

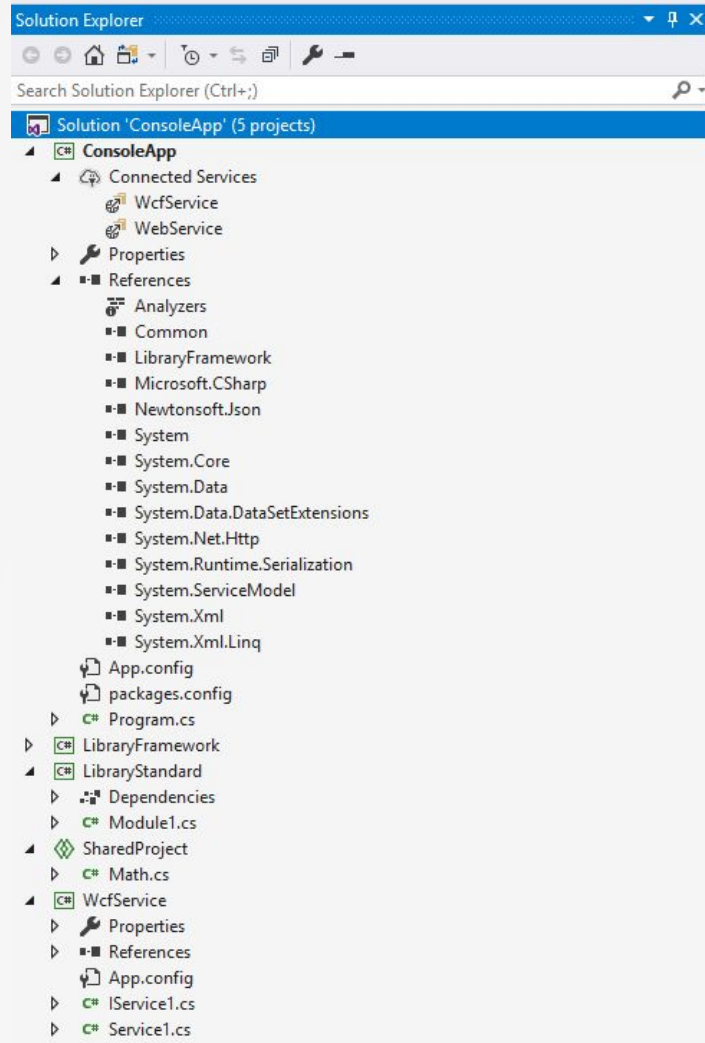
Dependency Management



Speaker:

Igor

Solution



Solution is simply a container for one or more related projects, along with build information and Visual Studio window settings.

Project

In a logical sense, a project contains all the source code files, icons, images, data files, etc. that are compiled into an executable, library, or website. A project is defined in an XML file with an extension such as .vbproj, .csproj, or .vcxproj. This file contains a virtual folder hierarchy, and paths to all the items in the project. It also contains the build settings.

```
LibraryFramework.csproj
4 <PropertyGroup>
5   <Configuration Condition=" '$(Configuration)' == '' >Debug</Configuration>
6   <Platform Condition=" '$(Platform)' == '' >AnyCPU</Platform>
7   <ProjectGuid>{8C224FE2-D18E-43C7-80DF-F527A75A8580}</ProjectGuid>
8   <OutputType>Library</OutputType>
9   <AppDesignerFolder>Properties</AppDesignerFolder>
10  <RootNamespace>LibraryFramework</RootNamespace>
11  <AssemblyName>LibraryFramework</AssemblyName>
12  <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
13  <FileAlignment>512</FileAlignment>
14 </PropertyGroup>
15 <PropertyGroup Condition=" '$(Configuration)' == 'Debug' >
24 <PropertyGroup Condition=" '$(Configuration)' == 'Debug|x86' >
33   <DebugSymbols>true</DebugSymbols>
34   <OutputPath>bin\x86\Debug</OutputPath>
35   <DefineConstants>TRACE;DEBUG;DoubleLibrary</DefineConstants>
36   <DebugType>full</DebugType>
37   <PlatformTarget>x86</PlatformTarget>
38   <ErrorReport>prompt</ErrorReport>
39   <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
40 </PropertyGroup>
41 <PropertyGroup Condition=" '$(Configuration)' == 'Release' >
50 <ItemGroup>
51   <Reference Include="log4net, Version=1.2.13.0, Culture=neutral, PublicKeyToken=669e0ddfb0b1aa2a, processorArchitecture=MSIL">
52     <HintPath>..\packages\log4net.2.0.3\lib\net40-full\log4net.dll</HintPath>
53   </Reference>
54   <Reference Include="Newtonsoft.Json, Version=4.5.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed, processorArchitecture=MSIL">
55     <HintPath>..\packages\Newtonsoft.Json.5.0.8\lib\net45\Newtonsoft.Json.dll</HintPath>
56   </Reference>
57   <Reference Include="SuperSocket.Common, Version=1.6.0.4, Culture=neutral, PublicKeyToken=6c8000676988ebb, processorArchitecture=MSIL">
58     <HintPath>..\packages\SuperWebSocket.0.9.0.2\lib\net40\SuperSocket.Common.dll</HintPath>
59   </Reference>
60   <Reference Include="SuperSocket.SocketBase, Version=1.6.0.4, Culture=neutral, PublicKeyToken=6c8000676988ebb, processorArchitecture=MSIL">
61     <HintPath>..\packages\SuperWebSocket.0.9.0.2\lib\net40\SuperSocket.SocketBase.dll</HintPath>
62   </Reference>
63   <Reference Include="SuperSocket.SocketEngine, Version=1.6.0.4, Culture=neutral, PublicKeyToken=6c8000676988ebb, processorArchitecture=MSIL">
64     <HintPath>..\packages\SuperWebSocket.0.9.0.2\lib\net40\SuperSocket.SocketEngine.dll</HintPath>
65   </Reference>
66   <Reference Include="SuperWebSocket, Version=0.9.0.0, Culture=neutral, PublicKeyToken=7ba53b9a7cef5d1c, processorArchitecture=MSIL">
67     <HintPath>..\packages\SuperWebSocket.0.9.0.2\lib\net40\SuperWebSocket.dll</HintPath>
68   </Reference>
69   <Reference Include="System" />
70   <Reference Include="System.Core" />
71   <Reference Include="System.Xml.Linq" />
72   <Reference Include="System.Data.DataSetExtensions" />
73   <Reference Include="Microsoft.CSharp" />
74   <Reference Include="System.Data" />
75   <Reference Include="System.Net.Http" />
76   <Reference Include="System.Xml" />
77 </ItemGroup>
```



When you compile your project, the MSBuild engine consumes the project file to create the executable.

