

A SharePoint Developer Introduction

Hands-On Lab

Lab Manual

SPCHOL307 – Developing SharePoint 2010 Sandboxed Solutions in Visual Studio 2010 – 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**

[SPCHOL307 – Developing SharePoint 2010 Sandboxed Solutions in Visual Studio 2010 2](#_Toc260142531)

[Lab Objective 2](#_Toc260142532)

[Additional Resources 2](#_Toc260142533)

[Getting Started 2](#_Toc260142534)

[Logging in to the Virtual Machine 2](#_Toc260142535)

[Locations 2](#_Toc260142536)

[Lab Pre-requisites 3](#_Toc260142537)

[Copying code samples from Word document 3](#_Toc260142538)

[Code Snippets 4](#_Toc260142539)

[Exercise 1 – Create a Web Part for a Sandboxed Solution 5](#_Toc260142540)

[Task 1 – Create a Sandboxed Solution Project with a web part 5](#_Toc260142541)

[Task 2 – Add code to provide querying and rendering functionality 7](#_Toc260142542)

[Task 3 – Build and Deploy the Sandboxed Solution 10](#_Toc260142543)

[Exercise 2 – Creating a Sandboxed Solution Web Part with overload code 15](#_Toc260142544)

[Task 1 – Create a Sandboxed Solution Project with a web part 15](#_Toc260142545)

[Task 2 – Add code to overload the Sandboxed Solution web part 17](#_Toc260142546)

[Task 3 – Set the SharePoint Quotas and Limits 18](#_Toc260142547)

[Task 4 – Build and Deploy the Sandboxed Solution 22](#_Toc260142548)

[Lab Summary 25](#_Toc260142549)

# SPCHOL307 – Developing SharePoint 2010 Sandboxed Solutions in Visual Studio 2010

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

## Lab Objective

The objective of this lab is to learn about how to create a web part that renders and updates list data that is deployed as a Sandboxed Solution.

A Sandboxed Solution can be deployed to a site by a site administrator without requiring intervention from the farm administrator. The solution has full access to the immediate site and restricted access to system resources and other sites.

## Additional Resources

This lab includes the following additional resources:

|  |  |  |
| --- | --- | --- |
| This Lab Manual | SPCHOL307\_Manual\_CS.docx | This document |
| Source Code | \Supporting Files\SPCHOL307\Completed\CS\Ex1\ \Supporting Files\SPCHOL307\Completed\CS\Ex2\ | Completed lab source code in C#. |
| Resources | \Supporting Files\SPCHOL307\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 Resources directory is **C:\Content Packs\Packs\SharePoint 2010 Developer Labs 1.0\Supporting Files\SPCHOL307\Resources**.

The default working folder for this lab is \SPHOLS\SPCHOL307.

### Lab Pre-requisites

Browse to base HOL directory ***Supporting Files\SPCHOL307\Resources***and execute the **optimize.ps1** PowerShell script:

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

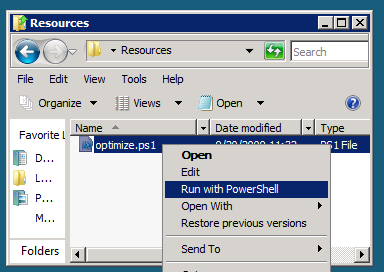


Figure 1 - Execute the PowerShell script

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 moments):

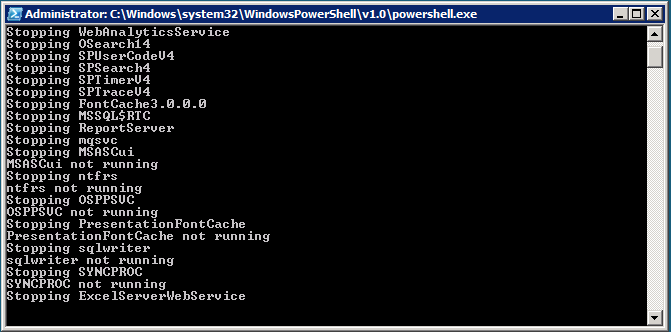


Figure 2 - PowerShell Window executing the script

### 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:**

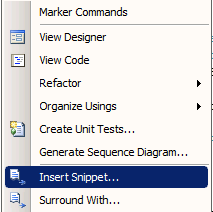


Figure 3 - Visual Studio Code Context Menu.

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

## Exercise 1 – Create a Web Part for a Sandboxed Solution

In this exercise, we will create a web part that renders and updates list data that is deployed as a Sandboxed Solution.

### Task 1 – Create a Sandboxed Solution Project with a web part

We will start by creating a standard Silverlight application project.

1. Open Visual Studio 2010 from **Start | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | New | Project**.
3. In the New Project dialog box, expand the Installed Templates left-hand menu to **Visual C# | SharePoint | 2010 | Empty SharePoint Project**.
4. Name the project **SBSolutionDemo**.
5. Change the location to **C:\SPHOLS\SPCHOL307\CS\Ex1\**

