

Windows Desktop Client

- [Utopia Windows Desktop Client Update Creation Process](#)
- [Update CefSharp to 64 bit \(when we want to do it\)](#)
- [Client Application Update Process](#)
- [Code Scanning](#)
- [How To Update CefSharp Version](#)

Utopia Windows Desktop Client Update Creation Process

To Create an update to the Utopia Windows Desktop Client, you will do the following steps:

1. Right-click the UtopiaWindowsDesktopClient project in Visual Studio and select Properties.
2. Under the Application tab on the left side, click the "Assembly Information" button.
3. Change both the Assembly version and the File version. (Yes, you must do both!)
4. Click Ok, Save All.
5. Select the UtopiaWindowsDesktopClientSetup and go to the properties window.
6. Change the "Version" property to match the version set in Assembly Information.
7. Save All, this will prompt you to automatically change the Product Code, Accept this. (We want it to change the Product Code. NEVER Change the Upgrade Code)
8. To create a new exe:
 1. Clean and rebuild the solution in Debug
 2. Switch to Release mode
 3. Right click UtopiaWindowsDesktopClientSetup and Rebuild or Build
9. Take the resulting "RubexInstaller.exe", found in UtopiaClientApplications\UtopiaWindowsDesktopClient\UtopiaWindowsDesktopClientSetup\Release within your local repo, and copy it to the Utopia Web Project's ClientApplicationInstallers/Windows folder. (Replacing the existing installer)
10. Edit the WindowsClientVersion.json file in the given folder, change the Version property to the correct value.

Congratulations, You've updated the Windows Desktop Client!

- Note: You may also have to update the file versions in the other projects if they change. (i.e. UtopiaWindowsDesktopClientShared)

Update CefSharp to 64 bit (when we want to do it)

When updating the build of this application, you must change the version number in the Assembly Information of the UtopiaDesktopClientApplication AND the UtopiaWindowsDesktopClientSetup version.

Most of this should build and run out of the box. However, there are some CefSharp caveats, first and foremost, CefSharp cares about x86 vs x64. Everything is currently targeting x86, but if and when the day comes that we do a x64 version, here are some of the changes needed (May want to consider making x86 and x64 versions with a shared project, same with installer):

1. Change the targeting to x64 in the configuration manager.
2. You may have to change the references in the UtopiaWindowsDesktopClient project to target the x64 versions of the cefsharp dlls (CefSharp, CefSharp.Core, CefSharp.Wpf)
3. The following files directly referenced in the setup project will need to be pointed at the x64 versions:

- a. CefSharp.BrowserSubprocess.Core.dll
- b. CefSharp.BrowserSubprocess.exe
- c. cef.pak
- d. cef_100_percent.pak
- e. cef_200_percent.pak
- f. cef_extensions.pak
- g. chrome_elf.dll
- h. d3dcompiler_47.dll
- i. icudtl.dat
- j. libcef.dll
- k. libEGL.dll
- l. libGLESv2.dll
- m. natives_blob.bin
- n. snapshot_blob.bin
- o. v8_context_snapshot.bin

- use their current reference paths as a starting location, then change with the corresponding x64 paths.

----- IF YOU UPDATE CEFSHARP-----

Update the above files to point at the new version's file paths.

Client Application Update Process

Untitled Diagram.png

Code Scanning

Resources

- **Code Scan Pipeline:**
https://dev.azure.com/eFileCabinet/Utopia/_apps/hub/ms.vss-ciworkflow.build-ci-hub?_a=edit-build-definition&id=132&view=Tab_Tasks
- **Migrating from packages.config:** <https://learn.microsoft.com/en-us/nuget/consume-packages/migrate-packages-config-to-package-reference#migration-steps>

Background

We wanted to do a code scan of the Revver Desktop App, but after some research and some trial and error, we found that this was going to be more difficult than anticipated.

In order to be able to use ADO advanced security to code scan, we needed to build the app in an ADO pipeline. There are two issues with this. 1) you can't build a .vsproj project (Visual Studio Installer Project) in ADO pipelines (well, we probably could if we had our own build machines), and 2) our projects store nuget packages in a packages.config file and the ADO pipelines have issues restoring nuget packages when they are configured this way.

