



.NET Migration Guide

GstarCAD 2027

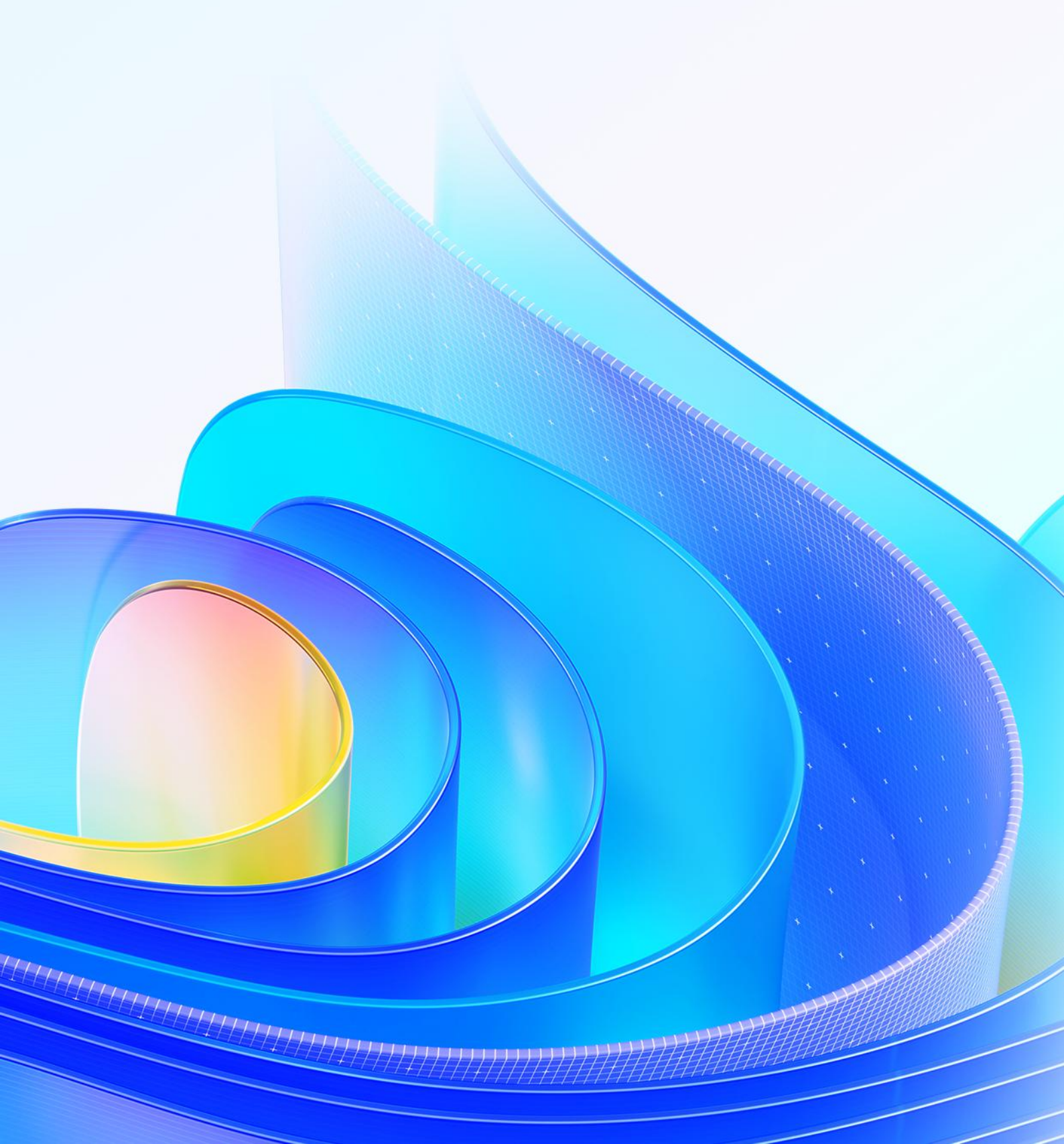


Table of Contents

1. Introduction	1
2. Programming Environment	2
3. Requirement on Original Source Code	3
4. Install SDK	4
5. Visual Studio 2022 C# .NET Program Migration Process	5
5.1. Change Reference File	5
5.1.1. Remove Reference	5
5.1.2. Add Reference	5
5.1.3. Load Reference File	7
5.2. Modify Name in Namespace	9
5.3. Compile Program	9
6. Visual Studio 2022 VB .NET Program Migration Process	10
6.1. Change Reference File	10
6.1.1. Remove Reference	10
6.1.2. Add Reference	10
6.1.3. Load Reference File	11
6.2. Modify Name in Namespace	13
6.3. Compile Program	13
7. Other Usage	14
7.1. Abtain GCAD Application Object under Visual Basic Environment	14
7.2. C# .NET and VB .NET Namespace Modification	14
7.3. Sample of Replacement	15
7.3.1. Sample of Namespace Replacement	15
7.3.2. Sample of Class Name and Property Name Replacement	15
8. Copyright	16

1. Introduction

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

2. Programming Environment

- Microsoft® Visual Studio 2022 (version 17.8.0)
- Windows SDK 10.0 (latest installed version)
- CPU:
 - Basic: 1.6 GHz CPU
 - 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)
- Monitor Resolution:
 - 1028x800 and above true color display, including 4K (3840x2160) display
- GRXSDK
- Microsoft .NET 8.0

3. Requirement on Original Source Code

The original source codes should be .NET application source codes of .NET ObjectARX 2025.

4. Install SDK

Download SDK.

