.NET Development Guide

GstarCAD 2022

GstarCAD

Table of Contents

1. Brief Introduction

.NET is the latest application development package introduced by GstarCAD®, which 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# on GstarCAD application development.

2. Development Environment

- Microsoft Visual Studio Enterprise 2017（Version 15.9.17）
- CPU：

Basic Requirement：1.6 GHz CUP，Recommend：3.0 GHz CPU and above

- RAM：

Basic Requirement：2 GB，Recommend：8 GB and above

- Operation System

Windows 10 version 1507 and advanced version：Including home version, professional version, education version and enterprise version（not support LTSC and Windows 10 S）

Windows 8.1（with updated 2919355）： Core version, professional version and enterprise version

Windows 7 SP1（with latest Windows updated）： Including home version, professional version, enterprise version and Ultimate version

- Monitor Resolution:

Traditional monitor：1028 x 800 and above CRT monitor, including 4K (3840 x 2160) monitor

- GRX SDK 2022
- GstarCAD 2022
- .NET Framework 4.8 and above

3. Install .NET SDK

Download GstarCAD SDK package from our official website:

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

After you get the GRXSDK.ZIP file, please extract it to your disk directory, for instance, C:\grxsdk. Afterwards you will get 6 files, which they are inc,inc-x64,inc-x86,lib-x64,lib-x86 and samples.

- The Inc file includes header files
- The inc-x64 file includes the files used in COM and .NET for 64 bit
- The inc-x86 file includes the files used in COM and .NET for 32 bit
- The lib-x64 file includes the files referenced from GRX library
- The lib-x86 file includes the files referenced from GRX library
- Samples includes project files, they are dotnet，fact_dg，HelloADS，HelloARX，SimplePalette, etc.

The instruction of sample projects:

- Dotnet includes multiple dotnet samples，including addline，hello，vbhello, etc.
  1) Addline is the .NET sample for adding solid line.
  2) Hello is the.NET sample for prompting the information of output.
  3) Vbhello is the VB.NET sample for how to develop with .NET API.
- fact_dg is the sample for defining LSP function.
- HelloADS is the project sample to develop the type of ADS.
- HelloARX includes includes the example for how to develop with ARX API.
- SimplePalette is the sample to develop the type modules for palette.

4. VS2017 C# .NET Development Sample

The following takes the creation of the "Hello" project as an example to illustrate the creation process. And assumes that the GstarCAD SDK is installed in the “c:\grxsdk” directory.

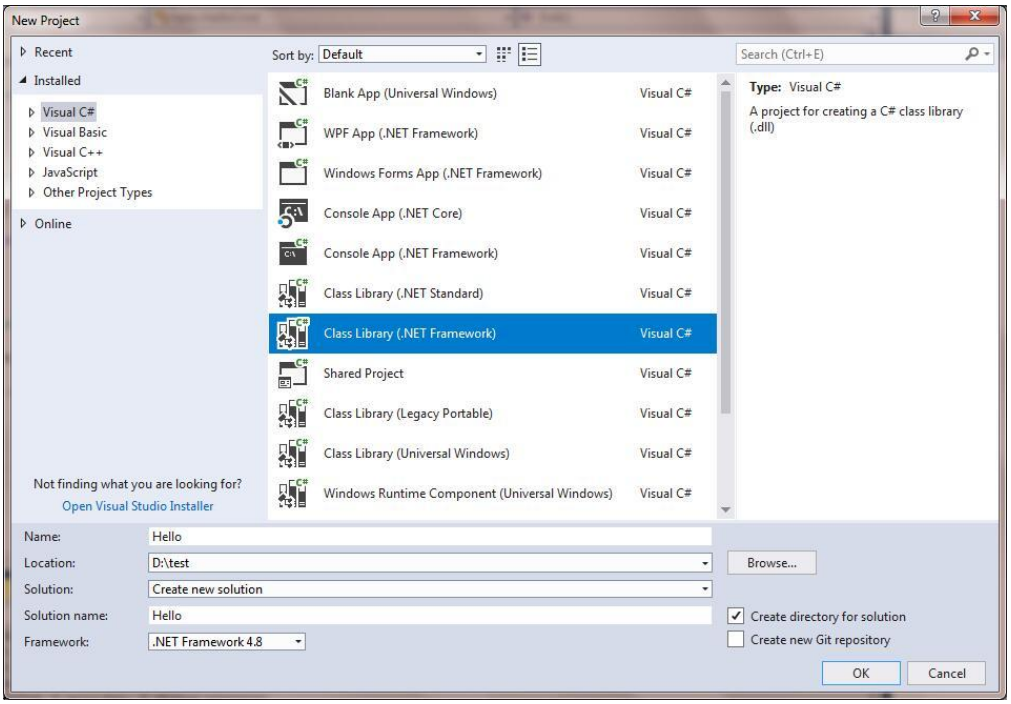
4.1. Create .Net Project

4.1.1. Run Microsoft Visual Studio 2017

Please click [File] → [New] → [Project]. After clicked the menu Project, the New Project dialog box will pops out. Then select Visual C# in the catalogue Installed Templates and click [Class Library (.Net Framework)].

4.1.2. Input Project Saved Path and Project Name

Type "Hello" at the name label in "New Project" dialog box and choose .NET Framework4.8, as shown below:



Remark： Please choose **.NET Framework 4.8** at the frame pull-down list.

