# Developing Shared Add-Ins with Microsoft Visual Studio 2008 Express

* [8 Comments](http://www.squidoo.com/access-shared-add-in#module94454261)
* [Like](http://www.squidoo.com/access-shared-add-in) 2

Ranked #11,777 in [Computers & Electronics](http://www.squidoo.com/topics/computers-and-electronics), #204,905 overall

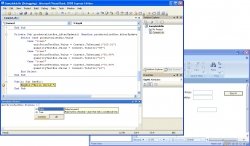
[Ads by Google](http://www.google.com/url?ct=abg&q=https://www.google.com/adsense/support/bin/request.py%3Fcontact%3Dabg_afc%26url%3Dhttp://www.squidoo.com/access-shared-add-in%26gl%3DCH%26hl%3Den%26client%3Dca-pub-9879162776784828%26ai0%3DCnhvhreLYUNmoIo22_Qb274CoA8uHtOME0_untl_AjbcBEAEgs6LFAygDUPyelqoCYPWVzoHgBKABreC62APIAQGpArSrKfjmU7I-qAMBqgRvT9BOkauuSYFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fIZk8lv5ttHPXr_DlQtsheilA0GBzCOEWEdZ1wXqsWyyfAUciAYBgAe7n8Un%26ai1%3DC-kxAreLYUNmoIo22_Qb274CoA_vegM4Ci7KFnzzAjbcBEAIgs6LFAygDUIq9mv8FYPWVzoHgBKABqd_a_gPIAQGpAgI9xrBZIrY-qAMBqgRoT9BuququSoFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fKZly7_ynefJbLkLY9ZpbetFA0GBzGNUTUKWCyCIBgGAB7-gpQE%26ai2%3DCZYi-reLYUNmoIo22_Qb274CoA6TM_84C_OyG3DbAjbcBEAMgs6LFAygDULj817n8_____wFg9ZXOgeAEyAEBqQK0qyn45lOyPqgDAaoEak_QHs5Bt1C6ZxRc27HopPW_zVDrkNQ7bWBujDRBTnFRlZxDCKfQCejg0x7yLY8LD39jNdKpQDUFu52vHOgAl5zmkql_ObYYjV0u9H9PjhJgH3Y6EoFSKLYgam3rhTa3gswjBL2kXwfUWlKAB9TA9gE&usg=AFQjCNG85R6MsRTuFlhOGUxc8MAKEm3O6Q)

[Office-Programme](http://www.googleadservices.com/pagead/aclk?sa=L&ai=CnhvhreLYUNmoIo22_Qb274CoA8uHtOME0_untl_AjbcBEAEgs6LFAygDUPyelqoCYPWVzoHgBKABreC62APIAQGpArSrKfjmU7I-qAMBqgRvT9BOkauuSYFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fIZk8lv5ttHPXr_DlQtsheilA0GBzCOEWEdZ1wXqsWyyfAUciAYBgAe7n8Un&num=1&cid=5Gi-lwl0hE5jEIrZc-yeM8mI&sig=AOD64_1IwBf5NVRB6kupevcjtXMMFZJFSA&client=ca-pub-9879162776784828&adurl=http://www.klubschule.ch/Angebote/Informatik-Neue-Medien/Office-Anwendungen/level%401N11621--suche%40ECDL--) Microsoft Office für alle Niveaus: Jetzt bei der Klubschule anmelden! [Klubschule.ch/Office](http://www.googleadservices.com/pagead/aclk?sa=L&ai=CnhvhreLYUNmoIo22_Qb274CoA8uHtOME0_untl_AjbcBEAEgs6LFAygDUPyelqoCYPWVzoHgBKABreC62APIAQGpArSrKfjmU7I-qAMBqgRvT9BOkauuSYFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fIZk8lv5ttHPXr_DlQtsheilA0GBzCOEWEdZ1wXqsWyyfAUciAYBgAe7n8Un&num=1&cid=5Gi-lwl0hE5jEIrZc-yeM8mI&sig=AOD64_1IwBf5NVRB6kupevcjtXMMFZJFSA&client=ca-pub-9879162776784828&adurl=http://www.klubschule.ch/Angebote/Informatik-Neue-Medien/Office-Anwendungen/level%401N11621--suche%40ECDL--)