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, HelloARX 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.

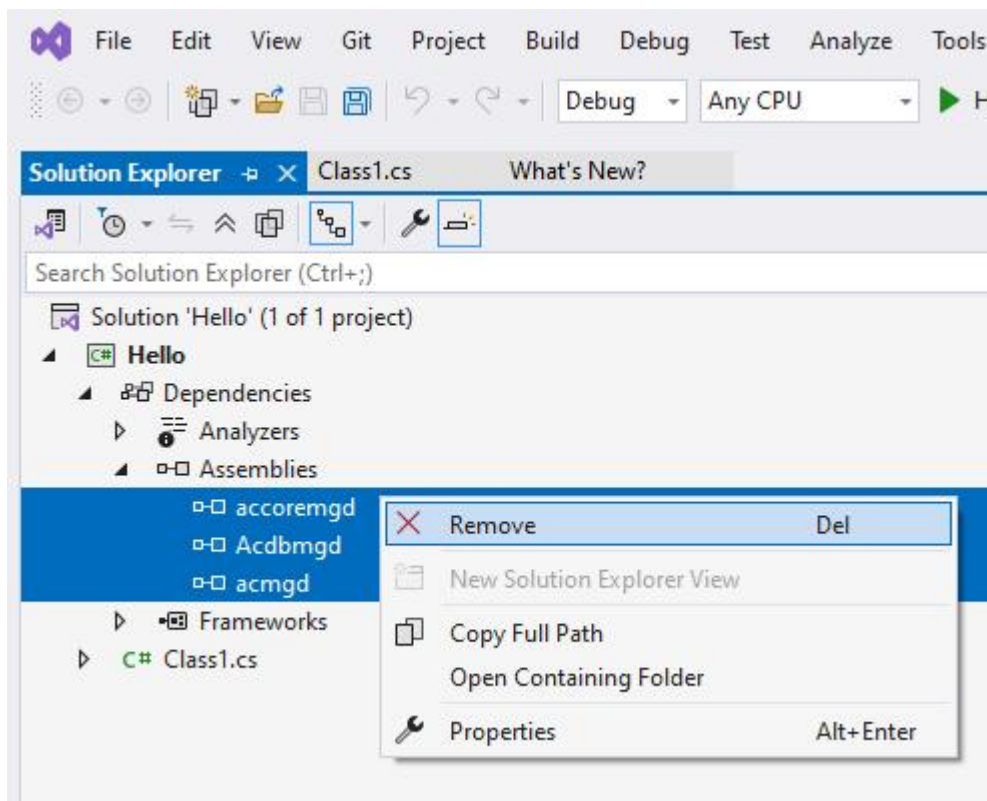
5. Visual Studio 2022 C# .NET Program Migration Process

