

Contents

<i>About the Author</i>	<i>xi</i>
 Chapter 1 <i>Debugging Introduction</i>	1
<i>Identifying the Problem</i>	1
Think Before You Leap	2
Was It Worth It?	2
Improving Debugging Skills	3
<i>What .NET Adds to Debugging</i>	4
No More Memory Problems	5
Language Independence	5
Remote Debugging and Cross-Machine Debugging	7
Detach from Process	8
ASP.NET	8
<i>Still More to Come</i>	9
 Chapter 2 <i>Quickies: Six Pieces</i> <i>of Debugging Advice</i>	 11
1) <i>Never Ignore a Bug That Occurs Early in a Test</i> <i>to Investigate a Bug That Occurs Later</i>	12
2) <i>Never Forget the Goal Is to Make the Program Work—</i> <i>Fixing the Bug Is the Means, Not the End</i>	15
When Goals Are Dependent on Subgoals, Which Are Dependent on Even More Subgoals... ..	17
When the Going Gets Tough, the Tough Look for an Easier Way	17
3) <i>When Something Works One Way but Fails</i> <i>in a Different Way, Focus on the Differences</i>	18
Focus on the Differences	19
When Something Works One Way... ..	20
4) <i>Never Assume the Problem Is a Bug in the Hardware,</i> <i>API, or OS Without Reasonable Proof</i>	21
The Burden of Proof	23
5) <i>Keep a Few Test Computers on Which Debugging Tools</i> <i>Are NEVER Installed—No Exceptions. Period.</i>	24
Verifying Your Code Works on a Non-Developer Environment	25
Ensuring Your Product Can Be Debugged at a Customer Site	28
Giving Your Team Practice in Other Forms of Debugging	28

6) <i>Step Over All New Code in the Debugger As Soon As You Finish Writing It</i>	29
Forgetting to Fill In the Details of a Function	30
Discovering Your Error Handling Code Was Hiding a Serious Problem	30
Changing Code As a Test but Accidentally Checking That Code In Anyway	31
Missing a Chance for an Optimization	32
The Bottom Line	34
<i>Highlights</i>	34
 Chapter 3 Always Have a Plan	35
<i>Brute-Force Debugging</i>	35
<i>Three Guesses</i>	37
Still Don't Believe Me? I'll Prove It to You	38
Set Up a Test Case	41
Guessing Wisely	41
Ever Seen Something Like This Bug Before?	41
<i>The Cell Phone Problem</i>	43
When You Assume, You Make an Ass out of "U" and "Me"	43
Gather Information—Things Are Working Until WHEN?	46
Reasoning About Implementation Details	49
Drilling Deeper into the Problem	53
The Payoff—What Was the Source of All This Trouble?	55
<i>Highlights</i>	56
 Chapter 4 Assertion Debugging	59
<i>What Are Asserts?</i>	59
The Problems with Message Boxes	60
Using Asserts to Launch the Debugger	61
<i>"But I Always Test My Code, So Why Do I Need Asserts?"</i>	63
Finding the Point of Failure	63
Side Benefits of Asserting	66
<i>.NET's Debug and Trace Classes</i>	68
<i>Using Asserts Aggressively</i>	71
<i>Asserting Performance Bottlenecks</i>	72
<i>Don't Assert Legal Cases</i>	75

