

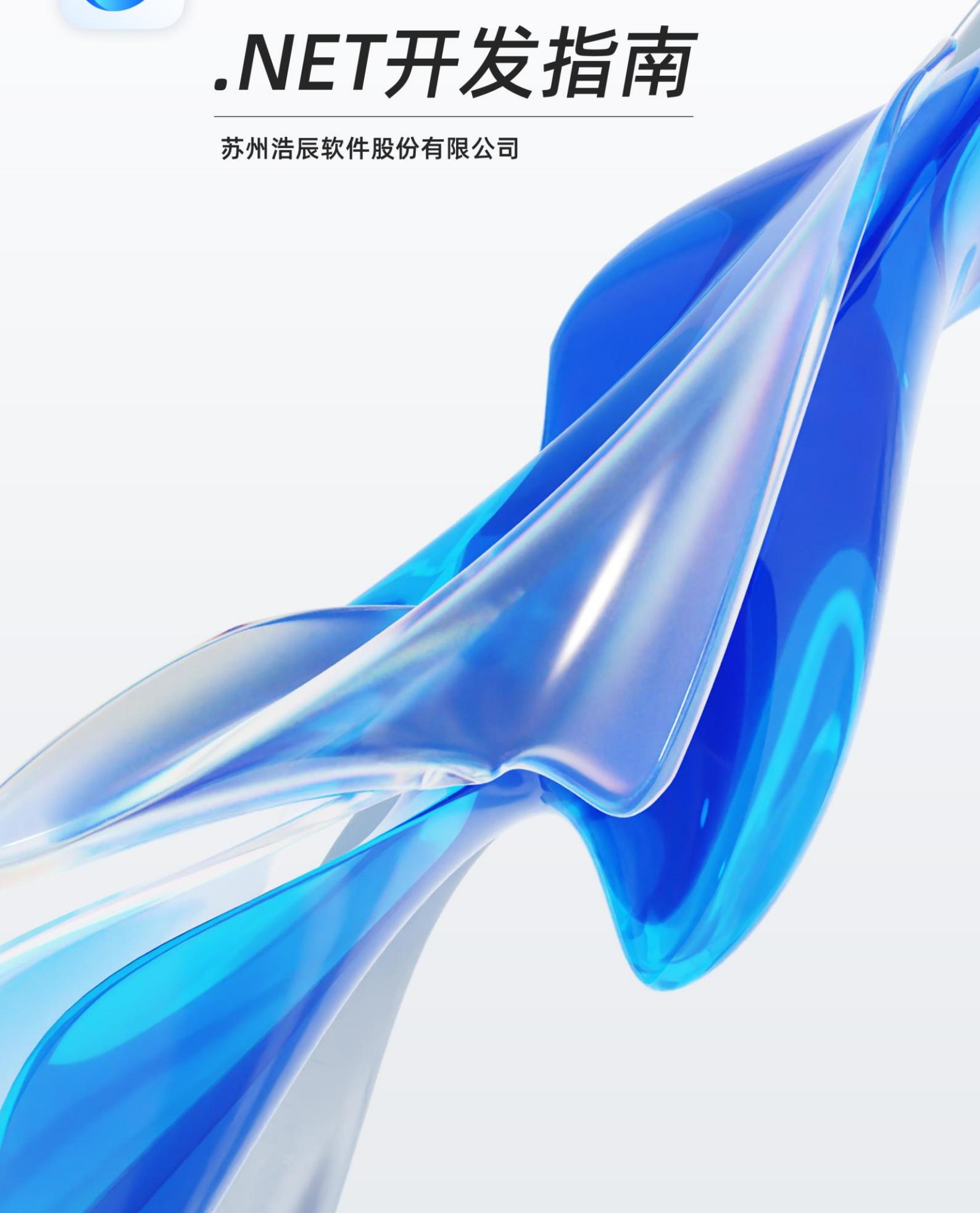


浩辰CAD 2025

# .NET开发指南

---

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 安装.NET SDK.....	4
4 VS2017 C#.NET 开发样例.....	5
4.1 创建一个.NET 项目 .....	5
4.2 设置引用库.....	6
4.3 添加代码.....	10
4.4 编译程序.....	11
4.5 运行程序.....	11
5 VS2017 VB.NET 开发样例.....	14
5.1 创建一个.NET 项目 .....	14
5.2 设置引用库.....	15
5.3 添加代码.....	19
5.4 编译程序.....	20
5.5 运行程序.....	20
6 DotNet 二次开发的特殊应用描述.....	22
6.1 using 使用 .....	22
6.2 ResultBuffer 使用 .....	22
6.3 导入非托管 arx 函数的替换方法.....	22
6.4 模式对话框点击按钮后在当前图选择的处理方法.....	22
6.5 C#.NET 与 VB.NET 命名空间变更 .....	23
7 版权声明.....	24

## 1 简介

浩辰 CAD.NET API 提供了一系列托管的外包类(Managed Wrapper Class)，提供了以.NET 为基础，面向对象的开发环境及应用程序接口。开发人员可在 Microsoft .NET Framework 下，使用支持.NET 的语言，如 VB.NET，C#等对浩辰 CAD 进行二次开发，其优点是完全面向对象，方便易用。

## 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

Windows11

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 安装. NET 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。

## 4 VS2017 C#.NET 开发样例

以下以创建“Hello”工程为例，说明创建过程，并假设浩辰 CAD 的 SDK 安装在 c:\grxsdk 目录下。

### 4.1 创建一个.NET 项目

#### 4.1.1 运行 Microsoft Visual Studio 2017

选择【文件】菜单，再选择【新建】子菜单，最后选择【新建】菜单的子菜单【项目】。点击【项目】菜单项，会弹出“新建项目”对话框中，在左边的【已安装】树中，选择【Visual C#】，右边列表中选择【类库 (.NET Framework)】。