We attempted migrating from the packages.config to package references, and this was successful, however, it broke the installer project. We did spend several hours trying to see if we could reconfigure things in the installer project so that when you built the installer, things would work, but did not want to spend more effort trying to get this to work, so eventually gave up on this.

In the end, a pipeline was created (referenced at the top of this page), but it cannot currently be used with the master branch. In order to do a code scan you need to branch off of master, migrate the packages.config to package references, and run the pipeline on that branch. Keep in mind this will not scan the installers, just the build projects.

Doing a code scan

1. Create a branch of master (or the branch you want to create a new update installer off of)
2. Migrate from packages.config to package references
 - you can use the reference at the top of this page for instructions on how to do this (but long story short, you right click the packages.config file and select **Migrate packages.config to PackageReference...**)
3. Commit the changes to your branch and push it to ADO
4. Run the '**Revver Windows Desktop App Code Scan**' build pipeline on your branch

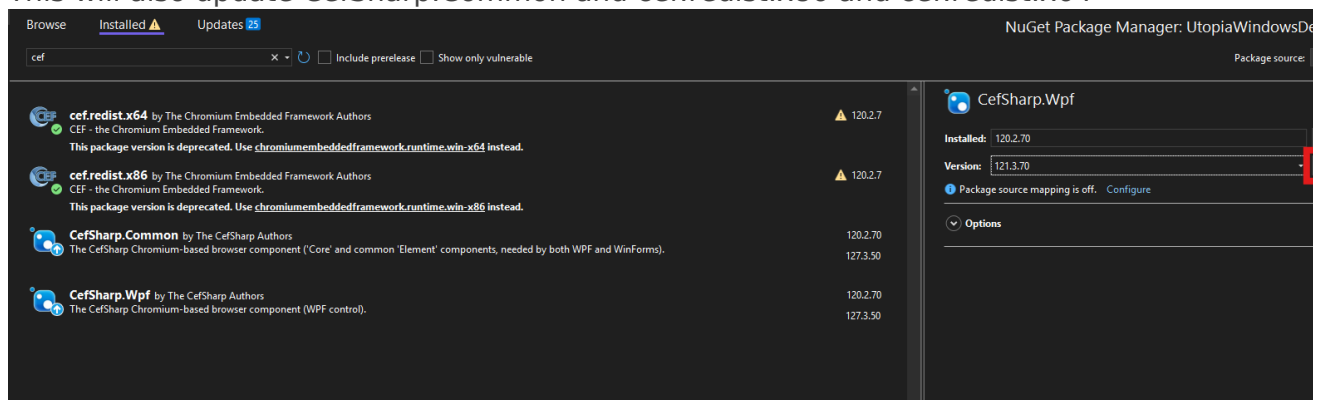
How can we automate this in the future?

I think the best way to do this would be to build the installer using wixsharp, rather than a visual studio installer project.

