MASS C++ Developer’s Guide

Updated March 14, 2020

# Setup

## Location and project setup

MASS C++ can be found on the DSLab computers named Hermes. Currently, these machines are running CENTOS and are the main nodes being used for development.

The master copy of MASS C++ is stored in bitbucket and should be cloned and branched as needed.

The codebase is broken up into several folders. The work\_ folders are currently unused and present only for historical reasons. Development occurs with git branches based on the source directory. The redhat and ubuntu folders contain Makefiles specific to those systems. The redhat folder is also currently unused.

## Getting a copy of the code for development

To get a local copy of MASS C++, git clone the repository on your machine as shown:

git clone https://bitbucket.org/mass\_library\_developers/mass\_cpp\_core.git

# Making Changes

## Making the change

If you are unfamiliar with Git, checkout the Git training files for more indepth discussion on how to use in in development. The training files can be found in the ~/Training/GitTraining folder.

MASS C++ is using the Git Flow model for updates. To add a feature, branch off of the develop branch. You can then push your branch to origin and test it on the Hermes machines.

Here is the basic workflow in the command line, which should be very close to any GUI Git application’s process.

git checkout –b <MY\_BRANCH\_NAME>

git add <CHANGED\_FILES>

git commit

git push origin <MY\_BRANCH\_NAME>

## Building the change

Currently there is one way to build your changes

### Hermes machines

To build a particular branch, you will need to clone the repository into a folder so you can checkout the correct branch you need to build.

For the Hermes machines, you can do the following:

ssh [<username>@hermes<#>.uwb.edu](mailto:dslab@uw1-320-lab.uwb.edu)

git clone https://bitbucket.org/mass\_library\_developers/mass\_cpp\_core.git

You will now be able to checkout your branch. To build it, enter the ubuntu directory and type

make

This will build the MASS library.

**Warning: This only builds your changes. Runtime errors will not be detected. Do not assume that you have not broken anything until you run your branch on one of the Hermes machines!**

**Note**: When building MASS C++, if errors occur regarding libssh2, libgcrypt, or OPENSSL, this is probably an issue with libssh2. Follow these steps to configure and reinstall this library.

2.2.2 Installing libssh2

Enter the ubuntu folder and use command **pwd** to get the current working directory.

Enter the libssh2-1.9.# folder and enter:

./configure --prefix=/<pwd\_output>/ssh2

make

make install

make distclean

This should compile the ssh2 library in the ssh2 folder locally for use by MASS C++. Try compiling MASS C++ again.

## Testing the change

To run MASS with a sample program, you will need to have setup a clone of the repository as shown in Method 2. In the ubuntu/Appl/<application\_name> folder, type

./compile.sh

./run.sh

This will run MASS with a test program so you can catch obvious runtime errors.

# Releasing a New Version

## Tagging in Git For Release

## Copy to the Release Directory