#### 4.1.2 输入项目的存放路径和项目名称

在“新建项目”对话框中“名称”标签后填入“Hello”，并选择.NET Framework4.8，如图 4.1 所示。

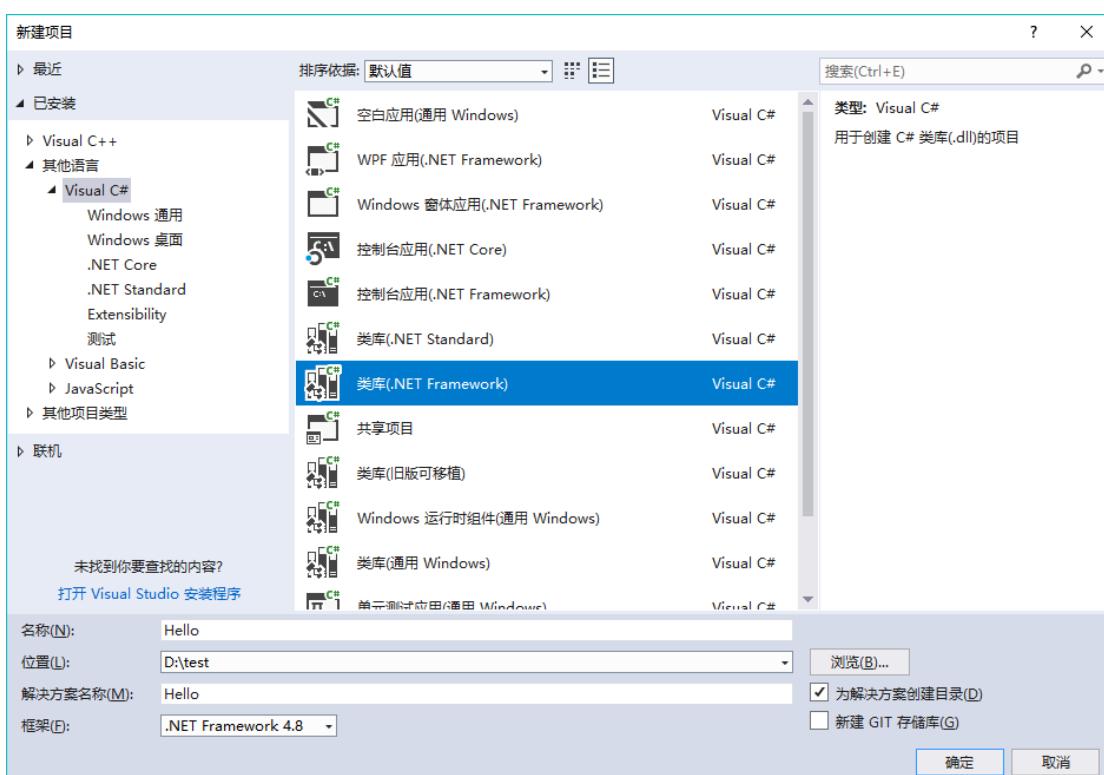


图 4.1

注意：此处应选择 framework 4.8

#### 4.1.3 完成项目创建

进行完第二步操作后，单击对话框中的【确定】，完成了一个新项目的创建。

### 4.2 设置引用库

#### 4.2.1 添加引用

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称“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 4.2、图 4.3 所示。

