

Developing Advanced Web Parts for SharePoint 2010 with Visual Studio 2010

Hands-On Lab

Lab Manual – SPCHOL301

SPCHOL301– Advanced Web Parts – C#

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft. All rights reserved.

**Contents**

[Lab Objective 2](#_Toc259797261)

[Additional Resources 2](#_Toc259797262)

[Getting Started 2](#_Toc259797263)

[Logging in to the Virtual Machine 2](#_Toc259797264)

[Locations 2](#_Toc259797265)

[Lab Pre-requisites 3](#_Toc259797266)

[Copying code samples from Word document 3](#_Toc259797267)

[Code Snippets 4](#_Toc259797268)

[Exercise 1 – Build a Visual Web Part using SharePoint controls 5](#_Toc259797269)

[Task 1 – Create a SharePoint project of type Visual Web Part 5](#_Toc259797270)

[Task 2 – Create a Projects Listing Web Part 7](#_Toc259797271)

[Task 3 – Deploy and Test 8](#_Toc259797272)

[Exercise 2 – XSLT List View Web Part 12](#_Toc259797273)

[Task 1 – Open SharePoint Designer 12](#_Toc259797274)

[Task 2 – Add XSLT List View Web Part 14](#_Toc259797275)

[Task 3 – Modify XSLT List View Web Part 17](#_Toc259797276)

[Task 4 – Deploy and Test 19](#_Toc259797277)

[Lab Summary 20](#_Toc259797278)

**Developing Advanced Web Parts for SharePoint 2010 with Visual Studio 2010**

Estimated time to complete this lab: **30 minutes**

## Lab Objective

One of the core ASP.NET technologies used by SharePoint to present dynamic information to users is Web Parts. They are the most common customization created for SharePoint. A Web Part is a reusable component that exists on a Web Part Page and can present any type of web-based information.

The objective of this lab is to learn about some of the more advanced Web Part functionality specific to the SharePoint system.

* Create a basic SharePoint Web Part which us SharePoint Controls to display information from within the SharePoint site.
* Create and customize an XsltListViewWebPart in SharePoint Designer 2010.

## Additional Resources

This lab includes the following additional resources:

|  |  |  |
| --- | --- | --- |
| This Lab Manual | SPCHOL301\_Manual\_CS.docx | This document |
| Source Code | Completed\CS\Ex1 | Completed lab source code. |
| Resources | *Resources\CS* | Various resources used throughout this lab. |

## Getting Started

### Logging in to the Virtual Machine

Please log into the virtual machine as the following user:

**Username:** Administrator

**Password:** pass@word1

### Locations

This Hands-On Lab contains a number of additional resources in fixed locations. By default, it is assumed that the base HOL directory is **C:\Content Packs\Packs\SharePoint 2010 Developer Labs 1.0\SUPPORTING FILES\SPCHOL301\Resources**..

The default working folder for this lab is **C:\SPHOLS\SPCHOL301**.

### Lab Pre-requisites

Browse to base HOL directory ***Supporting Files\SPCHOL301\Resources***and execute the **optimize.ps1** PowerShell script:

1. Right click on **optimize.ps1** and select **Run with PowerShell**:

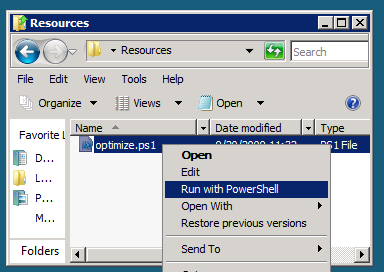


Figure 1 - Execute the PowerShell script

1. This will open a PowerShell window to execute the script. Please wait until the PowerShell script completes executing the script and closes the PowerShell window (this may take a few minutes):

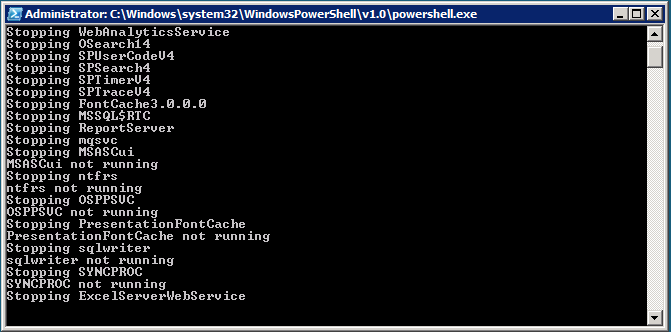


Figure 2 - PowerShell Window executing the script

### Copying code samples from Word document

Copying and pasting code from this Word document to Visual Studio is only safe for the sections of formatted code, e.g.:

Console.WriteLine("This is safe code!");

Code not in these sections may contain Unicode or other invisible characters that are not valid XML or C#/VB code, e.g.:

**Console.WriteLine(“This is NOT safe code !!”);**

### Code Snippets

You can also use Code Snippets to insert the appropriate code in the lab.

