

Walkthrough: Creating an Outlook Form Region

Custom form regions extend standard or custom Microsoft Office Outlook 2007 forms. In this walkthrough, you will design a custom form region that contains a button, and add it to a task window. When you click the button, it displays a message.

For information about form regions, see [Outlook Form Regions Overview](#).

This walkthrough illustrates the following tasks:

- Designing a form region.
- Adding the form region to an Outlook 2007 add-in project.
- Connecting the form region custom code with the add-in.
- Creating a form region manifest.
- Adding form region registry entries.

Note

The dialog boxes and menu commands you see might differ from those described in Help depending on your active settings or edition. To change your settings, select **Import and Export Settings** on the **Tools** menu. For more information, see [Visual Studio Settings](#).

Prerequisites

You need the following components to complete this walkthrough:

- One of these two development environments:
 - Microsoft Visual Studio 2005 Tools for the Microsoft Office System (VSTO 2005).
- or-
- Visual Studio 2005 Professional Edition.
- Microsoft Visual Studio 2005 Tools for the 2007 Microsoft Office System (VSTO 2005 SE).
- Microsoft Office Outlook 2007.

Designing a Form Region

Forms regions can be designed only by using Microsoft Office Outlook 2007, so for this step, you will not need Visual Studio. Instead, you will create the form region in Outlook 2007, save it to an easy-to-find location, and then import it into Visual Studio.

To design a form region in Outlook 2007

1. Start Microsoft Office Outlook 2007.
2. On the **File** menu, point to **New**, and then click **Task**.
3. Click the **Microsoft Office Button**, and then click **Editor Options**.
4. Click **Popular** and then select **Show Developer Tab in the Ribbon**.
5. Click **OK**.
6. Click the **Microsoft Office Button**, and then click **Close**.
7. On the **Tools** menu, point to **Forms** and then click **Design a Form**.
8. In the **Design Form** box, click **Task** and then click **Open**.
9. On the **Developer** tab, in the **Design** group, click **Form Region** and then **New Form Region**.

A new form region opens. If the **Field Chooser** does not appear, click **Field Chooser** in the **Tools** group.

10. Drag the **Subject** field from the **Field Chooser** to the form region.
11. In the **Tools** group, click **Control Toolbox** to open the Toolbox.
12. Drag a **CommandButton** control from the **Toolbox** to the form region.

Note

Steps in this walkthrough refer to the default names of these two controls.

13. On the **Developer** tab, in the **Design** group, click **Form Region**, and then click **Save Form Region As**. Name the form region **TaskFormRegion** and save it to your My Documents directory.
14. Exit Outlook 2007.

Creating a New Outlook 2007 Add-in Project

You are now ready to customize the form region using VSTO 2005 SE. The Outlook add-in project template contains all the form region custom logic.

To create a new Outlook 2007 add-in project

1. Start Visual Studio.
2. Create a Microsoft Outlook 2007 Add-in project with the name **FormRegionAddIn**. Make sure that you use the Outlook Add-in project template for the 2007 Microsoft Office system.
3. In the **New Project** dialog box, make sure that the **Create directory for solution** checkbox is selected.
4. Save the project to [%USERPROFILE%]\My Documents\Visual Studio 2005\Projects\FormRegionWalkthrough\.

For more information, see [How to: Create Visual Studio Tools for Office Projects](#).

Adding the Form Region to the Add-in Project

In this step, you will import the form region you designed in Outlook 2007 into the add-in project as a resource.

To add the form region to the Outlook 2007 add-in project

1. In **Solution Explorer**, right-click the **FormRegionAddin** project and then click **Properties**.
2. Click the **Resources** tab.
3. On the Resource Designer toolbar, point to **Add Resource**, click the arrow, and then click **Add Existing File**.
4. In the **Add existing file to resources** dialog box, browse to your My Documents directory, select **TaskFormRegion.ofs** and then click **Open**.

TaskFormRegion.ofs is added to the Resources directory in **Solution Explorer**.

Adding Code to Customize the Form Region Behavior

In this step, you will add code to display a message box when the user clicks the button on the form region.

To accomplish this, you must implement the **Microsoft.Office.Interop.Outlook.FormRegionStartup** interface defined in the Microsoft.Office.Interop.Outlook assembly. For more information, see [Outlook Form Regions Overview](#).

To add code to customize the form region behavior

1. In **Solution Explorer**, right-click the **FormRegionAddin** project and then click **Add Reference**.

The **Add Reference** dialog box appears.

