



.NET Migration Guide

GstarCAD 2024

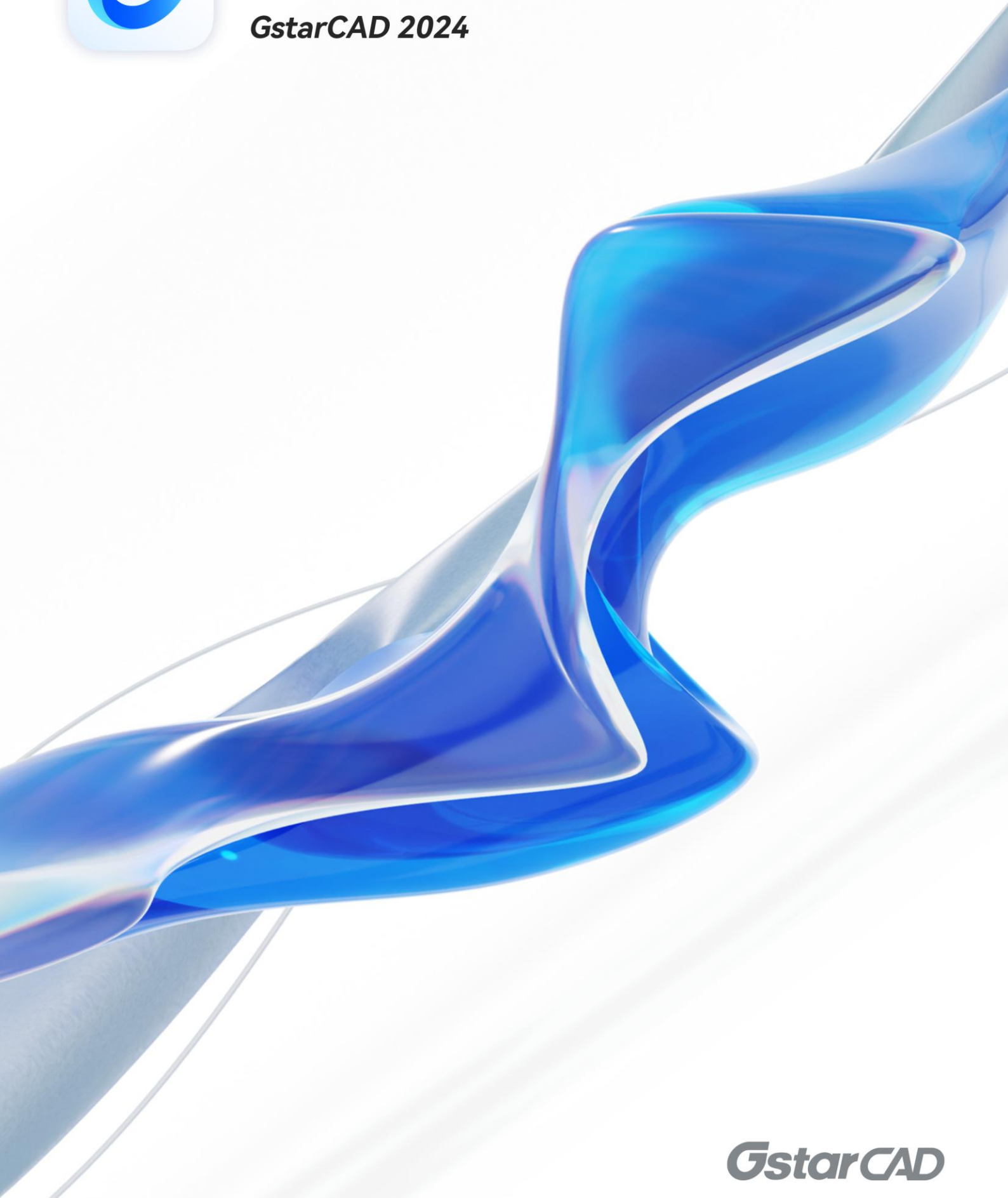


Table of Contents

1.	Introduction	4
2.	Programming Environment	5
3.	Requirement on Original Source Code.....	6
4.	Install GstarCAD SDK.....	7
5.	Visual Studio 2017 C# .NET Program Migration Process.....	8
5.1.	Change Reference File	8
5.1.1.	Remove Reference.....	8
5.1.2.	Add Reference.....	8
5.1.3.	Load Reference File	10
5.2.	Change Target Framework	11
5.3.	Modify Name in Namespace.....	12
5.4.	Compile Program	12
6.	Visual Studio 2017 VB .NET Program Migration Process.....	13
6.1.	Change Reference File	13
6.1.1.	Remove Reference.....	13
6.1.2.	Add Reference.....	13
6.1.3.	Load Reference File	14
6.2.	Change Target Framework	16
6.3.	Modify Name in Namespace.....	16
6.4.	Compile Program	16
7.	Special Usage of GstarCAD .NET Programming	17
7.1.	How to Use ' <i>using</i> '	17
7.2.	How to Use ' <i>ResultBuffer</i> '	17
7.3.	Alternative Method of Importing Unmanaged ARX Function	17
7.4.	Method of Selecting in Current Drawing after Clicking Button in Modal Dialog Box.....	17
7.5.	Other Usage	18
7.5.1.	Abtain GCAD Application Object under Visual Basic Environment.....	18
7.5.2.	C# .NET and VB .NET Namespace Modifaction.....	19
7.6.	Sample of Replacement	20

7.6.1.	Sample of Namespace Replacement.....	20
7.6.2.	Sample of Class Name and Property Name Replacement.....	20
8.	Copyright	21

1. Introduction

There are a large number of .NET applications based on AutoCAD®. With GstarCAD SDK, it is easy to migrate .NET application programs running on AutoCAD® to GstarCAD with little change to the original source codes, keeping the source-code-level compatibility.

2. Programming Environment

- Microsoft® Visual Studio Enterprise 2017 (Version 15.9.17)
- Windows SDK 10.0.17763.0
- CPU:
 - Basic: 1.6 GHz CUP
 - Recommended: 3.0 GHz CPU and above
- RAM:
 - Basic: 2 GB
 - Recommended: 8 GB and above
- Operation System (OS)
 - Windows 11
 - Windows 10 (version 1507 and above):
 - Home, Professional, Education and Enterprise (not support LTSC and Windows 10 S)
 - Windows 8.1 (with 2919355 update):
 - Core, Professional and Enterprise
 - Windows 7 SP1 (with the latest update):
 - Home, Professional, Enterprise and Ultimate
- Monitor Resolution:
 - 1028x800 and above true color display, including 4K (3840x2160) display
- GstarCAD SDK 2024
- GstarCAD 2024
- .NET Framework 4.8 and above

3. Requirement on Original Source Code

The original source codes should be .NET application source codes of .NET ObjectARX 2010 or higher versions.

