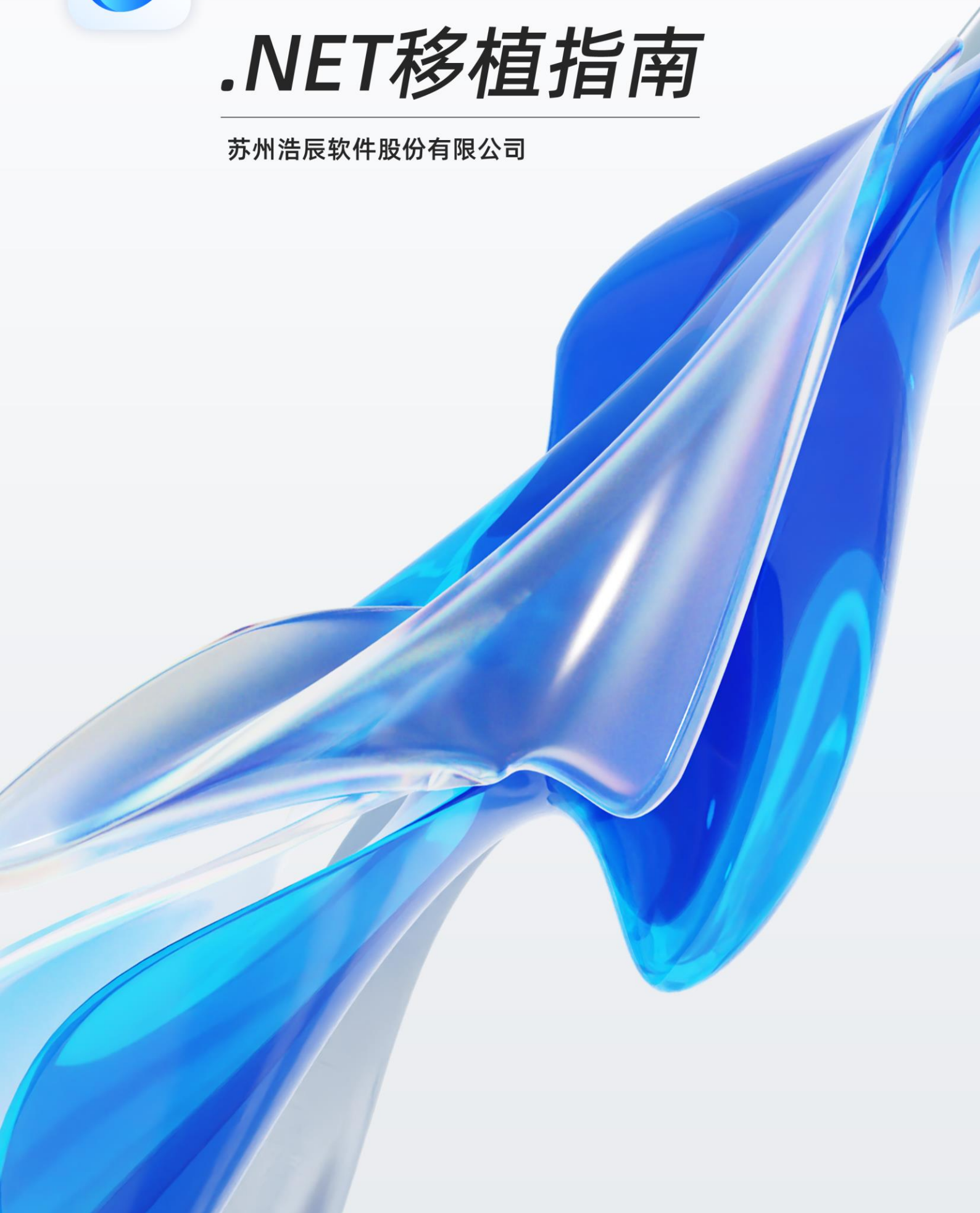




浩辰CAD 2025

.NET移植指南

苏州浩辰软件股份有限公司



目 录

1 简介.....	3
2 开发环境.....	3
3 原始代码.....	4
4 安装浩辰 SDK.....	4
5 VS2017 的 C#.NET 移植步骤.....	5
5.1 修改引用文件.....	5
5.2 修改目标框架.....	10
5.3 修改命名空间名称.....	11
5.4 编译运行.....	11
6 VS2017 VB.NET 移植步骤.....	12
6.1 修改引用文件.....	12
6.2 修改目标框架.....	16
6.3 修改命名空间名称.....	17
6.4 编译运行.....	18
7 DotNet 二次开发的特殊应用描述.....	19
7.1 using 使用.....	19
7.2 ResultBuffer 使用.....	19
7.3 导入非托管 arx 函数的替换方法.....	19
7.4 模式对话框点击按钮后在当前图选择的处理方法.....	19
8 其他事项.....	21
8.1 VB 环境下获取 GCAD Application 对象.....	21
8.2 C#.NET 与 VB.NET 命名空间变更.....	21
9 替换样例.....	23
9.1 命名空间替换样例.....	23
9.2 类名及属性名替换样例.....	23
10 版权声明.....	24

1 简介

国内外各行业的二次开发商已经在 AutoCAD 上开发了大量的 .Net 应用程序，而浩辰 CAD .Net 的一大亮点是用户在不改代码的情况下，就可以把在 AutoCAD 平台上运行的 .Net 程序移植到浩辰 CAD 上来，实现了与 AutoCAD®.Net 程序源码级兼容，即实现“一套代码两个平台”。为了方便客户移植代码，提供本说明。

2 开发环境

➤ Microsoft Visual Studio 2017 （版本 15.9.17）

➤ Windows SDK 10.0.17763.0

➤ 处理器：

基本要求：1.6 GHz 处理器，建议：3.0 GHz 处理器及以上

➤ 内存：

基本要求：2 GB，建议：8 GB 及以上

➤ 操作系统

Windows 11

Windows 10 版本 1507 或更高版本：家庭版、专业版、教育版和企业版（不支持 LTSC 和 Windows 10 S）

Windows 8.1 （带有更新 2919355）：核心版、专业版和企业版

Windows 7 SP1 （带有最新的 Windows 更新）：家庭普通版、家庭高级版、专业版、企业版和旗舰版

显示器分辨率：

传统显示器：1028 x 800 及以上真彩色显示器，包括 4K (3840 x 2160) 显示器

➤ GRX SDK 2025

➤ 浩辰 CAD 2025

➤ .NET Framework 4.8 及以上版本

3 原始代码

移植的源代码应为.NET ObjectARX 2010 或更高版本的.NET 应用程序源代码。

4 安装浩辰 SDK

将 GRXSDK.ZIP 解压到磁盘中，例如 C:\grxsdk。解压完成后会在 C:\grxsdk 路径下生成 5 个文件夹：arx、inc、inc-x64、lib-x64、utils。其中：

➤ arx 目录包含从 arx 程序移植到 GRX 程序使用的头文件，库文件及样例程序等。包含以下目录

- inc 包含从 arx 程序移植到 GRX 程序使用的头文件
- inc-x64 包含 64 位下 com 和 dotnet 使用的文件。
- lib-x64 包含 64 位下引用的 GRX 库文件。
- Samples 包含例子工程文件， 分别是 dotnet, fact_dg, HelloADS, HelloARX, SimplePalette。其中：
 - ✧ Dotnet 文件夹包含 dotnet 使用实例，分别为 addline, hello, vbhello。其中：
 - Addline 为 dotnet 增加实体线的实例。
 - Hello 为 dotnet 输出提示信息的实例。
 - Vbhello 为如何在 vb.net 下使用 dotnet 接口进行开发的实例。
 - ✧ fact_dg 为 lisp 函数定义实例。
 - ✧ HelloADS 为如何开发 ads 类型工程的实例。
 - ✧ HelloARX 为如何使用 arx 接口进行开发的实例。
 - ✧ SimplePalette 为如何使用 palette 类型模块的实例。
- utils 目录包含作为 GRX 扩展的应用程序的子目录，包括用于边界表示的 brep 等扩展功能开发接口。