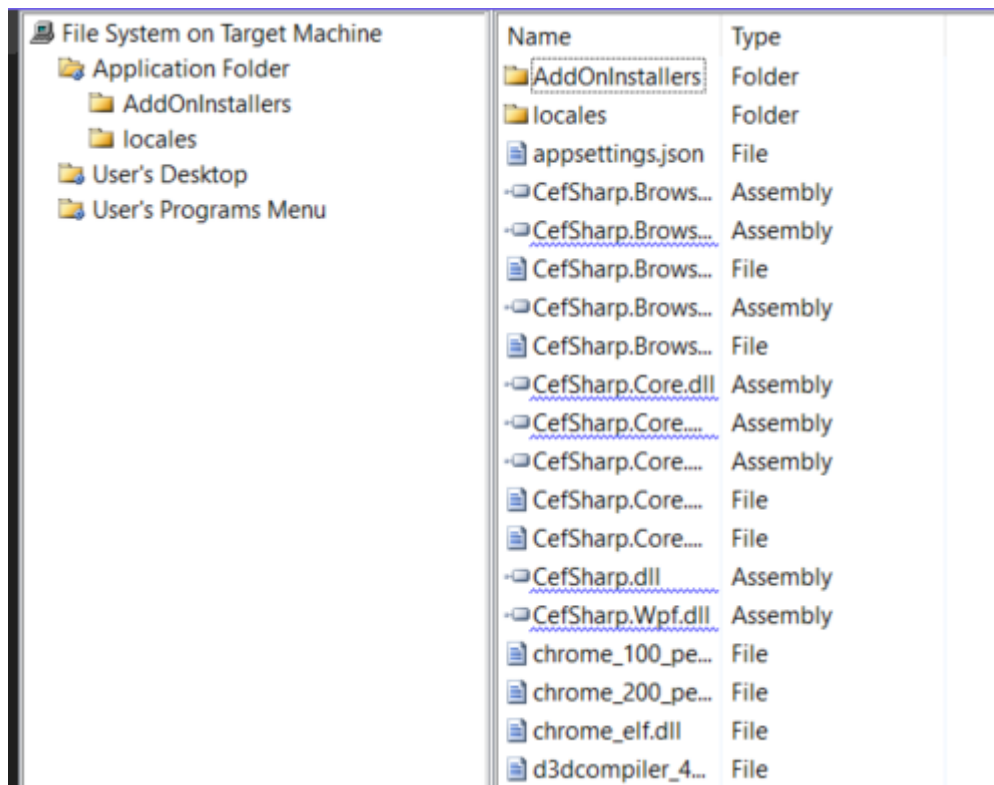
How To Update CefSharp Version

This update process may not work for future updates to CefSharp. The next version of CefSharp starts using some different libraries (chromiumembeddedframework instead of cef.redist.x86). Knowing these steps will likely still be important though because we will still likely need to update the setup / installer project

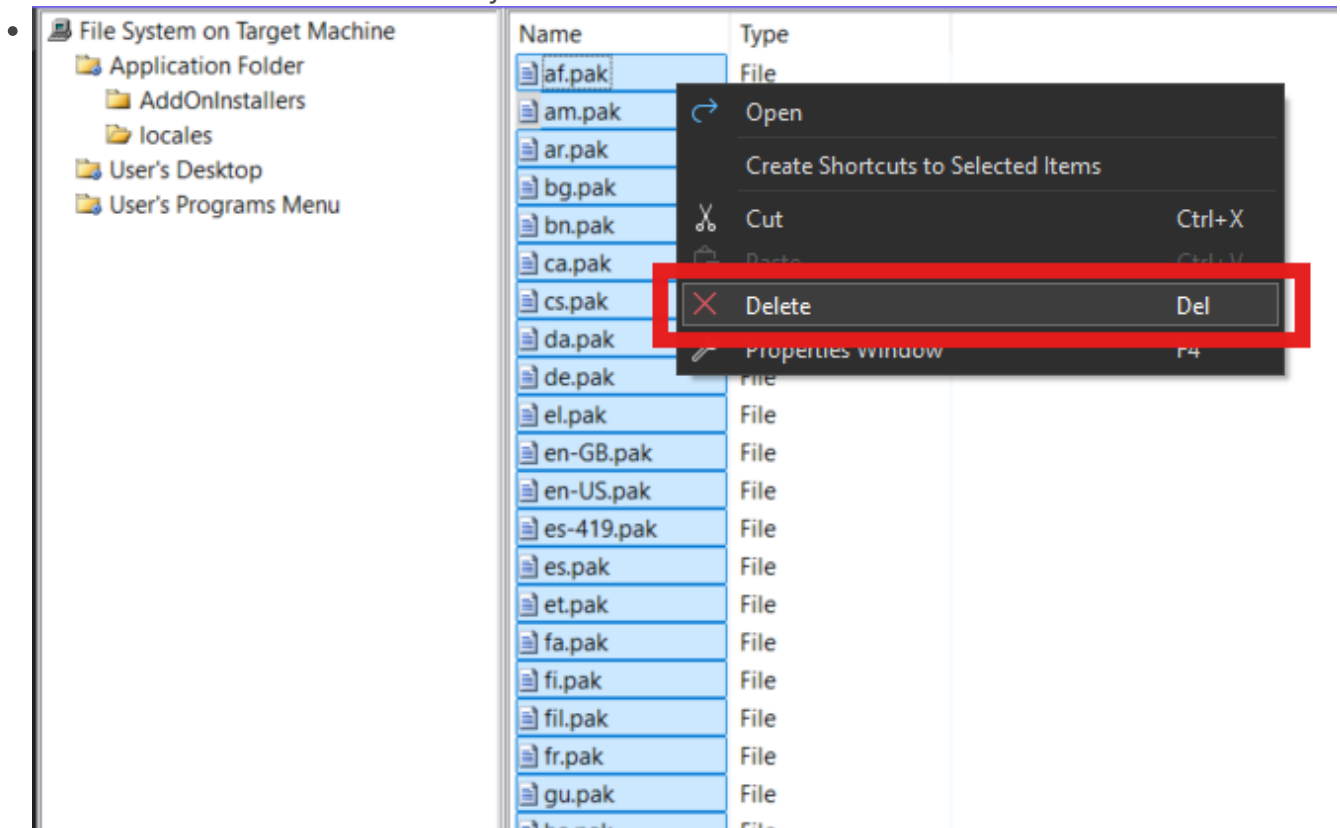
1. Update the version of CefSharp.Wpf to the desired version
 - This will also update CefSharp.Common and cef.redist.x86 and cef.redist.x64



