



.NET Programming Guide

GstarCAD 2024



Table of Contents

1.	Introduction	4
2.	Programming Environment	5
3.	Install GstarCAD SDK	6
4.	Visual Studio 2017 C# .NET Programming Sample	7
4.1.	Create .NET Project	7
4.1.1.	Run Microsoft® Visual Studio 2017	7
4.1.2.	Input Project Save Path and Project Name	7
4.1.3.	Finish Creating New Project	7
4.2.	Set Reference Library	8
4.2.1.	Add Reference	8
4.2.2.	Choose and Add Reference File	9
4.3.	Add Code	12
4.4.	Compile Program	13
4.5.	Run Program	14
5.	Visual Studio 2017 VB .NET Programming Sample	15
5.1.	Create .NET Project	15
5.1.1.	Run Microsoft® Visual Studio 2017	15
5.1.2.	Input Project Save Path and Project Name	15
5.1.3.	Finish Creating New Project	15
5.2.	Set Reference Library	16
5.2.1.	Add Reference	16
5.2.2.	Choose and Add Reference File	17
5.3.	Add Code	20
5.4.	Compile Program	21
5.5.	Run Program	21
6.	Special Usage of GstarCAD .NET Programming	23
6.1.	How to Use ' <i>using</i> '	23
6.2.	How to Use ' <i>ResultBuffer</i> '	23
6.3.	Alternative Method of Importing Unmanaged ARX Function	23

6.4.	Method of Selecting in Current Drawing after Clicking Button in Modal Dialog Box.....	23
6.5.	C# .NET and VB .NET Namespace Modification.....	24
7.	Copyright	25

1. Introduction

GstarCAD .NET API provides a series of Managed Wrapper Class, Object-oriented development environment and API on the basic of .NET. Developers can use the languages which support .NET language, such as VB .NET, C# to develop application on GstarCAD.

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. 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.

4. Visual Studio 2017 C# .NET Programming Sample

The following is an example of how to create a project called 'Hello', assuming that the GstarCAD SDK is installed in the 'C:\grxsdk'.

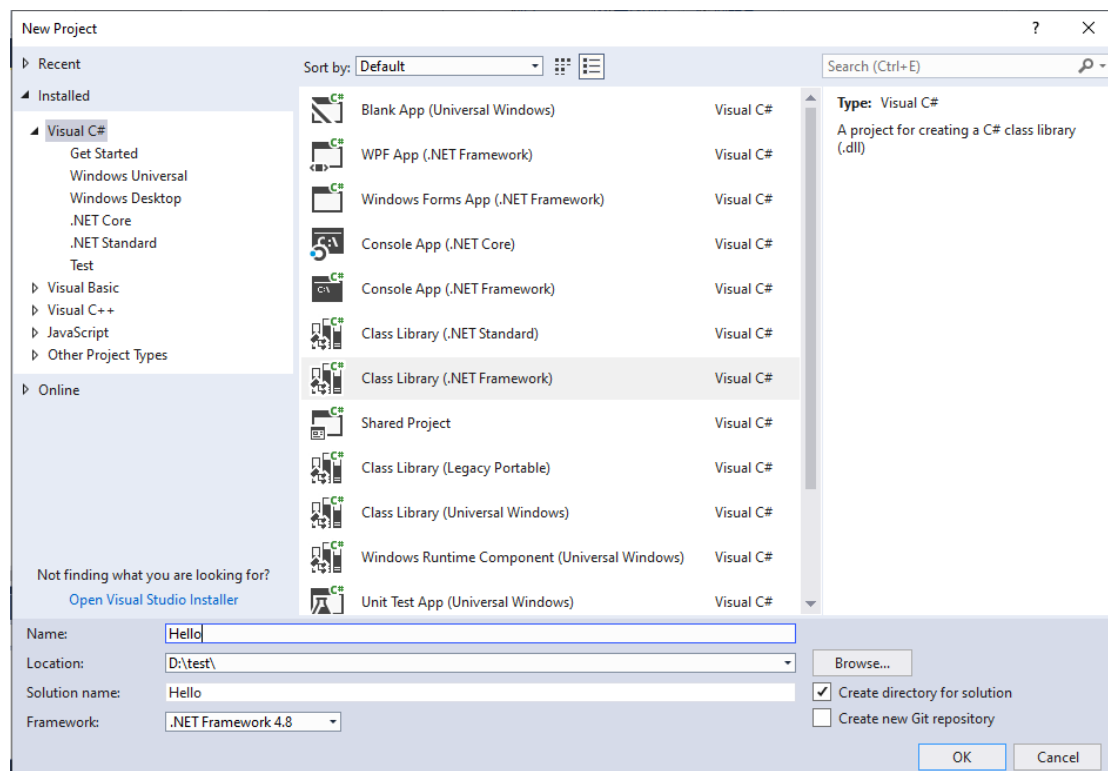
4.1. Create .NET Project

4.1.1. Run Microsoft® Visual Studio 2017

Click **File**→**New**→**Project** to launch the **New Project** dialog window. Select **Visual C#** in the **Installed** on the left side and click **Class Library (.NET Framework)** in the middle window.

4.1.2. Input Project Save Path and Project Name

Input 'Hello' at the **Name** field in the **New Project** dialog window and choose **.NET Framework 4.8** at the **Framework** field, as shown below:



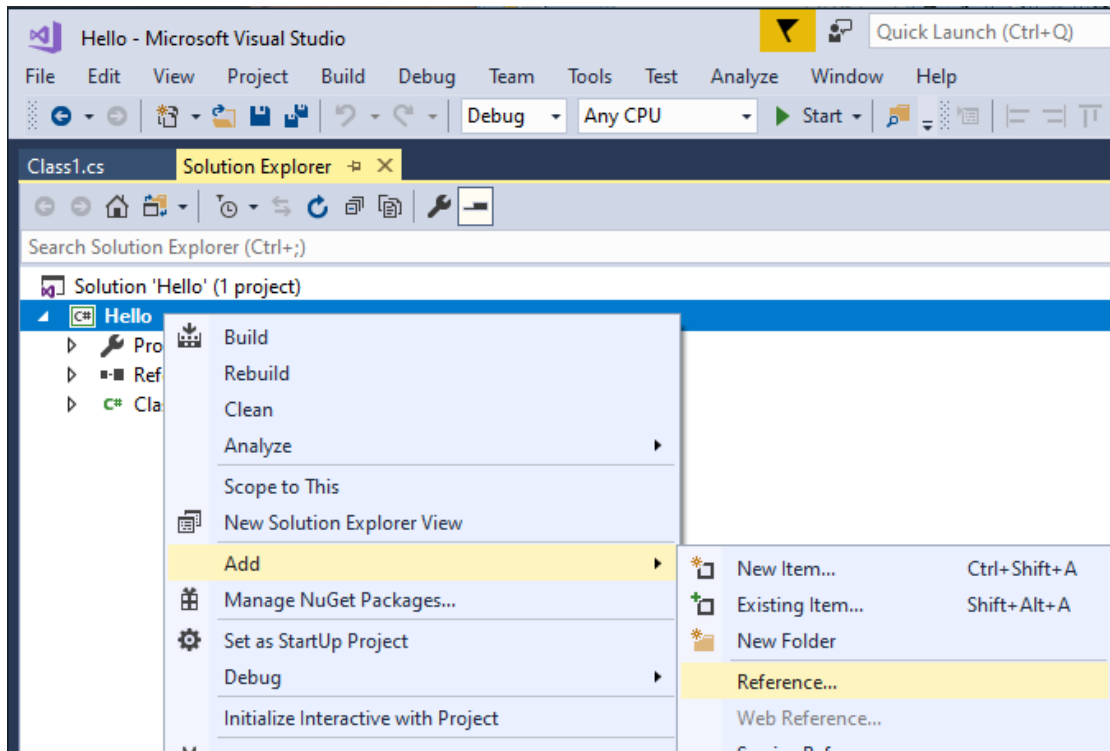
4.1.3. Finish Creating New Project

Click **OK** button to finish creating the new project 'Hello'.