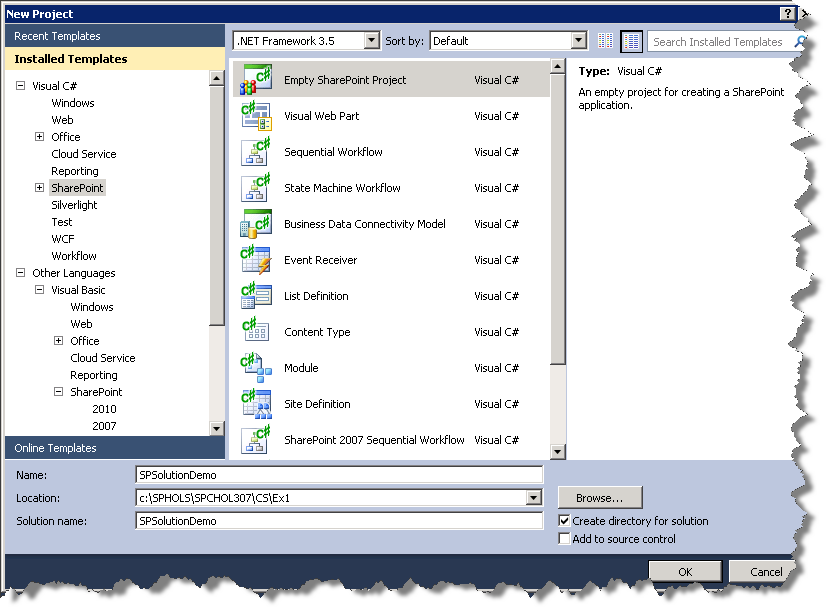


Figure 4 - New Project Dialog

1. Press **OK** to continue.
2. In the SharePoint Customization Dialog, change the local site to use for debugging to **http://intranet.contoso.com/**.
3. Leave the trust level for the SharePoint solution as **Deploy as a sandboxed solution**.

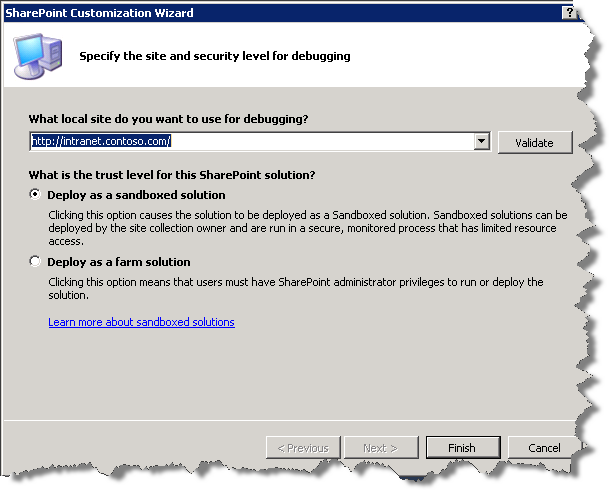


Figure 5 - SharePoint Customization Wizard Step1.

1. Press **Finish** to continue. Visual Studio will create the new project and add the necessary files.
2. Right-click on the **SBSolutionDemo** project in the Solution Explorer and select **Add | New item**.
3. Ensure **Visual C# | SharePoint | 2010 |** is selected in the Installed Templates pane.
4. In the Add New Item dialog, select to add a new **Web Part** and name it **SBWebPart**.

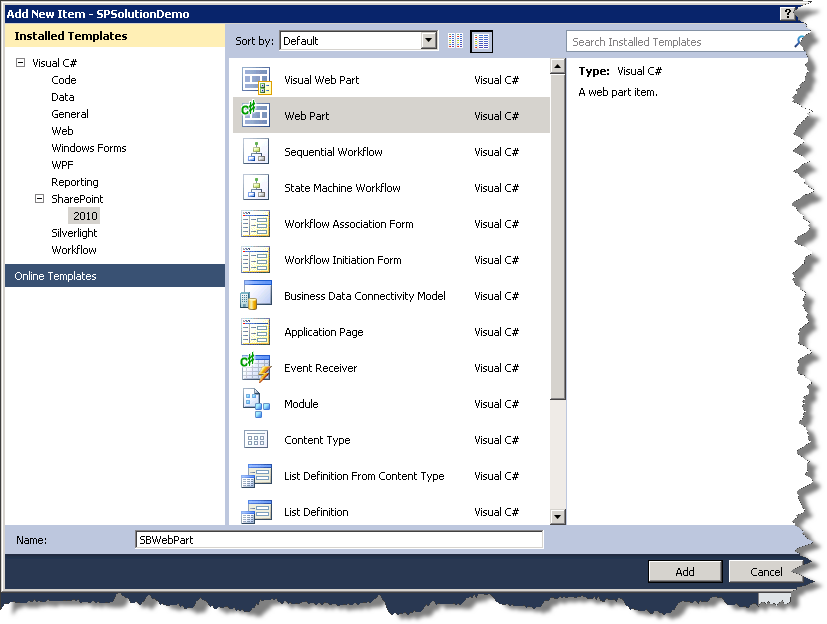


Figure 6 - Add New Item Dialog

1. Press **Add** to add the web part to the project.

### Task 2 – Add code to provide querying and rendering functionality

1. Open **SBWebPart.cs** and add the following using statement after the other using statements:

using System.Web.UI.HtmlControls;

**Code Snippet**: *My Code Snippets | spchol307\_ex1\_webpart\_namespaces*

1. Add the following variables to the **SBWebPart** class:

DropDownList \_ddlProjects = new DropDownList();

TextBox \_tbDescription = new TextBox();

TextBox \_tbDueDate = new TextBox();

**Code Snippet**: *My Code Snippets | spchol307\_ex1\_webpart\_variables*

1. Add the following new methods within the **SBWebPart** class:

protected override void OnLoad(EventArgs e)

{

base.OnLoad(e);

if (!Page.IsPostBack)

GetProjectDetails();

}

/\* Populate the text boxes with the selected project details \*/

private void GetProjectDetails()

{

EnsureChildControls();

if (\_ddlProjects.SelectedValue != "-- Select a Project --")

{

SPList pList = SPContext.Current.Web.Lists["Projects"];

int nProjectID = Convert.ToInt32(\_ddlProjects.SelectedValue);

SPListItem spliProject = pList.GetItemById(nProjectID);

\_tbDescription.Text = spliProject["Description"].ToString();

DateTime dueDate = Convert.ToDateTime(spliProject["Due\_x0020\_Date"]);

\_tbDueDate.Text = dueDate.ToShortDateString();

}

else

{

\_tbDescription.Text = String.Empty;

\_tbDueDate.Text = String.Empty;

}

}

**Code Snippet**: *My Code Snippets | spchol307\_ex1\_webpart\_getprojectdetails*

1. Replace the existing **CreateChildControls** method with the following code:

protected override void CreateChildControls()

{

base.CreateChildControls();