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

5.1. Change Reference File

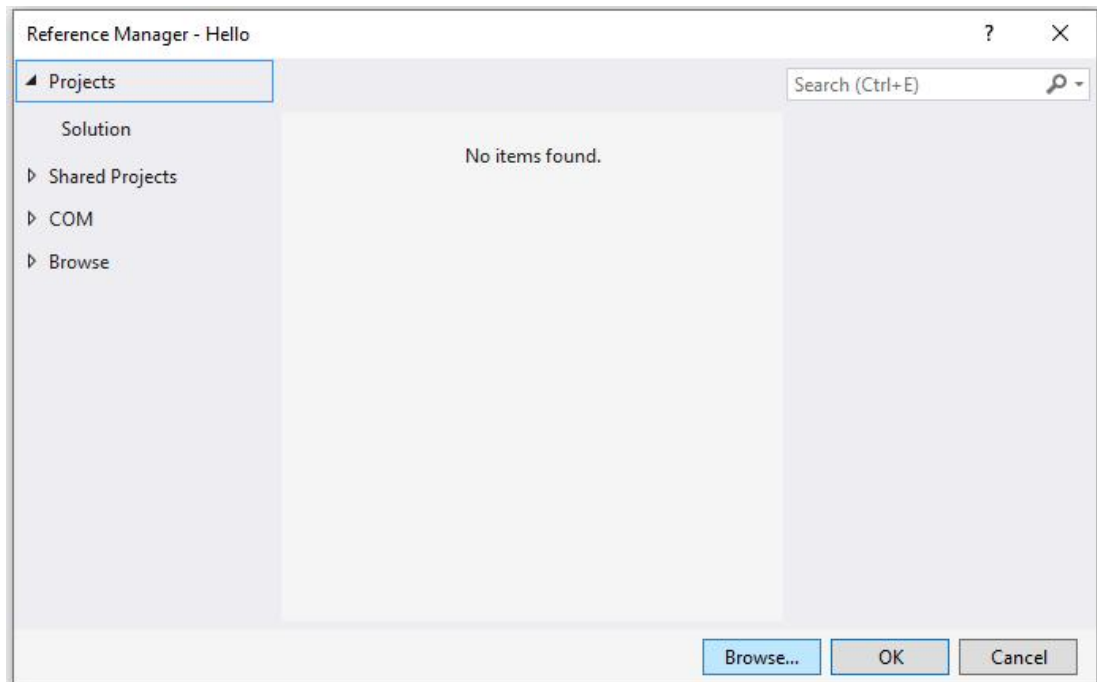
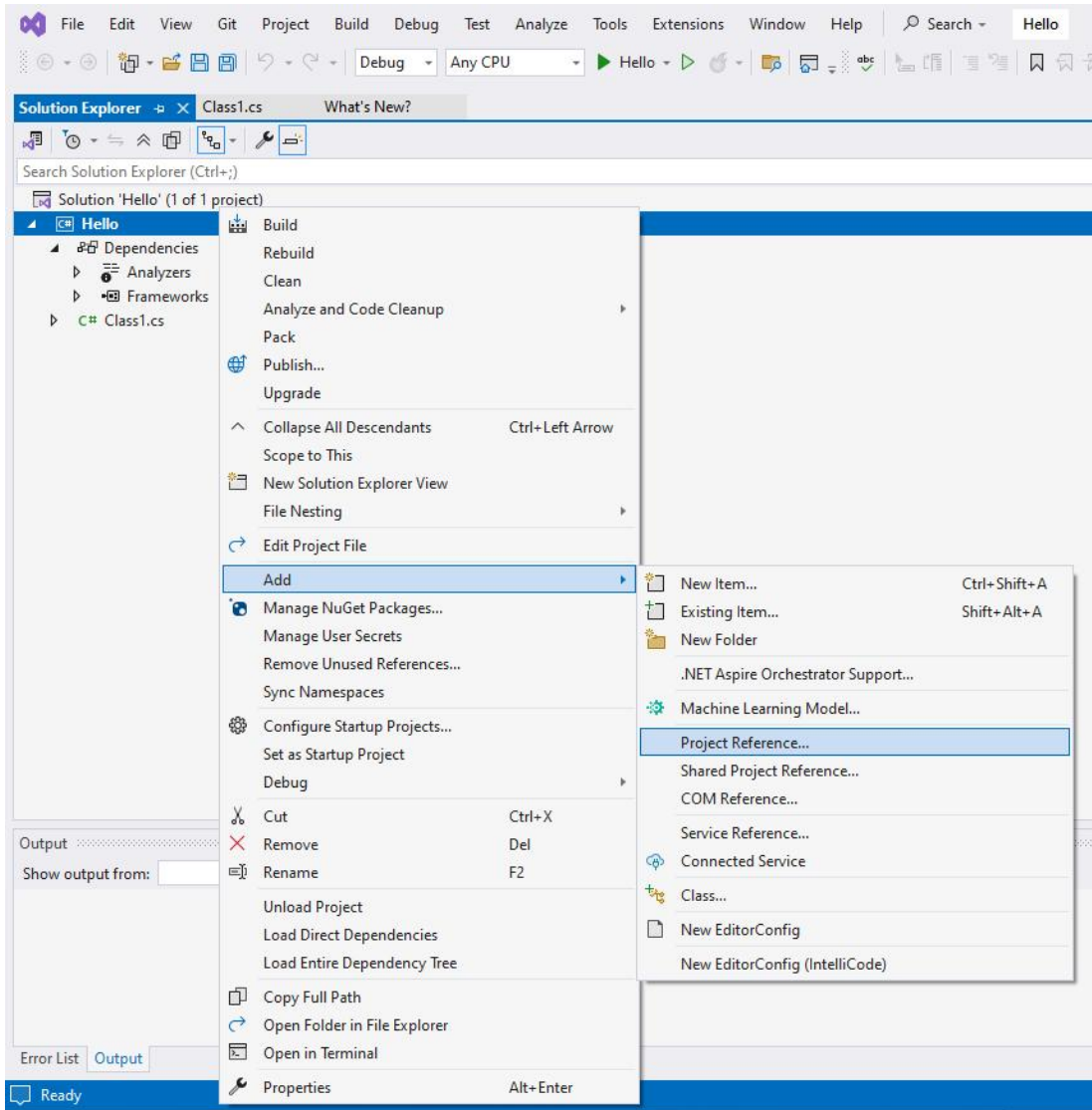
5.1.1. Remove Reference

From **Solution Explorer** of Visual Studio 2022, find project 'Hello' and remove relevant ARX reference files, as shown below.