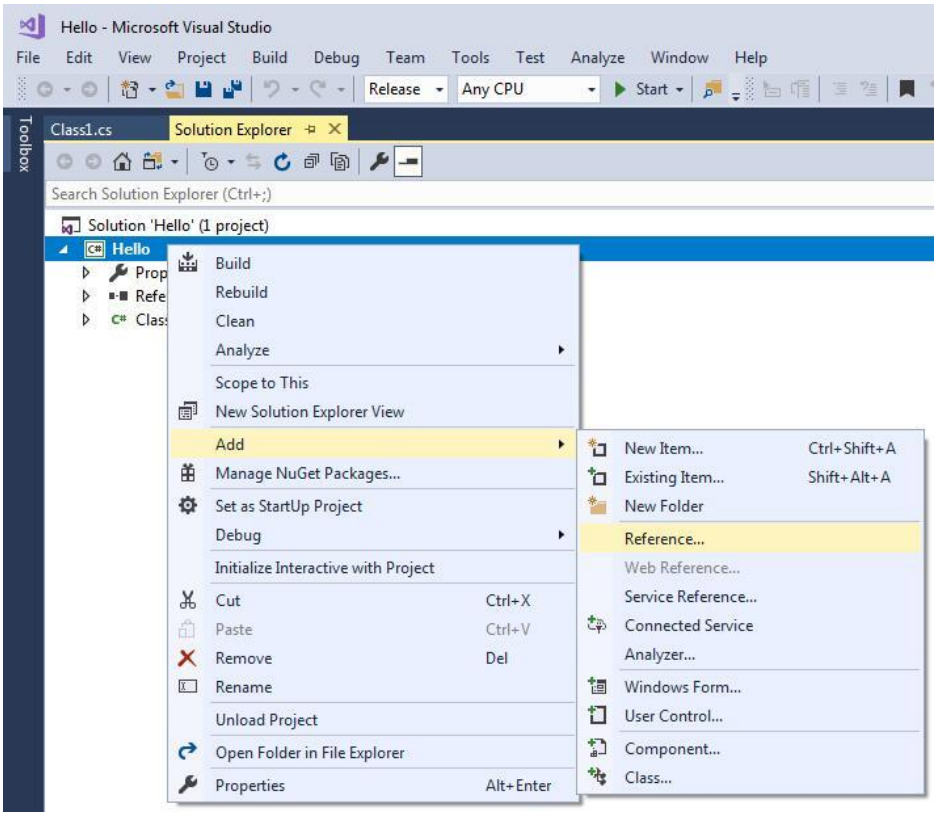
4.1.3. Finish to Create a New Project

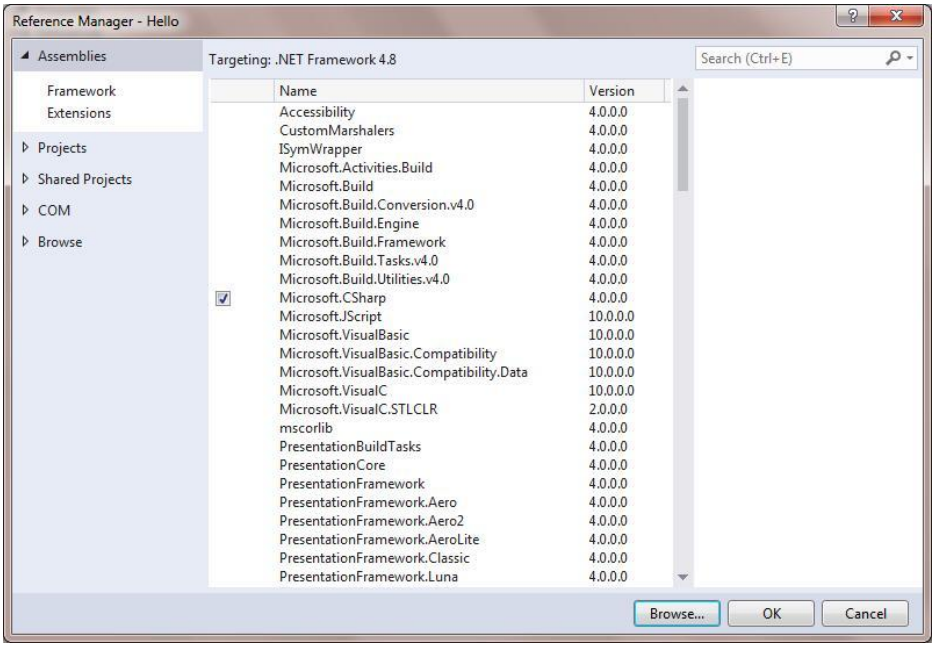
After finished operation above, click OK button in the dialog to finish the new project creating.

4.2. Set the Reference Library

4.2.1. Add Reference

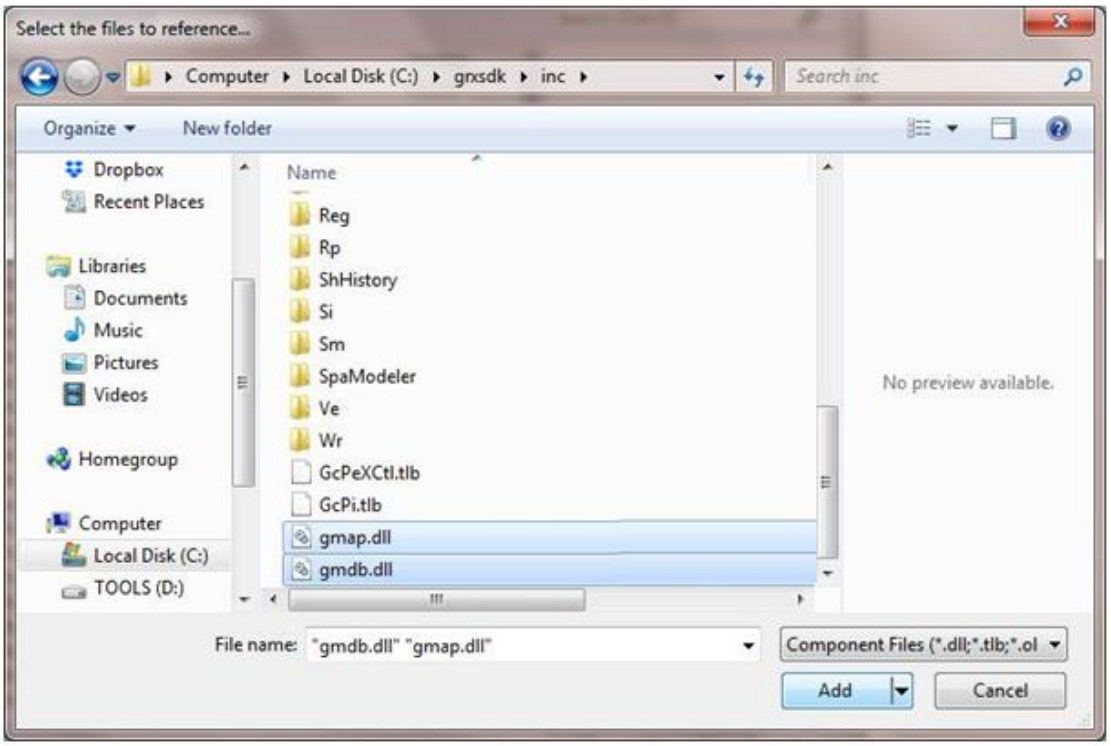
In VS 2017, find the Hello project you have been created on the solution explorer. Select the Hello project and right click to select Add Reference. After the "Reference Manager" dialog box pops out, click Browse button as shown below.