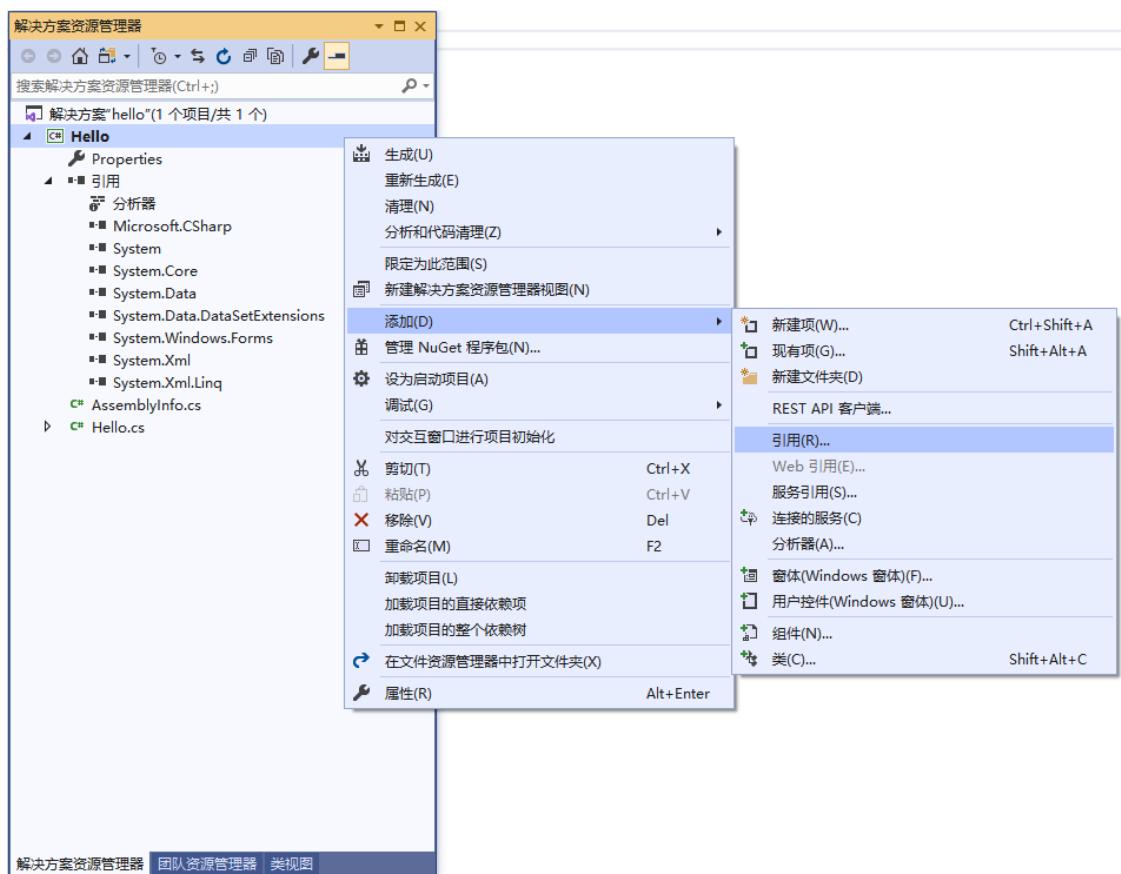


图 4.2

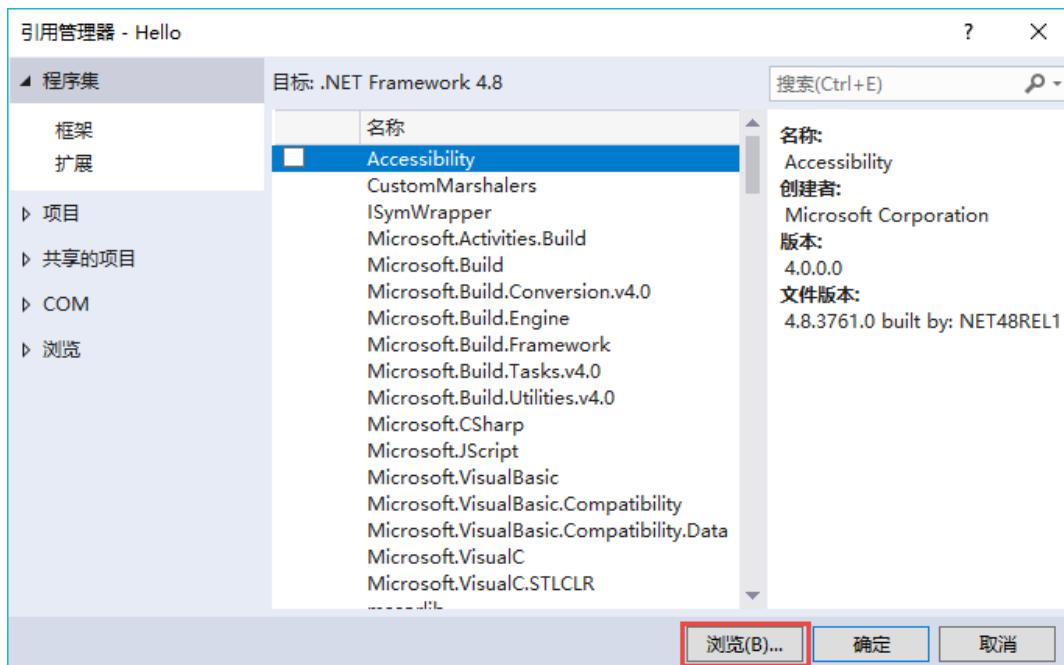


图 4.3

#### 4.2.2 选择并添加文件

找到浩辰 CAD 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 4.4、图 4.5 所示。

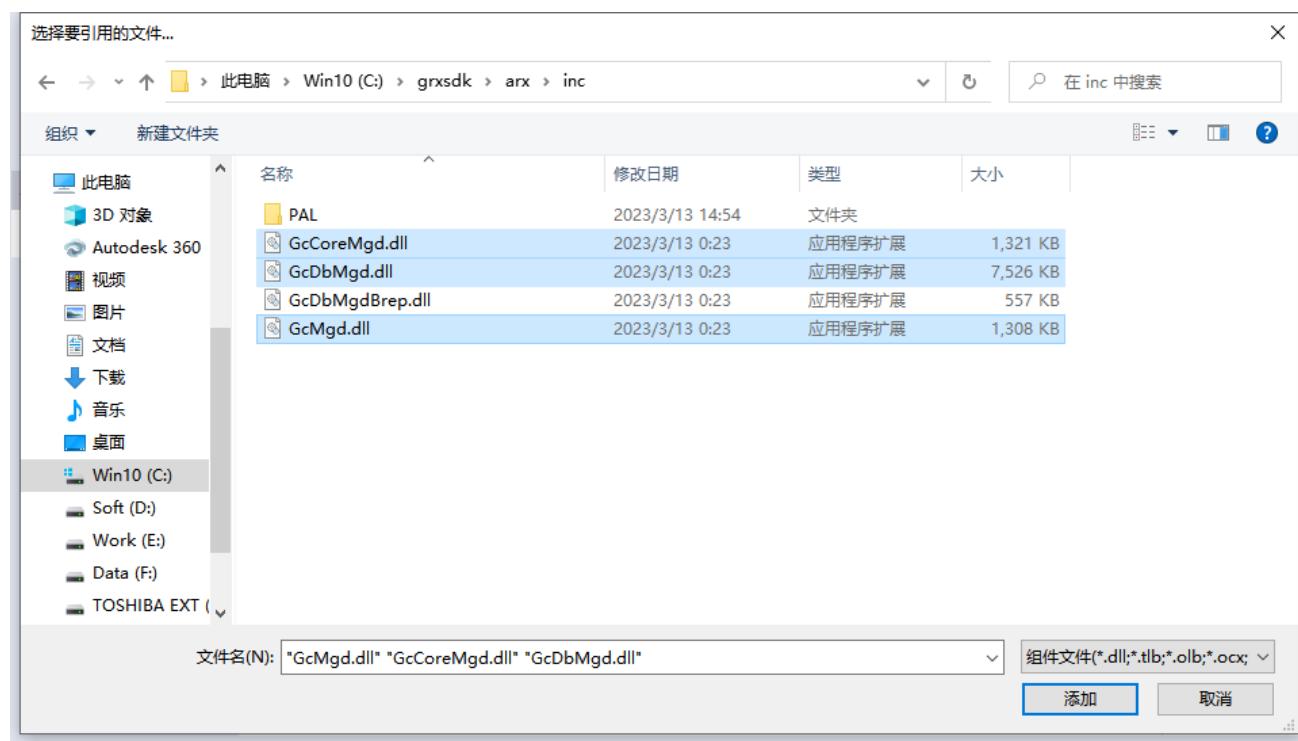


图 4.4

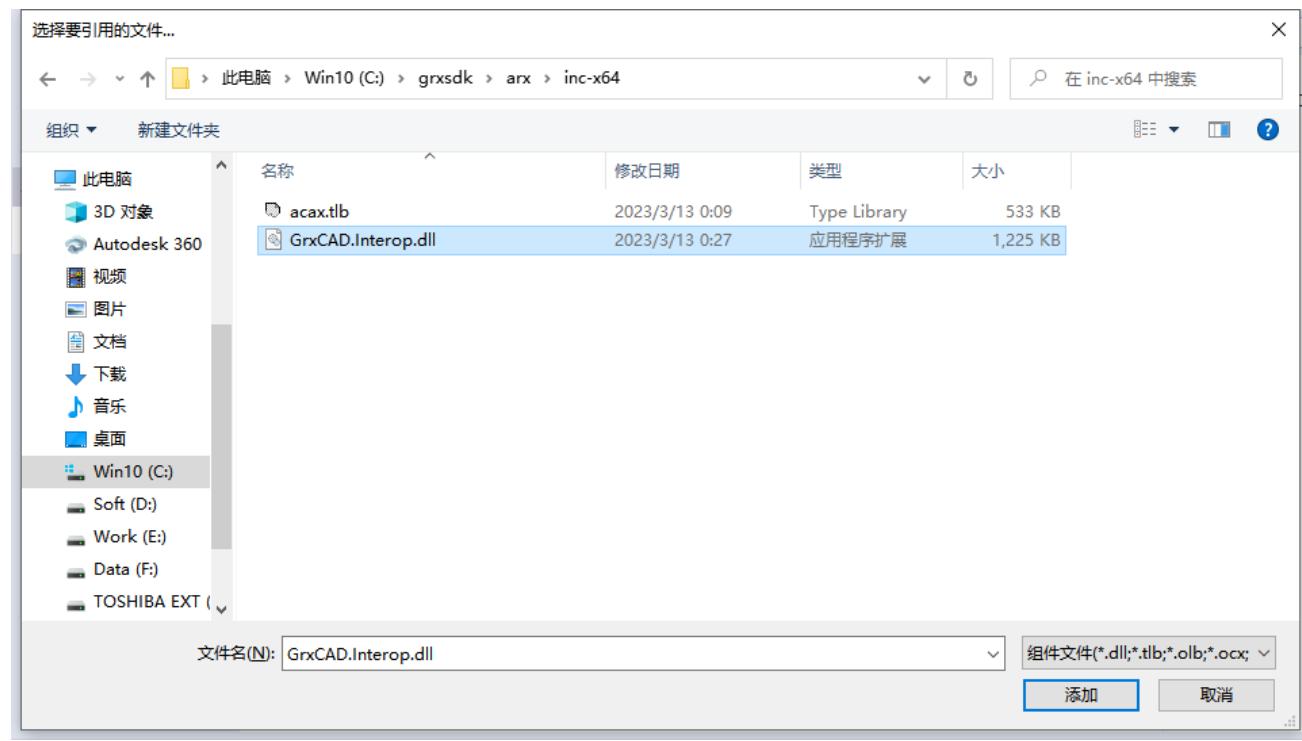


图 4.5

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 4.6、图 4.7 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。

