**Microsoft Office SharePoint Server (MOSS) for Developers**

**(Part 1)**

**Topic 1: Introduction to Web Parts**



[**http://msdn.microsoft.com/rampup**](http://msdn.microsoft.com/rampup)

**Introduction**

Microsoft ASP.NET 2.0 provides a Web Part infrastructure that helps you build custom Web Parts and deploy them to Web sites built using Microsoft Windows SharePoint Services 3.0.

This article provides recommendations for best practices to choose between ASP.NET 2.0 Web Parts and Windows SharePoint Services 3.0 Web Parts according to business needs, reviews ASP.NET 2.0 Web Part infrastructure, and walks you through how to build custom Web Parts that work in both standard ASP.NET 2.0 Web sites and Windows SharePoint Services 3.0. The article also outlines deployment and security considerations to address before adding Web Parts to SharePoint sites.

You will need to install the [**Windows SharePoint Services 3.0 SDK**](http://msdn.microsoft.com/en-us/library/ms441339.aspx)

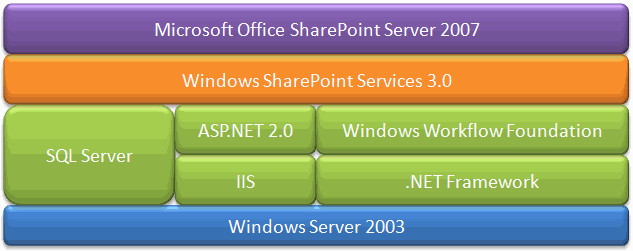
**Web Part Frameworks**

SharePoint Web Parts are UI elements that support both customization and personalization. A *Web Part* is an ASP.NET server control designed to be edited and modified by knowledge workers in the browser. It works as a component of a SharePoint site that presents information pulled from multiple data sources. With Web Parts, you can create information dashboards on corporate portals and Web sites. Web Part pages allow you to personalize information relevant to your needs by updating the value of specific Web Part properties.

The Web Part infrastructure in Microsoft Windows SharePoint Services 2.0 and Microsoft Office SharePoint Portal Server 2003 allowed you to build custom Web Parts targeted to Microsoft SharePoint Products and Technologies by using managed code.

ASP.NET 2.0 includes a new Web Part control set that allows site customization and personalization on custom Web sites that are independent of SharePoint Products and Technologies.

The Web Part infrastructure in Windows SharePoint Services 3.0 exists on a layer above the ASP.NET 2.0 Web Part infrastructure. This introduces a next-generation Web Part infrastructure for the Microsoft Windows Server 2003 platform as shown in the following figure.



**Figure 1. ASP.NET 2.0 Web Part infrastructure**

**Choosing Between ASP.NET 2.0 Web Parts and Windows SharePoint Services 3.0 Web Parts**

Office SharePoint Server 2007 and Windows SharePoint Services 3.0 are built on top of the ASP.NET 2.0 Framework. Windows SharePoint Services 3.0 uses the ASP.NET 2.0 Web Part infrastructure to take advantage of features such as master pages and custom Web Part development while providing a complete set of Windows SharePoint Services features, such as document services, events, workflow, search, site columns, content types and more.

You can build Web Parts for Windows SharePoint Services 3.0 in two ways:

* Create custom ASP.NET 2.0 Web Parts.
* Create SharePoint-based Web Parts.

You should create ASP.NET 2.0 Web Parts whenever you can. However, there are a few exceptions where using SharePoint-based Web Parts might offer advantages... The following table provides a decision matrix to help you choose the best option depending on your business needs.

|  |  |
| --- | --- |
| **Create a custom ASP.NET 2.0 Web Part** | **Create a SharePoint-based Web Part** |
| * For most business needs. * To distribute your Web Part to sites that run ASP.NET 2.0 or SharePoint sites. * When you want to reuse one or more Web Parts created for ASP.NET 2.0 sites on SharePoint sites. * To use data or functionality provided by Windows SharePoint Services 3.0. For example, you are creating a Web Part that works with site or list data. | * When you want to migrate a set of Web Parts using the SharePoint-based Web Part infrastructure to Windows SharePoint Services 3.0. * To create cross page connections. * To create connections between Web Parts that are outside of a Web Part zone. * To work with client-side connections (Web Part Page Services Component). * To use a data-caching infrastructure that allows caching to the content database. |

**Table 1. Decision matrix for creating Web Parts**

# **Creating a New Web Part Solution**

The Web Part project template included in the extensions is customized especially for creating Web Parts for the Windows SharePoint Services environment.

### To create a new Web Part solution

1. Open Microsoft Visual Studio 2008.
2. On the **File** menu, click **New**, and then select **Project**.
3. In **Project types**, select **Visual C#**, and then select **SharePoint**.
4. In **Templates**, select **Web Part**.
5. Specify a new name, location, or solution name for your Web Part, and then click **OK**. The extensions create a new Web Part solution project, which includes the following:
   * References to the necessary DLLs
   * AssemblyInfo.cs, a file that enables you to specify company and product information for the Web Part assembly, and version information
   * Temporary.snk, a temporary signature key file for the Web Part assembly
   * A Visual C# code file for the Web Part, named the same as the Web Part name

By default, the code file for the Web Part contains the following:

* + **using** statements for the necessary **System** and Windows SharePoint Services namespaces, such as **Microsoft.SharePoint**, **Microsoft.SharePoint.WebControls**, and **Microsoft.SharePoint.WebPartPages**
  + An empty Web Part class that inherits from System.Web.UI.WebControls.WebParts.WebPart

1. Add the code you want to the Web Part class.

# 

# **Customizing the Web Part Solution Package**

The Web Part project template automatically generates the XML files needed to package your Web Part as a Windows SharePoint Services 3.0 Feature for deployment and activation. The required GUIDs contained in the XML files are also generated automatically.

