

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

Web Parts enable users to directly modify the content, appearance, and behavior of SharePoint site pages by using a browser. This walkthrough shows you how to create a Web Part by using the **Web Part** item template in Visual Studio 2010.

The Web Part displays employees in a data grid. The user specifies the location of the file that contains the employee data. The user can also filter the data grid so that employees who are managers appear in the list only.

This walkthrough illustrates the following tasks:

- Creating a Web Part by using the Visual Studio **Web Part** item template.
- Creating a property that can be set by the user of the Web Part. This property specifies the location of the employee data file.
- Rendering content in a Web Part by adding controls to the Web Part controls collection.
- Creating a new menu item, referred to as a *verb*, that appears in the verbs menu of the rendered Web part. Verbs enable the user to modify the data that appears in the Web Part.
- Testing the Web Part in SharePoint.

Note: Your computer might show different names or locations for some of the Visual Studio user interface elements in the following instructions. The Visual Studio edition that you have and the settings that you use determine these elements.

Prerequisites

You need the following components to complete this walkthrough:

- Supported editions of Microsoft Windows and SharePoint. For more information, see Requirements for Developing SharePoint Solutions.
- Visual Studio 2010 Professional or an edition of Visual Studio Application Lifecycle Management (ALM).

Creating an Empty SharePoint Project

First, create a Empty SharePoint project. Later, you will add a Web Part to the project by using the **Web Part** item template.

To create an Empty SharePoint Project

1. Start Visual Studio 2010 by using the **Run as Administrator** option.
2. On the **File** menu, point to **New**, and then click **Project**. The **New Project** dialog box appears.
3. Open the **New Project** dialog box, expand the **SharePoint** node under the language that you want to use, and then select the **2010** node.
4. In the **Templates** pane, select **Empty SharePoint Project**, and then click **OK**. The **SharePoint Customization Wizard** appears. This wizard enables you to select the site that you will use to debug the project and the trust level of the solution.

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

5. Select **Deploy as a farm solution**, and then click **Finish** to accept the default local SharePoint site.

Adding a Web Part to the Project

Add a **Web Part** item to the project. The **Web Part** item adds the Web Part code file. Later, you will add code to the Web Part code file to render the contents of the Web Part.

To add a Web Part to the Project

1. On the **Project** menu, click **Add New Item**.
2. In the **Add New Item** dialog box, in the **Installed Templates** pane, expand the **SharePoint** node, and then click **2010**.
3. In the list of SharePoint templates, select **Web Part**, and then click **Add**. The **Web Part** item appears in **Solution Explorer**.

Rendering Content in the Web Part

You can specify which controls you want to appear in the Web Part by adding them to the controls collection of the Web Part class.

To render content in the Web Part

1. In **Solution Explorer**, double-click WebPart1.vb (in Visual Basic) or WebPart1.cs (in C#). The Web Part code file opens in Code Editor.
2. Add the following statements to the top of the Web Part code file.

```
using System.Data;
```

3. Add the following code to the WebPart1 class. This code declares the following fields:

- A data grid to display employees in the Web Part.
- Text that appears on the control that is used to filter the data grid.
- A label that displays an error if the data grid is unable to display data.
- A string that contains the path of the employee data file.

```
private DataGrid grid;  
private static string verbText = "Show Managers Only";  
private Label errorMessage = new Label();  
protected string xmlFilePath;
```

4. Add the following code to the WebPart1 class. This code adds a custom property named DataFilePath to the Web Part. A custom property is a property that can be set in SharePoint by the user. This property gets and sets the location of a XML data file that is used to populate the data grid.

```
[Personalizable(PersonalizationScope.Shared), WebBrowsable(true),  
WebDisplayName("Path to Employee Data File"),  
WebDescription("Location of the XML file that contains employee data")]  
public string DataFilePath  
{  
    get  
    {  
        return xmlFilePath;  
    }  
}
```

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

```
set
{
    xmlFilePath = value;
}
}
```