2. Attempt to rebuild the UtopiaWindowsDesktopClient (Release | x86), you should get build errors in the output console indicating that it couldn't find files specific to the previous version of CefSharp.
 - example error
 - ERROR: Unable to find source file
'C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\cef.redist.x86.109.1.11\CEF\locales\ru.pak' for file 'ru.pak', located in
'[TARGETDIR]\locales', the file may be absent or locked.
 - You may also get build errors you need to resolve because CefSharp updates are not always backwards compatible.
3. In the UtopiaWindowsDesktopClient project, double click the first file (af.pak currently) to open up a view that looks like the following screen shot

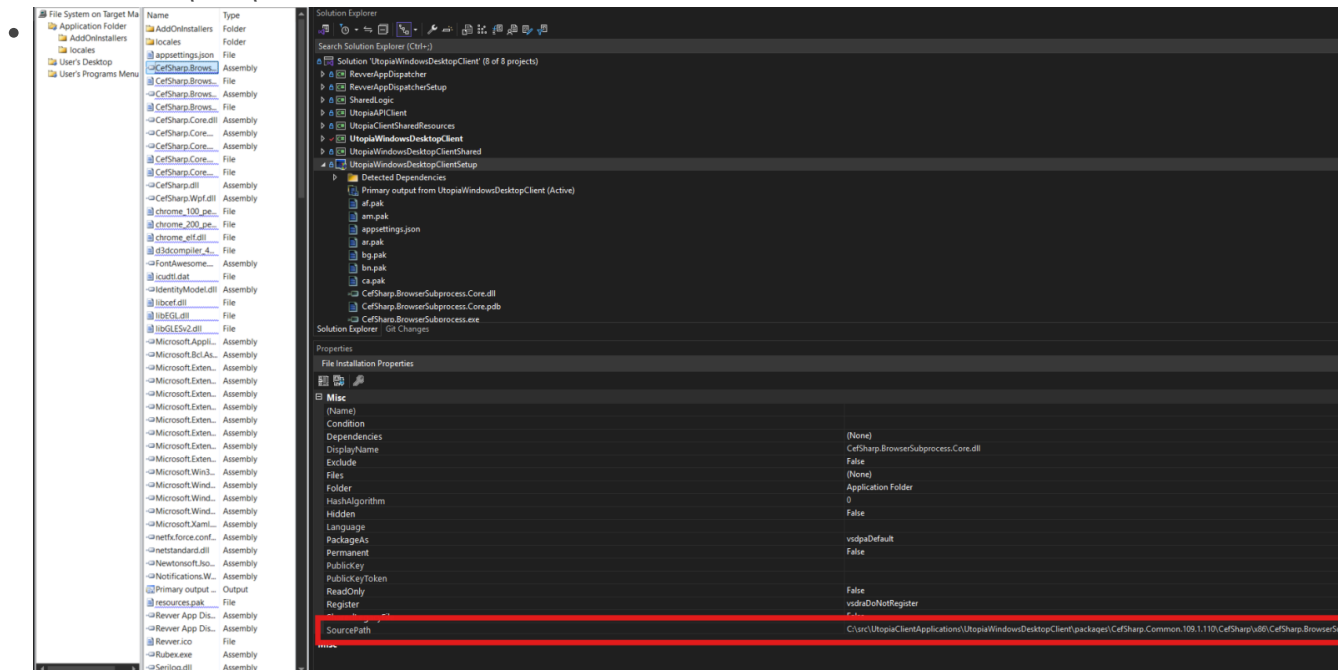


