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

**Topic 6: Page Navigation**





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

**Introduction**

SharePoint is built right on top of .NET framework 3.0. The previous version of SharePoint was not set up this way from an architectural standpoint. Instead, it was attached into ASP.NET. But thanks to a lot of additions, or a lot of changes that the .NET team did in the .NET 30 release, specifically ASP.NET 2.0, allowed the SharePoint team to simply change the way ASP.NET works in the context of a SharePoint site. SharePoint navigation is much more of a pluggable infrastructure, and makes applications more scaleable.

When discussing page navigation most web sites employ some form of visual navigation to help users move around the site easily, and find the information and Web pages they require. Though the look and feel can vary widely from site to site, the same basic elements are usually present, in the form of navigation bars, or menu lists, that direct users to specific parts of the web site.

ASP.NET 1.x out of the box provided little in the way of support for site navigation, leading many developers and web designers to either build their own navigation system, or buy third-party controls to meet their requirements. All this changes with ASP.NET 2.0, which introduces a navigation system that uses a pluggable framework for exposing the site hierarchy, and controls that plug into this new model, making it easy to build a high quality menu and navigation system.

This paper describes how the navigation system in ASP.NET 2.0 works, and shows how it can be extended beyond simple XML files, which is the default mechanism used in Visual Studio 2008.

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

**Understanding the Navigation System in ASP.NET 2.0**

In addition to creating a compelling navigation model that would appeal to both developers and web site designers, one of the other goals for the ASP.NET 2.0 navigation system was to create an architecture that provided an extensibility capability that was flexible enough to address a wide range of needs. It is based on a provider model that is used throughout the ASP.NET 2.0 framework, which provides a standard mechanism for plugging in different data sources.

The ASP.NET 2.0 navigation framework can be broken down into a couple of areas:

* The web navigation controls (Menu, TreeView, and SiteMapPath) that developers use on the actual web pages. These can be customized to change their look and feel.
* The SiteMapDataSource control, which the TreeView and Menu navigation controls bind to, providing an abstract layer between the Web navigation controls and the underlying provider of the navigation information.
* The Site Map Provider, which is the pluggable provider that exposes the actual information that describes the layout of the web site. ASP.NET ships with one provider, the XmlSiteMapProvider, which uses an XML file with a specific schema as its data store

This layered architecture creates a looser coupling between the underlying site hierarchy and the controls on the web site, provides greater flexibility, and makes architectural and design changes easier as sites evolve.

The diagram below shows the relationship between the provider and the controls.



**Figure 1. Navigation architecture**

In the case of the navigation system, the data source describes the hierarchy of the web site pages that users can navigate to, and how this information should be displayed to the users. It is referred to as a site map. The layout for a simple web site might be:

Home

 Products

 Product A

 Product B

Product C

 Latest Offers

 Contact Us

 Email

 Visit us

**Custom Navigation and New User Interface Elements**

Windows SharePoint Services 3.0 provides exciting new features for navigation that enhance user awareness of site context. Two new breadcrumb controls located in the upper left and central regions of the page provide users awareness of the site structure above and within the current site. The top navigational control is now located beneath the banner and uses a tab-like structure for display. Both top navigation and the **Quick Launch** area (left navigation) are now highly customizable through the user interface or through the object model. All these new features improve the navigational experience of users, providing both power users and developers robust mechanisms for customizing navigation.

As in the previous version of Windows SharePoint Services, the contents of navigational controls can be changed by modifying the [NavBars](http://msdn.microsoft.com/en-us/library/ms462632.aspx) element in the [Onet.xml](http://msdn.microsoft.com/en-us/library/ms474369.aspx) file of a site definition, or by modifying the controls through Microsoft Office SharePoint Designer 2007. However, in the new version you can also modify the controls through the user interface, through markup on .aspx pages, or through types and members of the [Microsoft.SharePoint.Navigation](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.aspx) namespace. Administrators can toggle on or off a **Menu** or **TreeView** control for the left navigational area.

Windows SharePoint Services 3.0 is built on Microsoft ASP.NET 2.0. In addition to the changes in navigation, ASP.NET 2.0 supplies the new master pages, site map providers, and site map controls, simplifying and greatly improving navigation.