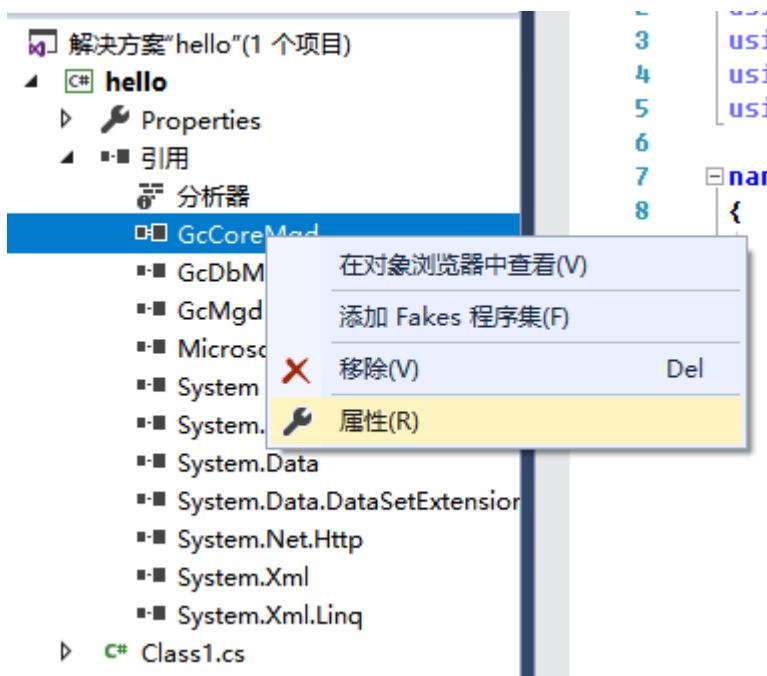


图 4.6

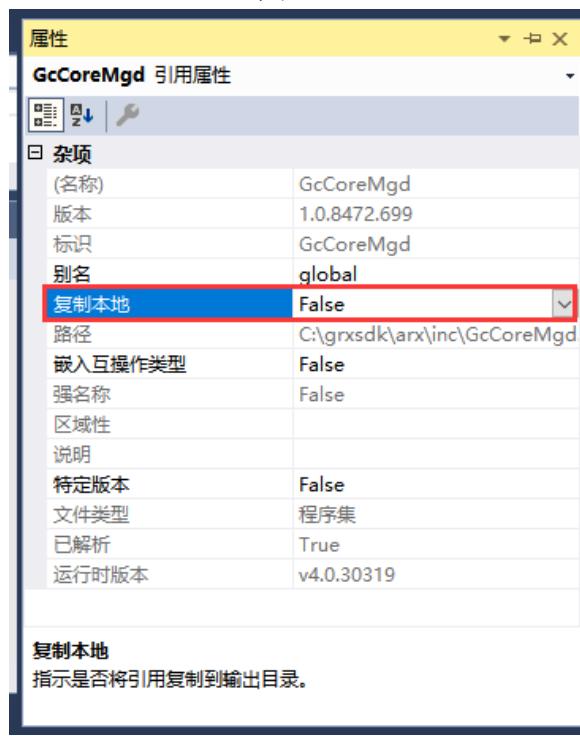


图 4.7

- GcCoreMgd.dll 包括命名空间为:  
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 包括命名空间为:

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 包括命名空间为:

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 表示 com 的对应的.NET 接口。

### 4.3 添加代码

在 Hello.cs 里，拷入如下代码:

```
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();
            }
        }
    }
}
```

## 4.4 编译程序

点击 Visual Studio 2017 的菜单项【生成】->【生成解决方案】，会在 F:\test\Hello\Hello\bin\Release（目录可设定）目录下生成 Hello.dll。