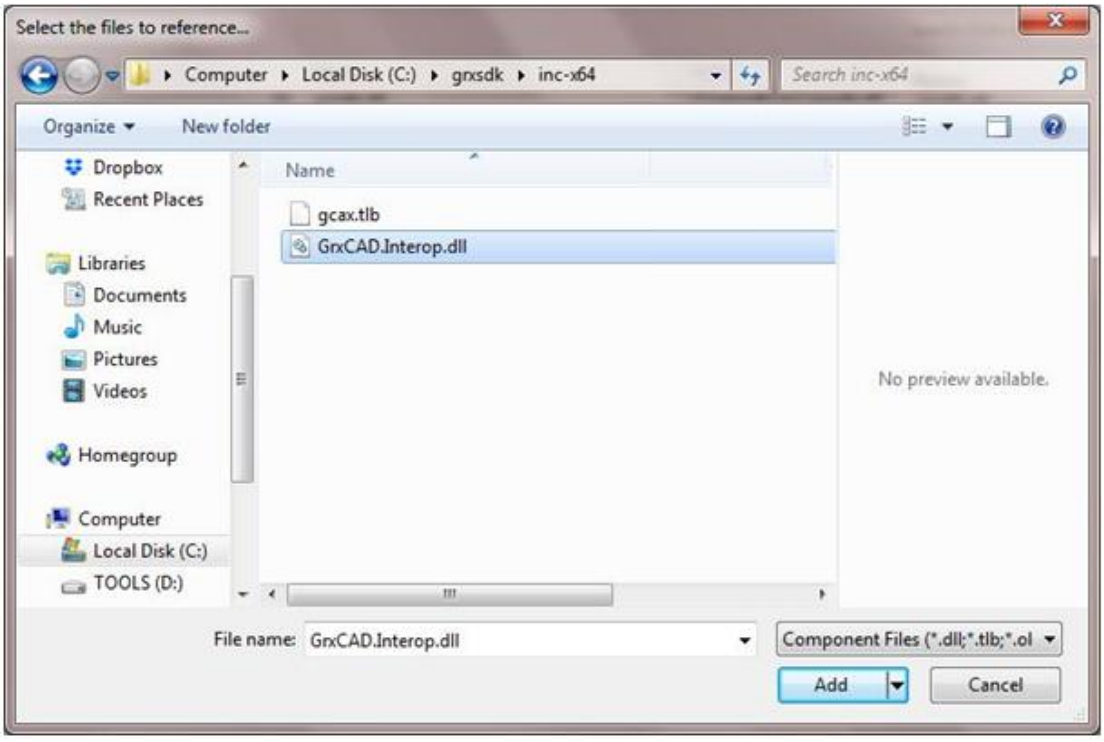




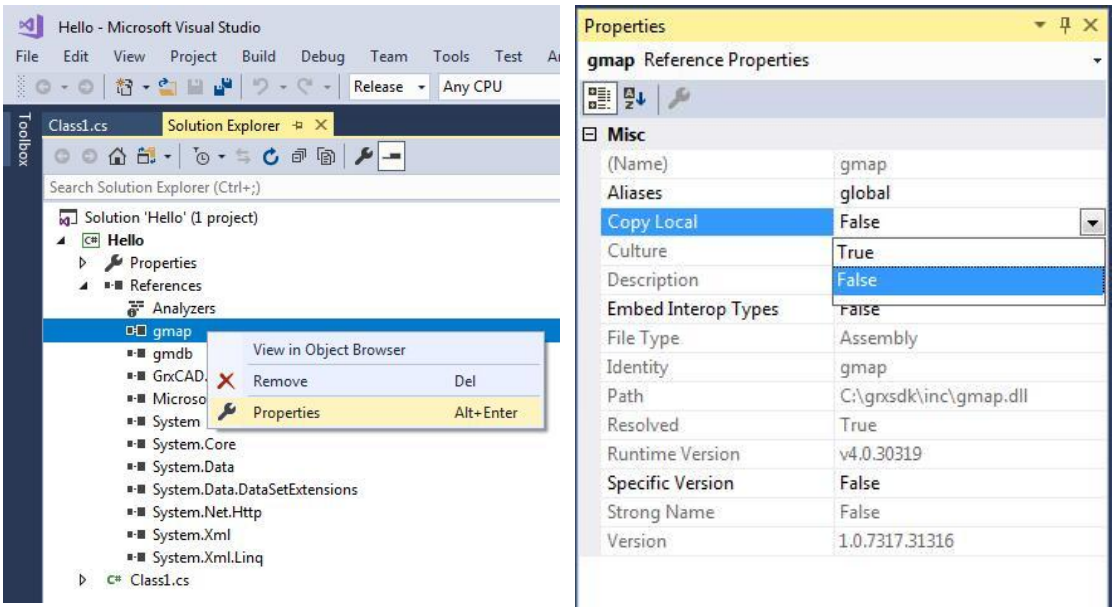
4.2.2. Choose and Add Reference Files

Find the GstarCADSDK installation path, for example, “C:\grxsdk\inc”. Select gmap.dll and gmdb.dll files in “c:\grxsdk\inc-x86” and add them by clicking Add button（If you use GstarCAD 64-bit version, please select GrxCAD.Interop.dll in inc-x64）.





Remark：gmap.dll and gmdb.dll have to be added but file GrxCAD.Interop.dll is optional. After adding the files, right click each added reference file, click Property at the context menu, then the "Reference Properties" palette pops up, change the "Copy Local" property from True to False. The following image shows how to change "Copy Local" property for gmap.dll file. Please take it as reference to change gmdb.dll file's Copy Local property.



- gmap.dll file includes the following name space:

GrxCAD.ApplicationServices
GrxCAD.EditorInput
GrxCAD.Internal
GrxCAD.PlottingServices
GrxCAD.Publishing
GrxCAD.Runtime
GrxCAD.Windows

- gmdb.dll file includes the following name space:

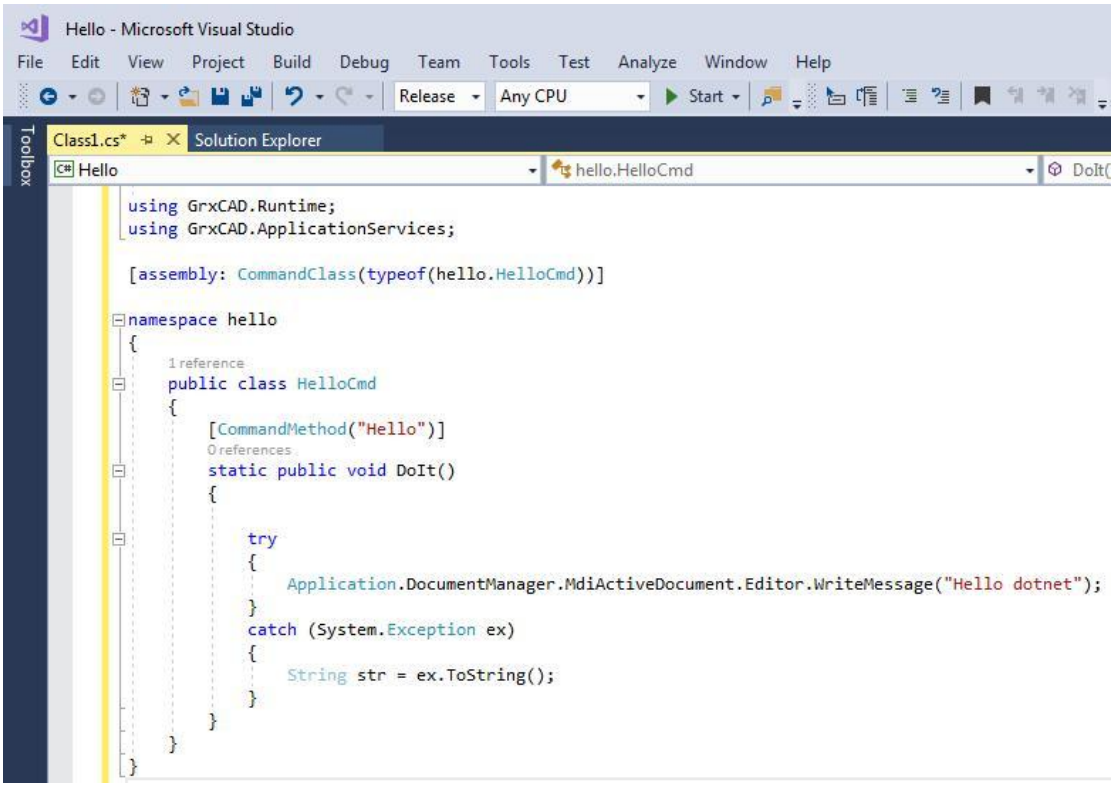
GrxCAD.Colors
GrxCAD.DatabaseServices
GrxCAD.DatabaseServices.Filters
GrxCAD.Export_Import
GrxCAD.Geometry
GrxCAD.GraphicsInterface
GrxCAD.GraphicsSystem
GrxCAD.LayerManager
GrxCAD.Runtime

- GrxCAD.Interop.dll files shows the com corresponding .NET interface.