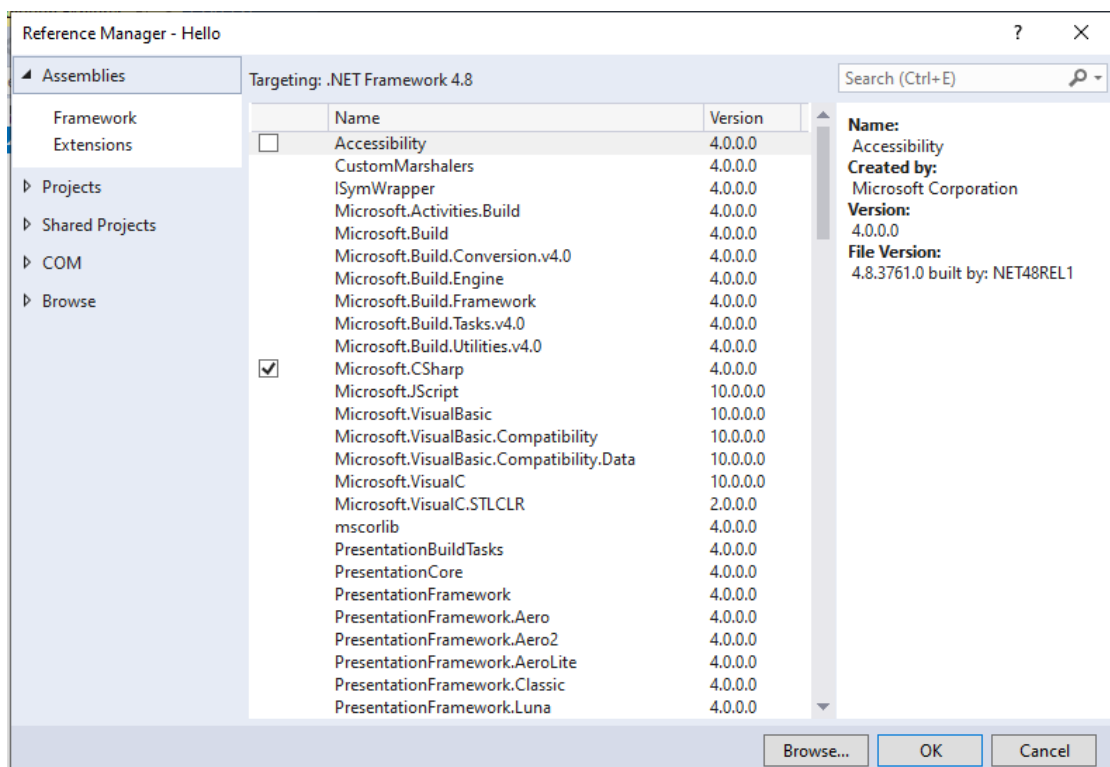
4.2. Set Reference Library

4.2.1. Add Reference

In Visual Studio 2017, find the 'Hello' project in the **Solution Explorer**, right-click on it and select **Add**→**Reference**.

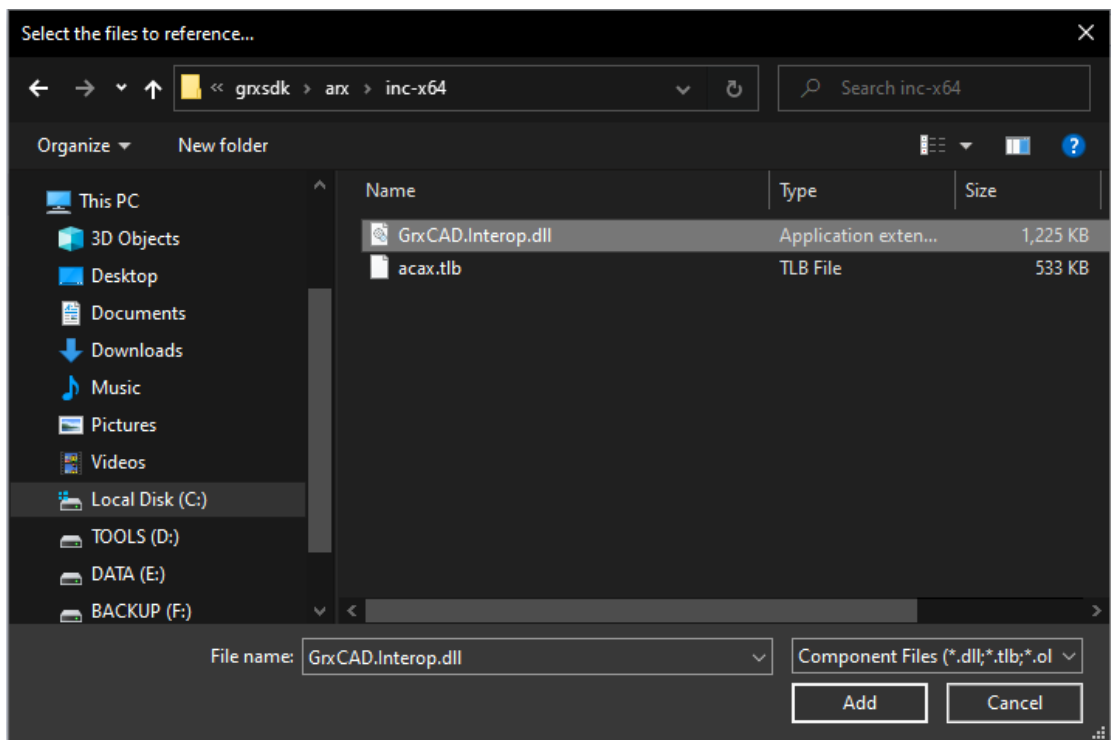
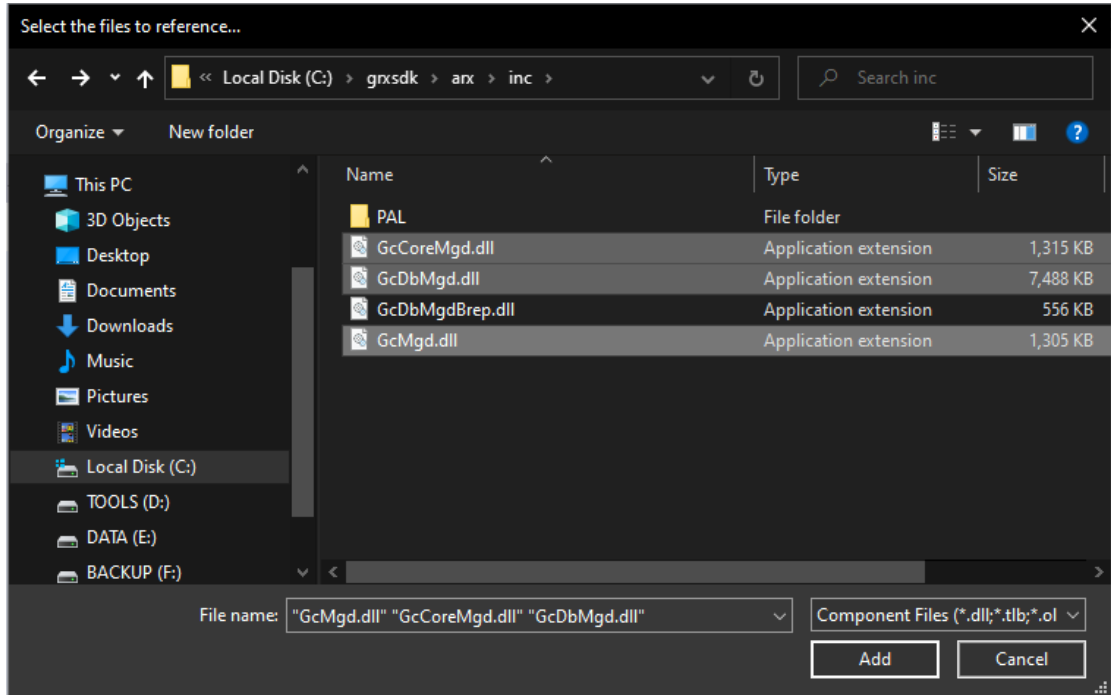


