The Elements of C# Style

The Elements of C# Style is for all C# practitioners, especially those working in teams where consistency is critical. Just as Strunk and White’s The Elements of Style provides rules of usage for writing in the English language, this text furnishes a set of rules for writing in C#. The authors offer a collection of standards and guidelines for creating solid C# code that will be easy to understand, enhance, and maintain.

This book provides conventions for

- formatting
- naming
- documentation
- programming
- and packaging

for C# 2.0 and includes discussion of advanced topics such as generics.

Kenneth Baldwin is currently President of CenterSpace Software, a developer of C# numerical libraries for the .NET platform. He has previously written documentation for C++ and Java class libraries for Rogue Wave Software, mathematical and statistical tools for MathSoft Inc., 3D modeling and animation tools for Cubicomp, and semantic networks for Microsoft. He has also developed Web-based learning systems for Medlink and the Stanford University School of Education. He holds a PhD in Cognitive Science from the University of California, San Diego, and a BA in Linguistics from the University of California, Berkeley.

Andrew Gray is a coauthor of The Elements of C++ Style. He is currently an Executive Partner with Evolution Software Design, focusing on .NET, Tablet PCs, and XML/Web Services. He cohosts TabletDev (www.tabletdev.com), a resource site for the Tablet PC development community. He has many years of Windows development experience, working for IntellChem, Rogue Wave Software, ACCEL Technologies, and Microsoft. He holds an MS degree in Computer Science from the University of California, San Diego, and a BS degree in Computer Science from Harvey Mudd College.

Trevor Misfeldt is a coauthor of The Elements of Java Style and The Elements of C++ Style. He has held programming positions for a variety of firms using C#, C++, Java, and other technologies. He is currently CEO of CenterSpace Software, a developer of C# numerical libraries for the .NET platform, and has worked previously for Rogue Wave Software, CleverSet Inc., and ProWorks. He holds a BSc in Computer Science from the University of British Columbia and a BA in Economics from the University of Western Ontario.
Contents

Preface .................................................. ix
Audience .................................................. ix

Introduction ............................................ 1
Disclaimer ............................................... 2
Acknowledgments ....................................... 3

1. General Principles ................................... 4

2. Formatting ........................................... 7
   2.1 White Space ...................................... 7
   2.2 Braces .......................................... 14
   2.3 Class Organization ............................... 19

3. Naming ............................................. 22
   3.1 General ......................................... 22
   3.2 Abbreviations .................................... 25
   3.3 Preprocessor Symbols ............................ 26
   3.4 Types and Constants ............................. 27
   3.5 Enumerations ................................... 29
   3.6 Interfaces ........................................ 29
   3.7 Properties ...................................... 30
   3.8 Methods ........................................ 31
   3.9 Variables and Parameters ..................... 32
   3.10 Attributes ..................................... 35
   3.11 Namespaces .................................... 35
vi CONTENTS

3.12 Event Handling .................................. 35
3.13 Exceptions ....................................... 36

4. Documentation ................................ 37
4.1 General .......................................... 37
4.2 API .............................................. 40
4.3 Internal Code ..................................... 43

5. Design .......................................... 49
5.1 Engineering .................................... 49
5.2 Class Design .................................... 54
5.3 Thread Safety and Concurrency ................. 63
5.4 Efficiency ....................................... 66

6. Programming .................................. 70
6.1 Types ........................................... 70
6.2 Statements and Expressions ........................ 74
6.3 Control Flow .................................... 76
6.4 Classes .......................................... 80
6.5 Life Cycle ....................................... 85
6.6 Fields and Properties .............................. 90
6.7 Methods .......................................... 92
6.8 Attributes ....................................... 94
6.9 Generics ......................................... 96
6.10 Enumerations .................................... 96
6.11 Type Safety, Casting, and Conversion ............... 99
6.12 Error Handling and Debugging ................... 100
6.13 Events, Delegates, and Threading .................. 106

7. Packaging .................................... 108
7.1 Files ............................................ 108
7.2 Namespaces ..................................... 109
7.3 Assemblies ....................................... 112
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>115</td>
</tr>
<tr>
<td>Glossary</td>
<td>125</td>
</tr>
<tr>
<td>Bibliography</td>
<td>139</td>
</tr>
<tr>
<td>Index</td>
<td>141</td>
</tr>
</tbody>
</table>
As commercial developers of software components, we always strive to have good, consistent style throughout our code. Since source code is usually included in our final products, our users often study our code to learn not just how the components work, but also how to write good software.

This fact ultimately led to the creation of *The Elements of Java Style* and *The Elements of C++ Style*. The positive reception of those books, coupled with recurring questions about C# and .NET style issues, resulted in this edition for C#.

If you've read the earlier books in this series (or even if you haven't), much of the advice in this book will probably be familiar. This is deliberate, as many of the programming principles described are timeless and valid across programming languages. However, the content has been reworked and expanded here to address the unique characteristics of the C# language.

**Audience**

We wrote this book for anyone writing C# code, but especially for programmers who are writing C# as part of a team. For a team to be effective, everyone must be able to read and

---


understand everyone else’s code. Having consistent style conventions is a good first step!

This book is not intended to teach you C#, but rather it focuses on how C# code can be written in order to maximize its effectiveness. We therefore assume you are already familiar with C# and object-oriented programming.