4. Install GstarCAD SDK

Download GstarCAD SDK ('GRXSDK.ZIP' file) from GstarCAD website:

<https://www.gstarcad.net/download/>

Unzip GRXSDK.ZIP file to the local disk (e.g. 'C:\grxsdk') and there will be 5 directories generated (in 'C:\grxsdk') which are: **arx**, **inc**, **inc-x64**, **lib-x64** and **utils**.

arx contains the header files, library files and sample programs used for porting ARX programs to GRX programs. It contains the following directories:

- **Inc:** Header files used for porting from ARX to GRX
- **inc-x64:** Files used by COM and .NET (for 64-bit)
- **lib-x64:** GRX libraries (for 64-bit)
- **Samples:** Sample projects, including Dotnet, fact_dg, HelloADS, HelloA and SimplePalette.
 - **Dotnet:** .NET programming samples
 - 1) **Addline:** .NET programming sample of adding solid lines
 - 2) **Hello:** .NET programming sample of outputting prompt information
 - 3) **Vbhello:** Sample of .NET programming with VB .NET
 - **fact_dg:** Sample of LISP function definition
 - **HelloADS:** Sample of ADS programming
 - **HelloARX:** Sample of GRX programming
 - **SimplePalette:** Programming sample of how to create a set Palette windows
- **Utils:** Directory contains sub-directories of GRX extended applications, including APIs for extended function development, e.g. BREP for boundary representation.

Inc: Header files used for programming the GRX

inc-x64: Files used by COM and .NET (for 64-bit)

lib-x64: GRX libraries (for 64-bit)

Utils: Directory contains subdirectories of GRX extended applications, including APIs for extended function development, e.g. BREP for boundary representation.

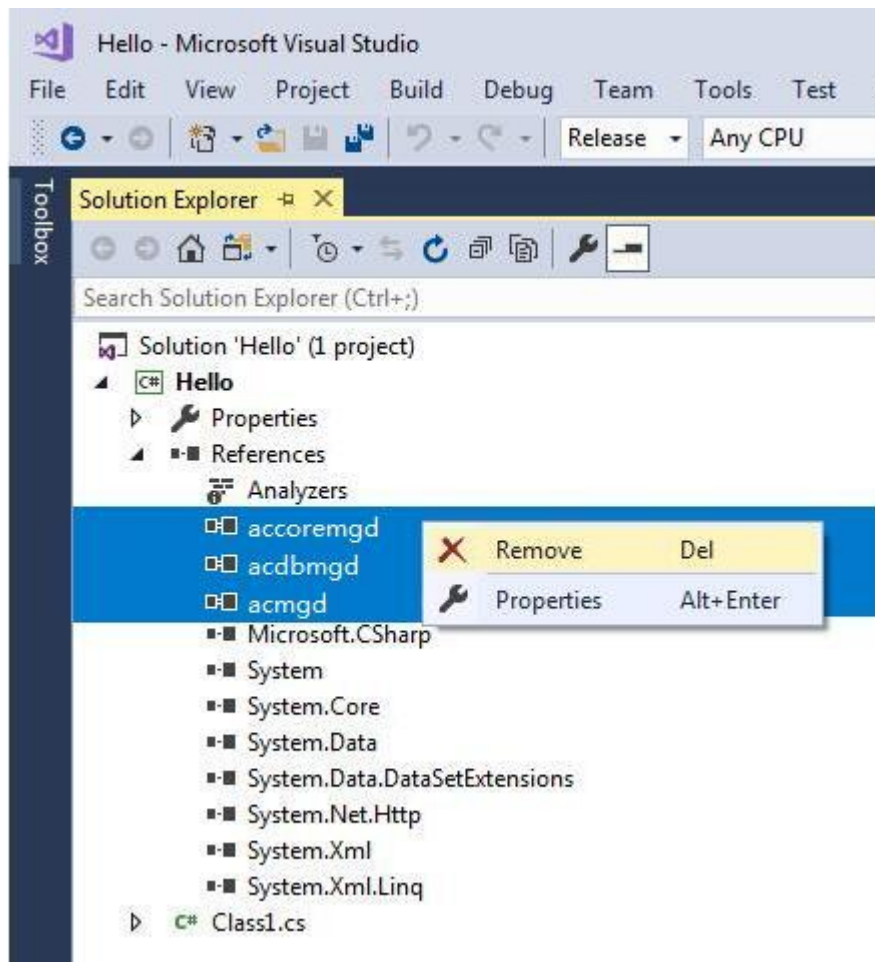
5. Visual Studio 2017 C# .NET Program Migration Process

The following illustrates the migration process by taking the 'Hello' project as an example, and assumes that the GstarCAD SDK is installed in 'C:\grxsdk'.