[Neu: Visual Studio 2012](http://www.googleadservices.com/pagead/aclk?sa=L&ai=C-kxAreLYUNmoIo22_Qb274CoA_vegM4Ci7KFnzzAjbcBEAIgs6LFAygDUIq9mv8FYPWVzoHgBKABqd_a_gPIAQGpAgI9xrBZIrY-qAMBqgRoT9BuququSoFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fKZly7_ynefJbLkLY9ZpbetFA0GBzGNUTUKWCyCIBgGAB7-gpQE&num=2&cid=5Gi-lwl0hE5jEIrZc-yeM8mI&sig=AOD64_24C73sG4VsNR0uEv8TJRmShAwKIw&client=ca-pub-9879162776784828&adurl=http://www.sos-software.com/de/product_info.php%3Fpname%3DVisual%2520Studio) Jetzt die vielen Highlights nutzen und inkl. MSDN sichern. [www.sos-software.com](http://www.googleadservices.com/pagead/aclk?sa=L&ai=C-kxAreLYUNmoIo22_Qb274CoA_vegM4Ci7KFnzzAjbcBEAIgs6LFAygDUIq9mv8FYPWVzoHgBKABqd_a_gPIAQGpAgI9xrBZIrY-qAMBqgRoT9BuququSoFfpu_VyeikrOLPVPed3zssLWKOg5pkMk2VmAkdp4gJ6LmOHPYxggAPPi450OsONwW9i69Cthuej_GT4DM8tlHJWsB2fKZly7_ynefJbLkLY9ZpbetFA0GBzGNUTUKWCyCIBgGAB7-gpQE&num=2&cid=5Gi-lwl0hE5jEIrZc-yeM8mI&sig=AOD64_24C73sG4VsNR0uEv8TJRmShAwKIw&client=ca-pub-9879162776784828&adurl=http://www.sos-software.com/de/product_info.php%3Fpname%3DVisual%2520Studio)

[Qlikview erleben bei uns](http://googleads.g.doubleclick.net/aclk?sa=L&ai=CZYi-reLYUNmoIo22_Qb274CoA6TM_84C_OyG3DbAjbcBEAMgs6LFAygDULj817n8_____wFg9ZXOgeAEyAEBqQK0qyn45lOyPqgDAaoEak_QHs5Bt1C6ZxRc27HopPW_zVDrkNQ7bWBujDRBTnFRlZxDCKfQCejg0x7yLY8LD39jNdKpQDUFu52vHOgAl5zmkql_ObYYjV0u9H9PjhJgH3Y6EoFSKLYgam3rhTa3gswjBL2kXwfUWlKAB9TA9gE&num=3&sig=AOD64_3O8ap8ifdQoanSLP2cMqZVAbSmCw&client=ca-pub-9879162776784828&adurl=http://www.qlikview-praxis-workshop.ch) mit Ihren Daten - kostenlos 1 Tag Schulung + Applikationsaufbau [www.qlikview-praxis-workshop.ch](http://googleads.g.doubleclick.net/aclk?sa=L&ai=CZYi-reLYUNmoIo22_Qb274CoA6TM_84C_OyG3DbAjbcBEAMgs6LFAygDULj817n8_____wFg9ZXOgeAEyAEBqQK0qyn45lOyPqgDAaoEak_QHs5Bt1C6ZxRc27HopPW_zVDrkNQ7bWBujDRBTnFRlZxDCKfQCejg0x7yLY8LD39jNdKpQDUFu52vHOgAl5zmkql_ObYYjV0u9H9PjhJgH3Y6EoFSKLYgam3rhTa3gswjBL2kXwfUWlKAB9TA9gE&num=3&sig=AOD64_3O8ap8ifdQoanSLP2cMqZVAbSmCw&client=ca-pub-9879162776784828&adurl=http://www.qlikview-praxis-workshop.ch)

## Introduction



In an [MSDN article](http://msdn.microsoft.com/en-us/library/aa663367.aspx), Microsoft says, "The Visual Studio 2005 Express Editions do not contain the Shared Add-In template. However, you can write a shared add-in, it is just a bit more work to get going." However, what they fail to tell us is exactly what is required to write a shared add-in in the Express Edition.  
  
This guide fills in that missing piece of the puzzle, and will help get you started with a fully integrated development IDE for Microsoft Office Shared Add-Ins in the Express edition - including full use of the Visual Studio debugger.  
  
