

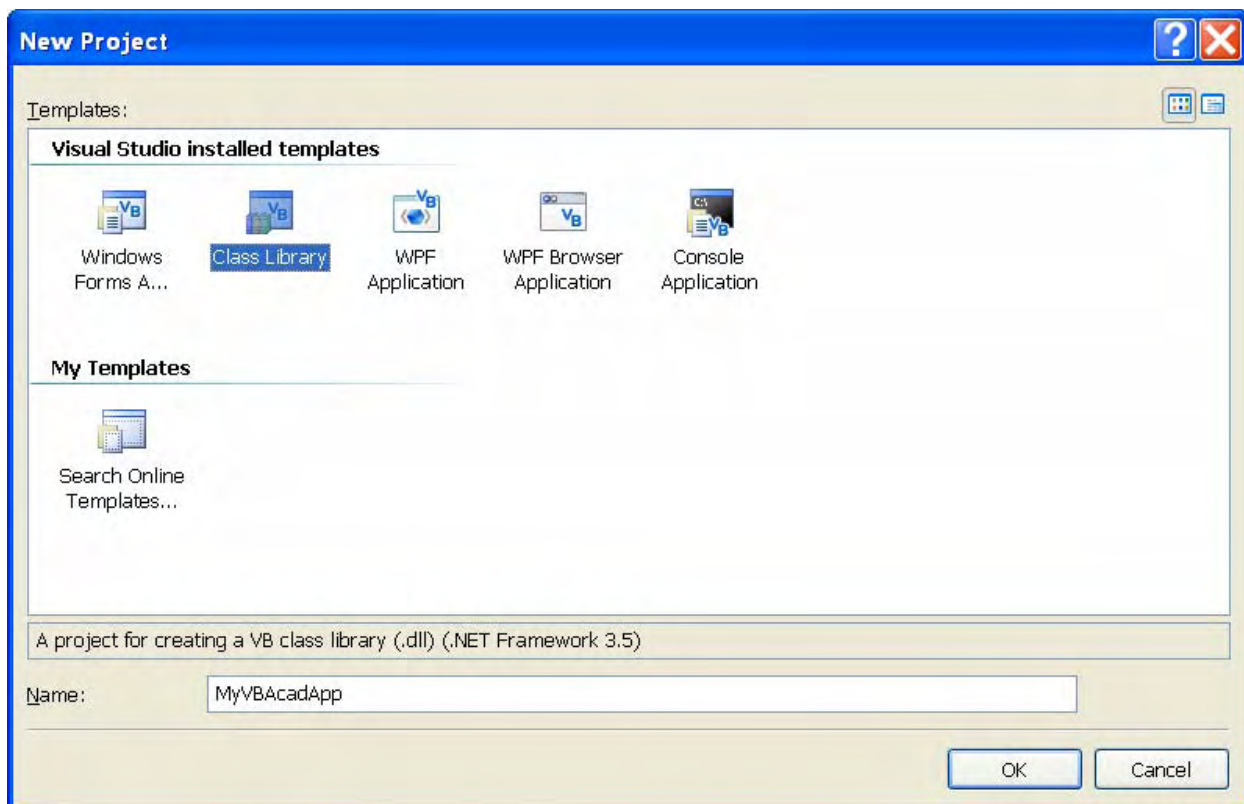
# Using Visual Basic Express to create an AutoCAD .NET addin

## Introduction

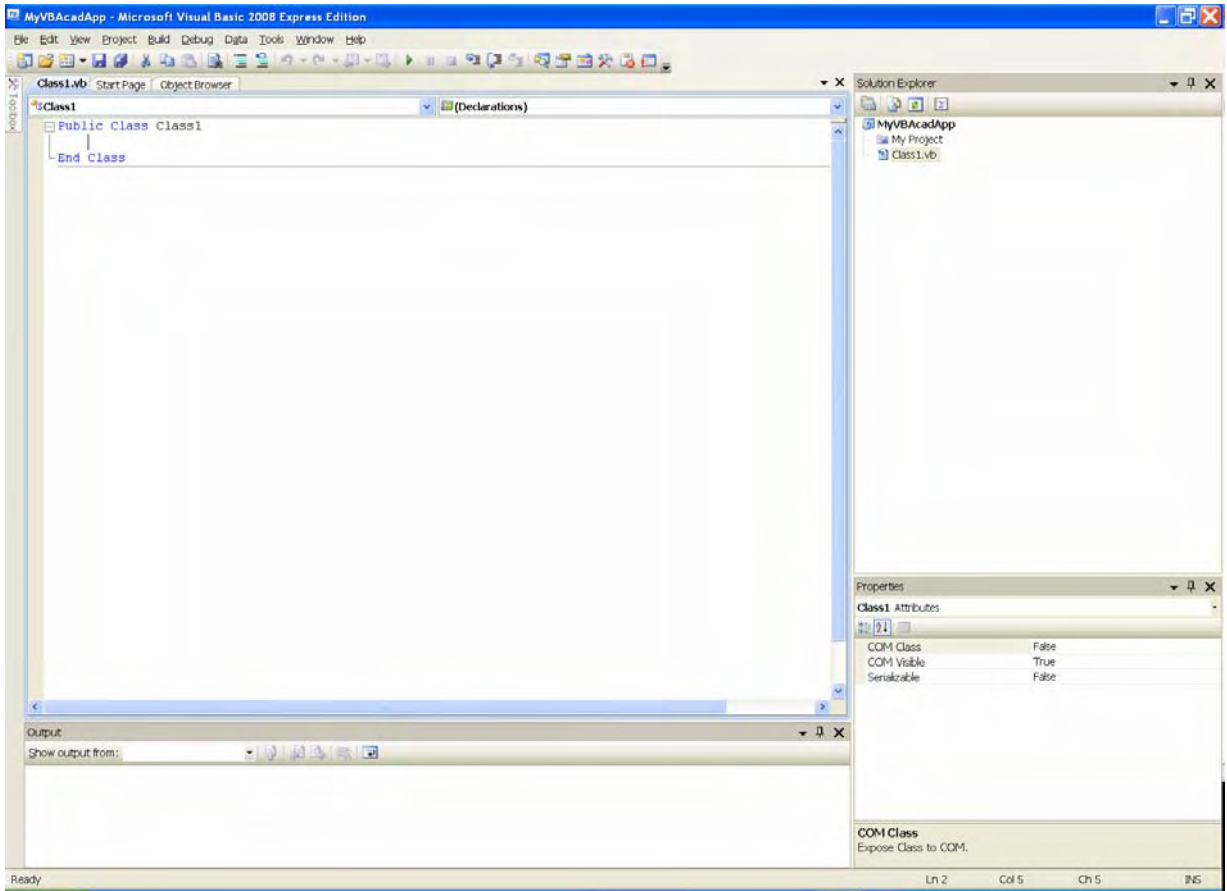
This document will provide a step by step tutorial on how to create, load and debug a simple VB .NET application for AutoCAD using the Visual Basic Express development environment. The document focuses on setting up and using the Visual Basic Express environment. We do not elaborate on the AutoCAD .NET API itself. We provide references at the end of this tutorial to resources for learning the AutoCAD .NET API itself.