To use the required code snippet for this lab:

* Right click on the code file where you want to insert the code snippet.
* Select **Insert Snippet:**

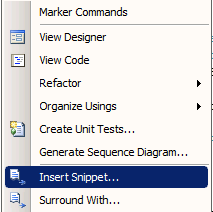


Figure 3 - Visual Studio Code Context Menu.

* Select code snippets from **My Code Snippets** gallery.

## Exercise 1 – Build a Visual Web Part using SharePoint controls

**Exercise Duration**: 15 minutes

**Exercise Overview**

This example illustrates how to create a Visual Web Part and use SharePoint controls in Visual Studio 2010.

### Task 1 – Create a SharePoint project of type Visual Web Part

In this task a project you will create a Visual Web Part solution and use SharePoint user controls.

1. Open **Visual Studio 2010** by going to **Start Menu | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu select **File | New | New Project**.
3. Select the **Visual C# | SharePoint | 2010 |** **Visual Web Part** project template.
4. Enter **ProjectsListWebPart**in the Name textbox.
5. Enter **C:\SPHOLS\SPCHOL301\CS\Ex1** in the Location textbox.

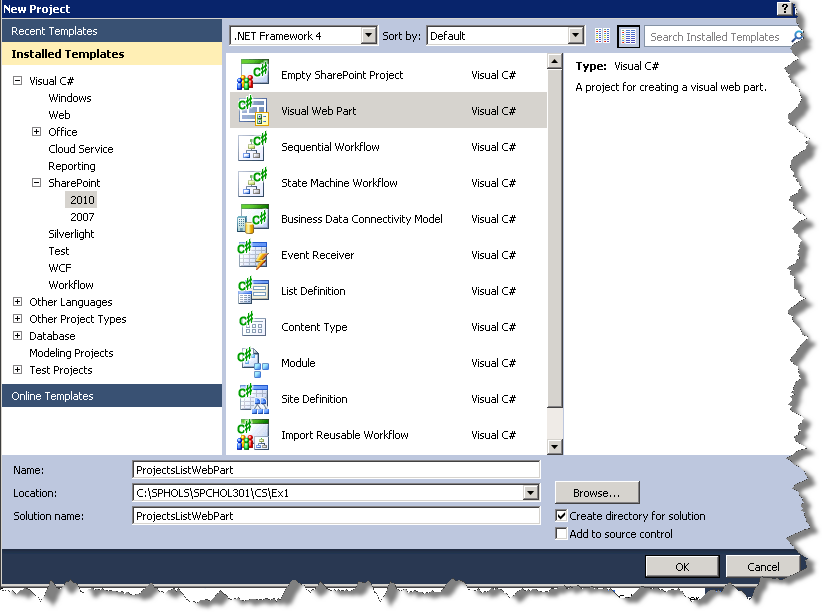
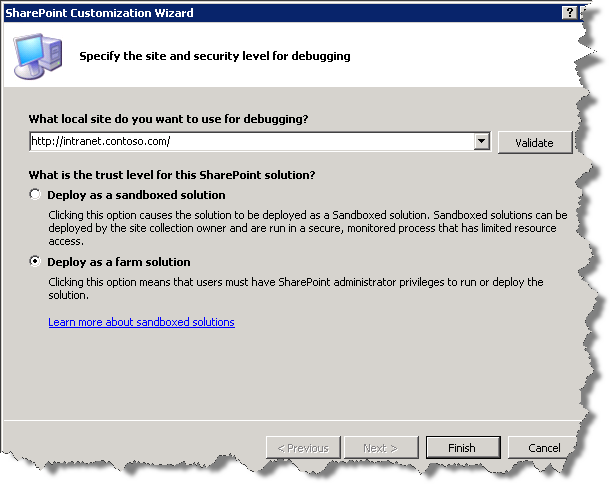


Figure 4 - New Project Dialog

1. Press **OK** to create the project.
2. A **SharePoint Customization Wizard** dialog box will appear.
3. Enter [**http://intranet.contoso.com**](http://intranet.contoso.com)**/** in the “What local site do you want to use for debugging?” textbox.
4. Select “**Deploy as farm solution**” radio button from the “What is the trust level for this SharePoint solution?” drop-down