## 4.5 运行程序

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 Hello.dll，加载后如图 4.8 所示。

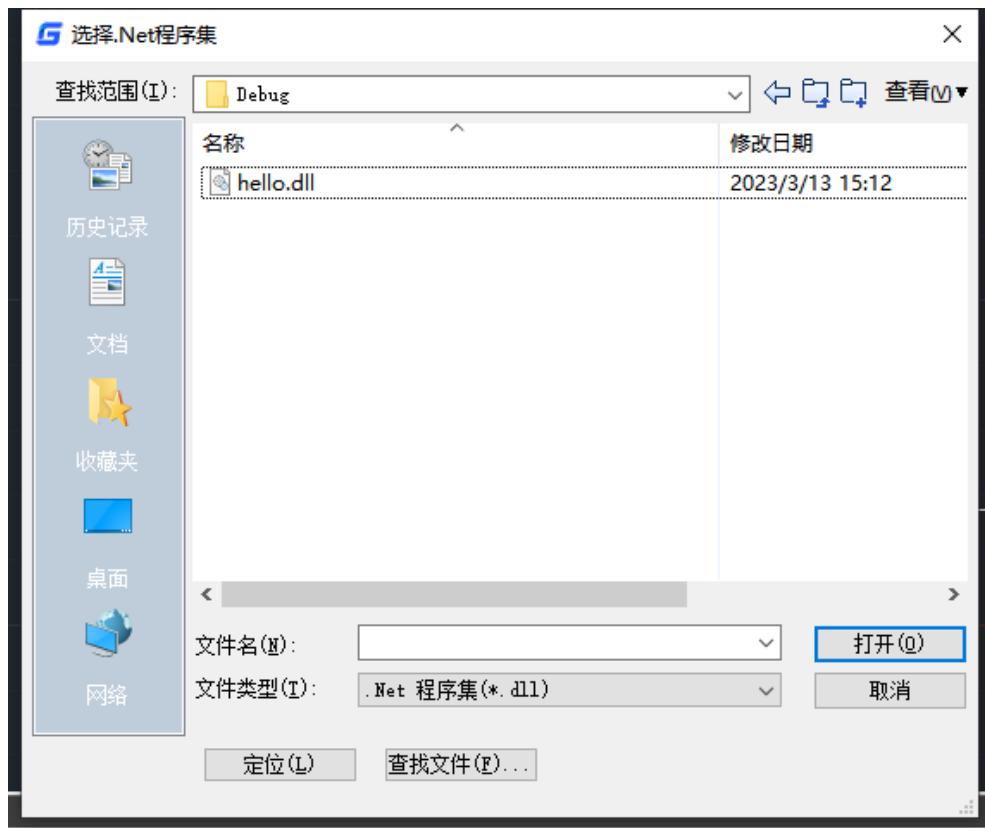


图 4.8

关闭加载程序对话框，在浩辰 CAD 命令行输入“hello”命令。将在命令行上打印输出：“Hello dotnet”。运行效果如图 4.9 所示。

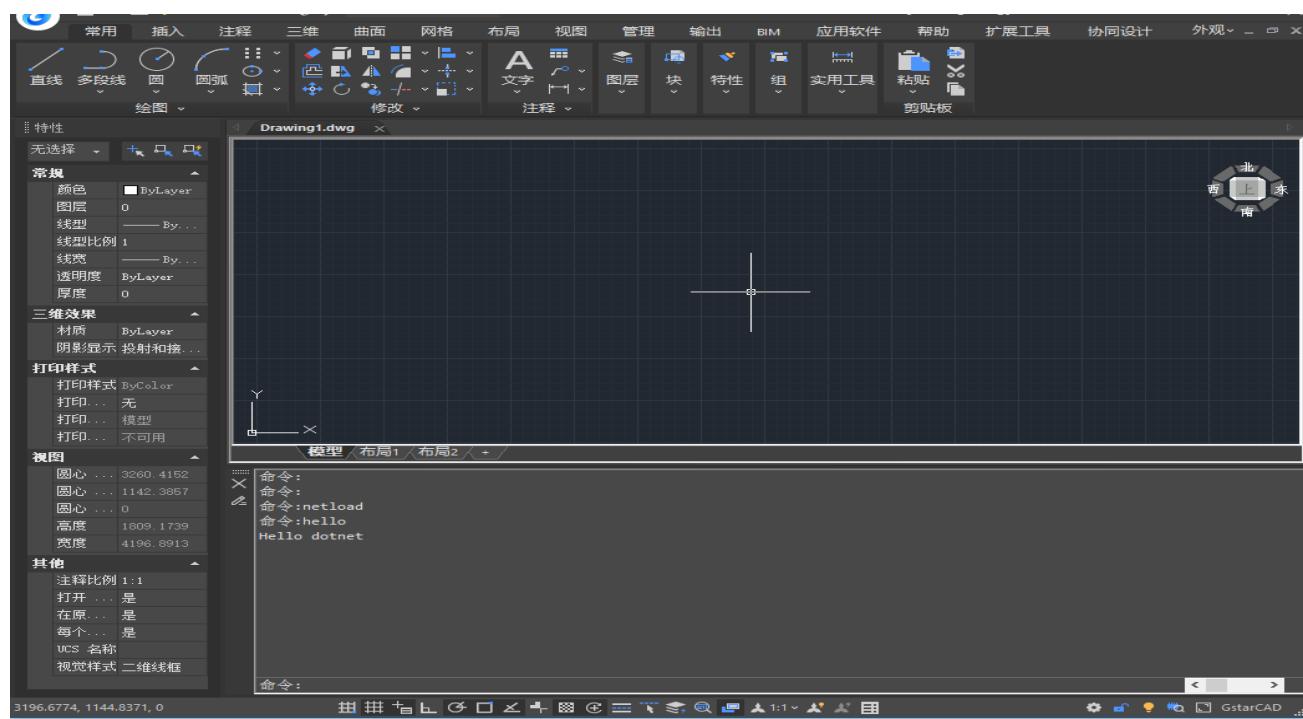


图 4.9



## 5 VS2017 VB.NET 开发样例

以下以创建“Hello”工程为例，说明创建过程，并假设浩辰 CAD 的 SDK 安装在 c:\grxsdk 目录下。

### 5.1 创建一个.NET 项目

#### 5.1.1 运行 Microsoft Visual Studio 2017

选择【文件】菜单，再选择【新建】子菜单，最后选择【新建】菜单的子菜单【项目】。点击【项目】菜单项，会弹出“新建项目”对话框中，在左边的【已安装】树中，选择【Visual Basic】，右边列表中选择【类库 (.NET Framework)】。

#### 5.1.2 输入项目的存放路径和项目名称

在“新建项目”对话框中“名称”标签后填入“vbHello”。选择.NET Framework4.8，如图 5.1 所示。

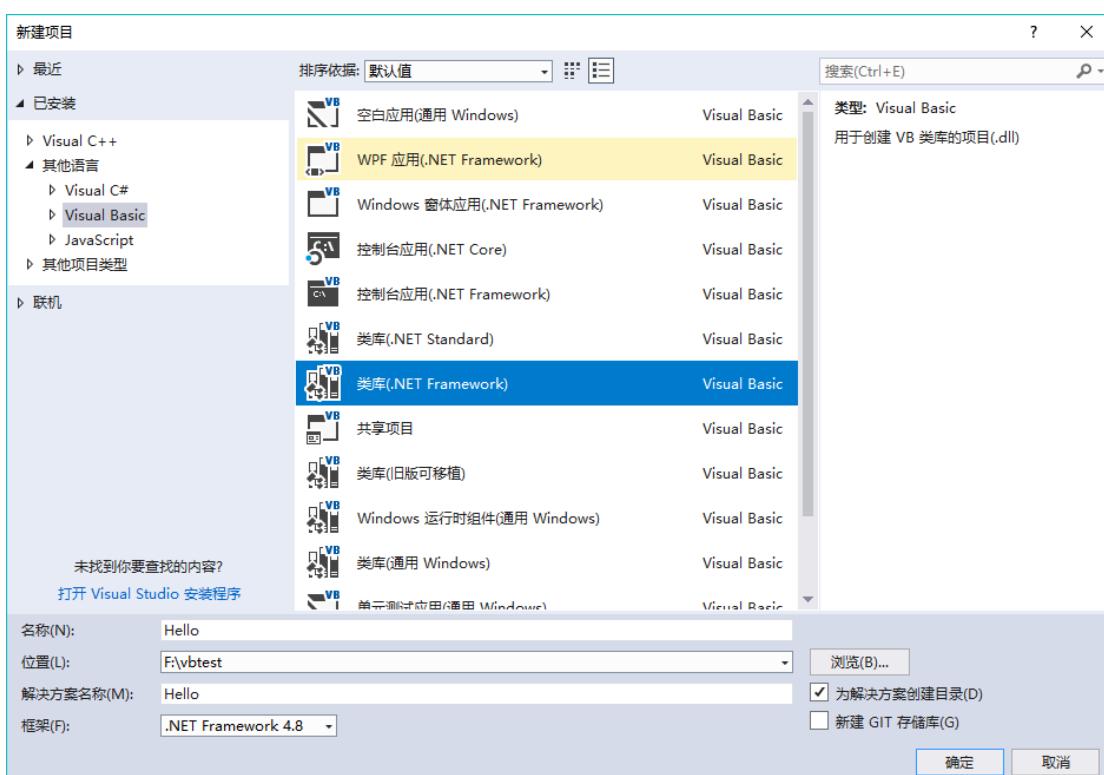


图 5.1

**注意：此处应选择 framework4.8**

### 5.1.3 完成项目创建

进行完第二步操作后，单击对话框中的【确定】按钮，完成了一个新项目的创建。

## 5.2 设置引用库

### 5.2.1 添加引用

在 Visual Studio 2017 的“解决方案资源管理器”中，选中项目名称“vbHello”，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。