5.1. Change Reference File

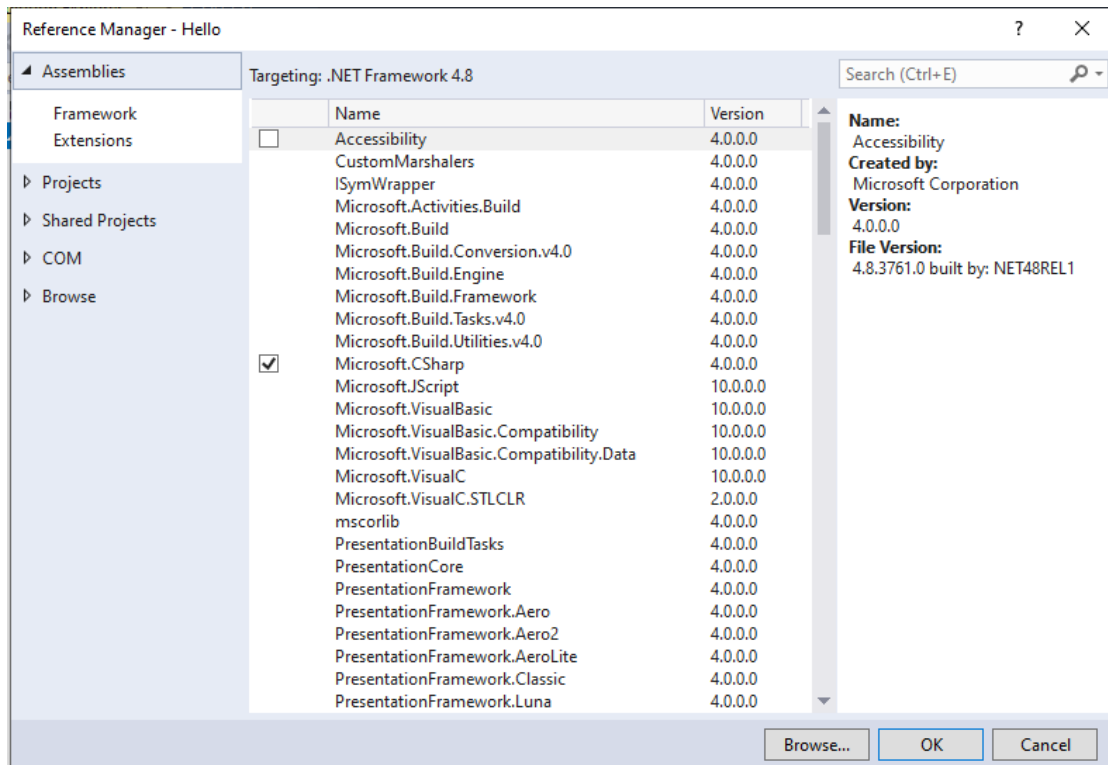
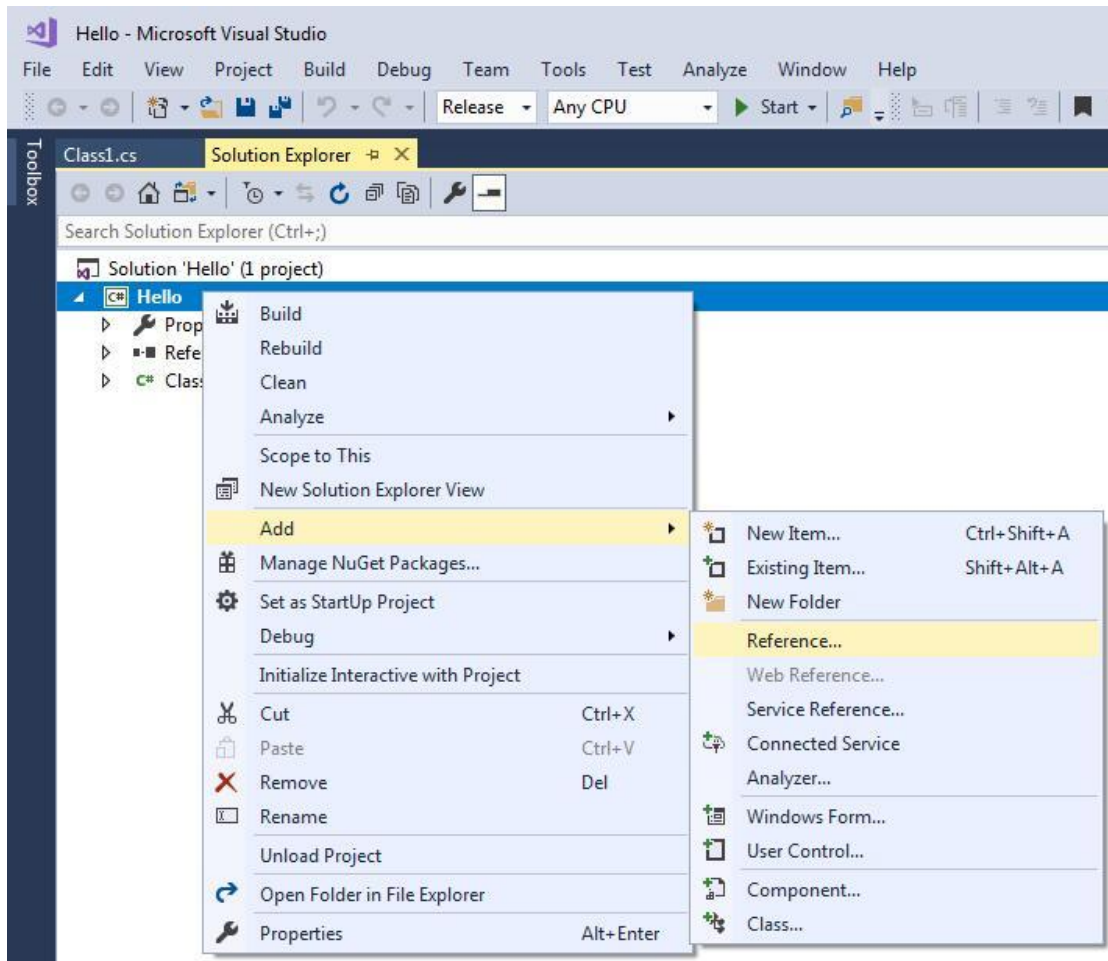
5.1.1. Remove Reference

From **Solution Explorer** of Visual Studio 2017, find project 'Hello' and remove relevant ARX reference files, as shown below.



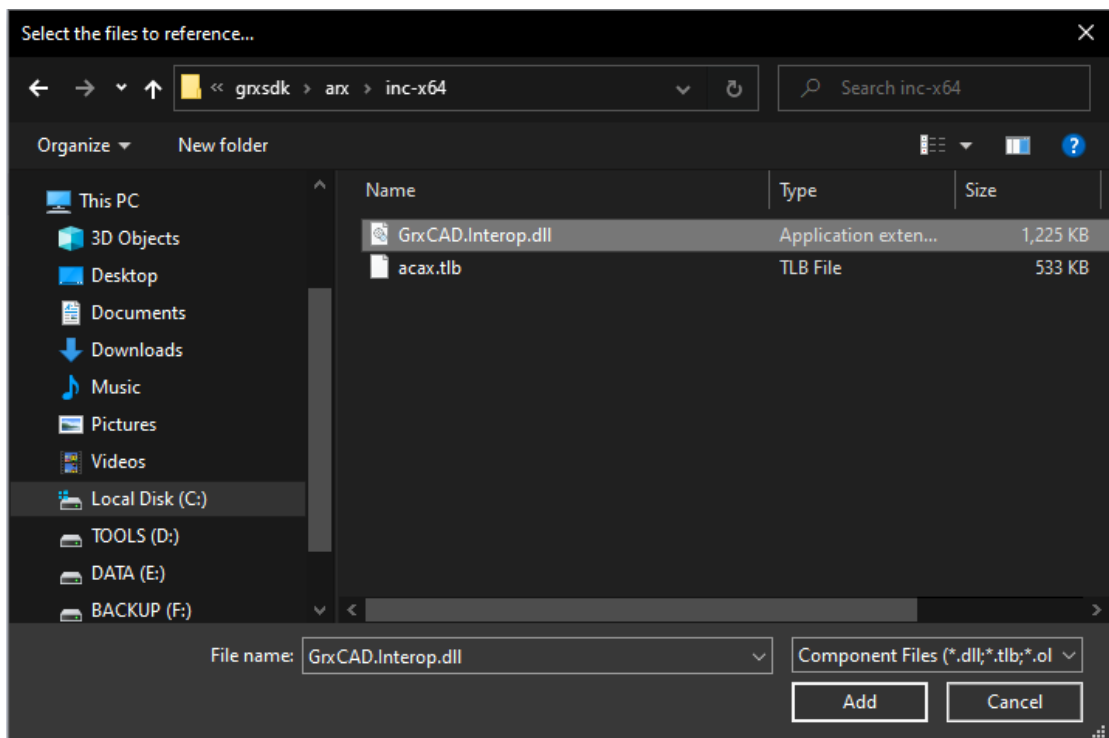
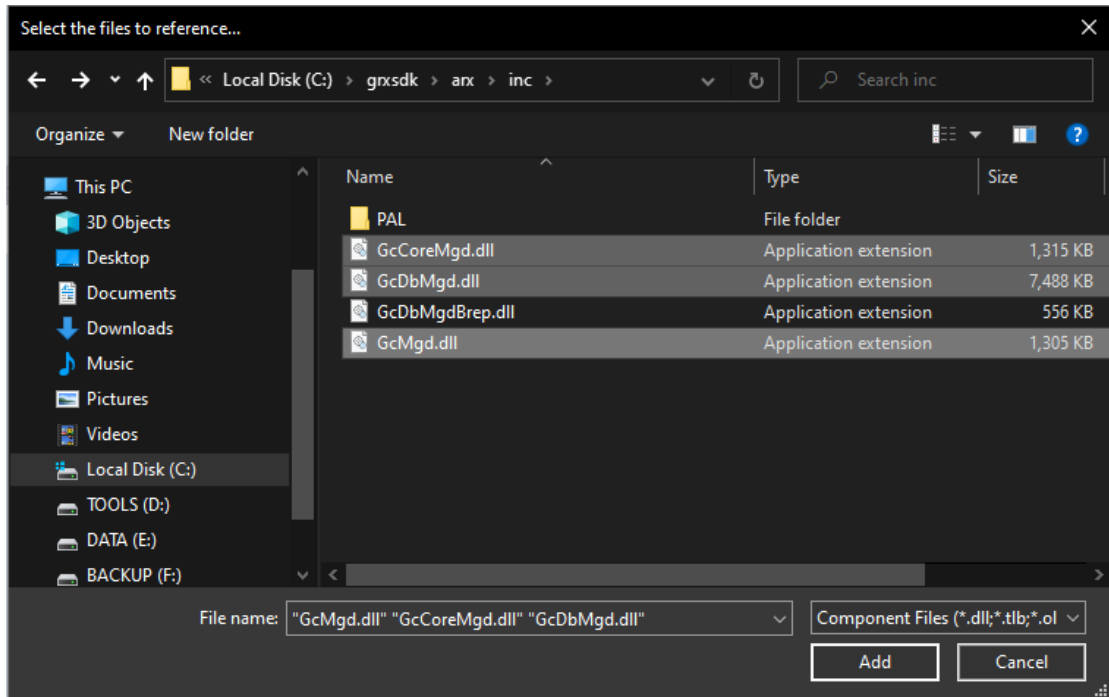
5.1.2. Add Reference

