Update of a .NET (C#) project for Autocad 2025 (.NET 8.0)

(This Migration Guide was originally written by Gilles Chanteau in French and posted to the AutoCAD .NET forum. Translated to English by Jeff_M with permission.)

Beginning with the 2025 version, the .NET plugins for AutoCAD must target .NET Core 8.0 (or .NET 8.0) instead of .Net Framework.

The plugins targeting previous versions of the .NET framework (4.8 for Autocad 2024) must therefore be upgraded to operate on AutoCAD 2025 (and subsequent versions).

This is a break in compatibility, the second since the integration of the .NET programming in AutoCAD (the first had taken place with the 2013 version and the Acad.exe division in two Files: Acad.exe and ACCore.dll).

The objective of this document is to show how to update a project targeting the .net framework so that it is compatible with AutoCAD 2025. The process requires the use of Visual Studio 2022 (version 17.8 or later) and, of course, an AutoCAD version 2025. As with the previous versions, it is best to reference AutoCAD libraries from the SDK ObjectARX 2025

Downloadable from this page.

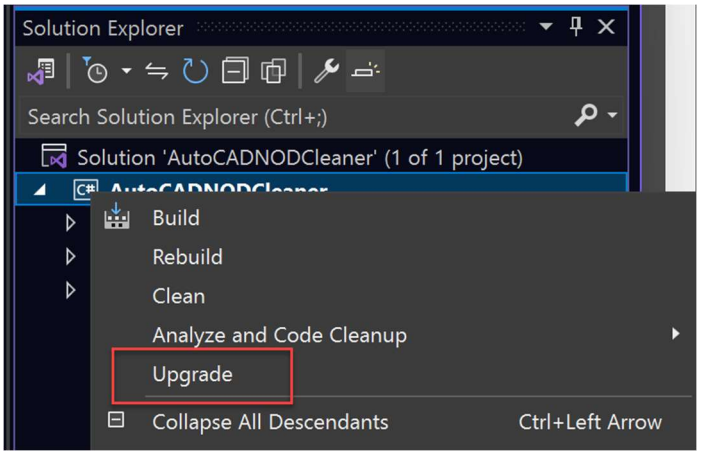
The .NET upgrade assistant

To facilitate the update, Microsoft provides an upgrade assistant. The simplest is to install the Visual Studio extension (see information).

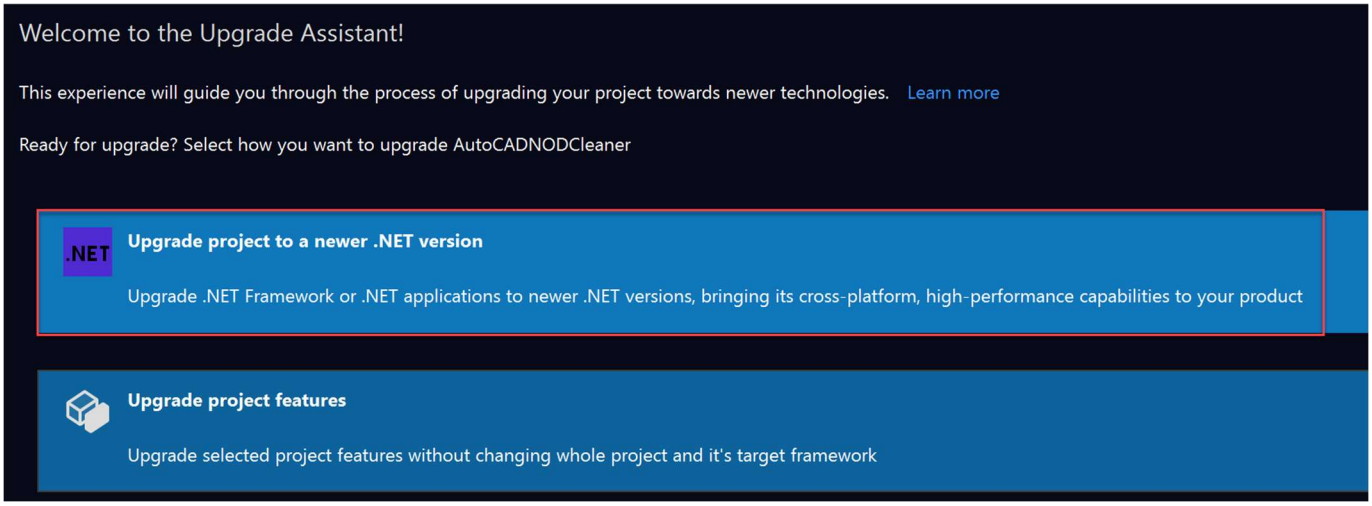
Migration from Visual Studio

Open the .NET project to migrate to Visual Studio 2022.