Visual Studio uses two file types to store settings for solutions:

```
VisualStudioVersion = 15.0.27703.2035
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ConsoleApp", "ConsoleApp\ConsoleApp.csproj", "{532D0597-AA42-4806-B287-101ACD014544}"
  ProjectSection(ProjectDependencies) = postProject
    (03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D) = {03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}
    (42D9A0AA-0C81-4A9C-86F3-7586098919CC) = {42D9A0AA-0C81-4A9C-86F3-7586098919CC}
  EndProjectSection
EndProject
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "LibraryFramework", "LibraryFramework\LibraryFramework.csproj", "{8C224FE2-D18E-43C7-80DF-F527A75A8580}"
EndProject
Project("{9A19103F-16F7-4668-BE54-9A1E7A4F7556}") = "LibraryStandard", "LibraryStandard\LibraryStandard.csproj", "{03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}"
EndProject
Project("{D954291E-2A0B-460D-934E-DC6B0785DB48}") = "SharedProject", "SharedProject\SharedProject.shproj", "{42C12773-E218-4FE1-AD12-02A0526085B2}"
EndProject
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "WcfService", "WcfService\WcfService.csproj", "{42D9A0AA-0C81-4A9C-86F3-7586098919CC}"
EndProject
Global
  GlobalSection(SharedMSBuildProjectFiles) = preSolution
    SharedProject\SharedProject.projitems*{42c12773-e218-4fe1-ad12-02a0526085b2}*SharedItemsImports = 13
    SharedProject\SharedProject.projitems*{8c224fe2-d18e-43c7-80df-f527a75a8580}*SharedItemsImports = 4
  EndGlobalSection
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {532D0597-AA42-4806-B287-101ACD014544}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {532D0597-AA42-4806-B287-101ACD014544}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {532D0597-AA42-4806-B287-101ACD014544}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {532D0597-AA42-4806-B287-101ACD014544}.Release|Any CPU.Build.0 = Release|Any CPU
    {8C224FE2-D18E-43C7-80DF-F527A75A8580}.Debug|Any CPU.ActiveCfg = Debug|x86
    {8C224FE2-D18E-43C7-80DF-F527A75A8580}.Debug|Any CPU.Build.0 = Debug|x86
    {8C224FE2-D18E-43C7-80DF-F527A75A8580}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {8C224FE2-D18E-43C7-80DF-F527A75A8580}.Release|Any CPU.Build.0 = Release|Any CPU
    {03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}.Debug|Any CPU.ActiveCfg = Debug|x64
    {03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}.Debug|Any CPU.Build.0 = Debug|x64
    {03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {03BBE03D-5D7B-4B3B-B3C4-B4D8EC26EC3D}.Release|Any CPU.Build.0 = Release|Any CPU
    {42D9A0AA-0C81-4A9C-86F3-7586098919CC}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {42D9A0AA-0C81-4A9C-86F3-7586098919CC}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {42D9A0AA-0C81-4A9C-86F3-7586098919CC}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {42D9A0AA-0C81-4A9C-86F3-7586098919CC}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
  GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {EF415ECD-33E6-4F44-B539-305E3FCF8462}
```

.sln - Visual Studio Solution.

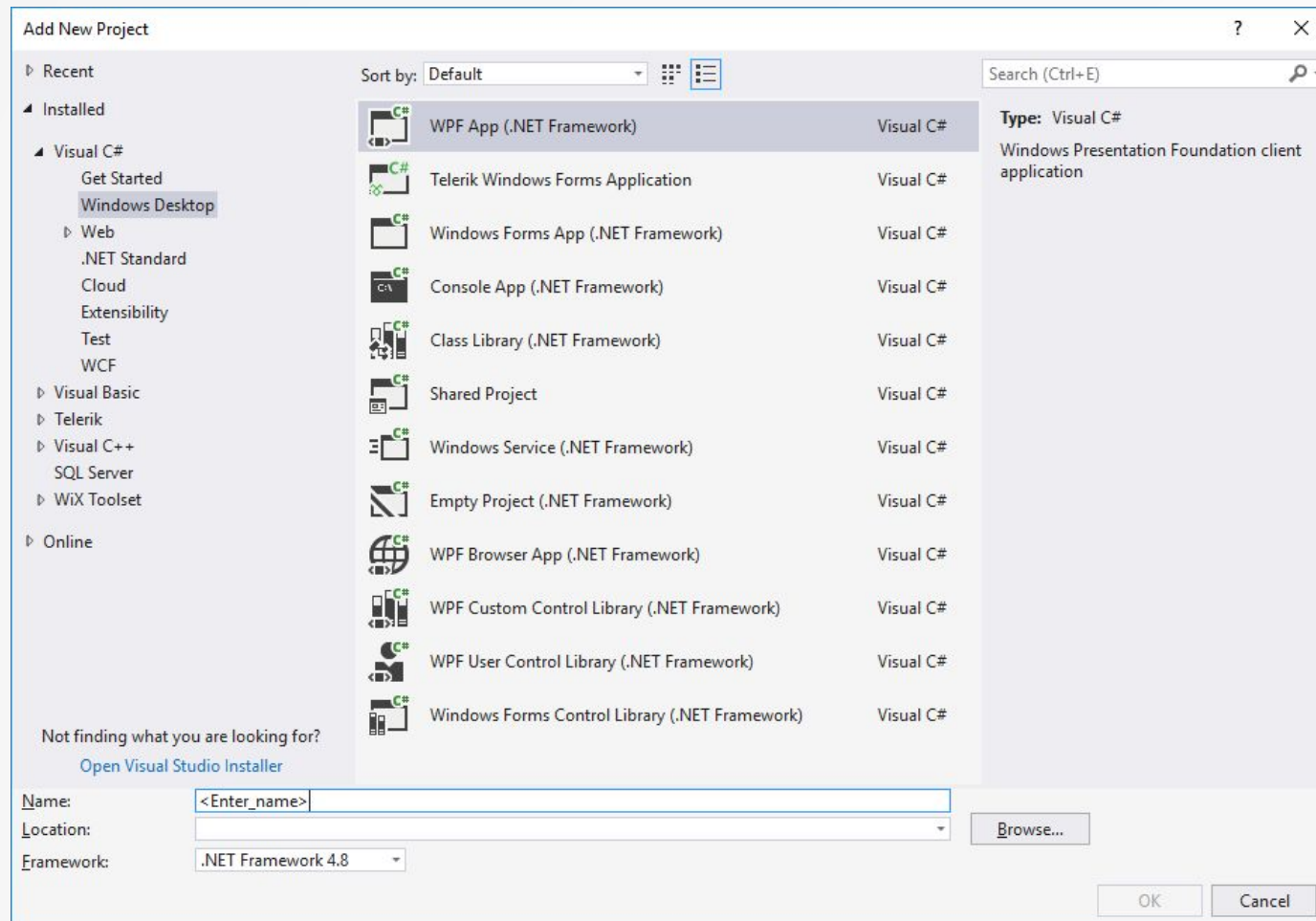
Organizes projects, project items, and solution items in the solution.

