# Contributor Guide

Contributions to NRules, whether new features or bug fixes, are welcome and are greatly appreciated.

\*\*Your contributions must be your own work and licensed under the same terms as NRules project\*\*

## Process

- \*\*File an issue.\*\* Either suggest a feature or note a defect. If it's a new feature, explain the motivation behind it and how you think it should work. If it's a defect, include a description and reproduction (ideally one or more failing unit tests).

- \*\*Design discussion.\*\* For new features, some discussion on the issue will take place to determine if it's something that should be included with NRules. For defects, discussion may happen around whether the issue is truly a defect or if the behavior is correct.

- \*\*Implementation.\*\* Fork the repository, implement and test the change.

- \*\*Pull request.\*\* Create a pull request on the `develop` branch of the repository to submit changes. Pull requests need to pass the CI build and follow coding standards. All pull requests should include accompanying unit tests to verify the work.

- \*\*Code review.\*\* Some iteration may take place requiring updates to the pull request.

- \*\*Pull request acceptance.\*\* The pull request will be accepted into the `develop` branch and pushed to `master` with the next release.

### Questions

If you just want to ask a question about how to use NRules, instead of creating an issue consider asking your question on [Stack Overflow](http://stackoverflow.com/questions/tagged/nrules), [Discussion Group](http://groups.google.com/group/nrules-users) or [Gitter Chat](https://gitter.im/NRules/NRules).

## License

By contributing to NRules, you assert that:

1. The contribution is your own original work.

2. You have the right to assign the \*copyright\* for the work (it is not owned by your employer, or you have been given copyright assignment in writing).

3. You license it under the terms applied to the rest of the NRules project.

## Development Guidelines

### Development Environment

- Visual Studio 2019 (with latest patches/updates).

- [Sandcastle Help File Builder](https://github.com/EWSoftware/SHFB)

### Dependencies

NRules core assemblies depend \*only\* on the .NET Base Class Library (BCL). It should be possible to build the project straight out of Git (no additional installation needs to take place on the developer's machine). This means NuGet package references (which should be restored during the build process).

Integration and extension projects may have additional dependencies, consistent with the nature of those projects.

Unit tests are written in xUnit and Moq.

### Coding Standards

- NRules uses standard Microsoft .NET coding guidelines. See the [Framework Design Guidelines](https://msdn.microsoft.com/en-us/library/ms229042.aspx) for suggestions.

- Use four spaces for code indentation (no tabs).

- For private fields, use a camelCase notation, prefixing field names with an underscore.

- Generally, use comment-free coding approach, where code is self-explanatory. Comments may be added for complex areas of code, and should explain "why" not "how".

- Maintain clean code - no commented out sections, no dangling or inconsistent line breaks, ensure consistent indentation and bracing.

- Treat compiler warnings as errors.

- If you have ReSharper, make sure to address all warnings and errors that it produces.

- Be cognizant of performance-critical code.

### Documentation

Public types and members must have XML documentation.