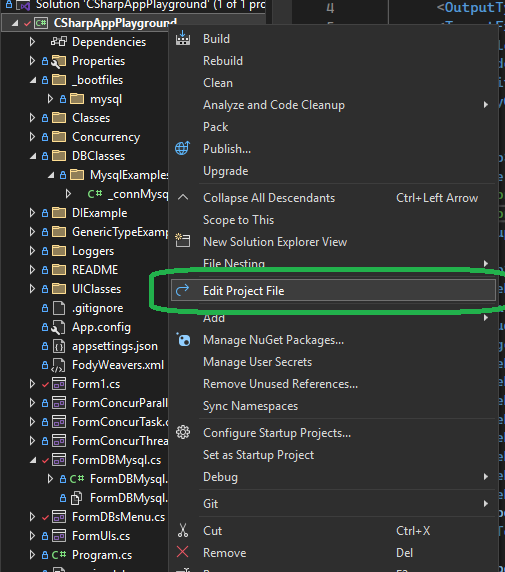
C# Project - quick guide

This document will be quick guide for common C# project files

# .csproj (xml file)



1. Right click on project > Edit Project File
2. .csproj xml view

*Some common attributes to know*

### 1. Including files into the project

This is useful for copying file over the build folder, example a Readme document or config files.

Note: some type of files like appsettings.json are built-in so C# proj already know to set those attributes in automatically

<ItemGroup>

<None Update="appsettings.json">

<CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>

</None>

<None Update="version.txt">

<CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>

</None>

<None Include="version.txt">

<CopyToOutputDirectory>Always</CopyToOutputDirectory>

</None>

</ItemGroup>

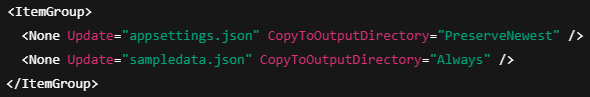
Property for `CopyToOutputDirectory`

* PreserveNewest = copy file on compile when file changed or doesn’t exist
* Always = always copy file on compile
* Never = never copy the file over

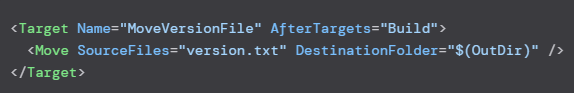
Options

* <None Update=”” = file with updatable meta data and by default included in project
* <None Include=”” = file to be included into project

Alt format



### 1b. Including files into the build (another option)



### 2. Ignoring files and folder in project

Sometimes we need to put some test files or database initialization files in the project just for better book keeping and maintenance. But these database or test files should not be parsed, compiled or included in the build.

In the .csproj file these attributes can help control the files and folders.

* <Compile Remove || Include
  + Compile: This item should not be compiled, DEFAULT is include. So by default files does not need to be included. Only Remove=”” needs to be explicit
* <Content Remove || Include
  + Content: Useful for web application to be copy to a dist folder etc… Same default and explicit Remove call as Compile
* <None Remove || Include
  + This is for unknown file types, like files that are not .cs etc. Same logic for Remove and Include as other tags

<ItemGroup>

<Compile Remove="\_bootfiles\\*\*\\*" />

<Content Remove="\_bootfiles\\*\*\\*" />

<None Remove="\_bootfiles\\*\*\\*" />

</ItemGroup>

Other Tip and Usage:

1. When building or compiling reference files like SQL gets processed, so use <Compile> to remove them from the build process. It can take specific file or folder.

2. Using None to remove the files will also remove from the solution window

|  |  |
| --- | --- |
|  |  |

3. Use Link to copy files from a file somewhere outside to another folder in the build

<ItemGroup>

<None Include="..\Shared\Resources\config.json" Link="Resources\config.json" />

</ItemGroup>

<Folder> is used by C# proj to keep track of an empty folder, so it can be displayed in the IDE solution listing. If a file is added to that folder then either this is removed or it is redundant.