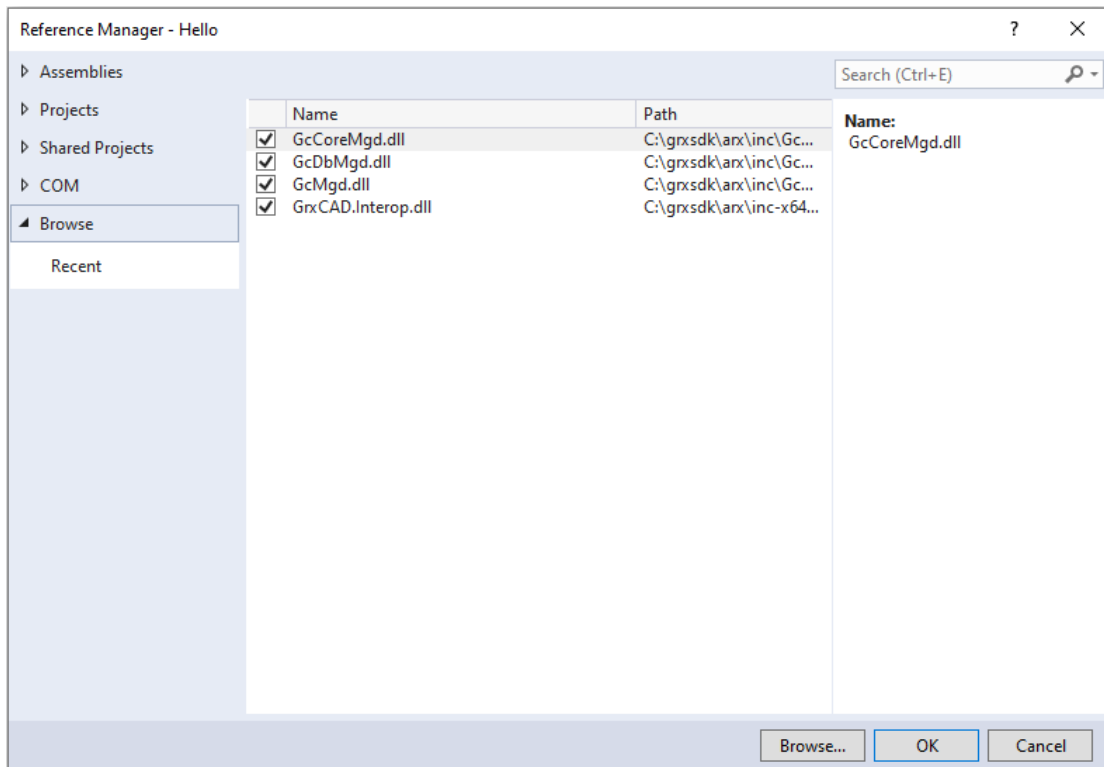
After the **Reference Manager** dialog window pops out, click **Browse** as shown below.



4.2.2. Choose and Add Reference File

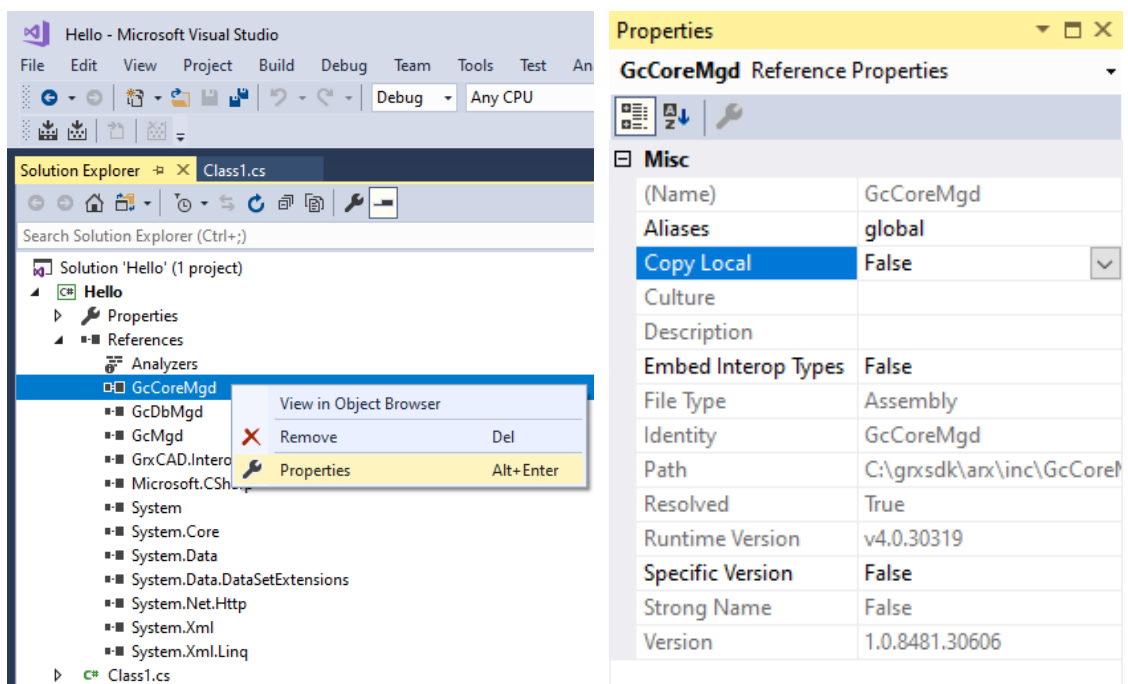
From the GstarCAD SDK installation directory, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll'(or 'acax.tlb') in '\arx\inc-x64', and add them by clicking **Add** button.





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*'.



- '*GcCoreMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad.ApplicationServices
 - Gssoft.Gscad.ApplicationServices.Core
 - Gssoft.Gscad.EditorInput
 - Gssoft.Gscad.GraphicsSystem
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.PlottingServices
 - Gssoft.Gscad.Publishing
 - Gssoft.Gscad.Runtime
 - Gssoft.Gscad.Windows

- '*GcDbMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad
 - Gssoft.Gscad.Colors
 - Gssoft.Gscad.ComponentModel
 - Gssoft.Gscad.DatabaseServices
 - Gssoft.Gscad.DatabaseServices.Filters
 - Gssoft.Gscad.DatabaseServices.Internal
 - Gssoft.Gscad.Geometry
 - Gssoft.Gscad.GraphicsInterface
 - Gssoft.Gscad.GraphicsSystem
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.LayerManager
 - Gssoft.Gscad.Runtime

- '*GcMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad.ApplicationServices
 - Gssoft.Gscad.EditorInput
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.Internal.Calculator
 - Gssoft.Gscad.Internal.DatabaseServices
 - Gssoft.Gscad.Internal.Forms
 - Gssoft.Gscad.Internal.PreviousInput
 - Gssoft.Gscad.Internal.PropertyInspector
 - Gssoft.Gscad.Internal.Reactors
 - Gssoft.Gscad.Internal.Render.RapidRT
 - Gssoft.Gscad.Internal.Windows
 - Gssoft.Gscad.Windows
 - Gssoft.Gscad.Windows.Data
 - Gssoft.Gscad.Windows.Data.Render.RapidRT
 - Gssoft.Gscad.Windows.ToolPalette

- 'GrxCAD.Interop.dll' shows the corresponding .NET APIs of COM.

