## CefSharp Project Contribution Guide

Thanks for your interest in contributing to the project! Please follow these simple guidelines:

### General

- Please use [![Gitter](https://badges.gitter.im/Join%20Chat.svg)](https://gitter.im/cefsharp/CefSharp?utm\_source=badge&utm\_medium=badge&utm\_campaign=pr-badge) as your first point of call for basic/simple questions.

- When creating an issue please use the issue template provided.

- \*\*Please read the full contents of [the FAQ](https://github.com/cefsharp/CefSharp/wiki/Frequently-asked-questions) before submitting an issue\*\*. Also searching the `GitHub` project is a \*\*must\*\*. It's quite likely your question has already been answered before. If something is unclear in the FAQ, of course feel free to ask; the idea is just to reduce the level of "noise" we have to go through, reading the same questions over and over again.

- Please make sure to \*\*test out the current version\*\* of `CefSharp` to see whether the problem you are encountering still exists.

- Don't cross-post: if you create an issue, and all the information is contained there, that's enough. There's no reason to also post it to Stackoverflow; it just creates "line noise". The project maintainers are very busy people like you and me, and things will sometimes take a few weeks (or in worst case, more) to answer. If you are in a rush - do your very best to investigate the problem thoroughly; if possible, fix the bug yourself and submit a pull request.

- Before creating a GitHub issue or pull request, try looking through the list & issue archives to make sure the issue at hand hasn't been raised before. [Google](http://www.google.com) can also be helpful: just typing "cefsharp appdomain" for example (when trying to see whether AppDomain support has been discussed before) will often give you helpful results.

- We do appreciate cultural/languages differences, that being said \*\*never\*\* demand that someone help you, this is not a commercial application with paid support! Please and thank you go a long way.

- This is a volunteer project, we give of our time freely and we ask for you to do the same. Contributions can be simple like updating/adding new entries in the [FAQ](https://github.com/cefsharp/CefSharp/wiki/Frequently-asked-questions), creating new pages in the [WIKI](https://github.com/cefsharp/CefSharp/wiki), updating the examples. (Anyone with a `GitHub` account can edit the `WIKI`)

### `CefSharp` vs `Chromium Embedded Framework(CEF)`

`CefSharp` simply takes the building blocks provided by `CEF` and attempts to provide a usable '.Net' implementation.

The upstream [`CEF` forum](http://magpcss.org/ceforum/) is a valuable resource, if your issues seems fairly low level, then please conduct some research before posting.

It maybe helpful to run the `cefclient` application and compare output with `CefSharp`. The `WinForms` and `WPF` versions use two different rendering modes, `WPF` uses Offscreen Rendering (`OSR`). `OffScreen` also uses `OSR` mode.

- Download one of the following:

- For x86 download http://opensource.spotify.com/cefbuilds/cef\_binary\_81.3.10%2Bgb223419%2Bchromium-81.0.4044.138\_windows32\_client.tar.bz2

- For x64 download http://opensource.spotify.com/cefbuilds/cef\_binary\_81.3.10%2Bgb223419%2Bchromium-81.0.4044.138\_windows64\_client.tar.bz2

- Extract and run cefclient.exe

- If you are using WPF/OffScreen run

```

cefclient.exe --multi-threaded-message-loop --off-screen-rendering-enabled --enable-gpu --no-sandbox --disable-site-isolation-trials

```

- If you are using WinForms run

```

cefclient.exe --multi-threaded-message-loop --no-sandbox --disable-site-isolation-trials

```

- \*\*MAKE SURE TO TEST WITH THE COMMAND LINE ARGS LISTED ABOVE\*\*

- If you can reproduce the problem with `cefclient` then you'll need to report the bug on https://bitbucket.org/chromiumembedded/cef/overview there is no point opening an issue here. (Make sure you search before opening an issue)

### What should I include when creating an `Issue`?

When creating an issue please use the provided by report template (the field will be pre-populated).

A bug is a \_demonstrable problem\_ that is caused by the code in the repository. Ideally each issue is a useful resource for references purposes (Don't take offence if someone edits your description).

Your bug report should \*\*always follow this template\*\*:

- \*\*What is the expected output? What do you see instead?\*\*

- \*\*What steps will reproduce the problem?\*\*

- \*\*Are you using `WinForms`, `WPF` or `OffScreen`?\*\*

- \*\*What version of the product are you using? On what operating system? x86 or x64?\*\*

- What version are you using? Nuget? CI Nuget? build from a branch? If so which branch?

- Win7, Win 8, Win10, etc?

- \*\*Please provide any additional information below.\*\*

- A stack trace if available, any Exception information.

- Does the cef log provide any relevant information? (By default there should be a debug.log file in your bin directory)

- Any other background information that's relevant? Are you doing something out of the ordinary? 3rd party controls?

