Deploying and Optimizing a SharePoint Web Part That Calls Excel Web Services

**Summary:** Learn how to deploy a Web Part by using deployment CABs, and how to improve the appearance and configurability of a Web Part. (10 printed pages)

Dan Battagin, Microsoft Corporation

November 2006

**Applies to:** Microsoft Office SharePoint Server 2007, Microsoft Office Excel 2007

Download: [MOSS2007ExcelWebPart.exe](http://www.microsoft.com/downloads/details.aspx?familyid=2c5919e4-0757-4db1-9798-e0e4032f12a8&displaylang=en)

Watch the [SharePoint 2007 Video: Excel Services Mortgage Calculator](http://wm.microsoft.com/ms/msdn/office/sharepoint2007excelservicesmortgagecalc/sharepoint2007excelservicesmortgagecalc.wmv).

**Contents**

* [Introduction to Deploying and Optimizing a SharePoint Web Part](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_Introduction)
* [Creating a Deployment Project to Install Your Web Part](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_DeploymentProject)
* [Creating the CAB Project](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_CABProject)
* [Deploying the Web Part and Adding It to a Web Part Page](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_Deploying)
* [Improving the Appearance and Configurability of the Web Part](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_Appearance)
* [Redeploying the Updated Mortgage Calculator Web Part](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_Redeploy)
* [Conclusion](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_Conclusion)
* [Additional Resources](http://msdn2.microsoft.com/en-us/library/aa973805.aspx#Office2007ExcelServicesWebPart2_AdditionalResources)

**Introduction to Deploying and Optimizing a SharePoint Web Part**

In [Using Excel Web Services in a SharePoint Web Part](http://msdn2.microsoft.com/en-us/library/aa973804.aspx), you learn how to create a Web Part that calls Excel Web Services. The next step is to package and deploy the Web Part. This article describes how to build a deployment project that you can use to deploy the mortgage calculator Web Part that you created. You also learn how to configure and improve the appearance of a Web Part.

**Systems Requirements**

To create and run the samples, you need the following software installed on your computer:

* Visual Studio 2005
* Microsoft Office Excel 2007
* Office SharePoint Server 2007

**Creating a Deployment Project to Install Your Web Part**

One of the easiest ways to deploy your Web Part to Office SharePoint Server 2007 is to create a CAB file that can be deployed by using the Stsadm.exe tool. In this section, you create a CAB project in Visual Studio 2005 that will contain the elements of your Web Part.

**Creating Support Files for the CAB Project**

Before you create a CAB project, you must create the following files in your XlMortageCalc project:

* Manifest.xml, which specifies the contents of the CAB
* MortgageCalculator.dwp, which is a Web Part definition file that is used by Office SharePoint Server 2007

**To create the Manifest.xml file**

1. In Solution Explorer, right-click the **XlMortageCalc** project, point to **Add**, and then click **New Item**.
2. Select **XML File**, and name the file **Manifest.xml**.
3. Click **Add**.
4. Add the following content to the file:

Xml

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl05xml');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl05xml');)

<?xml version="1.0"?> <!-- You need only one manifest per CAB project for Web Part Deployment.--> <!-- This manifest file can have multiple assembly nodes.--> <WebPartManifest xmlns="http://schemas.microsoft.com/WebPart/v2/Manifest"> <Assemblies> <Assembly FileName="XlMortgageCalc.dll"> <!-- Use the <ClassResource> tag to specify resources like image files or Microsoft JScript files that your Web Parts use. --> <!-- Note that you must use relative paths when specifying resource files. --> <ClassResources></ClassResources> <SafeControls> <SafeControl Namespace="XlMortgageCalc" TypeName="\*" /> </SafeControls> </Assembly> </Assemblies> <DwpFiles> <DwpFile FileName="XlMortgageCalc.dwp"/> </DwpFiles> </WebPartManifest>