- inc 包含开发 GRX 程序使用的头文件。
- inc-x64 包含 64 位下 com 和 dotnet 使用的文件。
- lib-x64 包含 64 位下引用的 GRX 库文件。
- utils 目录包含作为 GRX 扩展的应用程序的子目录，包括用于边界表示的 brep 等扩展功能开发接口。

5 VS2017 的 C#.NET 移植步骤

以下以移植“Hello”工程为例，说明移植过程，并假设浩辰 CAD 的 SDK 安装在 c:\grxsdk 目录下。

5.1 修改引用文件

5.1.1 移除引用文件

移除 ARX 中的引用库文件，如图 5.1 所示。

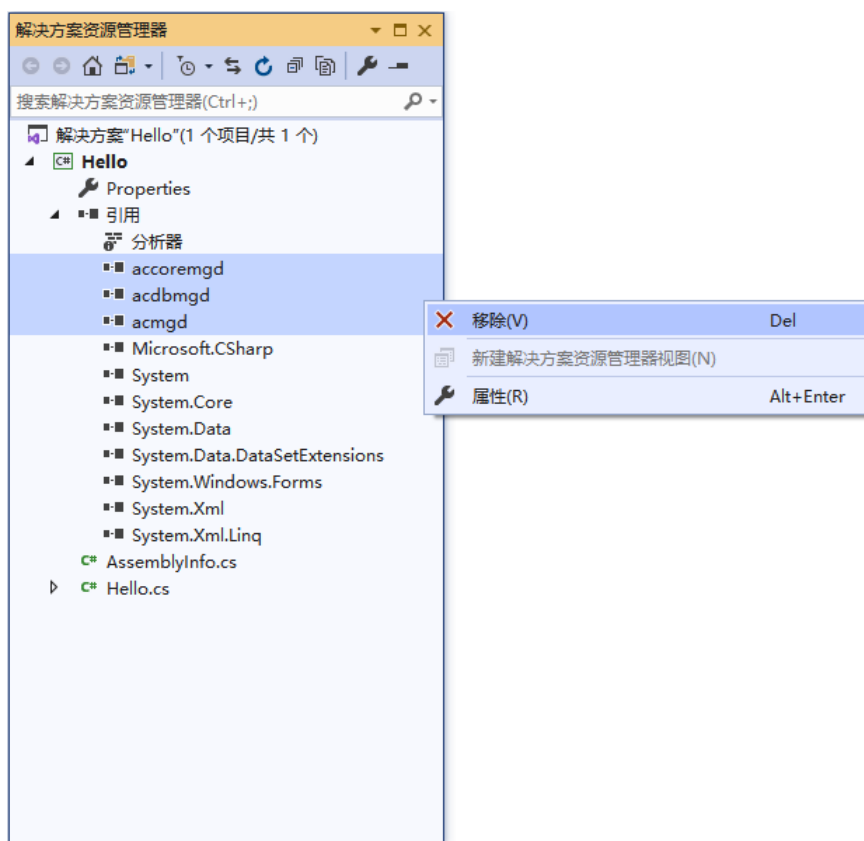


图 5.1

5.1.2 选择引用文件

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称，例如“Hello”，点右键，点【添加】->【引用】菜单项，会弹出“引用管理器”对话框，选择底部【浏览】如图 5.2、图 5.3 所示。