4.3. Add Codes

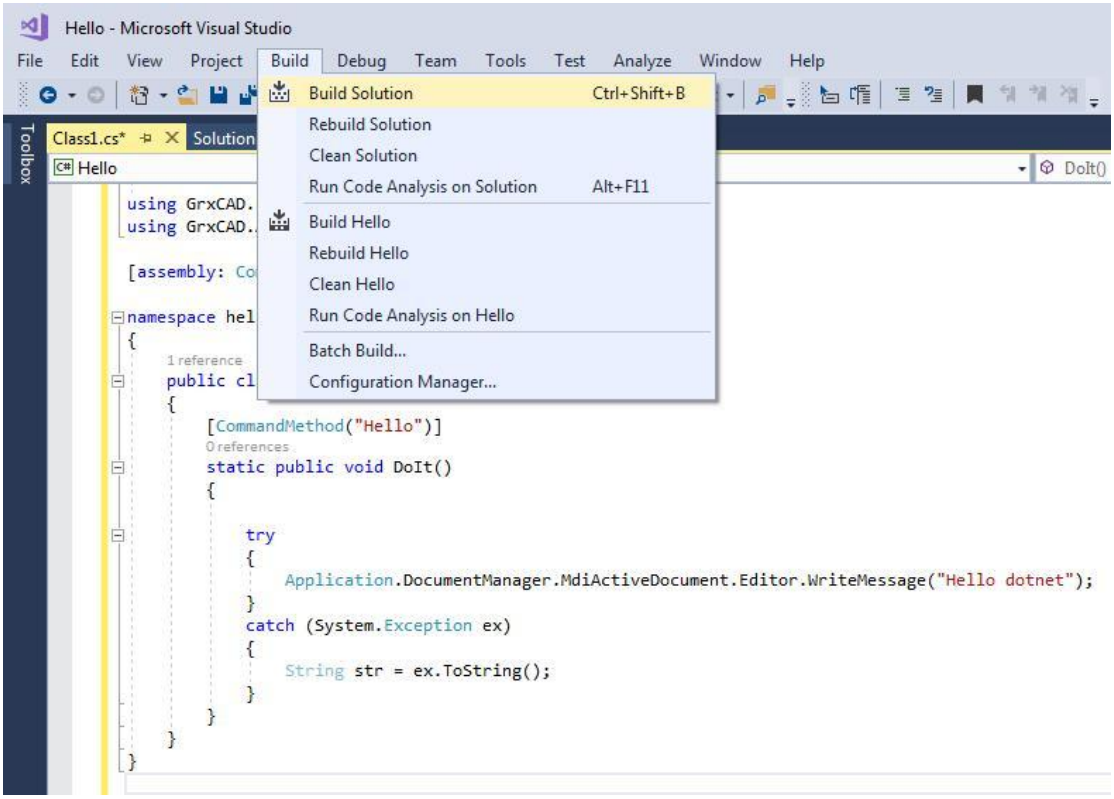
Using the code below to replace the existing code in Class1.cs file in Hello project.

```
using System;

using GrxCAD.Runtime;
using GrxCAD.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();
            }
        }
    }
}
```



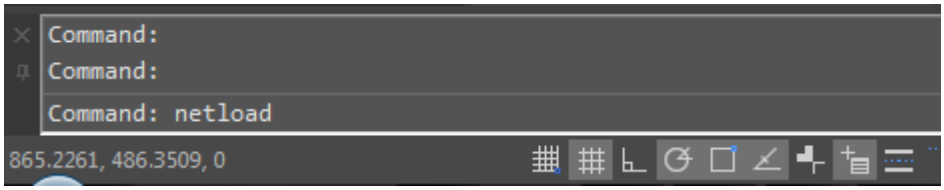
4.4. Compile Program

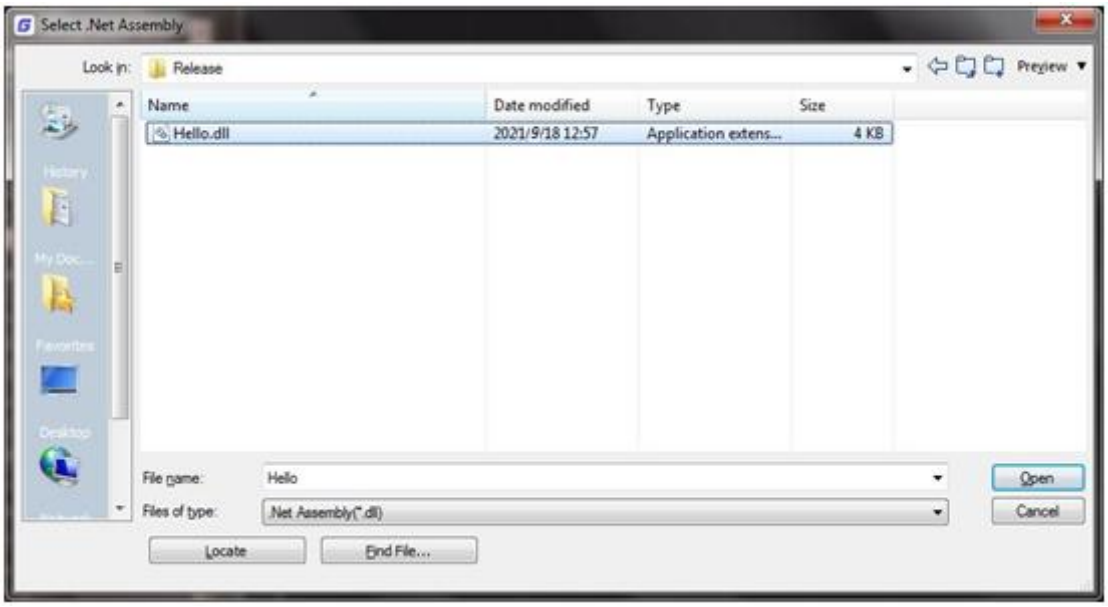
Click Visual Studio 2017 menu item [Build] » [Build solution], then you will get a “Hello.dll” file from D:\test\Hello\Hello\bin\Release directory.