**To create the MortgageCalculator.dwp file**

1. In Solution Explorer, right-click the **XlMortageCalc** project, point to **Add**, and then click **New Item**.
2. Select **XML File**, and name the file **XlMortgageCalc.dwp**.
3. Click **Add**.
4. Add the following content to the file:

Xml

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl06xml');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl06xml');)

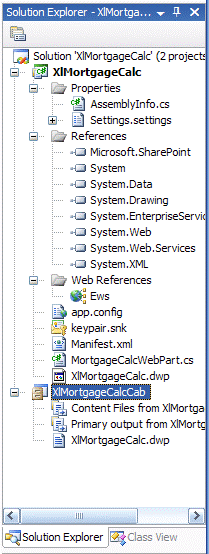
<?xml version="1.0" encoding="utf-8"?> <WebPart xmlns="http://schemas.microsoft.com/WebPart/v2" > <Title>Mortgage Calculator</Title> <Description>Mortgage Calculator Web Part that uses Excel Services to calculate monthly mortgage payments.</Description> <Assembly>XlMortgageCalc, Version=1.0.0.0, Culture=neutral, PublicKeyToken=cf4fe2436d2bd078</Assembly> <TypeName>XlMortgageCalc.XlMortgageCalc</TypeName> <!-- Specify initial values for any additional base class or custom properties here. --> </WebPart>

|  |
| --- |
| **NoteNote:** |
| The PublicKeyToken must be set to the public key that is specified for your XlMortgageCalc.dll file. Follow these steps to determine the public key:  1. Open a Visual Studio 2005 Command Prompt window.  2. Type **sn.exe -T** *full path to your XlMortgageCalc.dll file*.  3. Copy the public key token that is returned. |

**Creating the CAB Project**

CAB projects are simple projects that gather several files from your Web Part project into a single CAB file that can be deployed by Office SharePoint Server 2007, as shown in Figure 1.

**Figure 1. Solution Explorer with CAB project displayed**



**To create a CAB project**

1. Start Visual Studio.
2. On the **File** menu, point to **New**, and then click **Project**.

The **New Project** dialog box appears.

1. In the **Project Type** pane, from the **Other Project Types** node, select **Setup and Deployment**.
2. In the **Templates** pane, click **Cab Project**.
3. Name the CAB project **XlMortgageCalcCab**.
4. Click **OK**.

Next, you add the necessary files to the CAB project.

**To add files to the CAB project**

1. In Solution Explorer, right-click the **XlMortgageCalcCab** project.
2. Point to **Add**, and then click **Project Output**.
3. Press CTRL+C (to select multiple items) and then select the **Primary output** and **Content** files.
4. Click **OK**.
5. In Solution Explorer, again right-click the **XlMortgageCalcCab** project, point to **Add**, and then click **File**.
6. Browse and select the **XlMortgageCalc.dwp** file that you created earlier.
7. Save and build the solution.

**Deploying the Web Part and Adding It to a Web Part Page**

Follow these steps to deploy your Web Part to Office SharePoint Server 2007.

**To deploy the Web Part to SharePoint Server 2007**

1. For ease of deployment, copy the XlMortgageCalcCab.cab file to the following location, where Office SharePoint Server 2007 is installed:

C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\bin

This should be the same as the location of Stsadm.exe. Ensure that you update the path on your computer if it does not match the default location.

1. Click **Start**, click **Run**, and then type **cmd**.
2. Type **cd "C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\bin"**
3. Run the following command to deploy your Web Part:

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl07other');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl07other');)

stsadm -o addwppack -filename XlMortgageCalcCab.cab -url http://localhost -globalinstall -force

1. After the command returns with "Operation completed successfully," run the following command:

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl08other');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl08other');)

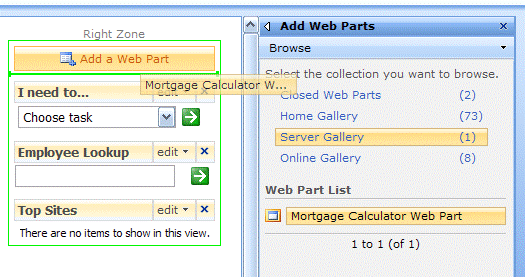
iisreset /timeout:0

Your Web Part is now successfully deployed to Office SharePoint Server 2007; the only remaining step is to add it to a Web page. In this case, you add it to the home page, but the following steps work for any SharePoint page.