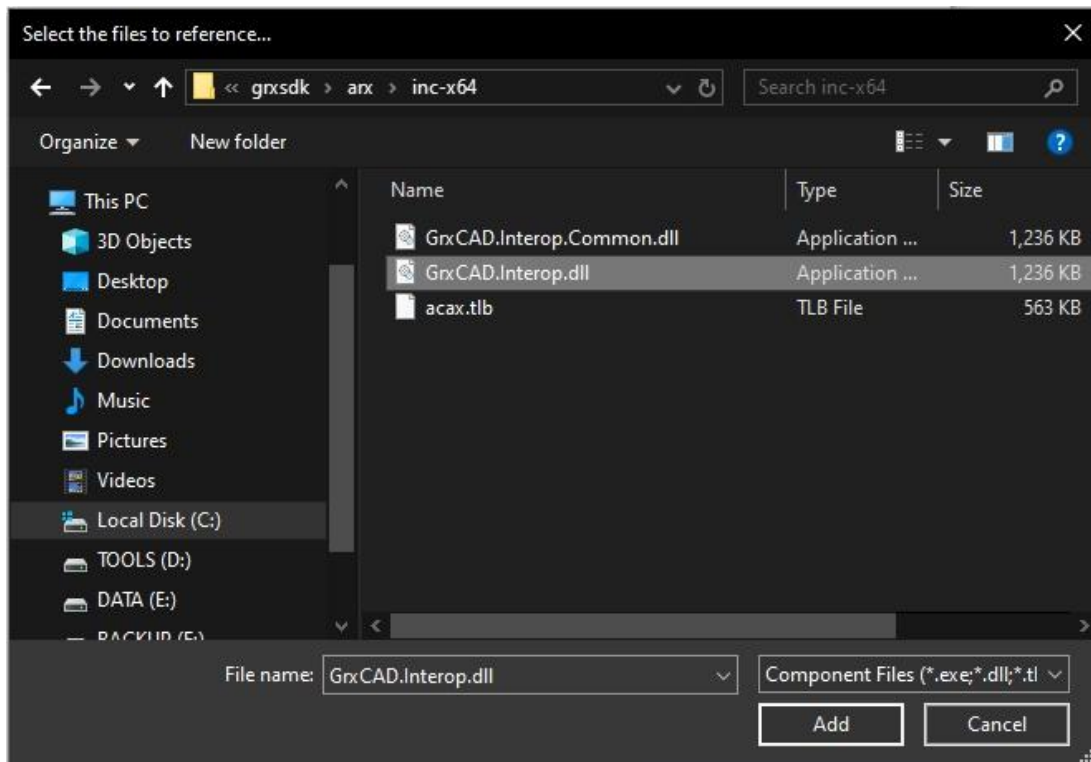
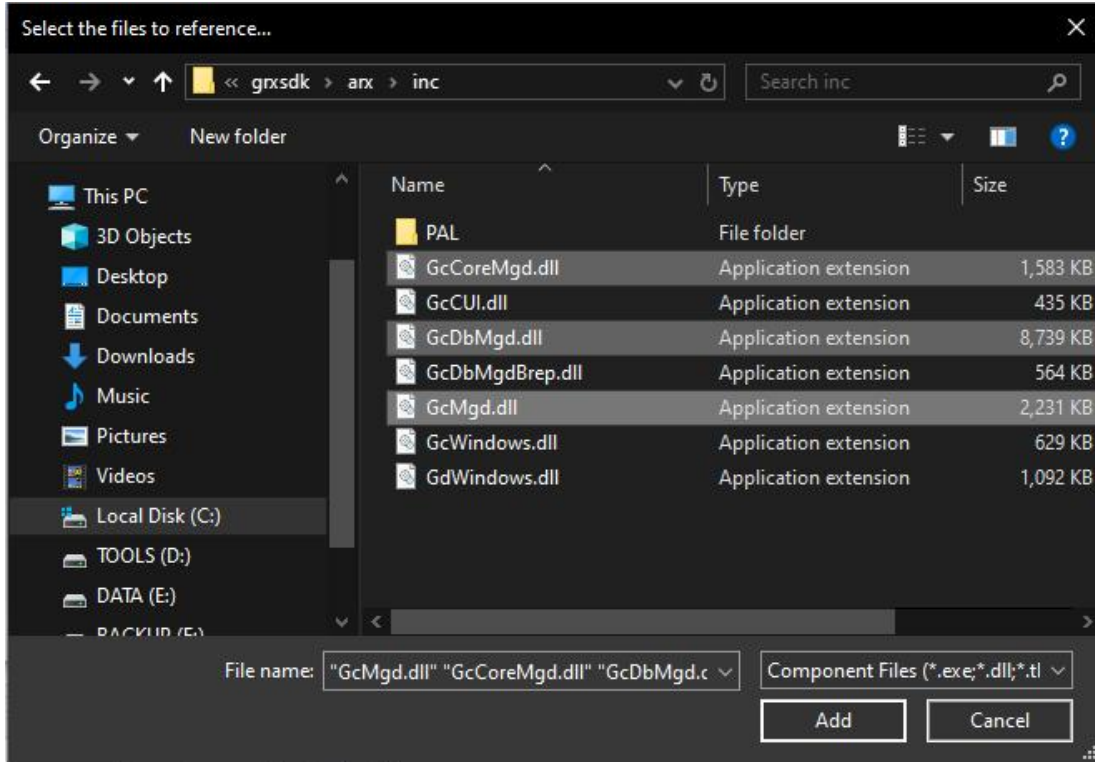
5.1.2. Add Reference

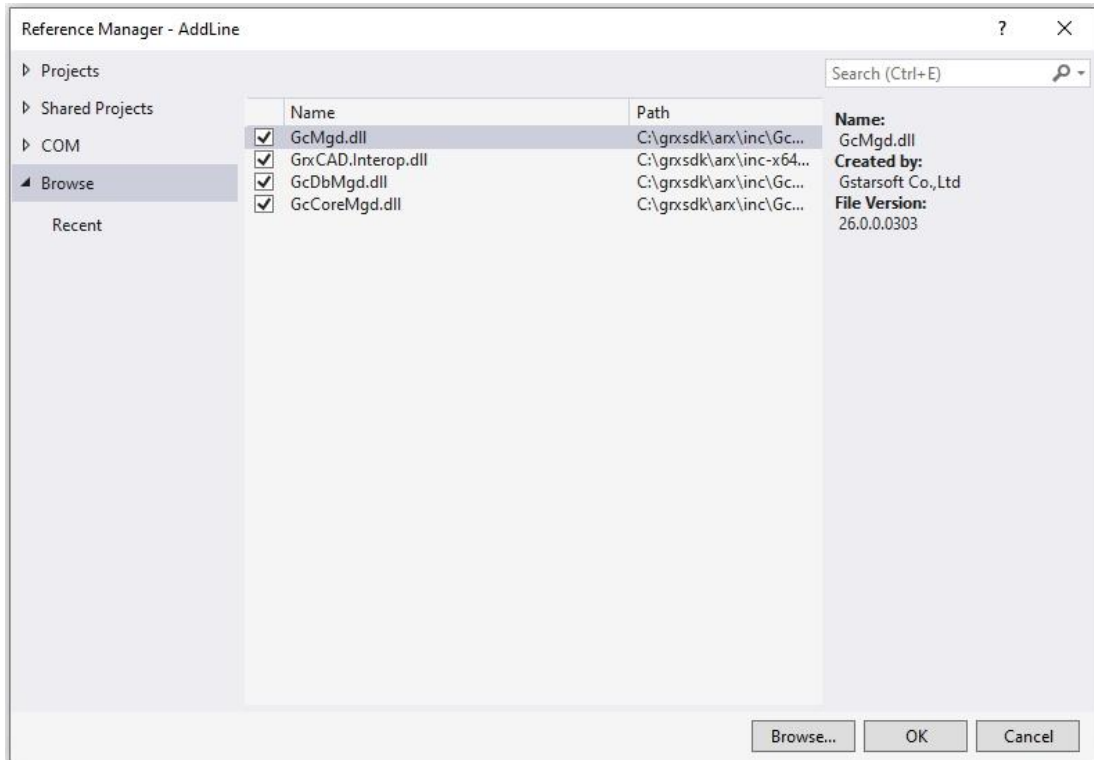
In **Solution Explorer** of Visual Studio 2022, right-click on project 'Hello' and select **Add Reference**. After the **Reference Manager** dialog window pops out, click **Browse...** button as shown below.