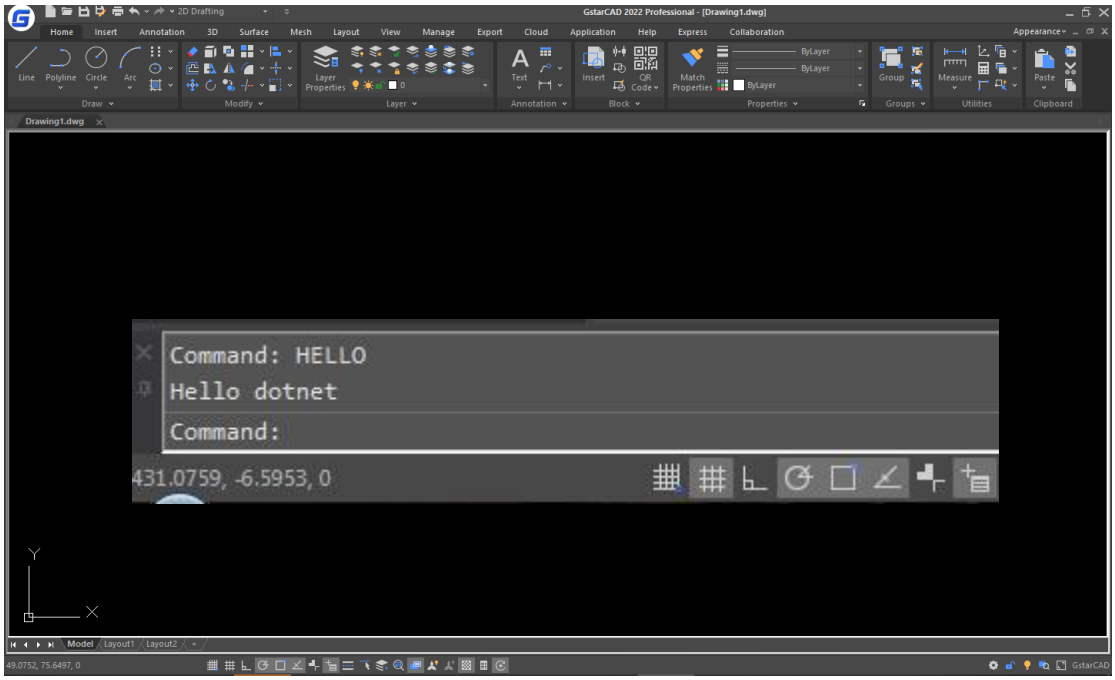
4.5. Run Program

Run GstarCAD and input "netload" at command line, there will a "Select .Net Assembly" dialog box pops out, select the "Hello.dll" file and click Open button to load it.





Input "hello" command at command line, “Hello dotnet” will display in the command line.



5. VS2017 VB .NET Development Sample

The following takes the creation of the "Hello" project as an example to illustrate the creation process. And assumes that the GstarCAD SDK is installed in the "c:\grxsdk" directory.

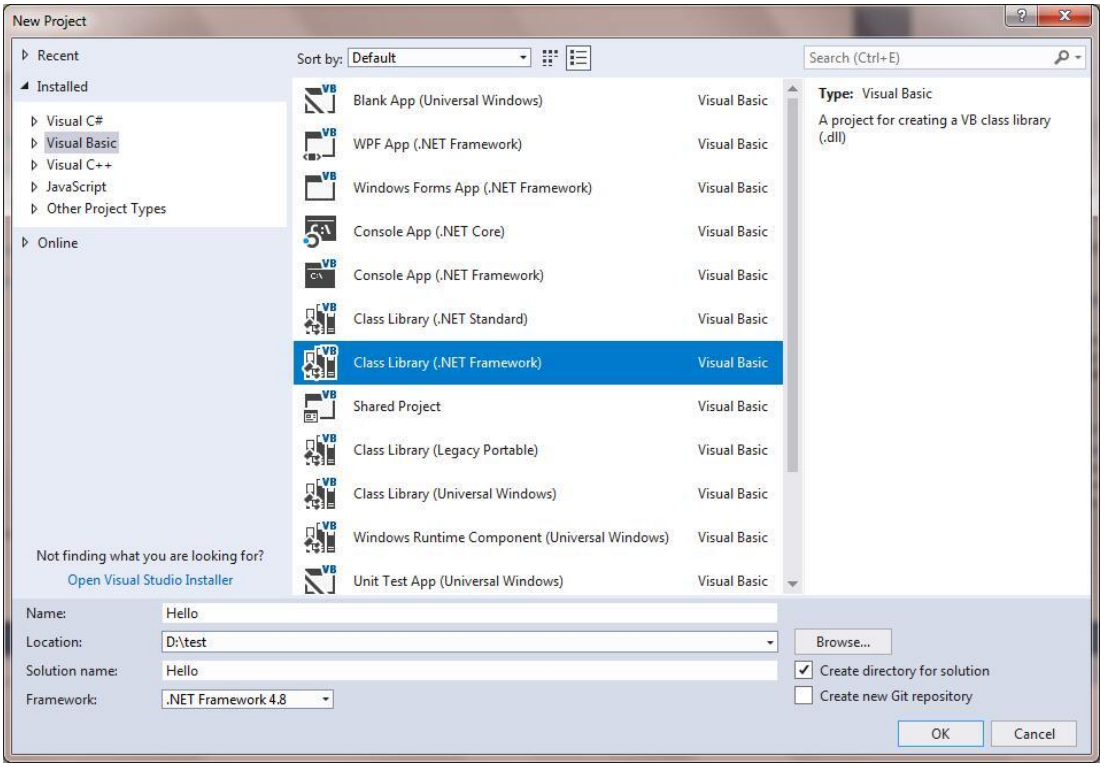
5.1. Create a .NET Project

5.1.1. Run Microsoft Visual Studio 2017

Please click [File] → [New] → [Project]. After clicked the menu Project, the New Project dialog box will pops out. Then select Visual C# in the catalogue Installed Templates and click [Class Library (.Net Framework)].

5.1.2. Input Project Saved Path and Project Name

Type "Hello" at the name label in "New Project" dialog box and choose .NET Framework4.8, as shown below:



Remark： Please choose **.NET Framework 4.8** at the frame pull-down list.