In **Solution Explorer** of Visual Studio 2017, right-click on project 'Hello' and select **Add Reference**. After the **Reference Manager** dialog window pops out, click **Browse...** button as shown below.



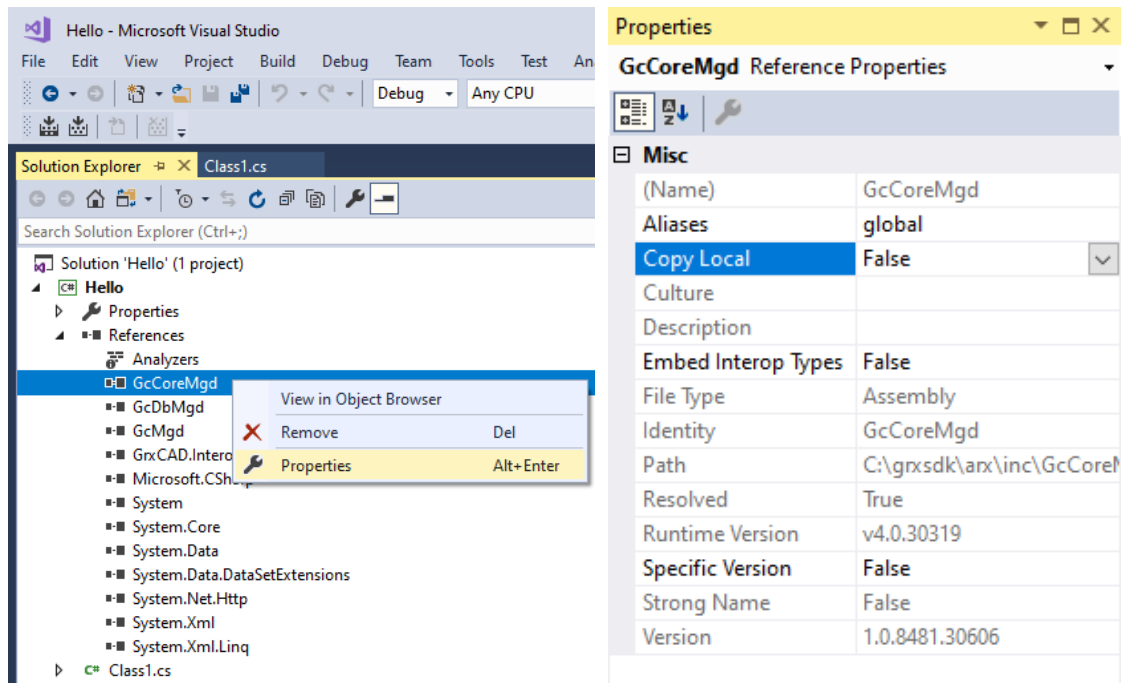
5.1.3. Load Reference File

From the installation path of GstarCAD SDK, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll' or 'acax.tlb' in '\arx\inc-x64', and add them by clicking **Add** button.



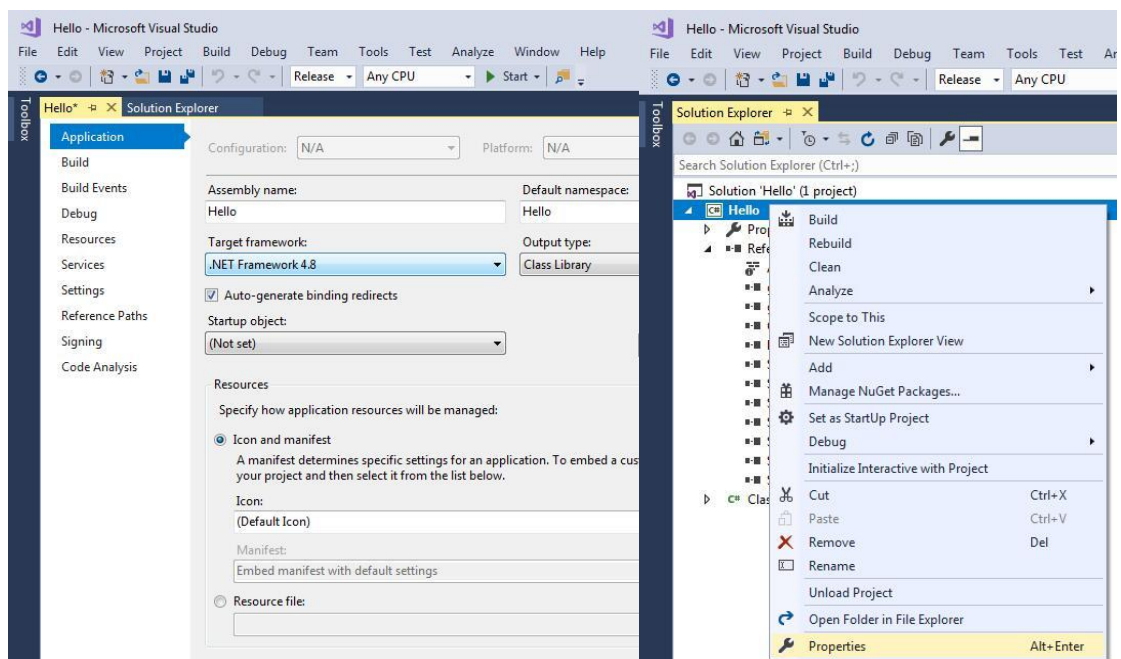
NOTE: 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' must be added while 'GrxCAD.Interop.dll' (or 'acax.tlb') is optional (needed when COM is used in the program).