<i>When Asserts Can't Be Easily Used</i>	76
Asserts with Windows Services	76
Asserts with ASP.NET Pages and Web Services	77
Asserts with Remote Objects	80
<i>Customizing Asserts with a TraceListener</i>	80
When the Customized Assert Fires	82
Using Customized Asserts in a Service	84
<i>Highlights</i>	85
 Chapter 5 Debugging with Logs	87
<i>Printf Debugging</i>	87
<i>Program Logs</i>	88
What Information Should Be in the Logs?	90
Making Logs Readable	95
<i>.NET's Support for Logging</i>	100
Listeners and Switches in .NET	103
XML Config Files	106
<i>Windows Event Logs</i>	108
<i>Highlights</i>	111
 Chapter 6 ASP.NET and SQL Debugging	113
<i>ASP.NET and the Debugger</i>	114
Attaching the Debugger	115
Remote Debugging	119
Detaching the Debugger	120
<i>Logging with ASP.NET</i>	122
ASP.NET Tracing Data	123
Enabling ASP.NET Tracing	130
<i>Debugging SQL Stored Procedures</i>	135
Stepping Into Stored Procedures Directly	136
Debugging Stored Procedures in Your Application	137
Remote SQL Debugging	138
<i>Highlights</i>	139

Chapter 7 Debugging Remote Customer Sites	141
<i>Why Is It So Hard to Debug a Remote Customer Issue?</i>	141
Configuration Issues	143
You Must Try to Reproduce the Bug	145
<i>“Why Are You Asking Me All These Questions?”</i>	
<i>Why Can’t You Just Fix It?!?”</i>	146
What the Customer Expects	146
Why the Customer Can’t Get That	147
<i>Automated Diagnostic Utilities</i>	147
Automatically Gathering the Data	148
Design Decisions About the Diagnostic Utility	148
<i>Third-Party Tools Let You Watch the Customer</i>	
Reproduce the Bug	151
Seeing the Bug with Your Own Eyes	151
Taking Control of the User’s Desktop	152
<i>For Very Hard Bugs, Let Developers Participate</i>	
<i>in the Tech Support Phone Calls</i>	153
Each Involved Person Filters Out More Information	153
What Developers Should Say to the Customer	154
<i>Make a New Build with More Logging—and a Way</i>	
<i>to Verify Whether It Gets Installed</i>	155
.NET Security	157
Limiting Access	157
How CAS Works	159
Handling SecurityExceptions	161
Permissions Are Granted on a Per-Assembly Basis	162
Highlights	165
 Chapter 8 Multithreaded Debugging	 167
<i>What Is Multithreading?</i>	167
<i>Viewing Threads in the Debugger</i>	170
<i>Causes of Common Threading Problems</i>	171
Race Conditions	172
Deadlocks	176
Starvation	177
<i>Debugging Threading Problems</i>	180
Debugging Race Conditions	183
Debugging Deadlocks	189
<i>Final Thoughts About Threading</i>	197
Highlights	197

Chapter 9 Defect Tracking Programs	199
<i>Why Do You Need Defect Tracking Software?</i>	200
All the Details You're Not Aware Of	201
What Is a Defect Tracking Program?	202
<i>Development Workflows</i>	205
Common Problems with Workflows	206
Enforcing a Workflow	207
Handling Bounce-Backs in the Workflow	209
<i>Searching and Reporting</i>	209
Duplicate PRs	210
Searching Past Bugs for Clues About the Present	211
Reporting on PRs	213
<i>Communicating with Teammates</i>	216
Collaborating with Other Developers	216
Collaborating with Testers	217
Collaborating with Tech Support	220
<i>Know Your Tools</i>	221
<i>Highlights</i>	221
 Chapter 10 Source Control Debugging	 223
<i>Introduction to Source Control</i>	224
Using Visual SourceSafe	225
Allow Developers to Work Together	227
See a History of All Changes	229
Fetch a Particular Version or Branch a Copy	231
Protect Against Accidental Deletion of Files	232
<i>Debugging with SourceSafe</i>	233
Debugging with Change Histories	233
Bug Fixing with Branch Management	242
<i>Learn to Love SourceSafe</i>	249
<i>Highlights</i>	249
 Epilogue	 251
 Index	 253



<http://www.springer.com/978-1-59059-059-1>

Debugging Strategies For .NET Developers

Dillon, D.

2003, XI, 272 p., Softcover

ISBN: 978-1-59059-059-1

A product of Apress