Figure 5 - SharePoint Customization Wizard dialog

1. Press **Finish** to complete the customization wizard.
2. Visual Studio will add the necessary files.

### Task 2 – Create a Projects Listing Web Part

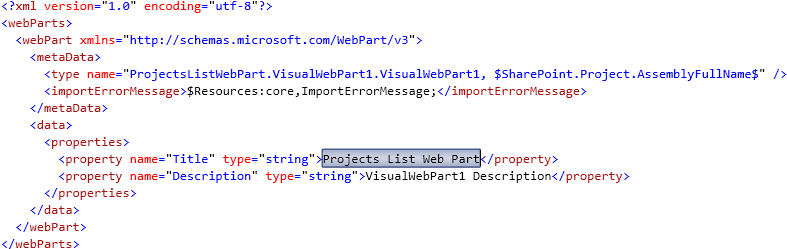
1. In the Solution Explorer, double-click on **VisualWebPart1.webpart**.
2. Change the **Title** property in the VisualWebPart1.webpartfile to **Projects List Web Part**.

Figure 6 - VisualWebPart1.webpart after title change

1. Add the following code to the end of VisualWebPart1UserControl.ascx.This code adds a **ListViewByQuery** SharePoint control, a **Label** ASP control, and a **DateTime** SharePoint control. The same can be achieved by dragging the control from the **Toolbox** onto the **VisualWebPart1UserControl.ascx** files canvas. You’ll have to make sure the **IDs** match the ones in the code below.

<SharePoint:ListViewByQuery runat="server" ID="ProjectsListView" />

<br />

<asp:Label ID="Label1" runat="server" Text="Items due by:" />

<SharePoint:DateTimeControl ID="DueDate" AutoPostBack="true" OnDateChanged="OnDate\_Changed"

DateOnly="true" runat="server">

</SharePoint:DateTimeControl>

**Code Snippet:** *My HTML Snippets | spchol301\_ex1\_usercontrol*

1. In the Solution Explorer, right-click on **VisualWebPart1UserControl.ascx** and select **View Code.**
2. Add the following **using** statements.

using Microsoft.SharePoint;

using Microsoft.SharePoint.Utilities;

using System.Web;

**Code snippet:** *My Code Snippets | spchol301\_ex1\_namespaces*

1. Add the following code to the body of the **Page\_Load** method.

DateTime dueDate = DateTime.Now;

if (!string.IsNullOrEmpty(Request.QueryString["date"]))

{

dueDate = DateTime.Parse(Request.QueryString["date"]);

}

DueDate.SelectedDate = dueDate;

SPWeb home = SPContext.Current.Web;

ProjectsListView.List = home.Lists["Projects"];

SPQuery query = new SPQuery(ProjectsListView.List.DefaultView);

query.ViewFields = "<FieldRef Name=\"Title\" /><FieldRef Name=\"Due\_x0020\_Date\" />";

query.Query = string.Format(

"<Where><Leq><FieldRef Name=\"Due\_x0020\_Date\" /><Value Type=\"DateTime\">{0}</Value></Leq></Where>"

, dueDate.ToString("s"));

ProjectsListView.DisableFilter = true;

ProjectsListView.DisableSort = true;

ProjectsListView.Query = query;

**Code snippet:** *My* *Code Snippets | spchol301\_ex1\_pageload*

1. Add the following code after the **Page\_Load** method.

protected void OnDate\_Changed(object sender, EventArgs e)

{

SPUtility.Redirect(SPContext.Current.ListItem.Name

, SPRedirectFlags.Default

, HttpContext.Current

, string.Format("date={0}", DueDate.SelectedDate.ToString("d")));