Panel parent = new Panel();

parent.Style.Add("border", "solid 1px Navy");

parent.Style.Add("background-color", "#EEEEEE");

parent.Style.Add("width", "250px");

\_ddlProjects.ID = "ddlProjects";

\_ddlProjects.AutoPostBack = true;

\_ddlProjects.SelectedIndexChanged += new

EventHandler(ddlProjects\_SelectedIndexChanged);

PopulateProjects();

parent.Controls.Add(\_ddlProjects);

Panel panel = new Panel();

Label label = new Label();

label.Text = "Description";

panel.Controls.Add(label);

parent.Controls.Add(panel);

panel = new Panel();

panel.Controls.Add(\_tbDescription);

parent.Controls.Add(panel);

label = new Label();

label.Text = "Due Date";

panel = new Panel();

panel.Controls.Add(label);

parent.Controls.Add(panel);

panel = new Panel();

panel.Controls.Add(\_tbDueDate);

parent.Controls.Add(panel);

panel = new Panel();

Button bUpdateProject = new Button();

bUpdateProject.Text = "Update Project";

bUpdateProject.Click += new EventHandler(bUpdateProject\_Click);

panel.Controls.Add(bUpdateProject);

parent.Controls.Add(panel);

Controls.Add(parent);

}

**Code Snippet**: *My Code Snippets | spchol307\_ex1\_webpart\_createchildcontrols*

1. Add the following methods below **CreateChildControls**:

private void PopulateProjects()

{

SPList splProjects = SPContext.Current.Web.Lists["Projects"];

\_ddlProjects.Items.Add("-- Select a Project --");

foreach (SPListItem spli in splProjects.Items)

{

\_ddlProjects.Items.Add(new ListItem(spli.Title, spli.ID.ToString()));

}

}

void ddlProjects\_SelectedIndexChanged(object sender, EventArgs e)

{

GetProjectDetails();

}

/\* Update the current project \*/

void bUpdateProject\_Click(object sender, EventArgs e)

{

EnsureChildControls();

int nProjectID = Convert.ToInt32(\_ddlProjects.SelectedValue);

SPListItem spliProject =

SPContext.Current.Web.Lists["Projects"].GetItemById(nProjectID);

spliProject["Description"] = \_tbDescription.Text;

spliProject["Due\_x0020\_Date"] = \_tbDueDate.Text;

spliProject.Update();

}

**Code Snippet**: *My Code Snippets | spchol307\_ex1\_webpart\_populateprojects*

### Task 3 – Build and Deploy the Sandboxed Solution

A sandboxed solution is deployed using the SharePoint web site.

1. Right-click the **SBSolutionDemo** project and select the **Package** to create a .wsp file.
2. Open Internet Explorer and browse to <http://intranet.contoso.com>.
3. Open the **Site Actions** menu and select **Site Settings**.
4. Under the **Galleries** section select **Solutions**.

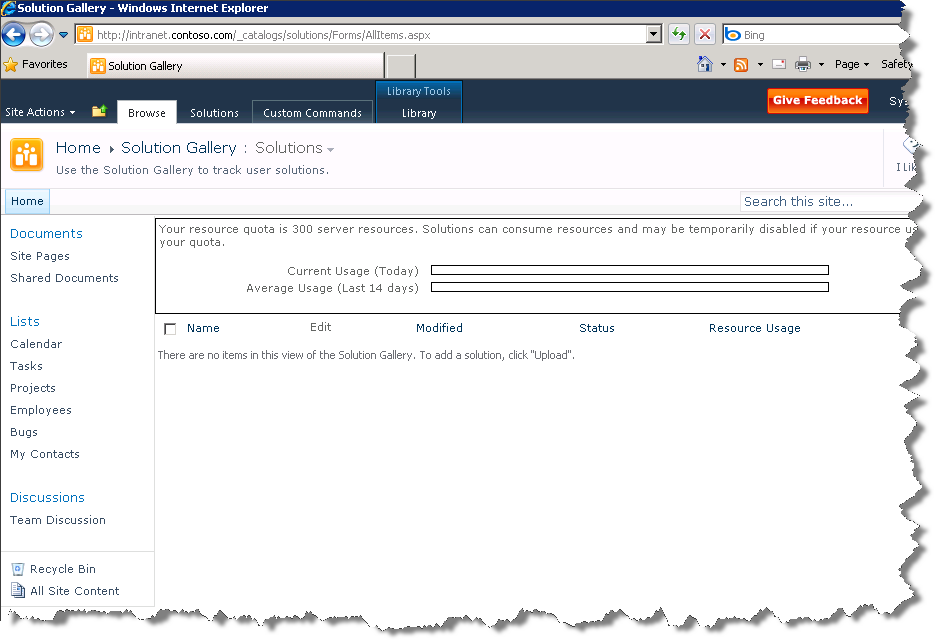


Figure 7 - Solutions Gallery

1. Select the **Solutions** tab.
2. On the Solutions tab, select **Upload Solution**.
3. In the Upload Document dialog that pops up, browse to the .wsp file at **C:\SPHOLs\SPCHOL307\CS\Ex1\SBSolutionDemo\SBSolutionDemo\bin\Debug\SBSolutionDemo.wsp**.
4. Click **OK** to upload the SBSolutionDemo.wsp to SharePoint. Leave the **Overwrite existing files** box checked.
5. Click **Activate** in the Solution Gallery - Activate Solution dialog. The Sandboxed Solution web part is now ready to be used.
6. Open the **Site Actions** menu and select **More Options**.
7. Under the **Page** section of the Create dialog, select **Web Part Page**. Click **Create**.
8. Name the new Web Part Page **SBSolutionDemoPage**, set the Layout to **Full Page, Vertical**. And the Save Location to **Shared Documents**.

