.NET Programming Guide

GstarCAD 2027

Table of Contents

1. Introduction 1

2. Programming Environment 2

3. Install SDK 3

4. Visual Studio 2022 C# .NET Programming Sample 4

    4.1. Create .NET project 4

        4.1.1. Runn Microsoft® Visual Studio 2022 4

        4.1.2. Input Project Save Path and Project Name 4

        4.1.3. Finish Creating New Project 5

    4.2. Set Reference Library 6

        4.2.1. Add Reference 6

        4.2.2. Choose and Add Reference File 7

    4.3. Add Code 11

    4.4. Compile Program 12

    4.5. Run Program 12

5. Visual Studio 2022 VB .NET Programming Sample 14

    5.1. Create .NET project 14

        5.1.1. Run Microsoft® Visual Studio 2022 14

        5.1.2. Input Project Save Path and Project Name 14

        5.1.3. Finish Creating New Project 15

    5.2. Set the Reference 16

        5.2.1. Add Reference 16

        5.2.2. Choose and Add Reference File 17

    5.3. Add code 21

    5.4. Compile Program 21

    5.5. Run Program 21

6. C# .NET and VB .NET Namespace Modification 24

7. Copyright 25

1. Introduction

This .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.

2. Programming Environment

- Microsoft® Visual Studio 2022 (version 17.8.0)
- Windows SDK 10.0 (latest installed version)
- CPU:

  Basic: 1.6 GHz CPU

  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)
- Monitor Resolution:

  1028x800 and above true color display, including 4K (3840x2160) display
- GRXSDK
- Microsoft .NET 8.0

3. Install SDK

Download SDK.

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, HelloARX 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.

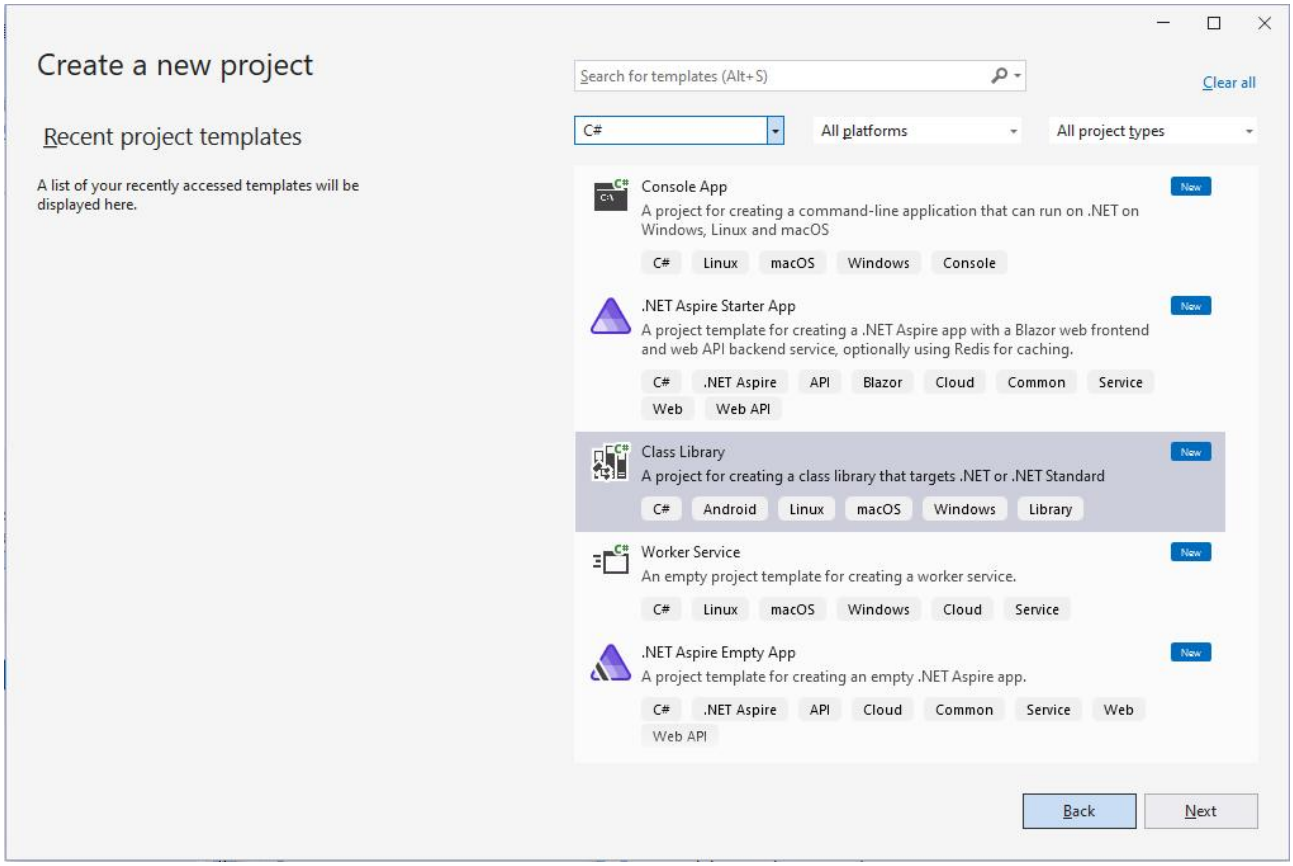
4. Visual Studio 2022 C# .NET Programming Sample

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

4.1. Create .NET project

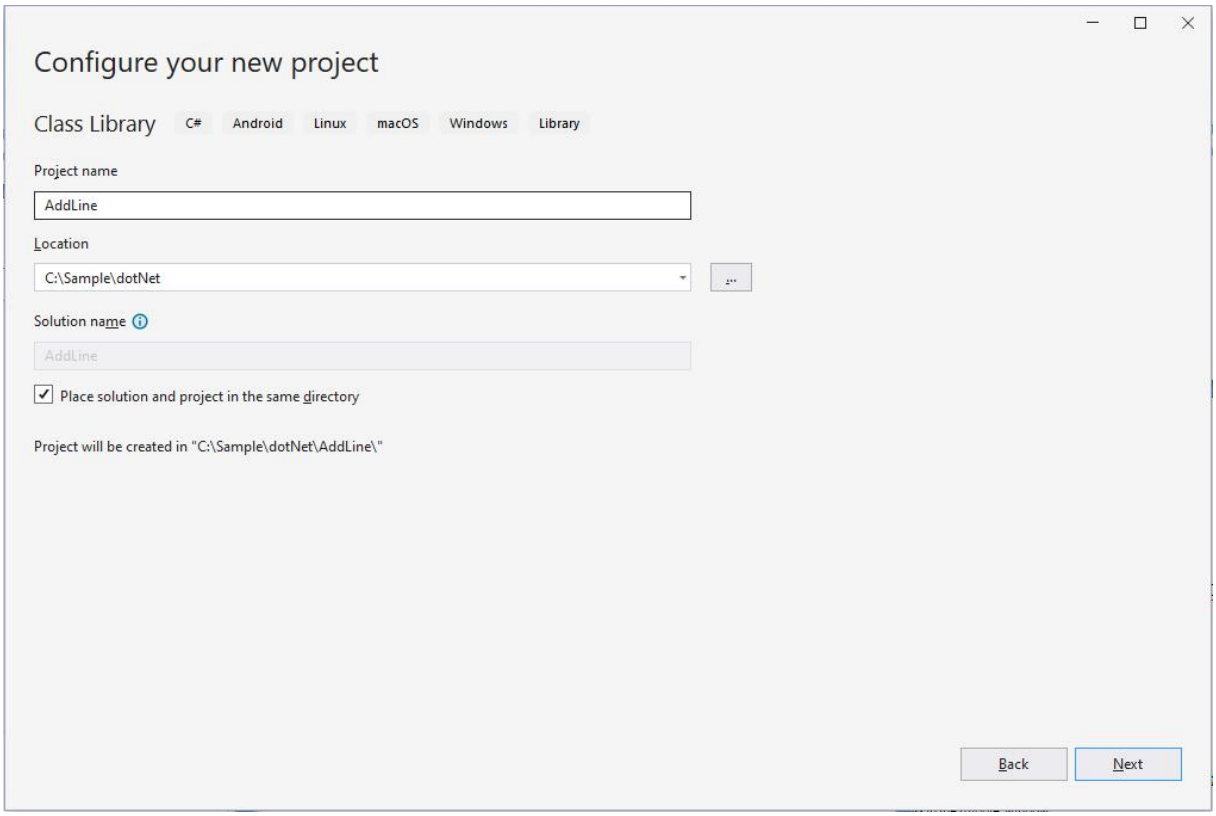
4.1.1. Runn Microsoft® Visual Studio 2022

Click **File→New→Project** to launch the **New Project** dialog window. Select **C#** from the drop down list on the right side and click **Class Library** from the list frame.



