# Contribute to the Win2D Project

Thank you for your interest in the Win2D project!

\* [Before we can accept your pull request](#before-we-can-accept-your-pull-request)

\* [Source Directory Structure](#source-directory-structure)

\* [Contribute](#contribute)

## Before we can accept your pull request

Microsoft asks that all contributors sign a contributor license agreement (CLA).

CLAs are generally common and accepted in most open source software projects.

We all want Microsoft's open source projects to be as widely used and

distributed as possible. We also want its users to be confident about the

origins and continuing existence of the code. The CLA helps us achieve that

goal by ensuring that we have the agreement of our contributors to use their

work, whether it be code, or documentation.

The CLA permits Microsoft to distribute your code without restriction. It

doesn't require you to assign to us any copyright you have, the ownership of

the copyright remains with you. You cannot withdraw permission for use of the

contribution at a later date.

We are generally seeking originally authored code and documentation as

contributions. Should you wish to submit materials that are not your original

work, you may submit them separately to the Project in accordance with the terms

of the CLA.

Our Azure Pull Request Bot will automatically check for a signed CLA when you

submit a pull request as described below in [Contribute](#contribute).

If there isn't a CLA on file, it will walk you through an all electronic process.

\*\*Note\*\*: your employer may also have to complete an on-line form.

If you would rather use an off-line process, email

[win2dcla@microsoft.com](mailto:win2dcla@microsoft.com) and we will send you the

form to sign, scan and return.

## Source Directory Structure

Note: For the bulk of interesting product code, go to /winrt/lib.

\*\*/bin\*\* - Binary files generated by the build

\*\*/build\*\* - Shared, build-related resources that are common across projects

- /Assets - Common Windows Store assets used for tests and samples

- /nuget - Resources needed for building the NuGet package

\*\*/obj\*\* - Intermediate files generated by the build

\*\*/samples\*\* - Example projects showing how to use Win2D

\*\*/tools\*\* - Standalone utilities, separate from Win2D product code

- /checkcode - Verifies the right copyright banner appears at the top of each source file

- /codegen - Generates headers+IDL from XML resource files (you don't have to build this to build Win2D)

- /docs - Tools and scripts for building API reference documentation

\*\*/winrt\*\* - Win2D product code and automated tests

- /dll - Build resources for the product dll

- /docsrc - Source files used to create the reference documentation

- /inc - Common includes

- /lib - All the projected IDL and runtime class implementations

- /published - Headers that are included directly in the NuGet package for 3rd part consumption

- /test.external - Automated tests that use only the public WinRT interface

- /test.internal - Automated tests that link directly with winrt.lib to access internals of the implementation

- /test.managed - Automated tests written in C#

- /test.nativecomponent - C++/CX component that exposes native functionality for use by test.managed

## Contribute

In order for the contribution process to be as seamless as possible, the

following procedure has been established.

1. Create a new branch

2. Add new content or edit existing content

3. Verify your changes

4. Submit a pull request to the main repository

5. Delete the branch

Each branch should be limited to a single feature/bug fix both to streamline

work flow and reduce the possibility of merge conflicts.

#### Create a new branch

This can be done in Visual Studio:

- Go to 'View' -> 'Team Explorer' -> 'Branches'

- Click 'New Branch'

- Enter the name of your new branch

- Check the check box 'Checkout branch'

- Click 'Create Branch'

#### Add new content or edit existing content

Using Visual Studio or your favorite editor, add your new content or edit

existing files.

You can commit multiple times while you are doing your work, or you can wait

and commit only once when you're done.

#### Verify your changes

If you have added new functionality please make sure to add tests to the

appropriate test suites.

To validate your changes, run "msbuild /m /nr:false Win2D.proj" from a

Visual Studio Developer Command Prompt. This will:

- Build all flavors of all projects (debug, release, x86, x64, ARM, and ARM64)

- Run all automated tests

- Generate updated reference documentation

- Make sure all source files include the correct MIT copyright header

To run Store unit tests, you must have a Windows Store development license

installed on your computer, and must install the test package certificate from

build\Assets\TemporaryKey.pfx in your Win2D enlistment:

1. Right-click on the .pfx file, and select \*\*Install PFX\*\*

2. Choose \*\*Local Machine\*\*, then click \*\*Next\*\* three times (leaving the password blank)

3. Select \*\*Place all certificates in the following store\*\*

4. Click the \*\*Browse\*\* button, and select \*\*Trusted People\*\*

5. Click \*\*OK\*\*, \*\*Next\*\*, and then \*\*Finish\*\*

#### Submit a pull request to the main repository

When you are done with your work and are ready to have it merged into the central

repository follow these steps.

1. Push your branch back to GitHub

2. On the GitHub site, navigate in your fork to the new branch

3. Click the \*\*Pull Request\*\* button at the top of the page

4. Ensure that the Base branch is 'Win2D@master' and the Head branch is

'<your username>/Win2D@<branch name>'

5. Click the \*\*Update Commit Range\*\* button

6. Give your pull request a Title, and describe all the changes being made.

If your change fixes a GitHub issue make sure to reference it in the description.

7. Submit the Pull Request

One of the site administrators will now process your pull request. Your pull

request will surface on the Win2D site under Issues. When the Pull Request is

accepted, the issue will be resolved.

#### Delete a branch

Once your changes have been successfully merged into the central repository you

can delete the branch you used, as you will no longer need it. Any further work

requires a new branch.

Congratulations, you have successfully contributed to the project.