**NOTE:** This guide is geared strongly toward developing Shared Add-Ins for Microsoft Access 2007, however, with some small source-code changes, you can use the same template to write Shared Add-Ins for any Microsoft Office application.

### Contents at a Glance

1. [Step 1 - Creating The Template](http://www.squidoo.com/access-shared-add-in#module94289871)
2. [Step 2 - Export The Template](http://www.squidoo.com/access-shared-add-in#module94399781)
3. [Step 3 - Configuring The External Tools](http://www.squidoo.com/access-shared-add-in#module94400091)
4. [Using The Template](http://www.squidoo.com/access-shared-add-in#module94404441)
5. [Getting You Started With a Sample Add-In](http://www.squidoo.com/access-shared-add-in#module94440621)
6. [Amazon](http://www.squidoo.com/access-shared-add-in#module151866311)

[More](http://www.squidoo.com/access-shared-add-in)

## Step 1 - Creating The Template

### Since the Express Edition does not contain the Shared Add-In template, we'll have to create one manually. Here's how:

Start **Visual Basic**, and create a new **Class Library** project, named "**AccessAddInTemplate**."  
  
From the **Project** menu, click on **Add Reference**, select **extensibility** from the **.NET** tab, and click on **OK**.  
  
Open the **Add Reference** dialog box again, select **Microsoft Access 12.0 Object Library** from the **COM** tab, and click on **OK**.  
  
In the **Solution Explorer**, rename **Class1.vb** to **Connect.vb**.  
  
Erase all of the existing source code from **Connect.vb**, and replace it with the following source code:

#Region "Imports"  
Imports Microsoft.Office  
Imports Microsoft.Office.Core  
Imports Microsoft.Office.Interop  
Imports Extensibility  
Imports System.Runtime.InteropServices  
Imports System.Reflection  
Imports Access = Microsoft.Office.Interop.Access  
#End Region  
  
<GuidAttribute("82359690-b3c7-4c6c-b5eb-d075f270bcb8"), \_  
ProgIdAttribute("AccessAddInTemplate.Connect")> \_  
Public Class Connect  
    Implements Extensibility.IDTExtensibility2  
    Private applicationObject As Access.Application  
    Private addInInstance As Microsoft.Office.Core.COMAddIn  
  
    Public Sub OnBeginShutdown( \_  
      ByRef custom As System.Array) \_  
      Implements Extensibility.IDTExtensibility2.OnBeginShutdown  
    End Sub  
  
    Public Sub OnAddInsUpdate( \_  
      ByRef custom As System.Array) \_  
      Implements Extensibility.IDTExtensibility2.OnAddInsUpdate  
    End Sub  
  
    Public Sub OnStartupComplete( \_  
      ByRef custom As System.Array) \_  
      Implements Extensibility.IDTExtensibility2.OnStartupComplete  
    End Sub  
  
    Public Sub OnDisconnection( \_  
      ByVal RemoveMode As Extensibility.ext\_DisconnectMode, \_  
      ByRef custom As System.Array) \_  
      Implements Extensibility.IDTExtensibility2.OnDisconnection  
        GC.Collect()  
        GC.WaitForPendingFinalizers()  
    End Sub  
  
    Public Sub OnConnection( \_  
      ByVal application As Object, \_  
      ByVal connectMode As Extensibility.ext\_ConnectMode, \_  
      ByVal addInInst As Object, ByRef custom As System.Array) \_  
      Implements Extensibility.IDTExtensibility2.OnConnection  
        applicationObject = CType(application, Access.Application)  
        addInInstance = CType(addInInst, Microsoft.Office.Core.COMAddIn)  
        addInInstance.Object = Me  
    End Sub  
End Class

From the **Project** menu, click on **AccessAddInTemplate Properties**.  
  
Click on the **Compile** tab, and change the **Build output path** to **bin\Debug\**.  
  
Click on the **Application** tab, and click on **Assembly Information**. Check the **Make COM Visible** check box, and copy the **GUID** into the clipboard.  
  
In the code window, replace the value in **GuidAttribute** found after the "**Imports**" region with the **GUID** you copied from the **Assembly Information** dialog box.  
  
Save your project, and close **Visual Basic**.  
  
