



# **.NET Migration Guide**

---

**GstarCAD 2022**



## Table of Contents

1.	Brief Introduction .....	4
2.	Development Environment .....	5
3.	Original Code .....	6
4.	Install GstarCAD SDK .....	7
5.	VS2017 C# .NET Migration Process .....	8
5.1.	Change Reference Files .....	8
5.1.1.	Remove the Reference.....	8
5.1.2.	Add Reference .....	8
5.1.3.	Load Reference File .....	10
5.2.	Change the Target Framework.....	11
5.3.	Modify the Name in Name Space .....	12
5.4.	Compile Program .....	12
6.	VS2017 VB.NET Migration Process .....	13
6.1.	Change Reference Files .....	13
6.1.1.	Remove the Reference.....	13
6.1.2.	Add Reference .....	13
6.1.3.	Load Reference Files .....	14
6.2.	Change the Target Frame .....	16
6.3.	Modify the Name in Name Space .....	16
6.4.	Compile Program .....	16
7.	DotNet Application Development Special Usage.....	17
7.1.	Using.....	17
7.2.	ResultBuffer .....	17
7.3.	Alternative method to import Unmanaged ARX Function .....	17
7.4.	Alternative Method .....	17
7.5.	Other Usages.....	19
7.5.1.	Abtain GCAD Application Object under VB enveriment	19
7.5.2.	C#.NET and VB.NET Name space Modifaction.....	20
7.6.	Replacement Sample.....	21

7.6.1. Namespace Replacement .....	21
7.6.2. Class Name and Property Name Replacement Sample 21	
8. Copyright.....	22

## **1. Brief Introduction**

Application Developers from various industries have developed a large number of ARX applications based on AutoCAD®, and one of the highlights of GstarCAD GRX is GstarCAD can migrate the ARX program which runs on the AutoCAD platform to GstarCAD with almost no code changing. Realized source-level compatibility with AutoCAD®ARX programs, which means one set of code can be suitable for two platforms. Developers can refer this guide to easily migrate the code to GstarCAD.

## 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. Original Code**

The original code should be DOTNET application source code which is from the DONET ObjectARX 2010 or higher version.

## 4. Install GstarCAD 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 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 is the sample to develop with ARX API.
- SimplePalette is the sample to develop the type modules for palette