.suo - Solution User Options.

Stores user-level settings and customizations, such as breakpoints.

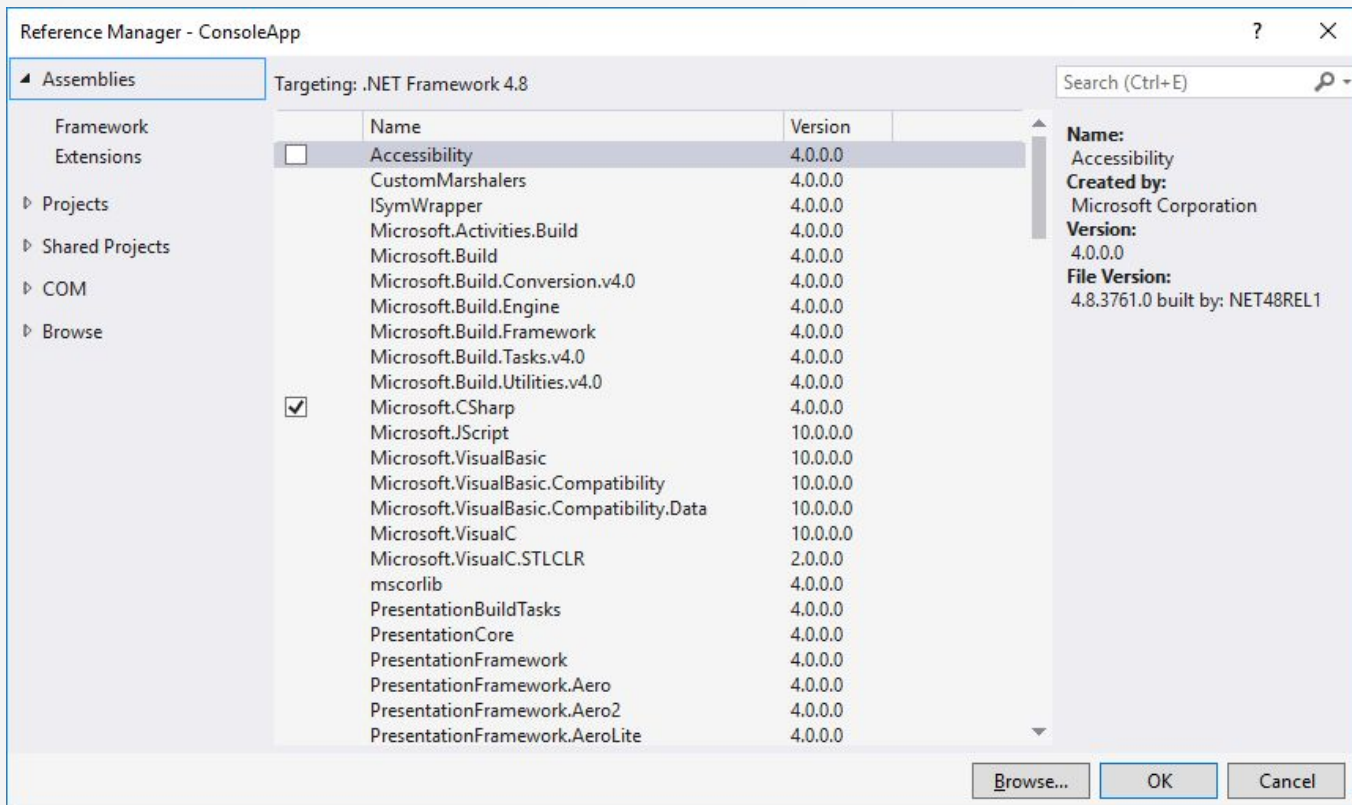
Create new projects

The easiest way to create a new project is to start from a project template for a particular type of application or website. A project template consists of a basic set of pre-generated code files, config files, assets, and settings.



