# Contributing

Discord.Net is an open-source project, and we appreciate any and all

contributions made by our community. However, please conform to the

following guidelines when possible:

## Development Cycle

We prefer all changes to the library to be discussed beforehand,

either in a GitHub issue, or in a discussion in our Discord channel

with library regulars or other contributors.

Issues that are tagged as "up for grabs" are free to be picked up by

any member of the community.

### Pull Requests

We prefer pull-requests that are descriptive of the changes being made

and highlight any potential benefits/drawbacks of the change, but these

types of write-ups are not required. See this [merge request](https://github.com/RogueException/Discord.Net/pull/793)

for an example of a well-written description.

## Semantic Versioning

This project follows [Semantic Versioning](http://semver.org/). When

writing changes to this project, it is recommended to write changes

that are SemVer compliant with the latest version of the library in

development.

The working release should be the latest build off of the `dev` branch,

but can also be found on the [development board](https://github.com/RogueException/Discord.Net/projects/1).

We follow the .NET Foundation's [Breaking Change Rules](https://github.com/dotnet/corefx/blob/master/Documentation/coding-guidelines/breaking-change-rules.md)

when determining the SemVer compliance of a change.

Obsoleting a method is considered a \*\*minor\*\* increment.

## Coding Style

We attempt to conform to the .NET Foundation's [Coding Style](https://github.com/dotnet/corefx/blob/master/Documentation/coding-guidelines/coding-style.md)

where possible.

As a general rule, follow the coding style already set in the file you

are editing, or look at a similar file if you are adding a new one.

### Documentation Style for Members

When creating a new public member, the member must be annotated with sufficient documentation. This should include the

following, but not limited to:

\* `<summary>` summarizing the purpose of the method.

\* `<param>` or `<typeparam>` explaining the parameter.

\* `<return>` explaining the type of the returned member and what it is.

\* `<exception>` if the method directly throws an exception.

The length of the documentation should also follow the ruler as suggested by our

[Visual Studio Code workspace](Discord.Net.code-workspace).

#### Recommended Reads

\* [Official Microsoft Documentation](https://docs.microsoft.com)

\* [Sandcastle User Manual](https://ewsoftware.github.io/XMLCommentsGuide/html/4268757F-CE8D-4E6D-8502-4F7F2E22DDA3.htm)