In Windows Explorer, navigate to the folder where you saved your project, and open the file **.\AccessAddInTemplate\AccessAddInTemplate.vbproj** with Notepad, or an XML editor of your choice.  
  
We need to add some additional information under these two nodes:  
  
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">  
and  
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">  
  
Paste the following lines of code directly under both of those nodes:

    <StartAction>Program</StartAction>   
    <StartProgram>c:\Program Files\Microsoft Office\Office12\MSACCESS.EXE</StartProgram>   
    <StartArguments />

Save the file, and close the editor.

## Step 2 - Export The Template

### Now we need to make our template usable from the New Project dialog box. Here's how:

Start **Visual Basic** and open the **AccessAddInTemplate** project.  
  
From the **File** menu, click on **Export Template**.  
  
Make sure **Project Template** is selected, and the **AccessAddInTemplate** project selected, and click on **Next**.  
  
Change the **template name** to "**MS Access Shared Add-In**", and change the **description** to "**A project for creating a managed Shared Add-In for Microsoft Office Access 2007**."  
  
Check the box "**Automatically import the template into Visual Studio**," and click on **Finish**.

## Step 3 - Configuring The External Tools

### We are going to need create additional tools to simplify the use of our template. Here's how:

From the **Tools** menu, click on **External Tools**.  
  
Click on **Add**.  
  
Enter **Create GUID** in the **Title** text box.  
  
Enter **C:\Program Files\Microsoft Visual Studio 9.0\Common7\Tools\guidgen.exe** in the **Command** text box.  
  
Click on **Add**.  
  
Enter **Register Add-In** in the **Title** text box.  
  
Enter **C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\regasm.exe** in the **Command** text box.  
  
Enter **/codebase $(BinDir)$(TargetName).dll** in the **Arguments** text box.  
  
Check the **Use Output window** check box.  
  
Click on **Add**.  
  
Enter **Unregister Add-In** in the **Title** text box.  
  
Enter **C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\regasm.exe** in the **Command** text box.  
  
Enter **/unregister $(BinDir)$(TargetName).dll** in the **Arguments** text box.  
  
Check the **Use Output window** check box.  
  
Click on **Add**.  
  
Enter **Create Add-In .reg File** in the **Title** text box.  
  
Enter **C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\regasm.exe** in the **Command** text box.  
  
Enter **/regfile:$(BinDir)$(TargetName).reg /codebase $(BinDir)$(TargetName).dll** in the **Arguments** text box.  
  
Check the **Use Output window** check box.  
  
Click on **OK**  
  
You should now see four additional menu items in the **Tools** menu: **Create GUID**, **Register Add-In**, and **Unregister Add-In**, and **Create Add-In .reg File**.

## Using The Template

### Here's how to create, and debug a new Add-In using the Shared Add-In template:

**Overview:** The steps you will need to take are creating your new project, assigning it a new GUID, registering the assembly, and installing the Add-In into Microsoft Access via the system registry.  
  
From the **File** menu click on **New Project**.  
  
Select the template from the **My Templates** section, and name it and click on **OK**.  
  
From the **Tools** menu, click on **Create GUID**.  
  
Select option **4** from the **GUID Format** list, click on **New GUID**, and then click on **Copy** and close the **Create GUID** window.  
  
Just after the **Imports** region in the **Connect.vb** source code, replace GUID in the **GuidAttribute** with the one in the clipboard. Make sure to erase the { brackets } surrounding the GUID.  
  
From the **Project** menu, open your project's properties.  
  
Click on the **Application** tab.  
  
Click on the **Assembly Information** button, and paste the GUID into the appropriate text box. Once again, make sure to erase the { brackets } surrounding the GUID.  
  
Save and build the project.  
  
From the **Tools** menu, click on **Register Add-In**.  
  
Open the **Registry Editor** - regedit - and navigate to **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Office\Access\Addins**  
  
Create a new **Key** with the same name of your project followed by **.Connect**. For example, if your project name is **MyAccessAddIn** then the key you want to create is **MyAccessAddIn.Connect**.  
  
Under your new key, create a new **String** value named **FriendlyName** and change its value to what you want the end-user to see for the Add-In name.  
  
