Table of Contents

cppbestpractices

Collaborative Collection of C++ Best Practices

This document is available as a download via gitbook

For more information please see the Preface.

This book has inspired an O'Reilly video: Learning C++ Best Practices

Preface

C++ Best Practices: A Forkable Coding Standards Document

This document is meant to be a collaborative discussion of the best practices in C++. It complements books such as *Effective C++* (Meyers) and *C++ Coding Standards* (Alexandrescu, Sutter). We fill in some of the lower level details that they don't discuss and provide specific stylistic recommendations while also discussing how to ensure overall code quality.

In all cases brevity and succinctness is preferred. Examples are preferred for making the case for why one option is preferred over another. If necessary, words will be used.

Disclaimer

This document is based on my personal experiences. You are not supposed to agree with it 100%. It exists as a book on GitHub so that you can fork it for your own uses or submit back proposed changes for everyone to share.

This book has inspired an O'Reilly video: Learning C++ Best Practices

Use The Tools Available

An automated framework for executing these tools should be established very early in the development process. It should not take more than 2-3 commands to checkout the source code, build, and execute the tests. Once the tests are done executing, you should have an almost complete picture of the state and quality of the code.

Source Control

Source control is an absolute necessity for any software development project. If you are not using one yet, start using one.

- GitHub - allows for unlimited public repositories, must pay for a private repository.
- Bitbucket - allows for unlimited private repositories with up to 5 collaborators, for free.
- SourceForge - open source hosting only.
- GitLab - allows for unlimited public and private repositories, unlimited CI Runners included, for free.
- Visual Studio Online (http://www.visualstudio.com/what-is-visual-studio-online-vs) - allows for unlimited public repositories, must pay for private repository. Repositories can be git or TFVC. Additionally: Issue tracking, project planning (multiple Agile templates, such as SCRUM), integrated hosted builds, integration of all this into Microsoft Visual Studio. Windows only.

Build Tool

Use an industry standard widely accepted build tool. This prevents you from reinventing the wheel whenever you discover / link to a new library / package your product / etc. Examples include:

- CMake
  - Consider: https://github.com/sakra/cotire/ for build performance
  - Consider: https://github.com/toeb/cmakepp for enhanced usability
- Conan - a crossplatform dependency manager for C++
- C++ Archive Network (CPPAN) - a crossplatform dependency manager for C++
- Waf
- FASTBuild
- Ninja - can greatly improve the incremental build time of your larger projects. Can be used as a target for CMake.

- Bazel - Note: MacOS and Linux only.
- gyp - Google's build tool for chromium.
- maiken - Crossplatform build tool with Maven-esque configuration style.
- Qt Build Suite - Crossplatform build tool From Qt.
- meson - Open source build system meant to be both extremely fast, and, even more importantly, as user friendly as possible.

Remember, it's not just a build tool, it's also a programming language. Try to maintain good clean build scripts and follow the recommended practices for the tool you are using.

Continuous Integration

Once you have picked your build tool, set up a continuous integration environment.

Continuous Integration (CI) tools automatically build the source code as changes are pushed to the repository. These can be hosted privately or with a CI host.

- Travis CI
  - works well with C++
  - designed for use with GitHub
  - free for public repositories on GitHub
- AppVeyor
  - supports Windows, MSVC and MinGW
  - free for public repositories on GitHub
- Hudson CI / Jenkins CI
  - Java Application Server is required
  - supports Windows, OS X, and Linux
  - extendable with a lot of plugins
- TeamCity
  - has a free option for open source projects
- Decent CI
  - simple ad-hoc continuous integration that posts results to GitHub
  - supports Windows, OS X, and Linux
  - used by ChaiScript
- Visual Studio Online (http://www.visualstudio.com/what-is-visual-studio-online-vs)
  - Tightly integrated with the source repositories from Visual Studio Online
  - Uses MSBuild (Visual Studio's build engine), which is available on Windows, OS X and Linux
  - Provides hosted build agents and also allows for user-provided build agents
  - Can be controlled and monitored from within Microsoft Visual Studio
  - On-Premise installation via Microsoft Team Foundation Server