The extensions generate a .webpart file as the element.xml file for the Feature containing the Web Part. Every Web Part should have a .webpart file, which is an XML file that describes the Web Part. The .webpart file also makes your Web Part appear in the Web Part gallery in Windows SharePoint Services.

You can customize the information in these XML files by editing solution, Feature, and element properties on the **SharePoint Solution** tab in your project's **Properties** dialog box, as follows:

* **Solution**   Information in this node is written into the manifest.xml file for the Feature. You can set the following solution property:
  + **Name**   The name of your Web Part solution.
* **Feature**   Information in this node is written into the feature.xml file for the Feature. You can set the following feature properties:
  + **Folder Name**   The name of the folder to contain the files for this Feature.
  + **Title**   The title of the Feature. Limited to 255 characters.
  + **Description**   A longer representation of what the Feature does.
  + **Version**   Specifies a **System.Version**-compliant representation of the version of a Feature. This can be up to four numbers delimited by decimals that represent a version.
  + **Scope**   The Feature scope. Can contain one of the following values: **Farm** (farm), **WebApplication** (Web application), **Site** (site collection), or **Web** (Web site).
  + **Hidden**   Hides the feature. This attribute is **FALSE** by default.
  + **Default Resource File**   Indicates a common resource file for retrieving Feature XML resources.
* **Element**   Information in this node is written into the element.xml file for the Feature. As mentioned earlier, this file contains the information usually contained in a .webpart file. You can set the following element properties:
  + **Title**   The title of your Web Part as you want it to appear in the Windows SharePoint Services user interface.
  + **Description**   The description of your Web Part that you want to appear in the Windows SharePoint Services user interface.
  + **ImportErrorMessage**   The error string to display if Windows SharePoint Services is unable to import your Web Part solution.

### Customizing the Web Part solution feature package

1. In Solution Explorer, right-click the Web Part project, and then select **Properties**.
2. In the Properties window, select **SharePoint Solution**.

This tab lists the nodes that contain the properties for the solution package that the extensions will generate.

1. Expand the **Solution** node, and then expand the **Feature** node and the **Element** node.
2. To edit properties for a node, click a node, and then edit the properties you want.

For example, click the **Element** node, and then edit the values for the **Title**, **Description**, and **ImportErrorMessage** keys.

1. To commit your changes for a node, click in the **Description** value field.

# 

# **Deploying and Debugging the Web Part**

You can deploy and start debugging your Web Part simply by pressing F5.

When you press F5, the extensions automatically build, deploy, and initiate debugging of your Web Part solution Feature. To accomplish this, the extensions perform the following actions automatically:

* Build the Web Part assembly.
* Package the Web Part solution as a Windows SharePoint Services Feature.
* If this Web Part solution was deployed previously, the extensions retract the previous version of the Web Part solution Feature.
* Install your Web Part assembly to the global assembly cache.
* Deploy and activate your Web Part solution Feature in Windows SharePoint Services.
* Add your Web Part to the SafeControls list.
* Restart Microsoft Internet Information Services (IIS) by invoking the **iisreset** command.
* Attach to the **w3wp** processes to enable debugging.

### Deploying and debugging your Web Part

1. Press F5. The extensions automatically build, deploy, and initiate debugging of your Web Part solution Feature.
2. Navigate to the Web Part Page on your SharePoint site where you want to add the Web Part.
3. In the Web Part page, click **Site Actions**, and then select **Edit Page**.
4. In the Web Part zone where you want to add your Web Part, click **Add a Web Part**.
5. In the **Add Web Parts** dialog box, in the **All Web Parts** section, under **Miscellaneous**, select your Web Part and then click **OK**.

# **Additional resources and references**

* [Working with ASP.NET 2.0 Web Parts and Windows SharePoint Services 3.0](http://msdn.microsoft.com/en-us/library/bb153523.aspx)
* [Creating a Windows SharePoint Services 3.0 Web Part Using Visual Studio](http://msdn.microsoft.com/en-us/library/aa973249.aspx)
* [Introduction to SharePoint Products and Technologies for the Professional .NET Developer](http://msdn.microsoft.com/en-us/library/cc537498.aspx)
* [Windows SharePoint Services Developer Center](http://msdn2.microsoft.com/en-us/SharePoint)
* [Windows SharePoint Services TechCenter](http://technet.microsoft.com/en-us/office/sharepointserver/default.aspx)
* [SharePoint Server Developer Center](http://msdn2.microsoft.com/en-us/office/aa905503.aspx)
* [SharePoint Server TechCenter](http://technet.microsoft.com/en-us/windowsserver/sharepoint/default.aspx)