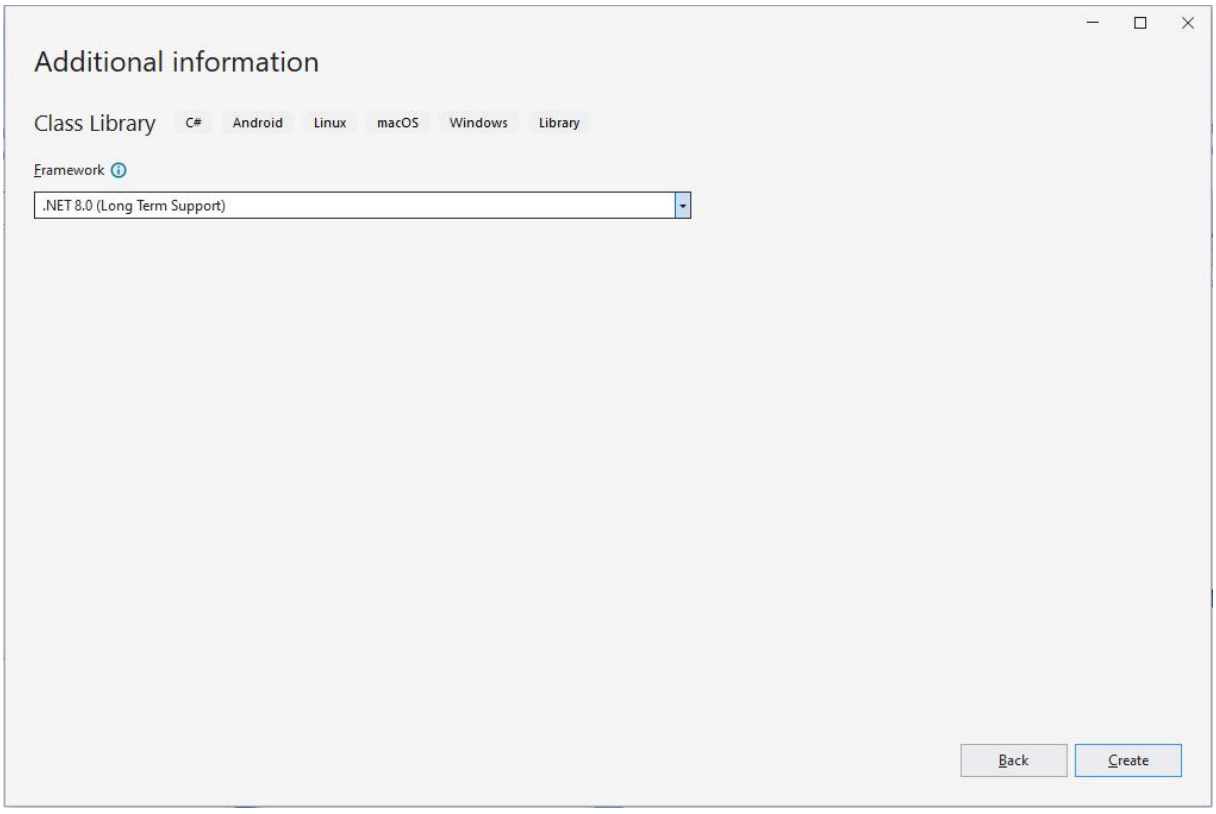
4.1.2. Input Project Save Path and Project Name

Input '*AddLine*' at the **Project Name** field in the **New Project** dialog box, enter '*C:\Sample\dotNet*" in the **Location** field, and click **Next**. As shown below.



4.1.3. Finish Creating New Project

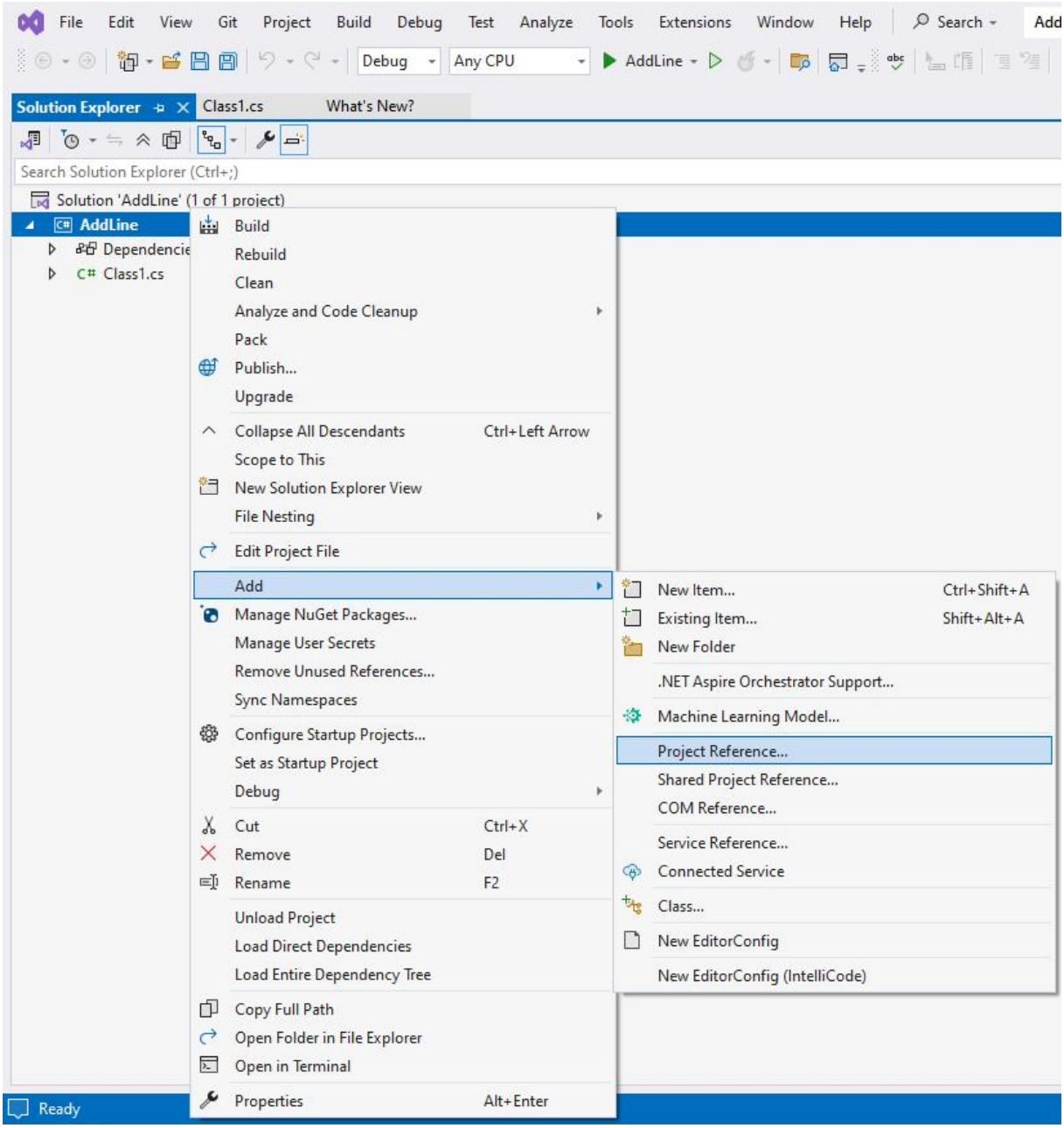
Select **.NET 8.0 (Long Term Support)** in the drop-down list and click the **Create** button to finish creating the new project '*AddLine*'.



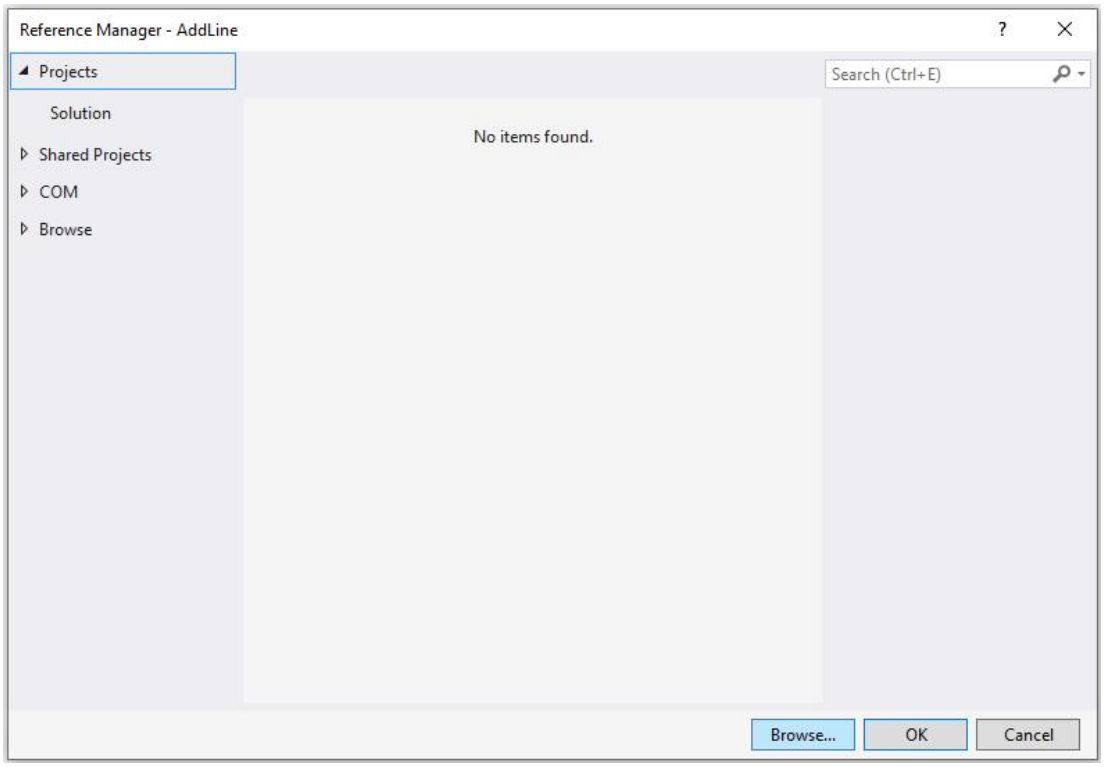
4.2. Set Reference Library

4.2.1. Add Reference

In Visual Studio 2022, find the '*AddLine*' project in the **Solution Explorer**, right-click on it and select **Add→Reference**.