**To add the Web part to a Web page**

1. Open your Web browser and browse to "http://YourServer" as a user who has administrative permissions to Office SharePoint Server 2007.
2. Click **Site Actions**, and then click **Edit Page**.
3. In the **Right** zone, click **Add Web Part**.
4. In the **Add Web Parts - Web Page Dialog** page, at the lower-right corner, click **Advanced Web Part gallery and options** to display the **Add Web Parts** gallery.
5. Click **Server Gallery**.
6. In the **Server Gallery**, you see the mortgage calculator Web Part. Drag the mortgage calculator Web Part to the **Right** zone.

**Figure 2. Dragging the Web Part to the SharePoint home page**



1. At the top of the page, click **Submit for approval**.
2. After the page refreshes, click **Approve**.

You should now have a working mortgage calculator on your Office SharePoint Server 2007 home page that uses Excel Services to perform the actual calculation. To change the calculation, you just update the workbook; no coding is involved.

**Improving the Appearance and Configurability of the Web Part**

You now have a working mortgage calculator Web Part within Office SharePoint Server 2007. Currently, it cannot be configured to use a different workbook (located at a different location on your server) to perform the calculation. It is also not a very good looking Web Part. By using a bit more code and some cascading style sheet (CSS) styles, you can improve the appearance and configurability of the Web Part so that you can specify the workbook to use through the Web browser.

**Updating the Look of the Web Part**

You might have noticed in the code that was added to the Web Part that the following CSS classes were already specified:

* **textInput**
* **total**
* **mainTable**
* **error**

In this section, you add definitions for these CSS classes to your project.

**To add CSS class definitions**

1. Inside your XlMortgageCalc class, add the following internal class (**XlMortgageCalc.Constants**):

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl09CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl09CSharp');)

internal class Constants { public static string Styles = @" .mainTable { background-position: center center; background-attachment: fixed; background-image: url('\_WPR\_/house.jpg'); background-repeat: no-repeat; background-color: #FFFFFF } .mainTable td, .mainTable input { font-family:Verdana; font-size:8pt } .total { font-weight:bold;width: 100; } .error { font-weight:bold; color:red; } .textInput { text-align:right; width: 75px; font-family:Verdana; font-size:8pt }"; }

1. At the beginning of the **MortgageCalcPart\_Load** method, add the following lines of code:

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl10CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl10CSharp');)

// Add our style sheet. HtmlGenericControl stylesheet = new HtmlGenericControl("style"); stylesheet.InnerHtml = ReplaceTokens(Constants.Styles); this.Controls.Add(stylesheet);

You might notice that you are referencing an image, house.jpg, in the new styles. It is included in the download, and is stored in the \_WPR\_ (Web Part Resources) folder for this Web Part. The \_WPR\_ folder is determined at run time by the **ReplaceTokens** method that is called in step 2. Next, you must add the house.jpg image to the project, and to the CAB file.

**To add an image to the project**

1. Drag the house.jpg image that is included in the download (or any other image you want to use) onto the **XlMortgageCalc** node in Solution Explorer.
2. In the Manifest.xml file, replace the following line:

Xml

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl11xml');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl11xml');)

<ClassResources></ClassResources>

With these lines:

Xml

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl12xml');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl12xml');)

<ClassResources> <ClassResource FileName="house.jpg"/> </ClassResources>

1. Rebuild your solution to ensure that everything builds correctly.

**Adding Web Part Properties to Make the Web Part Configurable**

In this section, you add a Web Part property that can be configured in the browser to specify the workbook that the Web Part should use to perform the calculation. You call this property **WorkbookUrl**.