4.3. Add Code

Add the following codes to 'Class1.cs':

```
using System;

using Gssoft.Gscad.Runtime;
using Gssoft.Gscad.ApplicationServices;

[assembly: CommandClass(typeof(hello.HelloCmd))]

namespace hello
{
    public class HelloCmd
    {
        [CommandMethod("Hello")]
        static public void DoIt()
        {
            try
            {
                Application.DocumentManager.MdiActiveDocument.Editor.WriteMessage("Hello dotnet");
            }
            catch (System.Exception ex)
            {
                String str = ex.ToString();
            }
        }
    }
}
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Hello
{
    0 references
    public class Class1
    {
    }
}

using System;

using Gssoft.Gscad.Runtime;
using Gssoft.Gscad.ApplicationServices;

[assembly: CommandClass(typeof(hello.HelloCmd))]

namespace hello
{
    0 references
    public class HelloCmd

```

4.4. Compile Program

In Visual Studio 2017, click **Build**→**Build Solution** to generate 'Hello.dll' file.

```

using System;
using Gssoft.Gsca
using Gssoft.Gsca

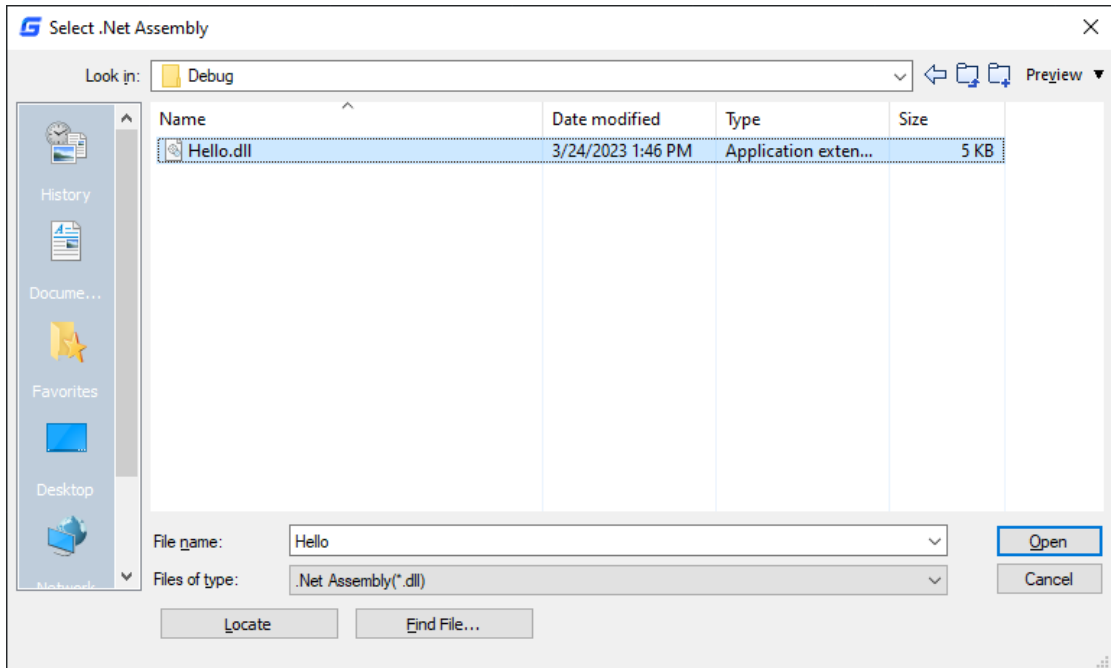
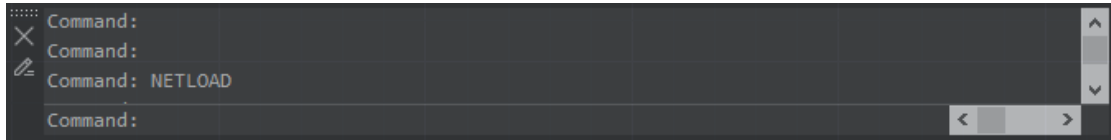
[assembly: Commar

namespace hello
{
    1 reference
    public class HelloCmd
    {
        [CommandMethod("Hello")]
        0 references
        static public void DoIt()
        {
            try
            {
                Application.DocumentManager.MdiActiveDocument.Editor.WriteMessage("Hello dotnet");
            }
            catch (System.Exception ex)
            {
                String str = ex.ToString();
            }
        }
    }
}

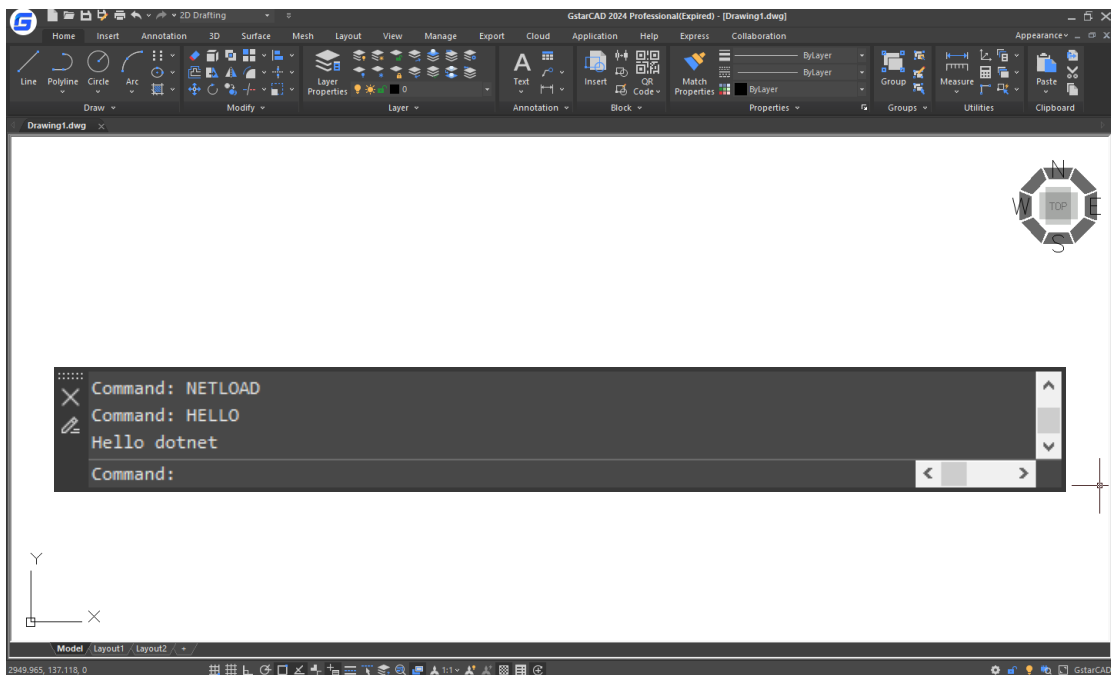
```

4.5. Run Program

Run GstarCAD and input *'netload'* at command line to launch **Select .Net Assembly** dialog window, select the *'Hello.dll'* and click **Open** button to load it.



Input *'hello'* at command line, there will be a message *'Hello dotnet'* displayed in the command line.



5. Visual Studio 2017 VB .NET Programming Sample

The following is an example of how to create a project called 'Hello', assuming that the GstarCAD SDK is installed in the 'C:\grxsdk'.