Reference Manager

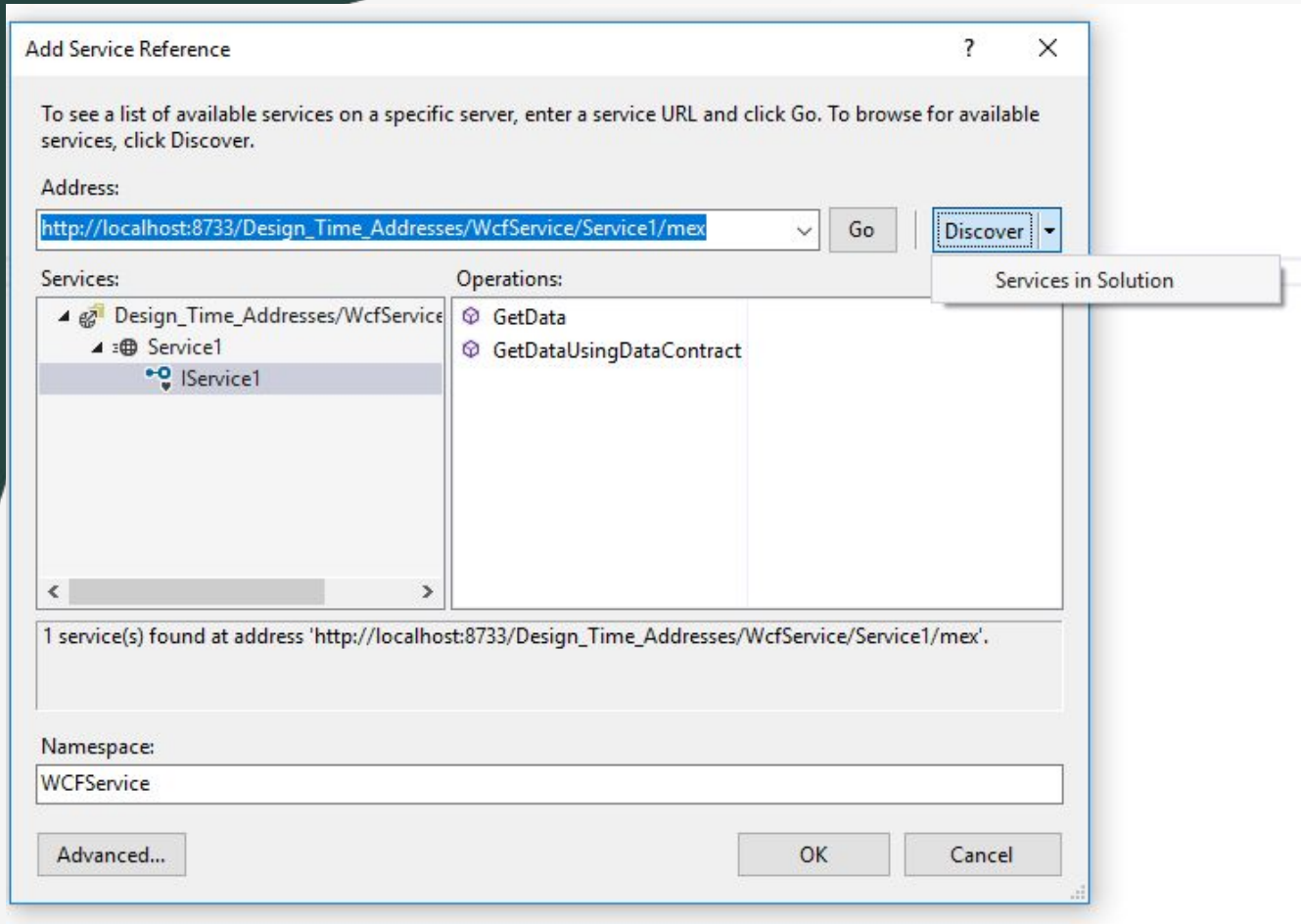
Reference Manager allows to add and manage references to components that you, Microsoft, or another company developed.



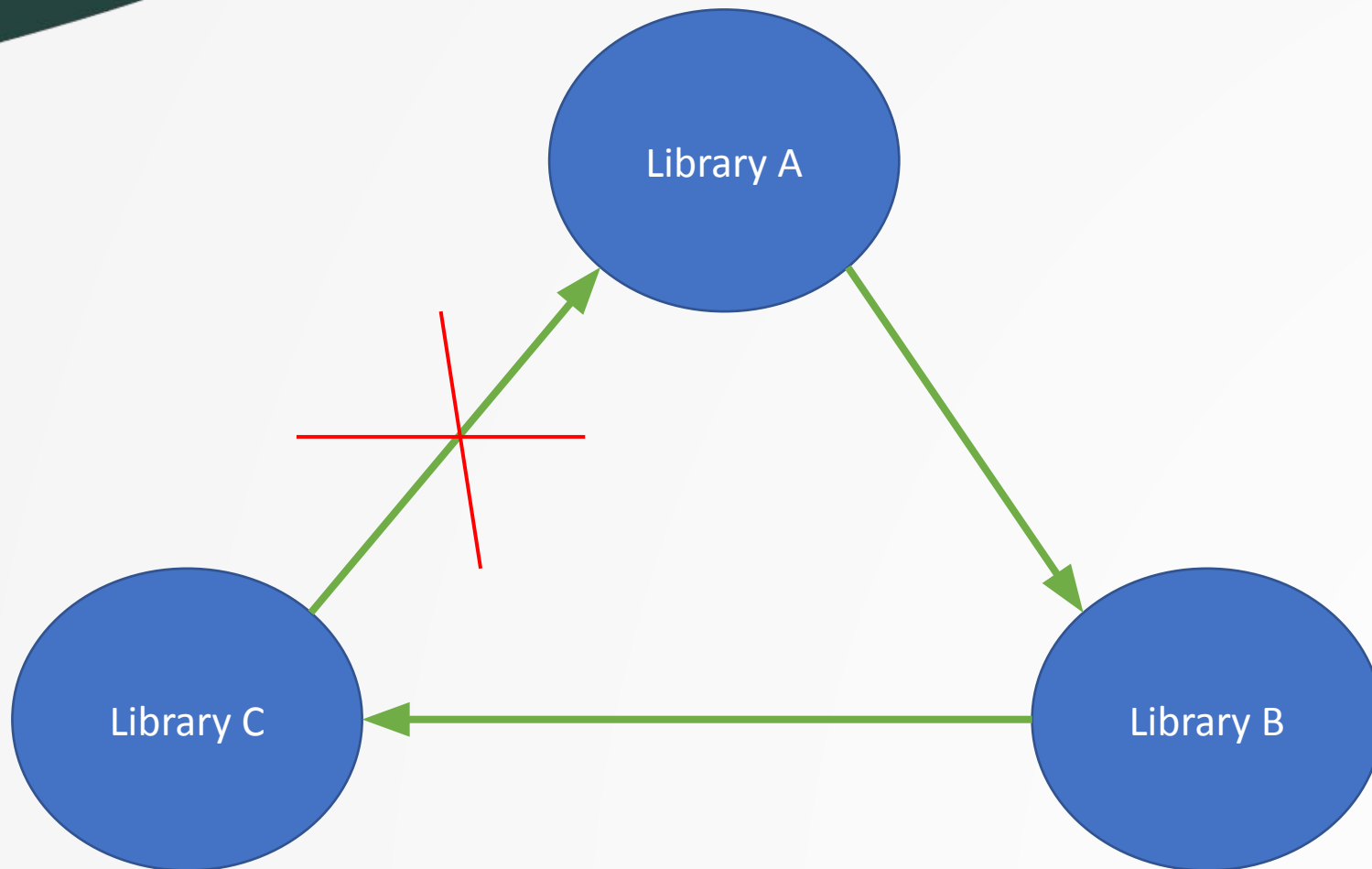
- > Assemblies
- > Projects
- > Shared project
- > Com
- > Browse