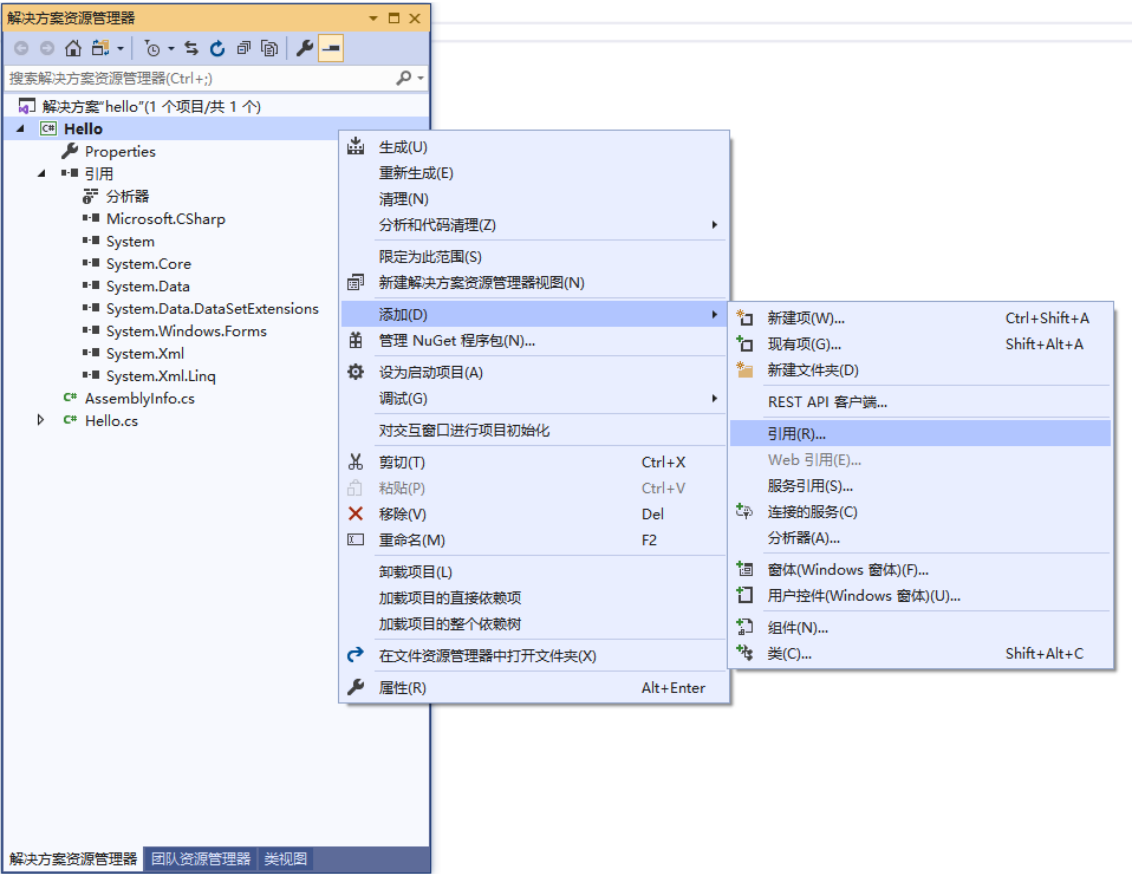


图 5.2

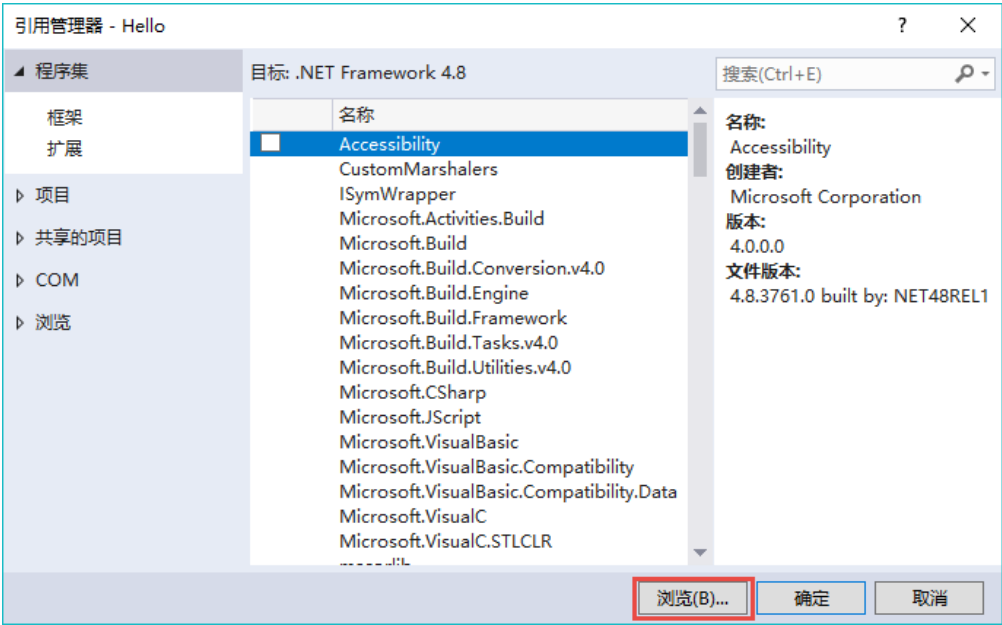


图 5.3

5.1.3 添加引用文件

找到浩辰 CAD 的 sdk 目录，例如在 c:\grxsdk\arx\inc 下选择文件 GcCoreMgd.dll、GcDbMgd.dll 和 GcMgd.dll，在 c:\grxsdk\arx\inc-x64 下选择文件 GrxCAD.Interop.dll (或 acax.tlb) 进行添加。如图 5.4、图 5.5 所示。

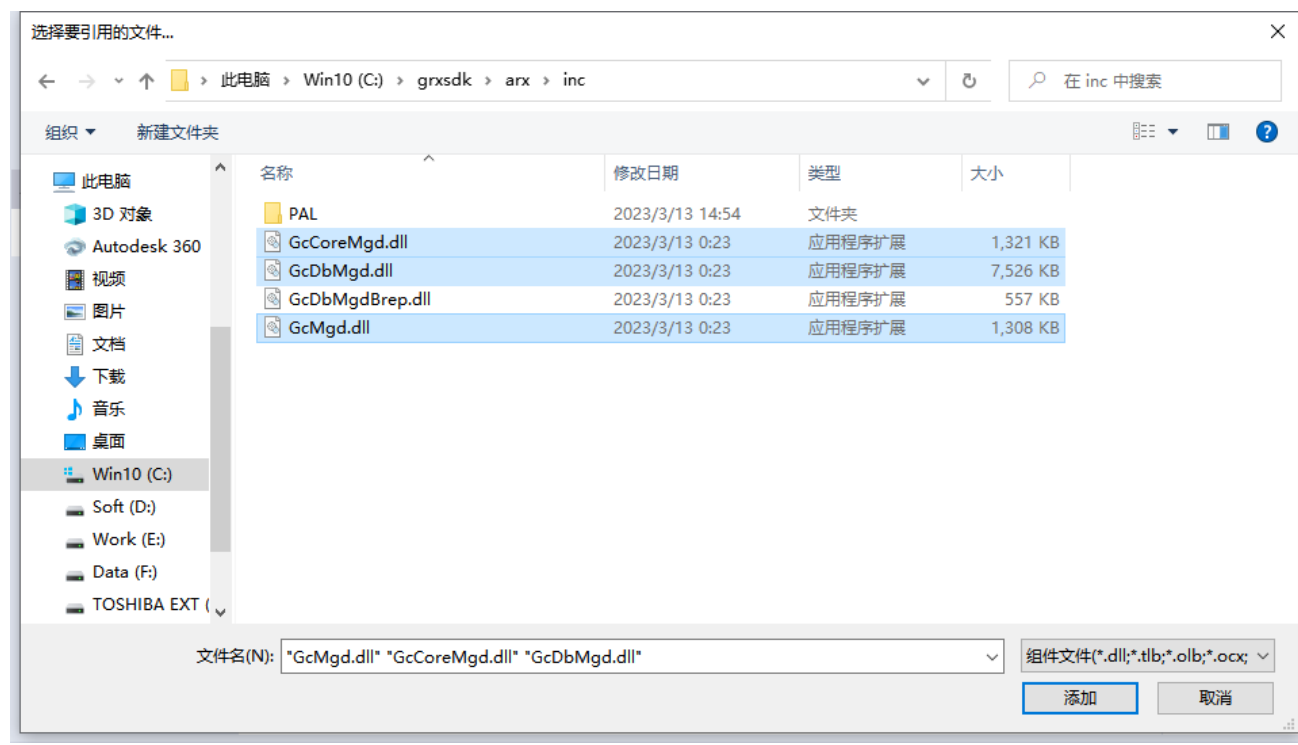


图 5.4

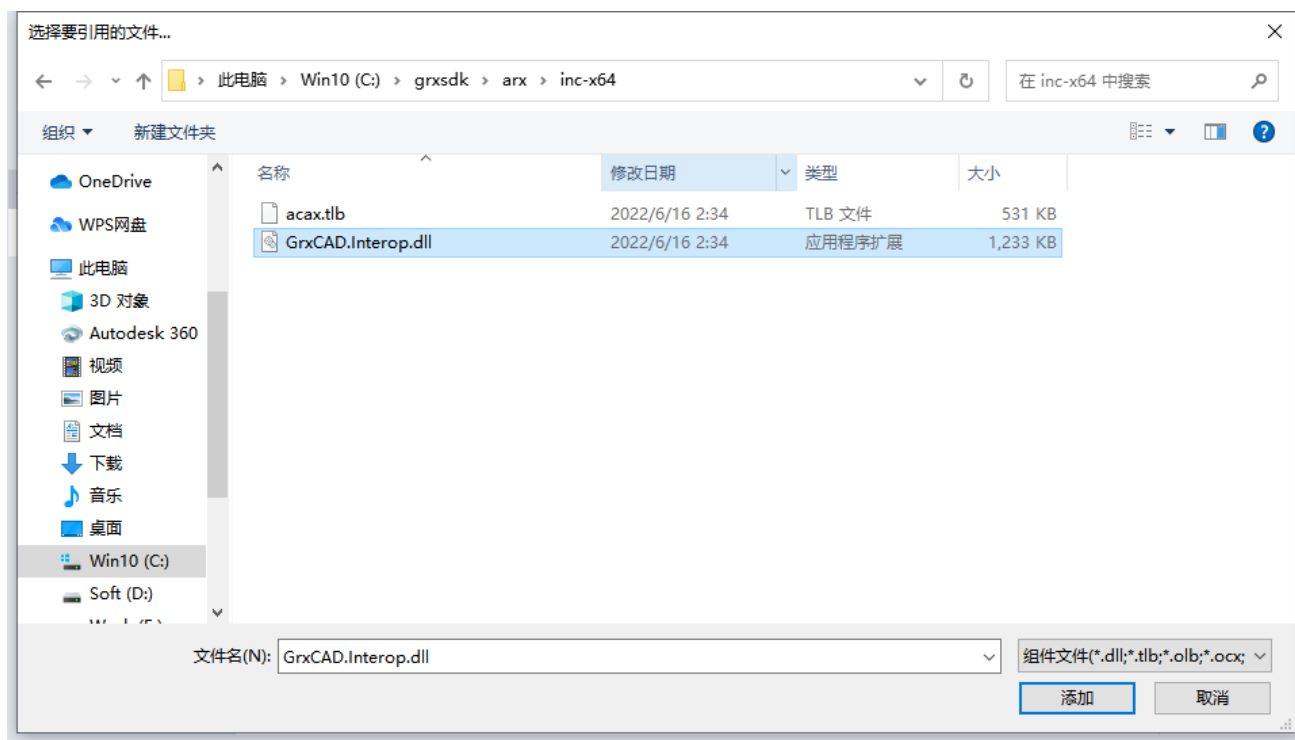


图 5.5

说明：GcCoreMgd.dll、GcDbMgd.dll 和 GcMgd.dll 必须添加，GrxCAD.Interop.dll (或 acax.tlb) 可以选择添加。

