BuiltSteady Engineering Guide

Contents

[Source Control 2](#_Toc313958492)

[Repositories 2](#_Toc313958493)

[Documentation 2](#_Toc313958494)

[Structure 2](#_Toc313958495)

[Source Code 3](#_Toc313958496)

[Structure 3](#_Toc313958497)

[Enlisting and Making Changes 4](#_Toc313958498)

[Naming Conventions 4](#_Toc313958499)

[Build Environment 4](#_Toc313958500)

[Build 4](#_Toc313958501)

[Deployment 4](#_Toc313958502)

[Test 4](#_Toc313958503)

# Source Control

## Repositories

All documentation and source code are stored in a GitHub repository.

Directions for setting up GitHub for Windows are located at: <http://help.github.com/win-set-up-git>

Directions for setting up GitHub for Mac OSX are located at: <http://help.github.com/mac-set-up-git>

## Documentation

Product documentation and corporate website are stored in a GitHub repository named builtsteady.

To enlist in the builtsteady repository, open a git bash command window and clone the repository to your local machine:

git clone git@github.com:ogazitt/builtsteady.git

### Structure

The builtsteady repository is divided into two primary directories; documents and corpweb.

The documents directory contains informational documentation for builtsteady corporation and the product. This includes legal documentation, marketing materials, presentations, videos, etc… This repository does not contain technical documentation, as that is stored in the zaplify repository with the source code. The documents tree is organized by area; legal, product, etc…

The corpweb directory contains the projects and source code for the corporate website.

## Source Code

The projects, source code, and technical documentation are stored in a GitHub repository named zaplify.

To enlist in the zaplify repository, open a git bash command window and clone the repository to your local machine:

git clone git@github.com:ogazitt/zaplify.git

### Structure

The zaplify repository is divided into four primary directories; source, test, tools, and documents.

The source directory contains the projects and code for the product. The source tree is organized by area: services, devices, packages, and shared. The packages directory contains external open-source code utilized by the product. The shared directory contains internally developed code utilized across areas of the product.

The test directory contains the projects and code for testing the product. The test tree is organized by area mirroring the source tree, with the addition of a clients directory for implementing test clients for various services.

The tools directory contains external and internal tools used to development, management, deployment, etc.. of the product.

The documents directory contains technical documentation pertaining to architecture, design, building and deploying the product, management of cloud services, test plans, etc...

## Enlisting and Making Changes

The following are the common git commands used for adding, removing, or updating files in a repository:

git add **filename** // add or update file in local changes

git add **directory** // add or update entire directory in local changes

git rm **filename** // remove file from local changes

git mv **filename** // rename file in local changes

git status // review local changes

git commit –m '**comment**' // commit local changes

git push // push committed changes to remote repository

## Naming Conventions

There are a few naming conventions that should be followed when adding new directories or files to the repository. External tools and packages are exempt from these conventions.

* Directory names **should** be lower-case and **must not** contain spaces. This is to eliminate confusion when interacting with GitHub which is case-sensitive.
* The names of files that contain code **should not** contain spaces. (includes code in source, test, and tools)
* The names of files that contain code **should** use pascal-casing. ( MyCodeModule.cs, Web.config, etc. )
* The names of files in a documents directory **may** contain spaces.

# Build Environment

# Build

# Deployment

# Test