In the solutions explorer, right click on the project and then click on Upgrade.



Choose: Upgrade Project to a Newer .Net version.



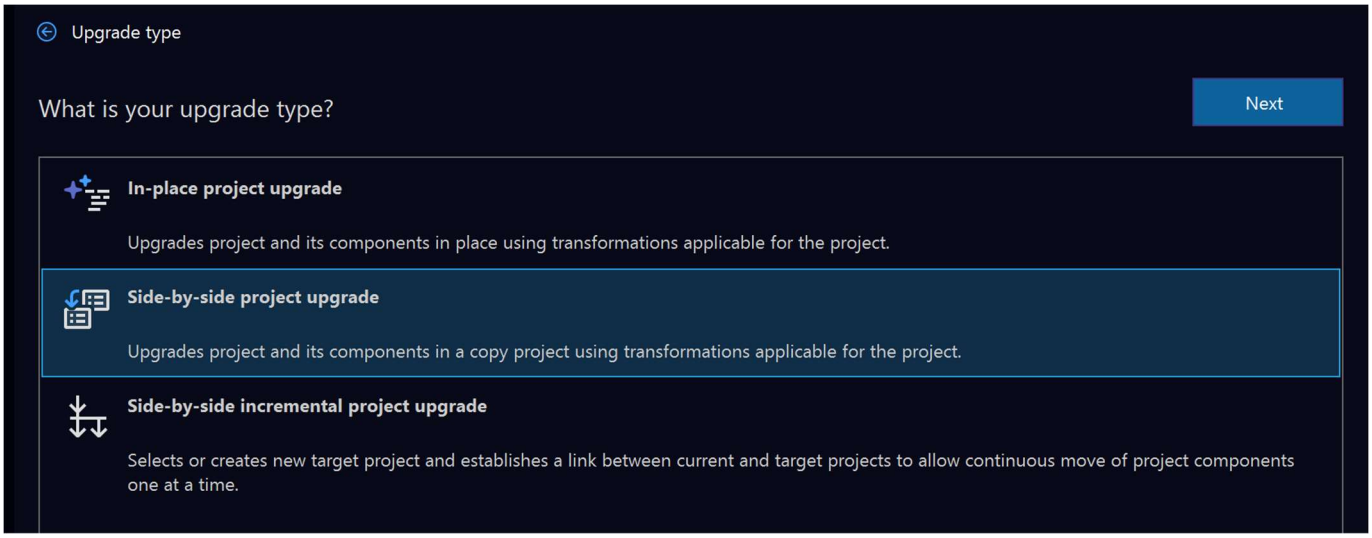
Choose the type of upgrade

In-Place Project Upgrade to replace the existing project

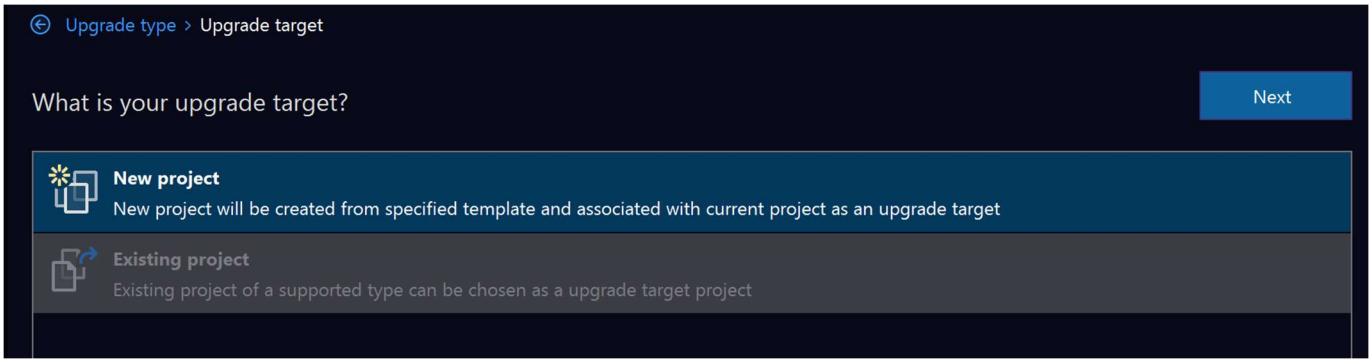
Side-By-Side Project Upgrade to create a new updated project while retaining the original project