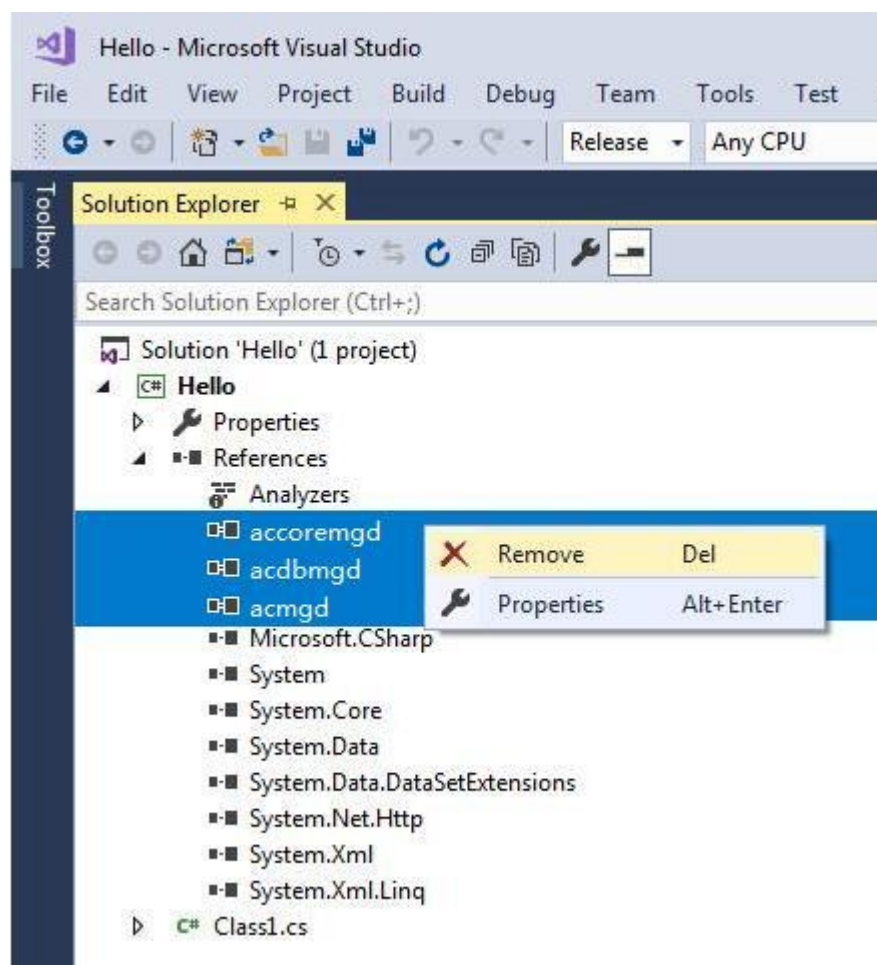
## 5. VS2017 C# .NET Migration Process

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.Change Reference Files

#### 5.1.1. Remove the Reference

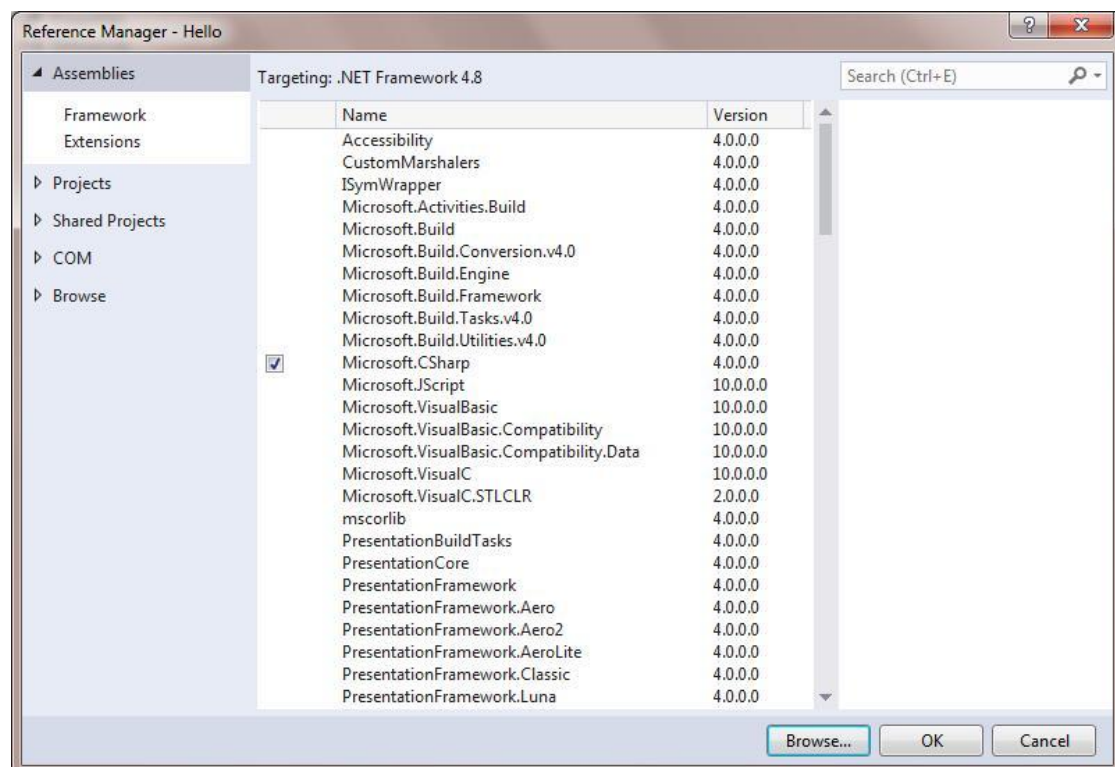
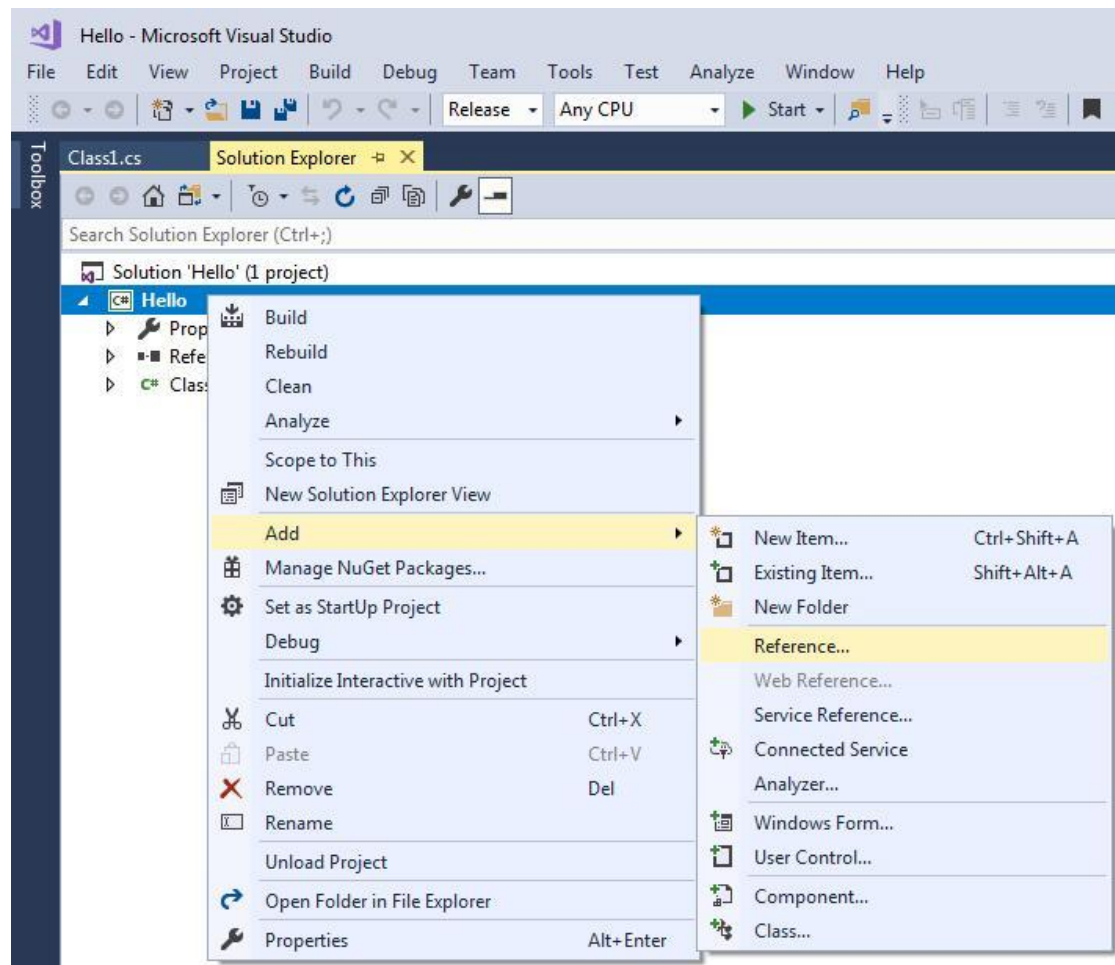
Remove acad reference files, as shown below.