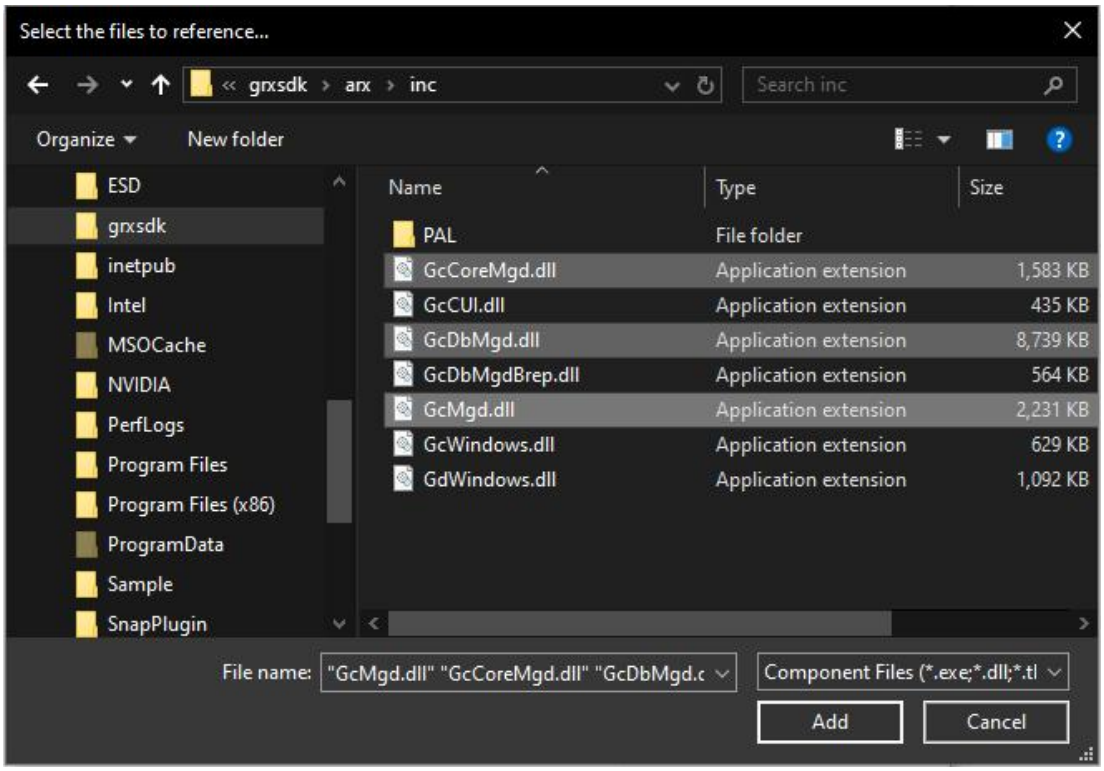


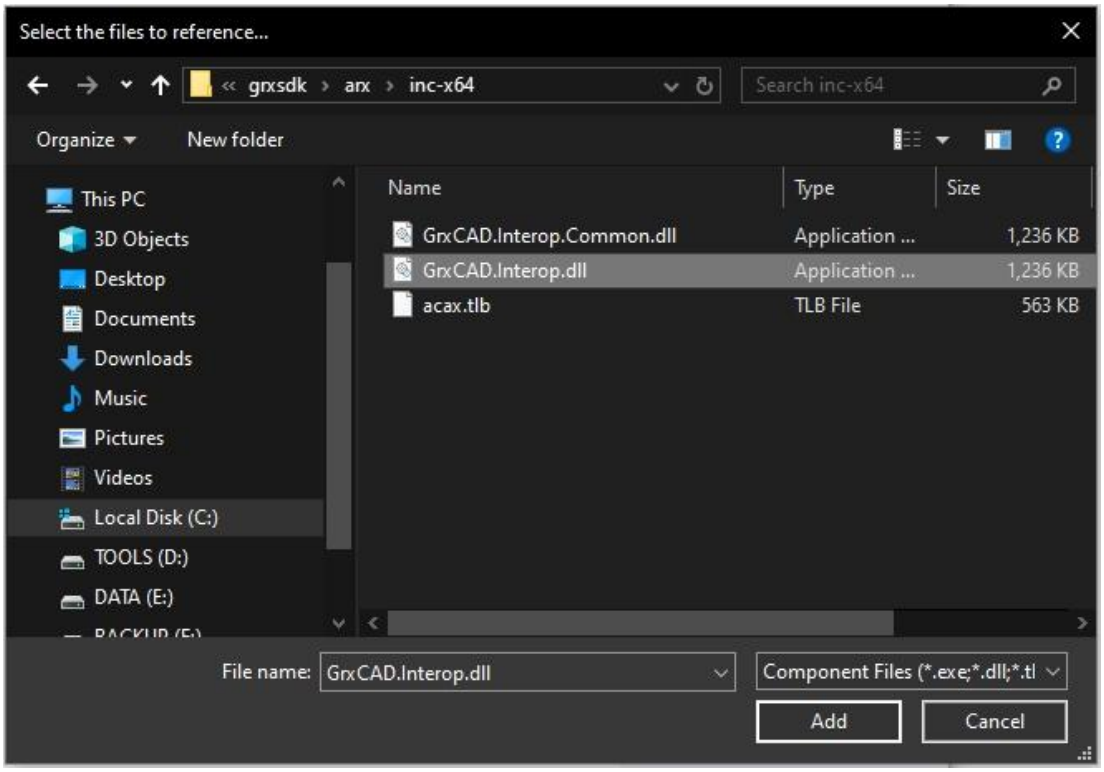
After the **Reference Manager** dialog window pops out, click **Browse** as shown below.

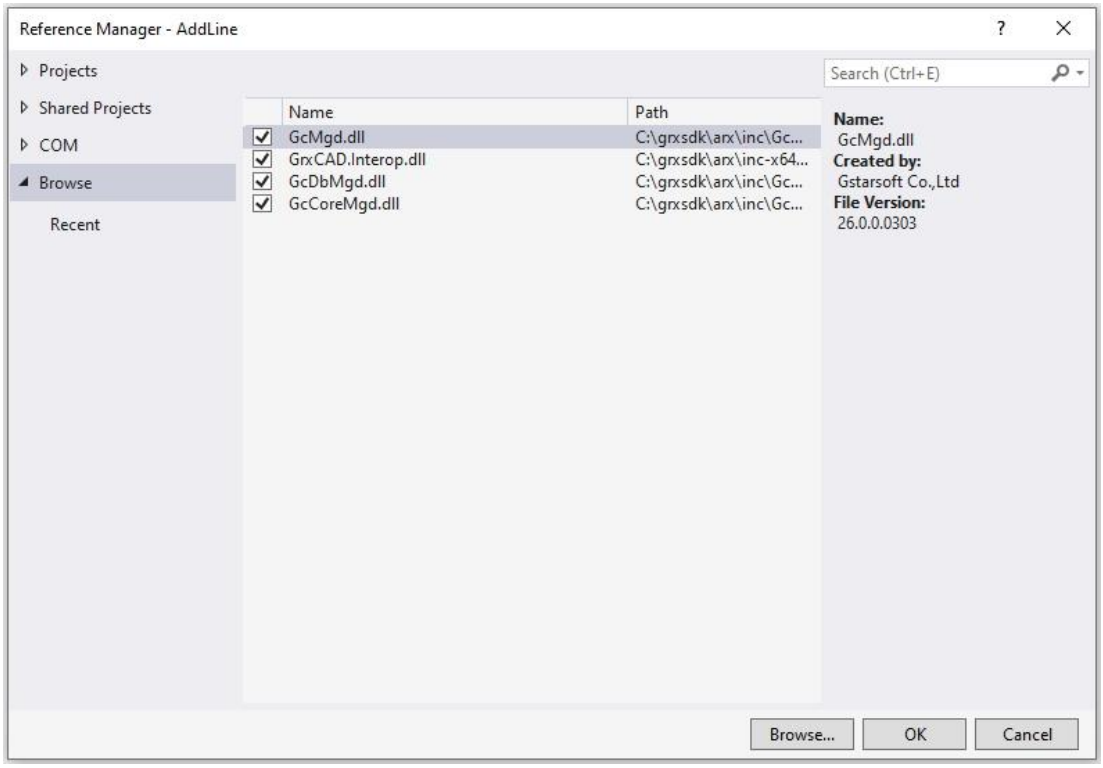


4.2.2. Choose and Add Reference File

From the 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.