在添加完毕后，分别选中引用文件，点击右键，选择【属性】菜单项，将弹出的【属性】-【复制本地】中的 true 更改为 false，如图 5.6、图 5.7 所示，本处只给出 GcCoreMgd.dll 的修改样例，其他引用程序集的修改与 GcCoreMgd.dll 一致。

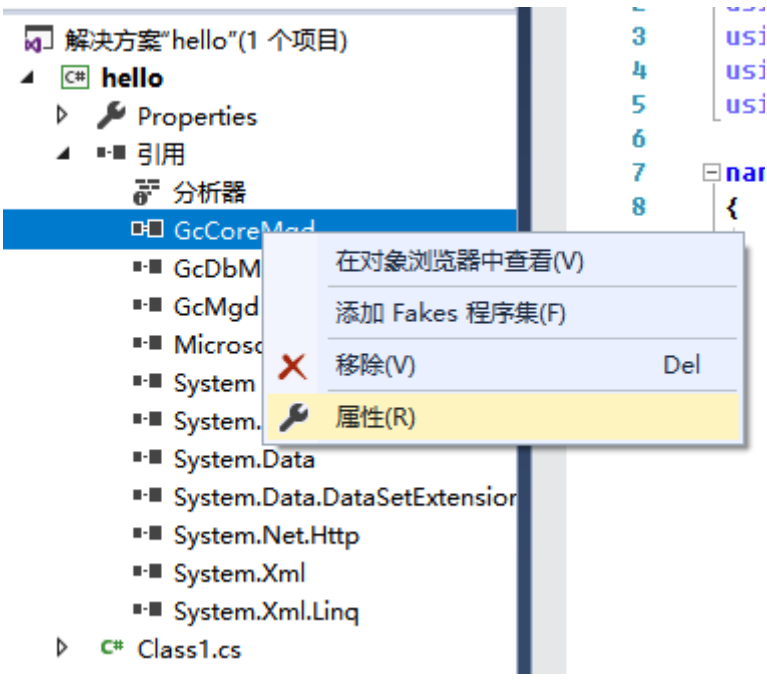


图 5.6

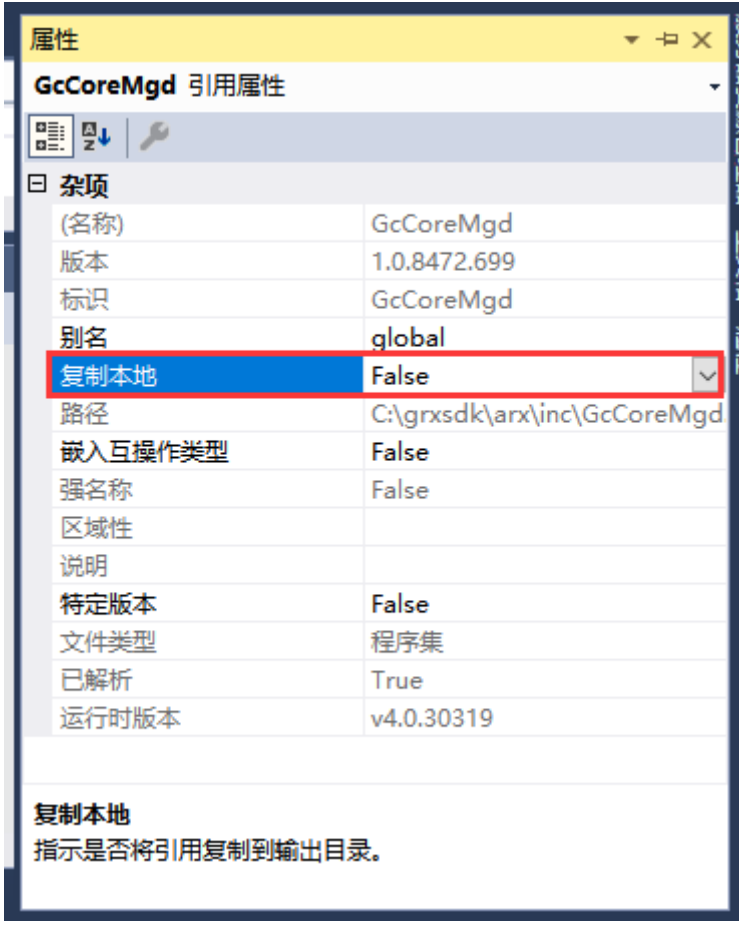


图 5.7

5.2 修改目标框架

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称，点击右键，点击【添加】->【属性】菜单项，在新页面的【应用程序】->【目标框架】中将目标框架修改为 Framework 4.8，如图 5.8、图 5.9 所示。