#### 5.1.2. Add Reference

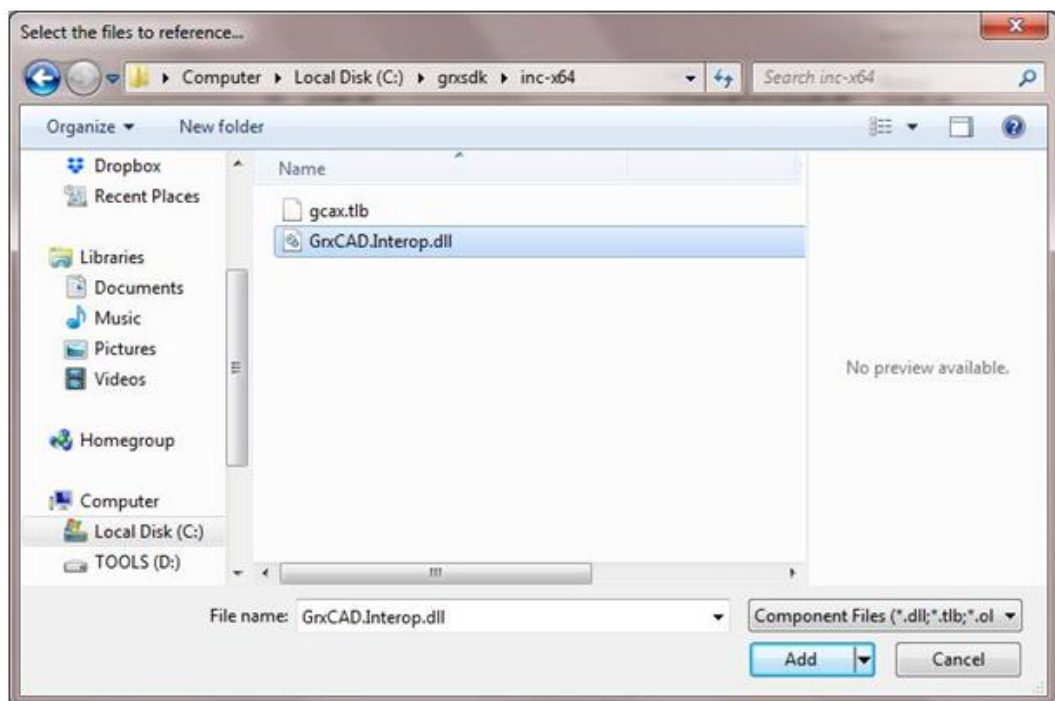
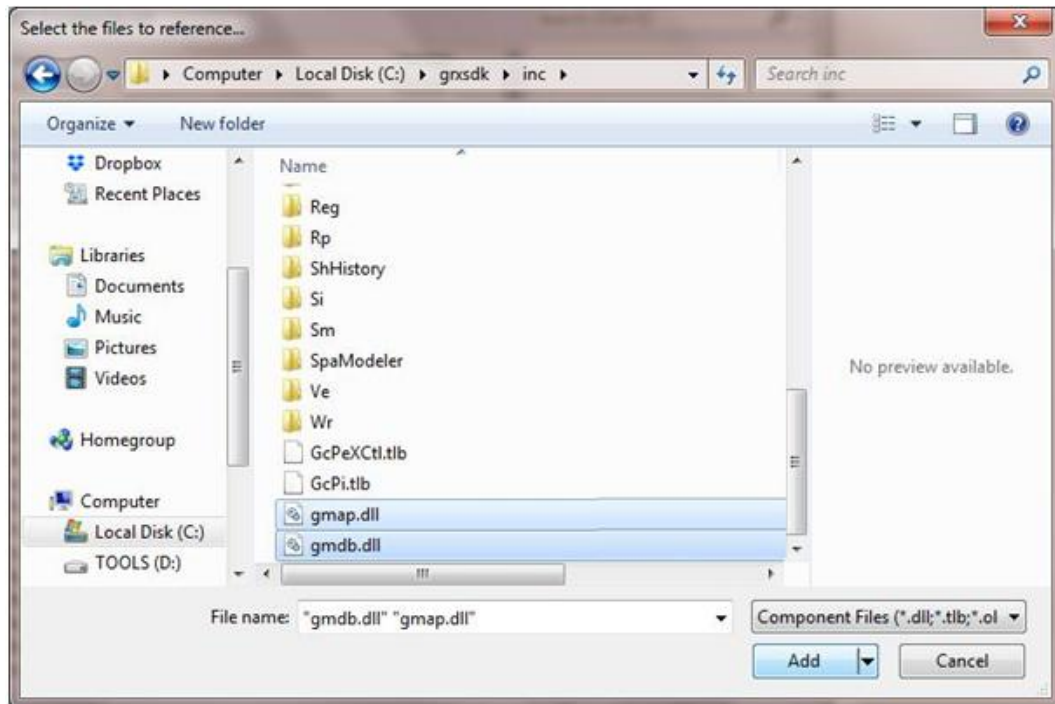
In VS 2017, find the Hello project you have been created on the solution explorer. Select the Hello project and right click to select [Add Reference](#). After the "Reference Manager" dialog box pops out, click [Browse](#) button as shown below.