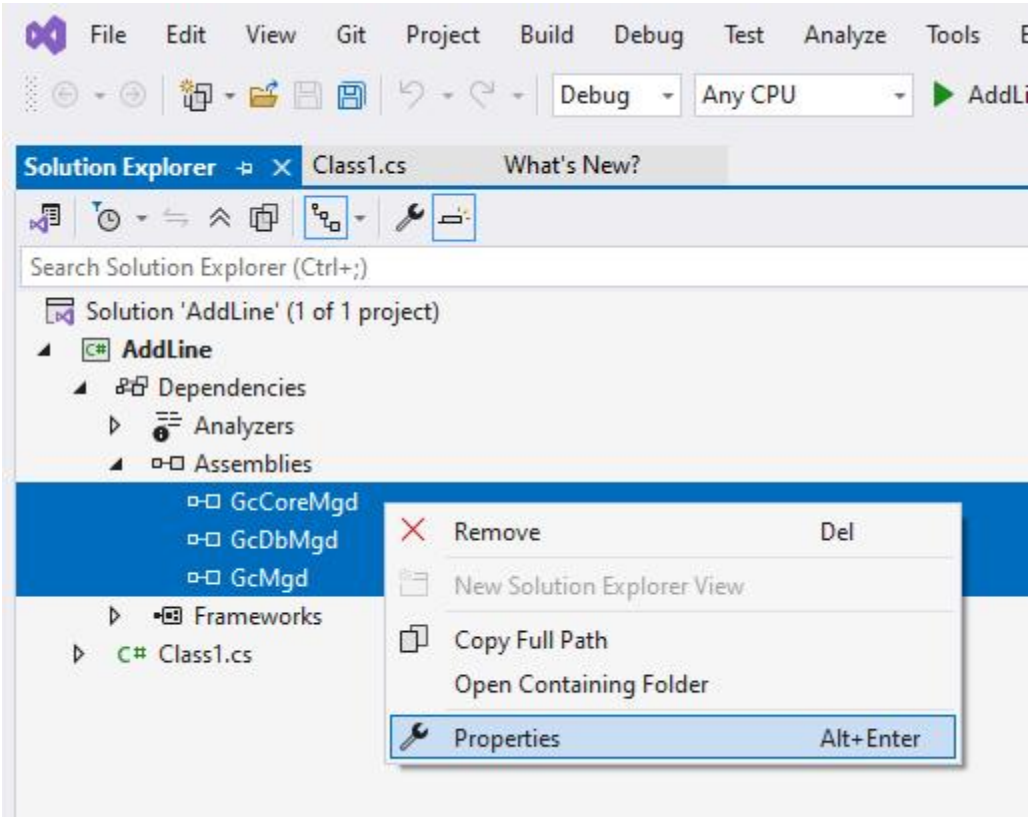


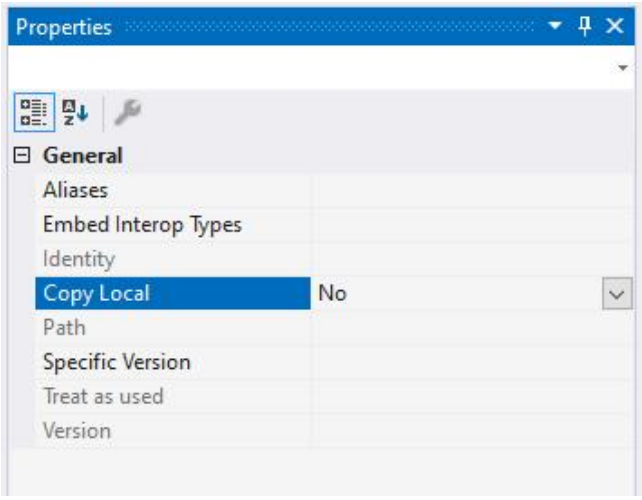




NOTE: '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' must be added while '*GrxCAD.Interop.dll*' (or '*acax.tlb*') is optional (needed when COM is used in the program).

After adding the files, right-click added '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' files 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.PAL.FontUtils
  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.PAL.FontUtils

  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.PAL.FontUtils

  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 Gssoft.Gscad.ApplicationServices;
using Gssoft.Gscad.DatabaseServices;
using Gssoft.Gscad.Geometry;
using Gssoft.Gscad.Runtime;

namespace AddLine
{
    public class Class1
    {
        [CommandMethod("addline")]
        static public void DoIt()
        {
            Point3d st = new Point3d(100.0, 100.0, 0.0);
            Point3d ed = new Point3d(200.0, 200.0, 0.0);
            Line lin = new Line(st, ed);
            Database db = HostApplicationServices.WorkingDatabase;
            Transaction trans = db.TransactionManager.StartTransaction();
            try
            {
                BlockTable bt = (BlockTable)trans.GetObject(db.BlockTableId,
OpenMode.ForRead);
                BlockTableRecord btr =
(BlockTableRecord)trans.GetObject(bt[BlockTableRecord.ModelSpace], OpenMode.ForWrite);
                btr.AppendEntity(lin);
                trans.AddNewlyCreatedDBObject(lin, true);
                trans.Commit();
            }
            catch (System.Exception ex)
            {

