# Contributing to Roslyn

Guidelines for contributing to the Roslyn repo.

## Submitting Pull Requests

- \*\*DO\*\* ensure submissions pass all Jenkins legs and are merge conflict free.

- \*\*DO\*\* follow the [.editorconfig](http://editorconfig.org/) settings for each directory.

- \*\*DO\*\* submit language feature requests as issues in the [C# language](https://github.com/dotnet/csharplang#discussion) / [VB language](https://github.com/dotnet/vblang) repos. Once a feature is championed and validated by LDM, a developer will be assigned to help begin a prototype on this repo inside a feature branch.

- \*\*DO NOT\*\* submit language features as PRs to this repo first, or they will likely be declined.

- \*\*DO\*\* submit issues for other features. This facilitates discussion of a feature separately from its implementation, and increases the acceptance rates for pull requests.

- \*\*DO NOT\*\* submit large code formatting changes without discussing with the team first.

When you are ready to proceed with making a change, get set up to build (either on [Windows](https://github.com/dotnet/roslyn/blob/master/docs/contributing/Building%2C%20Debugging%2C%20and%20Testing%20on%20Windows.md) or on [Unix](https://github.com/dotnet/roslyn/blob/master/docs/contributing/Building%2C%20Debugging%2C%20and%20Testing%20on%20Unix.md)) the code and familiarize yourself with our developer workflow.

These two blogs posts on contributing code to open source projects are good too: [Open Source Contribution Etiquette](http://tirania.org/blog/archive/2010/Dec-31.html) by Miguel de Icaza and [Don’t “Push” Your Pull Requests](https://www.igvita.com/2011/12/19/dont-push-your-pull-requests/) by Ilya Grigorik.

## Creating Issues

- \*\*DO\*\* use a descriptive title that identifies the issue to be addressed or the requested feature. For example, when describing an issue where the compiler is not behaving as expected, write your bug title in terms of what the compiler should do rather than what it is doing – “C# compiler should report CS1234 when Xyz is used in Abcd.”

- \*\*DO\*\* specify a detailed description of the issue or requested feature.

- \*\*DO\*\* provide the following for bug reports

- Describe the expected behavior and the actual behavior. If it is not self-evident such as in the case of a crash, provide an explanation for why the expected behavior is expected.

- Provide example code that reproduces the issue.

- Specify any relevant exception messages and stack traces.

- \*\*DO\*\* subscribe to notifications for the created issue in case there are any follow up questions.

## Coding Style

The Roslyn project is a member of the [.NET Foundation](https://github.com/orgs/dotnet) and follows the same [developer guide](https://github.com/dotnet/runtime/blob/master/docs/coding-guidelines/coding-style.md). The repo also includes [.editorconfig](http://editorconfig.org) files to help enforce this convention. Contributors should ensure they follow these guidelines when making submissions.

### CSharp

- \*\*DO\*\* use the coding style outlined in the [.NET Runtime Coding Guidelines](https://github.com/dotnet/runtime/blob/master/docs/coding-guidelines/coding-style.md)

- \*\*DO\*\* use plain code to validate parameters at public boundaries. Do not use Contracts or magic helpers.

```csharp

if (argument == null)

{

throw new ArgumentNullException(nameof(argument));

}

```

- \*\*DO\*\* use `Debug.Assert()` for checks not needed in release builds. Always include a “message” string in your assert to identify failure conditions. Add assertions to document assumptions on non-local program state or parameter values, e.g. “At this point in parsing the scanner should have been advanced to a ‘.’ token by the caller”.

- \*\*DO\*\* avoid allocations in compiler hot paths:

- Avoid LINQ.

- Avoid using `foreach` over collections that do not have a `struct` enumerator.

- Consider using an object pool. There are many usages of object pools in the compiler to see an example.

### Visual Basic Conventions

- \*\*DO\*\* apply the spirit of C# guidelines to Visual Basic when there are natural analogs.

- \*\*DO\*\* place all field declarations at the beginning of a type definition

### Tips 'n' Tricks

Our team finds using [this enhanced source view](http://sourceroslyn.io/) of Roslyn helpful when developing.