SIDE-CONSIDE INCREMENTAL PROJECT UPGRADE to create a new updated project by linking the code of New project to that of the original. This option allows you to have to modify the code only in a project, but makes more complicated the use of new features of C# 12 which do not exist in the original project (C# 7 at most).

Then Next.

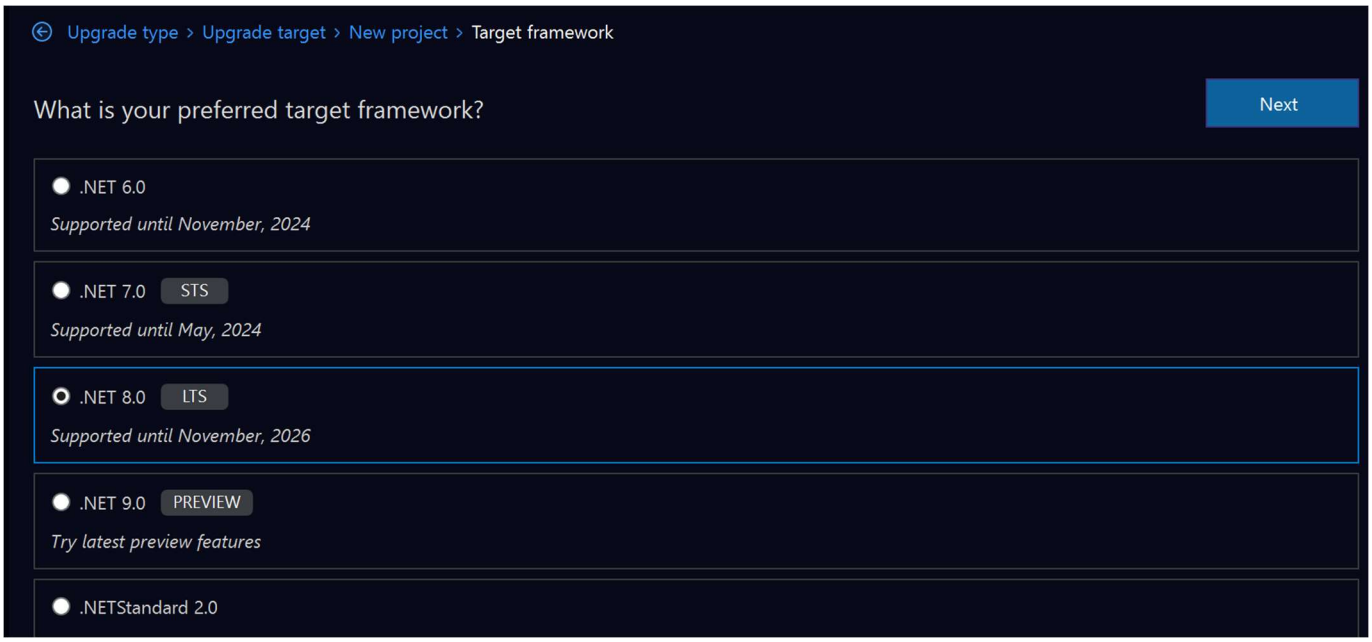


Select New Project, then Next:

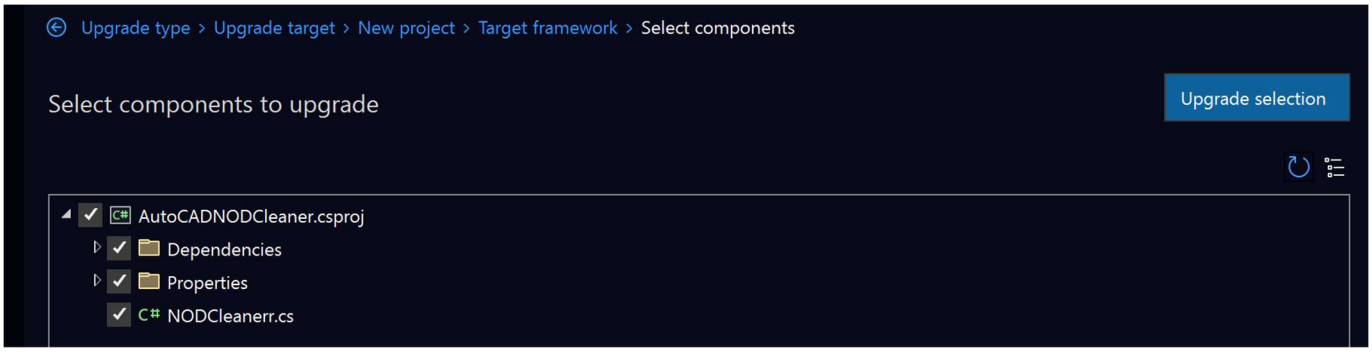


Give the new project a name, then Next:

Select .NET 8, then Next:



Leave all the boxes checked, then select upgrade.



Modification of the .csProj file

Once the migration with the assistant is completed, some changes to the file must be made .csProj. This file is opened by double clicking on the new project in the solution explorer.
In the first PropertyGroup tag, replace:

<TargetFramework>net8.0</TargetFramework>
With:
<TargetFramework>net8.0-windows</TargetFramework>

And, to be able to use the Nullable Reference types (optional), add a Nullable tag:
<Nullable>enable</Nullable>

Add an itemgroup to avoid conflict errors between .NET Framework and .NET 8.0.
<ItemGroup>
<FrameworkReference Include="Microsoft.WindowsDesktop.App" />
</ItemGroup>

Modify the paths of the AutoCAD libraries (ACCOREMGD.DLL, ACDBMGD.DLL, ACMGD.DLL,...) to Reference those of AutoCAD 2025.
If the file contains a propertygroup tag containing StartAction, StartProgram, or StartArguments tags to automatically start AutoCAD in Debug mode, they can be deleted. With .Net Core, the start -up actions and options are saved in a JSON file (see below).

The .csproj file should look, more or less, like this:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0-windows</TargetFramework>
    <OutputType>Library</OutputType>
    <GenerateAssemblyInfo>false</GenerateAssemblyInfo>
    <UseWPF>true</UseWPF>
    <ImportWindowsDesktopTargets>true</ImportWindowsDesktopTargets>
      <Nullable>enable</Nullable>
  </PropertyGroup>
      <ItemGroup>
            <FrameworkReference Include="Microsoft.WindowsDesktop.App" />
      </ItemGroup>
      <ItemGroup>
    <Reference Include="AcCoreMgd">
```