Add Service Reference Tool

This tool retrieves metadata from a web service in the current solution, on a network location, or from a WSDL file, and generates a .NET Core compatible source file containing Windows Communication Foundation (WCF) client proxy code that you can use to access the web service.

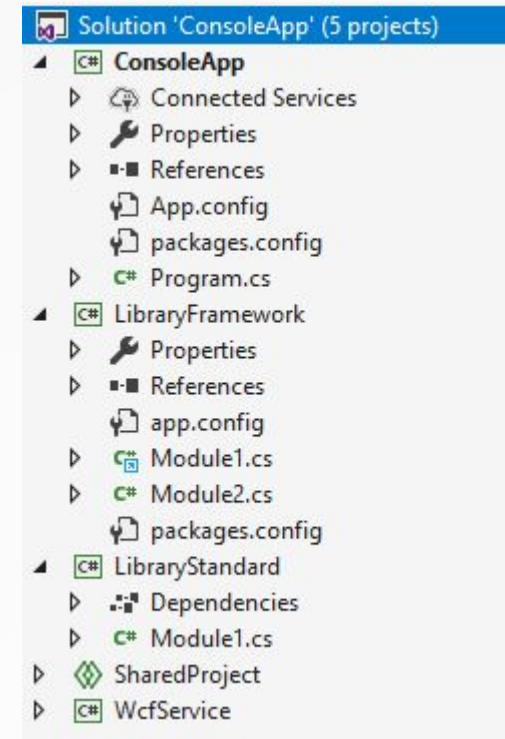
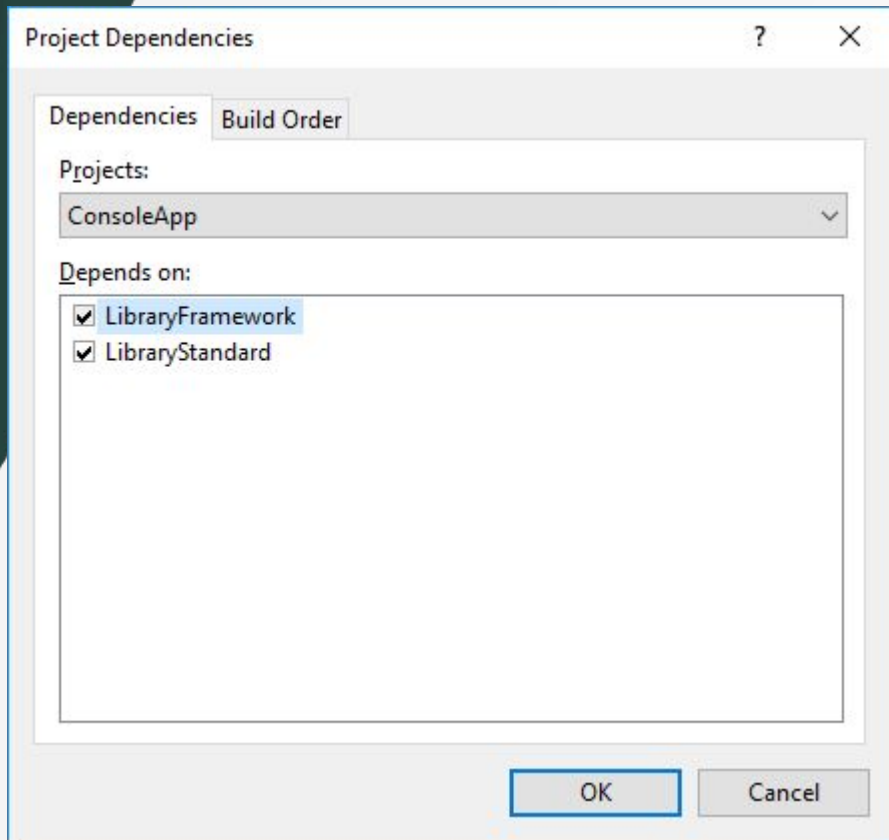


Circular references



Project Dependencies

When building a solution that contains multiple projects, it can be necessary to build certain projects first, to generate code used by other projects. When a project consumes executable code generated by another project, the project that generates the code is referred to as a project dependency of the project that consumes the code. Such dependency relationships can be defined in the Project Dependencies dialog box.



Common Mistakes

Solution 'ConsoleApp' Property Pages

Configuration: Active(Debug) Platform: Active(Any CPU) Configuration Manager...

Common Properties
Configuration Properties
Configuration

Project contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build	Deploy
ConsoleApp	Debug	Any CPU	<input checked="" type="checkbox"/>	<input type="checkbox"/>
LibraryFramework	Debug	x86	<input checked="" type="checkbox"/>	<input type="checkbox"/>
LibraryStandard	Debug	x64	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Exception Unhandled