After adding the files, right-click added 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' files one by one and select **Properties** at the context menu, the **Reference Properties** palette pops up. Set the value of **Copy Local** property to **False**. The following screenshot shows how to change **Copy Local** property of 'GcCoreMgd.dll'.



5.2. Change Target Framework

In **Solution Explorer** of Visual Studio 2017, right-click on project 'Hello' and select **Properties**. After the dialog window pops out, select **Application** and change the **Target framework** to **.NET Framework 4.8** from the pull-down list.



5.3. Modify Name in Namespace

Replace the *'Autodesk.AutoCAD'* with corresponding *'Gssoft.Gscad'* in .NET source codes.

5.4. Compile Program

In Visual Studio 2017, click **Build**→**Build Solution** to regenerate the DLL file in corresponding directory. Run GstarCAD and input *'netload'* at command line, select the relative DLL file from **Select .Net Assembly** dialog window and click **Open** button to load it.

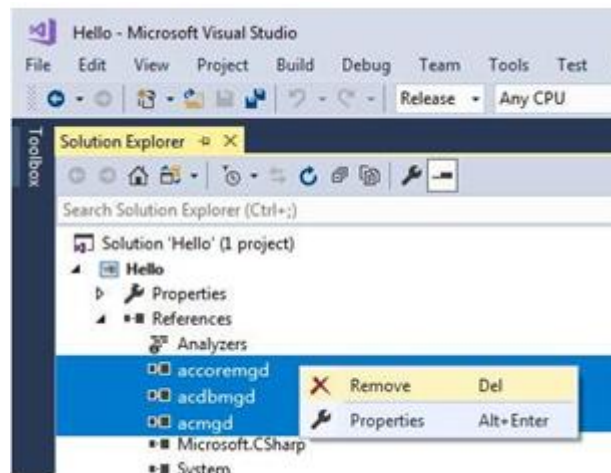
6. Visual Studio 2017 VB .NET Program Migration Process

The following illustrates the migration process by taking the 'Hello' project as an example, and assumes that the GstarCAD SDK is installed in 'C:\grxsdk'.

6.1. Change Reference File

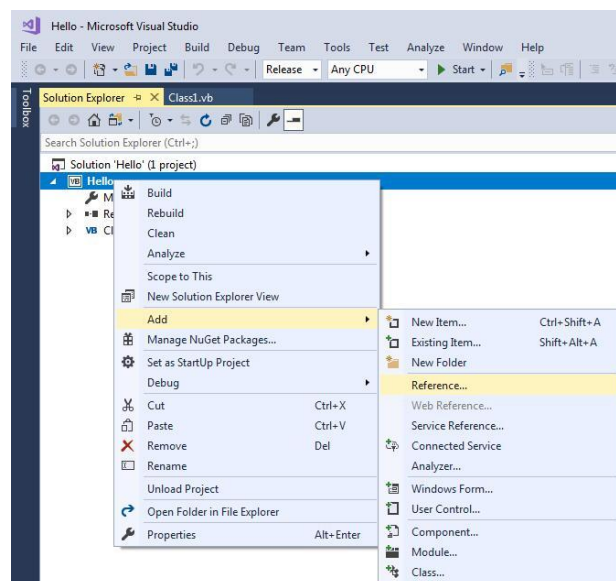
6.1.1. Remove Reference

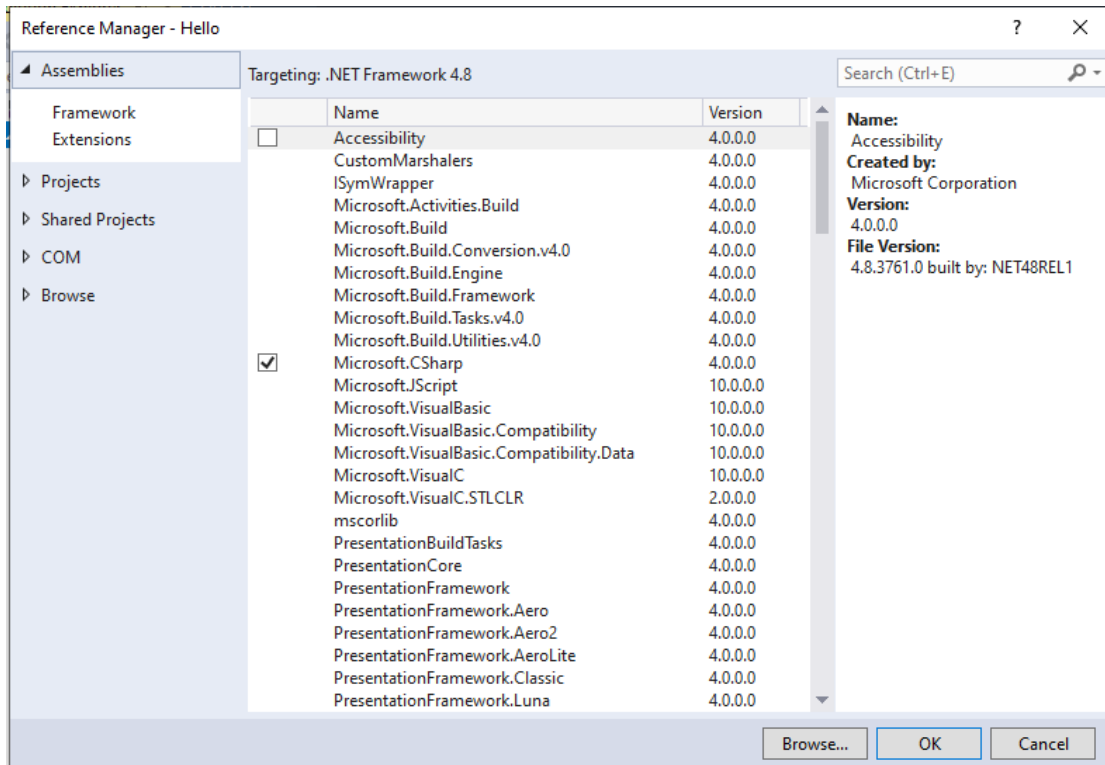
From **Solution Explorer** of Visual Studio 2017, find project 'Hello' and remove relevant ARX reference files, as shown below.