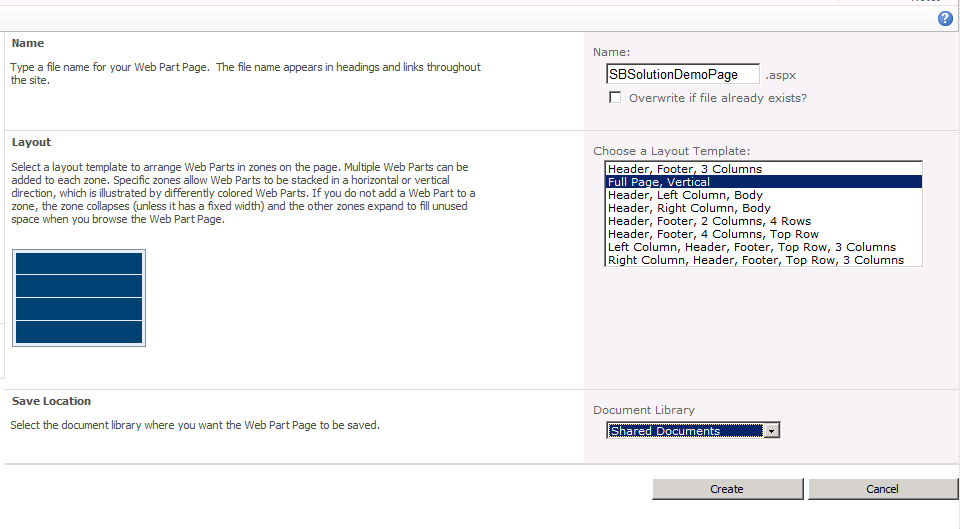


Figure 8 - New Web Part Page

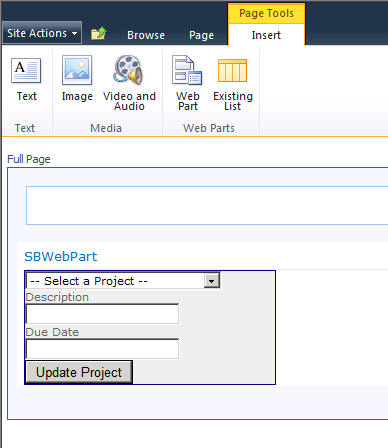
1. Click **Create** to create the new Web Part Page.
2. Select the middle area of the new Web Part Page and then click the new **Insert** tab that appears in the top toolbar.
3. Select **Web Part** and then under Categories **Custom** and under Web Parts choose **SBWebPart**.
4. Click **Add** to add the Sandboxed Solution web part to the page.
5. In the Ribbon, click **Page**. Next, click **Stop Editing** in the toolbar  
     
   

Figure 9 - SBWebPart

1. Select **System Account** in the toolbar at the top right hand side of the page and choose **Sign in as Different User**.
2. For the User name enter **andyj** and for the password enter **pass@word1**.



Figure 10 - Windows Login Dialog

1. Click **OK**.
2. The Sandboxed Solution web part is now ready to use. Select different things from the drop-down to see. When you are finished, close Internet Explorer.

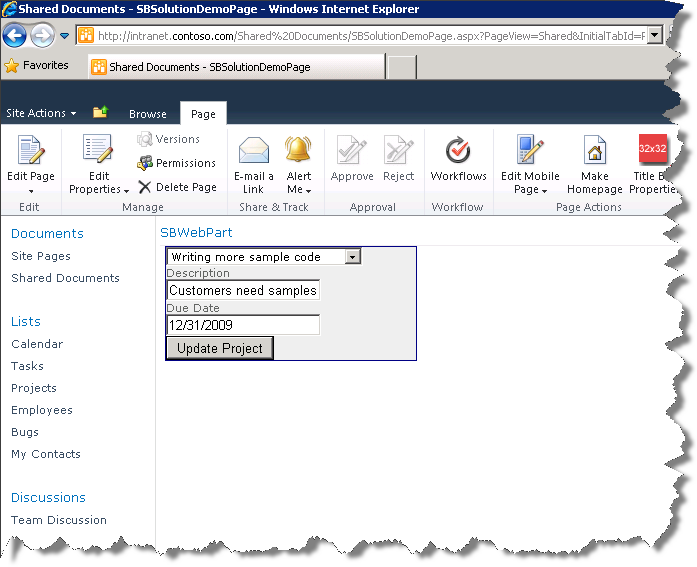


Figure 11 - The Sandboxed Solution Web Part

## Exercise 2 – Creating a Sandboxed Solution Web Part with overload code

In exercise two we will again create a Sandboxed Solution web part. This web part will be designed to use a high amount of CPU resources to demonstrate SharePoint’s resource quota system.

### Task 1 – Create a Sandboxed Solution Project with a web part

We will start by creating a standard Silverlight application project.

1. Open Visual Studio 2010 from **Start | All Programs | Microsoft Visual Studio 2010 | Microsoft Visual Studio 2010**.
2. From the menu, select **File | New | Project**.
3. In the New Project dialog box, expand the Installed Templates left hand menu to **Visual C# | SharePoint | 2010 | Empty SharePoint Project**.
4. Name the project **SBSolutionOverload**.
5. Change the location to **C:\SPHOLS\SPCHOL307\CS\Ex2\**