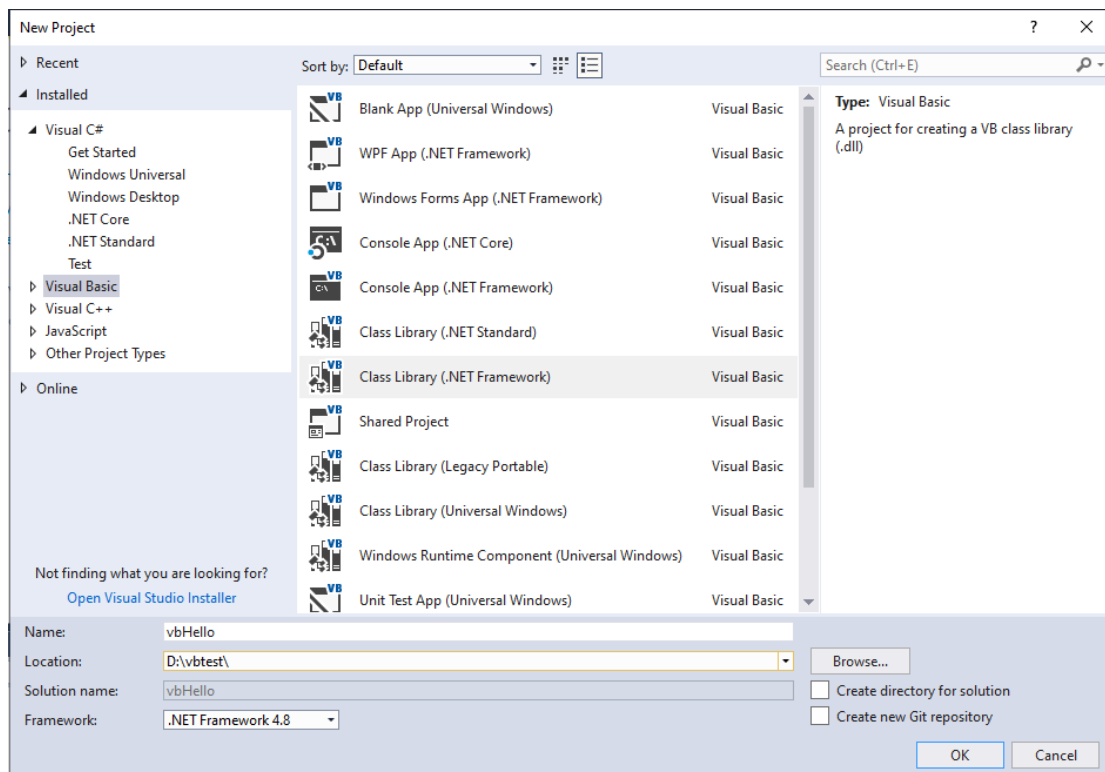
5.1. Create .NET Project

5.1.1. Run Microsoft® Visual Studio 2017

Click **File**→**New**→**Project** to launch the **New Project** dialog window. Select **Visual Basic** in the **Installed** on the left side and click **Class Library (.NET Framework)** in the middle window.

5.1.2. Input Project Save Path and Project Name

Input 'vbHello' at the **Name** field in the **New Project** dialog window and choose **.NET Framework 4.8** at the **Framework** field, as shown below:



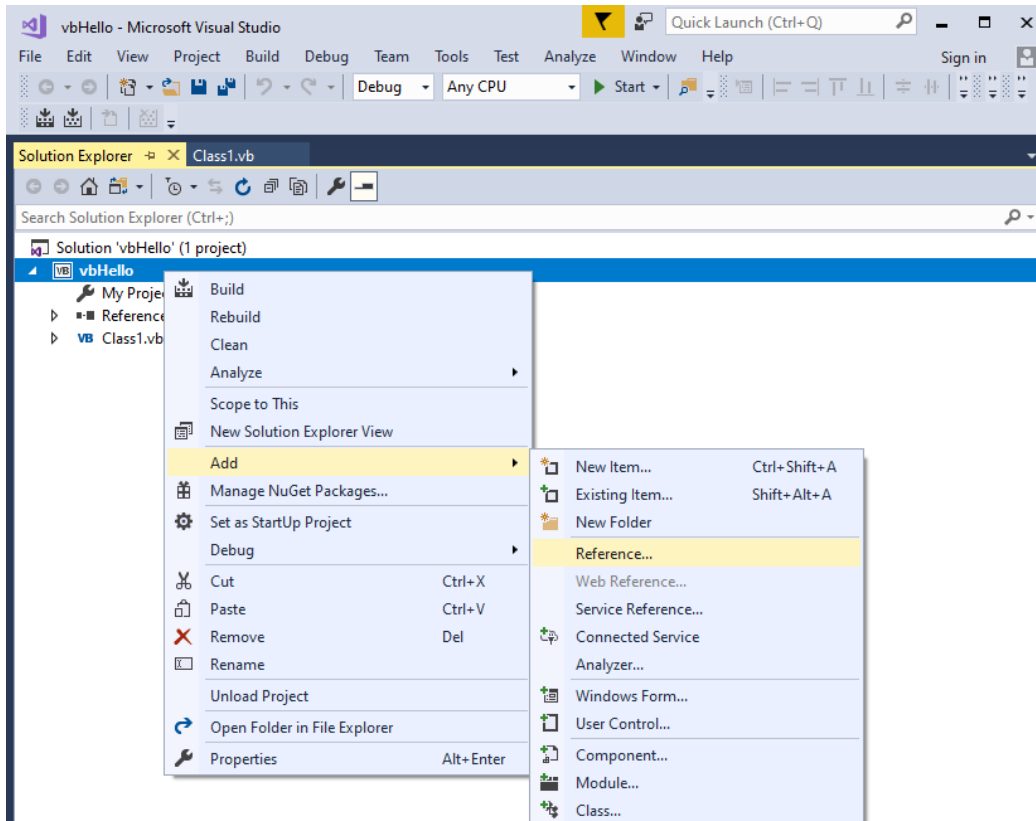
5.1.3. Finish Creating New Project

Click **OK** button to finish creating the new project 'vbHello'.

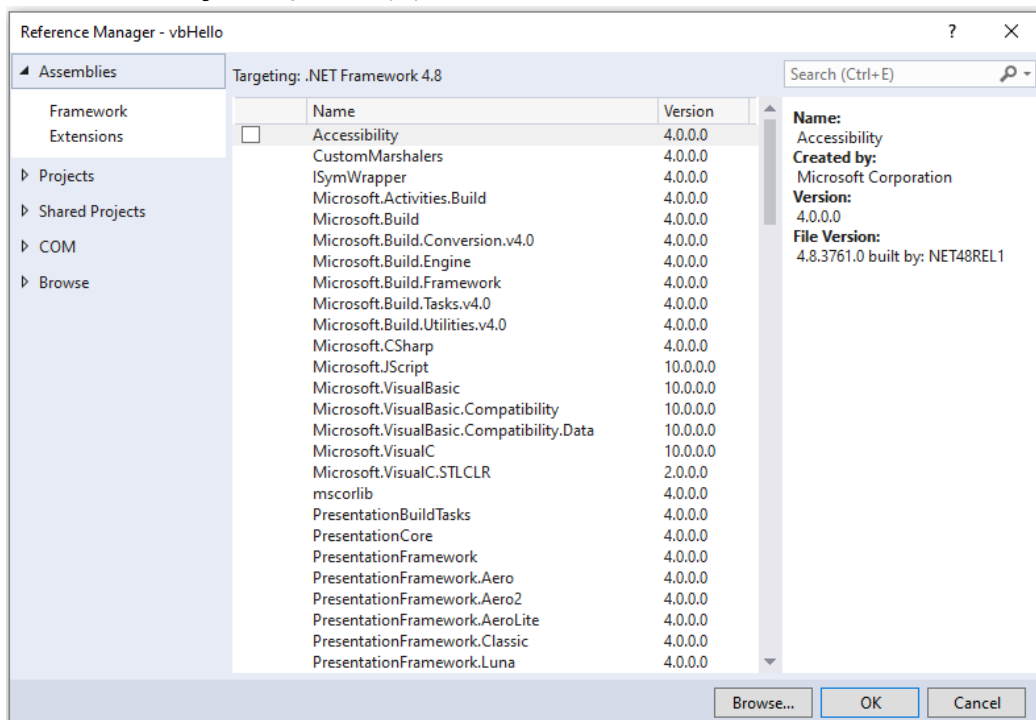
5.2. Set Reference Library

5.2.1. Add Reference

In Visual Studio 2017, find the 'Hello' project in the **Solution Explorer**, right-click on it and select **Add→Reference**.