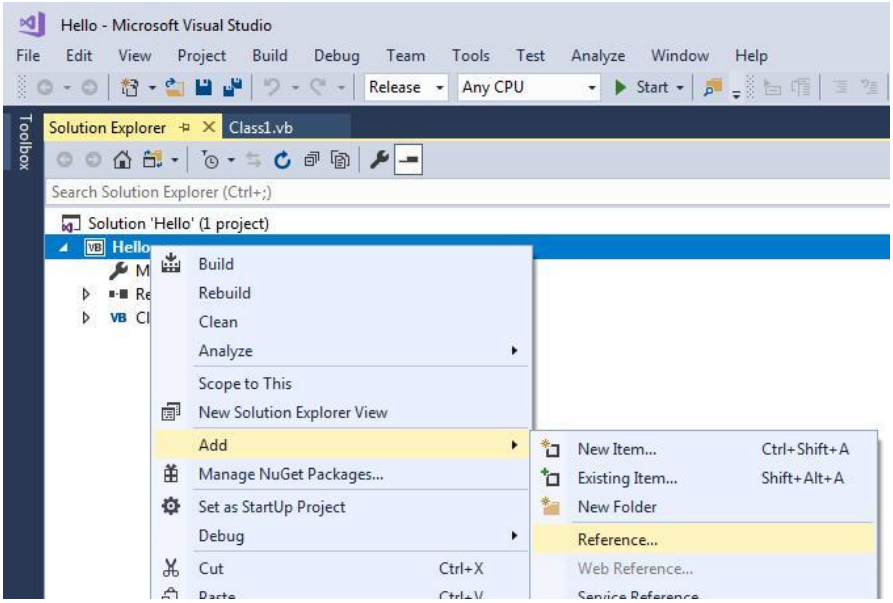
5.1.3. Finish to Create a New Project

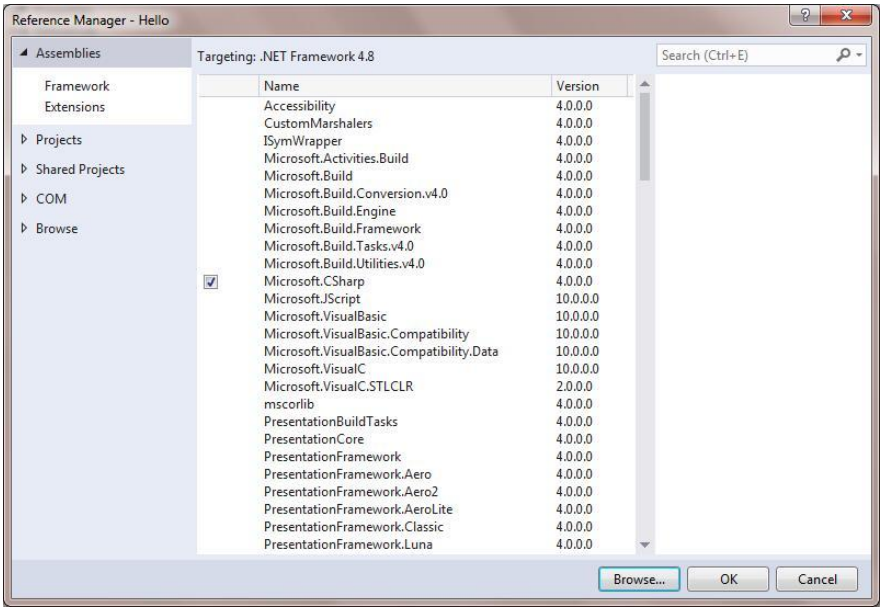
After finished operation above, click OK button in the dialog to finish the new project creating.

5.2. Set the Reference Library

5.2.1. Add Reference

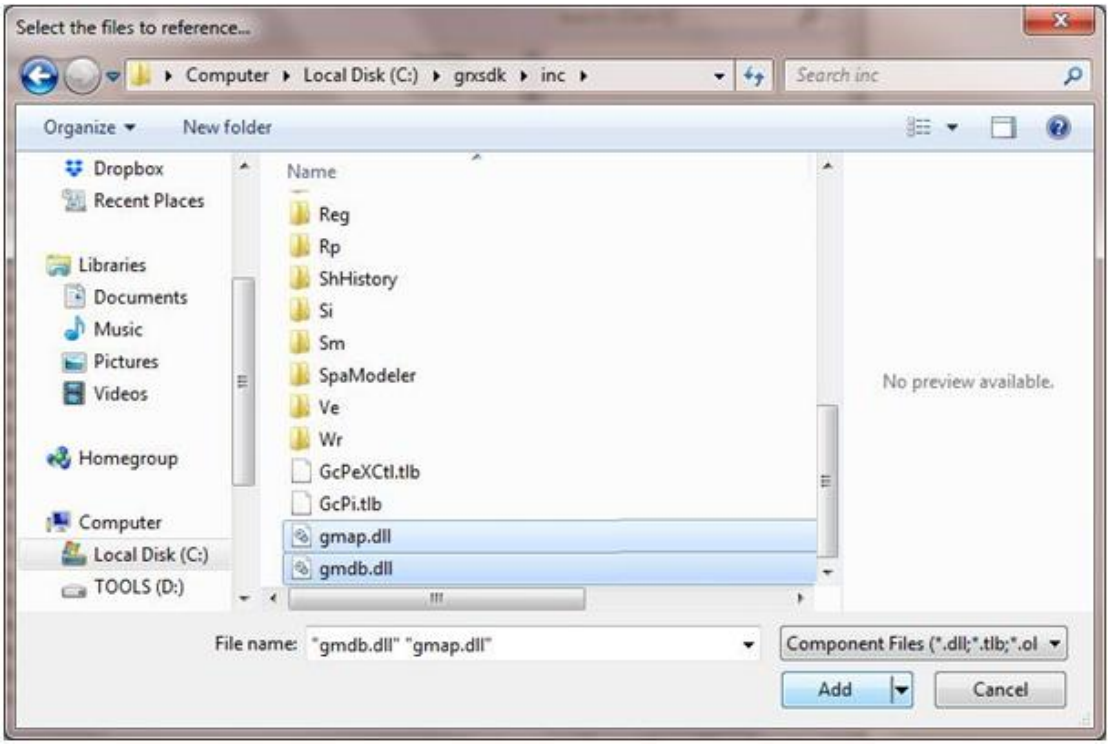
In VS 2017, find the Hello project you have been created on the solution explorer. Select the Hello project and right click to select Add Reference. After the "Reference Manager" dialog box pops out, click Browse button as shown below.