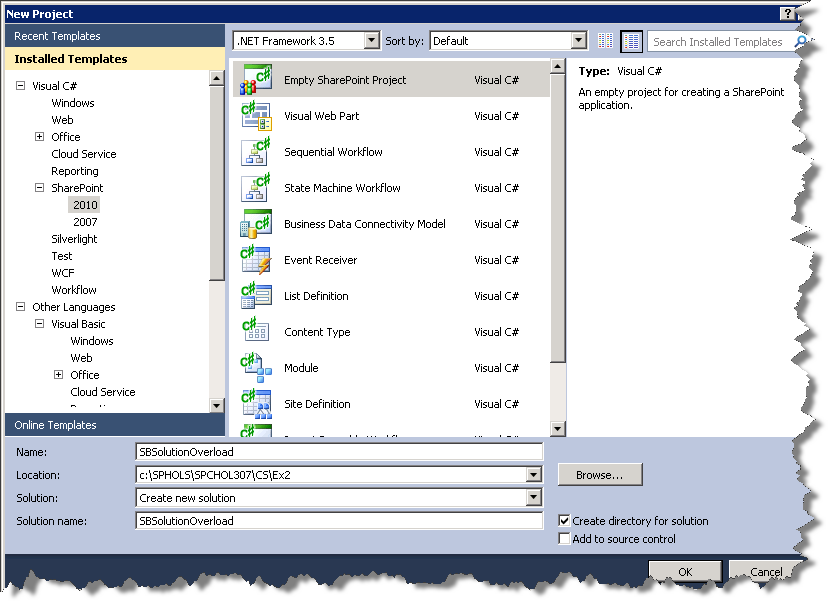


Figure 12 - New Project Dialog

1. Press **OK** to continue.
2. Change the local site to use for debugging to **http://intranet.contoso.com/**.
3. Leave the trust level for the SharePoint solution as **Deploy as a sandboxed solution**.

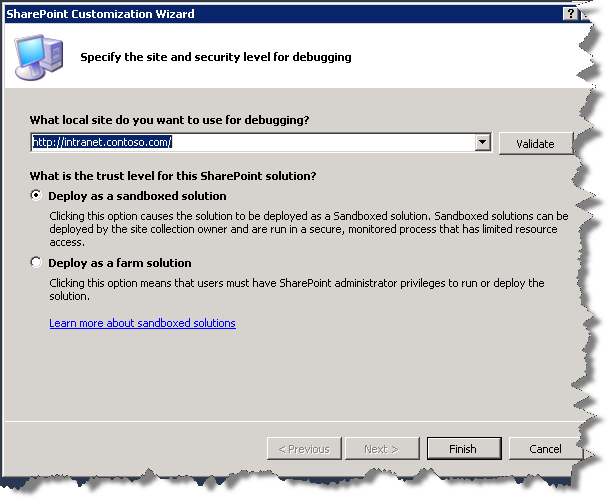


Figure 13 - SharePoint Customization Wizard Step1.

1. Press **Finish** to continue. Visual Studio will create the new project and add the necessary files.
2. Right-click on the **SBSolutionOverload** project and select **Add | New item**.
3. In the Add New Item dialog, ensure **Visual C# | SharePoint | 2010 |** is selected in the Installed Templates pane.
4. In the middle pane, select **Web Part** and name it **SBWebPart**.

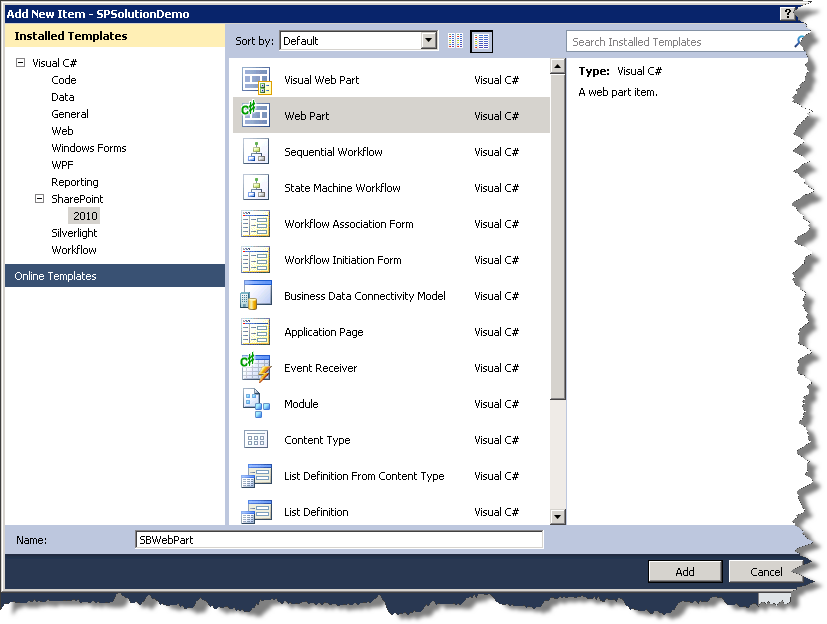


Figure 14 - Add New Item Dialog

1. Press **Add** to add the web part to the project.

### Task 2 – Add code to overload the Sandboxed Solution web part

The web part we are creating will have a button, text box, and label. Clicking the button will cause SharePoint to do work for the number of seconds specified in the text box.

1. Open **SBWebPart.cs** and replace the **CreateChildControls** method with the following code:

protected override void CreateChildControls()

{

base.CreateChildControls();

Label lbl = new Label();

TextBox txt = new TextBox();

Button btn = new Button();

DateTime dt = DateTime.Now;

btn.Text = "Show All Lists";

btn.Click += delegate

{

do

{

SPWebCollection Webs;

SPListCollection lists;

Webs = SPContext.Current.Site.AllWebs;

foreach (SPWeb web in Webs)

{

lists = web.Lists;

foreach (SPList list in lists)

lbl.Text = lbl.Text + "<br>" + list.Title;

}

}

while (dt.AddSeconds(int.Parse(txt.Text)).CompareTo(DateTime.Now) > 0);

};

Controls.Add(txt);

Controls.Add(btn);

Controls.Add(lbl);

}

**Code Snippet**: *My Code Snippets | spchol307\_ex2\_webpart\_createchildcontrols*

### Task 3 – Set the SharePoint Quotas and Limits

SharePoint allows you to configure quotas and limits on Sandboxed Solutions. We will now enter the Central Administration site and configure a web applications quota.