4. Delete all the files that are currently in the 'locales' folder.



5. Right click the 'locales' folder on the left, and in the dropdown select 'Add' -> 'File', and add all the files from 'C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\cef.redist.x86.{UPDATED_TO_VERSION}\CEF\locales'
6. Now select the 'Application Folder' folder on the left.
7. Delete any files that start with the following paths (be vary careful when you do this so you don't accidentally delete a file you weren't supposed to)

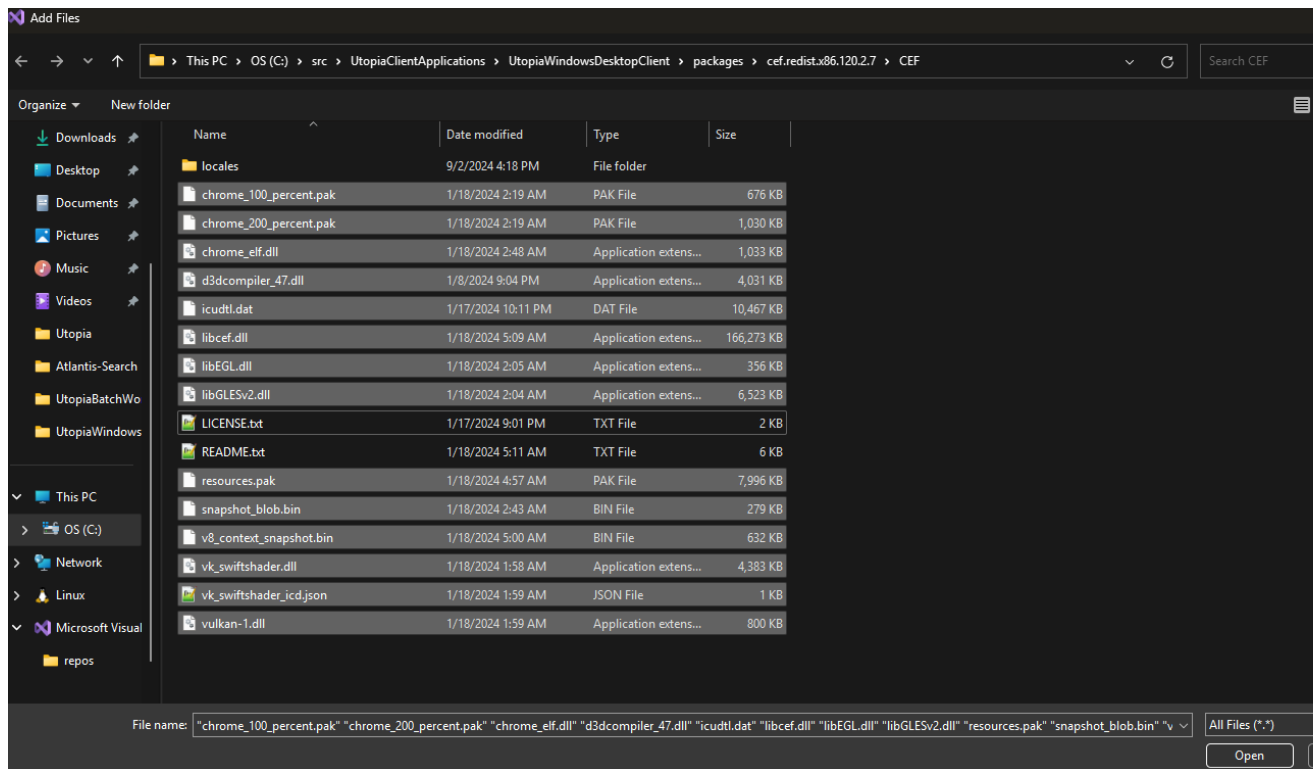
- C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\CefSharp.Common.109.1.110\CefSharp\x86\
- 'C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\cef.redist.x86.109.1.11\CEF\



