Demo outline

In this demo you will go through tasks required to create a SharePoint Project Items like Web parts and Visual Web Parts using the SharePoint Tools templates.

The provided sample code is the final solution that will be created during this demo. Also, this demo project is based on the SharePoint Tools Project template demo included in this Training Kit.

**Run Source\Setup\Setup.bat to create the required SharePoint site for this demo.**

### Steps:

1. Open the MetroWebParts project and inspect the existing items. If required, set the Site Url property of the project in the Properties window.
2. Add the Images SharePoint mapped folder to the project by right-clicking on the project node and selecting the Add SharePoint “Images” Mapped Folder.
   1. This will automatically create the MetroWebParts folder.
   2. Right-click on the MetroWebParts folder and select Add Existing Item and select, from the Assets folder of this demo, the FeatureIcon.gif and WebPartIcon.gif image files.
3. Open the feature included in the Features folder by right-clicking on it and opening the Feature designer.
   1. In the Properties window, edit the image Url property and set it to MetroWebParts\FeatureIcon.gif to use the recently added image file.
4. Add a new SharePoint Web Part by right-clicking on the project node and using the Add New item context menu.
   1. Edit the elements.xml and change the Group property of the File element to something more meaningful like “MetroWebParts”
   2. Edit the .webpart file of the Web Part and modify the Title and Description properties.
   3. Include the ChromeType, CatalogIconImageUrl and TitleIconImageUrl properties.

<property name="ChromeType" type="chrometype">TitleAndBorder</property>

<property name="CatalogIconImageUrl" type="string">\_layouts/images/MetroWebParts/WebPartIcon.gif</property>

<property name="TitleIconImageUrl" type="string">\_layouts/images/MetroWebParts/WebPartIcon.gif</property>

1. Add a new SharePoint Visual Web Part by right-clicking on the project node and using the Add New Item context menu.
   1. Do the same as with the Web Part previously created.
      * Edit elements.xml file, edit .webpart file modifying existing properties and adding new properties
   2. Add a SharePoint TreeView to the user control included in the template. (Code Snippets at the end of the current file)
   3. Override and Implement the OnPreRender method to load the TreeView with the hierarchical virtual file system for the current site. (Code Snippets at the end of the current file)
   4. Import the Microsoft.Sharepoint namespace in the user control’s class if required.
2. Change the Active Deployment Configuration to No Activation using the SharePoint tab in the Project properties page.
   1. Right-click the MetroWebParts project node and click Deploy. Review the Output window to see the details.
3. Open Internet Explorer and browse to the configured SharePoint site’s home page. For example <http://localhost/sites/SharePointToolsDemo>
4. Activate the feature in the Site by using the Site Settings - Site Collection Administration page in SharePoint. Deactivate the feature first if it is already activated.
5. Using the Page ribbon in the site’s home page to use the Edit Page command and add a new web part.
   1. Select the MetroWebParts category and check the two created Web Parts are present.
   2. Add both web parts and the use the Stop Editing command to check the new site’s home page look and feel.

<SharePoint:SPTreeView ID="treeSiteFiles" runat="server" ImageSet="XPFileExplorer" NodeIndent="15">

<HoverNodeStyle Font-Underline="True" ForeColor="#6666AA" />

<NodeStyle Font-Names="Tahoma" Font-Size="8pt" ForeColor="Black" HorizontalPadding="2px"

NodeSpacing="0px" VerticalPadding="2px" />

<ParentNodeStyle Font-Bold="False" />

<SelectedNodeStyle BackColor="#B5B5B5" Font-Underline="False" HorizontalPadding="0px"

VerticalPadding="0px" />

</SharePoint:SPTreeView>

C#

const string SITE\_IMG = @"\\_layouts\images\FPWEB16.GIF";

const string FOLDER\_IMG = @"\\_layouts\images\FOLDER16.GIF";

const string GHOSTED\_FILE\_IMG = @"\\_layouts\images\NEWDOC.GIF";

const string UNGHOSTED\_FILE\_IMG = @"\\_layouts\images\RAT16.GIF";

protected override void OnPreRender(EventArgs e)

{

treeSiteFiles.Nodes.Clear();

SPWeb site = SPContext.Current.Web;

SPFolder rootFolder = site.RootFolder;

TreeNode rootNode = new TreeNode(site.Url, site.Url, SITE\_IMG);

LoadFolderNodes(rootFolder, rootNode);

treeSiteFiles.Nodes.Add(rootNode);

treeSiteFiles.ExpandDepth = 1;

}

protected void LoadFolderNodes(SPFolder folder, TreeNode folderNode)

{

foreach (SPFolder childFolder in folder.SubFolders)

{

TreeNode childFolderNode = new TreeNode(childFolder.Name, childFolder.Name, FOLDER\_IMG);

childFolderNode.NavigateUrl = SPContext.Current.Site.MakeFullUrl(childFolder.Url);

LoadFolderNodes(childFolder, childFolderNode);

folderNode.ChildNodes.Add(childFolderNode);

}

foreach (SPFile file in folder.Files)

{

TreeNode fileNode;

if (file.CustomizedPageStatus == SPCustomizedPageStatus.Uncustomized)

{

fileNode = new TreeNode(file.Name, file.Name, GHOSTED\_FILE\_IMG);

}

else

{

fileNode = new TreeNode(file.Name, file.Name, UNGHOSTED\_FILE\_IMG);

}

fileNode.NavigateUrl = SPContext.Current.Site.MakeFullUrl(file.Url);

folderNode.ChildNodes.Add(fileNode);

}

}

Visual Basic

Const SITE\_IMG As String = "\\_layouts\images\FPWEB16.GIF"

Const FOLDER\_IMG As String = "\\_layouts\images\FOLDER16.GIF"

Const GHOSTED\_FILE\_IMG As String = "\\_layouts\images\NEWDOC.GIF"

Const UNGHOSTED\_FILE\_IMG As String = "\\_layouts\images\RAT16.GIF"

Protected Overrides Sub OnPreRender(ByVal e As System.EventArgs)

treeSiteFiles.Nodes.Clear()

Dim site = SPContext.Current.Web

Dim rootFolder = site.RootFolder

Dim rootNode = New TreeNode(site.Url, site.Url, SITE\_IMG)

LoadFolderNodes(rootFolder, rootNode)

treeSiteFiles.Nodes.Add(rootNode)

treeSiteFiles.ExpandDepth = 1

End Sub

Protected Sub LoadFolderNodes(ByVal folder As SPFolder, ByVal folderNode As TreeNode)

For Each childFolder As SPFolder In folder.SubFolders

Dim childFolderNode = New TreeNode(childFolder.Name, childFolder.Name, FOLDER\_IMG)

childFolderNode.NavigateUrl = SPContext.Current.Site.MakeFullUrl(childFolder.Url)

LoadFolderNodes(childFolder, childFolderNode)

folderNode.ChildNodes.Add(childFolderNode)

Next

For Each file As SPFile In folder.Files

Dim fileNode As TreeNode

If (file.CustomizedPageStatus = SPCustomizedPageStatus.Uncustomized) Then

fileNode = New TreeNode(file.Name, file.Name, GHOSTED\_FILE\_IMG)

Else

fileNode = New TreeNode(file.Name, file.Name, UNGHOSTED\_FILE\_IMG)

End If

fileNode.NavigateUrl = SPContext.Current.Site.MakeFullUrl(file.Url)

folderNode.ChildNodes.Add(fileNode)

Next

End Sub