tools—whether from Microsoft, IBM, Borland, or another major player—every day. When they think about building tools of their own to respond to problems, what’s more natural than to solve their own problems first? It’s this natural urge, together with the desire to share (and perhaps to show off), that gives us the rich variety of tools showcased in the present volume. If you’re only using shrinkwrapped tools from major vendors, no matter how good they are, you’re missing out on an amazing collection of useful code.

Not every developer will need every tool in this book, of course. There’s the ever-present danger of getting so loaded down by your toolbox that you can barely move. But if there are rough edges in your process, places where you feel vaguely unhappy and inefficient, the chances are pretty good that someone else has had the same problem before you. The solution may be just a few pages away. James Avery and Jim Holmes have done great work in locating and documenting a vast collection of useful tools here, saving you the work of hunting them down one at a time. The half-dozen that make their way into your daily process will more than repay the time that it takes you to read their work.

—Mike Gunderloy
Editor, Larkware.com

Credits

About the Authors

James Avery is an accomplished author and .NET architect. In addition to having coauthored this book, James is the author of *Visual Studio Hacks* (O'Reilly) as well as books for Microsoft Press and Wrox. He has also contributed articles to *MSDN Magazine*, *Dr. Dobb's Journal*, and *ASP Today*.

After working for a number of large corporations and consulting companies, James left his last job to go out on his own and formed his own company, Infozerk, Inc. (<http://www.infozerk.com>). James also started and is the current president of the Cincinnati .NET User Group, and he cofounded the Cincinnati-Dayton Code Camp (whose name is still in dispute).

James lives in Cincinnati, OH with his wonderful wife Tammy and their four cats. When he is not working on his laptop, you can find him fiddling with his digital cameras, reading, or playing Xbox 360.

James is a frequent blogger, and you can find his blog at <http://www.dotavery.com/blog/>. James can also be contacted by email at javery@infozerk.com.

Jim Holmes, a Microsoft MVP for C#, spent 11 years running radar systems in flight on E-3 AWACS aircraft and has nearly 25 years of experience in the IT industry, including network management, systems analysis, and software development in Perl, Java, C++, and .NET. Jim has also worked as a retail wine sales clerk, so he can help you decide what wine you should drink with the pizza you're eating as you're working overtime to meet your software delivery deadline. Currently, Jim is a principal consultant for NuSoft Solutions (<http://www.NuSoftSolutions.com>).

Jim is the founder of the Dayton .NET Developers Group and cofounder of the Dayton-Cincinnati Code Camp, the proper name for the conference he created with James Avery. He writes regular columns for VisualStudioHacks.com, including the *Ask the Pros* series, where industry leaders talk about how they get the most out of

Visual Studio. Jim is a frequent poster at his blog, FrazzledDad (<http://frazzleddad.com>), where he talks about trying to work from home while taking care of two small children, rose gardening, cooking, and doing software development.

Jim can be reached by email at Jim@IterativeRose.com.

Contributors

George Aroush has over 15 years of experience in the IT domain. He has held positions in engineering, architecture, and R&D, and has served as lead for various teams and projects. George has been active with open source solutions since 2000 and has been leading the Lucene.Net project since 2003. Today, as his day job, George works for IBM Rational on the award-winning product ClearQuest. He holds a master's degree in computer science from Northeastern University and a bachelor's degree from Tufts University.

Nigel Atkinson is one of the founders of Neoworks, the software development company that originally created log4net. Nigel has worked as a software engineer, project manager, and consultant in the software industry for the past 10 years and is currently Neoworks's Business Development Director.

Alexander Avdonin is the author of TaskSwitchXP and several other utilities. You can find his home page at <http://www.ntwind.com>.

Thomas Bandt is coauthor of UrlRewritingNet.UrlRewrite and a specialist for ASP.NET e-commerce and content-management applications. He loves Microsoft .NET and programming with the Framework, as well as other computer stuff—almost as much as the German national soccer team. Thomas lives in Germany. You'll find his blog (in German) at <http://blog.thomasbandt.de>.

Nino Benvenuti is a Microsoft MVP (Visual Developer - Device Application Development) who is focused on creating .NET-based mobility solutions. Throughout his career, he has created a variety of desktop, server, and device solutions using a range of technologies for a diverse set of clients. Nino is active in the Cincinnati .NET community as well as in the Pocket PC, Smartphone, and .NET CF newsgroups. You'll find his blog and other items at <http://www.nino.net>.

Jeff Blankenburg is a freelance web developer based in Columbus, OH. Since 1998, Jeff has been passionately pursuing innovative new ways to create a user-centric experience using web technologies. His specialties include CSS, XHTML, JavaScript, and C#. You can contact him directly at innovation@jexed.com or look for his latest efforts in a browser near you.

Peter Boey is a software engineer with years of experience working for international companies like Nike and Philips. In 1999, Peter founded Blacksun Software, which creates tools and utilities for the Microsoft Windows platform. These tools were primarily created for Blacksun's own needs but quickly became very popular on the Internet.

Simone Busoli is enrolled in the Computer Science Engineering program at the University of Modena and Reggio Emilia, Italy. He likes working on web applications and especially on all aspects of code reusability. This interest has led him to become involved in the development of open source projects, with a focus on ASP.NET web control libraries.

Brian Button is the VP of Engineering and Director of Agile Methodologies at Asynchrony Solutions, Inc. Previously, he was a consultant for patterns and practices on the Microsoft Enterprise Library project from its inception until its final delivery. Brian is currently responsible for Agile training, mentoring, and consulting practices, and for providing Agile project management advice. He can be reached through his blog at <http://oneagilecoder.asolutions.com> or via email at brian.button@asolutions.com.