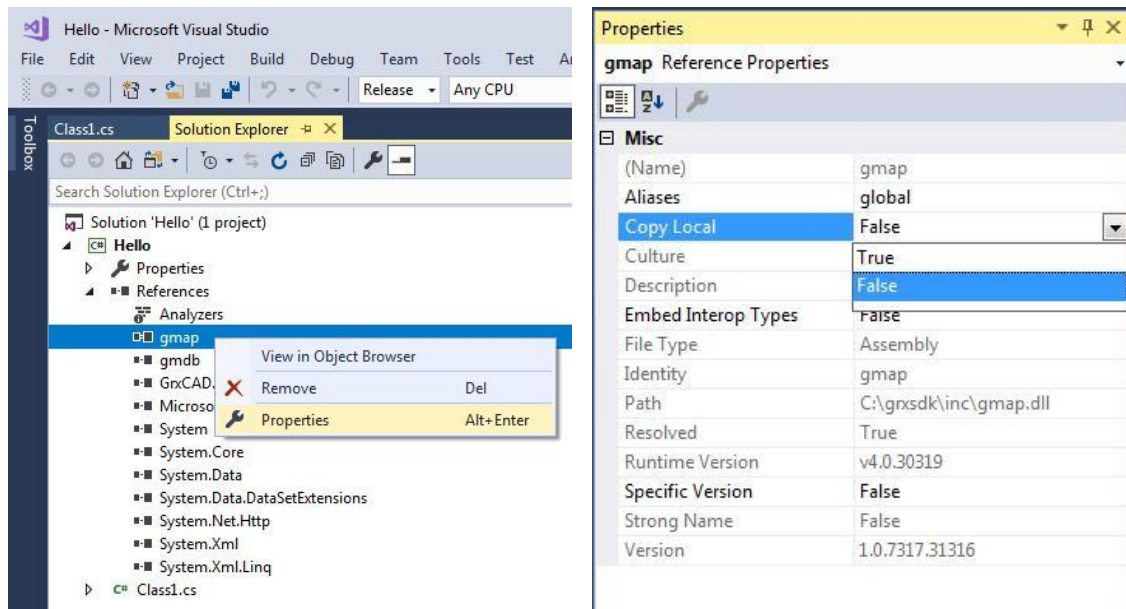
### 5.1.3. Load Reference File

Find the GstarCAD SDK 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)



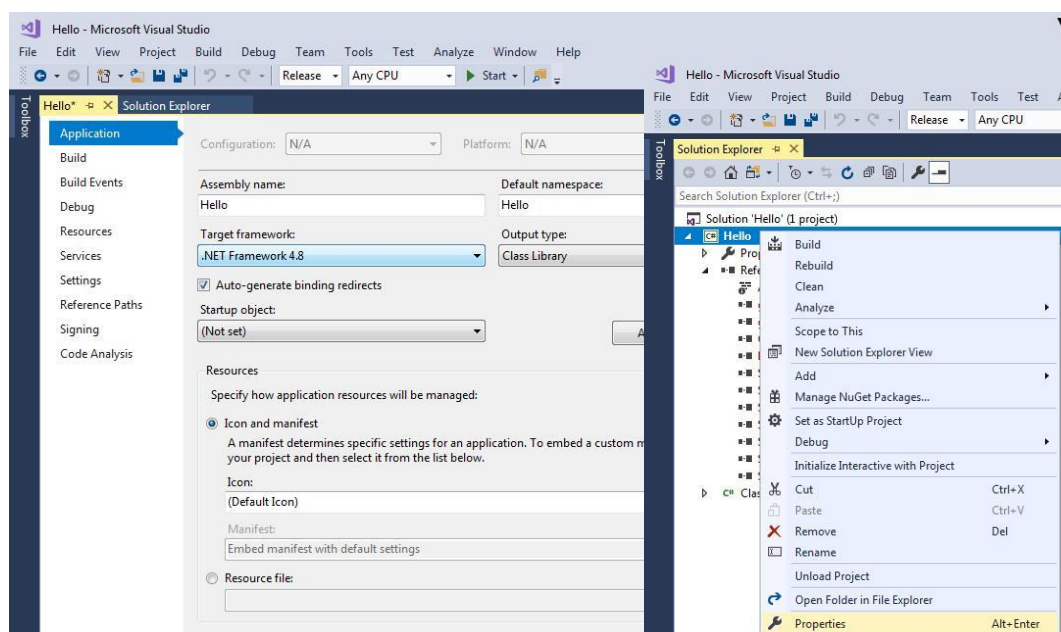
Remark: gmap.dll and gmdb.dll have to be added but file GrxCAD.Interop.dll is optional.  
After adding the files, right click each added reference file, click [Property](#) at the context