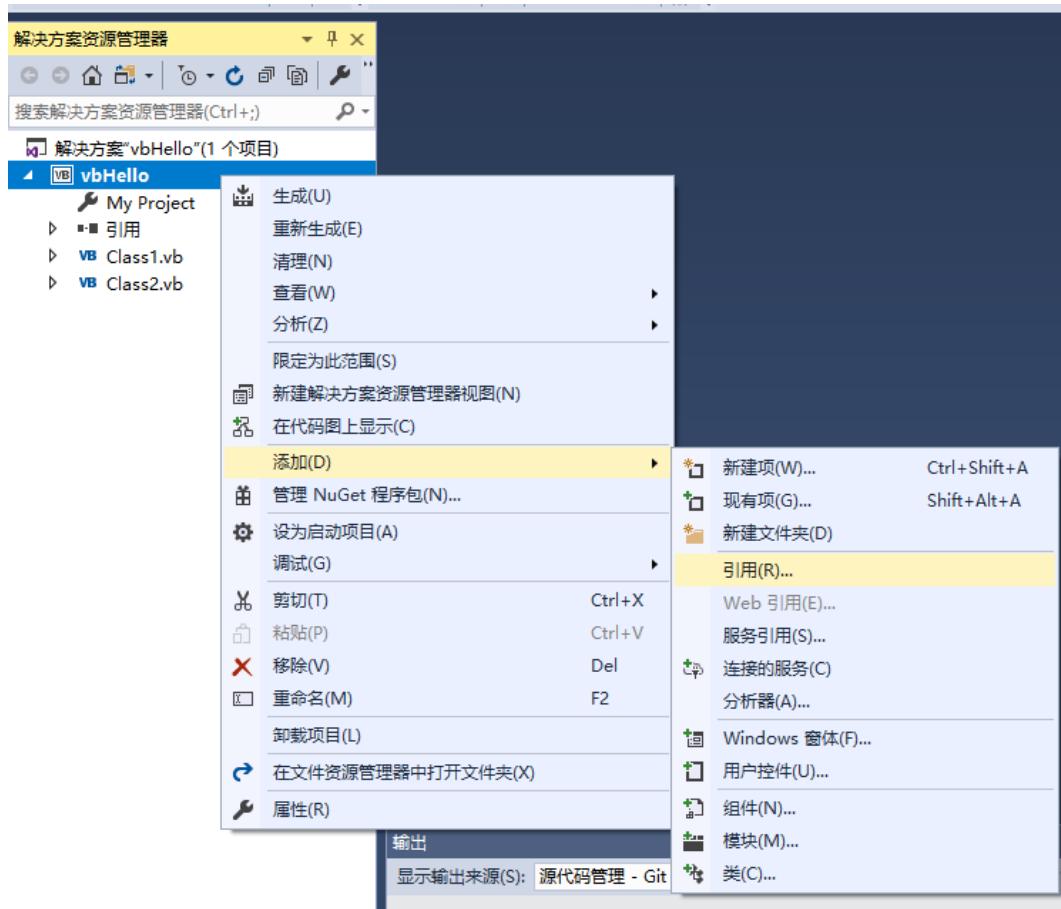


图 5.2

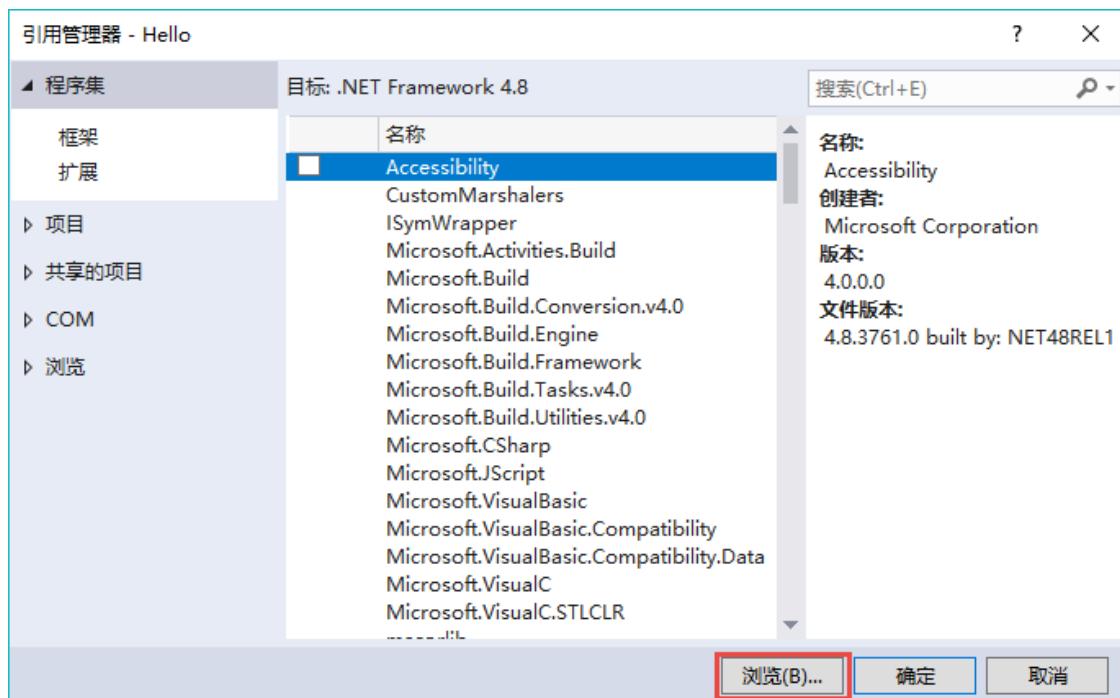


图 5.3

### 5.2.2 选择并添加文件

找到浩辰 CADsdk 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。

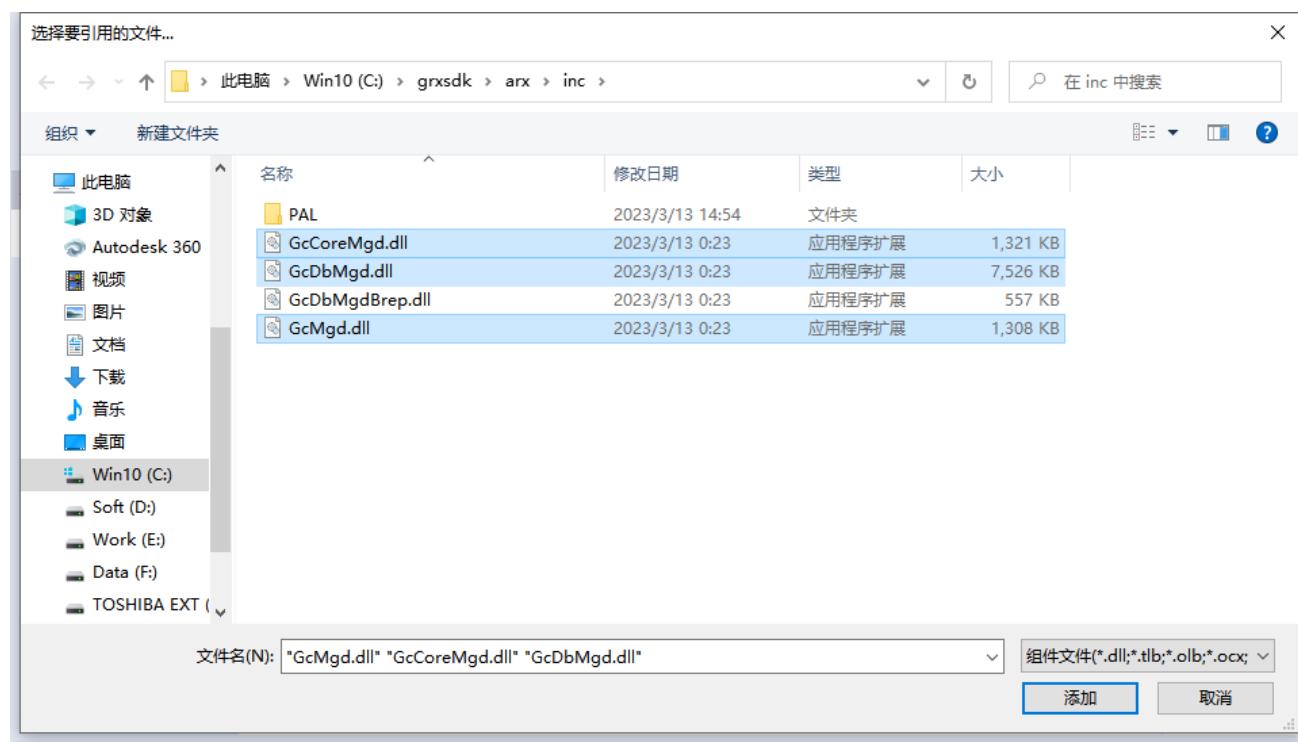


图 5.4

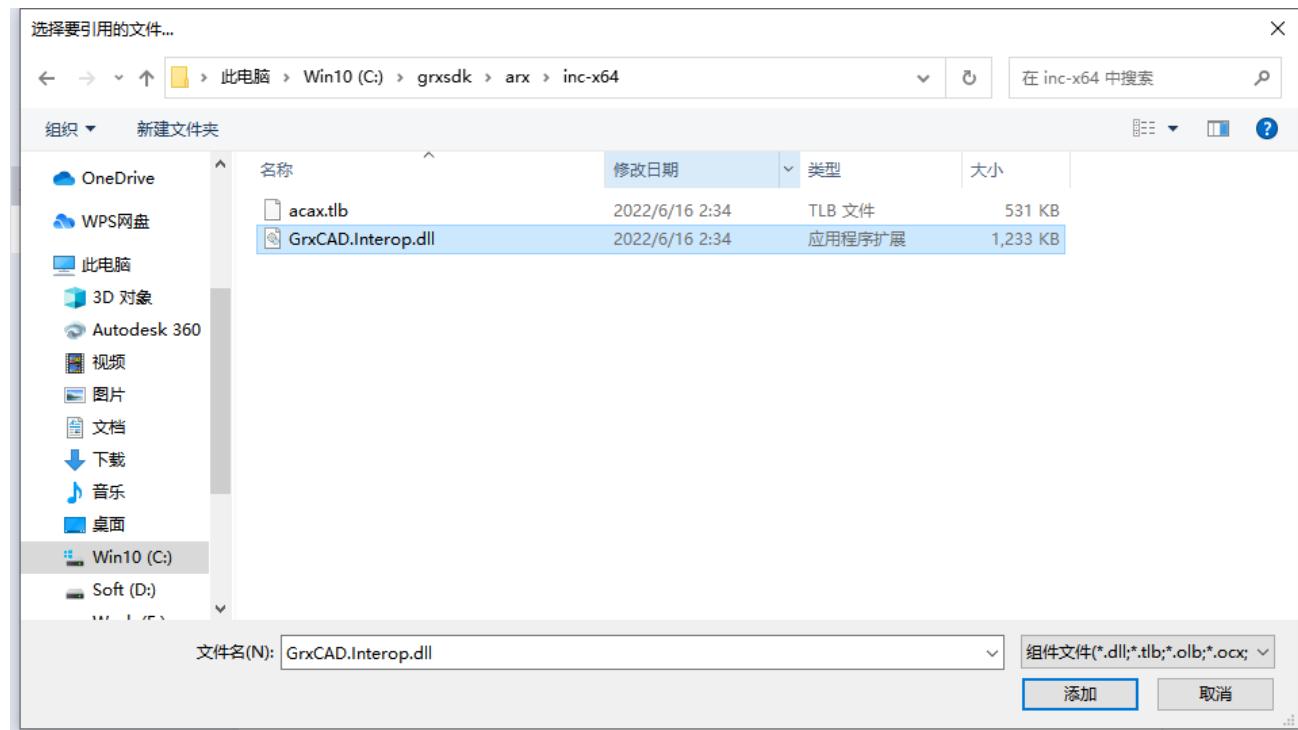


图 5.5

说明：GcCoreMgd. d11、GcDbMgd. d11 和 GcMgd. d11 必须添加，GrxCAD. Interop. d11(或 