## Contributing

Before you can contribute, EA must have a Contributor License Agreement (CLA) on file that has been signed by each contributor.

You can sign here: [Go to CLA](https://electronicarts.na1.echosign.com/public/esignWidget?wid=CBFCIBAA3AAABLblqZhByHRvZqmltGtliuExmuV-WNzlaJGPhbSRg2ufuPsM3P0QmILZjLpkGslg24-UJtek\*)

### Pull Request Policy

All code contributions to EASTL are submitted as [Github pull requests](https://help.github.com/articles/using-pull-requests/). All pull requests will be reviewed by an EASTL maintainer according to the guidelines found in the next section.

Your pull request should:

\* merge cleanly

\* come with tests

\* tests should be minimal and stable

\* fail before your fix is applied

\* pass the test suite

\* code formatting is encoded in clang format

\* limit using clang format on new code

\* do not deviate from style already established in the files

### Getting the Repository

EASTL uses git submodules for its dependencies as they are seperate git repositories. Recursive clones will continue until HD space is exhausted unless they are manually limited.

It is recommended to use the following to get the source:

```bash

git clone https://github.com/electronicarts/EASTL

cd EASTL

git submodule update --init

```

### Running the Unit Tests

EASTL uses CMake as its build system.

\* Create and navigate to "your\_build\_folder":

\* mkdir your\_build\_folder && cd your\_build\_folder

\* Generate build scripts:

\* cmake eastl\_source\_folder -DEASTL\_BUILD\_TESTS:BOOL=ON

\* Build unit tests for "your\_config":

\* cmake --build . --config your\_config

\* Run the unit tests for "your\_config" from the test folder:

\* cd test && ctest -C your\_config

Here is an example batch file.

```batch

set build\_folder=out

mkdir %build\_folder%

pushd %build\_folder%

call cmake .. -DEASTL\_BUILD\_TESTS:BOOL=ON -DEASTL\_BUILD\_BENCHMARK:BOOL=OFF

call cmake --build . --config Release

call cmake --build . --config Debug

call cmake --build . --config RelWithDebInfo

call cmake --build . --config MinSizeRel

pushd test

call ctest -C Release

call ctest -C Debug

call ctest -C RelWithDebInfo

call ctest -C MinSizeRel

popd

popd

```

Here is an example bash file

```bash

build\_folder=out

mkdir $build\_folder

pushd $build\_folder

cmake .. -DEASTL\_BUILD\_TESTS:BOOL=ON -DEASTL\_BUILD\_BENCHMARK:BOOL=OFF

cmake --build . --config Release

cmake --build . --config Debug

cmake --build . --config RelWithDebInfo

cmake --build . --config MinSizeRel

pushd test

ctest -C Release

ctest -C Debug

ctest -C RelWithDebInfo

ctest -C MinSizeRel

popd

popd

```

The value of EASTL\_BUILD\_BENCHMARK can be toggled to `ON` in order to build projects that include the benchmark program.