Create a new **DWORD** value named **LoadBehavior** and choose its value from one of these options:

* **0** = Disconnect - Is not loaded.
* **1** = Connected - Is loaded.
* **2** = Bootload - Load on application startup.
* **8** = DemandLoad - Load only when requested by user.
* **16** = ConnectFirstTime - Load only once (on next startup).

**NOTE:** You can combine those options by adding the values together. For example, the typical **LoadBehavior** is **3**, which is Connected + Bootload. This would load the Add-In when Access starts - in other words, your projects **OnConnection** procedure would be called when the user opens Microsoft Access.  
  
Close the **Registry Editor**, and return to **Visual Basic**.  
  
Press **F5**, or click on the **run** button. **Microsoft Access** should open up, and **Visual Studio** will switch to debug mode.  
  
Close **Microsoft Access**, and you will return to **Visual Studio** in edit mode.  
  
**NOTE:** To have **Microsoft Access** open up using a specific database, change the **Command line arguments** option in the **Debug** tab of your project's properties to the full path of your database file. Make sure to encapsulate the file name in quotes.

## Getting You Started With a Sample Add-In

### Okay, I have my template, I've created a new Shared Add-In... Now what?

Let's make a very simple sample Shared Add-In for Microsoft Access to see how all of this is put together.  
  
Start by following the steps in the **Using The Template** section to create a project named **SampleAddIn**.  
  
Press **F5** to open **Microsoft Access**, and create a new database in your project's directory named **Test.accdb**.  
  
From the **Create** tab in Access, click on **Form Design**.  
  
Drop a button, named **Button1** and a text box named **Text1** on the form, and save the form with its default name, **Form1**.  
  
Close **Microsoft Access**.  
  
In **Visual Studio**, open the **Properties** of your project, and enter the full path of your new database, including its file name, surrounded by quotes into the **Command line arguments** text box on the **Debug** tab.  
  
In **Connect.vb**, below the line that reads:

    Private addInInstance As Microsoft.Office.Core.COMAddIn

paste these lines of code:

    Private WithEvents Button1 As Access.CommandButton  
    Private Text1 As Access.TextBox  
  
    Public Sub HookupControls(ByVal button As Access.CommandButton, ByVal textbox As Access.TextBox)  
        Button1 = button  
        Button1.OnClick = "[Event Procedure]"  
        Text1 = textbox  
    End Sub  
  
    Private Sub Button1\_OnClick() Handles Button1.Click  
        Text1.Value = "Hello World"  
    End Sub

At the end of the **OnDisconnection** procedure, after the line that reads:

        GC.WaitForPendingFinalizers()

paste these two lines of code:

        Marshal.FinalReleaseComObject(Button1)  
        Marshal.FinalReleaseComObject(Text1)

Build and run your project.  
  
In **Microsoft Access**, open **Form1** in **Design View**.  
  
Add a new **Event Procedure** for the Form's **On Load** event, and add to it these lines of code:

    With COMAddIns("SampleAddIn.Connect")  
        .Connect = True  
        .Object.HookupControls Button1, Text1  
    End With

Open the form in **Form View**, and click on the button. The text box should populate with the text "Hello World"  
  
Close **Microsoft Access** to return to **Visual Studio**.

-----------------------------------------------------------

|  |  |  |  |
| --- | --- | --- | --- |
| I have asked this elsewhere, but have never found anyone knows how to build an add-in for VBA IDE using VB.NET. Is it even possible? Could someone point me to an example?  [vb.net](http://stackoverflow.com/questions/tagged/vb.net) [vba](http://stackoverflow.com/questions/tagged/vba) [add-in](http://stackoverflow.com/questions/tagged/add-in) [vbe](http://stackoverflow.com/questions/tagged/vbe)   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/q/1942626)|[improve this question](http://stackoverflow.com/posts/1942626/edit) | edited [Dec 23 '09 at 16:32](http://stackoverflow.com/posts/1942626/revisions) | asked Dec 21 '09 at 21:04  [[http://www.gravatar.com/avatar/74c236638d43b713794e529cd59e9aa1?s=32&d=identicon&r=PG](http://stackoverflow.com/users/102270/oorang)](http://stackoverflow.com/users/102270/oorang)  [Oorang](http://stackoverflow.com/users/102270/oorang) 2,6901429  88% accept rate | |
|  | feedback |