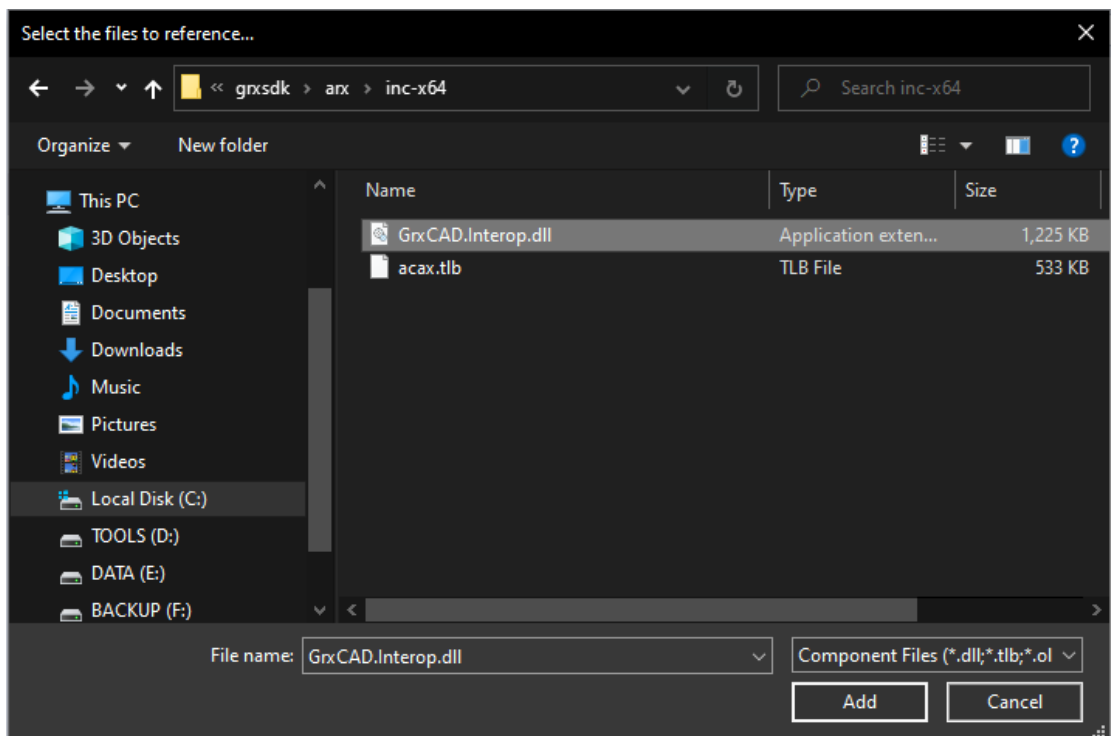
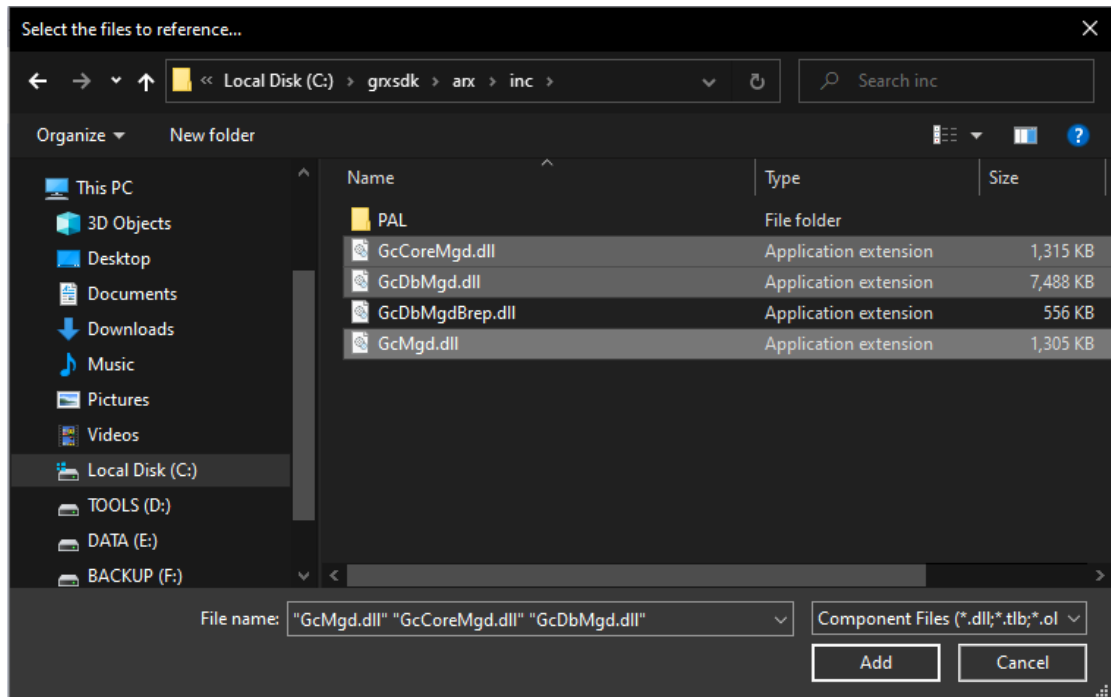


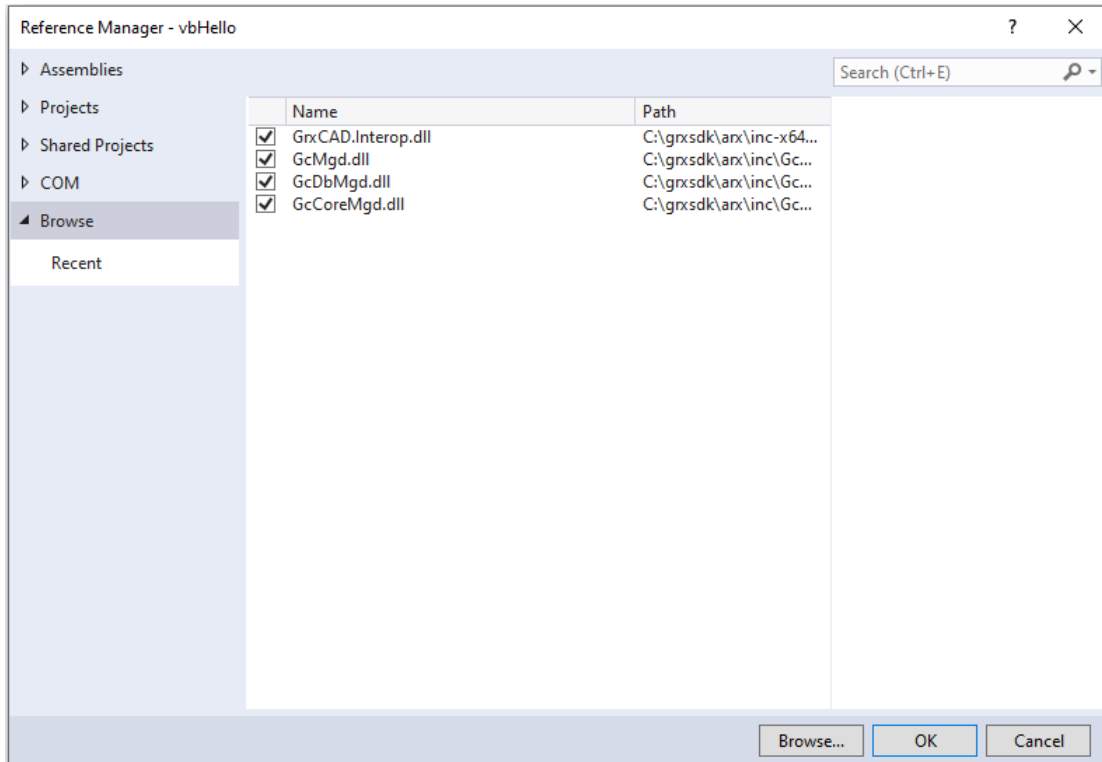
After the **Reference Manager** dialog window pops out, click **Browse** as shown below.