}

**Code snippet:** My Code Snippets | spchol301\_ex1\_ondatechanged

### Task 3 – Deploy and Test

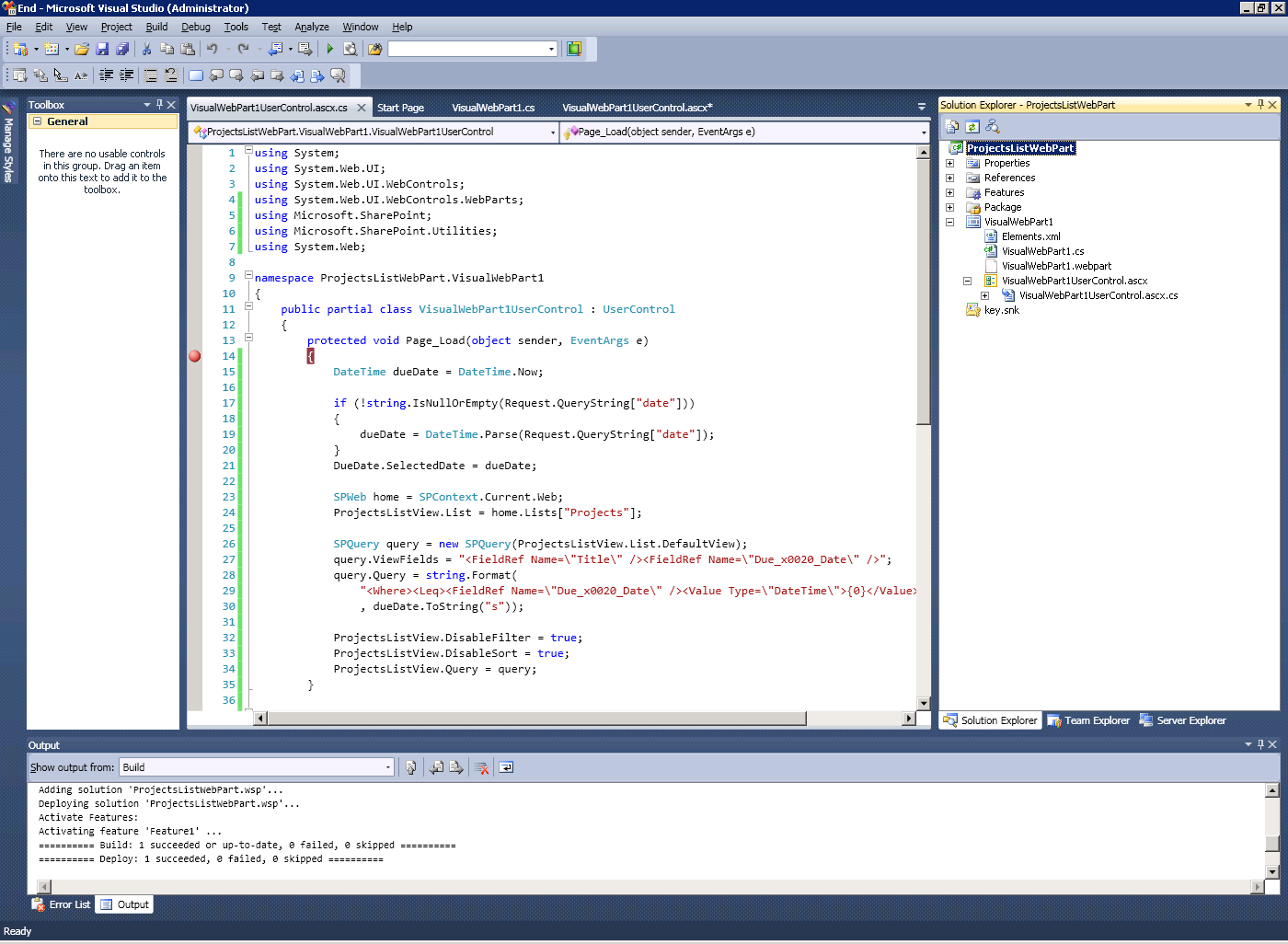
1. You are now ready to deploy the Web Part. Right-click on the **Project** in **Solution Explorer**. Select **Deploy.**
2. If you view the **Output** window you can see that the deployment succeeded.  
     
   

Figure 7 - Build Output window

1. Open a web browser to **http://intranet.contoso.com**. You are now going to add the Web Part to the home page.
2. Click the **Edit** icon in the ribbon.

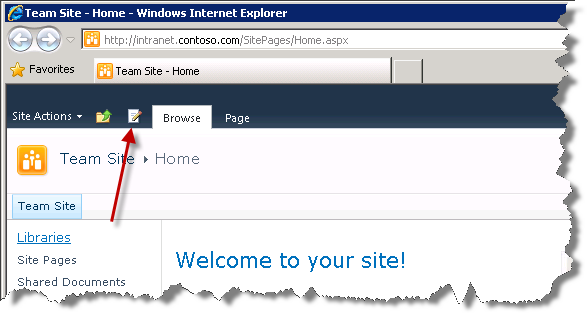


Figure 8 - Edit icon on the landing page

1. Click the **Insert** tab under **Editing Tools** in the Ribbon, as shown below.

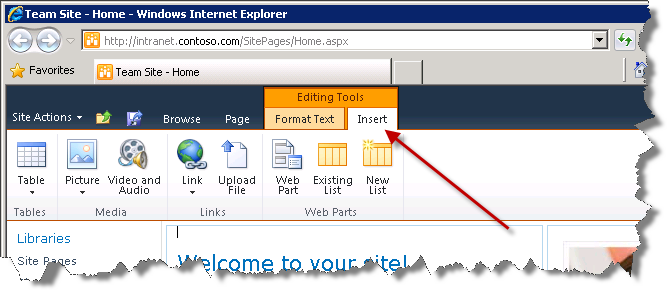


Figure 9 - Insert tab on the edit ribbon

1. Click the **Web Part** button. Then under **Categories** select **Custom. Projects List Web Part** should appear in the list of Web Parts. Select **Add** on the right hand side.