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.



## 5.2.Change the Target Framework

In VS 2017, find the Hello project you have been created on the solution explorer. Right click to select [Properties](#). After the new page opens, please change the "Target framework" to [Framework 4.8](#) from the pull-down list in [Application](#) menu.



### **5.3.Modify the Name in Name Space**

Replace the Autodesk.AutoCAD corresponding Name sapce to GrxCAD.

### **5.4.Compile Program**

Click Visual Studio 2017 menu item [\[Build\]](#) » [\[Build solution\]](#), regenerate the dll file in the corresponding directory.

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

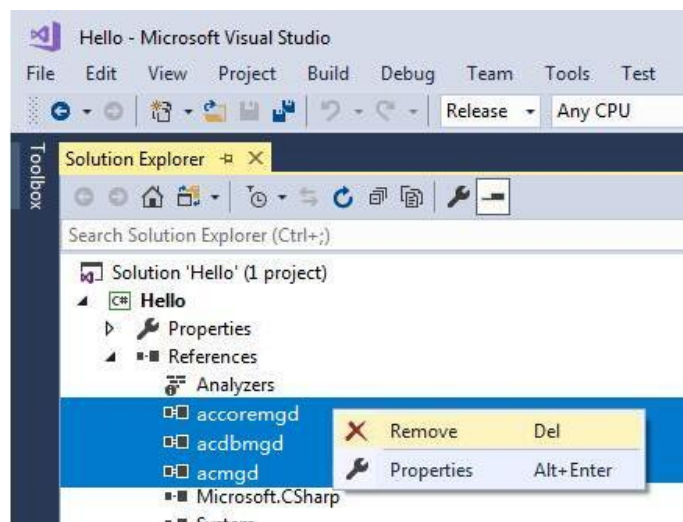
## 6. VS2017 VB.NET Migration Process

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.

### 6.1.Change Reference Files

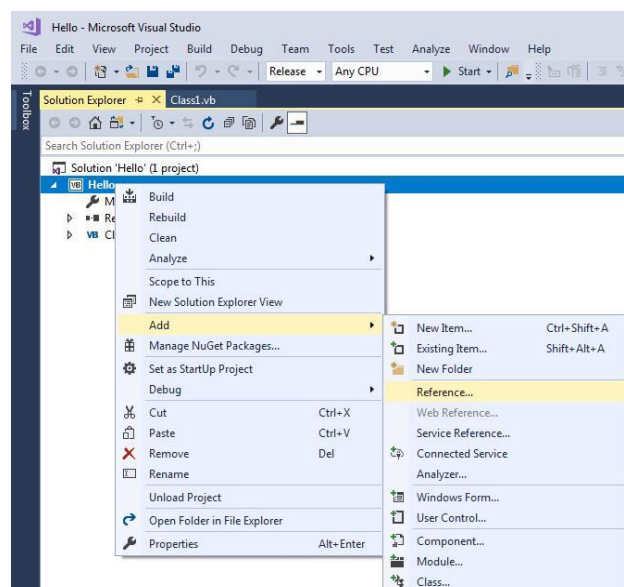
#### 6.1.1. Remove the Reference

Remove acad reference files, as shown below.