**4 Answers**

[active](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net?answertab=active#tab-top) [oldest](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net?answertab=oldest#tab-top) [votes](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net?answertab=votes#tab-top)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| up vote 5 down vote accepted | It is possible you need to write a com addin using IDTExtensibility2 interface, select the shared addin project template from new project.  **EDIT**  Otherwise to create this addin from scratch you will need to do the following:   1. Create a new project class library 2. Add references to "Extensibility", it should be in the list. You may need to download the PIAs for your version of office. (and perhaps VSTO but i am unsure on this point) 3. Add references to "Microsoft.Vbe.Interop" again should be with the PIAs. 4. Check the box "Register for Com Interop" in the properties tab. 5. **OPTIONAL** In the debug settings tab change the startup to external program and enter the path to the excel exe in the programfiles folder (if this is intended for excel) this is to allow the project to be debuggable. 6. **OPTIONAL** In the command options add a entry to a worksheet, or word doc that will show the addin dialog using a macro on startup, for development this makes sense to streamline the debugging experience. eg "C:\vbe.xlsm" 7. **OPTIONAL** Also set the startup path to the worksheet directory eg "C:\" 8. Implement the interface "IDTExtensibility2" found in "Extensibility" assembly. 9. Call this class "Connect" (this is just a preference) 10. Attribute the class with the following   [ComVisible(true), Guid("YourGeneratedGuid"), ProgId("YourAddinName.Connect")]  Heres an implementation to get you started, Firstly replace the "YourAddinName" with your AppName and Create a Guid for "YourGeneratedGuid". You will need to register the Addin into the right Registry location, see the registry keys that follow to get an idea, also replace some vars in the registry keys.  Imports System  Imports System.Drawing  Imports System.Linq  Imports System.Runtime.InteropServices  Imports Extensibility  Imports Microsoft.Vbe.Interop  Namespace VBEAddin  ''' <summary>  ''' The object for implementing an Add-in.  ''' </summary>  ''' <seealso class='IDTExtensibility2' />  <Guid("YourGeneratedGuid"), ProgId("YourAddinName.Connect")> \_  Public Class Connect  Implements IDTExtensibility2  Private \_application As VBE 'Interop VBE application object  #Region "IDTExtensibility2 Members"  ''' <summary>  ''' Implements the OnConnection method of the IDTExtensibility2 interface.  ''' Receives notification that the Add-in is being loaded.  ''' </summary>  ''' <param term='application'>  ''' Root object of the host application.  ''' </param>  ''' <param term='connectMode'>  ''' Describes how the Add-in is being loaded.  ''' </param>  ''' <param term='addInInst'>  ''' Object representing this Add-in.  ''' </param>  ''' <seealso class='IDTExtensibility2' />  Public Sub OnConnection(ByVal application As Object, ByVal connectMode As ext\_ConnectMode, ByVal addInInst As Object, ByRef [custom] As Array)  \_application = CType(Application,VBE)  End Sub  Private Sub onReferenceItemAdded(ByVal reference As Reference)  'TODO: Map types found in assembly using reference.  End Sub  Private Sub onReferenceItemRemoved(ByVal reference As Reference)  'TODO: Remove types found in assembly using reference.  End Sub  Private Sub BootAddin()  'Detect change in active window.  End Sub  ''' <summary>  ''' Implements the OnDisconnection method of the IDTExtensibility2 interface.  ''' Receives notification that the Add-in is being unloaded.  ''' </summary>  ''' <param term='disconnectMode'>  ''' Describes how the Add-in is being unloaded.  ''' </param>  ''' <param term='custom'>  ''' Array of parameters that are host application specific.  ''' </param>  ''' <seealso class='IDTExtensibility2' />  Public Sub OnDisconnection(ByVal disconnectMode As ext\_DisconnectMode, ByRef [custom] As Array)  End Sub  ''' <summary>  ''' Implements the OnAddInsUpdate method of the IDTExtensibility2 interface.  ''' Receives notification that the collection of Add-ins has changed.  ''' </summary>  ''' <param term='custom'>  ''' Array of parameters that are host application specific.  ''' </param>  ''' <seealso class='IDTExtensibility2' />  Public Sub OnAddInsUpdate(ByRef [custom] As Array)  End Sub  ''' <summary>  ''' Implements the OnStartupComplete method of the IDTExtensibility2 interface.  ''' Receives notification that the host application has completed loading.  ''' </summary>  ''' <param term='custom'>  ''' Array of parameters that are host application specific.  ''' </param>  ''' <seealso class='IDTExtensibility2' />  Public Sub OnStartupComplete(ByRef [custom] As Array)  'Boot dispatcher  End Sub  ''' <summary>  ''' Implements the OnBeginShutdown method of the IDTExtensibility2 interface.  ''' Receives notification that the host application is being unloaded.  ''' </summary>  ''' <param term='custom'>  ''' Array of parameters that are host application specific.  ''' </param>  ''' <seealso class='IDTExtensibility2' />  Public Sub OnBeginShutdown(ByRef [custom] As Array)  End Sub  #End Region  End Class  End Namespace  Here is the script to register the Addin, note you will need to change some of the settings in order to register it properly.  Windows Registry Editor Version 5.00  [HKEY\_CURRENT\_USER\Software\Microsoft\VBA\VBE\6.0\Addins\YourAddinName.Connect]  "CommandLineSafe"=dword:00000000  "Description"="Description for your new addin"  "LoadBehavior"=dword:00000000  "FriendlyName"="YourAddinName"  [HKEY\_CLASSES\_ROOT\CLSID\{YourGeneratedGuid}]  @="YourAddinName.Connect"  [HKEY\_CLASSES\_ROOT\CLSID\{YourGeneratedGuid}\Implemented Categories]  [HKEY\_CLASSES\_ROOT\CLSID\{YourGeneratedGuid}\InprocServer32]  @="mscoree.dll"  "ThreadingModel"="Both"  "Class"="YourAddinName.Connect"  "Assembly"="YourAssemblyNameFullTypeName"  "RuntimeVersion"="v2.0.50727"  "CodeBase"="file:///PathToAssembly"  [HKEY\_CLASSES\_ROOT\CLSID\{YourGeneratedGuid}\ProgId]  @="YourAddinName.Connect"  **NOTE** the tokens "YourGeneratedGuid" must have the curly braces {} included and be the same as the Guid in the attrib above, the token "YourAssemblyNameFullTypeName" must be the Assembly full name, the token "YourAddinName.Connect" must be the same ProgId set in the attrib above.  **SIDE NOTE**  Also found this helpful, might save you couple hours googling.  'HKEY\_CURRENT\_USER\Software\Microsoft\VBA\6.0\Common  'FontFace=Courier New (STRING - Default if missing)  'FontHeight=10 (DWORD - Default if missing)   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/1942651)|[improve this answer](http://stackoverflow.com/posts/1942651/edit) | edited [Dec 29 '09 at 8:50](http://stackoverflow.com/posts/1942651/revisions) | answered Dec 21 '09 at 21:11  [[http://www.gravatar.com/avatar/b742ed82f89ec89f88f2bccc022b4e18?s=32&d=identicon&r=PG](http://stackoverflow.com/users/104424/almog-ori)](http://stackoverflow.com/users/104424/almog-ori)  [almog.ori](http://stackoverflow.com/users/104424/almog-ori) 3,8251227 | | |
|  | |  |  | | --- | --- | |  | Hi Ori, the problem I am confronting is that I am using the Express version of VB.Net. One of the limitations of the Express version is a reduced template set. I'd imagine you can still do it, you just need to create one from scratch. Any ideas where to start on that? I checked out the MZ-Tools resources but their info doesn't apply to the VBA IDE from .Net. Which is the problem I keep hitting everywhere. All the info I can find is re:VB6. – [Oorang](http://stackoverflow.com/users/102270/oorang) [Dec 22 '09 at 1:27](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net#comment1854225_1942651) | |  | Wow, that's quite a lot. +1 for effort. I need time to try it out, if works, I'll accept:) Thanks:) – [Oorang](http://stackoverflow.com/users/102270/oorang) [Dec 22 '09 at 22:57](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net#comment1860950_1942651) | |  | Hi ORI, sorry for the delay. I'm painfully new at this, so please bear with me. I started an attempt to implement these instructions and the first road-block I hit was that (using the 2010 express version of VB) there is not a "shared addin project template" available (nor in the online collection). I know it's in the full version of VS, so I assume it's just missing from the Express addition. Do you know of a workaround to this problem? Sorry to complicate things. – [Oorang](http://stackoverflow.com/users/102270/oorang) [Dec 29 '09 at 1:21](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net#comment1887003_1942651) | |  | See my expanded answer – [almog.ori](http://stackoverflow.com/users/104424/almog-ori) [Dec 29 '09 at 8:52](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net#comment1888176_1942651) |   feedback | |