1. Open Internet Explorer and browse to <http://demo2010a:2010/>.

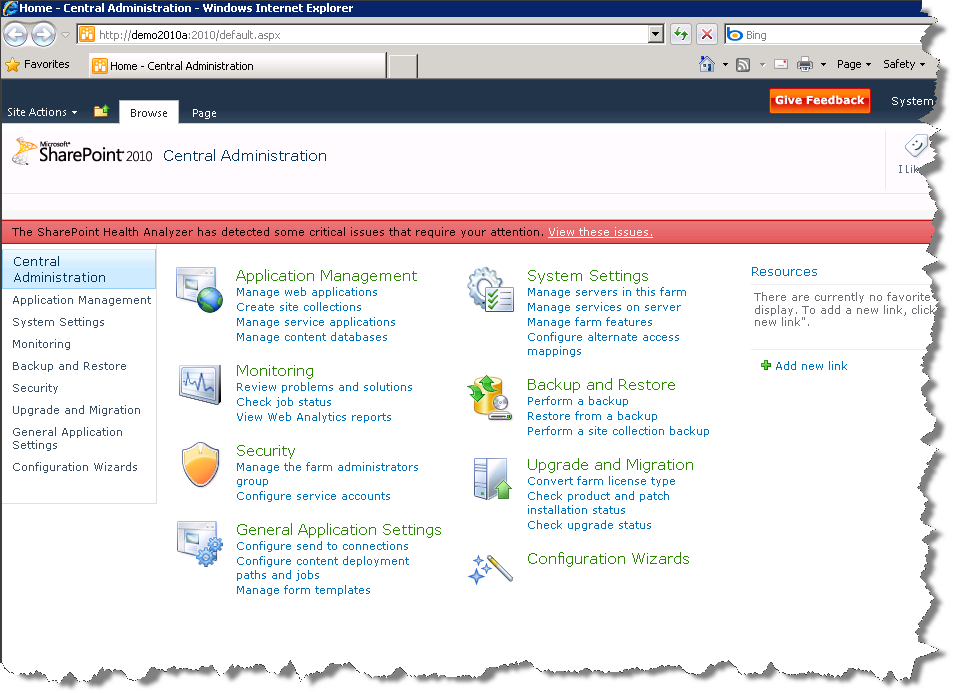


Figure 15 - Central Administration Page

1. On the Central Administration homepage select **Application Management**.
2. Under the **Site Collections** section select **Specify quota templates**.
3. Select the **Create a new quota template** radio button and name the new template **SBSolutionOverloadTemplate**.
4. Set the site storage limit to a maximum of **200** megabytes and the storage warning email to **10** megabytes.
5. Check the checkbox in the **Sandboxed Solutions With Code Limits** section.
6. Set the limit maximum usage per day to **5** points and the warning email to **5** points.

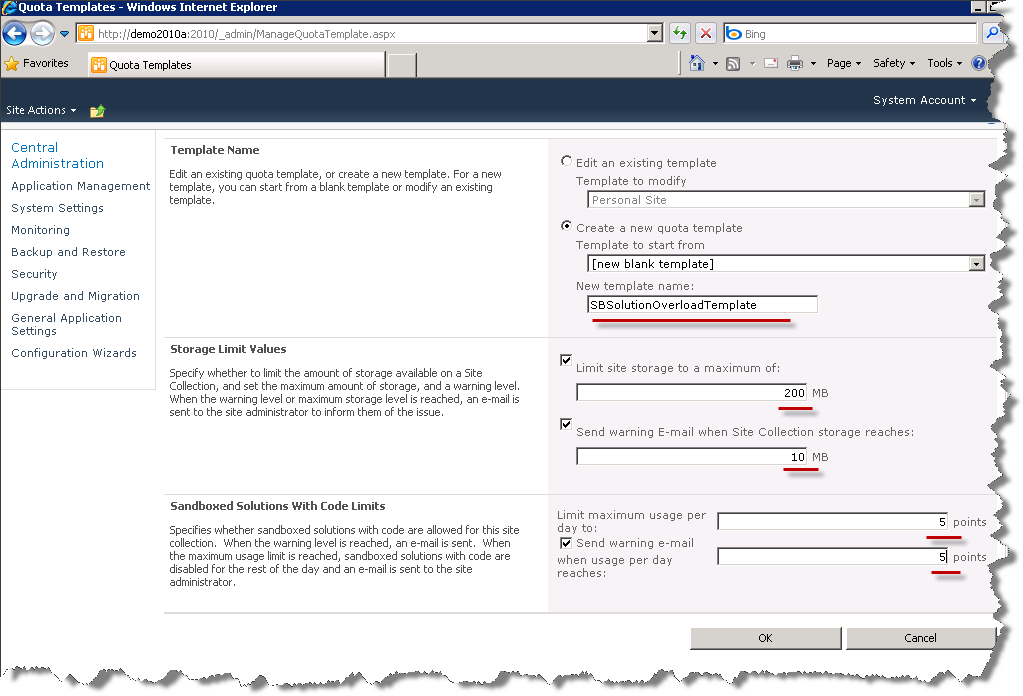


Figure 16 - Quota Templates Page

1. Click **OK** to create the quota template.
2. In the **Application Management** page, under the **Site Collections** section, select **Configure quotas and locks**.
3. Click the **Site Collection** dropdown link and select **Change Site Collection**.
4. Click the **Web Application** dropdown link and select **Change Web Application**.
5. In the Select Web Application dialog select <http://intranet.contoso.com>.