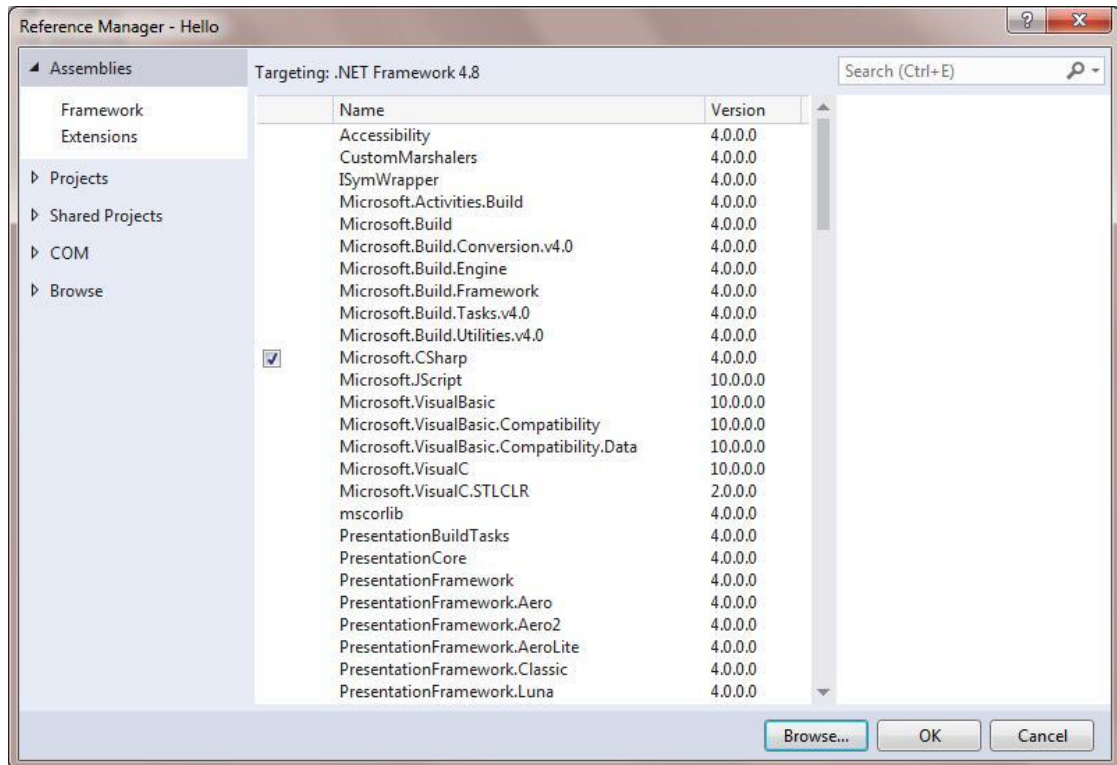


#### 6.1.2. Add Reference

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.