Application.DocumentManager.MdiActiveDocument.Editor.WriteMessage(ex.ToString());
                trans.Abort();
            }
        }
    }
}
```

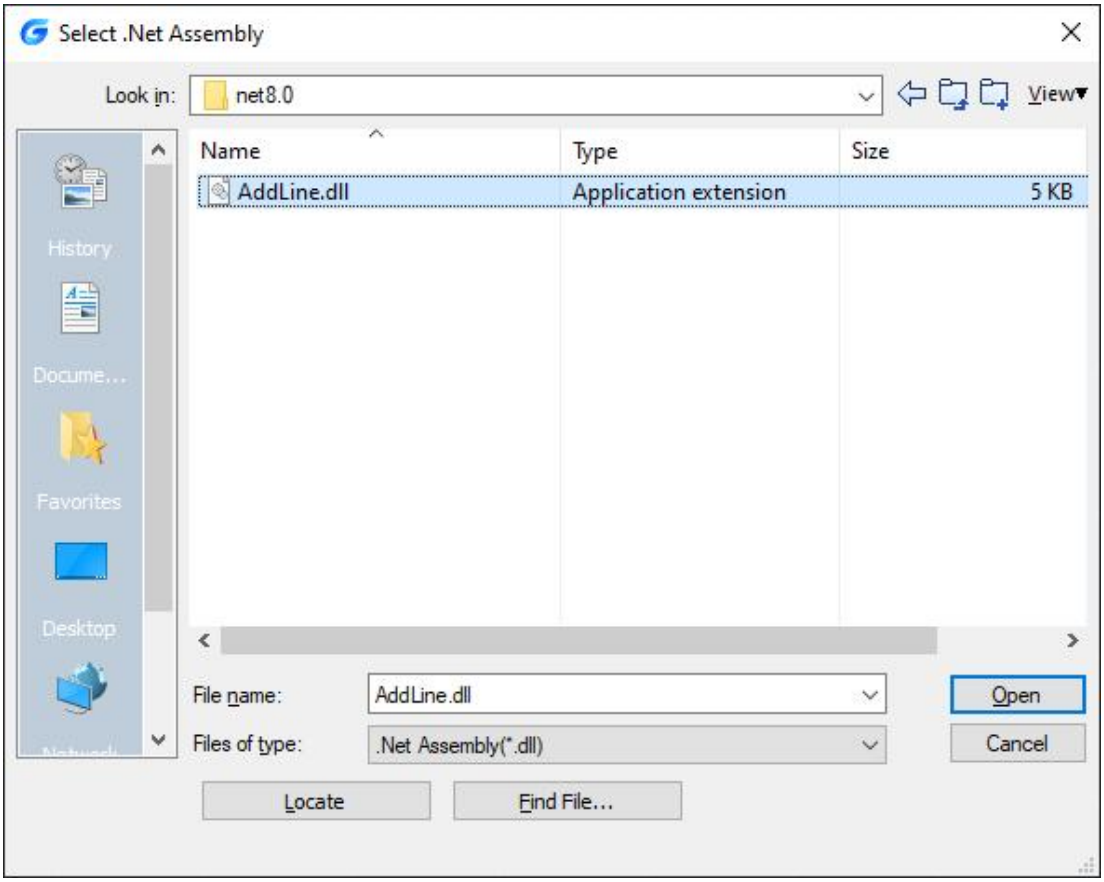
4.4. Compile Program

In Visual Studio 2022, click **Build→Rebuild Solution** to generate '*Addline.dll*' file.

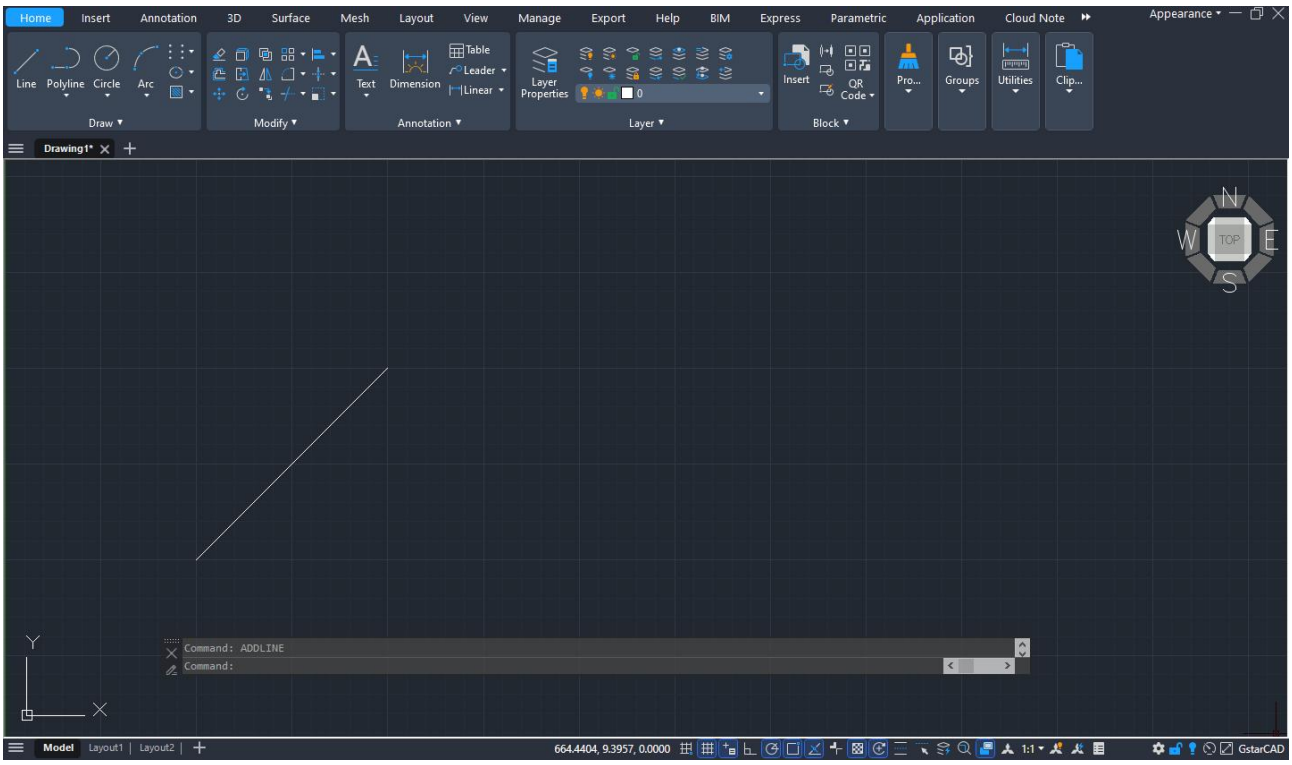
And AddLine.dll will be generated in the path 'C:\Sample\dotNet\AddLine\bin\Debug\net8.0 '(the path can be set).

4.5. Run Program

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



Input '*AddLine*' at command line, there will be a straight line generated.


Home
Insert
Annotation
3D
Surface
Mesh
Layout
View
Manage
Export
Help
BIM
Express
Parametric
Application
Cloud Note
Appearance
Line
Polyline
Circle
Arc
Draw
Modify
Text
Dimension
Table
Leader
Linear
Annotation
Layer Properties
Layer
Insert
QR Code
Block
Pro...
Groups
Utilities
Clip...
Drawing1*
N
W
TOP
E
S
Command: ADDLINE
Command:
Model
Layout1
Layout2
664.4404, 9.3957, 0.0000
1:1
GstarCAD

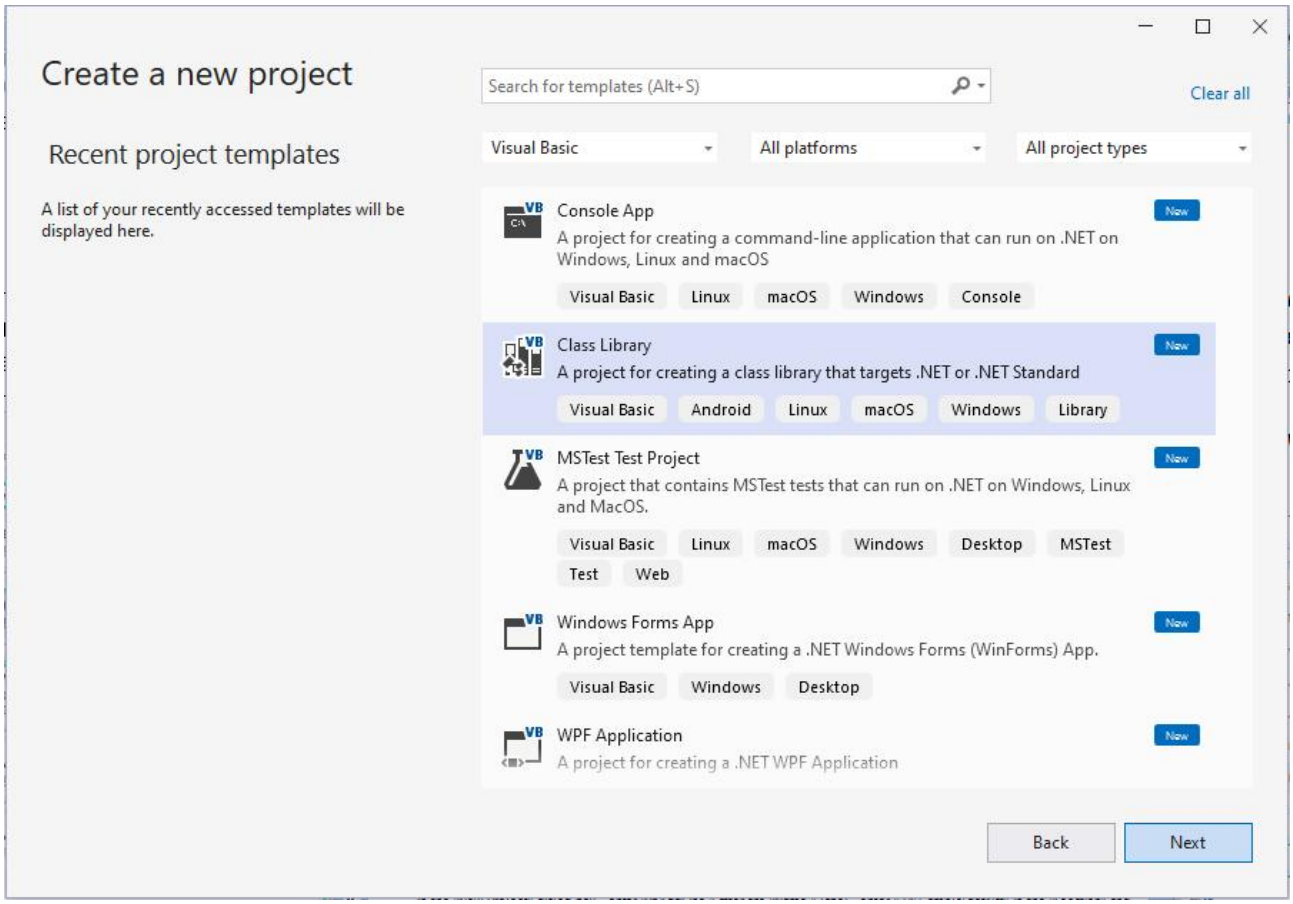