```
      <HintPath>D:\Autodesk\ObjectARX\ObjectARX 2025\inc\AcCoreMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="AcDbMgd">
      <HintPath>D:\Autodesk\ObjectARX\ObjectARX 2025\inc\AcDbMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="AcMgd">
      <HintPath>D:\Autodesk\ObjectARX\ObjectARX 2025\inc\AcMgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
  </ItemGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.CSharp" Version="4.7.0" />
    <PackageReference Include="System.Data.DataSetExtensions" Version="4.5.0" />
  </ItemGroup>
</Project>
```

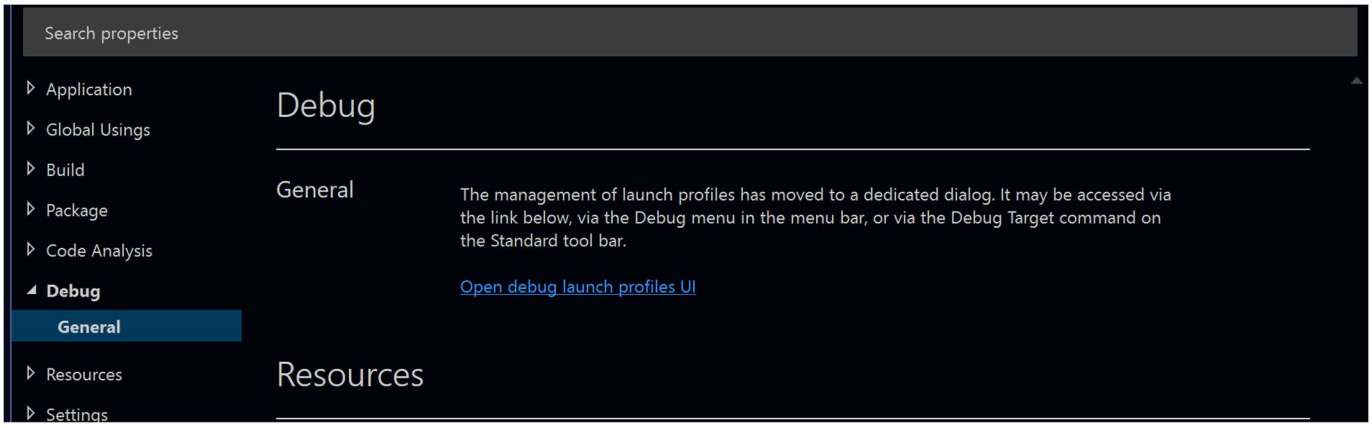
Save the changes.

Change project properties

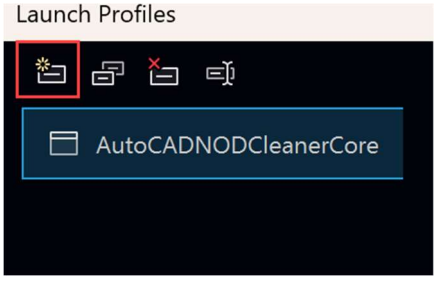
Some of the modifications made to the .csproj file could also have been made from the project properties.

To automatically launch AutoCAD at start-up in Debug mode, actions must be changed and Start-up options by creating a "Debug launch profile".

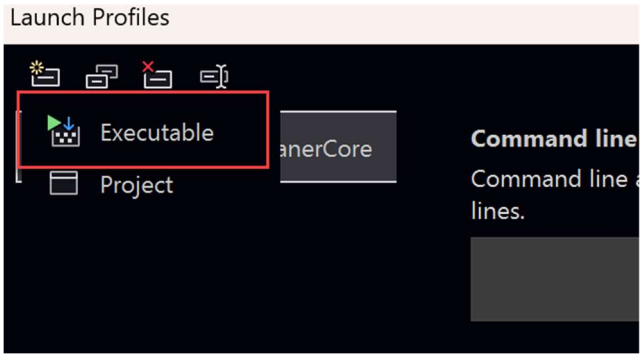
In the properties of the project, open the Debug section.



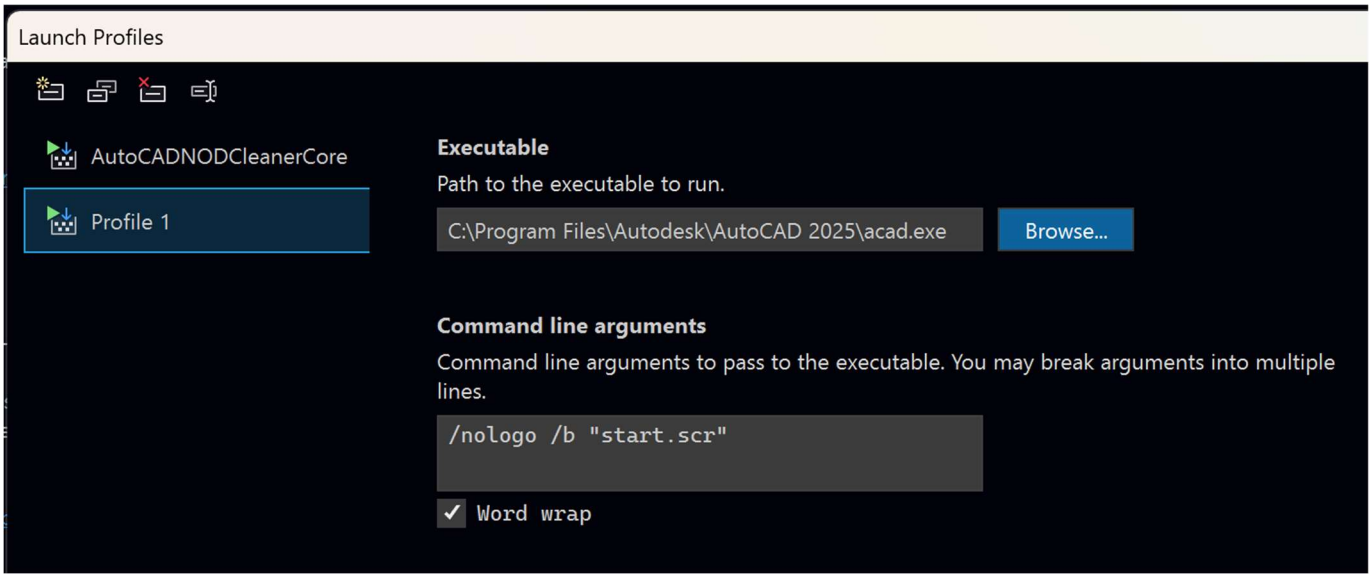
Then click on Open debug launch profiles to open the dialog box profiles. Launch and click on Create a profile:



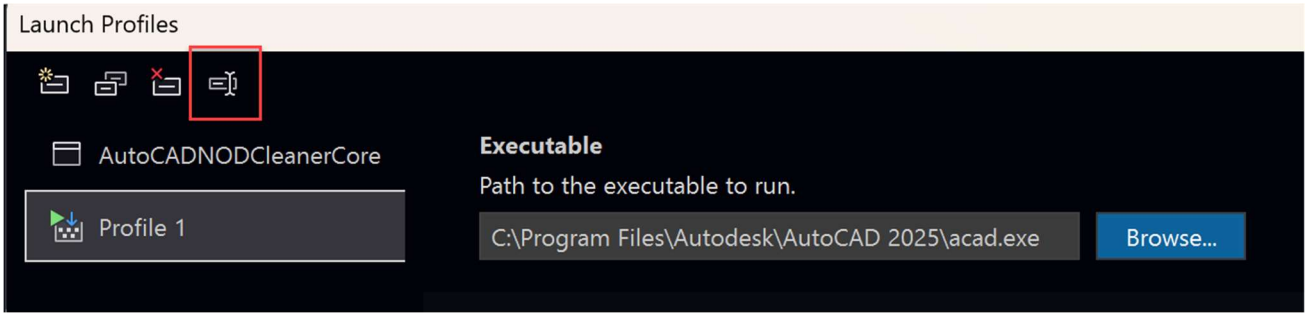
Select Executable:



Specify the path of the AutoCAD version to be opened and, possibly any command line arguments, for example to launch a script at AutoCAD startup that will load the project.