## Creating the project:

- 1) Launch Visual Basic 2008 Express Edition
- 2) Select the Menu item **File**→**"New Project..."** or type Ctrl+N. The New Project dialog will appear:

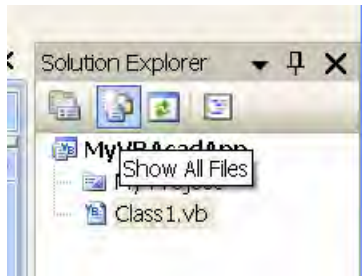


- 3) Select the **"Class Library"** Template (shown above) and enter a suitable name in the **"Name:"** text field. For Instance, we chose **"MyVBAcadApp"**.
- 4) A new Project is now created and the Visual Studio Express window should look something like this:

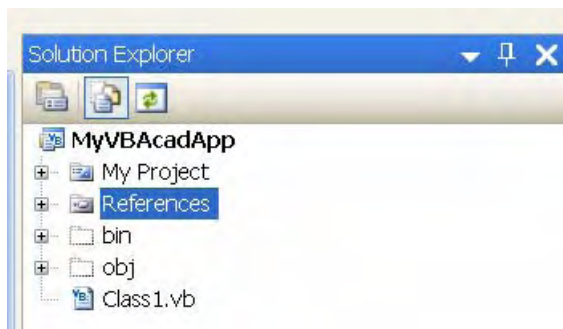


## Converting your Project into an AutoCAD VB .NET Application:

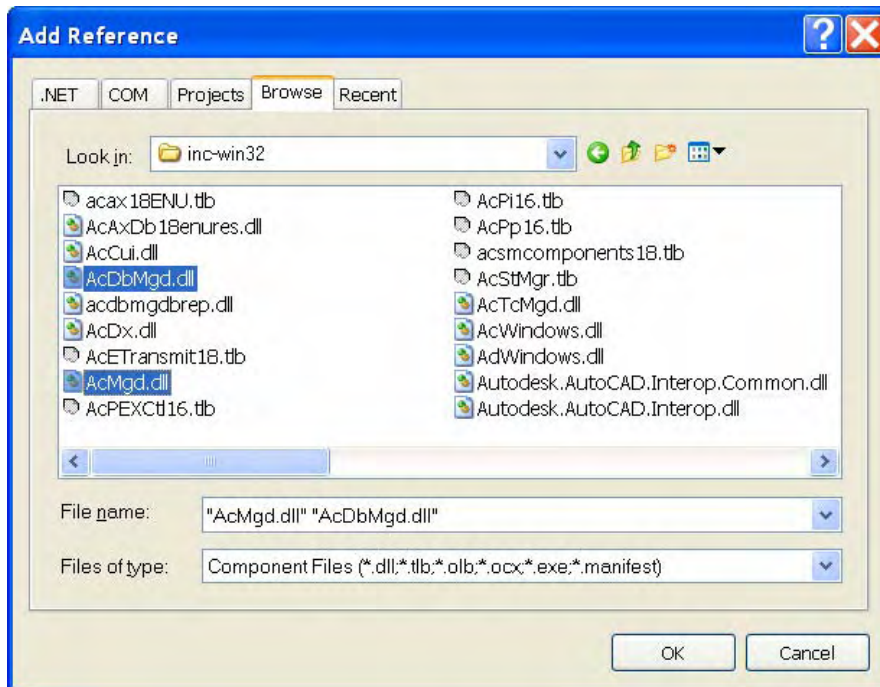
- 1) In your Visual Basic 2008 Express window, you should be able to see a "Solution Explorer" tab on the right. If you don't see it, click the "**View**→**Solution Explorer**" menu item or type the **Ctrl+R** keys.
- 2) Go to the **Solution Explorer** tab on the right side of the Visual Basic 2008 Express window select the "**Show All Files**" button as shown below