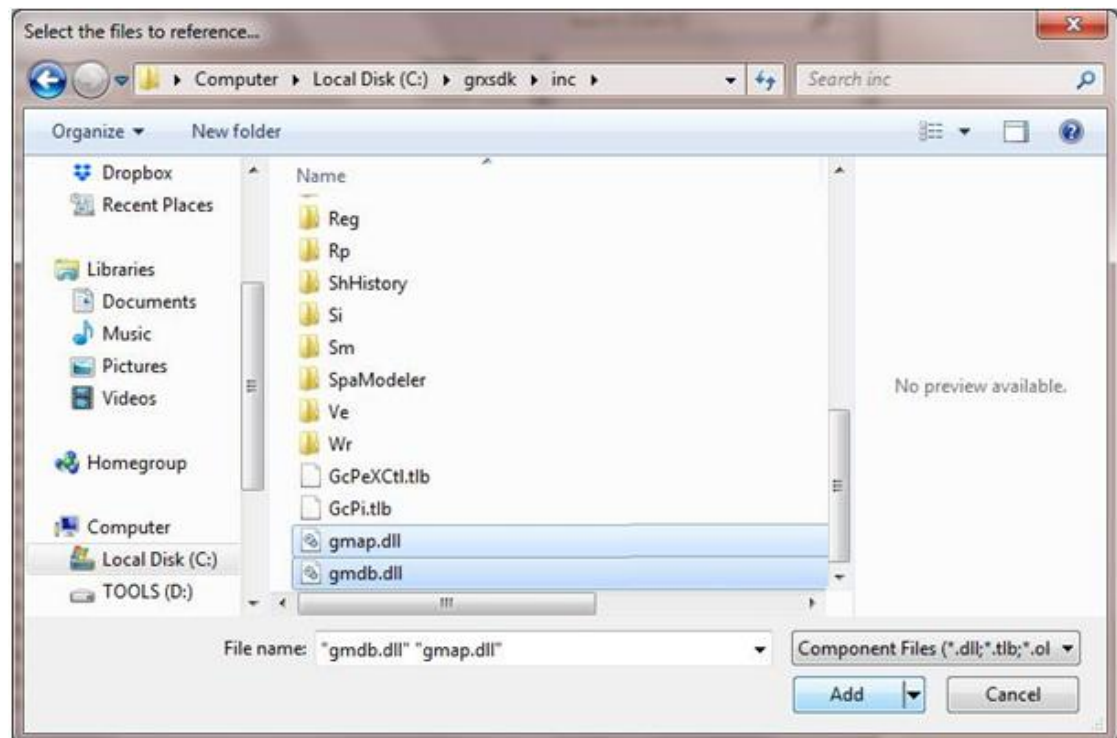


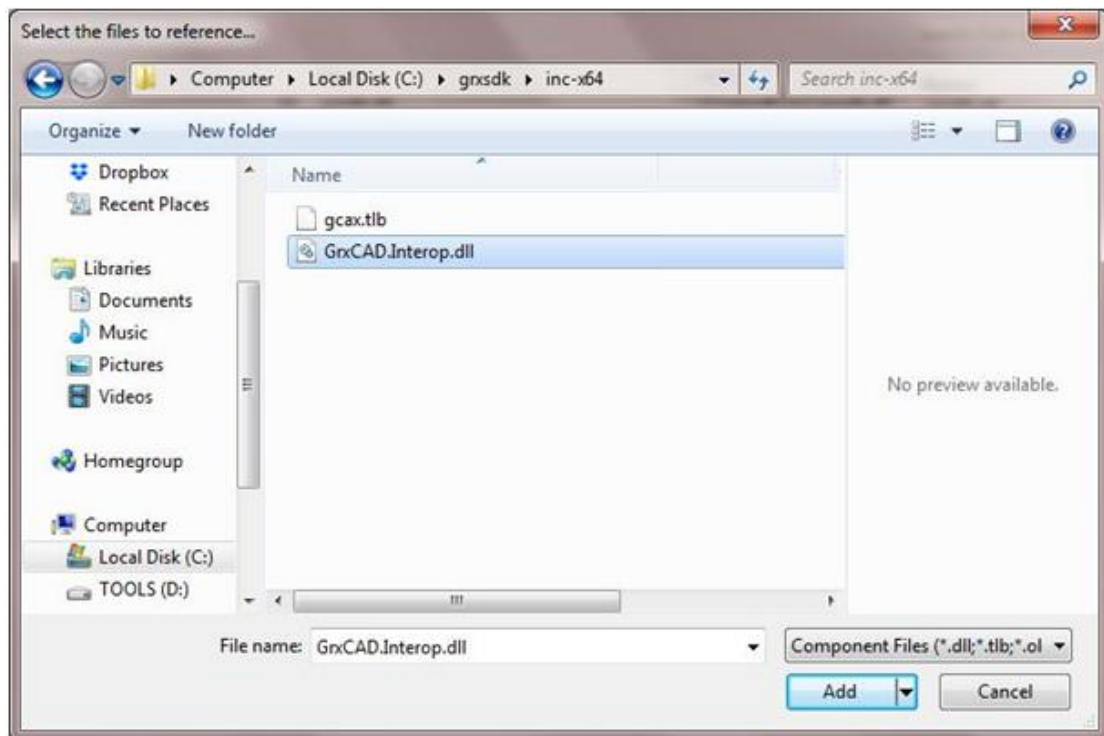




### 6.1.3. Load Reference Files

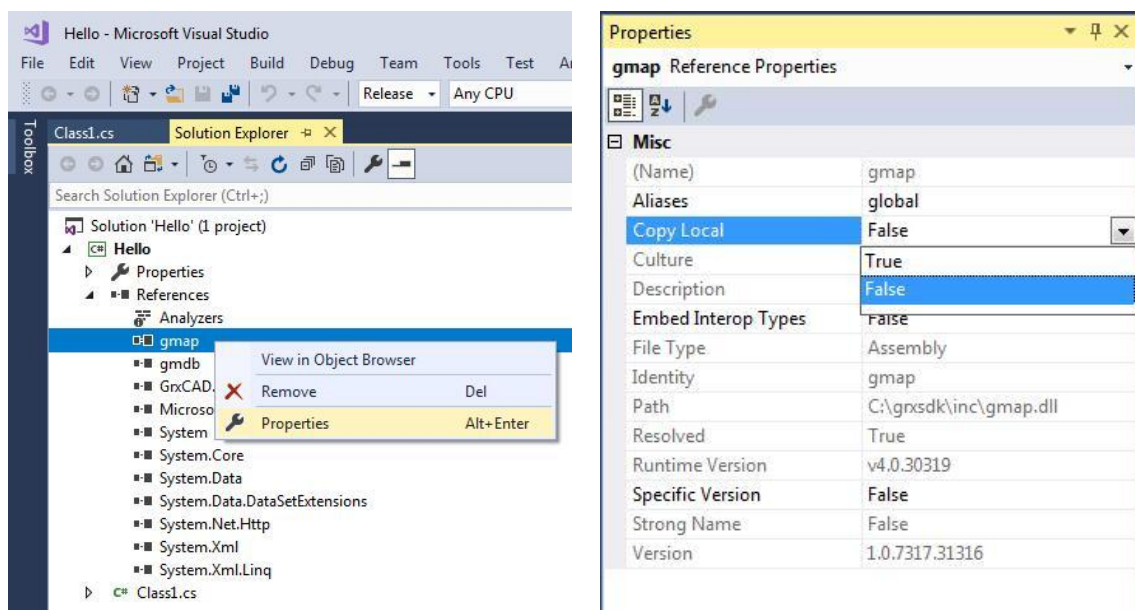
Find the GstarCADSDK installation path, for example, "C:\grxsdk\inc". Select gmap.dll and gmdb.dll files in "c:\grxsdk\inc-x86" and add them by clicking **Add** button (If you use GstarCAD 64-bit version, please select GrxCAD.Interop.dll in inc-x64)