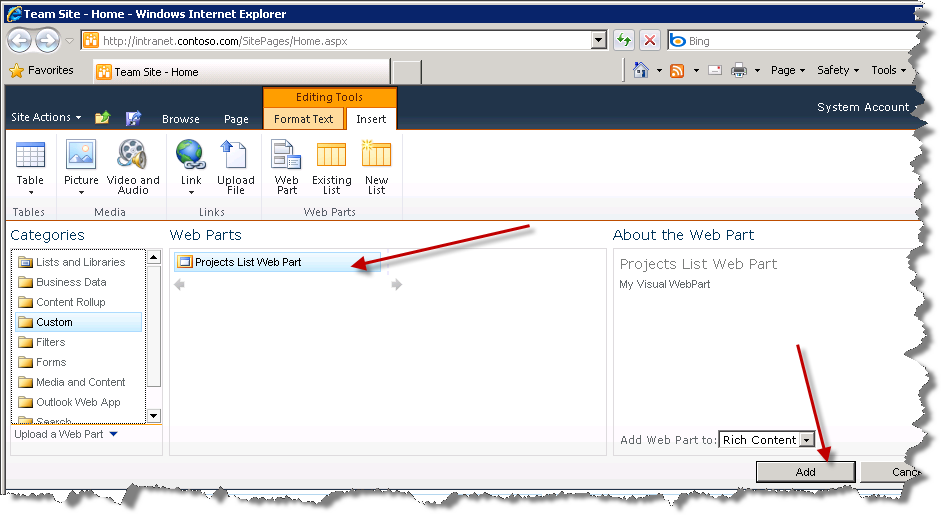


Figure 10 - Webpart insert form

1. Click on the **Save & Close** button in the ribbon. Select **Yes** if the **Save Changes** dialog appears.

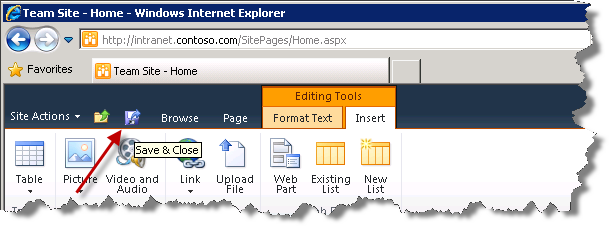


Figure 11 - Edit ribbon highligting the Stop Editing button

1. You now have your new Web Part listing projects on the home page.

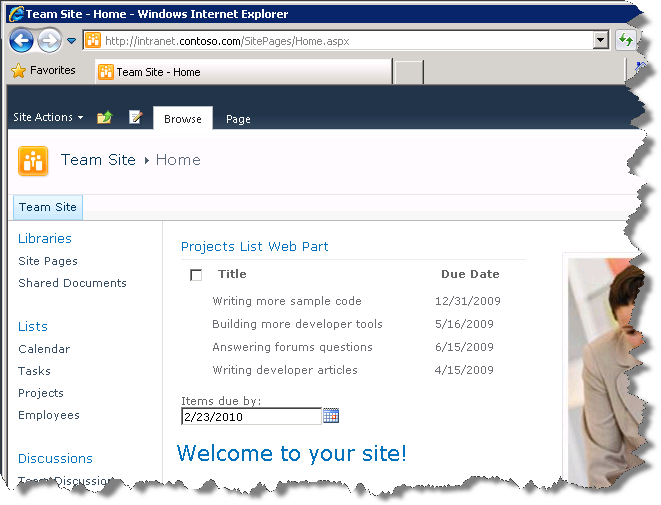


Figure 12 - The working webpart

In the past few minutes you have demonstrated building a Visual Web Part with SharePoint controls.

## Exercise 2 – XSLT List View Web Part

The XSLT List View Web Part is a new web part in SharePoint Foundation 2010. This web part is used in place of the Data View Web Part in Windows SharePoint Services v3.0 (The Data View Web Part is still available in SharePoint Foundation 2010). The XSLT List View Web Part is very easy to create in SharePoint Designer 2010.