6.1.2. Add Reference

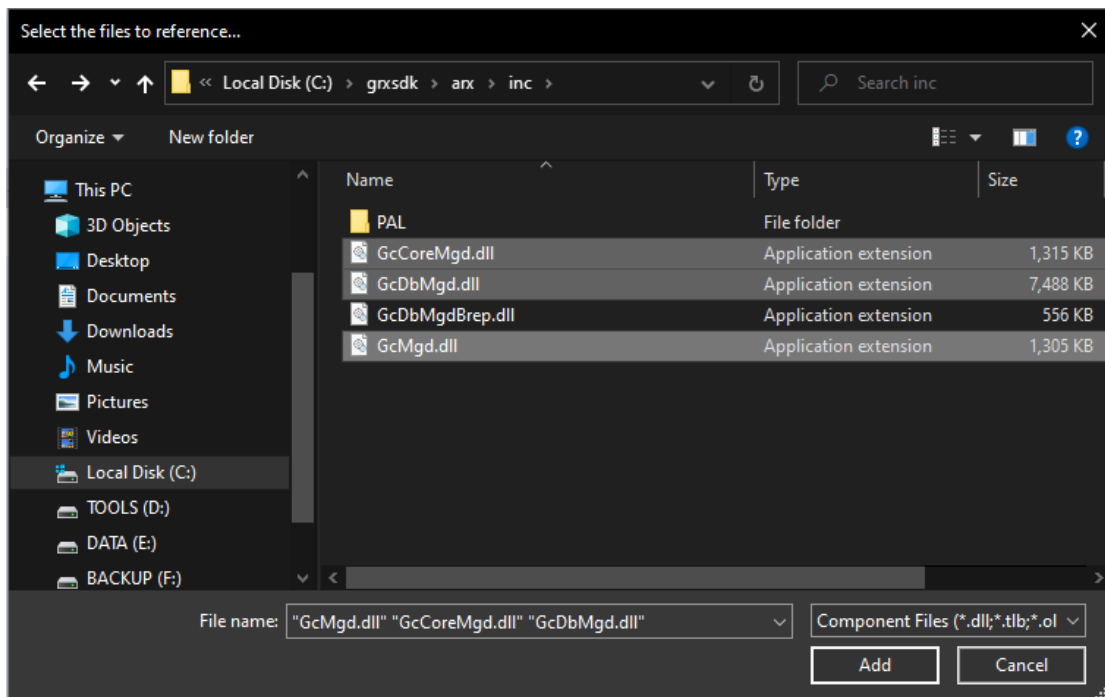
In **Solution Explorer** of Visual Studio 2017, right-click on project 'Hello' and select **Add Reference**. After the **Reference Manager** dialog window pops out, click **Browse...** button as shown below.