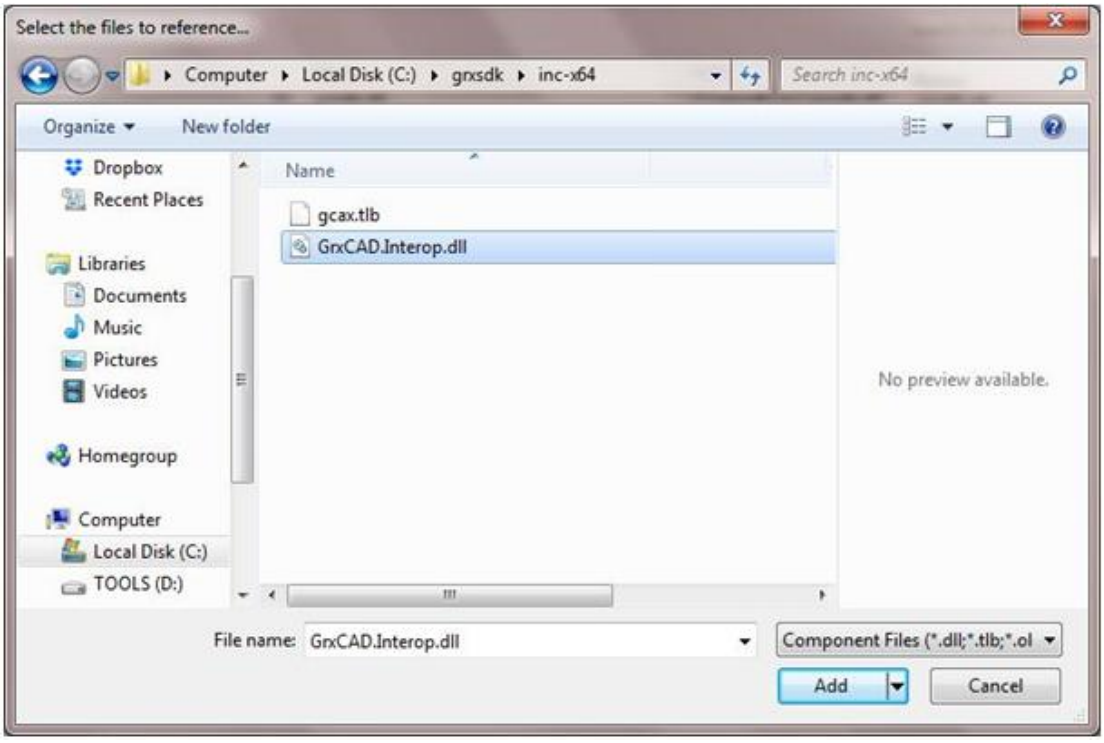




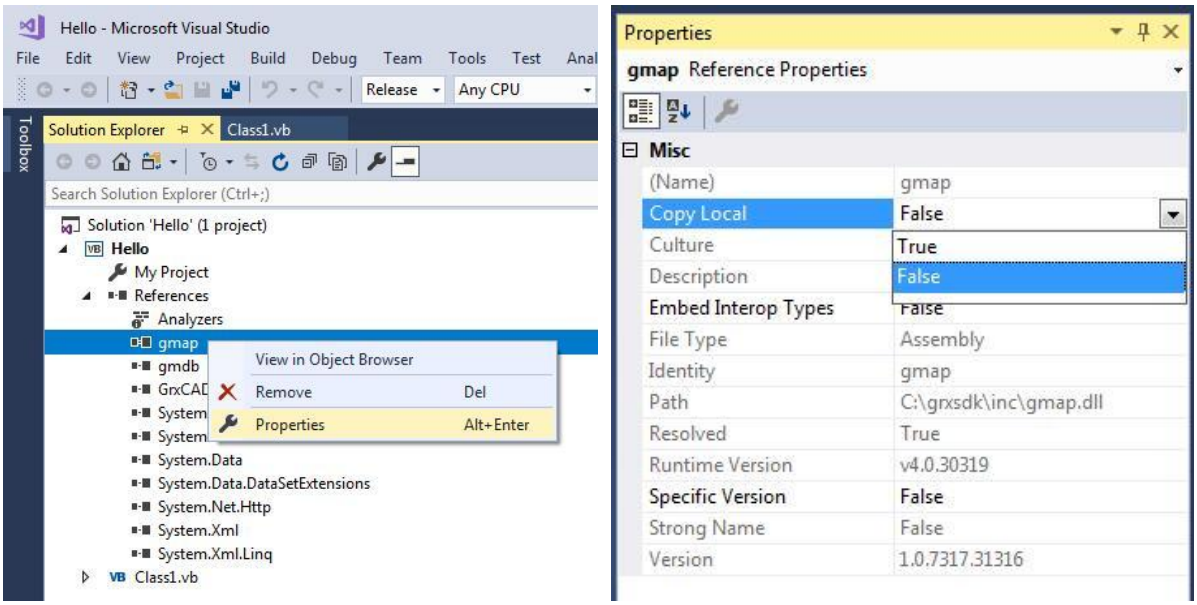
5.2.2. Choose and Add Reference Files

Find the GstarCADSDK installation path, for example, “C:\grxsdk\inc”. Select gmap.dll and gmdb.dll files in “c:\grxsdk\inc-x86” and add them by clicking Add button (If you use GstarCAD 64-bit version, please select GrxCAD.Interop.dll in inc-x64).





Remark: gmap.dll and gmdb.dll have to be added but file GrxCAD.Interop.dll is optional. After adding the files, right click each added reference file, click Property at the context menu, then the "Reference Properties" palette pops up, change the "Copy Local" property from True to False. The following image shows how to change "Copy Local" property for gmap.dll file. Please take it as reference to change gmdb.dll file's Copy Local property.



- gmap.dll includes the following name space:

GrxCAD.ApplicationServices
GrxCAD.EditorInput
GrxCAD.Internal
GrxCAD.PlottingServices
GrxCAD.Publishing
GrxCAD.Runtime
GrxCAD.Windows

- gmdb.dll includes the following name space:

GrxCAD.Colors
GrxCAD.DatabaseServices
GrxCAD.DatabaseServices.Filters
GrxCAD.Export_Import
GrxCAD.Geometry
GrxCAD.GraphicsInterface
GrxCAD.GraphicsSystem
GrxCAD.LayerManager
GrxCAD.Runtime

- GrxCAD.Interop.dll shows the com corresponding .NET interface.

5.3. Add Codes

Using the code below to replace the existing code in Class1.vb file in Hello project.

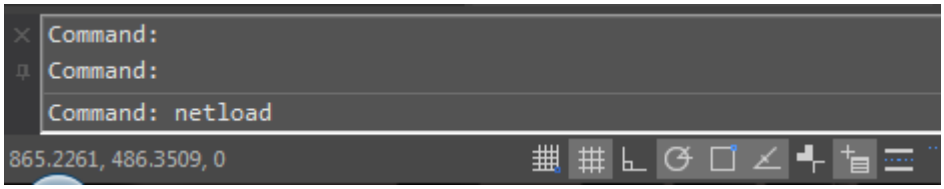
```
Imports System
Imports GrxCAD

Imports GrxCAD.Runtime

Imports GrxCAD.DatabaseServices
Imports GrxCAD.Geometry

Imports GrxCAD.ApplicationServices
Imports GrxCAD.EditorInput

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

Namespace GRXTest
Public Class MyCommands

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

End Class
End Namespace
```

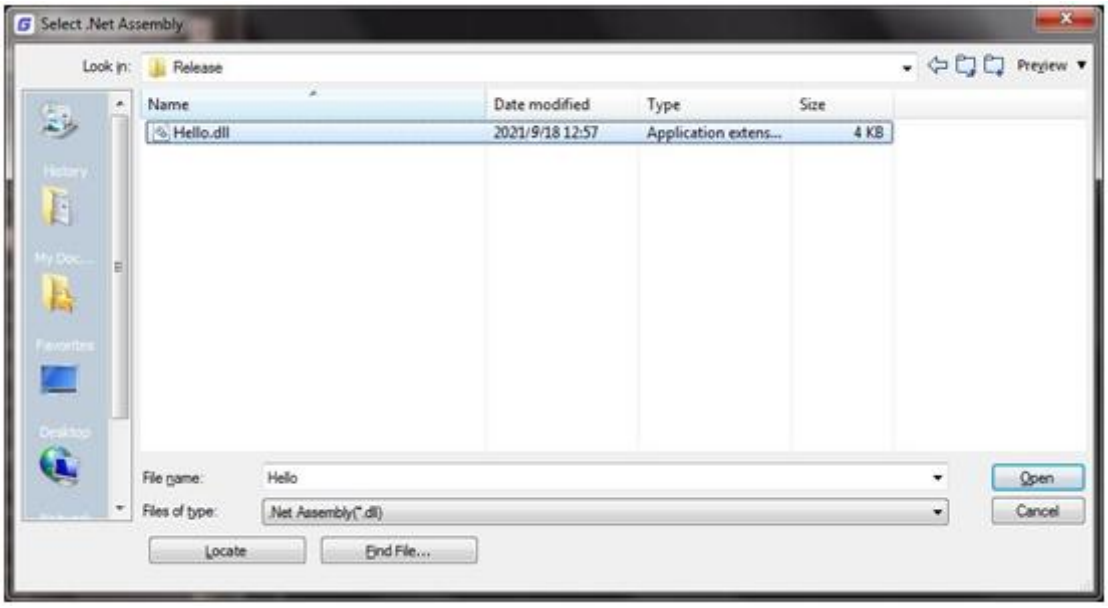
5.4. Compile Program

Click Visual Studio 2017 menu item [Build] » [Build solution], then you will get a “Hello.dll” file from D:\test\Hello\Hello\bin\Release directory.

