Using Visual Studio with Git

Visual Studio brings in lots of files and folders that don't belong in a checked in repository.

Here's a good gitignore file for Visual Studio projects:

```
# This works in Visual Studio projects
#   to ignore most VS generated output.
# NOTE: It does not work for VSCode.

# Ignore Build results...
**/[Bb]in/*
**/[Oo]bj/*
**/[Ll]og/*
**/[Ll]ogs/*
**/[Tt]emp/*
**/[Tt]mp/*

# Ignore Visual Studio specific...
**/.vs/*
*.user
*.suo
*.userosscache
*.sln.docstates

# Ignore publish folders...
[Pp]ublish/
```

For Existing Project

Changes to a Gitignore doesn't untrack files that have already been committed.
So. If you are retrofitting a gitignore to an existing project, you may need to follow this sequence, to ensure that files are properly ignored and committed.

Update your GitIgnore

Update the .gitignore file in your working copy with the above content.

Untrack All Files

Open a Powershell terminal, and navigate to the root of your working copy.
You may have to quote the folder path, for Powershell to recognize it.

Run this powershell from the checkout root, to untrack all files in Git:

```
# Remove all tracked bin, obj, .vs, and publish folders
git ls-files | Where-Object {
    $_ -match '\\(bin|obj|\.vs|publish)\\'
} | ForEach-Object {
    git rm -r --cached "$_"
}
```

Clear Cached and Add all

Then, open a command line window, also in the checkout root, and run this to add your gitignore file and commit the cleanup:

> NOTE: These must be run as three separate commands to property execute.

```
git rm -r --cached .
git add .
git commit -m "Refresh tracked files with updated .gitignore"
```

Once that is done, your checkout will think there is nothing to check in.
So, we need to force it to commit everything.

Rename to Force Checkin

Rename the solution folder, to ensure that the entire solution is considered as a new commit.

TortoiseGit will perceive every file as net-new, being in a different folder.

Checkin All

Open TortoiseGit, and check in all files.

> NOTE: Be sure to select All, to capture the unknown and deleted file sets.

Fix Rename

Rename the solution folder, back to its original.

And, do another commit.

> Again. Be sure to select All, to capture the unknown and added file sets.

Now, you should have everything properly checked in, with your gitignore file working.

Revision #6
Created 10 November 2025 21:21:13 by glwhite
Updated 10 November 2025 22:06:54 by glwhite