- 3) The above step will display the "**References**" folder in the Solution Explorer as shown below



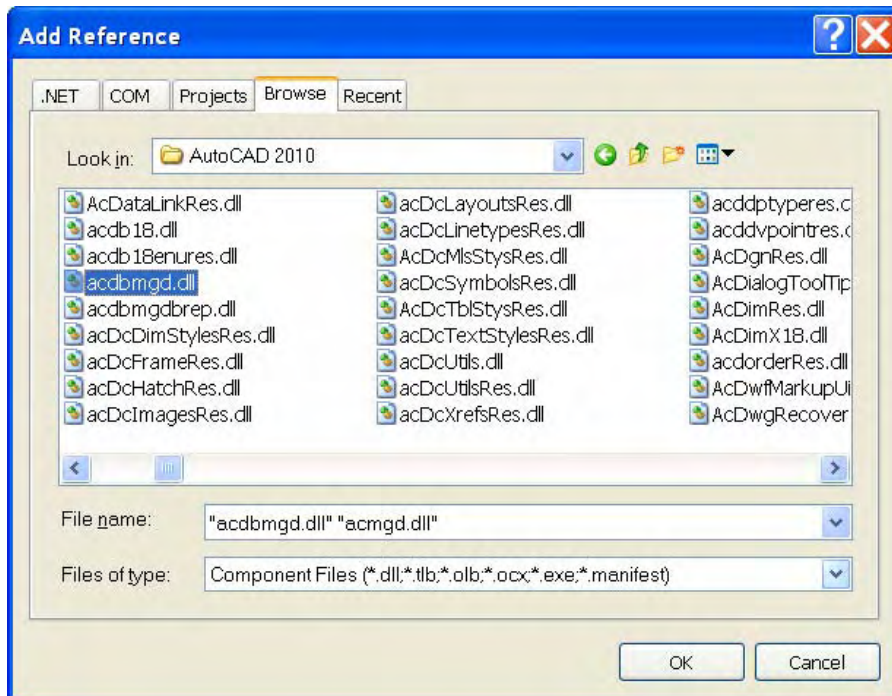
- 4) Right-click on this "References" folder and select "**Add Reference...**" item in the resulting context menu. An "**Add Reference**" dialog will be displayed:



- 5) Click on the “**Browse**” tab and Navigate to the location of the AutoCAD specific Reference files.

**Note:** You will find AutoCAD specific reference files in a couple of locations. If you have **ObjectARX SDK** installed, you can navigate to the Include folder (I would navigate to C:\ObjectARX\2010\inc-win32 for 32 bit platforms and C:\ObjectARX\2010\inc-x64) for the references. In the snapshot above, we navigated to the **inc-win32** folder.

Alternatively, you can navigate to the **AutoCAD 2010 Install Folder** (the default location is C:\Program Files\AutoCAD 2010) and find the references there as shown below:



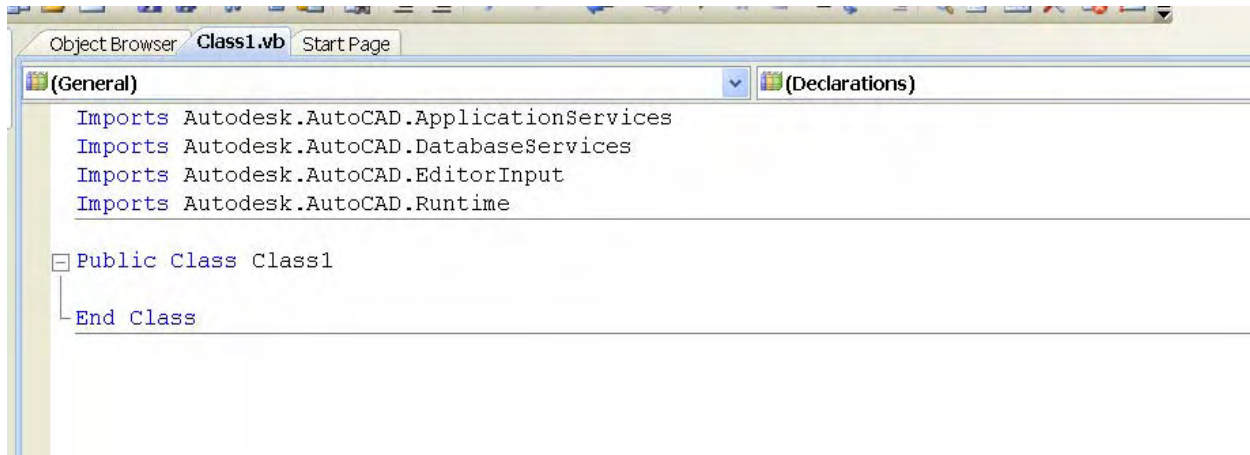
- 6) Select the DLL files **acmgsd.dll** and **acdbmgd.dll** from the location you browse to and click the **OK** button. These files contain all the necessary **.NET APIs** to create a generic AutoCAD .NET application.

**Note:** You can select multiple files (two files in this case) by pressing the “**Ctrl**” key while selecting them with the mouse.

- 7) Add the necessary **Namespaces** to the source files for convenient access to the AutoCAD .NET APIs. Here are the most commonly used AutoCAD .NET API specific namespaces:

```
Imports Autodesk.AutoCAD.ApplicationServices
Imports Autodesk.AutoCAD.DatabaseServices
Imports Autodesk.AutoCAD.EditorInput
Imports Autodesk.AutoCAD.Runtime
```

You can add these namespaces at the top of the “**Class1.vb**” source file as shown below.