2. On the **COM** page, select the Microsoft Forms 2.0 Object Library and then click **OK**.
3. In **Solution Explorer**, right-click the **FormRegionAddin** project, point to **Add**, and then click **Class**.
4. Change the name of the class to **FormRegionHookup**, and then click **Add**.
5. Add the following statements to the top of the **FormRegionHookup.cs** or **FormRegionHookup.vb** code file.

VB

```
Imports System.Runtime.InteropServices
Imports Outlook = Microsoft.Office.Interop.Outlook
Imports Microsoft.Vbe.Interop.Forms
```

```
Imports System.Windows.Forms
```

6. Replace the definition of the `FormRegionHookup` class with the following code. This code implements the **Microsoft.Office.Interop.Outlook.FormRegionStartup** interface, and applies the [ComVisibleAttribute](#), [GuidAttribute](#), [ProgIdAttribute](#), and [ClassInterfaceAttribute](#) attributes to the `FormRegionHookup` class.

VB

```
<ComVisible(True), _  
Guid("88F7BFBE-7666-4a0c-BCFD-2740E6625E04"), _  
ProgId("FormRegionAddIn.FormRegionHookup"), _  
ClassInterface(ClassInterfaceType.AutoDual)> _  
Public Class FormRegionHookup  
    Implements Outlook.FormRegionStartup
```

7. Declare the following fields at the top of the `FormRegionHookup` class. You will map these fields to controls in the form region.

VB

```
Private FormRegion As Outlook.FormRegion  
Private userForm As UserForm  
Private OlkTextBox1 As Outlook.OlkTextBox  
Private WithEvents CommandButton1 As Outlook.OlkCommandButton
```

8. Add the following code to the `FormRegionHookup` class. This code implements the **Microsoft.Office.Interop.Outlook.FormRegionStartup.GetFormRegionStorage** method of the **Microsoft.Office.Interop.Outlook.FormRegionStartup** interface and returns `TaskFormRegion ofs` from the resource collection of the **FormRegionAddin** project.

VB

```
Public Function GetFormRegionStorage( _  
    ByVal FormRegionName As String, _  
    ByVal Item As Object, _  
    ByVal LCID As Integer, _  
    ByVal FormRegionMode As Outlook.OlFormRegionMode, _  
    ByVal FormRegionSize As Outlook.OlFormRegionSize) As Object _  
    Implements Outlook.FormRegionStartup.GetFormRegionStorage  
    Application.DoEvents()  
    Select Case FormRegionName  
        Case "TaskFormRegion"  
            Dim ofsBytes As Byte()  
            ofsBytes = My.Resources.TaskFormRegion  
            Return ofsBytes  
        Case Else
```

```

        Return Nothing
    End Select
End Function

```

9. Add the following code to the `FormRegionHookup` class. This code implements the **Microsoft.Office.Interop.Outlook.FormRegionStartup.BeforeFormRegionShow** method of the **Microsoft.Office.Interop.Outlook.FormRegionStartup** interface and maps the fields you defined in the `FormRegionHookup` class to controls on the form region. Outlook 2007 calls **Microsoft.Office.Interop.Outlook.FormRegionStartup.BeforeFormRegionShow** and then displays the form region.

VB

```

Public Sub BeforeFormRegionShow( _
ByVal FormRegion As Outlook.FormRegion) _
Implements Outlook.FormRegionStartup.BeforeFormRegionShow
    Me.FormRegion = FormRegion
    Me.userForm = FormRegion.Form

    Try
        OlkTextBox1 = userForm.Controls.Item("OlkTextBox1")
        CommandButton1 = userForm.Controls.Item("CommandButton1")
    Catch ex As Exception
        MessageBox.Show(ex.ToString())
    End Try
End Sub

```

10. Add the following code to the `FormRegionHookup` class. This code overrides two methods in the **Microsoft.Office.Interop.Outlook.FormRegionStartup** interface. By default, Outlook does not call these methods and you will not add code to them in this walkthrough. However, you must override them to successfully compile the project.

VB

```

Public Function GetFormRegionManifest( _
ByVal formRegionName As String, _
ByVal LCID As Integer) As Object _
Implements Outlook.FormRegionStartup.GetFormRegionManifest
    Throw New Exception("The method or operation is not implemented.")
End Function

Public Function GetFormRegionIcon( _
ByVal formRegionName As String, _
ByVal LCID As Integer, _
ByVal icon As Outlook.OlFormRegionIcon) As Object _
Implements Outlook.FormRegionStartup.GetFormRegionIcon
    Throw New Exception("The method or operation is not implemented.")
End Function

```

11. Add the following code to the `FormRegionHookup` class. This code handles the **Microsoft.Office.Interop.Outlook.OlkCommandButtonsEvent_Event.Click** event of the `ComandButton1` control and displays the contents of the form region text box.