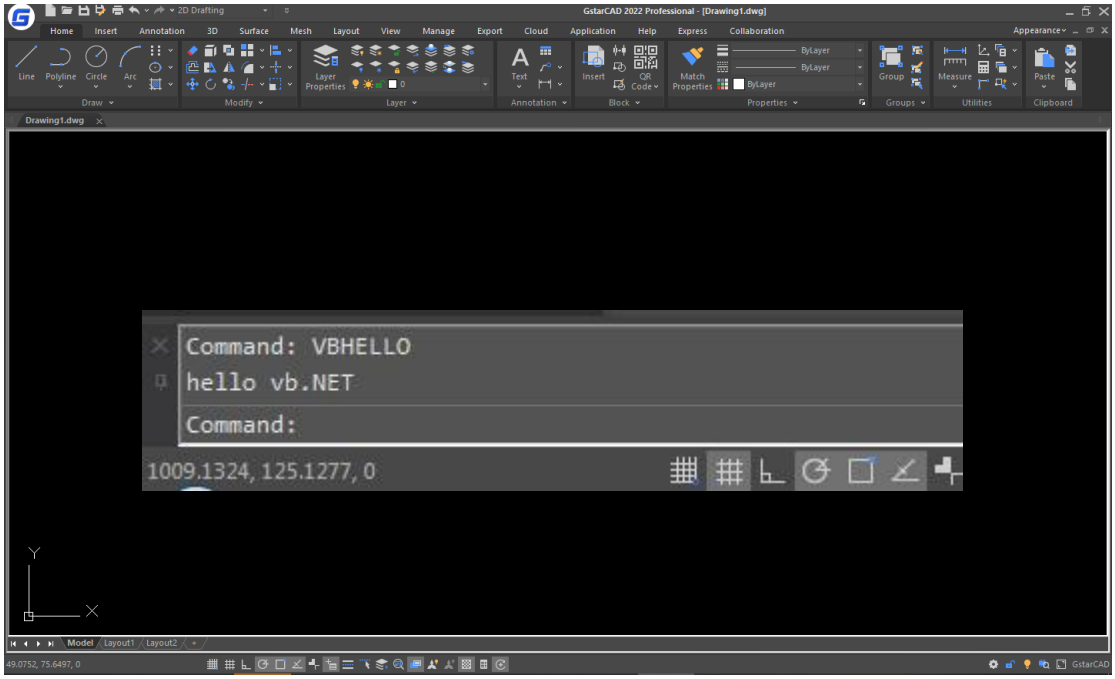
5.5. Run Program

Run GstarCAD and input "netload" at command line, there will a "Select .Net Assembly" dialog box pops out, select the "Hello.dll" file and click Open button to load it.





Then run GstarCAD, input "vbhello" command at command line, “hello vb.NET” will show in the command line.



6. DotNet Development Special Usage

Some differences between GstarCAD DotNet and AutoCAD DotNet:

6.1. Using

Code like “var mapping = new IdMapping()” need to add “using” function first.

For example “var mapping = new IdMapping()” in GstarCAD should be：

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

6.2. ResultBuffer

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

ResultBuffer.UnmanagedObject is the type of **OdDbResbuf**;

ResultBuffer.ResbufObject is the type of **resbuf**;

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

6.3. Alternative method to import Unmanaged ARX Function

The alternative method to import unmanaged arx function **acedCmd,acedCommand**:

Use **GrxCAD.EditorInput.Editor.Command(params object[])** to replace original functions.

6.4. Alternative Method

Autodesk.AutoCAD.EditorInput.Editor.TraceBoundary(Autodesk.AutoCAD.Geometry.Point3d, bool) alternatvie method：
GrxCAD.EditorInput.Editor.

If **TraceBoundary(Autodesk.AutoCAD.Geometry.Point3d, bool)** has not been provided，you can use the following method to replace.

```
[CommandMethod("TESTCMD")]
public void TestCommand()
{
    var db = Application.DocumentManager.MdiActiveDocument.Database;
    var ed = Application.DocumentManager.MdiActiveDocument.Editor;
```

```
    //new a DBObjectCollection for boundary set
    DBObjectCollection objColl = new DBObjectCollection();
    using (var tr = db.TransactionManager.StartTransaction())
    {
    var curSpace = tr.GetObject(db.CurrentSpaceId, OpenMode.ForRead) as
BlockTableRecord;
    var drawOrderTable = tr.GetObject(curSpace.DrawOrderTableId,
OpenMode.ForRead) as DrawOrderTable;
    ObjectIdCollection coll = drawOrderTable.GetFullDrawOrder(0);
    ObjectId entLast =    coll[coll.Count-1];

    //use "Y" or "N" whether or not to detect islands
    //instead of "0,0" when using another seed point
    ed.Command("-BOUNDARY", "A", "I", "Y", "", "0,0", "");

    coll = drawOrderTable.GetFullDrawOrder(0);
    int nIndex = coll.IndexOf(entLast);
    for (int i = nIndex; i >= 0; --i)
    {
        coll.RemoveAt(i);
    }
    //clone the boundaries created by -BOUNDARY command to DBObjectCollection
    foreach (ObjectId id in coll)
    {
        DBObject obj = tr.GetObject(id, OpenMode.ForRead) as DBObject;
        DBObject tmp = (DBObject)obj.Clone();
        objColl.Add(tmp);
    }
    //abort the transaction to avoid the boundaries created by -BOUNDARY command
posting to db
    tr.Abort();
    }

    //this is a test code to post the cloned boundaries to current space
    using (var tr = db.TransactionManager.StartTransaction())
    {
    var curSpace = tr.GetObject(db.CurrentSpaceId, OpenMode.ForWrite) as
BlockTableRecord;
    foreach (DBObject obj in objColl)
    {
        curSpace.AppendEntity((Entity)obj);
```

```
            tr.AddNewlyCreatedDBObject(obj, true);
        }
        tr.Commit();
        }

        //dispose the DBObjectCollection after used it,and dispose the item in it if
necessary,here we do
        //not dispose the DBOjbect in DBObjectCollection because we post it to current
space before
        objColl.Dispose();
}
```

6.5. C# .NET and VB .NET Name Space Modification

- VB.NET：

The content after **Imports**，change **Autodesk.AutoCAD** to **GrxCAD**

- C#.NET：

The content after **using**，change **Autodesk.AutoCAD** to **GrxCAD**

- If use COM：

With VB.net, add **Imports GcadVbaLib**
With C#.net, add **using GcadVbaLib**
Delete **Autodesk.AutoCAD.Interop.Common**

For the COM object name in the code, such as **AcadLWPolyline**, please change the prefix Acad to Gcad, for example **AcadLWPolyline** should be change to **GcadLWPolyline**.

If some codes not sure, you can check from the object browser. For example ACAD_COLOR, you can check from the browser whether there is GCAD_COLOR, if there is, just replace it.

7. Copyright

Copyright reserved: Gstarsoft Co.,Ltd

Usage License: Allows copying, quote any part of this document. Changes of any part of this document are forbidden without authorization. Please keep this statement when copying and quoting it, otherwise it will be held liable.

* AutoCAD® is a product of Autodesk® Company which is one of the CAD software solution providers. ARX is an abbreviated name of ObjectARX, which is AutoCAD® Runtime eXtension, and C++ SDK provided by Autodesk®.

GstarCAD 2022

https://www.gstarcad.net/

Gstarsoft