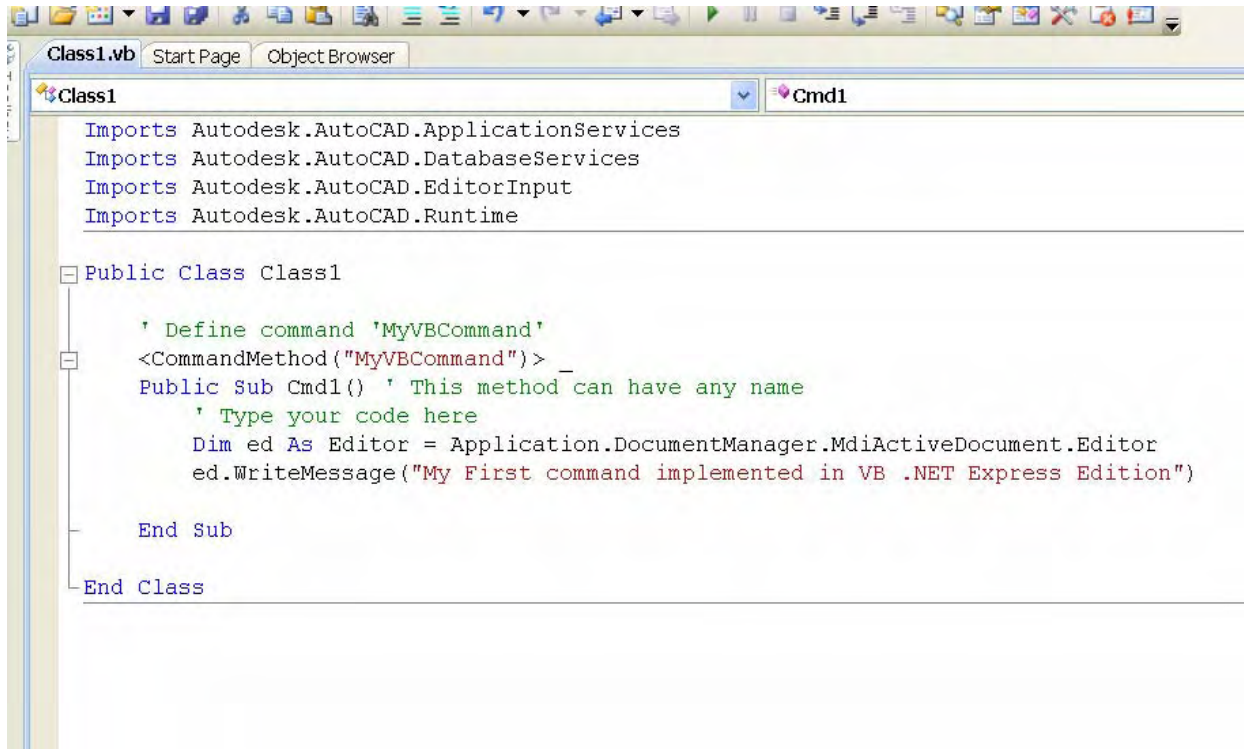
**Note:** We will not be elaborating on the purposes of the namespaces and the AutoCAD .NET APIs in this tutorial. You can find a lot more information on this in the references provided at the end of this tutorial.

### Creating a command:

Add the command method that implements the command(s) you need. Here is an example of the command method implementing the command we use:

```
' Define command 'MyVBCCommand'  
<CommandMethod("MyVBCCommand")> _  
Public Sub Cmd1() ' This method can have any name  
    ' Type your code here  
    Dim ed As Editor = Application.DocumentManager.MdiActiveDocument.Editor  
    ed.WriteMessage("My First command implemented in VB .NET Express Edition")  
  
End Sub
```

**Note:** We will not be elaborating on the creation of commands and the AutoCAD .NET APIs in this tutorial. You can find a lot more information on this in the references provided at the end of this tutorial. Here is a snapshot of how the command code would look like:



```
Imports Autodesk.AutoCAD.ApplicationServices
Imports Autodesk.AutoCAD.DatabaseServices
Imports Autodesk.AutoCAD.EditorInput
Imports Autodesk.AutoCAD.Runtime

Public Class Class1

    ' Define command 'MyVBCommand'
    <CommandMethod("MyVBCommand")> _
    Public Sub Cmd1() ' This method can have any name
        ' Type your code here
        Dim ed As Editor = Application.DocumentManager.MdiActiveDocument.Editor
        ed.WriteMessage("My First command implemented in VB .NET Express Edition")

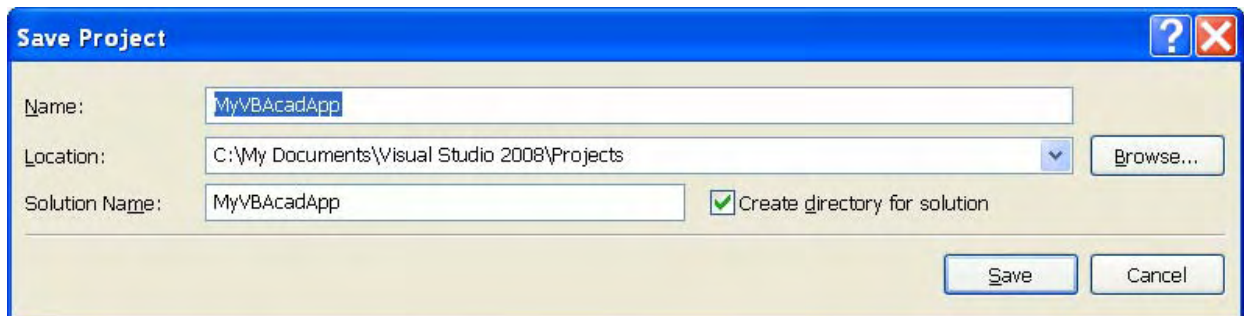
    End Sub

End Class
```

Now we have a functional AutoCAD VB .NET project!