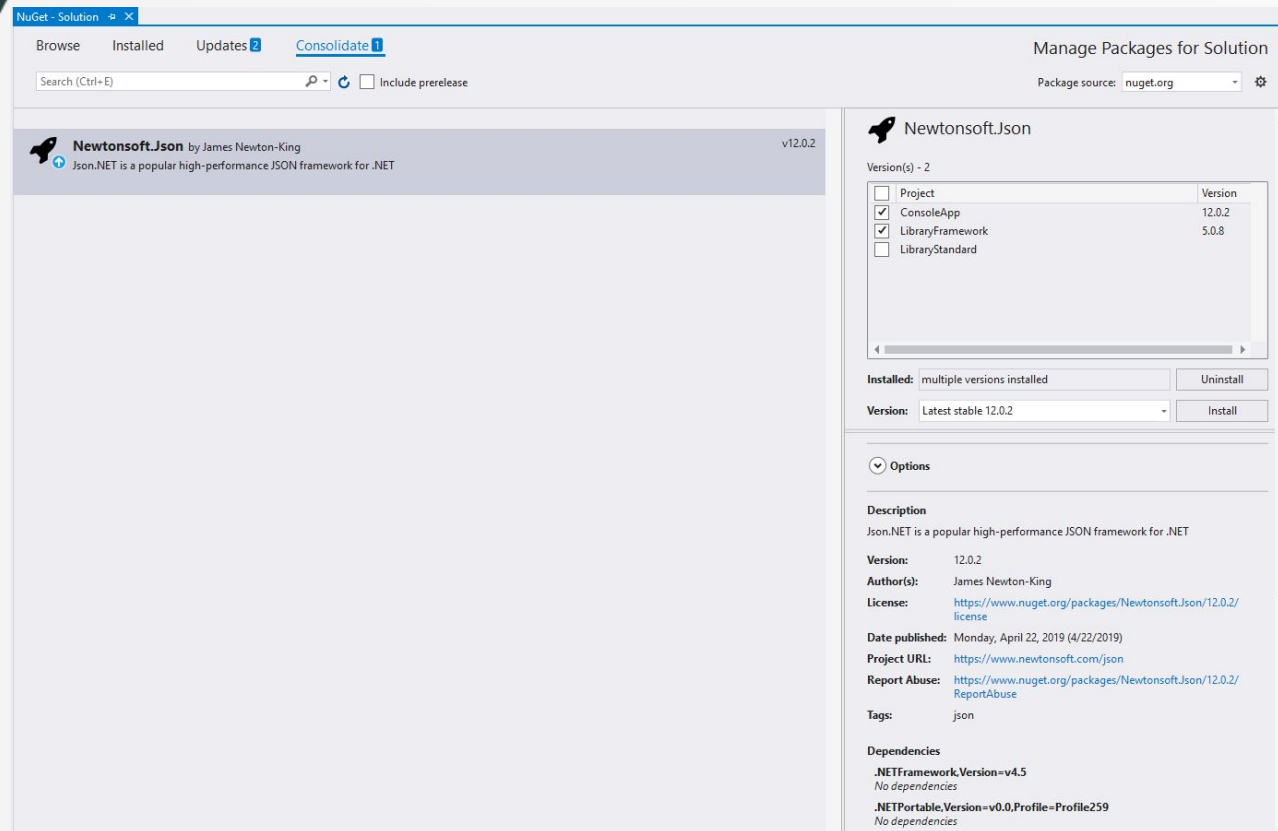
System.BadImageFormatException: 'Could not load file or assembly 'LibraryStandard, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null' or one of its dependencies. An attempt was made to load a program with an incorrect format.'

[Copy Details](#)
[Exception Settings](#)

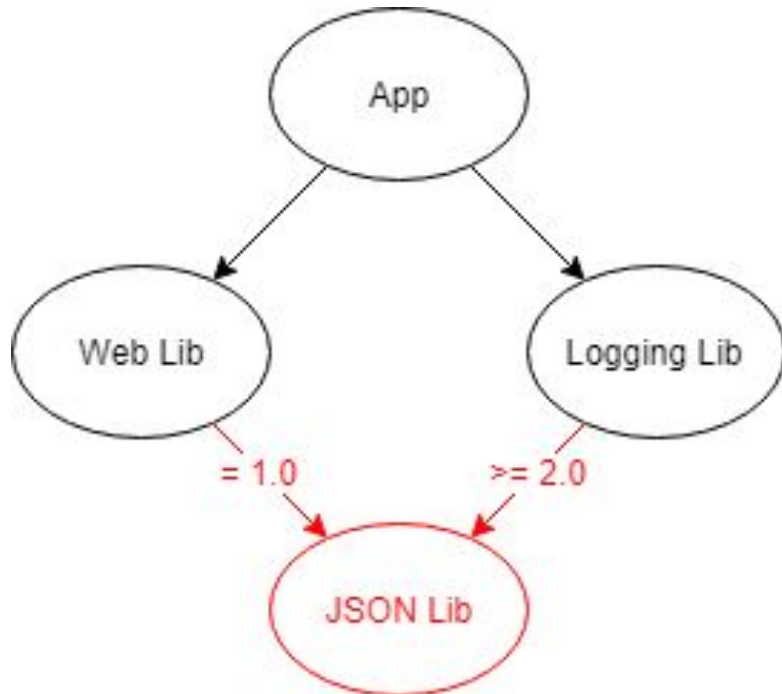
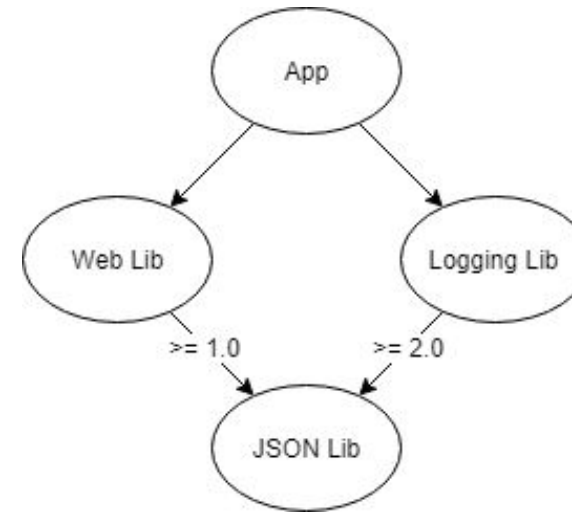
- ConsoleApp
 - Connected Services
 - Properties
 - References
 - Analyzers
 - Common
 - LibraryFramework
 - Microsoft.CSharp
 - Newtonsoft.Json
 - System
 - System.Core
 - System.Data
 - System.Data.DataSetExtensions
 - System.Net.Http
 - System.Xml
 - System.Xml.Linq
 - App.config
 - packages.config
 - Program.cs

Package Manager

The primary way of adding dependencies to a .NET library is referencing NuGet packages. NuGet package references allow you to quickly reuse and leverage already written functionality, but they're a common source of friction for .NET developers.