5.1.3. Load Reference File

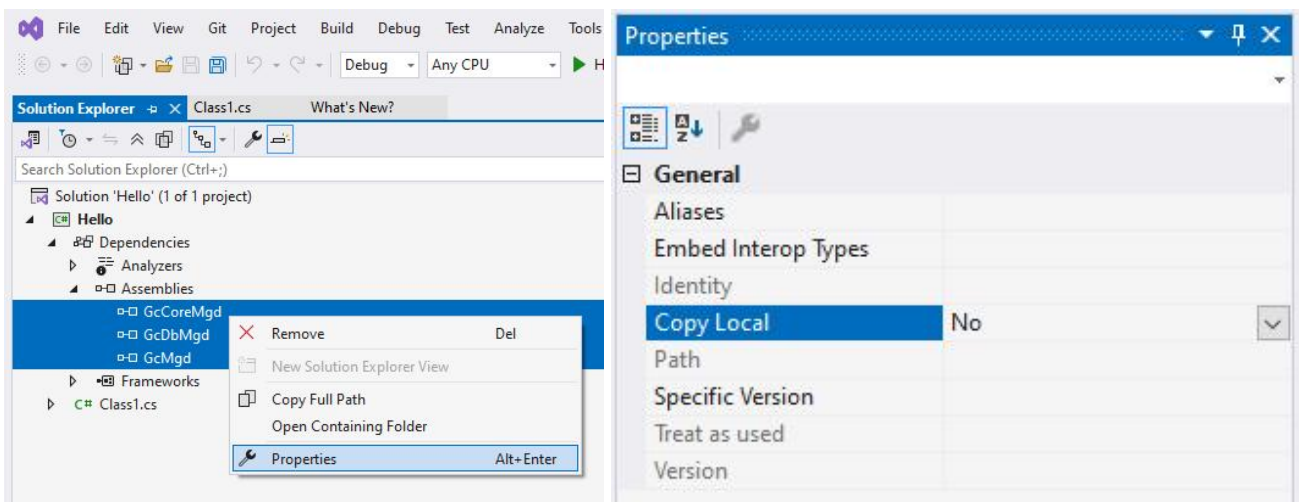
From the installation path of SDK, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll' or 'acax.tlb' in '\arx\inc-x64', and add them by clicking **Add** button.





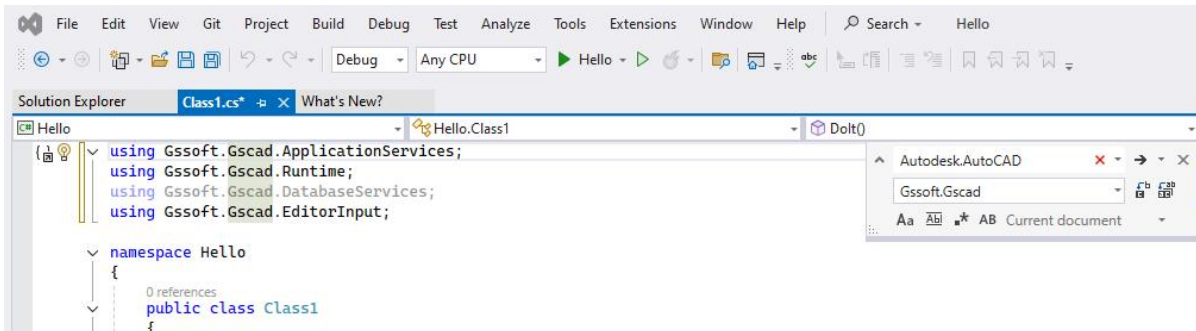
NOTE: 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' must be added while 'GrxCAD.Interop.dll' (or 'acax.tlb') is optional (needed when COM is used in the program).

After adding the files, right-click added 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' files 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. Modify Name in Namespace

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



5.3. Compile Program

In Visual Studio 2022, click **Build**→**Build Solution** to regenerate the DLL file in corresponding directory. Run CAD and input *'netload'* at command line, select the relative DLL file from **Select .Net Assembly** dialog window and click **Open** button to load it.