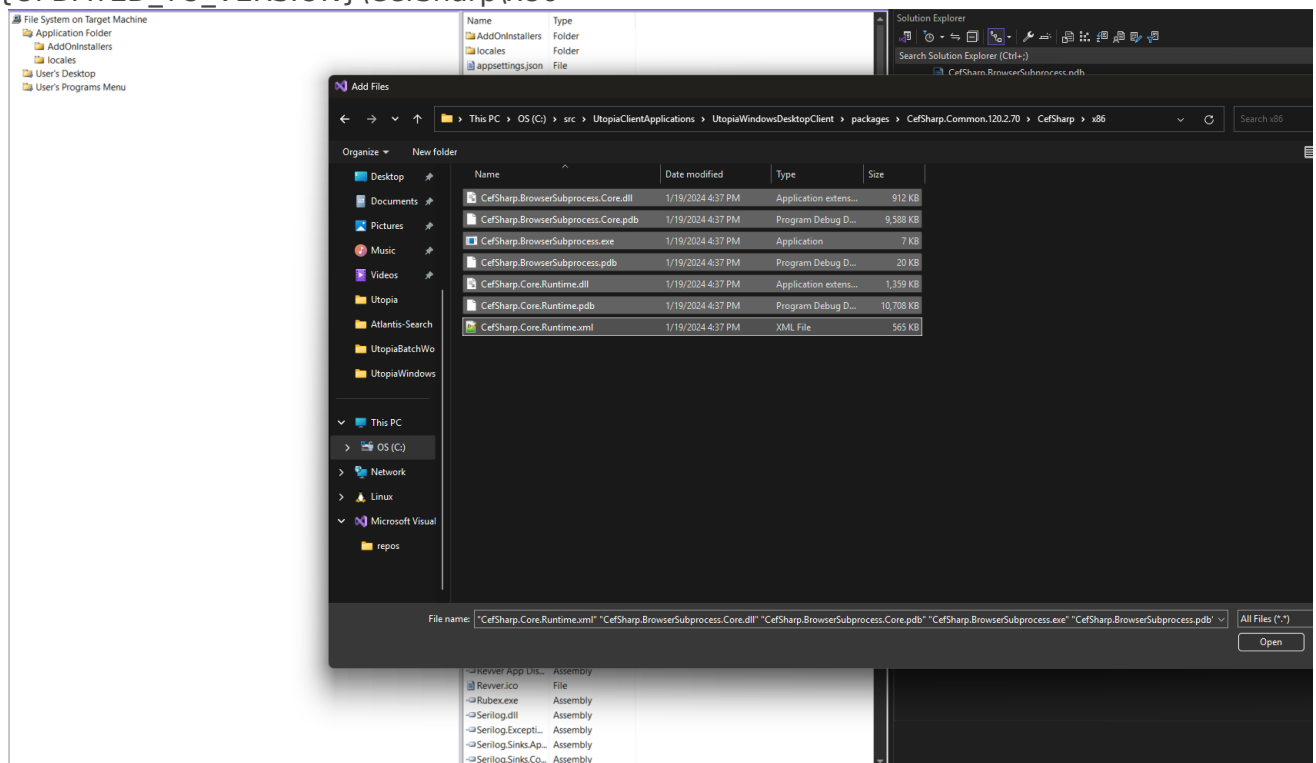
- you can reference the errors in the output console when building to see exactly what files it can't find and need to be deleted
- they may also be underlined in blue

8. Right click 'Application Folder' on the left, and in the dropdown select 'Add' => 'File', and all the files from 'C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\cef.redist.x86.109.1.11\CEF' except for the following

- locales folder
- README.txt
- LICENSE.txt
- maybe keeps eyes out for other .txt files that might be added the future (you basically want the .dlls, .pak, .dat, .bin, and .json files)



9. Right click 'Application Folder' on the left, and in the dropdown select 'Add' => 'File', and all the files from 'C:\src\UtopiaClientApplications\UtopiaWindowsDesktopClient\packages\CefSharp.Common.{UPDATED_TO_VERSION}\CefSharp\x86'



10. You may want to make sure that you now don't have any duplicate files in 'Application Folder' or any files that you missed deleting that referenced the old CefSharp version.
11. You should now be able to build the installers per usual.