For this exercise you will create an XSLT List View Web Part within SharePoint Designer 2010 that will show a list of employees on the Home Page of the Contoso Intranet. The list of employees will be pulled from the Employees list that exists within the Contoso site. The web part will be created completely in SharePoint Designer 2010. This will give us an understanding of the markup created by the XSLT List View Web Part.

### Task 1 – Open SharePoint Designer

1. Open **SharePoint Designer 2010** by going to **Start Menu | All Programs | SharePoint | Microsoft SharePoint Designer 2010.**.
2. **Click on the Open Site button. In the Open Site dialog window, enter http://intranet.contoso.com in the Site Name textbox..**

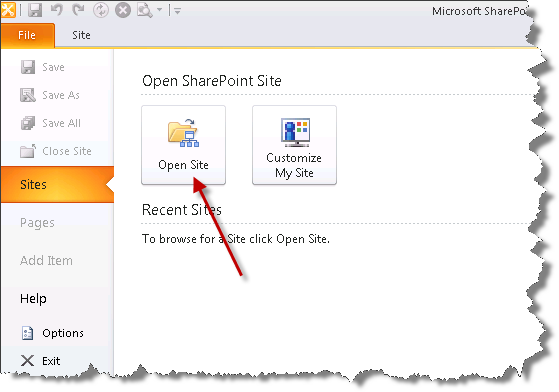
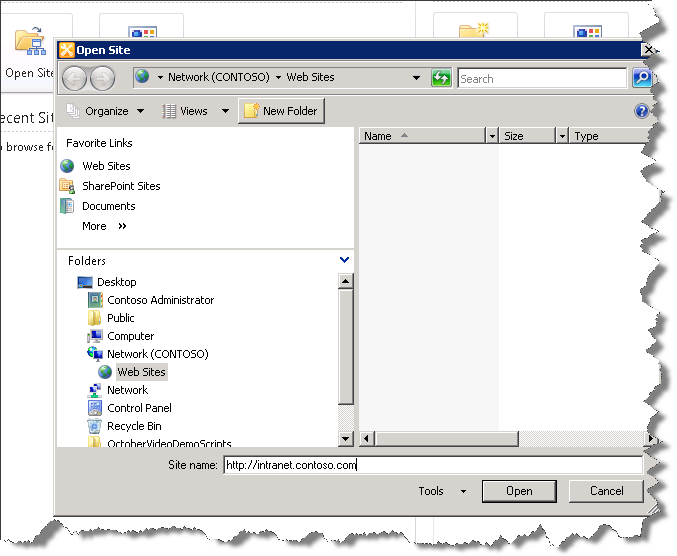


Figure 13 - SharePoint Designer 2010 Site menu

  
Figure 14 - Open Site dialog window

1. If prompted for authentication, enter the following details:

**Username**: administrator

**Password:** pass@word1

1. From the **SharePoint Site** left navigation select **Site Pages**, then **Home.aspx**.

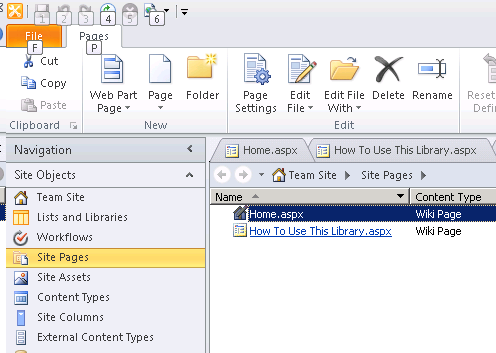


Figure 15 - Site Pages - Home Page

1. From the Ribbon menu, select **Edit File | Edit File in Advanced Mode**. Ignore any errors you may see on this screen, which opens to the Browse tab of the Designer pane. You will quickly be moving on.