- \*\*Does this problem also occur in the `CEF Sample Application` from http://opensource.spotify.com/cefbuilds/index.html?\*\*

Your bug report should include \*\*what you were doing\*\* in the software when you encountered it, \*\*what you were expecting\*\* to happen and \*\*what happened instead\*\*.

\*\*BE AWARE THAT BUG REPORTS MUST PROVIDE ALL OF THE INFORMATION STATED ABOVE!\*\*

#### Help us help you

1. Asking the same questions over and over again is \*\*Wasting our time\*\*, search open/closed issues to see if your issue has already been addressed.

2. Having to constantly query users to gather information is \*\*very frustrating\*\*! Use the bug template provided above.

3. We have \*\*very\*\* limited active contributors so please spend as much time looking into your own problem as possible, the more you help yourself, the quicker things will get fixed.

4. Please don't hijack issues, if your problem is distinct then please create a unique issue (after searching previous issues).

5. Good bug reports are extremely helpful. The more information you provide, the more likely your issue will be resolved.

6. Good bug reports shouldn't leave others needing to chase you up for more information. Be sure to include the

details of your environment.

7. 'Ask Don't Tell' : Ask how to achieve something, \*\*don't say it's broken\*\* just because you haven't got it working yet!

8. \*\*Isolate the problem\*\* &mdash; ideally create a reproducible example.

9. \*\*Include a screencast if relevant\*\* - Is your issue about a design or front end feature or bug? The most helpful thing in the world is if we can \*see\* what you're talking about. Use [LICEcap](http://www.cockos.com/licecap/) to quickly and easily record a short screencast (24fps) and save it as an animated gif! Embed it directly into your GitHub issue.

10. When including code limit to small chunks, large blocks post as a [Gist](http://gist.github.com/) or similar

11. Please no links to binaries, e.g. zip files, either contribute your example as a github project, a gist or another public code sharing service.

### Change Requests

Change requests cover both architectural and functional changes to how `CefSharp` works. If you have an idea for a refactor, or an improvement to a feature, etc - please be sure to:

1. \*\*Use the GitHub search\*\* and check someone else didn't get there first

2. Take a moment to think about the best way to make a case for, and explain what you're thinking as it's up to you to convince the project's leaders the change is worthwhile. Some questions to consider are:

- Is it really one idea or is it many?

- What problem are you solving?

- Why is what you are suggesting better than what's already there?

### Pull Requests/Feature Branches

Pull requests are \*\*awesome\*\*. If you're looking to raise a `PR` for something which doesn't have an open issue, please think carefully about raising an issue which your PR can close, especially if you're fixing a bug. This makes it more likely that there will be enough information available for your PR to be properly tested and merged.

- Please limit your changes to a logical grouping, keeping changesets small increases the likelihood they will be merged

- If you then want to make subsequent changes, it's actually best to not do them before the feature is merged. Please wait for feedback/review before progressing. This greatly improves our ability to review your changes and dramatically increases the likelihood they will be merged in a timely fashion.

- If you wish to keep progressing on your work, please maintain a feature branch independent of the branch referenced by your pull request. From your WIP branch you can selectively merge in changes to the PR branch as required.

- In general, it's much better to be "too granular" with PR's that contain "change-the-world"-kind of changes, which usually tend to lag behind a lot longer before getting merged (if they ever will...).

- Small (really, minimalistic) commits. Each individual commit only adds one specific thing. The basic approach to achieving this is to read your commit message. Do you feel the need to add multiple lines? Then you're doing too much at the same time.

- Keep the first line of commit message short (max 50 characters), then a blank line, then a full description if needed. Good example:

> Run tests in the default appdomain

>

> Due to limitations of VC++ we must run our tests in the default appdomain, it's possible to use xunit for testing. Re-add CefSharp.Test.csproj to solution Add some very basic unit tests using xunit, DisableTestParallelization, Set xunit.appDomain to denied in app.config to disable using appdomains (https://xunit.github.io/docs/configuring-with-xml.html) Add xunit vs test running packages, using 2.2.0 version as 2.3.0 was throwing an exception similar to https://github.com/xunit/xunit/issues/1514 TODO: Take a look at https://github.com/NuGet/NuGetGallery/tree/master/tests/NuGetGallery.Facts for some ideas on structuring xunit tests (look at implementing Theory)

### Coding Style

- \*\*Please\*\* follow existing coding style when submitting pull requests.

- Most importantly, \*\*use spaces\*\* - do not use tabs (which is the default setting for C++ projects in Visual Studio).

- The "preview" function when creating a Pull Request on GitHub is very useful for identifying whitespace issues (and for checking out the impact of your changes) - please use it.

- Max number of columns are 132, please format your code accordingly

- Please don't add #Regions to the code

- When adding new files, please prepend the standard license disclaimer (just copy and paste from another source file)

- Do your best to follow these guidelines but don't be afraid to make mistakes when trying to apply them. We are all novices in the beginning.