acax. t1b)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 5.6、图 5.7 所示，本处只给出 GcCoreMgd. d11 的修改样例，其他的引用程序集的修改与 GcCoreMgd. d11 一致。

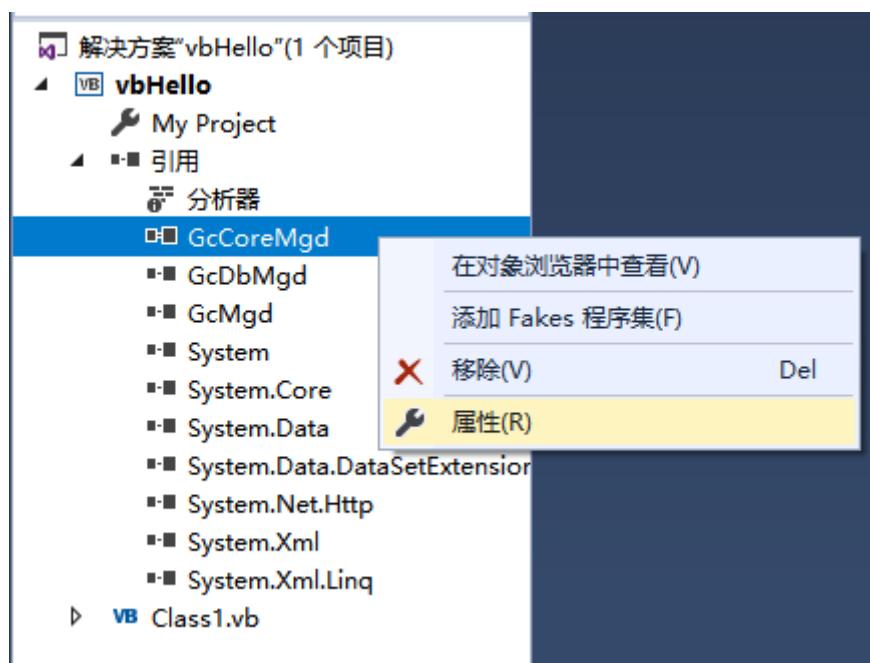


图 5.6

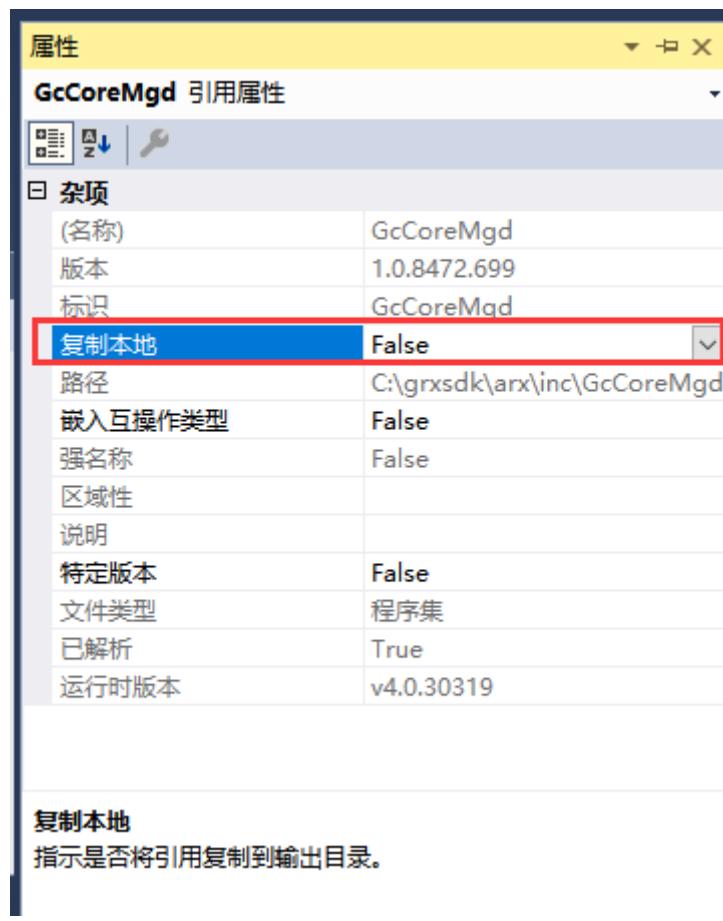


图 5.7

- GcCoreMgd.dll 包括命名空间为：  
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 包括命名空间为：  
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 包括命名空间为：