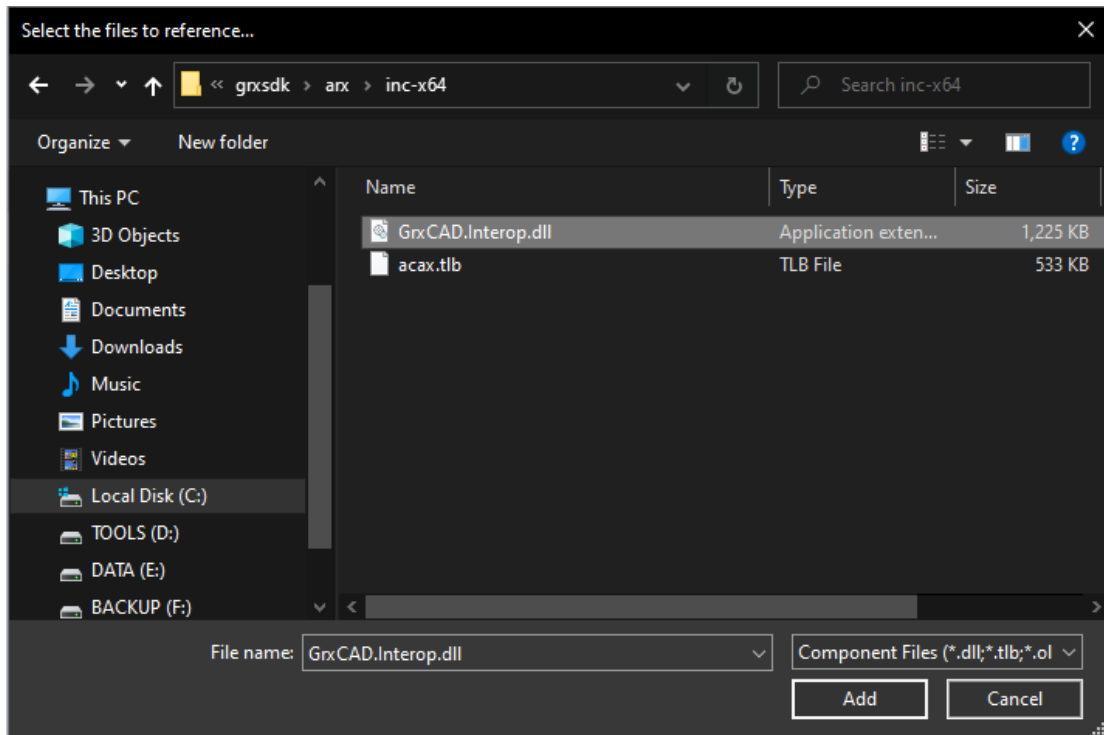




6.1.3. Load Reference File

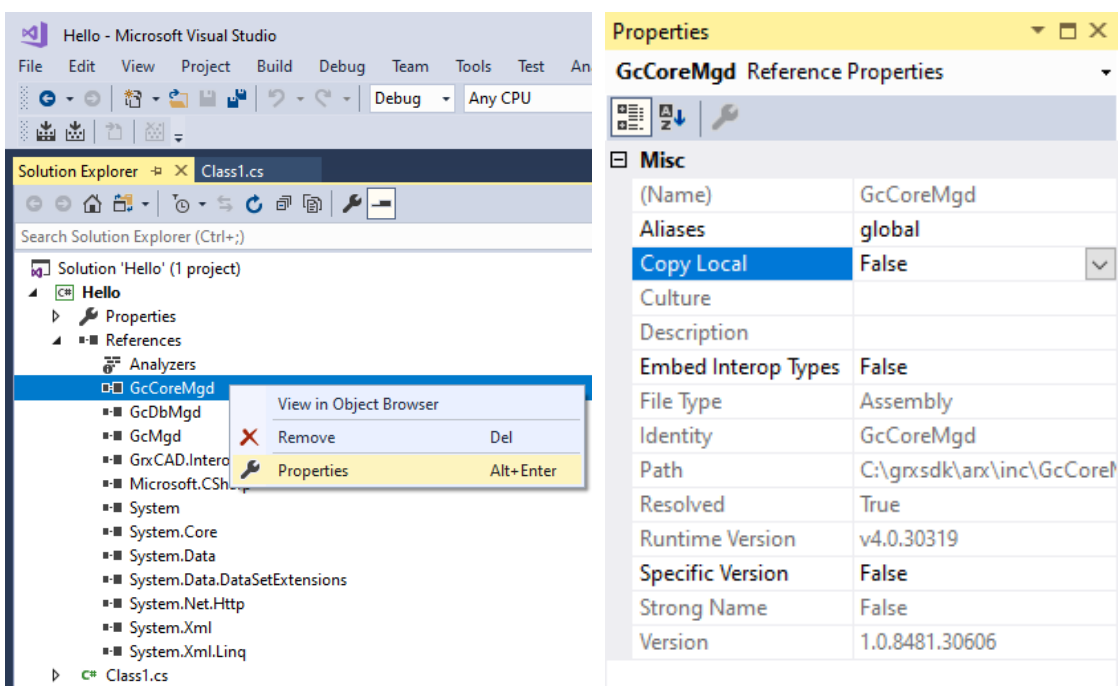
From the installation path of GstarCAD SDK, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll' or 'acax.tlb' in '\arx\inc-x64', and add them by clicking **Add** button.





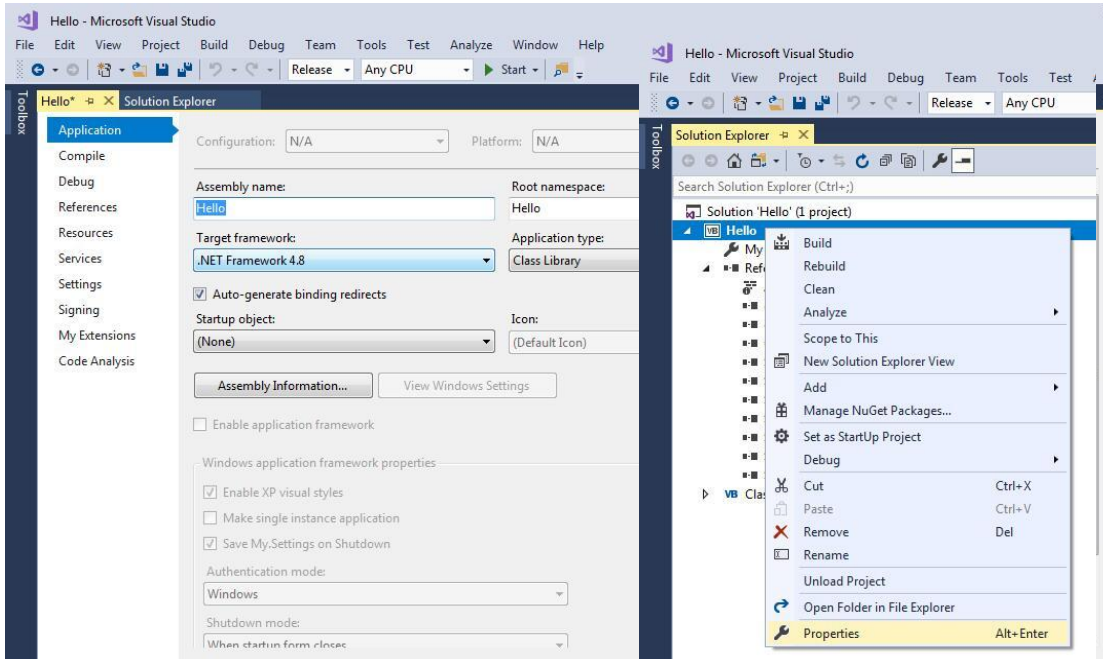
NOTE: 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' must be added while 'GrxCAD.Interop.dll' (or 'acax.tlb') is optional (needed when COM is used in the program).

After adding the files, right-click added 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' files one by one and select **Properties** at the context menu, the **Reference Properties** palette pops up. Set the value of **Copy Local** property to **False**. The following screenshot shows how to change **Copy Local** property of 'GcCoreMgd.dll'.



6.2. Change Target Framework

In **Solution Explorer** of Visual Studio 2017, right-click on project '*Hello*' and select **Properties**. After the dialog window pops out, select **Application** and change the **Target framework** to **.NET Framework 4.8** from the pull-down list.



6.3. Modify Name in Namespace

Replace the '*Autodesk.AutoCAD*' with corresponding '*Gssoft.Gscad*' in .NET source codes.

6.4. Compile Program

In Visual Studio 2017, click **Build**→**Build Solution** to regenerate the DLL file in corresponding directory. Run GstarCAD and input '*netload*' at command line, select the relative DLL file from **Select .Net Assembly** dialog window and click **Open** button to load it.

7. Special Usage of GstarCAD .NET Programming

The following describes the differences between GstarCAD .NET programming and AutoCAD® .NET programming.

7.1. How to Use 'using'

Codes like '*var mapping = new IdMapping()*' need to add '*using*' function first, e.g.:

```
using( var mapping = new IdMapping())  
{  
    ...  
}
```

7.2. How to Use 'ResultBuffer'

Use '*ResultBuffer.ResbufObject*' to replace '*ResultBuffer.UnmanagedObject*';

Type of *ResultBuffer.UnmanagedObject* is *GcResbuf*;

Type of *ResultBuffer.ResbufObject* is *resbuf*;

Use '*ResultBuffer.Create(IntPtr,bool)*' to replace '*ResultBuffer(IntPtr,bool)*'

7.3. Alternative Method of Importing Unmanaged ARX Function

Use '*Gssoft.Gscad.EditorInput.Editor.Command(params object[])*' to replace the unmanaged ARX functions *acedCmd* and *acedCommand*.

7.4. Method of Selecting in Current Drawing after Clicking Button in Modal Dialog Box

Select a point or entity in the current drawing after clicking the button in the modal dialog box, using:

using (*var edUI = doc.Editor.StartUserInteraction(this)*), e.g.:

```
private void button1_Click(object sender, EventArgs e)  
{  
    Document doc = AcadApp.DocumentManager.MdiActiveDocument;  
    Editor ed = AcadApp.DocumentManager.MdiActiveDocument.Editor;  
  
    using (var edUI = doc.Editor.StartUserInteraction(this))  
    {
```

```

        PromptPointOptions ppo = new PromptPointOptions("\nSelect point:");
        PromptPointResult ppr = ed.GetPoint(ppo);

        PromptSelectionOptions pso = new PromptSelectionOptions();
        PromptSelectionResult psr = ed.GetSelection(pso);
    }
}

```

7.5. Other Usage

7.5.1. Abtain GCAD Application Object under Visual Basic Environment

By adding '*C:\grxsdk\arx\inc-x64\acax.tlb*' as reference, it is possible to obtain GCAD application objects with following codes.

```

Imports Gsoft.Gscad.Runtime
Imports GcadVbaLib

```

```

Public Class Class1

```

```

    <CommandMethod("GetGcadApplication")>

```

```

    Public Sub GetGcadApplication()

```

```

        'Define abtain GcadApplication object, the object is null

```

```

        Dim GCadApp As GcadApplication

```

```

        ' Abtain the object address through the process name.