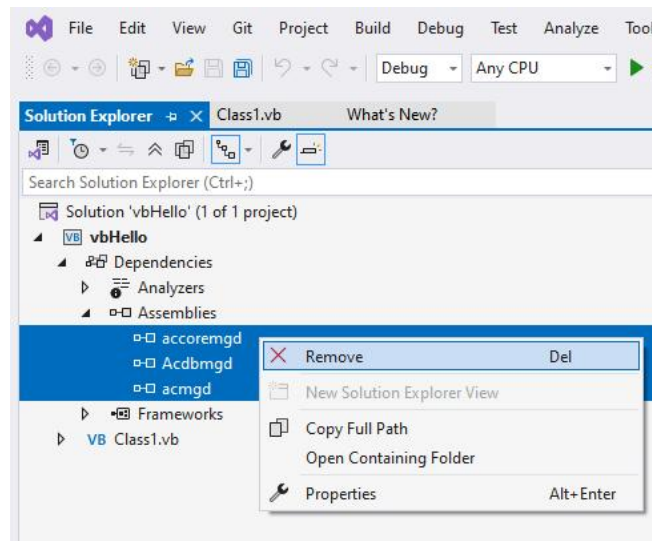
6. Visual Studio 2022 VB .NET Program Migration Process

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

6.1. Change Reference File

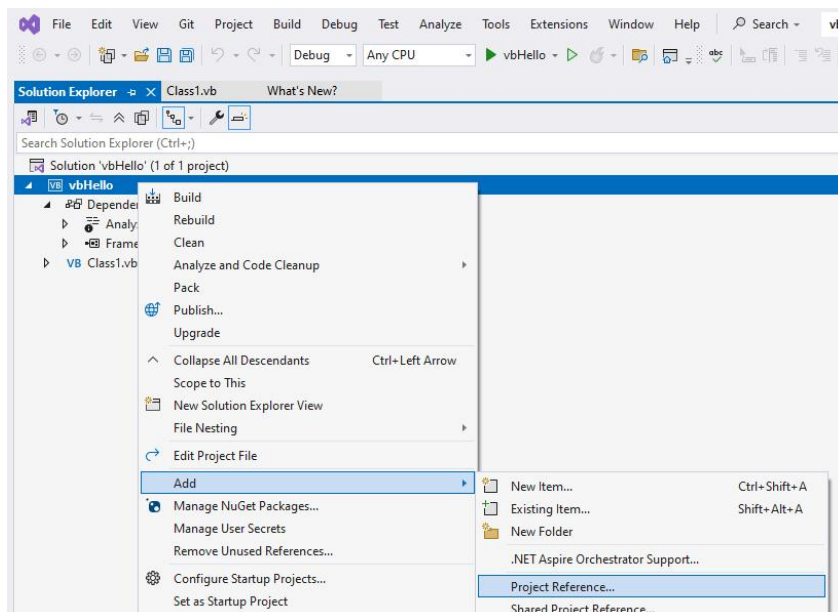
6.1.1. Remove Reference

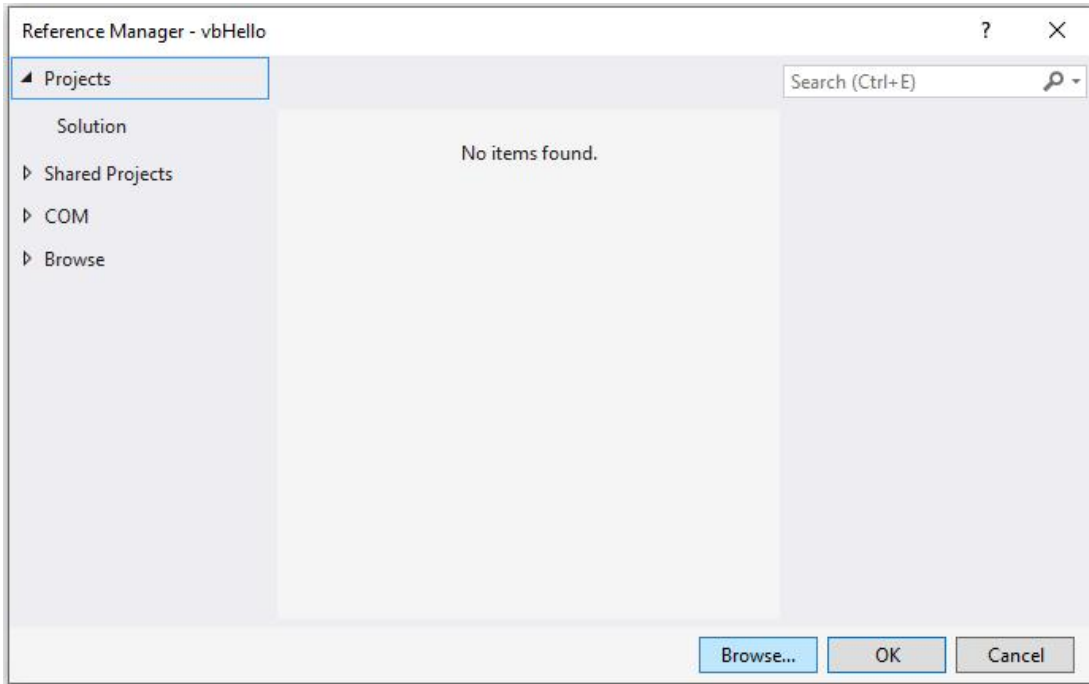
From **Solution Explorer** of Visual Studio 2022, find project 'vbHello' and remove relevant ARX reference files, as shown below.



