# Contributing

Contributions are highly welcome, however, except for very small changes, kindly file an issue and let's have a discussion before you open a pull request.

## Requirements

.NET SDK 2.2 https://dotnet.microsoft.com/download/dotnet-core/2.2

.NET SDK 3.1 https://dotnet.microsoft.com/download/dotnet-core/3.1

## Building the Project

Clone this repo:

git clone https://github.com/tonerdo/coverlet.git

cd coverlet

Building, testing, and packing use all the standard dotnet commands:

dotnet restore

dotnet build --no-restore

dotnet pack

dotnet test --no-build /p:CollectCoverage=true /p:Include=\"[coverlet.collector]\*,[coverlet.core]\*,[coverlet.msbuild.tasks]\*\" /p:Exclude=\"[coverlet.core.tests.samples.netstandard]\*,[coverlet.tests.xunit.extensions]\*\"

NB. You need to `pack` before testing because we have some integration testing that consume packages

## Performance testing

There is a simple performance test for the hit counting instrumentation in the test project `coverlet.core.performancetest`. Build the project with the msbuild step above and then run:

dotnet test /p:CollectCoverage=true test/coverlet.core.performancetest/

The duration of the test can be tweaked by changing the number of iterations in the `[InlineData]` in the `PerformanceTest` class.

For more realistic testing it is recommended to try out any changes to the hit counting code paths on large, realistic projects. If you don't have any handy https://github.com/dotnet/corefx is an excellent candidate. [This page](https://github.com/dotnet/corefx/blob/master/Documentation/building/code-coverage.md) describes how to run code coverage tests for both the full solution and for individual projects with coverlet from nuget. Suitable projects (listed in order of escalating test durations):

\* System.Collections.Concurrent.Tests

\* System.Collections.Tests

\* System.Reflection.Metadata.Tests

\* System.Xml.Linq.Events.Tests

\* System.Runtime.Serialization.Formatters.Tests

Change to the directory of the library and run the msbuild code coverage command:

dotnet test /p:Coverage=true

To run with a development version of coverlet call `dotnet run` instead of the installed coverlet version, e.g.:

dotnet test /p:Coverage=true /p:CoverageExecutablePath="dotnet run -p C:\...\coverlet\src\coverlet.console\coverlet.console.csproj"