5. Visual Studio 2022 VB .NET Programming Sample

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

5.1. Create .NET project

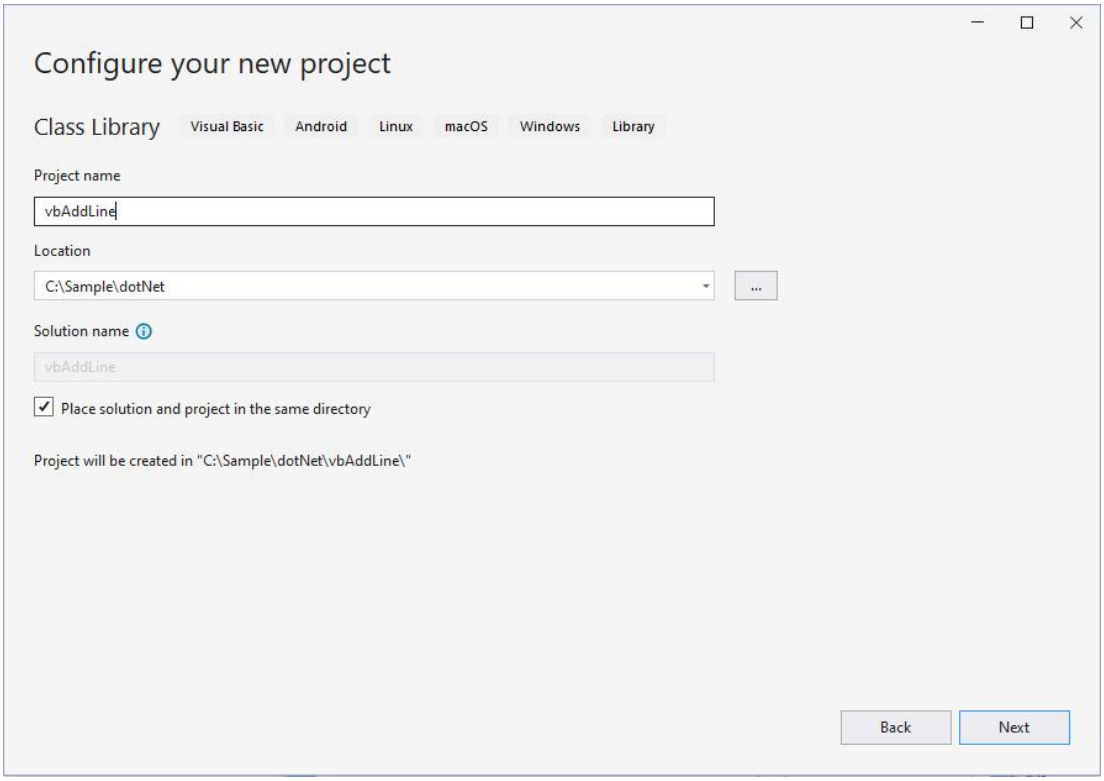
5.1.1. Run Microsoft® Visual Studio 2022

Click **File→New→Project** to launch the **New Project** dialog window. Select **Visual Basic** in the drop down list on the right side and click **Class Library** from the list frame.



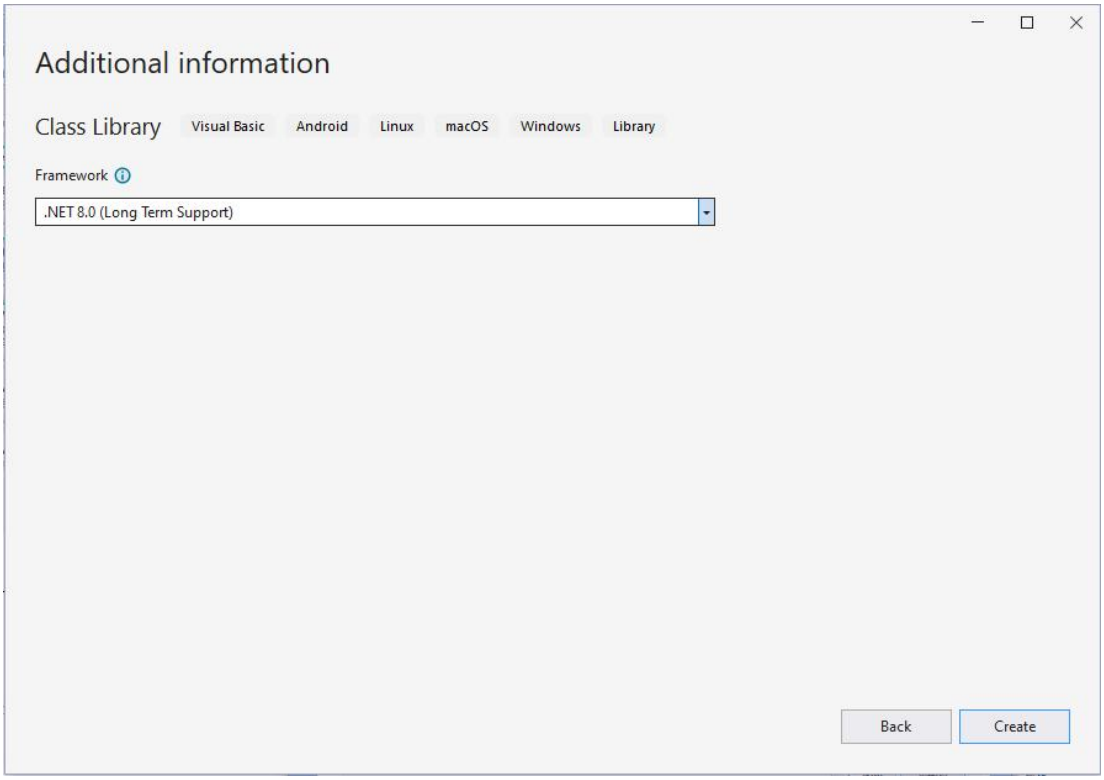
5.1.2. Input Project Save Path and Project Name

Input '*vbAddLine*' at the **Project Name** field in the **New Project** dialog box, enter '*C:\Sample\dotNet*' in the **Location** field, and click **Next**. As shown below.



5.1.3. Finish Creating New Project

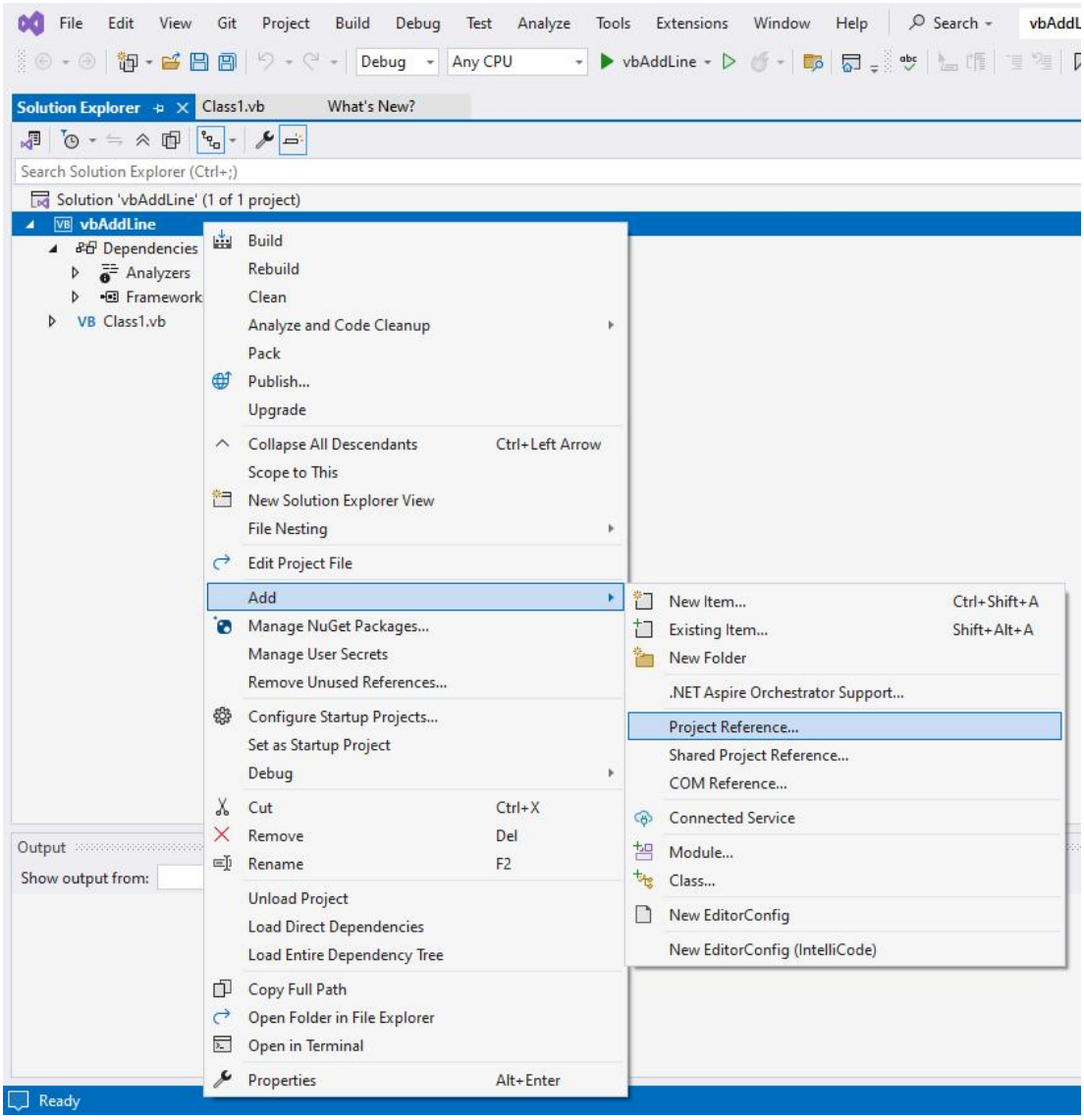
Select **.NET 8.0 (Long Term Support)** in the drop-down list and click the **Create** button to finish creating the new project '*AddLine*'.



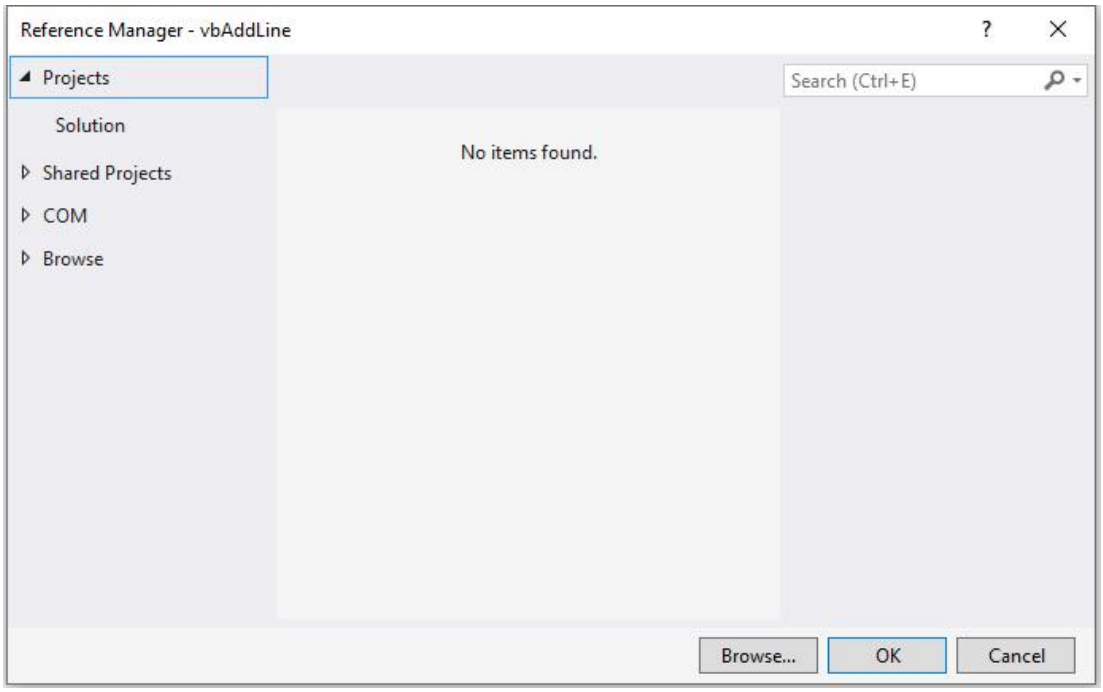
5.2. Set the Reference

5.2.1. Add Reference