VB

```
Private Sub CommandButton1_Click() Handles CommandButton1.Click
    MessageBox.Show(OlkTextBox1.Text)
End Sub
```

Connecting the Form Region Code with the Add-In

To connect the form region code with an add-in, create an instance of the **FormRegionHookup** class and pass that new object to Outlook 2007. To do this, override the [RequestService](#) method of the `ThisAddIn` class. For more information, see [Outlook Form Regions Overview](#).

To connect the form region with the add-in

1. In **Solution Explorer**, expand **FormRegionAddin**, expand **Outlook**, right-click **ThisAddin.cs** or **ThisAddin.vb**, and then click **View Code**.
2. Add the following code to the `ThisAddin` class. This code overrides the **RequestService** method and returns a **FormRegionHookup** object. Outlook 2007 calls this method during startup.

VB

```
Protected Overrides Function RequestService(ByVal serviceGuid As Guid)
    As Object
    If serviceGuid =
        GetType(Microsoft.Office.Interop.Outlook.FormRegionStartup).GUID Then
        Return New FormRegionHookup()
    Else
        Return MyBase.RequestService(serviceGuid)
    End If
End Function 'RequestService
```

Creating a Form Region Manifest

The form region manifest is an XML file that contains information that Outlook 2007 must have to load the form region. For more information about the XML schemas for form regions, download the schema reference ([2007 Office System: XML Schema Reference](#)) and see the sections "Form Region Manifest" and "Form Region Localization Manifest."

To create a form region manifest

1. In **Solution Explorer**, right-click **FormRegionAddIn**, point to **Add**, and then click **New Item**.
2. In the **Add New Item** dialog box, select **XML file**, name the file **Manifest.xml**, and click **Add**.
3. Replace the contents of **Manifest.xml** with the following XML.

Note

Outlook 2007 passes the value of the `<name>` element to the **Microsoft.Office.Interop.Outlook.FormRegionStartup.GetFormRegionStorage** method as the `FormRegionName` parameter.

```
<?xml version="1.0" encoding="utf-8"?>
<FormRegion
xmlns="http://schemas.microsoft.com/office/outlook/12/formregion.xsd">
  <name>TaskFormRegion</name>
  <title>Contoso</title>
  <formRegionType>separate</formRegionType>
  <showCompose>true</showCompose>
  <showRead>true</showRead>
  <showPreview>false</showPreview>
  <hidden>true</hidden>
  <addin>FormRegionAddIn</addin>
  <version>1.0</version>
</FormRegion>
```

1. In **Solution Explorer**, right-click **Manifest.xml**, and then click **Properties**.
2. In the **Properties** window, set the **Copy to Output Directory** property to **Copy Always**.

This makes sure that every build copies an updated version of the form region manifest to the output location.

Adding Form Region Registry Entries

Outlook 2007 uses registry entries to locate the form region manifest. A quick way to create these registry entries is to put them in a registry file and then run the file.

To add form region registry entries

1. Start Notepad and add the following text. `TaskFormRegion` is the location of the form region manifest.

```
Windows Registry Editor Version 5.00
```

```
[HKEY_CURRENT_USER\Software\Microsoft\Office\Outlook\FormRegions\IPM.Task]
"TaskFormRegion"="%USERPROFILE%\My Documents\Visual Studio
2005\Projects\FormRegionWalkthrough\FormRegionAddIn\FormRegionAddIn
\Manifest.xml"
```

2. On the **File** menu, click **Save**.
3. In the **File name** box, type **FormRegion.reg**.
4. Click the **Save as** type list, and then click **All Files**.
5. Choose a convenient location and click **Save**.
6. Double-click **FormRegion.reg**, click **Yes**, and then click **OK**.

This adds the information from **FormRegion.reg** to the registry. You can now delete this file, because you will not need it again.

Testing the Form Region

Test the form region by creating a new task and clicking the button on the form.

To test the form region

1. Activate Visual Studio, and then press CTRL+F5.

Microsoft Office Outlook 2007 starts.

2. In Outlook, on the **File** menu, point to **New**, and then click **Task**.
3. In the **Subject** box, type a name for the task.
4. On the **Task** tab, in the **Show** group, click **Contoso**.

The custom form region appears.

5. Click **CommandButton1**.

Outlook displays the name of the task subject.

See Also

Concepts

[Outlook Form Regions Overview](#)