### Save the Project:

You can save the project by clicking the "File→Save All" menu item or hitting the "Ctrl+Shift+S" keys. This will popup up the "Save Project" Dialog shown below.



You may choose the Name, Location and Solution Name of the project as you see fit.

### Build the Project:

The last step is to **Build** the application. Use the “**Build→Build <YourApplicationName>**” menu item to build your application. In the case of the sample we used, the menu item was “**Build→Build MyVBAcadApp**”.

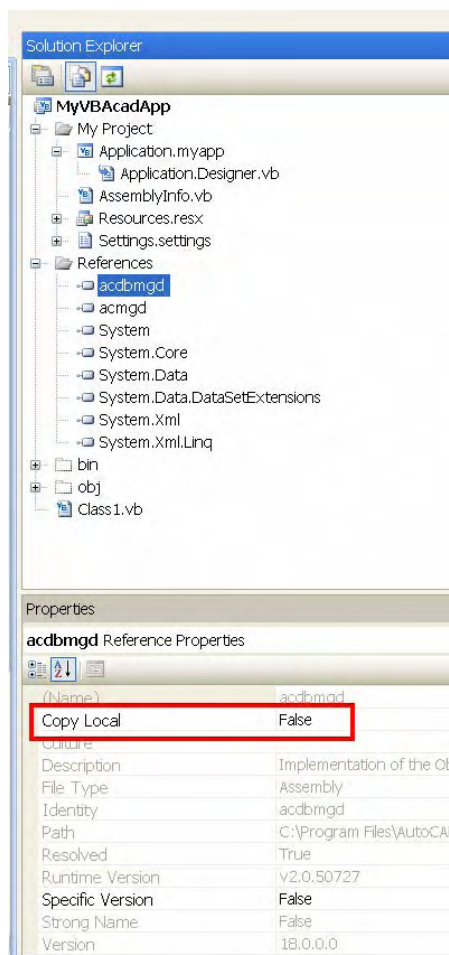
**You now have a functional AutoCAD VB .NET Application!**

### Debugging the Application:

You can load your compiled application into AutoCAD now using the NETLOAD command, but you’ll have to make a few additional changes to your project to be able to debug it when it’s running in AutoCAD.

### Set Copy Local property to False:

- 1) Open the Properties Tab by clicking the “**View→Properties Window**” menu item or the **F4** key.
- 2) Select the “**acdbmgd**” reference in the “References” folder in the “Solution Explorer” tab and set its “**Copy Local**” property in the **Properties** tab to “**False**” as shown below:





- 3) Do the above step for the "acmgd" reference as well

### Set the Debug Application Path:

The Visual Basic 2008 Express Edition does not provide any method through its user interface to specify a 'debug application path' to launch AutoCAD from the Visual Basic 2008 Express Edition environment. so we have to play a small trick here:

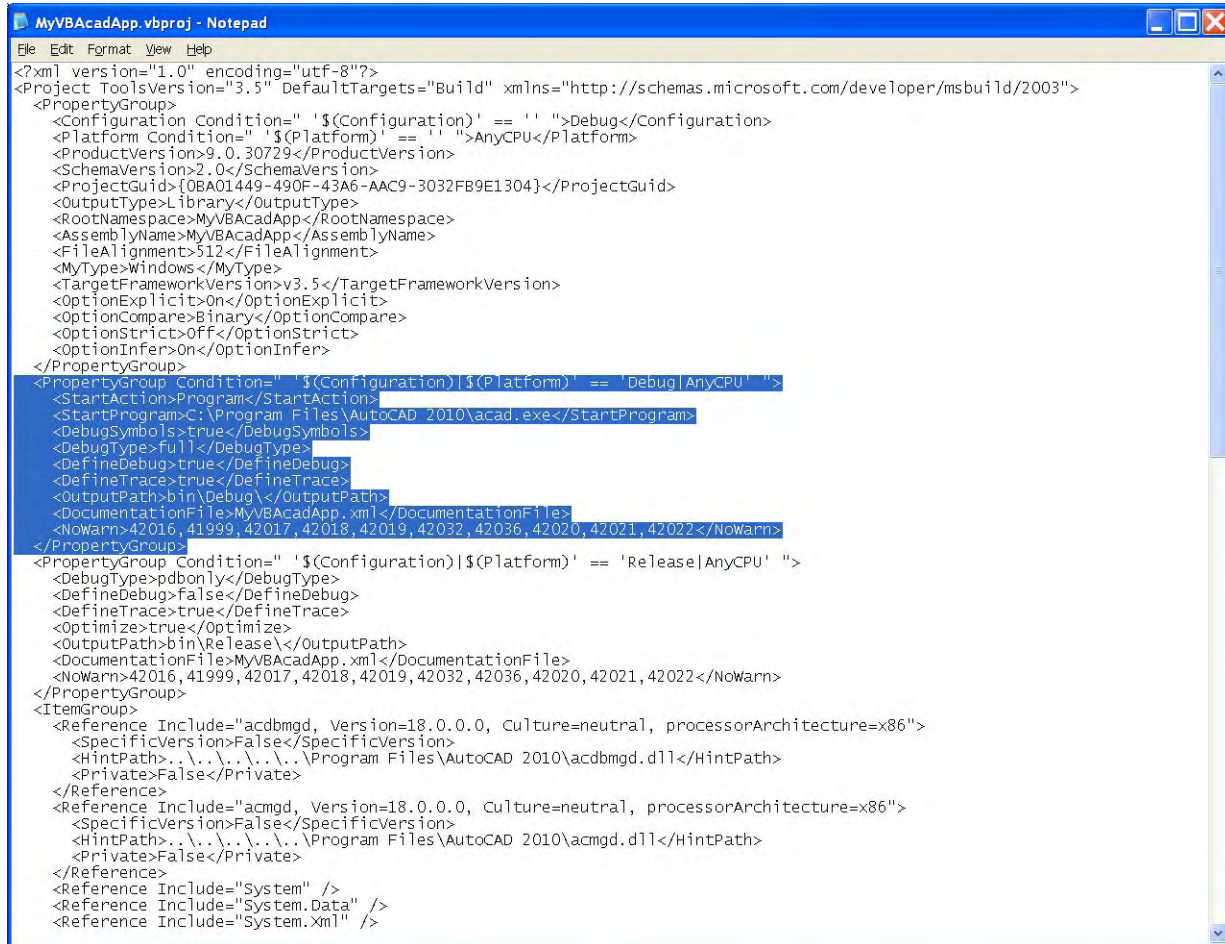
- 1) Save the project and close Visual Basic 2008 Express Edition
- 2) Navigate to the location of the project in **Windows Explorer**. My Project is in the folder C:\My Documents\Visual Studio 2008\Projects\MyVBACadApp\MyVBACadApp.
- 3) Open the Project file <ApplicationName.vbproj > in a text editor of your choice. I choose "Notepad" as the text editor and my application project file is named MyVBACadApp.vbproj. You will notice that it is an XML file. Under the XML Node "<PropertyGroup Condition=" '\$(Configuration) | \$(Platform)' == 'Debug | AnyCPU' ">" Add the following two XML Nodes:

```
<StartAction>Program</StartAction>  
<StartProgram>C:\Program Files\AutoCAD 2010\acad.exe</StartProgram>
```

Note: The "StartProgram" node stores the path of the AutoCAD application "acad.exe". You will have to modify this path to the correct path to the acad.exe executable on your machine. This section in my MyVBACadApp.vbproj file looks like this after I add my entries:

```
<PropertyGroup Condition=" '$(Configuration) | $(Platform)' == 'Debug | AnyCPU'  
">  
  <StartAction>Program</StartAction>  
  <StartProgram>C:\Program Files\AutoCAD 2010\acad.exe</StartProgram>  
  <DebugSymbols>>true</DebugSymbols>  
  <DebugType>full</DebugType>  
  <DefineDebug>>true</DefineDebug>  
  <DefineTrace>>true</DefineTrace>  
  <OutputPath>bin\Debug\</OutputPath>  
  <DocumentationFile>MyVBACadApp.xml</DocumentationFile>  
  
<NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoW  
arn>  
</PropertyGroup>
```

Here is a snapshot of my VBPROJ file opened in notepad:

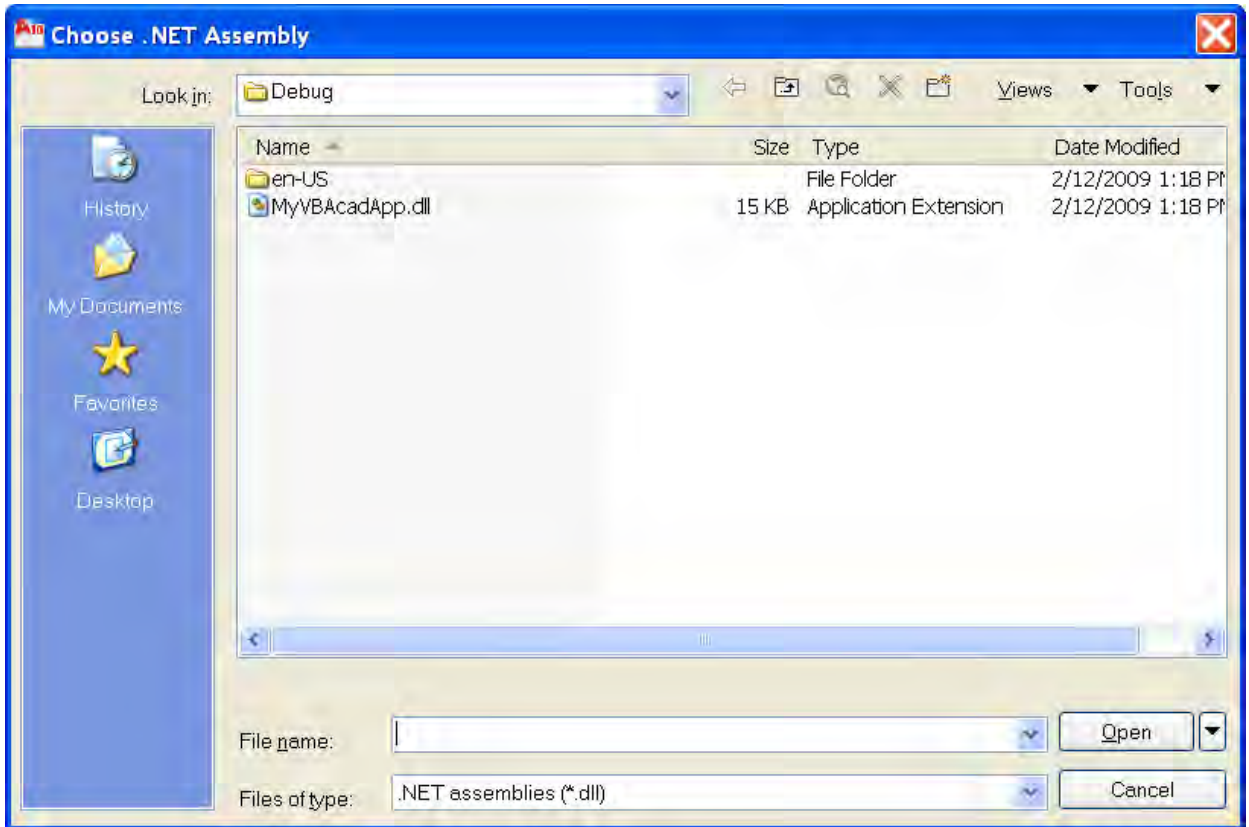


```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="3.5" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">AnyCPU</Platform>
    <ProductVersion>9.0.30729</ProductVersion>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{0BA01449-490F-43A6-AAC9-3032FB9E1304}</ProjectGuid>
    <OutputType>Library</OutputType>
    <RootNamespace>MyVBAcadApp</RootNamespace>
    <AssemblyName>MyVBAcadApp</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Windows</MyType>
    <TargetFrameworkVersion>v3.5</TargetFrameworkVersion>
    <OptionExplicit>On</OptionExplicit>
    <OptionCompare>Binary</OptionCompare>
    <OptionStrict>Off</OptionStrict>
    <OptionInfer>On</OptionInfer>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <StartAction>Program</StartAction>
    <StartProgram>C:\Program Files\AutoCAD 2010\acad.exe</StartProgram>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug</OutputPath>
    <DocumentationFile>MyVBAcadApp.xml</DocumentationFile>
    <NoWarn>42016, 41999, 42017, 42018, 42019, 42032, 42036, 42020, 42021, 42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
    <DefineDebug>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release</OutputPath>
    <DocumentationFile>MyVBAcadApp.xml</DocumentationFile>
    <NoWarn>42016, 41999, 42017, 42018, 42019, 42032, 42036, 42020, 42021, 42022</NoWarn>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="acdbmgd, Version=18.0.0.0, Culture=neutral, processorArchitecture=x86">
      <SpecificVersion>False</SpecificVersion>
      <HintPath>..\..\..\..\Program Files\AutoCAD 2010\acdbmgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="acmgd, Version=18.0.0.0, Culture=neutral, processorArchitecture=x86">
      <SpecificVersion>False</SpecificVersion>
      <HintPath>..\..\..\..\Program Files\AutoCAD 2010\acmgd.dll</HintPath>
      <Private>False</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
  </ItemGroup>
</Project>
```

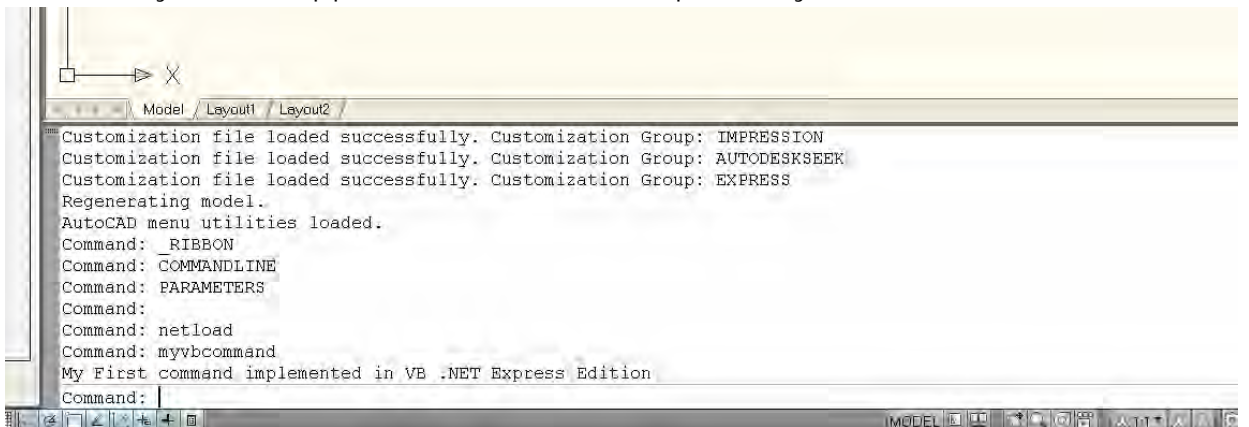
- 4) Save the VBPROJ file and open the project again in Visual Basic 2008 Express Edition.
- 5) Build the project again.
- 6) Select the “**Debug→Start Debugging**” Menu item or press **F5**. This will launch the AutoCAD application successfully from the Visual Basic 2008 Express Edition **debugging environment**.