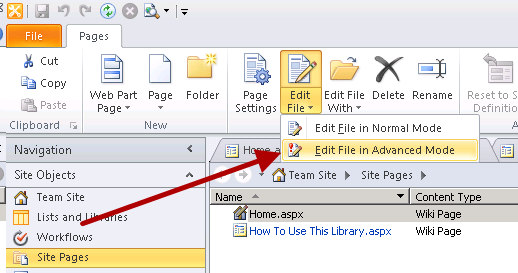


Figure 16 - Edit Page in Advanced Mode button in the ribbon

### Task 2 – Add XSLT List View Web Part

1. Click on the **Code** tab at the bottom of the designer pane to show the **Code** view.

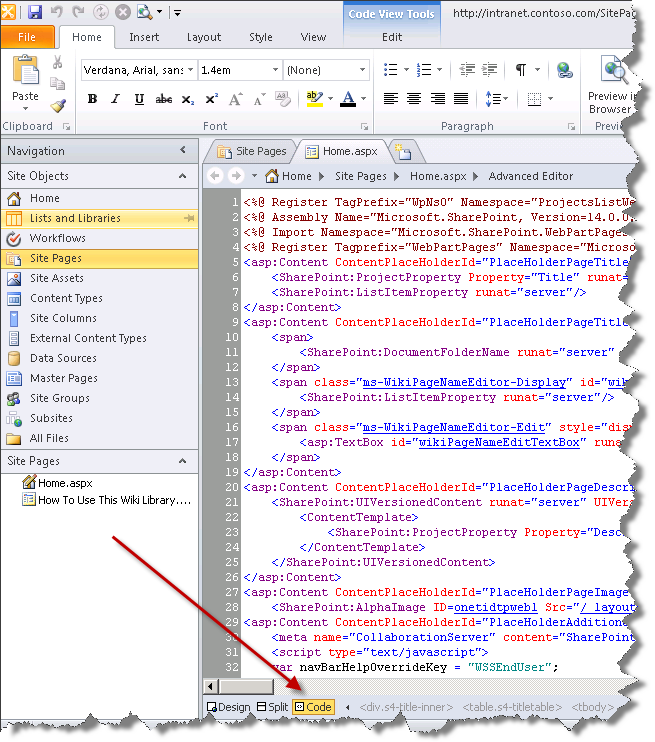


Figure 17 - Code view tab

1. In the **Code** view, scroll down to the bottom of the page and place your cursor before the </ContentTemplate> end tag (right after the WebPartPages:WebPartZone closing tag)

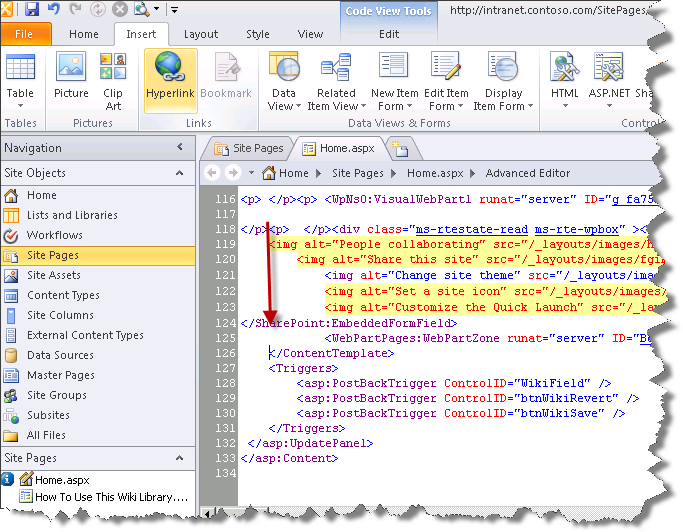


Figure 18 - Selecting the first div in the bottom web part zone

1. Select the **Insert** tab in the ribbon.
2. Click on the **Data View** item dropdown.

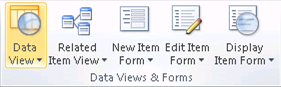


Figure 19 - Data view button in the ribbon

1. Select the **Employees** list.
2. SharePoint designer will add the new **XsltListViewWebPart** to the **HomePage.aspx**.
3. Click the **Save** icon at the top of the page.

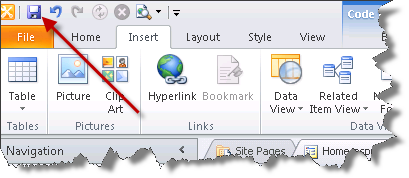


Figure 20 - Save button highlighted in the ribbon

1. If prompted for confirmation, click **Yes**.
2. Open a web browser to [**http://intranet.contoso.com**](http://intranet.contoso.com).
3. Ignore any warnings, if any.
4. You should now see the new XsltListViewWebPart.