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 表示 com 的对应的 .NET 接口。

### 5.3 添加代码

在 Class1.vb 里，用下面代码替换已存在的代码：

```
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(GRXTest.MyCommands))>
```

```
Namespace GRXTest
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.WriteLine("hello vb.NET")
End Sub

End Class
End Namespace
```

## 5.4 编译程序

点击 Visual Studio 2017 的菜单项【生成】->【生成解决方案】，会在 F:\test\Hello\Hello\bin\Debug（目录可设定）目录下生成 vbHello.dll。

## 5.5 运行程序

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 Hello.dll，加载后如图 5.8 所示。

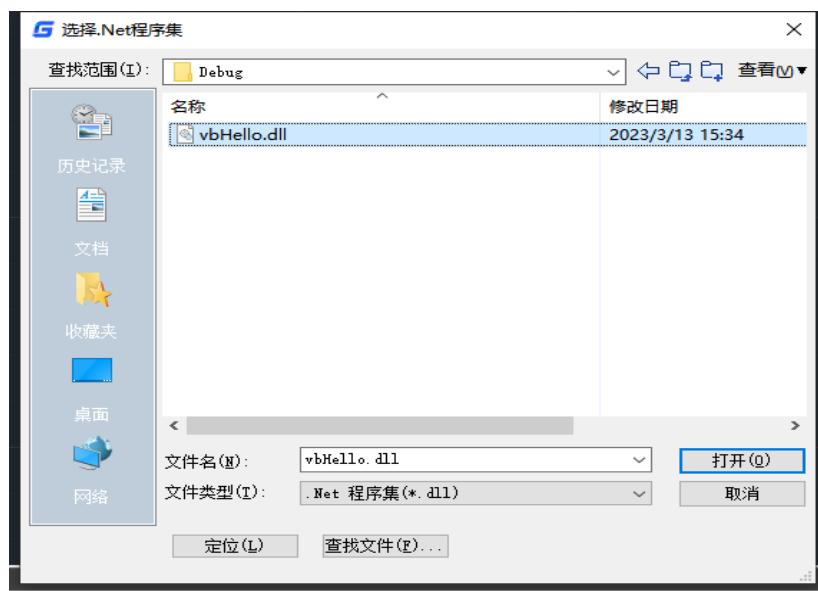


图 5.8

关闭加载程序对话框，在浩辰 CAD 命令行输入“vbhello”命令。会在命令行上打印输出：

“hello vb.NET”。运行效果如图 5.9 所示。

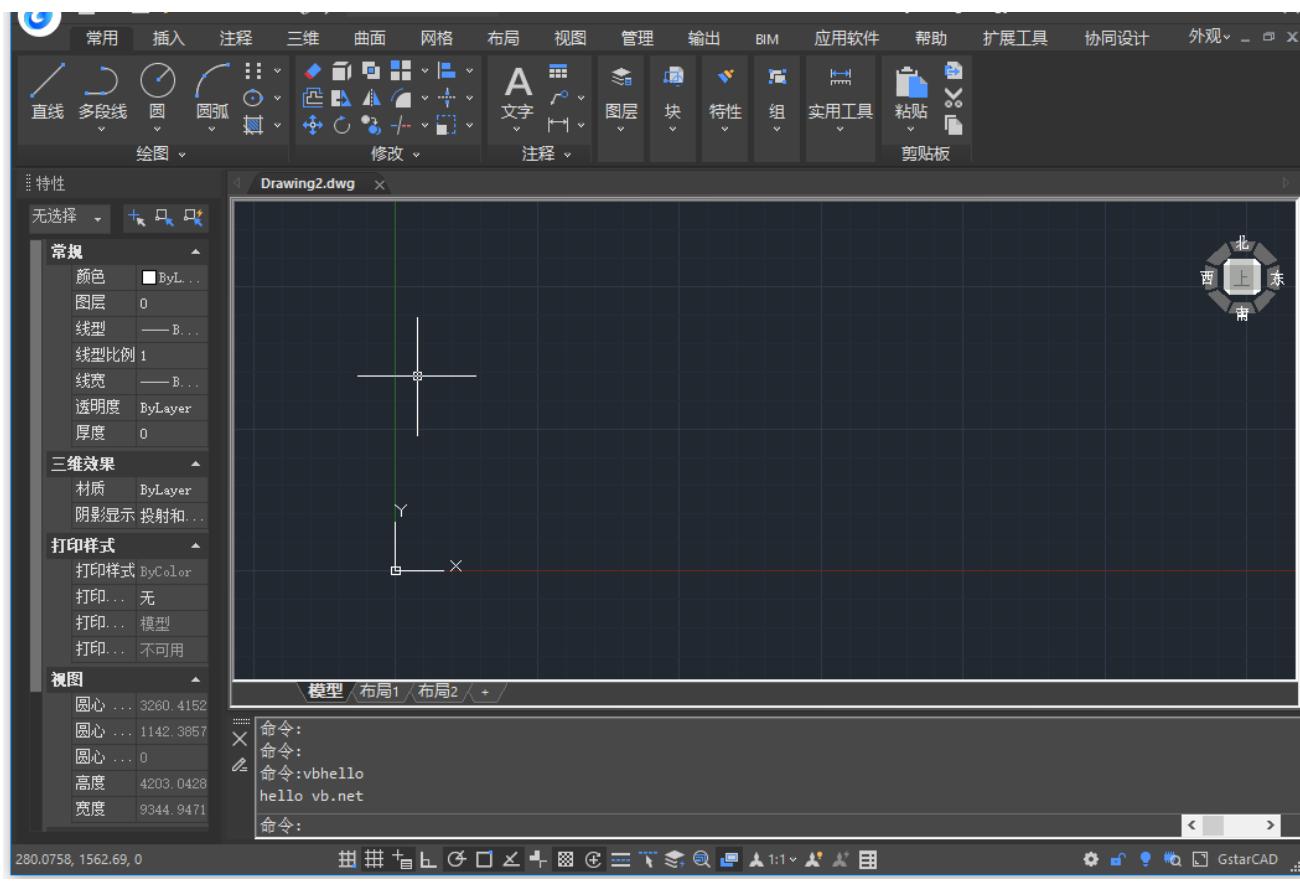


图 5.9

## 6 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：

### 6.1 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())
{
    ...
}
```

### 6.2 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)”。

### 6.3 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 Gsoft.Gscad.EditorInput.Editor.Command(params object[]) 代替。

### 6.4 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);

}

}
```

## 6.5 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。

## 7 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。