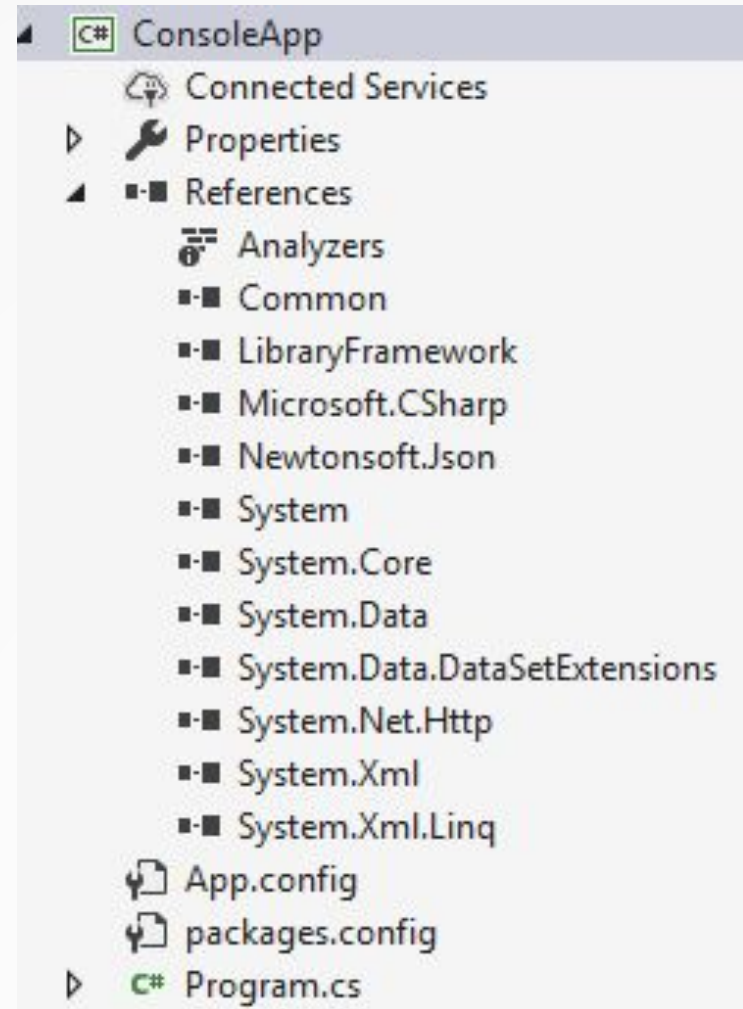
It's a common situation for a .NET project to have multiple versions of a package in its dependency tree. For example, an app depends on two NuGet packages, each of which depends on different versions of the same package.