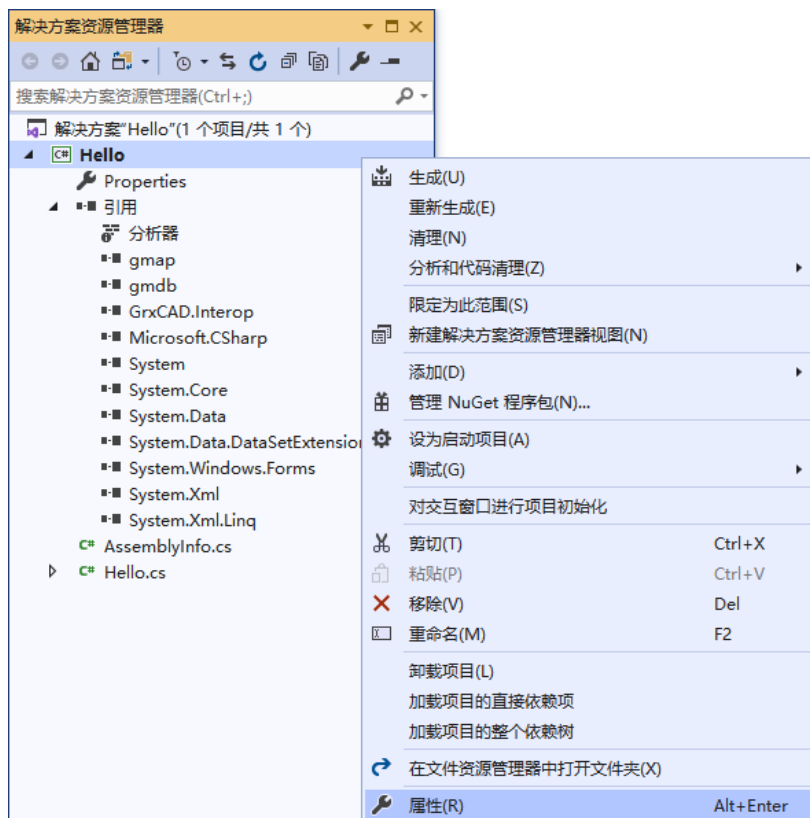


图 5.8

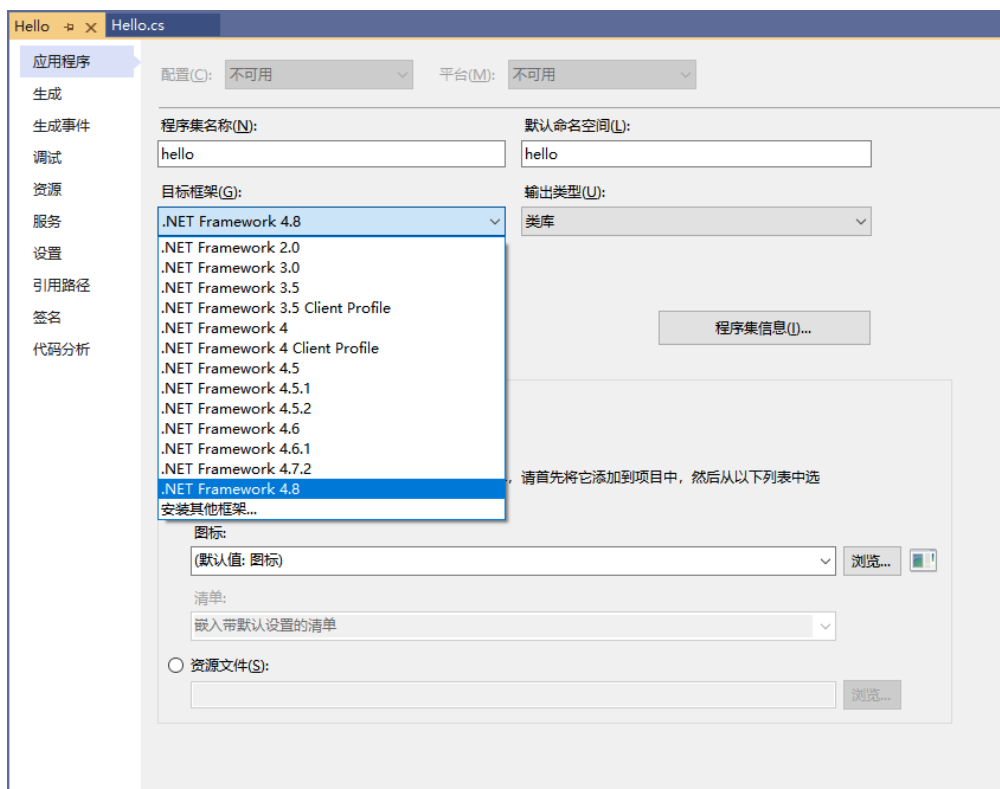


图 5.9

5.3 修改命名空间名称

将.NET 代码中对应的命名空间名称由“Autodesk.AutoCAD”替换为“Gssoft.Gscad”。

5.4 编译运行

点击 Visual Studio 2017 的菜单项【生成】->【生成解决方案】，在相应目录下重新生成 d11 文件，启动浩辰 CAD，在命令行输入 netload，将会出现“选择.NET 程序集”对话框，点击【打开】按钮，选择生成的 d11 文件即可加载移植后的应用程序。

6 VS2017 VB.NET 移植步骤

以下以移植“Hello”工程为例，说明移植过程，并假设浩辰 CAD 的 SDK 安装在 c:\grxsdk 目录下。

6.1 修改引用文件

6.1.1 移除引用文件

通过工程右键属性引用中移除 acad 的引用库文件，如图 6.1。

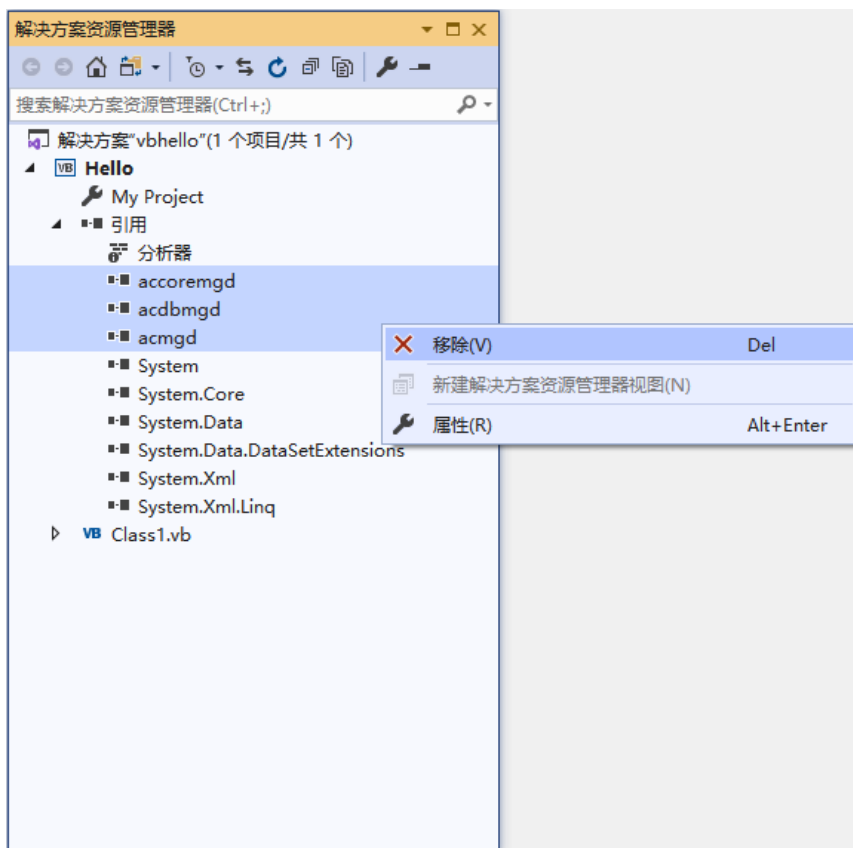


图 6.1

6.1.2 查询引用文件

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称，例如“Hello”，点右键，点【添加】->【引用】菜单项，会弹出“引用管理器”对话框，选择底部【浏览】，如图 6.2、图 6.3 所示。