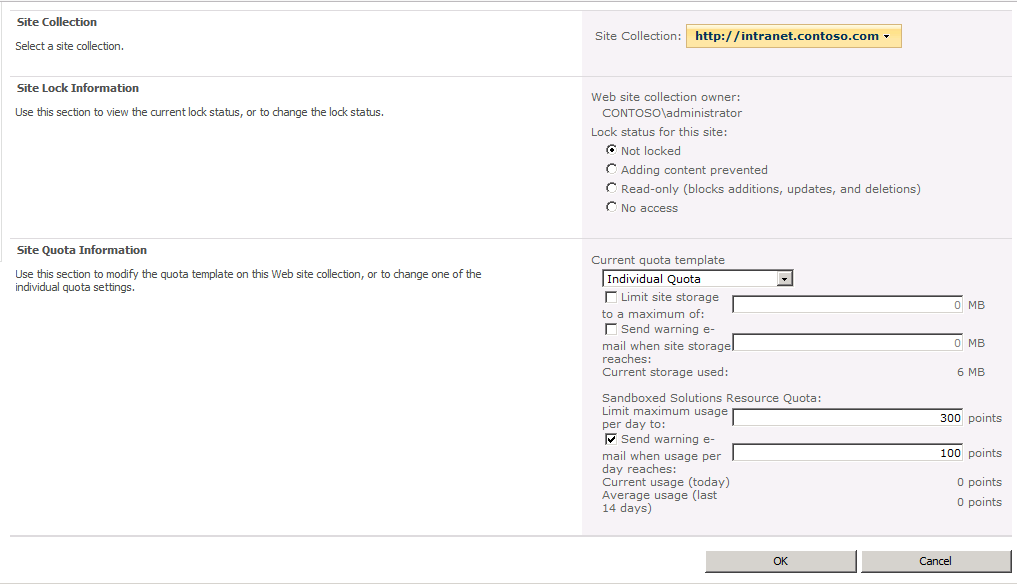


Figure 17 - Select Web Application Dialog

1. In the Select Web Application, dialog select [**http://intranet.contoso.com**](http://intranet.contoso.com) and click **OK**.
2. In the **Current quota template** dropdown, pick **SBSolutionOverloadTemplate**.

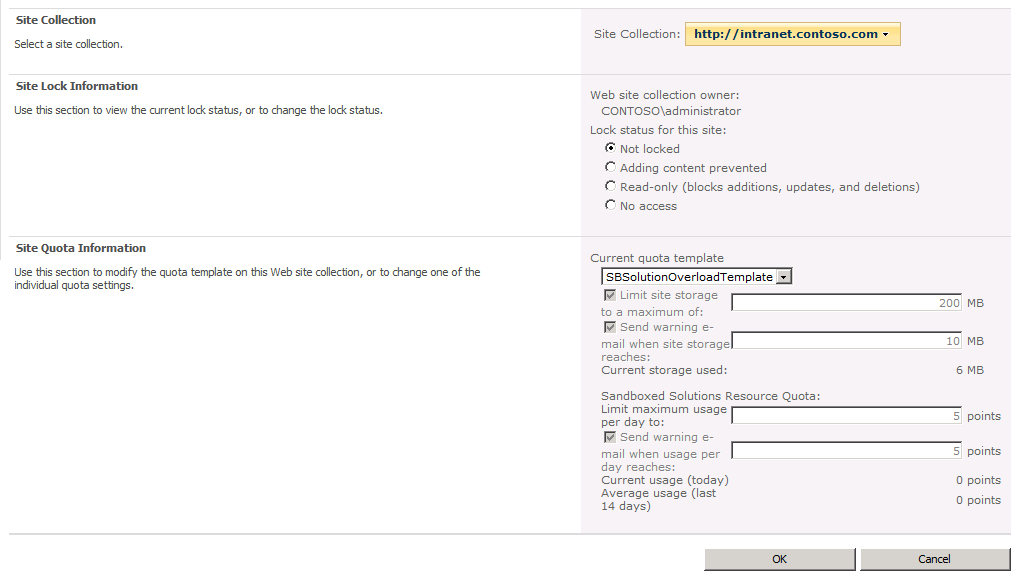


Figure 18 - Site Collection Quotas and Locks Page

1. Click **OK** to set the site collection’s quota information.

### Task 4 – Build and Deploy the Sandboxed Solution

A sandboxed solution is deployed using the SharePoint web site.

1. Return to Visual Studio, right-click the **SBSolutionOverload** project and select the **Package** to create a wsp file.
2. Open Internet Explorer and browse to <http://intranet.contoso.com>.
3. Open the **Site Actions** menu and select **Site Settings**.
4. Under the **Galleries** section, select **Solutions**.
5. On the top left, click **Solutions** (even though you are already on that page, this opens additional Ribbon options).