| up vote 0 down vote | | I would imagine you could call a .NET DLL from your VBA code (never actually done this myself). Just create a VB Class Library project and create a DLL to use in your VBA.  After a quick google it looks like you would need to set "Register for Com Interop" = True under Project Properties->Build, but like I said, I've never actually tried this before.   |  |  |  | | --- | --- | --- | | [share](http://stackoverflow.com/a/1942713)|[improve this answer](http://stackoverflow.com/posts/1942713/edit) | edited [Dec 22 '09 at 9:08](http://stackoverflow.com/posts/1942713/revisions)  [[http://www.gravatar.com/avatar/18e42b0ddc143340cfb072987f45ca50?s=32&d=identicon&r=PG](http://stackoverflow.com/users/63550/peter-mortensen)](http://stackoverflow.com/users/63550/peter-mortensen)  [Peter Mortensen](http://stackoverflow.com/users/63550/peter-mortensen) 5,55673062 | answered Dec 21 '09 at 21:24  [[http://www.gravatar.com/avatar/9cf92702c4843d9006fb9eb1460ba384?s=32&d=identicon&r=PG](http://stackoverflow.com/users/174499/smoore)](http://stackoverflow.com/users/174499/smoore)  [smoore](http://stackoverflow.com/users/174499/smoore) 1,867720 | |
|  | | |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | | 1 |  | | I think hes looking for a Addin to the development enviroment itself – [almog.ori](http://stackoverflow.com/users/104424/almog-ori) [Dec 21 '09 at 21:27](http://stackoverflow.com/questions/1942626/build-add-in-for-vba-ide-using-vb-net#comment1853063_1942713) |   feedback |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote 0 down vote | | I also found this reference to be helpful in making a VBA DLL from C# or VB.NET:   1. Create a new C# (or VB.Net) project and select Class Library as the template type. 2. using System; 3. using System.Collections.Generic; 4. using System.Linq; 5. using System.Text; 6. namespace SimpleCalc 7. { 8. public class Calc 9. { 10. private int numberOne = 0; 11. private int numberTwo = 0; 12. public void SetNumberOne(int number) 13. { 14. numberOne = number; 15. } 16. public void SetNumberTwo(int number) 17. { 18. numberTwo = number; 19. } 20. // Add two integers 21. public int Add() 22. { 23. return numberOne + numberTwo; 24. } 25. }   }   1. Configure project properties to make it COM visible. 2. Register for COM Interop. 3. Compile the project. 4. Copy the type library file to Windows system folder. 5. Reference the type library from Access VBA editor. 6. Use the DLL in your VBA code. 7. Public Function test() 8. Dim lngResult As Long 9. Dim objCalc As SimpleCalc.Calc 10. Set objCalc = New SimpleCalc.Calc 11. objCalc.SetNumberOne (3) 12. objCalc.SetNumberTwo (6) 13. lngResult = objCalc.Add()   End Function  Made available by [GeeksEngine.com](http://www.geeksengine.com/article/create-dll.html)   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/9526708)|[improve this answer](http://stackoverflow.com/posts/9526708/edit) | answered Mar 2 at 1:16  [[http://www.gravatar.com/avatar/3bc8b6aea2ee5ff74de362c0c8f0d4c6?s=32&d=identicon&r=PG](http://stackoverflow.com/users/3155/curtis-inderwiesche)](http://stackoverflow.com/users/3155/curtis-inderwiesche)  [Curtis Inderwiesche](http://stackoverflow.com/users/3155/curtis-inderwiesche) 1,82352250 | |
|  | | feedback |
| up vote -1 down vote | What about VSTO (Visual Studio Tools for Office)? It can be used to [create Office Add-Ins](http://msdn.microsoft.com/en-us/library/ms268878.aspx) in any .NET language.   |  | | --- | | [share](http://stackoverflow.com/a/1942866)|[improve this answer](http://stackoverflow.com/posts/1942866/edit) | | |