# Contributing Code

ChakraCore accepts bug fix pull requests. For a bug fix PR to be accepted, it must first have a tracking issue that has been marked approved. Your PR should link to the bug you are fixing. If you've submitted a PR for a bug, please post a comment in the bug to avoid duplication of effort.

ChakraCore also accepts new feature pull requests. For a feature-level PR to be accepted, it first needs to have design discussion. Design discussion can take one of two forms a) a feature request in the issue tracker that has been marked as approved or b) the PR must be accompanied by a full design spec and this spec is later approved in the open design discussion. Features are evaluated against their complexity, impact on other features, roadmap alignment, and maintainability.

These two blogs posts on contributing code to open source projects are a good reference: [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.

## Security

If you believe you have found a security issue in ChakraCore, please share it with us privately following the guidance at the Microsoft [Security TechCenter](https://www.microsoft.com/en-us/msrc/faqs-report-an-issue). Reporting it via this channel helps minimize risk to projects built with ChakraCore.

## Legal

You will need to complete a Contributor License Agreement (CLA) before your pull request can be accepted. This agreement testifies that you are granting us permission to use the source code you are submitting, and that this work is being submitted under appropriate license that we can use it.

You can complete the CLA by going through the steps at https://cla.microsoft.com. Once we have received the signed CLA, we'll review the request. You will only need to do this once.

## Housekeeping

Your pull request should:

\* Include a description of what your change intends to do

\* Be a child commit of a reasonably recent commit in the master branch

\* Pass all unit tests

\* Have a clear commit message

\* Include adequate tests

\* At least one test should fail in the absence of your non-test code changes. If your PR does not match this criteria, please specify why

\* Tests should include reasonable permutations of the target fix/change

\* Include baseline changes with your change

Submissions that have met these requirements will be assigned to a ChakraCore team member for additional testing. Submissions must meet functional and performance expectations, including meeting requirements in scenarios for which the team doesn’t yet have open source tests. This means you may be asked to fix and resubmit your pull request against a new open test case if it fails one of these tests. The ChakraCore team may verify your change by crawling the web with your change built into Chakra. Failures discovered when testing with this technique will not be analyzed by the team, but we will do our best to communicate the issue discovered to you. This approach needs further refinement, we acknowledge.

ChakraCore is an organically grown codebase. The consistency of style reflects this. For the most part, the team follows these [coding conventions](https://github.com/Microsoft/ChakraCore/wiki/Coding-Convention). Contributors should also follow them when making submissions. Otherwise, follow the general coding conventions adhered to in the code surrounding your changes. Pull requests that reformat the code will not be accepted.

## Running the tests

The unit tests can be run by following these steps:

\* Choose a build configuration to build and test, e.g. debug and x64.

\* Build `Chakra.Core.sln` for that config.

\* Specifically, running tests requires that `rl.exe`, `ch.exe`, and `ChakraCore.dll` be built.

\* Call `test\runtests.cmd` and specify the build config

e.g. `test\runtests.cmd -x64debug`

For full coverage, please run unit tests against debug and test for both x86 and x64:

\* `test\runtests.cmd -x64debug`

\* `test\runtests.cmd -x64test`

\* `test\runtests.cmd -x86debug`

\* `test\runtests.cmd -x86test`

`runtests.cmd` can take more switches that are useful for running a subset of tests. Read the script file for more information.

`runtests.cmd` looks for the build output in the default build output folder `Build\VcBuild\bin`. If the build output path is changed from this default then use the `-bindir` switch to specify that path.

## Code Flow into Microsoft Edge

Changes that make it into our ChakraCore GitHub master branch have a short journey to Chakra.dll. Code flows daily from GitHub to the internal repository from which builds of Chakra.dll are produced and then it flows into Windows and Microsoft Edge. While code flows quickly on this first leg of the journey, code flow from our internal branch to a Windows flighting branch is subject to any number of delays. So it is difficult to predict when your change in our GitHub repo will make it into a particular Windows flight.

## Issue Labels

- [`help wanted`](https://github.com/Microsoft/ChakraCore/labels/help%20wanted): these issues are specifically well suited for outside contributors.

- [`good first issue`](https://github.com/Microsoft/ChakraCore/labels/good%20first%20issue): these issues are small and appropriate for people who wish to familiarize themselves with GitHub pull requests and/or ChakraCore's contributor guidelines, build process, and running tests. We're here to help you get started in open source.

You are welcome to work on issues that are not tagged with these labels. However, issues without these labels are often deeply involved with the requirements of the various components of ChakraCore. Therefore, please be sure to touch base with a maintainer via comments on the issue before attempting to solve it.

Remember, for all issues you choose to work on please communicate on the issue that you are claiming it to avoid duplicated work.

To learn more about our GitHub labels see the [Label Glossary](https://github.com/Microsoft/ChakraCore/wiki/Label-Glossary) on our wiki page.