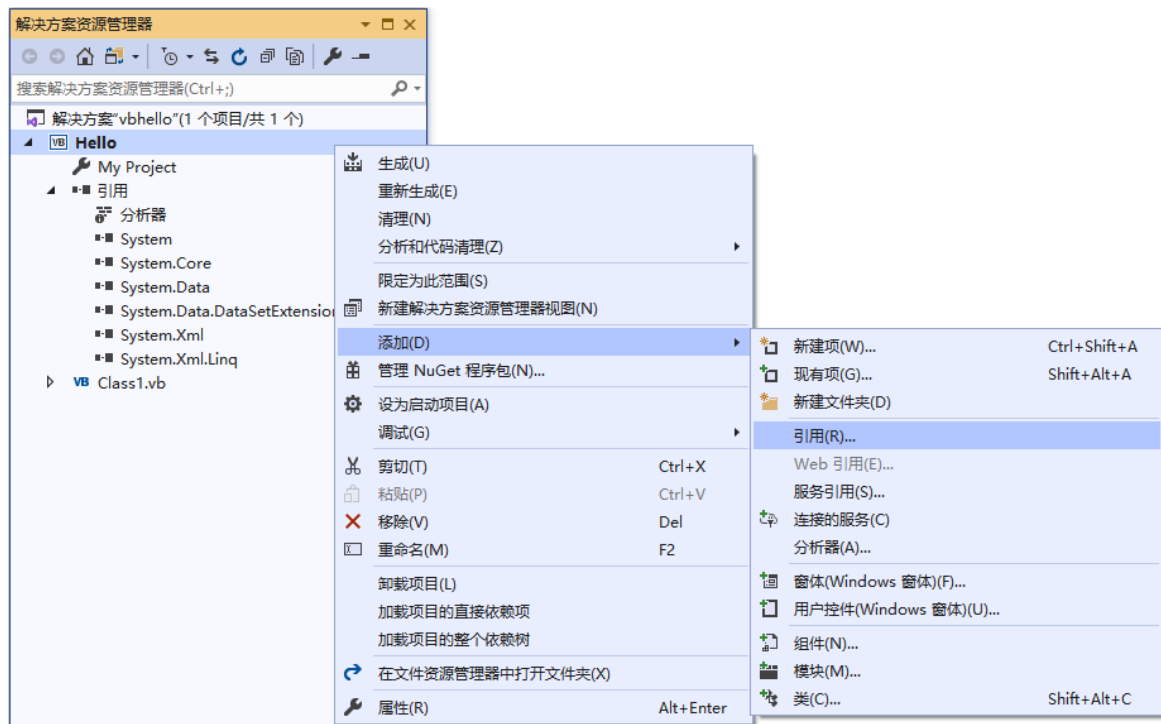


图 6.2

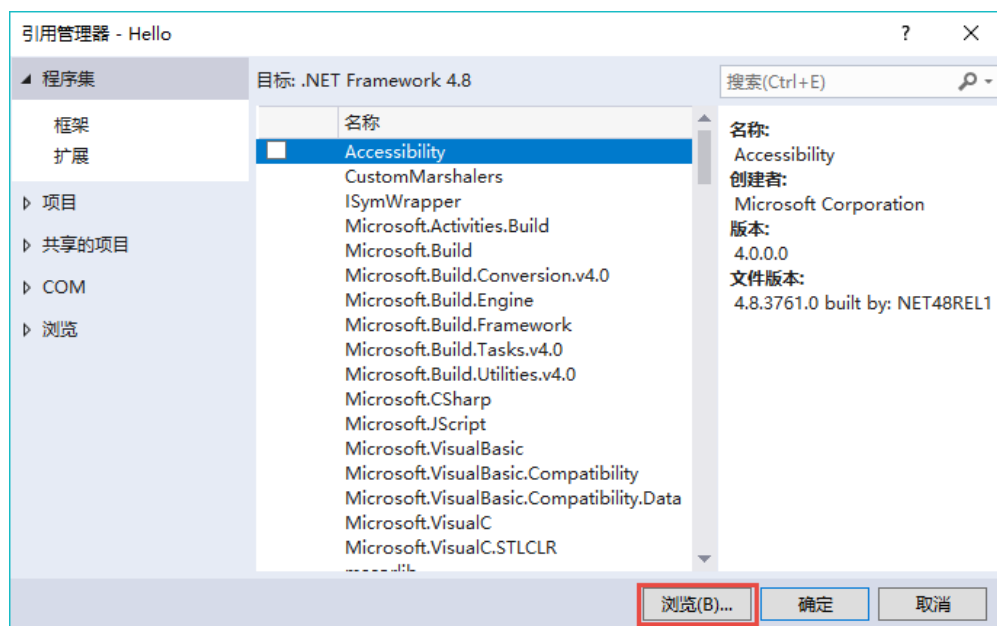


图 6.3

6.1.3 添加引用文件

找到浩辰 CAD 的 sdk 目录，例如在 c:\grxsdk\arx\inc 下选择文件 GcCoreMgd.dll、GcDbMgd.dll 和 GcMgd.dll，在 c:\grxsdk\arx\inc-x64 下选择文件 GrxCAD.Interop.dll (或 acax.tlb) 进行添加。如图 6.4、图 6.5 所示。