5.2.2. Choose and Add Reference File

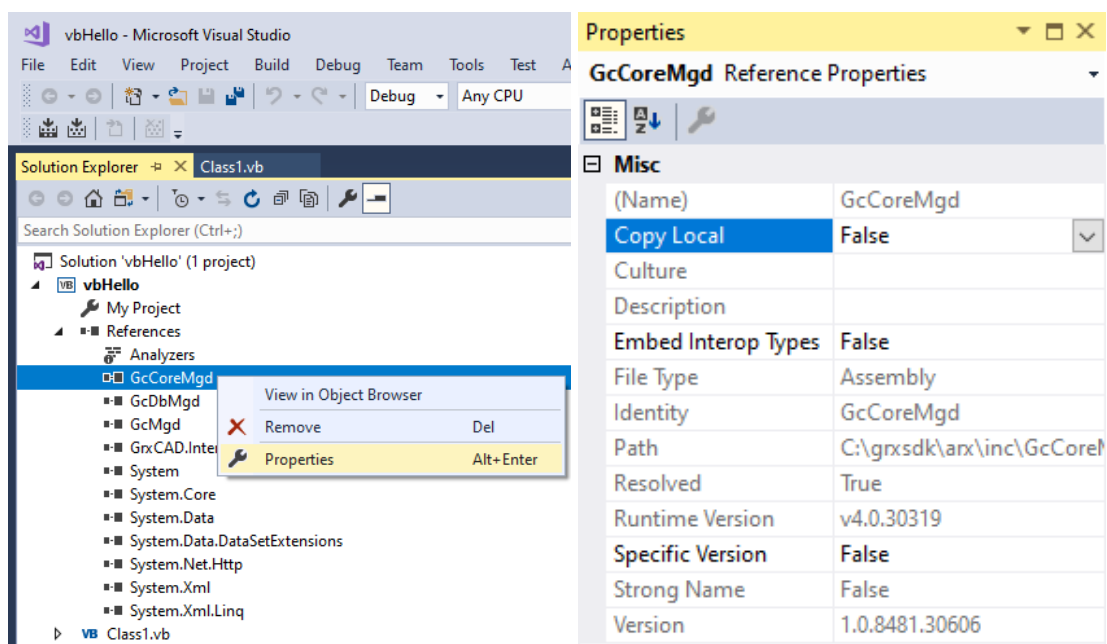
From the GstarCAD SDK installation directory, e.g. 'C:\grxsdk', select 'GcCoreMgd.dll', 'GcDbMgd.dll' and 'GcMgd.dll' in '\arx\inc' and 'GrxCAD.Interop.dll' (or 'acax.tlb') in '\arx\inc-x64', and add them by clicking **Add** button.





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*'.



- '*GcCoreMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad.ApplicationServices
 - Gssoft.Gscad.ApplicationServices.Core
 - Gssoft.Gscad.EditorInput
 - Gssoft.Gscad.GraphicsSystem
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.PlottingServices
 - Gssoft.Gscad.Publishing
 - Gssoft.Gscad.Runtime
 - Gssoft.Gscad.Windows

- '*GcDbMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad
 - Gssoft.Gscad.Colors
 - Gssoft.Gscad.ComponentModel
 - Gssoft.Gscad.DatabaseServices
 - Gssoft.Gscad.DatabaseServices.Filters
 - Gssoft.Gscad.DatabaseServices.Internal
 - Gssoft.Gscad.Geometry
 - Gssoft.Gscad.GraphicsInterface
 - Gssoft.Gscad.GraphicsSystem
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.LayerManager
 - Gssoft.Gscad.Runtime

- '*GcMgd.dll*' includes the following namespaces:
 - Gssoft.Gscad.ApplicationServices
 - Gssoft.Gscad.EditorInput
 - Gssoft.Gscad.Internal
 - Gssoft.Gscad.Internal.Calculator
 - Gssoft.Gscad.Internal.DatabaseServices
 - Gssoft.Gscad.Internal.Forms
 - Gssoft.Gscad.Internal.PreviousInput
 - Gssoft.Gscad.Internal.PropertyInspector
 - Gssoft.Gscad.Internal.Reactors
 - Gssoft.Gscad.Internal.Render.RapidRT
 - Gssoft.Gscad.Internal.Windows
 - Gssoft.Gscad.Windows
 - Gssoft.Gscad.Windows.Data
 - Gssoft.Gscad.Windows.Data.Render.RapidRT
 - Gssoft.Gscad.Windows.ToolPalette

- 'GrxCAD.Interop.dll' shows the corresponding .NET APIs of COM.

5.3. Add Code

Add the following codes in 'Class1.vb':

```
Imports System
Imports Gssoft.Gscad

Imports Gssoft.Gscad.Runtime

Imports Gssoft.Gscad.DatabaseServices
Imports Gssoft.Gscad.Geometry

Imports Gssoft.Gscad.ApplicationServices
Imports Gssoft.Gscad.EditorInput

' This line is not mandatory, but improves loading performances
<Assembly: CommandClass(GetType(GRXTTest.MyCommands))>

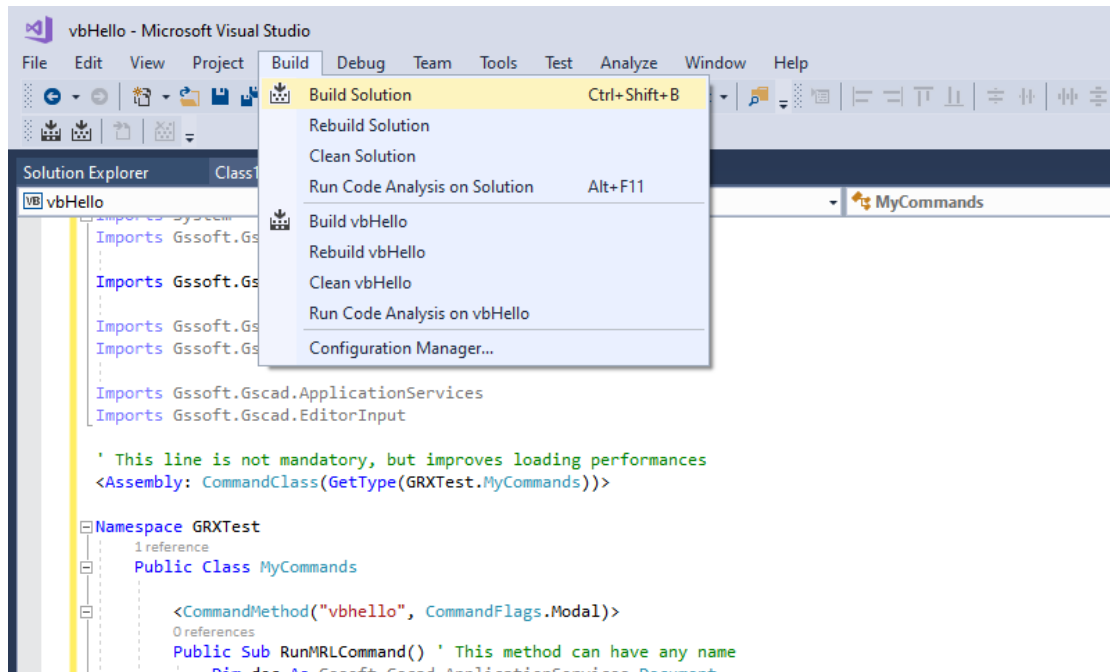
Namespace GRXTTest
Public Class MyCommands

    <CommandMethod("vbhello", CommandFlags.Modal)> _
    Public Sub RunMRLCommand() ' This method can have any name
        Dim doc As Gssoft.Gscad.ApplicationServices.Document
        Doc = Gssoft.Gscad.ApplicationServices.Application.DocumentManager.MdiActiveDocument
        doc.Editor.WriteMessage("hello vb.NET")
    End Sub

End Class
End Namespace
```