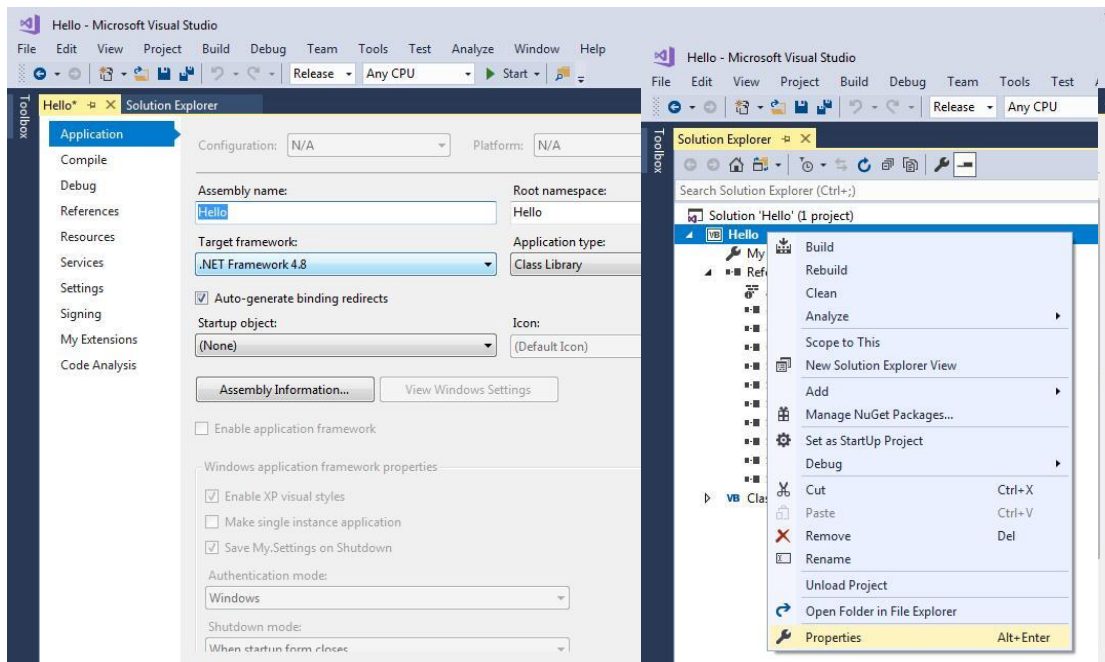
Remark: gmap.dll and gmdb.dll are necessary, GrxCAD.Interop.dll is alternative.

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.



## 6.2. Change the Target Frame

In VS 2017, find the Hello project you have been created on the solution explorer. Right click to select [Properties](#). After the new page opens, please change the "Target framework" to [Framework 4.8](#) from the pull-down list in [Application](#) menu.



## 6.3. Modify the Name in Name Space

Replace the Autodesk.AutoCAD corresponding Name sapce to GrxCAD.

## 6.4. Compile Program

Click Visual Studio 2017 menu item [\[Build\] » \[Build solution\]](#), regenerate the dll file in the corresponding directory.

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



## 7. DotNet Application Development Special Usage

DotNet Application Development Special Usage as shown below.

### 7.1. 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())  
{  
    ...  
}
```

### 7.2. 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)**

### 7.3. 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.

### 7.4. Alternative Method

**Autodesk.AutoCAD.EditorInput.Editor.TraceBoundary(Autodesk.AutoCAD.Geometry.Point3d, bool)** alternative 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 DBObject in DBObjectCollection because we post it to current space
    before
        objColl.Dispose();
    }

```

## 7.5.Other Usages

### 7.5.1. Abtain GCAD Application Object under VB enveriment

After adding C:\grxsdk\inc-x86\gcax.tlb (if your GstarCAD is 64-bit, please choose the 64-bit corrsponding file) as reference, you can abtain GCAD Application object.

```
Imports GrxCAD.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 obtained process window
```

```
        GCadApp.WindowState = GcWindowState.acMin
```

```
    End Sub
```

```
End Class
```

End Sub

End Class

### 7.5.2. C#.NET and VB.NET Name space Modifaction

➤ 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.

## 7.6.Replacement Sample

### 7.6.1. Namespace Replacement

AutoCAD	GstarCAD
Autodesk.AutoCAD.ApplicationServices	GrxCAD.ApplicationServices
Autodesk.AutoCAD.EditorInput	GrxCAD.EditorInput
Autodesk.AutoCAD.GraphicsSystem	GrxCAD.GraphicsSystem
Autodesk.AutoCAD.PlottingServices	GrxCAD.PlottingServices
Autodesk.AutoCAD.Publishing	GrxCAD.Publishing
Autodesk.AutoCAD.Runtime	GrxCAD.Runtime
Autodesk.AutoCAD.Windows	GrxCAD.Windows
Autodesk.AutoCAD.Colors	GrxCAD.Colors
Autodesk.AutoCAD.DatabaseServices	GrxCAD.DatabaseServices
Autodesk.AutoCAD.DatabaseServices.Filters	GrxCAD.DatabaseServices.Filters
Autodesk.AutoCAD.Geometry	GrxCAD.Geometry
Autodesk.AutoCAD.GraphicsInterface	GrxCAD.GraphicsInterface
Autodesk.AutoCAD.LayerManager	GrxCAD.LayerManager

### 7.6.2. Class Name and Property Name Replacement Sample

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

## 8. 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/>