5. Replace the CreateChildControls method with the following code. This code performs the following tasks:

- Adds the data grid and label that you declared in the previous step.
- Binds the data grid to an XML file that contains employee data.

```
protected override void CreateChildControls()
{
    // Define the grid control that displays employee data in the Web Part.
    grid = new DataGrid();
    grid.Width = Unit.Percentage(100);
    grid.GridLines = GridLines.Horizontal;
    grid.HeaderStyle.CssClass = "ms-vh2";
    grid.CellPadding = 2;
    grid.BorderWidth = Unit.Pixel(5);
    grid.HeaderStyle.Font.Bold = true;
    grid.HeaderStyle.HorizontalAlign = HorizontalAlign.Center;

    // Populate the grid control with data in the employee data file.
    try
    {
        DataSet dataset = new DataSet();
        dataset.ReadXml(xmlFilePath, XmlReadMode.InferSchema);
        grid.DataSource = dataset;
        grid.DataBind();
    }
    catch (Exception x)
    {
        errorMessage.Text += x.Message;
    }

    // Add control to the controls collection of the Web Part.
    Controls.Add(grid);
    Controls.Add(errorMessage);
    base.CreateChildControls();
}
```

6. Add the following method to the WebPart1 class. This code performs the following tasks:

- Creates a verb that appears in the Web Part verbs menu of the rendered Web part.
- Handles the event that is raised when the user clicks the verb in the verbs menu. This code filters the list of employees that appears in the data grid.

```
public override WebPartVerbCollection Verbs
{
    get
    {
        WebPartVerb customVerb = new WebPartVerb("Manager_Filter_Verb",
```

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

```
new WebPartEventHandler(CustomVerbEventHandler));

customVerb.Text = verbText;
customVerb.Description = "Shows only employees that are managers";

WebPartVerb[] newVerbs = new WebPartVerb[] { customVerb };

return new WebPartVerbCollection(base.Verbs, newVerbs);
}
}

protected void CustomVerbEventHandler(object sender, WebPartEventArgs args)
{
    int titleColumn = 2;

    foreach (DataGridItem item in grid.Items)
    {
        if (item.Cells[titleColumn].Text != "Manager")
        {
            if (item.Visible == true)
            {
                item.Visible = false;
            }
            else
            {
                item.Visible = true;
            }
        }
    }

    if (verbText == "Show Managers Only")
    {
        verbText = "Show All Employees";
    }
    else
    {
        verbText = "Show Managers Only";
    }
}
}
```

Testing the Web Part

When you run the project, the SharePoint site opens. The Web Part is automatically added to the Web Part Gallery in SharePoint. You can add the Web Part to any Web Part page.

To test the Web Part

1. Paste the following XML into a Notepad file. This XML file contains the sample data that will appear in the Web Part.

```
<?xml version="1.0" encoding="utf-8" ?>
  <employees xmlns="http://schemas.microsoft.com/vsto/samples">
    <employee>
      <name>David Hamilton</name>
      <hireDate>2001-05-11</hireDate>
      <title>Sales Associate</title>
    </employee>
    <employee>
```

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

```
<name>Karina Leal</name>
<hireDate>1999-04-01</hireDate>
<title>Manager</title>
</employee>
<employee>
  <name>Nancy Davolio</name>
  <hireDate>1992-05-01</hireDate>
  <title>Sales Associate</title>
</employee>
<employee>
  <name>Steven Buchanan</name>
  <hireDate>1955-03-04</hireDate>
  <title>Manager</title>
</employee>
<employee>
  <name>Suyama Michael</name>
  <hireDate>1963-07-02</hireDate>
  <title>Sales Associate</title>
</employee>
</employees>
```

2. In Notepad, click **File**, and then click **Save As**.
3. In the **Save As** dialog box, in the **Save as type** drop-down list, select **All Files**.
4. In the **File name** box, type data.xml.
5. Select any folder by using the **Browse Folders** button and then click **Save**.
6. In Visual Studio, press **F5**. The SharePoint site opens.
7. Click **Site Actions**, and then click **More Options**.
8. In the **Create** page, select **Web Part Page**, then click **Create**.
9. In the **New Web Part Page** page, name the page SampleWebPartPage.aspx, and then click **Create**. The Web Part page appears.
10. Select any zone on the Web Part page.
11. At the top of the page, click **Insert**, and then click **Web Part**.
12. In the **Categories** pane, click the **Custom** folder, click the **WebPart1** Web Part, and then click **Add**. The Web Part appears on the page.

Testing the Custom Property

To populate the data grid that appears in the Web Part, specify the path of the XML file that contains data about each employee.

To test the custom property

1. Click the arrow that appears in the corner of the Web Part, and then click **Edit Web Part**. A pane that contains properties for the Web Part appears on the right side of the page.

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

2. In the pane, expand the **Miscellaneous** node, type the path of the XML file that you created earlier, and click **Apply**, and then click **OK**. Verify that a list of employees appears in the Web Part.

Testing the Web Part Verb

Show and hide employees that are not managers by clicking an item that appears in the Web Part verbs menu.