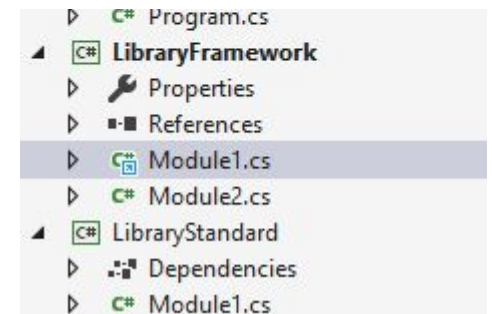
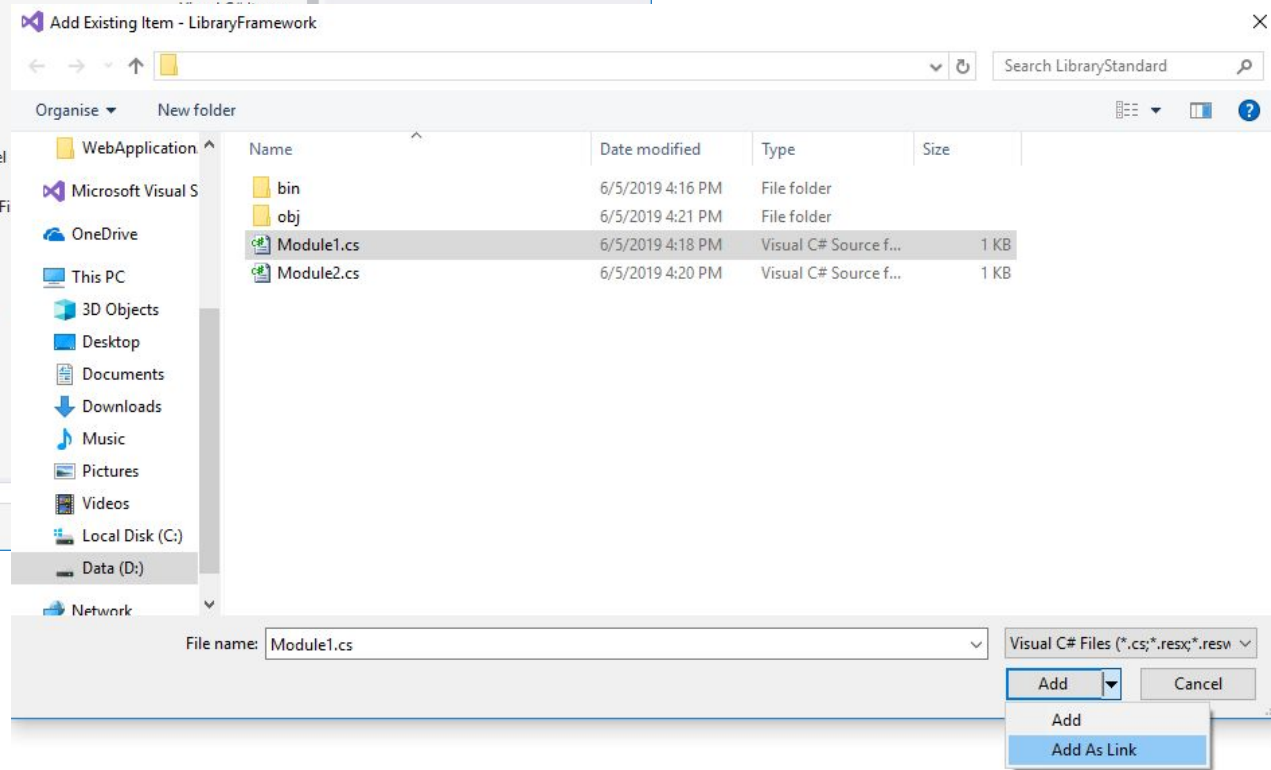
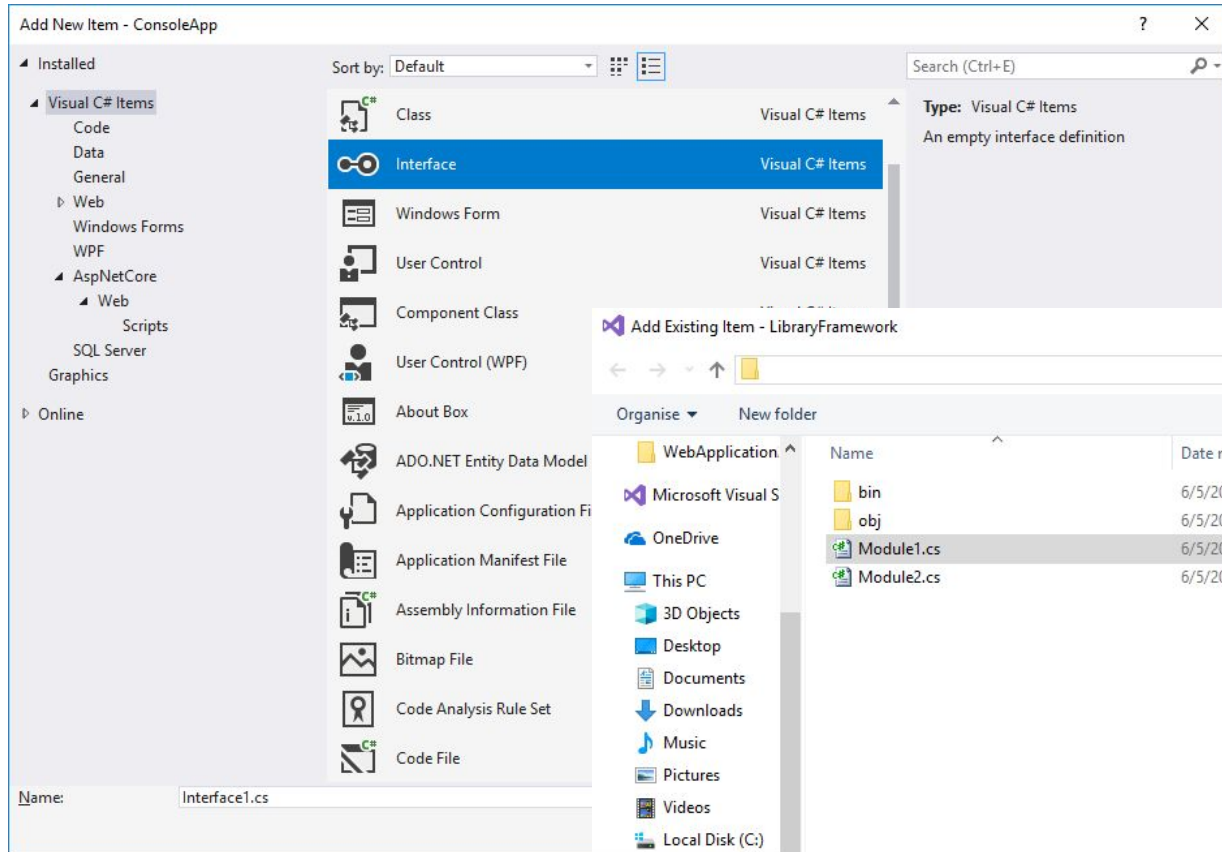
- > DO review your .NET library for unnecessary dependencies.
- > DO NOT have NuGet package references with no minimum version.
- > **AVOID NuGet package references that demand an exact version.**
- > AVOID NuGet package references with a version upper limit.
- > **NuGet always looks for the lowest applicable version. NuGet prefers the lowest applicable version over using the highest available because the lowest will have the least compatibility issues.**

References

```
<PlatformTarget>x64</PlatformTarget>
<ErrorReport>prompt</ErrorReport>
<CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
<Prefer32Bit>>true</Prefer32Bit>
</PropertyGroup>
<ItemGroup>
  <Reference Include="Common">
    <HintPath>..\Deploy\Common.dll</HintPath>
  </Reference>
  <Reference Include="Newtonsoft.Json, Version=12.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6ae"
    <HintPath>..\packages\Newtonsoft.Json.12.0.2\lib\net45\Newtonsoft.Json.dll</HintPath>
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="Microsoft.CSharp" />
  <Reference Include="System.Data" />
  <Reference Include="System.Net.Http" />
  <Reference Include="System.Xml" />
</ItemGroup>
<ItemGroup>
  <Compile Include="Program.cs" />
  <Compile Include="Properties\AssemblyInfo.cs" />
</ItemGroup>
<ItemGroup>
  <None Include="App.config" />
  <None Include="packages.config" />
</ItemGroup>
<ItemGroup>
  <ProjectReference Include="..\LibraryFramework\LibraryFramework.csproj">
    <Project>{8c224fe2-d18e-43c7-80df-f527a75a8580}</Project>
    <Name>LibraryFramework</Name>
  </ProjectReference>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
</Project>
```



Add New Item



Shared Project

```
Math.cs [X]
LibraryFramework SharedProject.Math
1 using System.Collections.Generic;
2 using System.Linq;
3
4 #if DecimalNumber
5 using Number = System.Double;
6 #endif
7 #if DoubleNumber
8 using Number = System.Decimal;
9 #endif
10
11 namespace SharedProject
12 {
13     public static class Math
14     {
15         public static (Number Min, Number Max) ValueRange(IEnumerable<Number> values)
16         {
17             if (!values.Any())
18                 return (0, 0);
19
20             return (values.Min(), values.Max());
21         }
22     }
23 }
24
```

LibraryFramework [X] Math.cs

Application Configuration: Active (Debug) Platform: Active (x86)

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

General

Conditional compilation symbols: DecimalNumber

Define DEBUG constant

Define TRACE constant

Platform target: x86



Thank you!