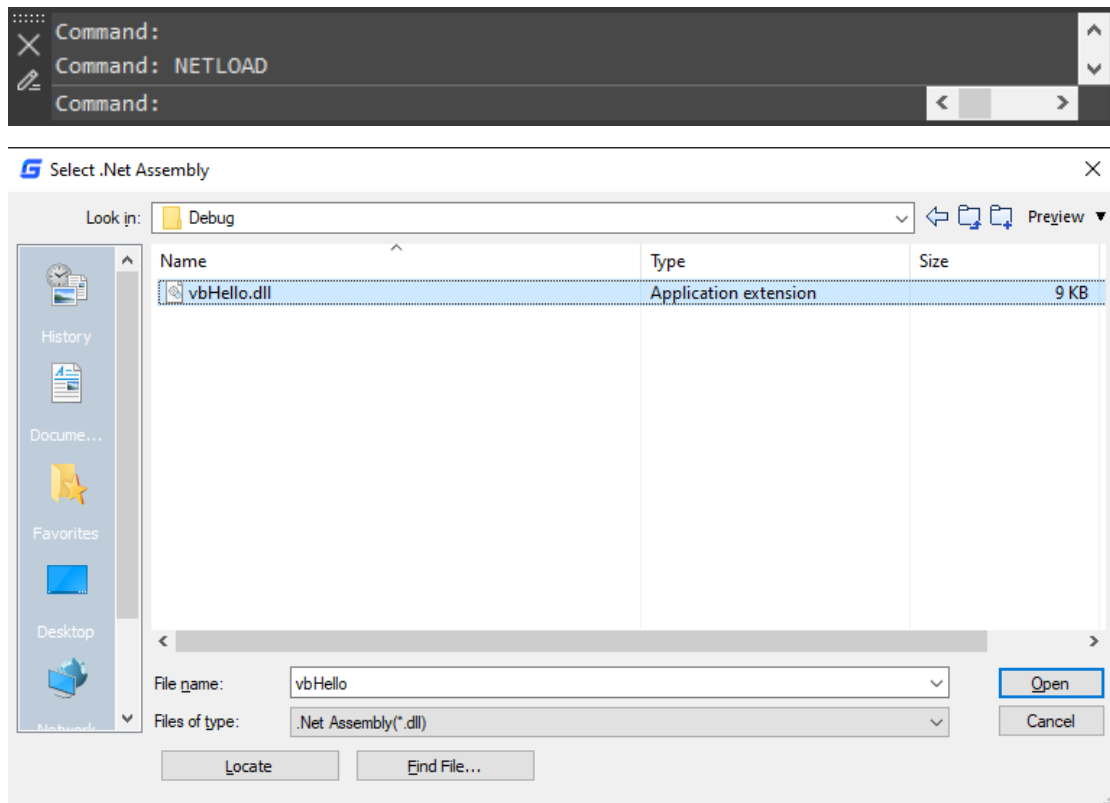
5.4. Compile Program

In Visual Studio 2017, click **Build**→**Build Solution** to generate '*vbHello.dll*' file.

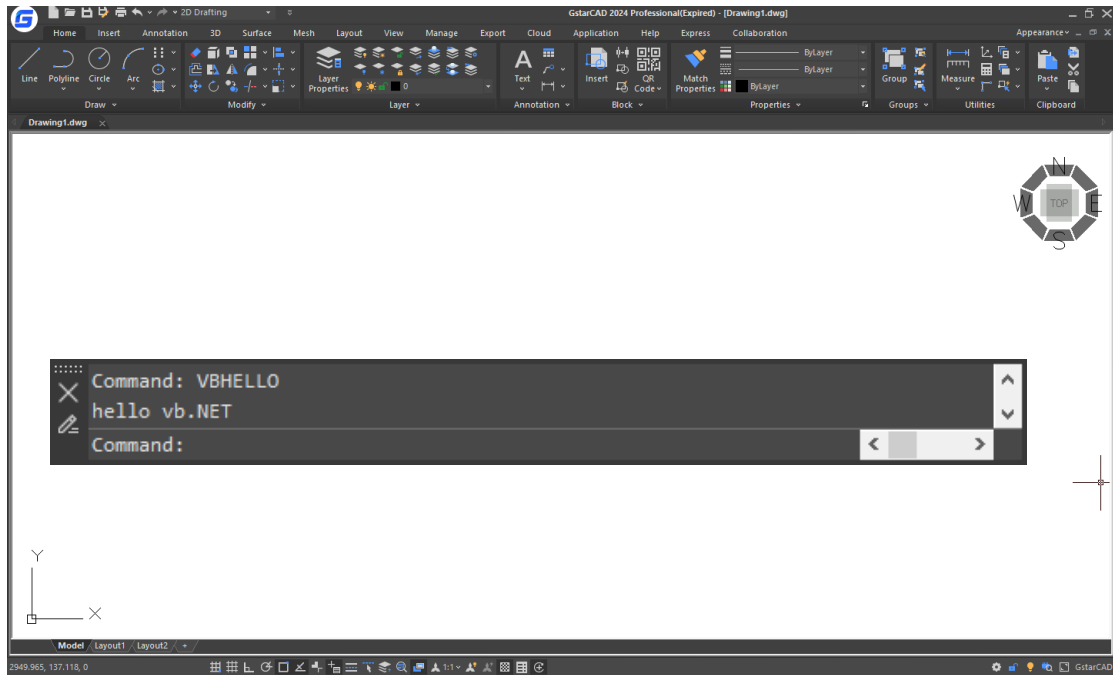


5.5. Run Program

Run GstarCAD and input '*netload*' at command line to launch **Select .Net Assembly** dialog window, select the '*Hello.dll*' and click **Open** button to load it.



Input 'vbhello' at command line, there will be a message 'hello vb.NET' displayed in the command line.



6. Special Usage of GstarCAD .NET Programming

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

6.1. How to Use 'using'

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

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

6.2. How to Use 'ResultBuffer'

Use `ResultBuffer.ResbufObject` to replace `ResultBuffer.UnmanagedObject`;

Type of `ResultBuffer.UnmanagedObject` is `OdDbResbuf`;

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

Use `ResultBuffer.Create(IntPtr,bool)` to replace `ResultBuffer(IntPtr,bool)`

6.3. Alternative Method of Importing Unmanaged ARX Function

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

6.4. 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);
    }
}

```

6.5. 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.

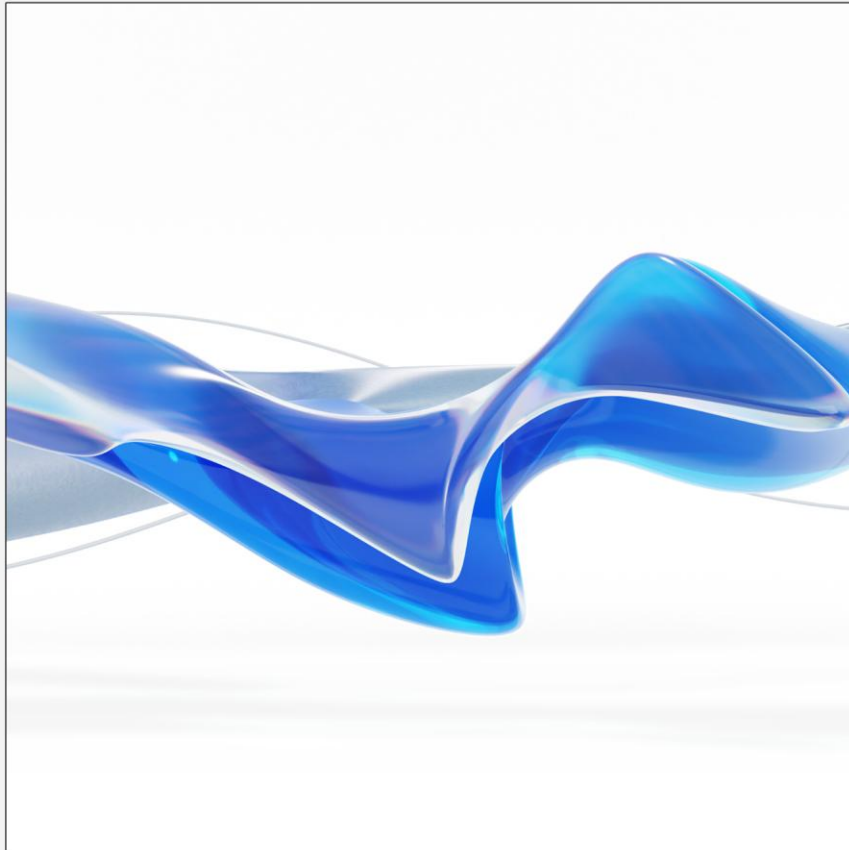
7. Copyright

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/>