In Visual Studio 2022, find the '*vbAddLine*' project in the **Solution Explorer**, right-click on it and select **Add→Reference**.

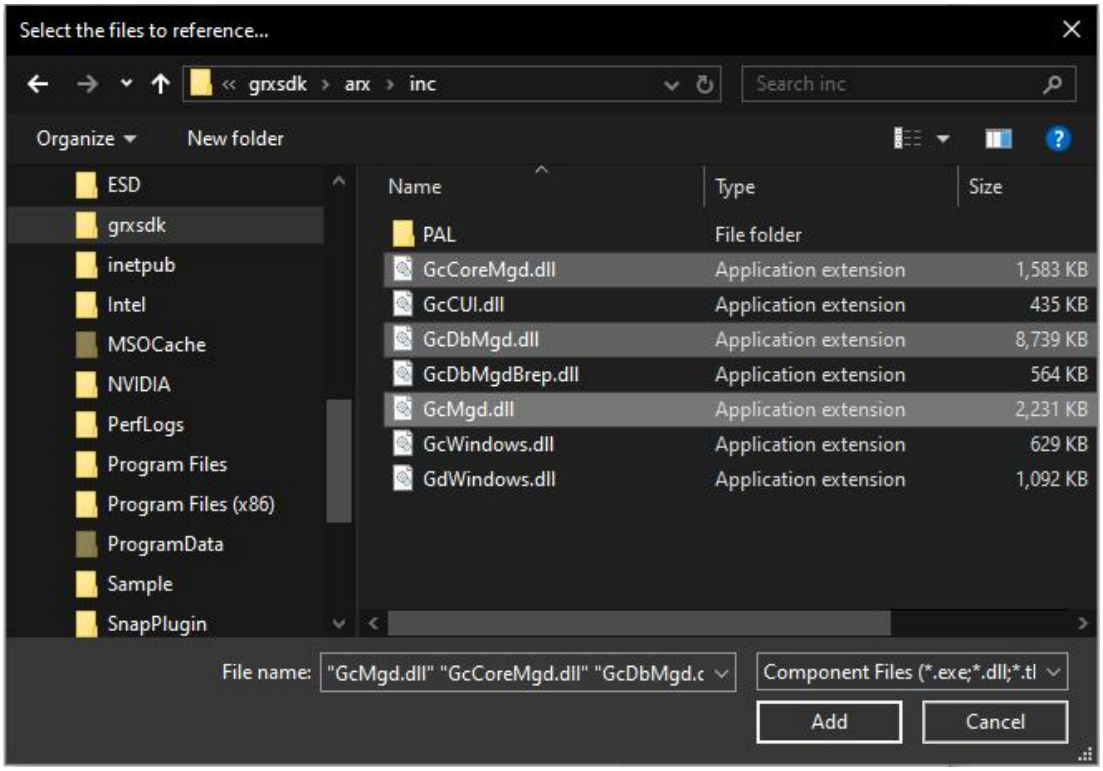


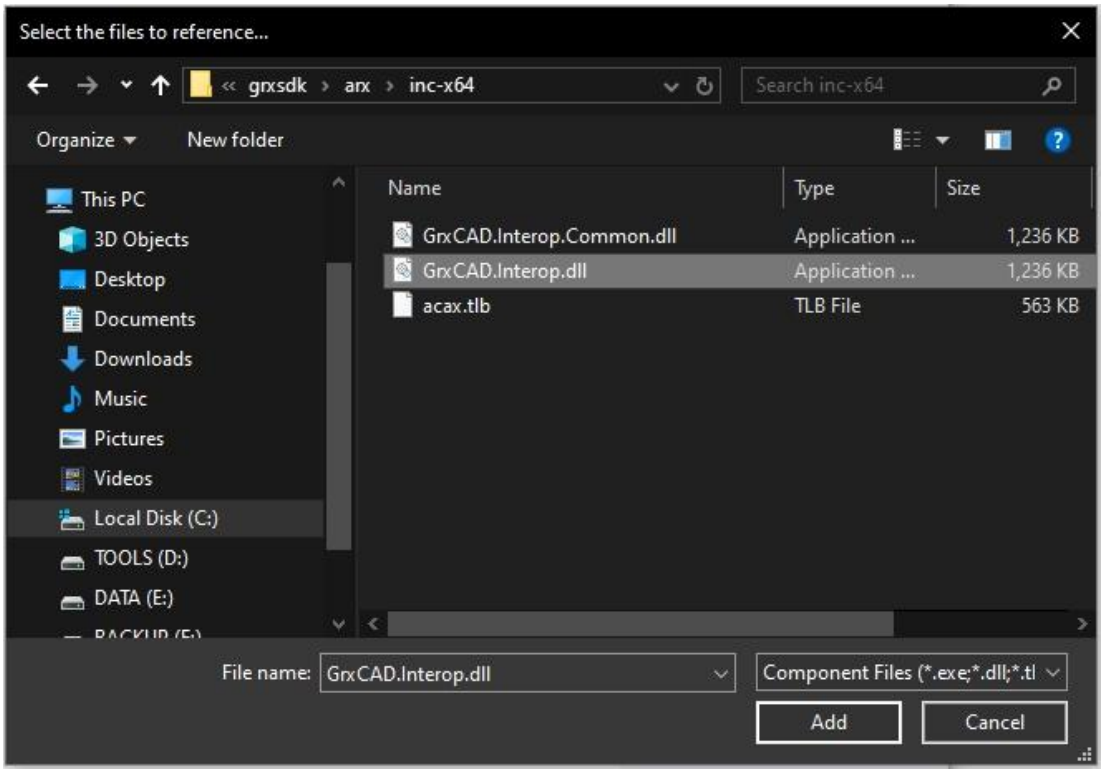
After the **Reference Manager** dialog window pops out, click **Browse** as shown below.

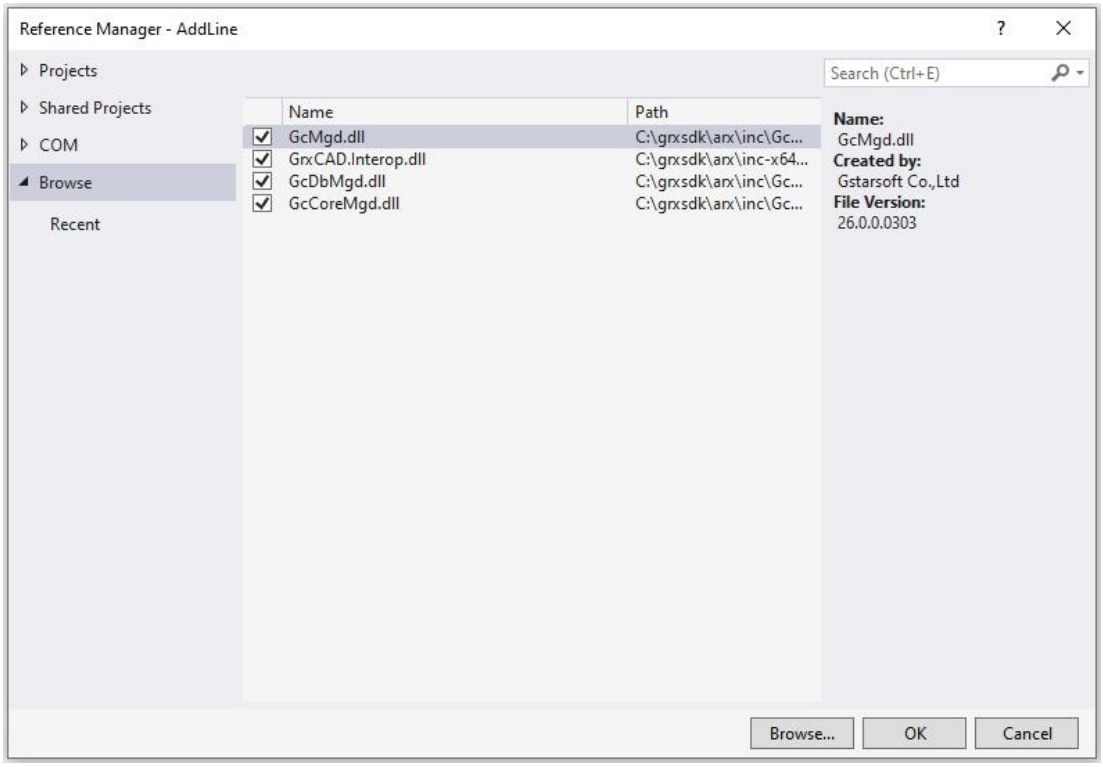


5.2.2. Choose and Add Reference File

From the 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.

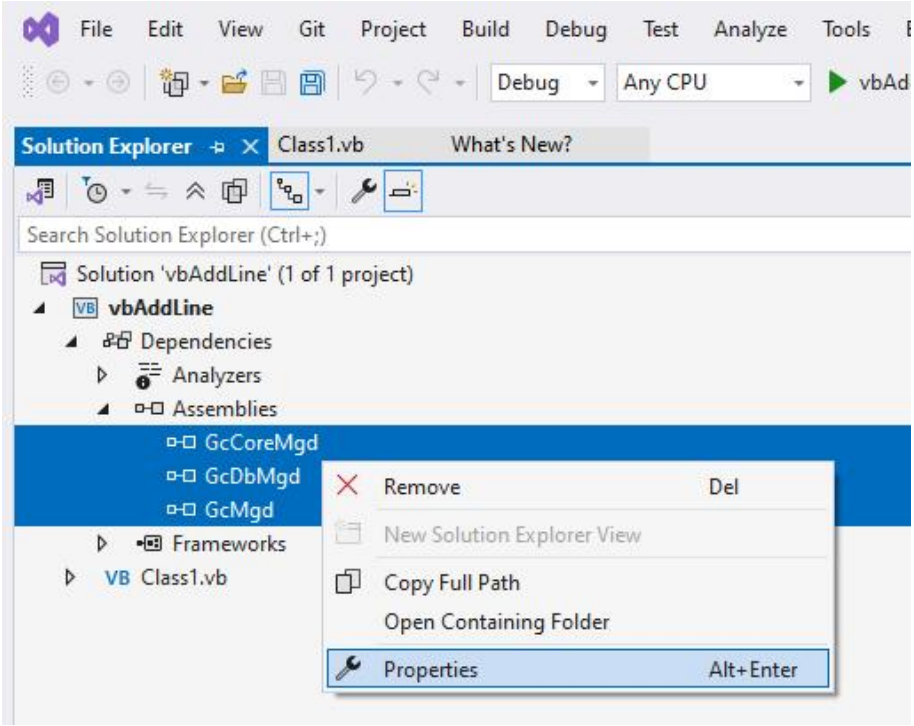


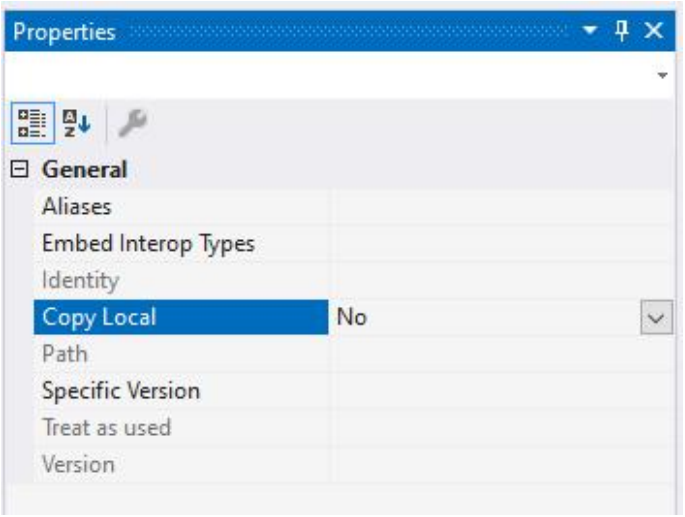




NOTE: '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' must be added while '*GrxCAD.Interop.dll*' (or '*acax.tlb*') is optional (needed when COM is used in the program).

After adding the files, right-click added '*GcCoreMgd.dll*', '*GcDbMgd.dll*' and '*GcMgd.dll*' files 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.PAL.FontUtils

  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.PAL.FontUtils

  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.PAL.FontUtils

  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 to '*Class1.vb*':