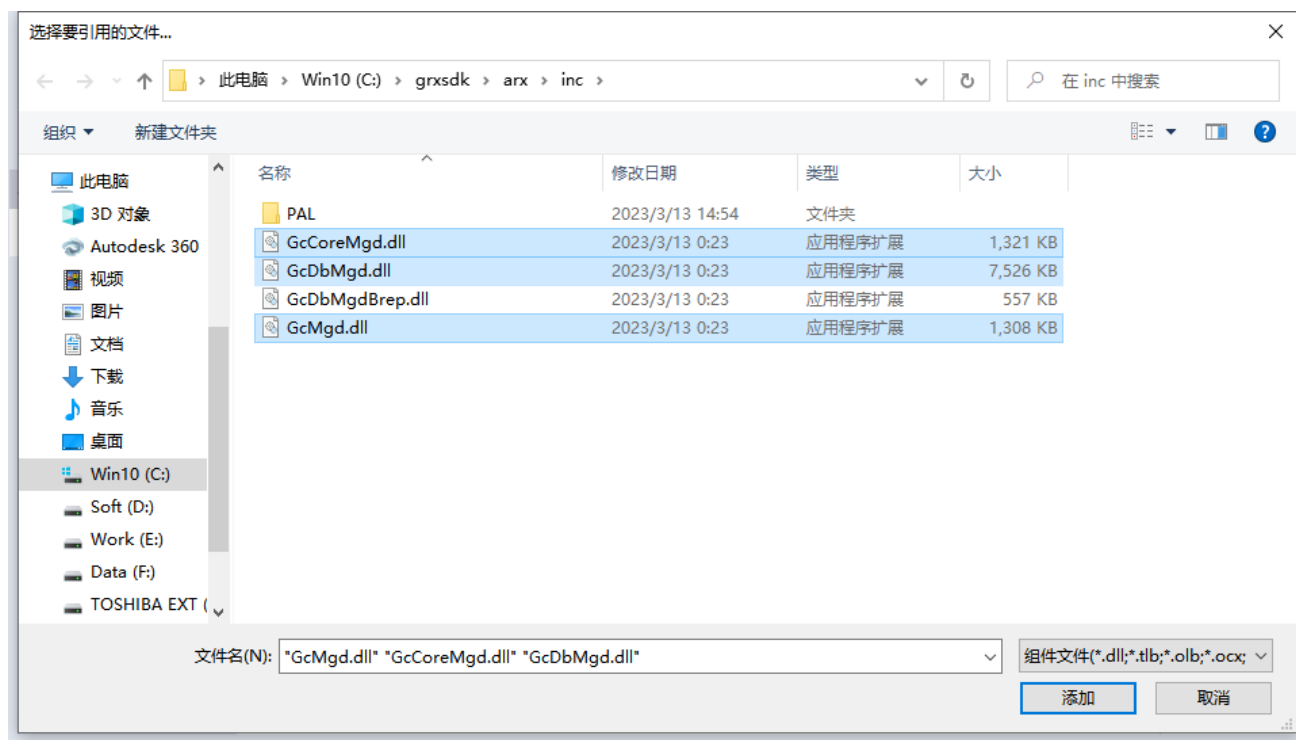


图 6.4

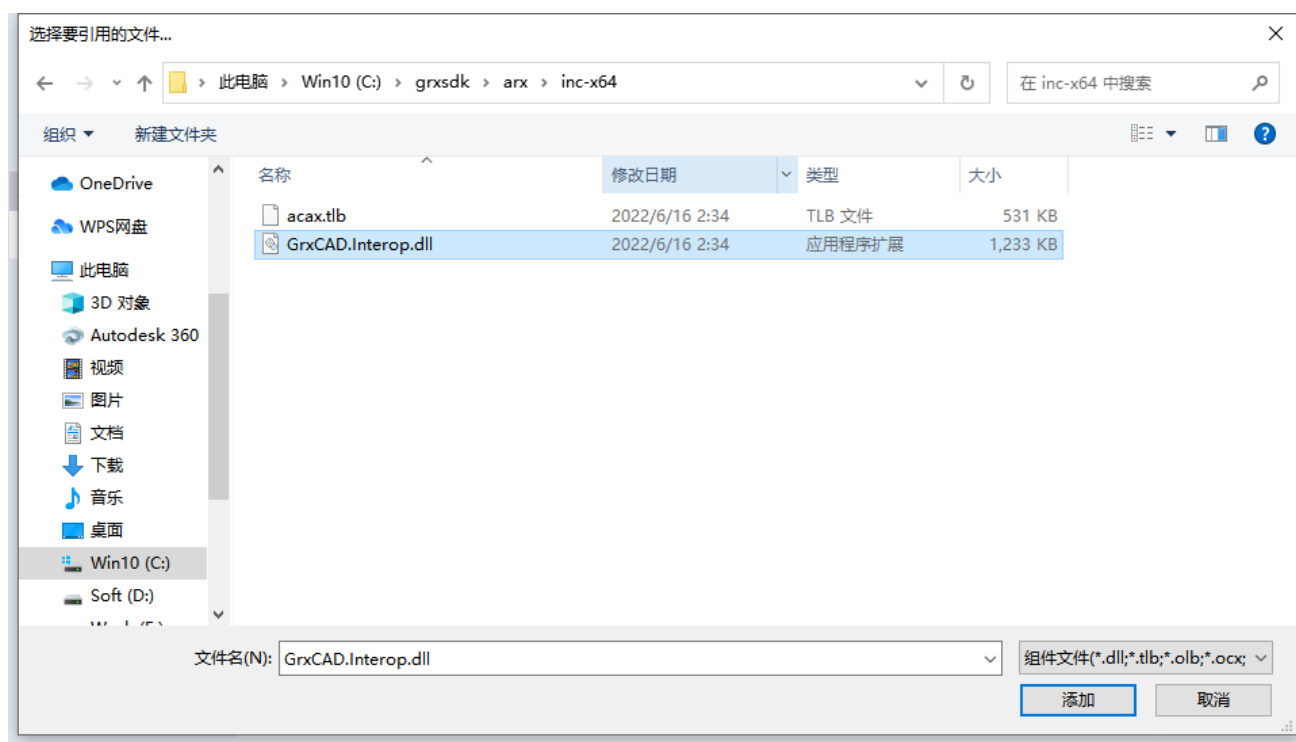


图 6.5

说明：GcCoreMgd.dll、GcDbMgd.dll 和 GcMgd.dll 必须添加，GrxCAD.Interop.dll（或

acax.tlb) 可以选择添加。

在添加完毕后，分别选中引用文件，点右键，选择【属性】菜单项，将弹出的【属性】-【复制本地】中的 true 更改为 false，如图 6.6、图 6.7 所示，本处只给出 GcCoreMgd.dll 的修改样例，其他的引用程序集的修改与 GcCoreMgd.dll 一致。

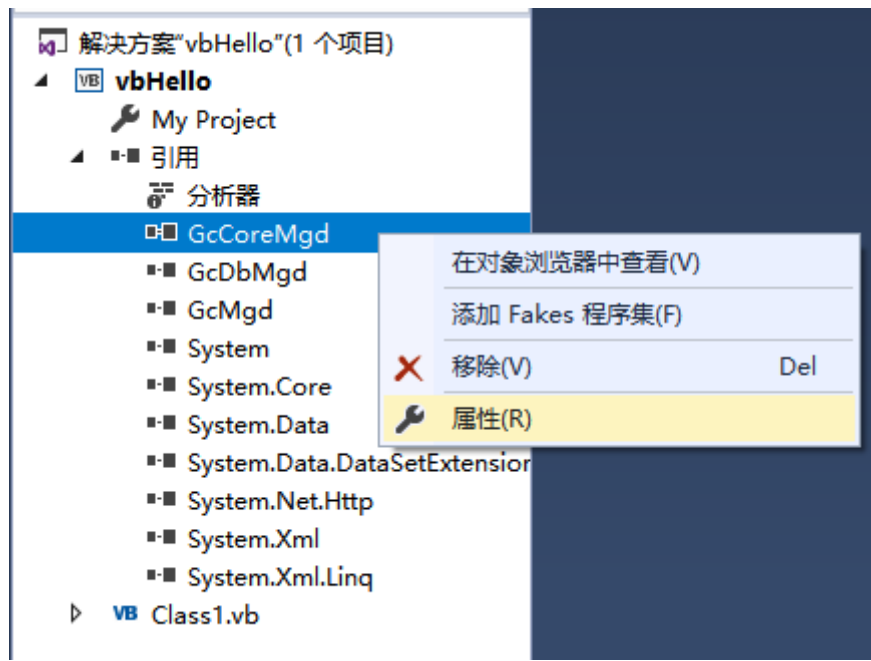


图 6.6

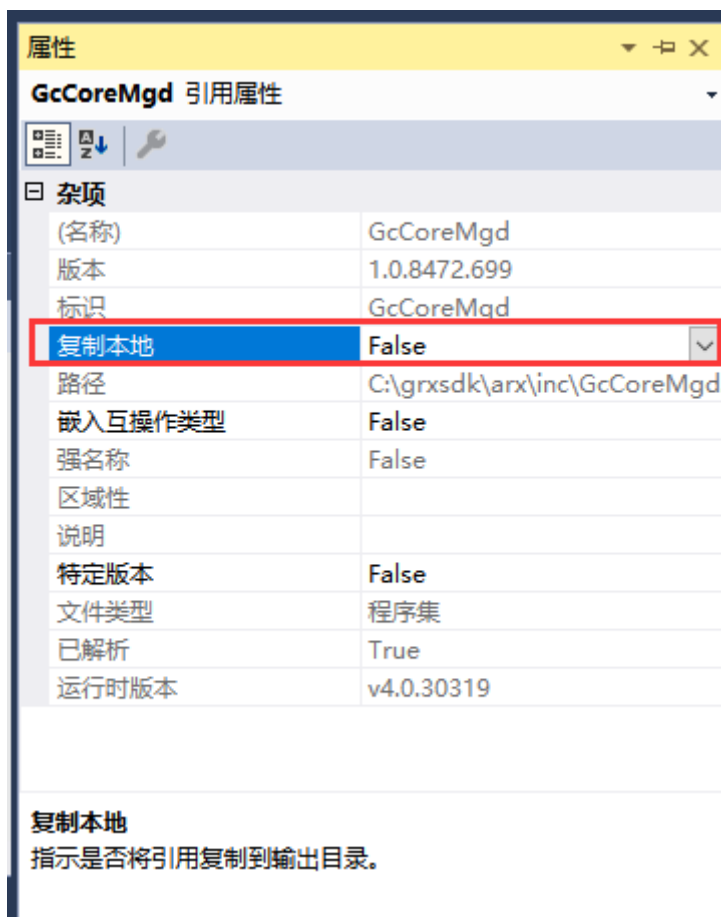


图 6.7

6.2 修改目标框架

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称，点右键，点【添加】->【属性】菜单项，在新页面的【应用程序】->【目标框架】中将目标框架修改为 Framework 4.8，如图 6.8、图 6.9 所示。