**To add a Web Part property**

1. Open MortgageCalcWebPart.cs.
2. In the **XlMortgageCalc.Constants** class, add the following constant:

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl13CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl13CSharp');)

public const string WorkbookUrl = @"http://localhost/shared documents/mortgagecalc.xlsx";

1. In the **XlMortgageCalc** class, add the following private member:

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl14CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl14CSharp');)

#region Private Members private string \_workbookUrl = Constants.WorkbookUrl; #endregion

1. In the **XlMortgageCalc** class, add the following public property.

Note that several attributes on this property tell Microsoft Windows SharePoint Services how to display the property when it is shown in the Web Part Page task pane.

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl15CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl15CSharp');)

#region Public Properties [Browsable(true), Category("Miscellaneous"), DefaultValue(Constants.WorkbookUrl), WebPartStorage(Storage.Personal), FriendlyName("Workbook URL"), Description("Enter the URL of the Excel workbook that should be used to perform the calculations for this Web Part.")] public string WorkbookUrl { get { return \_workbookUrl; } set { \_workbookUrl = value; } } #endregion

1. Finally, update the code in your project where this property should be used.

Change the following line of code:

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl16CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl16CSharp');)

sessionId = es.OpenWorkbook("http://TODOYourServer/Documents/MortgageCalc.xlsx", "en-US", "en-US", out status);

To be:

C#

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl17CSharp');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl17CSharp');)

sessionId = es.OpenWorkbook(this.WorkbookUrl, "en-US", "en-US", out status);

1. Rebuild your solution to ensure that everything builds correctly.

**Redeploying the Updated Mortgage Calculator Web Part**

Now that you have updated your Web Part, you must redeploy it and configure it to work in your environment. To do that, you first need to remove the original version of the Web Part that you installed to SharePoint, and then install and configure the new version.

**To redeploy the Web Part**

1. For ease of deployment, copy the XlMortgageCalcCab.cab file to the following location on the computer where Office SharePoint Server 2007 is installed:

C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\bin.

This should be the same location as Stsadm.exe. Ensure that you update the path on your computer if it does not match the default location.

1. Click **Start**, click **Run**, and then type **cmd**.
2. Type **cd "C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\bin"**
3. Run the following command to remove your old Web Part:

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl18other');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl18other');)

stsadm -o deletewppack -name XlMortgageCalcCab.cab -url http://localhost

1. Run the following command to install your new Web Part:

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl19other');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl19other');)

stsadm -o addwppack -filename XlMortgageCalcCab.cab -url http://localhost -globalinstall -force

1. After this command returns with "Operation completed successfully," run the following command:

[[http://msdn2.microsoft.com/msdn/Controls/CodeSnippet/en-us/copy_off.gif](javascript:CopyCode('ctl00_LibFrame_ctl20other');)Copy Code](javascript:CopyCode('ctl00_LibFrame_ctl20other');)

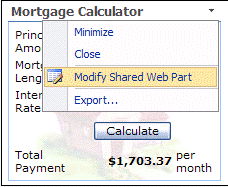
iisreset /timeout:0

Finally, you can configure your Web Part, as shown in Figure 3.

**To configure the Web Part**

1. In the upper-right corner of the Mortgage Calculator Web Part, click the drop-down menu, and then select **Modify Shared Web Part**.
2. Expand the **Miscellaneous** section (at the bottom of the task pane).
3. Enter the correct value for the location of your workbook, and then click **OK**.

**Figure 3. Configuring the Web Part properties for the Mortgage calculator Web Part**



**Conclusion**

In a short amount of time, and with a little bit of code, you created a Web Part deployment package. You also improved the appearance of the Web Part that you created in [Using Excel Web Services in a SharePoint Web Part](http://msdn2.microsoft.com/en-us/library/aa973804.aspx) and configured it to make it more user friendly.

**About the Author**

Dan Battagin is a lead program manager on the Excel team who also worked on Excel Services