### Loading our VB .NET Application into AutoCAD

- 1) Once AutoCAD is launched, run the **NETLOAD** command. This will display the “Choose .NET Assembly” dialog (as shown below):

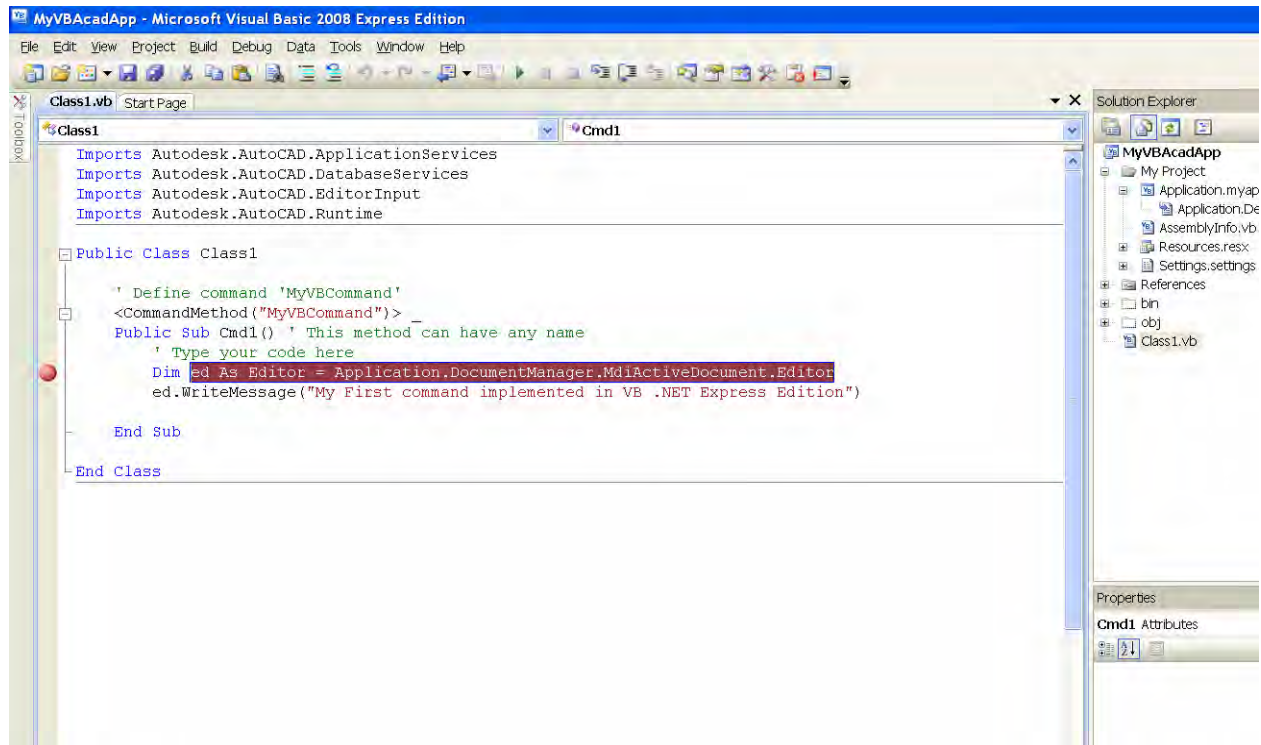


- 2) Navigate to the location of your built application. My application lived in the "C:\My Documents\Visual Studio 2008\Projects\MyVBAcadApp\MyVBAcadApp\bin\Debug" folder.
- 3) Select the application DLL and click "Open" button. The application is now loaded into AutoCAD
- 4) Run the command "MyVBCommand" or any other command name that you defined in your .NET application. Here is the output of my command:



Set a breakpoint in the code

- 1) Set the cursor at the line of code you want to break to.
- 2) Click the "**Debug**→**Toggle Breakpoint**" menu item or press the "**F9**" key. Here is a snapshot of the break point at the point of code where we set the Editor object (The Red dot indicates the breakpoint):



- 3) Run the command (in our case "MyVBCommand") on the AutoCAD command line and you will see the break point hit:

```
MyVBAcadApp (Debugging) - Microsoft Visual Basic 2008 Express Edition
File Edit View Project Build Debug Data Tools Window Help
Class1.vb
Class1
Imports Autodesk.AutoCAD.ApplicationServices
Imports Autodesk.AutoCAD.DatabaseServices
Imports Autodesk.AutoCAD.EditorInput
Imports Autodesk.AutoCAD.Runtime

Public Class Class1
    ' Define command 'MyVBCCommand'
    <CommandMethod("MyVBCCommand")>
    Public Sub Cmd1() ' This method can have any name
        ' Type your code here
        Dim ed As Editor = Application.DocumentManager.MdiActiveDocument.Editor
        ed.WriteMessage("My First command implemented in VB .NET Express Edition")
    End Sub
End Class
```

You can now use the debugger keys like F8 or Shift+F8 etc to traverse through the code

**Congratulations, you have successfully used Visual Basic 2008 Express Edition to create and use a .NET application in AutoCAD!**

#### References:

- 1) Developer Center for AutoCAD ([www.autodesk.com/developautocad](http://www.autodesk.com/developautocad)): This website provides a variety of learning resources such as “**DevTV: Introduction to AutoCAD .NET Programming**” which provided an excellent introduction to the AutoCAD .NET API and “**AutoCAD .NET Labs**” which provides a comprehensive self-learning tutorials. The Developer Center lists many other free development resources, including links to the Autodesk discussion groups for the various AutoCAD APIs.
- 2) “Through the Interface” Blog (<http://through-the-interface.typepad.com/>): A very interesting and engaging blog on all AutoCAD APIs and allied technologies
- 3) Autodesk Developer Network: For professional support for your software development work, consider joining the Autodesk Developer Network program. Visit [www.autodesk.com/joinadn](http://www.autodesk.com/joinadn) for more details of the program.