6.1.2. Add Reference

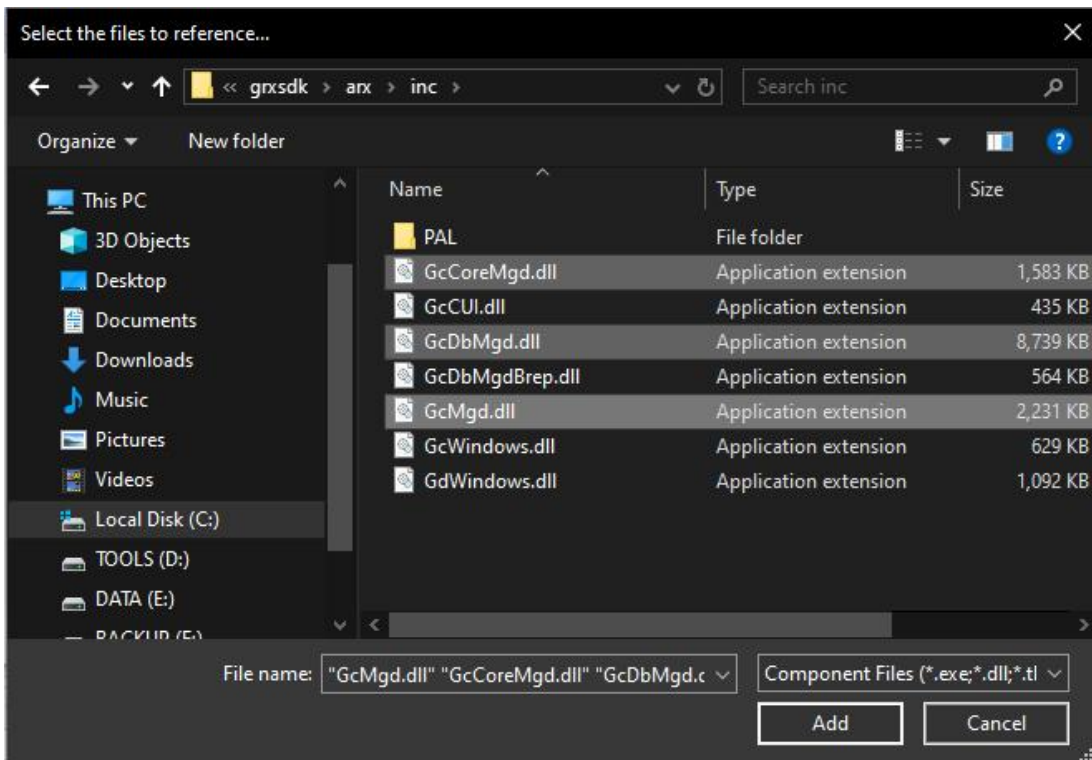
In **Solution Explorer** of Visual Studio 2022, right-click on project 'vbHello' and select **Add Reference**. After the **Reference Manager** dialog window pops out, click **Browse...** button as shown below.