To test the Web Part verb

1. Click the arrow that appears in the corner of the Web Part, and then click **Show Managers Only**. Only employees who are managers appear in the Web Part.
2. Click the arrow again, and then click **Show All Employees**. All employees appear in the Web Part.

Creating a Web Part for SharePoint by Using a Designer Visual Studio 2010

Web Parts enable users to directly modify the content, appearance, and behavior of SharePoint site pages by using a browser. This walkthrough shows you how to create a Web Part visually by using the SharePoint **Visual Web Part** project template in Visual Studio 2010.

The Web Part displays a monthly calendar view and a check box for each calendar list on the site. Users can choose which calendar lists to include in the monthly calendar view by selecting the checkboxes.

This walkthrough illustrates the following tasks:

- Creating a Web Part by using the **Visual Web Part** project template.
- Designing the Web Part by using the Visual Web Developer designer in Visual Studio.
- Adding code to handle the events of controls on the Web Part.
- Testing the Web Part in SharePoint.

Note: Your computer might show different names or locations for some of the Visual Studio user interface elements in the following instructions. The Visual Studio edition that you have and the settings that you use determine these elements.

Prerequisites

You need the following components to complete this walkthrough:

- Supported editions of Microsoft Windows and SharePoint. For more information, see Requirements for Developing SharePoint Solutions.
- Visual Studio 2010 Professional or an edition of Visual Studio Application Lifecycle Management (ALM).

Creating a Web Part Project

First, create a Web Part project by using the **Visual Web Part** project template.

To create a Visual Web Part Project

1. Start Visual Studio 2010 by using the **Run as Administrator** option.
2. On the **File** menu, point to **New**, and then click **Project**. If your IDE is set to use Visual Basic development settings, on the **File** menu, click **New Project**. The **New Project** dialog box appears.

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

3. Open the **New Project** dialog box, expand the **SharePoint** node under the language that you want to use, and then select the **2010** node.
4. In the **Visual Studio Installed Templates** pane, select **Visual Web Part**, and then click **OK**. The **SharePoint Customization Wizard** appears. This wizard enables you to select the site that you will use to debug the project and the trust level of the solution.
5. Click **Finish** to accept the default local SharePoint site and default trust level for the solution. By default, the Visual Web Developer displays the user control of the project in **Source** view where you can see the page's HTML elements.

Designing the Web Part

Design the Web Part by dragging controls from the **Toolbox** to the surface of the user control.

To design the layout of the Web Part

1. On the Visual Web Developer designer, click the **Design** tab to switch to Design view.
2. On the **View** menu, click **Toolbox**.
3. In the **Toolbox**, from the **Standard** group, drag a **CheckBoxList** and a **Button** to the designer.
4. In the designer, click **Button**.
5. On the **View** menu, click **Properties Window**.
6. In the **Properties** window, set the **Text** property of the button to Update.

Handling the Events of Controls on the Web Part

Add code that enables the user to add calendars to the master calendar view.

To handle events of controls on the Web Part

1. On the designer, double-click the **Update** button. The user control code file opens in Code Editor and the Button1_Click event handler appears. Later, you will add code to this event handler.
2. Add the following statements to the top of the user control code file.

```
using Microsoft.SharePoint;  
using Microsoft.SharePoint.WebControls;
```

3. Add the following line of code to the VisualWebPart1userControl class. This code declares a monthly calendar view control.

```
private MonthlyCalendarView MonthlyCalendarView1;
```

4. Replace the Page_Load method of the VisualWebPart1UserControl class with the following code. This code performs the following tasks:

- Adds a monthly calendar view to the user control.
- Adds a check box for each calendar list on the site.
- Specifies a template for each type of item that appears in the calendar view.

```
protected void Page_Load(object sender, EventArgs e)  
{  
    MonthlyCalendarView1 = new MonthlyCalendarView();  
    this.Controls.Add(MonthlyCalendarView1);  
    SPCalendarItemCollection items = new SPCalendarItemCollection();
```

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

```
SPWeb thisWeb = SPControl.GetContextWeb(Context);

if (CheckBoxList1.Items.Count == 0)
{
    foreach (SPList listItem in thisWeb.Lists)
    {
        if (listItem.BaseTemplate == SPListTemplateType.Events)
        {
            CheckBoxList1.Items.Add(new ListItem(listItem.Title));
        }
    }
}

MonthlyCalendarView1.ItemTemplateName =
    "CalendarViewMonthItemTemplate";
MonthlyCalendarView1.ItemAllDayTemplateName =
    "CalendarViewMonthItemAllDayTemplate";
MonthlyCalendarView1.ItemMultiDayTemplateName =
    "CalendarViewMonthItemMultiDayTemplate";
}
```