Then click on the rename the selected profile button:



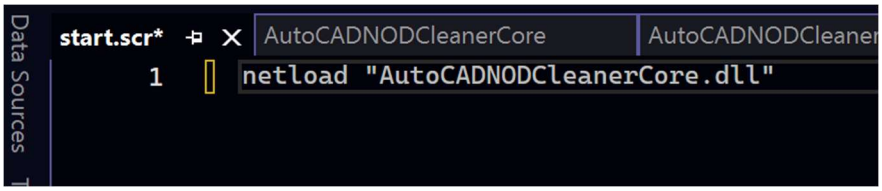
And enter the name of the original profile to overwrite it.

Validate and close the dialog box.

These changes have been saved in the launchsettings.json file of the Properties folder of project. It would have been possible to create or modify this file directly in Visual Studio.

```
{
  "profiles": {
    "AutoCADNODCleanerCore": {
      "commandName": "Executable",
      "executablePath": "C:\\Program Files\\Autodesk\\AutoCAD 2025\\acad.exe",
      "commandLineArgs": "/nologo /b \"start.scr\""
    }
  }
}
```

If in the command line arguments a script is launched to automatically load the Project, it is necessary to modify the name of the DLL so that it corresponds to that of the new project.



Modify the code of .cs files

During migration, the source file code was copied as is in the new project. Now correct any errors and use new features, modify the content of Files (the Errors List window and the IntelliSense provide precious help).

It is also preferable to change the name space to make it match the structure of Files (project name) with a search/replace in the existing project. This will solve any problems with files automatically generated by Visual Studio (files resources for example).

Define as a startup project the new project and launch it debugging it to test if everything works as hoped for.