Ben Carey has over nine years of experience working for various product-development and consulting companies. Ben specializes in technical leadership, architecture, patterns, test-driven development, and Agile methodologies.

Marc Clifton is the creator of MyXaml, an open source declarative XML instantiation engine. He is an industry consultant working primarily with companies interested in utilizing declarative programming concepts to add flexibility to n-tier architectures on web, CE, and desktop platforms. His other major open source project is the Advanced Unit Test framework. He operates his own web site, <http://www.marclifton.com>, where you will find many of his articles.

Hristo Deshev is a Group Product Manager for telerik, a leading vendor of ASP.NET controls (<http://www.telerik.com>). He has been working on component development for the past several years, focusing on rich, highly interactive Ajax solutions. A major goal of his team is “taming web development”—applying Agile software practices to create robust solutions targeting all modern web browsers and supporting multiple ASP.NET and Visual Studio versions. You can reach Hristo at hristo.deshev@telerik.com.

Peter Dettman is a graduate engineer living in Melbourne, Australia. He has 10 years of experience as a professional software developer, using a variety of languages (primarily Java and C#, of late). Specific areas of interest include distributed programming, concurrency, compilers, and programming language theory. For the last year or so, he has been the main developer on the C# port of the Bouncy Castle Cryptography API, now nearing its Version 1.0 release.

Michael Dobler started his career as a Civil Engineer but became attracted to the IT sector in 1995. He has programmed in various VB versions and exotic 4GL languages, always focusing on pure business applications. Currently, he is working as Software Development Manager for a large-scale CRM/BI solutions company. In 2004, Michael won the Code Hero award from <http://www.windowsforms.net> with his XPCCommonControls. You can find the controls and more information at his personal web site, <http://www.steeptvalley.net>.

David Dossot is the lead Java architect at Agile Partner S.A., a company he cofounded in Luxembourg. He is a were-programmer who enjoys Java during the day and .NET when night falls. He is also an occasional author for the late *Software Development Magazine* (now *Dr. Dobb's Journal*) and a judge for the Jolt Awards. Contact him at david@dossot.net or through his web site, <http://www.dossot.net>.

Grant Drake (a.k.a. “kiwidude”) is a developer from New Zealand who has been writing software for over 20 years. For the last six, he has resided in London, consulting to the corporate and banking industries as a senior developer specializing in Microsoft technologies. A development automation and tool junkie himself, his recent contribution back to the .NET community is NCoverExplorer. If dragged away from his keyboard, he spends his spare time inline slalom skating, snowboarding, and wondering when to leave for a better climate. He can be reached at grant@kiwidude.com.

Michael Dvoishes is the Technical and General Manager of Trivium Technologies. He began his career in programming during the late 1980s and received his M.S. degree in 1993. After working for several high technology startups, Michael started his own business, specializing in providing services to such companies. Currently, Michael spends his working time managing the day-to-day life of the company, keeping up to date with the latest technology advancements, and even doing some coding.

Dan Fernandez is Lead Product Manager for Visual Studio Express in the developer division at Microsoft. He has been with Microsoft since 2001 and has worked in multiple roles, including as the C# Product Manager and as a Developer Evangelist in the Mid-Atlantic district. Prior to joining Microsoft, Dan worked as a developer at several consulting firms, including IBM Global Services, specializing in web-based and mobile application development.

Jay Flowers is the creator of CI Factory.

Sara Ford is the Program Manager for the Power Toys for Visual Studio. Previously, she was a Software Design Engineer in Test for the Visual Studio Core team, where she drove the effort to make the Visual Studio 2005 product accessible to developers who are vision-impaired. Sara graduated from Mississippi State University with B.S. degrees in computer science and mathematics. Her blog can be found at <http://blogs.msdn.com/saraford/>.

Justin Greenwood has been a software developer for an Indianapolis-based consulting firm for the past six years. Like many others who love the field, he also enjoys working on side projects that pique his interest. In early 2004, he teamed up with coworker Mike Griffin to write the MyGeneration code generator. Since then, Justin, Mike, and a couple of close friends have supported the MyGeneration project and spawned several related projects, such as the dOODads, EntitySpaces, and Easy-Objects persistence frameworks.

sample content of Windows Developer Power Tools: Turbocharge Windows Development with more than 170 free tools

- [In Science's Shadow: Literary Constructions of Late Victorian Women pdf, azw \(kindle\), epub, doc, mobi](#)
- [download The Jews of Yugoslavia: A Quest for Community](#)
- [download online Lola and the Boy Next Door \(Anna and the French Kiss, Book 2\) pdf, azw \(kindle\), epub, doc, mobi](#)
- [Furniture Design pdf, azw \(kindle\), epub](#)
- [Heart of Gold pdf](#)
- [read online The Mayflower and the Pilgrims' New World here](#)

- <http://crackingscience.org/?library/Case-Files-Biochemistry--3rd-Edition---Lange-Case-Files-.pdf>
- <http://toko-gumilar.com/books/Inheritance--Inheritance-Cycle--Book-4-.pdf>
- <http://www.satilik-kopek.com/library/Lola-and-the-Boy-Next-Door--Anna-and-the-French-Kiss--Book-2-.pdf>
- <http://twilightblogs.com/library/Furniture-Design.pdf>
- <http://hasanetmekci.com/ebooks/Heart-of-Gold.pdf>
- <http://aneventshop.com/ebooks/The-Mayflower-and-the-Pilgrims--New-World.pdf>