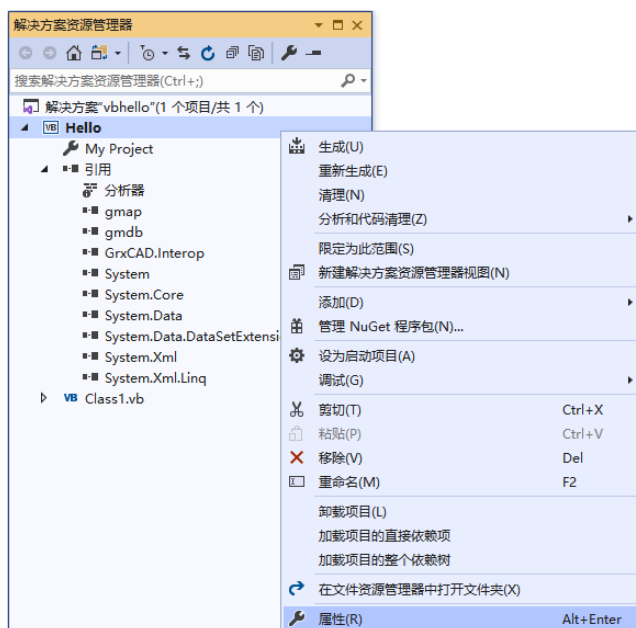


图 6.8

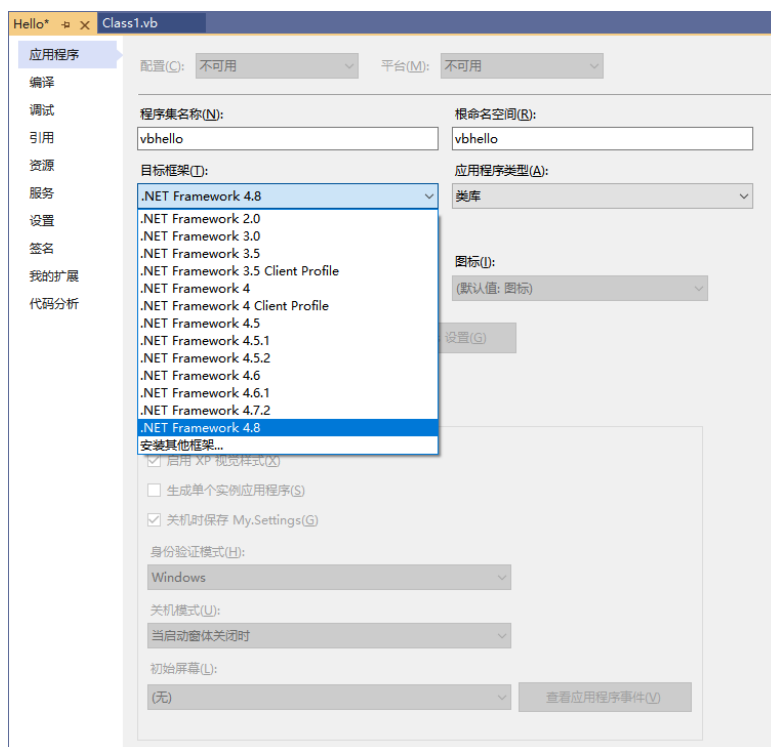


图 6.9

6.3 修改命名空间名称

将 .NET 代码中对应的命名空间名称 “Autodesk.AutoCAD” 直接替换为 “Gssoft.Gscad” 即可。

6.4 编译运行

点击 Visual Studio 2017 的菜单项【生成】->【生成解决方案】，在相应目录下重新生成 dll 文件，启动浩辰 CAD，在命令行输入 netload，将会出现“选择.NET 程序集”对话框，点击【打开】按钮，选择生成的 dll 文件即可加载移植后的应用程序。

7 DotNet 二次开发的特殊应用描述

DotNet 二次开发区别于 ACAD 的特殊用法，特殊应用描述如下。

7.1 using 使用

类似 `var mapping = new IdMapping();` 这种代码都需要使用 `using`，如：

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

7.2 ResultBuffer 使用

使用 “`ResultBuffer.ResbufObject`” 替代 “`ResultBuffer.UnmanagedObject`”；

`ResultBuffer.UnmanagedObject` 是 `GcResbuf` 类型；

`ResultBuffer.ResbufObject` 是 `resbuf` 类型；

使用 “`ResultBuffer.Create(IntPtr, bool)`” 替代 “`ResultBuffer(IntPtr, bool)`”。

7.3 导入非托管 arx 函数的替换方法

导入非托管 arx 函数 `acedCmd`, `acedCommand` 的替换方法：

使用 `Gssoft.Gscad.EditorInput.Editor.Command(params object[])` 代替。

7.4 模式对话框点击按钮后在当前图选择的处理方法

模式对话框点击按钮后在当前图选择点或实体，要使用

`using (var edUI = doc.Editor.StartUserInteraction(this))`，如下代码示例：

```
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("\n 选择点: ");
    PromptPointResult ppr = ed.GetPoint(ppo);

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

8 其他事项

8.1 VB 环境下获取 GCAD Application 对象

在此处需要添加引用 acax.tlb(路径为 C:\grxsdk\arx\inc-x86\acax.tlb, 如使用 64 位浩辰 CAD 请选择 inc-x64 下对应的文件), 然后使用如下代码可获取 GCAD Application 对象。

```
Imports Gssoft.Gscad.Runtime
Imports GcadVbaLib

Public Class Class1
    <CommandMethod("GetGcadApplication")>
        Public Sub GetGcadApplication()

            ' 定义获取 GcadApplication 的对象 此时对象为 null
            Dim GCadApp As GcadApplication

            ' 通过进程名获取到相应的对象地址
            GCadApp = GetObject(, "gcad.application")
            ' 将获取到的进程窗体最小化
            GCadApp.WindowState = GcWindowState.acMin

            End Sub
        End Class
```

8.2 C#.NET 与 VB.NET 命名空间变更

➤ VB.NET:

Imports 后的内容, 将 “Autodesk.AutoCAD” 改为 “Gssoft.Gscad”。

➤ C#.NET:

using 后的内容, 将 “Autodesk.AutoCAD” 改为 “Gssoft.Gscad”

➤ 如果使用 COM:

在 VB.net 下, 添加 Imports GcadVbaLib

在 C#.net 下, 添加 using GcadVbaLib

Autodesk.AutoCAD.Interop.Common 去掉

代码中一些 COM 对象的名称, 如 “AcadLWPolyline”, 需要将其前缀改为 “Gcad”, 比如改为 “GcadLWPolyline”。

一些不能确定的, 可以先在对象浏览器中查一下。比如 “ACAD_COLOR”, 可先在对象浏览器中查一下是否有 “GCAD_COLOR”, 如果查到了就应替换。

9 替换样例

9.1 命名空间替换样例

AutoCAD	浩辰 CAD
Autodesk.AutoCAD.ApplicationServices	Gssoft.Gscad.ApplicationServices
Autodesk.AutoCAD.EditorInput	Gssoft.Gscad.EditorInput
Autodesk.AutoCAD.GraphicsSystem	Gssoft.Gscad.GraphicsSystem
Autodesk.AutoCAD.PlottingServices	Gssoft.Gscad.PlottingServices
Autodesk.AutoCAD.Publishing	Gssoft.Gscad.Publishing
Autodesk.AutoCAD.Runtime	Gssoft.Gscad.Runtime
Autodesk.AutoCAD.Windows	Gssoft.Gscad.Windows
Autodesk.AutoCAD.Colors	Gssoft.Gscad.Colors
Autodesk.AutoCAD.DatabaseServices	Gssoft.Gscad.DatabaseServices
Autodesk.AutoCAD.DatabaseServices.Filters	Gssoft.Gscad.DatabaseServices.Filters
Autodesk.AutoCAD.Geometry	Gssoft.Gscad.Geometry
Autodesk.AutoCAD.GraphicsInterface	Gssoft.Gscad.GraphicsInterface
Autodesk.AutoCAD.LayerManager	Gssoft.Gscad.LayerManager

9.2 类名及属性名替换样例

AutoCAD	浩辰 CAD
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

10 版权声明

版权所有：苏州浩辰软件股份有限公司

使用许可：允许复制、引用本文档的任何部分。未经许可，不得更改本文档的任何部分。
在复制、引用时，请务必保留本声明，否则将追究法律责任。