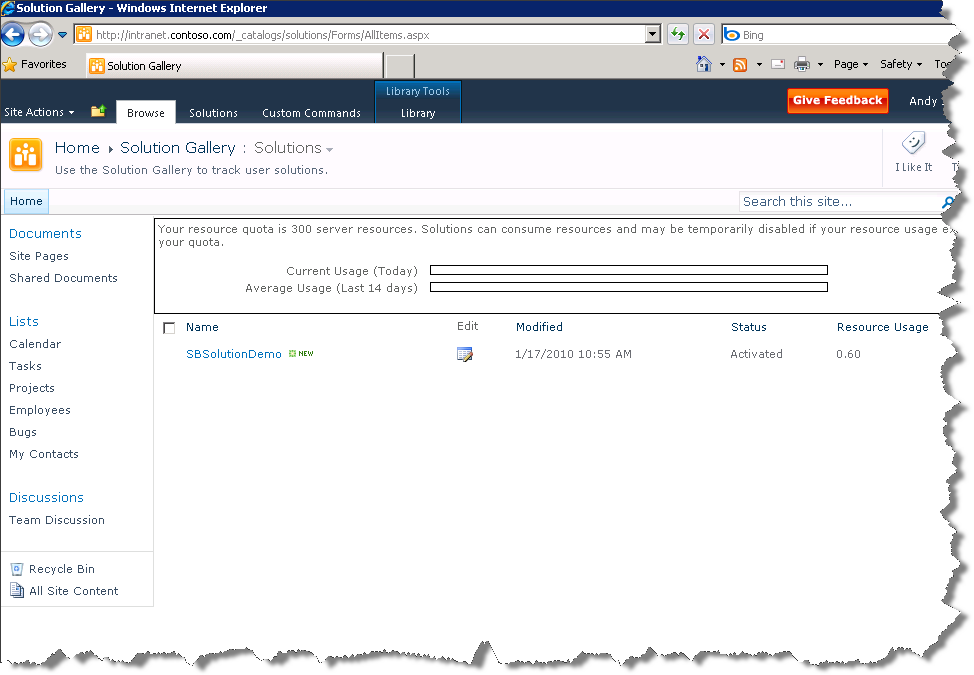


Figure 19 - Solution Gallery Page

1. On the **Solutions** page Solutions Ribbon, select **Upload Solution**.
2. In the Upload Document dialog that pops up, browse to the .wsp file at **C:\SPHOLs\SPCHOL307\CS\Ex2\SBSolutionOverload\SBSolutionOverload\bin\Debug\SBSolutionOverload.wsp**.
3. Click **OK** to upload the SBSolutionOverload.wsp to SharePoint. Leave the **Overwrite existing files** box checked.
4. Click **Activate** in the **Activate Solution** dialog. The Sandboxed Solution web part is now ready to be used.
5. Open the **Site Actions** menu and select **More Options..**.
6. Select **Page** from the left column, then select **Web Part Page** from the right pane, and click **Create**.
7. Name the new Web Part Page **SBSolutionOverloadPage**, set the Layout to **Full Page, Vertical**, and the Save Location to **Shared Documents**.

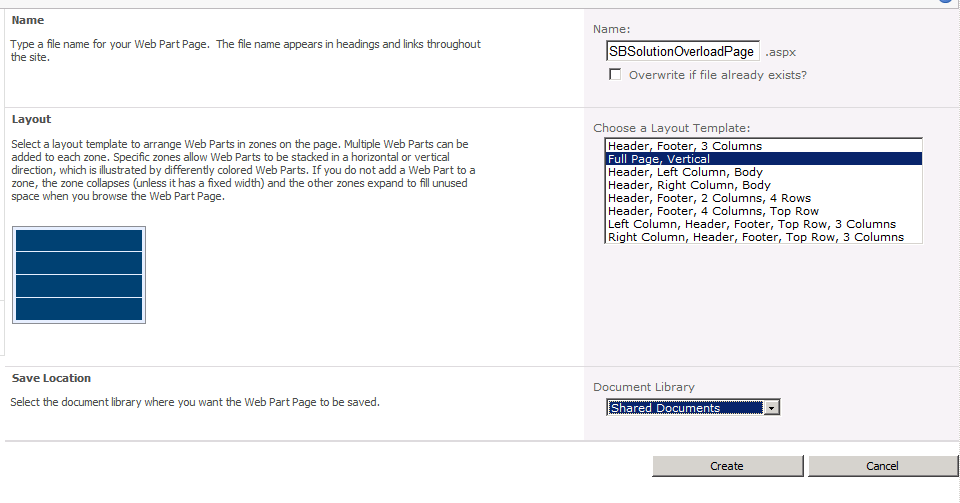


Figure 20 - New Web Part Page

1. Click **Create** to create the new Web Part Page.
2. Select the middle area of the new Web Part Page and then click the new Insert tab that appears in the top toolbar.
3. Select **Web Part** and then under Categories **Custom** and under Web Parts choose **SBWebPart**.
4. Click **Add** to add the Sandboxed Solution web part to the page.
5. Click on the **Page** tab in the Ribbon and click on **Stop Editing** button.
6. The Sandboxed Solution web part is now ready to use.

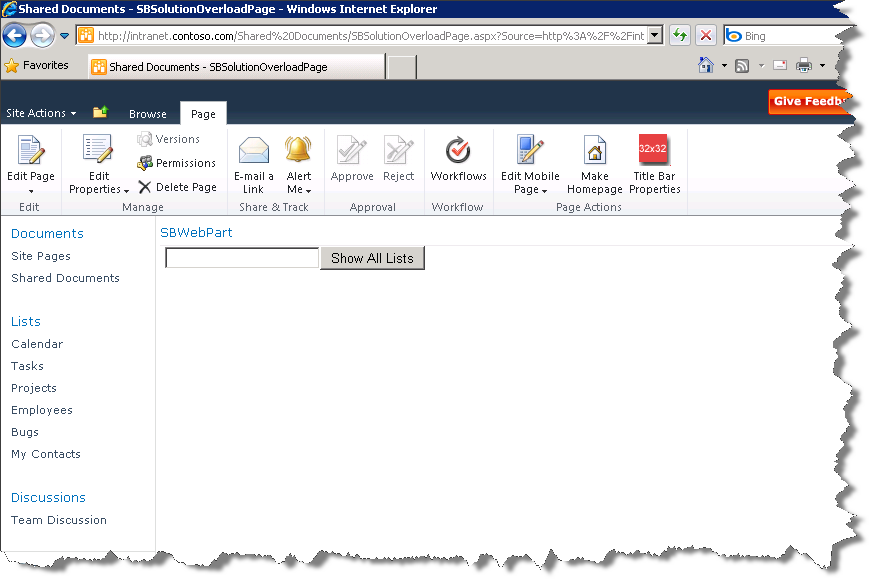


Figure 21 - The Sandboxed Solution Overload Web Part

1. Enter the number of seconds the web part should work for in the text box and then click **Show All Lists**. A high number entered in the textbox will result in SharePoint stopping execution of the web part.

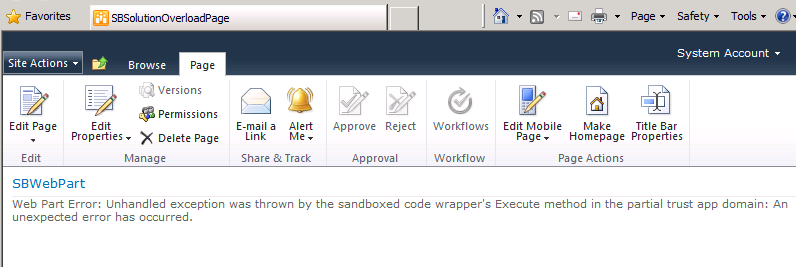


Figure 22 - Overload Web Part timing out

## Lab Summary

In this lab you performed the following exercises.

Created a Sandboxed Solution Web Part that read from and updated the local web site

Deployed a Sandboxed Solution Web Part

Configured quotas and limits for SharePoint web site

Created a Sandboxed Solution Web Part that exceeds quotas and is consequently disabled