# Contributing

Contributing back to `Elasticsearch.Net` and `NEST` is very much appreciated.

Whether you [feel the need to change one character](https://github.com/elastic/elasticsearch-net/pull/536) or have a go at

[mapping new APIs](http://github.com/elastic/elasticsearch-net/pull/376), no pull request (PR) is too small or too big.

In fact many of our most awesome features/fixes have been provided to us by

[these wonderful folks](https://github.com/elastic/elasticsearch-net/graphs/contributors) to which we are forever indebted.

It's usually best to open an issue first to discuss a feature or bug, before opening a pull request. Doing so can save time and help further ascertain the crux of an issue.

## Sign the CLA

We do ask that you sign the [Elasticsearch CLA](https://www.elastic.co/contributor-agreement) before we can accept pull requests from you.

## Coding Styleguide

Please install the [Editorconfig vs extension](https://visualstudiogallery.msdn.microsoft.com/c8bccfe2-650c-4b42-bc5c-845e21f96328)

this will automatically switch to our indentation, whitespace, newlines settings while working on our project

\*\*while leaving your default settings intact\*\*.

In most cases we won't shun a PR just because it uses the wrong indentation settings, though it'll be \*\*very\*\* much appreciated if it is already done!

## Tests

PRs with tests are more likely to be reviewed faster because it makes the job or reviewing the PR much easier. That being said,

we respect that you may be fixing a bug in your own time and may not have the time/energy to submit a PR with complete tests.

In those cases we tend to pull your bits locally and write tests ourselves, but this may mean your PR might sit idle longer than you would like.

## Branches

Convention:

- `master` reflects the latest server version, this is typically the `current latest major + 1`

- `N.x` where N represents the major version component of the Elasticsearch server release its integrating with; e.g. `7.x`

- `N.Y` where `N` is the major version and `Y` is the minor component, typically opened as integration branch for a specific minor leaving `N.x` free to do bug fixes.

Examples:

- `master` for the latest server version (currently \_8.x\_)

- `7.x` for 7.x compatible client

- `6.x` for 6.x compatible client

- `5.x` for 5.x compatible client

- `2.x` for 2.x compatible client (no longer maintained)

- `1.x` for 1.x compatible client (no longer maintained)

## Git

We do not require rebased/squashed commits although we do very much appreciate it!

Please submit your [Pull Requests](https://help.github.com/articles/creating-a-pull-request/) to

- [`master`](https://github.com/elastic/elasticsearch-net/tree/master) branch for master

- [`7.x`](https://github.com/elastic/elasticsearch-net/tree/7.x) branch for 7.x

- [`6.x`](https://github.com/elastic/elasticsearch-net/tree/6.x) branch for 6.x

- [`5.x`](https://github.com/elastic/elasticsearch-net/tree/5.x) branch for 5.x

# Building the solution

The solution uses a number of awesome Open Source software tools to ease development:

## Bullseye

[Bullseye](https://github.com/adamralph/bullseye) is used as the build automation system for the solution. To get started after cloning the solution, it's best to run the build script in the root

for Windows

```

.\build.bat

```

for OSX/Linux

```

./build.sh

```

This will

- Pull down all the dependencies for the build process as well as the solution

- Run the default build target for the solution

You can also compile the solution within Visual Studio if you prefer, but the build script is going to be \_much\_ faster.

## Tests

The `Tests` project contains both xunit unit and integration tests. A `tests.yaml` file within the root of the `Tests` project determines the test mode when running tests inside Visual Studio

- `u` for unit tests

- `i` for integration tests

- `m` for mixed mode i.e. unit and integration tests

The build script has a number of different build targets to run different types of tests, see the [`Targets.fs` file in the `scripts` project](https://github.com/elastic/elasticsearch-net/blob/master/build/scripts/Targets.fs) for the complete list, but the main ones are:

### Compile and run unit tests

```bat

.\build.bat

```

with no target will run the `Build` target, compiling the solution and running unit tests

### Compile

```bat

.\build.bat skiptests

```

This compiles the solution and skips running tests

### Quick Compile and run integration tests

```bat

.\build.bat integrate [Elasticsearch Version Number e.g. 5.0.0]

```

will quick compile the solution and run integration tests against the target Elasticsearch version. The first time this is run for a version of Elasticsearch, it will download Elasticsearch and unzip Elasticsearch, install the plugins necessary to run the integration tests, and start the node. Because of this, the first run may take some time to start.

## Troubleshooting

### Could not load file or assembly FSharp.Core

You may come across an exception similar to below when running the build script

>Unhandled Exception: System.IO.FileLoadException: Could not load file or assembly 'FSharp.Core, Version=4.3.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. The located assembly's manifest definition does not match the assembly reference. (Exception from HRESULT: 0x80131040)

?? at <StartupCode$FAKE>.$Program.main@()

The `1.x` and `master` branches have diverged dramatically as a result of changes in preparation for 2.0. This includes changes to the build process such that switching between the `master` and `1.x` branches and back again can change the versions of packages used within the build processes. To rectify this issue, try deleting the `packages` folder within the root of the solution and run the build script again.

If working on both 1.x and 2.x and 5.x versions of NEST, it is recommended to clone the git repository for each version into separate directories to avoid the need to switch between the divergent branches.