```
Imports Gssoft.Gscad.ApplicationServices
Imports Gssoft.Gscad.DatabaseServices
Imports Gssoft.Gscad.Geometry
Imports Gssoft.Gscad.Runtime

Public Class Class1
    <CommandMethod("vbAddLine")>
    Public Sub vbAddLine()
        Dim startpt As New Point3d(4.0, 2.0, 0.0)
        Dim endpt As New Point3d(10.0, 7.0, 0.0)
        Dim pLine As New Line(startpt, endpt)
        Dim lineid As ObjectId
        Dim db As Database = Application.DocumentManager.MdiActiveDocument.Database
        Dim tm As Gssoft.Gscad.DatabaseServices.TransactionManager = db.TransactionManager
        Dim ta As Transaction = tm.StartTransaction()
        Try
            Dim bt As BlockTable = tm.GetObject(db.BlockTableId, OpenMode.ForRead, False)
            Dim btr As BlockTableRecord = tm.GetObject(bt(BlockTableRecord.ModelSpace),
OpenMode.ForWrite, False)
            lineid = btr.AppendEntity(pLine)
            tm.AddNewlyCreatedDBObject(pLine, True)
            ta.Commit()
        Finally
            ta.Dispose()
        End Try
    End Sub
End Class
```

5.4. Compile Program

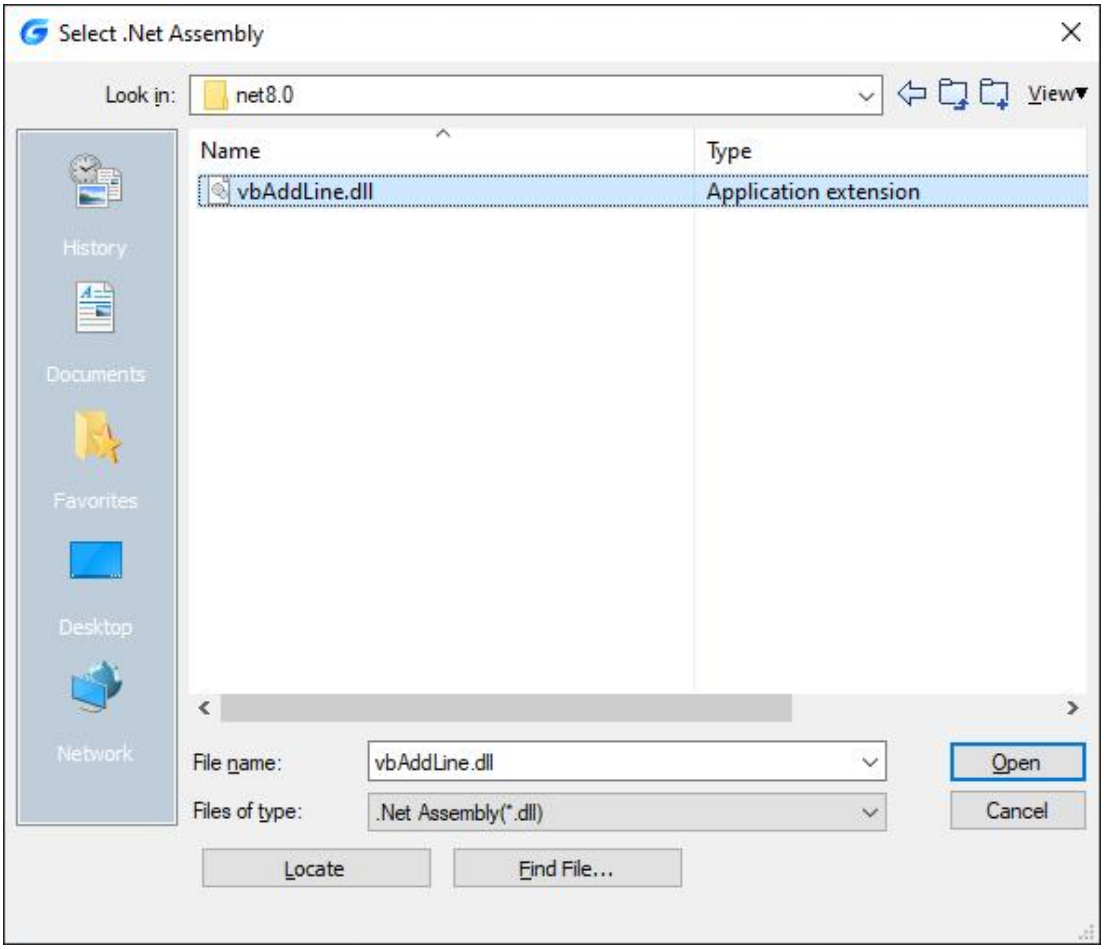
In Visual Studio 2022, click **Build→Rebuild Solution** to generate '*Addline.dll*' file.

And AddLine.dll will be generated in the path 'C:\Sample\dotNet\vbAddLine\bin\Debug\net8.0 '(the path can be set).

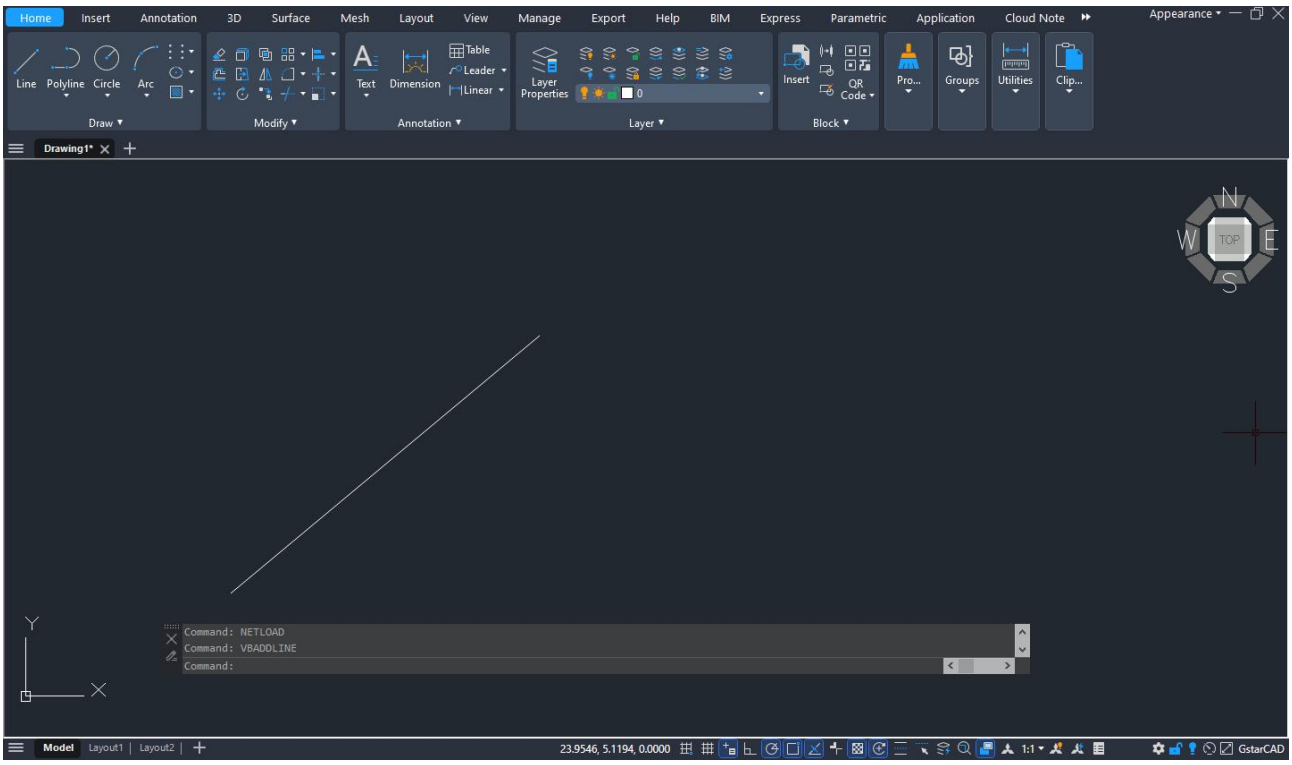
5.5. Run Program

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


Command:
Command:
Command: NETLOAD
Command:


Select .Net Assembly
Look in:
net8.0
View
Name
Type
vbAddLine.dll
Application extension
History
Documents
Favorites
Desktop
Network
File name:
vbAddLine.dll
Open
Files of type:
.Net Assembly(*.dll)
Cancel
Locate
Find File...


Command: NETLOAD
Command: VBADDLINE
Command:


Home
Insert
Annotation
3D
Surface
Mesh
Layout
View
Manage
Export
Help
BIM
Express
Parametric
Application
Cloud Note
Appearance
Line
Polyline
Circle
Arc
Draw
Modify
Text
Dimension
Table
Leader
Linear
Annotation
Layer Properties
Layer
Insert
QR Code
Block
Groups
Utilities
Drawing1*
N
W
TOP
E
S
Y
Command: NETLOAD
Command: VBADDLINE
Command:
Model
Layout1
Layout2
23.9546, 5.1194, 0.0000
1:1
GstarCAD

6. 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

GstarCAD 2027

https://www.gstarcad.net

Gstarsoft