```

```

        GCadApp = GetObject("gcad.application")

```

```

        'Minimize the abtained process window

```

```

        GCadApp.WindowState = GcWindowState.acMin

```

```

    End Sub

```

```

End Class

```

7.5.2. C# .NET and VB .NET Namespace Modification

- VB .NET:
Change '*Autodesk.AutoCAD*' to '*Gssoft.Gscad*' after the '*Imports*'
- C# .NET:
Change '*Autodesk.AutoCAD*' to '*Gssoft.Gscad*' after the '*using*'
- COM With VB .NET:
Add '*Imports GcadVbaLib*' and delete '*Autodesk.AutoCAD.Interop.Common*'
- COM With C# .NET:
Add '*using GcadVbaLib*' and delete '*Autodesk.AutoCAD.Interop.Common*'

For the COM object name in the codes, such as *AcadLWPolyline*, it is needed to change the prefix '*Acad*' to '*Gcad*', e.g. '*AcadLWPolyline*' should be change to '*GcadLWPolyline*'.

If it's not sure a COM object exists or not, then first check it from the object browser, e.g. for *ACAD_COLOR*, check whether there is corresponding *GCAD_COLOR* first from the browser. Then change '*ACAD_COLOR*' to '*GCAD_COLOR*' if *GCAD_COLOR* is found.

7.6. Sample of Replacement

7.6.1. Sample of Namespace Replacement

AutoCAD®	GstarCAD
<i>Autodesk.AutoCAD.ApplicationServices</i>	<i>Gssoft.Gscad.ApplicationServices</i>
<i>Autodesk.AutoCAD.EditorInput</i>	<i>Gssoft.Gscad.EditorInput</i>
<i>Autodesk.AutoCAD.GraphicsSystem</i>	<i>Gssoft.Gscad.GraphicsSystem</i>
<i>Autodesk.AutoCAD.PlottingServices</i>	<i>Gssoft.Gscad.PlottingServices</i>
<i>Autodesk.AutoCAD.Publishing</i>	<i>Gssoft.Gscad.Publishing</i>
<i>Autodesk.AutoCAD.Runtime</i>	<i>Gssoft.Gscad.Runtime</i>
<i>Autodesk.AutoCAD.Windows</i>	<i>Gssoft.Gscad.Windows</i>
<i>Autodesk.AutoCAD.Colors</i>	<i>Gssoft.Gscad.Colors</i>
<i>Autodesk.AutoCAD.DatabaseServices</i>	<i>Gssoft.Gscad.DatabaseServices</i>
<i>Autodesk.AutoCAD.DatabaseServices.Filters</i>	<i>Gssoft.Gscad.DatabaseServices.Filters</i>
<i>Autodesk.AutoCAD.Geometry</i>	<i>Gssoft.Gscad.Geometry</i>
<i>Autodesk.AutoCAD.GraphicsInterface</i>	<i>Gssoft.Gscad.GraphicsInterface</i>
<i>Autodesk.AutoCAD.LayerManager</i>	<i>Gssoft.Gscad.LayerManager</i>

7.6.2. Sample of Class Name and Property Name Replacement

AutoCAD®	GstarCAD
<i>Autodesk.AutoCAD.Interop.Common.ACAD_COLOR</i>	<i>GcadVbaLib.GCAD_COLOR</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadEntity</i>	<i>GcadVbaLib.GcadEntity</i>
<i>Autodesk.AutoCAD.Interop.Common.AcSelect</i>	<i>GcadVbaLib.GcSelect</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadRasterImage</i>	<i>GcadVbaLib.GcadRasterImage</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadText</i>	<i>GcadVbaLib.GcadText</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadMText</i>	<i>GcadVbaLib.GcadMText</i>
<i>Autodesk.AutoCAD.Interop.Common.AcWindowState</i>	<i>GcadVbaLib.GcWindowState</i>

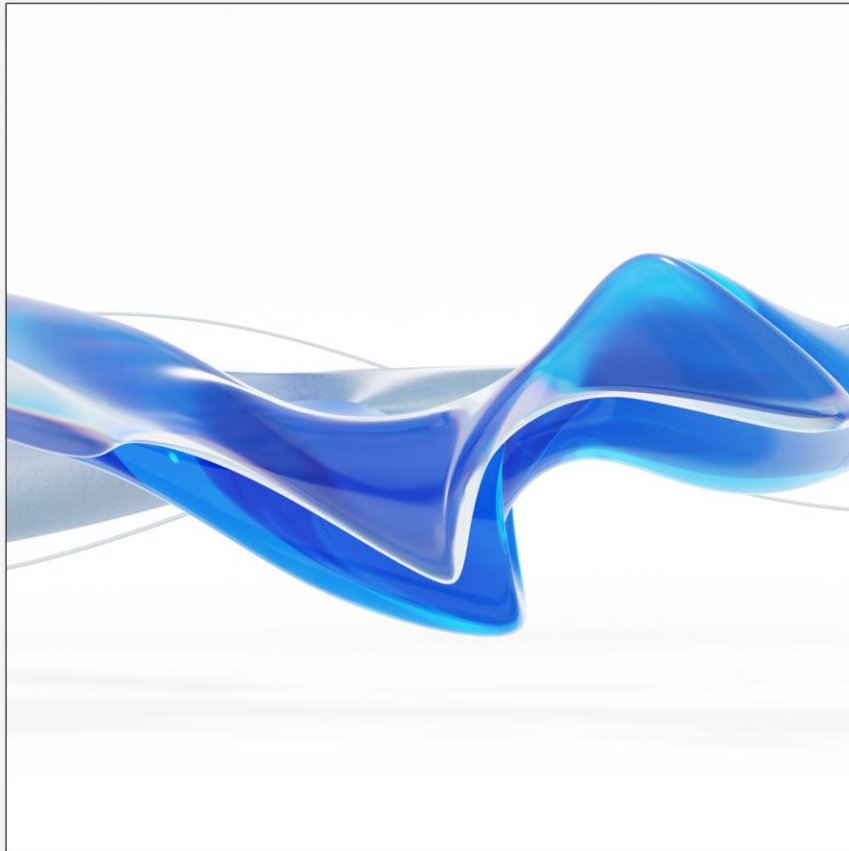
8. Copyright

Copyright reserved: Gstarsoft Co.,Ltd

Copying and referencing any part of this document is allowed. No part of this document may be changed without permission. Please keep this statement when copying or referencing this document.



GstarCAD 2024



■ <https://www.gstarcad.net/>