### Task 3 – Modify XSLT List View Web Part

1. Return to **SharePoint Designer 2010.** You are now going to make some modifications to the **XsltListViewWebPart.**
2. The **Fullname** column of the **Employees** list needs to stand out.
3. In the **Designer View**, select the table cell containing the word **Fullname.**



Figure 21 - Fullname Table Cell

1. Using the **Apply Styles** (if the **Apply Styles** window isn’t visible – go to the **View** tab and select **Apply Styles** under the **Task Panes** button) **window** on the bottom right**.** Scroll down to thein the **Select CSS style to apply:** to **.ms-bodyareaagemargin.** Right click **Apply Style.**

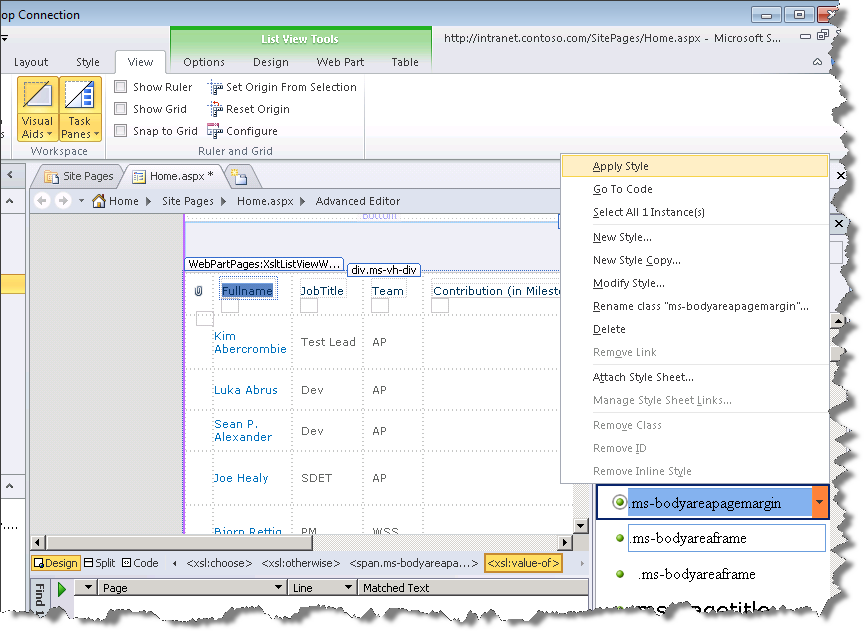


Figure 22 - Apply Style

1. In the **Code** view of the Designer scroll up to **RowLimit** and change the value **30** to *5.*

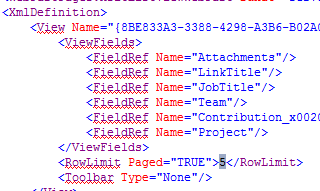


Figure 23 - Change row limit

1. In the code view of the Designer scroll up to **WebPartPages:XsltListViewWebPart** and add the **ChromeType** property and choose *TitleAndBorder* as the chrome type



Figure 24 - Add ChromeType Property

1. Click the **Save** icon at the top of the page.

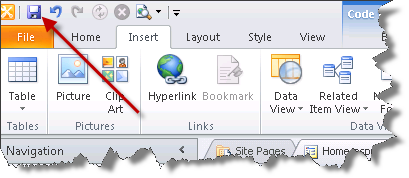


Figure 25 - Save button highlighted in the ribbon

### Task 4 – Deploy and Test

1. Open a web browser to **http://intranet.contoso.com**. You should now see the modified XsltListViewWebPart listing only 5 employees per page.

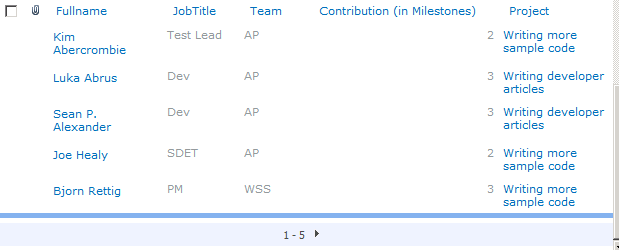


Figure 26 - Modified XsltListViewWebPart

## Lab Summary

In this lab you performed the following exercises.

Created a Visual Web Part using Visual Studio 2010.

Used a SharePoint Control from the toolbox.

Used the SharePoint ListViewByQuery control.

Used the integrated deployment within Visual Studio 2010 to deploy the SharePoint Visual Web Part project.

Used SharePoint Designer to create an XsltListViewWebPart.

Customized an XsltListViewWebPart.

Manipulated a SharePoint Web Site to host a Web Part.

In this lab, you learned how to create a SharePoint Visual Web Part from scratch. You learned how to use controls from both the Toolbox as well as SharePoint controls such as the ListViewByQuery and deployed the Web Part solution. Actions were also performed upon the SharePoint Web interface where you added a Web Part to an existing page and modified that Web Part accordingly. You also used the XsltListViewWebPart, and customized it using SharePoint Designer.