

Grok Code Book

@GrokCodeBook

<http://twitter.com/#!/GrokCodeBook>

August 12, 2011

Contents

Contents	i
Introduction	1
1 Setting Up Your Programming Environment	3
1.1 DIY	3
1.2 List of Tools	4
1.2.1 Build Automation	4
1.2.2 Unit Testing	4
1.2.3 Continuous Integration	4
Glossary	5
Acronyms	7

Introduction

The solitary nature of much of computer programming and the number of different avenues that may lead to a career in programming make it difficult for best practices to permeate the field. A responsible author can no longer assume that a person seeking to learn about programming has any background in either computer science or mathematics. All jargon should be introduced in a glossary at minimum; regardless of how many works the author has authored. If a programming text is to build on some assumed knowledge, it must explicitly define said knowledge from the outset, and ought contain a “Read First” section.

Also, the volume of information forthcoming about new technologiesframeworks, toolkits, tooling, etc.drowns out information about programming well. By providing this text free of charge, holding ourselves to the highest standard of authorship and editorship, trying to make this text accessible to all that might read it, and acting as stewards and care-takers of this living document, we hope to raise the bar for published works pertaining to computer programming.

The examples included hereafter are executed in C# and Microsoft.Net, but the concepts set forth therein are relevant to all OO languages and frameworks.

Chapter 1

Setting Up Your Programming Environment

1.1 DIY

One click installers for all-in-one Integrated Development Environments (IDEs) can be nice, but they tend to encourage dependency on automata. They also promote, “Well, if there isn’t an installer for it, it can’t be any good,” thinking. It should be noted that often times creating an installer that works for Windows can be prohibitively expensive for authors of tools who do not use Windows as their primary operating system or Visual Studio as their primary development environment.

Free/Libre/Open Source Software (FLOSS) software will be used in the examples hereafter; not because of its price-point, but rather because of the ability of open source software to transfer knowledge to its end user. Want to know the best way to write a task for your build engine? Why not see how the authors of your build engine did it? In order to write code, you must learn how to read code. By extension, to write code well, you must learn how to read code well. It is important to figure out why a piece of code was written the way it was and not just settle for duplicating and successfully compiling it. Reading the code behind open source software is a cheap and easy way to learn from the masters. Not Ant (NAnt) contains some of the most well written C# code out there.

FLOSS software can be a little more difficult to set up, but doing so is worth the extra effort. There are many freely available tutorials online that will walk you through how to configure a software package for first-time use. In the process of setting up software, you will learn a little bit more about your computer runs programs in addition to making your computer a little bit more your own.

The biggest reason for configuring your own environment is personal taste. You may prefer typing text on the command line in a program like Vim¹ over

¹<http://www.vim.org/>

using an IDE.

When it comes to programmer tools, one size definitely does not fit all. Something like ReSharper² may give you unparalleled refactoring and auto-completion support, but starting Visual Studio to change a configuration setting or one line of code is overkill. One theme you should expect to encounter repeatedly in the passages that follow is, “Use the right tool for the job at hand.”

If all you have is a hammer, everything looks like a nail.
—Bernard Baruch

Equip yourself appropriately.

1.2 List of Tools

1.2.1 Build Automation

NAnt will be used in the examples that follow. MSBuild has some advantages in Windows-only shops—i.e. it is installed by default on Windows Server machines. At the time of writing, it more of a hassle to run all the unit tests for a project from the command -line in MSBuild than in NAnt. Also, there seems to be more of an air of acceptance towards customization—specifically writing one’s own build tasks—in the NAnt community.

While there is virtue in the philosophy, “Why write what you can download for free?” sometimes the needs of customization outweigh the benefits of using easily downloadable/purchased software. The philosophy set forth herein is, “Weigh all options without bias, and make the logical choice.” Developers are—or at least always should be—part of a team. Sometimes, for the sake of progress, a developer needs to make decisions that compromise on their own happiness. For instance, not using your favorite build tool on a fresh project in a new job, because all the other developers are invested in the use of another tool.

1.2.2 Unit Testing

1.2.3 Continuous Integration

²<http://www.jetbrains.com/resharper/>

Glossary

Free/Libre/Open Source Software “Free/Libre/Open-source software (FLOSS) is liberally licensed to grant the right of users to use, study, change, and improve its design through the availability of its source code”³.⁷

Integrated Development Environment “An integrated development environment (IDE) (also known as integrated design environment, integrated debugging environment or interactive development environment) is a software application that provides comprehensive facilities to computer programmers for software development”⁴.⁷

NAnt A FLOSS build automation tool written in C#⁵.⁷

auto-completion “A feature [of software programs] that suggests text automatically based on the first few characters that a user types.”⁶.⁴

refactoring “The process of modifying source code for the purpose of improving its readability and maintainability while retaining the program’s functionality and behavior.”⁷.⁴

³http://en.wikipedia.org/wiki/Free_and_open_source_software

⁴http://en.wikipedia.org/wiki/Integrated_development_environment

⁵<http://nant.sourceforge.net/>

⁶[http://msdn.microsoft.com/en-us/library/dd921717\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/dd921717(v=office.12).aspx)

⁷http://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/Xcode_Glossary/20-Glossary/Glossary.html

Acronyms

FLOSS Free/Libre/Open Source Software. 3, 5

IDE Integrated Development Environment. 3–5

NAnt Short for Not Ant⁸. 3–5

⁸<http://nant.sourceforge.net/release/latest/help/introduction/fog0000000042.html>