5. Replace the Button1_Click method of the VisualWebPart1UserControl class with the following code.

This code adds items from each selected calendar to the master calendar view.

```
protected void Button1_Click(object sender, EventArgs e)
{
    SPCalendarItemCollection items = new SPCalendarItemCollection();
    SPWeb thisWeb = SPControl.GetContextWeb(Context);

    foreach (ListItem item in CheckBoxList1.Items)
    {
        if (item.Selected == true)
        {
            SPList calendarList = thisWeb.Lists[item.Text];
            DateTime dtStart = DateTime.Now.AddDays(-7);
            DateTime dtEnd = dtStart.AddMonths(1).AddDays(7);
            SPQuery query = new SPQuery();
            query.Query = String.Format(
                "<Query>" +
                "<Where><And>" +
                "<Geq><FieldRef Name=\"{0}\" />" +
                "<Value Type=\"DateTime\">{1}</Value></Geq>" +
                "<Leq><FieldRef Name=\"{0}\" />" +
                "<Value Type=\"DateTime\">{2}</Value></Leq>" +
                "</And></Where><OrderBy><FieldRef Name=\"{0}\" /></OrderBy>" +
                "</Query>",
                "Start Time",
                dtStart.ToShortDateString(),
                dtEnd.ToShortDateString());

            foreach (SPListItem listItem in calendarList.GetItems(query))
            {
                SPCalendarItem calItem = new SPCalendarItem();
                calItem.ItemID = listItem["ID"].ToString();
                calItem.Title = listItem["Title"].ToString();
                calItem.CalendarType = Convert.ToInt32(SPCalendarType.Gregorian);
            }
        }
    }
}
```

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

```
calItem.StartDate = (DateTime)listItem["Start Time"];
calItem.ItemID = listItem.ID.ToString();
calItem.WorkspaceLink = String.Format(
    "/Lists/{0}/DispForm.aspx", calendarList.Title);
calItem.DisplayFormUrl = String.Format(
    "/Lists/{0}/DispForm.aspx", calendarList.Title);
calItem.EndDate = (DateTime)listItem["End Time"];
calItem.Description = listItem["Description"].ToString();
if (listItem["Location"] != null)
{
    calItem.Location = listItem["Location"].ToString();
}
items.Add(calItem);
}
MonthlyCalendarView1.DataSource = items;
}
}
}
```

Testing the Web Part

When you run the project, the SharePoint site opens. The Web Part is automatically added to the Web Part Gallery in SharePoint. To test this project, you will perform the following tasks:

- Add an event to each of two separate calendar lists.
- Add the Web Part to a Web Part Page.
- Select lists to include in the monthly calendar view.

To add events to calendar lists on the site

1. In Visual Studio, press **F5**. The SharePoint site opens and the Microsoft SharePoint Foundation 2010 Quick Launch bar appears on the page.
2. In the Quick Launch bar, under **Lists**, click **Calendar**. The **Calendar** page appears.
3. Click **Events**, and then click **New Event**.
4. In the **Title** box, type Event in the default calendar, and then click **Save**.
5. Click **Site Actions**, and then click **More Options**.
6. In the **Create** page, click **Calendar**, name the calendar, and then click **Create**. The **Custom Calendar** page appears.
7. Add a new event to the custom calendar named Event in the custom calendar.

To add the Web Part to a Web Part Page

1. Click **Site Actions**, and then click **More Options**.
2. In the **Create** page, click **Web Part Page**, and then click **Create**.
3. In the **New Web Part Page** page, name the page SampleWebPartPage.aspx, and then click **Create**. The Web Part page appears.

Creating a Web Part for SharePoint in Visual Studio 2010

(Microsoft Corporation)

4. Select any zone on the Web Part page.
5. At the top of the page, click the **Insert** tab, and then click **Web Part**.
6. Click the **Custom** folder, click the **VisualWebPart1** Web Part, and then click **Add**. The Web Part appears on the page. The following controls appear on the Web Part:
 - A monthly calendar view.
 - An **Update** button.
 - A **Calendar** check box.
 - A **Custom Calendar** checkbox.

To select lists to include in the monthly calendar view

In the Web Part, select calendars that you want to include in the monthly calendar view, and then click **Update**. Events from all selected calendars appear in the monthly calendar view.