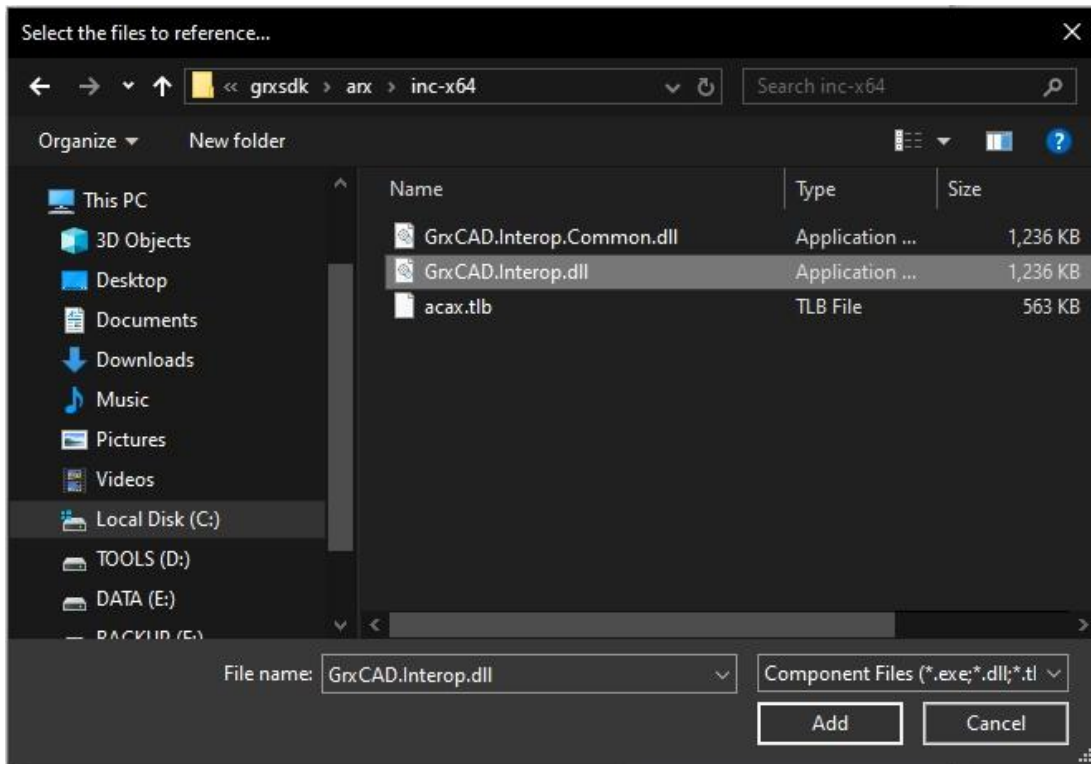




6.1.3. Load Reference File

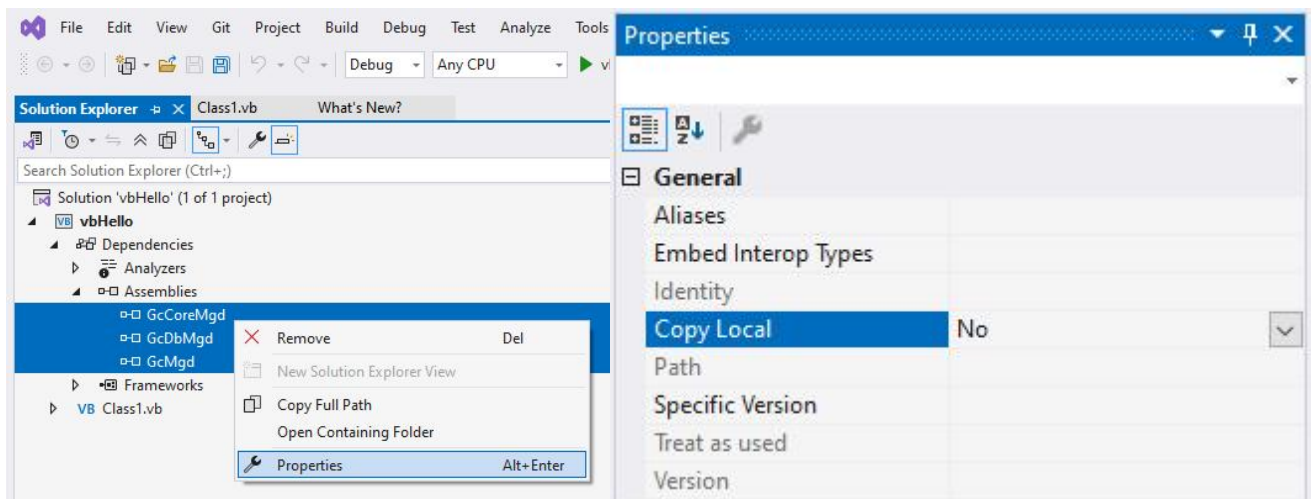
From the installation path of SDK, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll' or 'acax.tlb' in '\arx\inc-x64', and add them by clicking **Add** button.





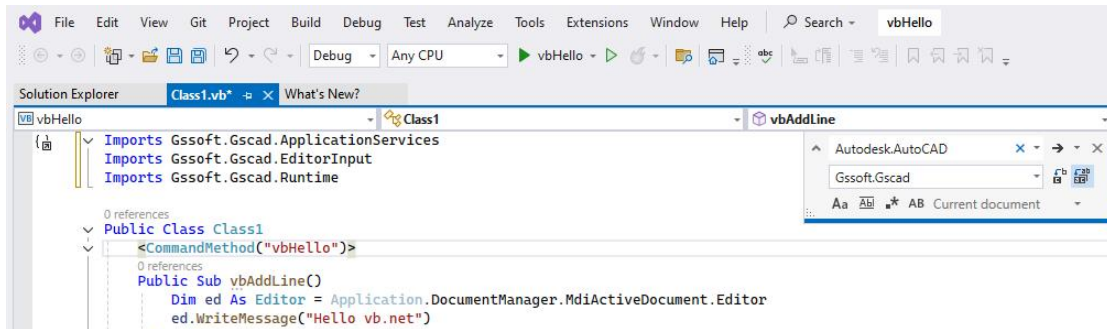
NOTE: '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' must be added while '*GrxCAD.Interop.dll*' (or '*acax.tlb*') is optional (needed when COM is used in the program).

After adding the files, right-click added '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' files 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. Modify Name in Namespace

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



6.3. Compile Program

In Visual Studio 2022, click **Build**→**Build Solution** to regenerate the DLL file in corresponding directory. Run CAD 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. Other Usage

7.1. Obtain 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 Gssoft.Gscad.Runtime
Imports GCAD

Public Class Class1
    <CommandMethod("GetGcadApplication")>
    Public Sub GetGcadApplication()
        'Define obtain GcadApplication object, the object is null
        Dim GCadApp As GcadApplication

        'Obtain the object address through the process name.
        GCadApp = GetObject("gcad.application")

        'Minimize the obtained process window
        GCadApp.WindowState = GcWindowState.acMin
    End Sub
End Class
```

7.2. 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 GCAD' and delete 'Autodesk.AutoCAD.Interop.Common'
- COM With C# .NET:
Add 'using GCAD' 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.

7.3. Sample of Replacement

7.3.1. Sample of Namespace Replacement

AutoCAD®	GsCAD
<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>

7.3.2. Sample of Class Name and Property Name Replacement

AutoCAD®	GsCAD
<i>Autodesk.AutoCAD.Interop.Common.ACAD_COLOR</i>	<i>GCAD.GCAD_COLOR</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadEntity</i>	<i>GCAD.GcadEntity</i>
<i>Autodesk.AutoCAD.Interop.Common.AcSelect</i>	<i>GCAD.GcSelect</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadRasterImage</i>	<i>GCAD.GcadRasterImage</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadText</i>	<i>GCAD.GcadText</i>
<i>Autodesk.AutoCAD.Interop.Common.AcadMText</i>	<i>GCAD.GcadMText</i>
<i>Autodesk.AutoCAD.Interop.Common.AcWindowState</i>	<i>GCAD.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 2027

<https://www.gstarcad.net>

Gstarsoft