# **New Navigational Components in Windows SharePoint Services 3.0**

Windows SharePoint Services 3.0 provides the following new features to improve site navigation:

* User-aware links; for example, removing the **Settings** links for users who cannot make particular changes, a capability that is provided through "link trimming."
* Breadcrumbs to provide users with additional information about their location within a site collection.
* Customization of the top navigation bar, ranging from adding and removing links to adding Microsoft JScript drop-down menus and fly-outs, which is provided by new shared navigation and master pages. Such menus can only be enabled by modifying a master page; there is no built-in support for enabling these menus.
* Customization of the left navigation bar, which includes adding and removing links to adding JScript drop-down menus and fly-outs, which is provided through ASP.NET master pages and navigational controls such as the **SiteMapPath**, **Menu** and **TreeView** controls. Such menus can only be enabled by modifying a master page; there is no out-of-box support for enabling these menus.
* Common navigation bars provided through master pages.

**Customizing Quick Launch and the Top Link Bar Through the User Interface**

You can customize the **Quick Launch** area and top link bar in the browser by making selections on site settings pages. To customize **Quick Launch** or the top link bar, click **Site Actions**, click **Site Settings**, and within the **Look and Feel** section, click the appropriate link for the area you want to customize.

* To add links, change the order of links, or inherit links from the parent site, click **Top link bar**, and on the **Top Link Bar** page, click the appropriate action on the toolbar.
* To add or customize links and headings in **Quick Launch**, click **Quick Launch**, and through the **Quick Launch** page, make the appropriate customizations.
* To display either **Quick Launch** or a folder view in the **Quick Launch** area, or both, click **Tree view**. On the **Tree view** page, select **Enable Quick Launch** to display **Quick Launch**. Select **Enable Tree View** to display a folder view for navigation.

**Adding Links through the Object Model**

* Windows SharePoint Services uses standard ASP.NET controls for navigation. For example, the **System.Web.SiteMapNode** class is implemented for the global breadcrumbs appearing at the top left of SharePoint pages. The **Menu** control is used for both top navigation links and links in **Quick Launch**, and the navigation hierarchy of SharePoint sites is exposed through the **System.Web.SiteMapProvider** class. Custom navigation controls can be bound to the provider, including **Treeview**, **Menu**, and **SiteMapPath** controls.
* You can use types and members of the [Microsoft.SharePoint.Navigation](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.aspx) namespace to customize the navigation of sites. You can use the Windows SharePoint Services object model to manage both the **Quick Launch** area and the top navigation area, reordering links, adding new links, or removing links. You can also manage site inheritance in relation to parent sites.
* The [Navigation](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.spweb.navigation.aspx) property of the [Microsoft.SharePoint.SPWeb](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.spweb.aspx) class gets a [Microsoft.SharePoint.Navigation.SPNavigation](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigation.aspx) object that contains all the navigation properties for a specified site, including its navigation inheritance status relative to the parent site, as well as page hierarchy and navigation node collections. The [QuickLaunch](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigation.quicklaunch.aspx) property of the **SPNavigation** object returns the collection of navigation nodes found in the **Quick Launch** area, the [Web](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigation.web.aspx) property gets the top-level root site of the parent site collection, and the [TopNavigationBar](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigation.topnavigationbar.aspx) property returns the collection of navigation nodes that represent links in the top navigation area. An [SPNavigationNodeCollection](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigationnodecollection.aspx) object represents a collection of the ordered navigation nodes for a given object and provides the ability to add or remove nodes programmatically.
* **Note**   Although a [SPNavigationNode](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.navigation.spnavigationnode.aspx) object can contain arbitrary URLs, link fix-up cannot be implemented on the object for these URLs. However, the object can contain documents and pages in the site collection, and link fix-up is supported.

# **Example**

* The following example creates a link and adds it to the top of the **Quick Launch** bar:

|  |
| --- |
| Visual Basic |
| Dim siteCollection As SPSite = SPControl.GetContextSite(Context)Dim site As SPWeb = siteCollection.AllWebs("TestSite")Dim subSite As SPWeb = site.Webs("SubTestSite")Dim nodes As SPNavigationNodeCollection = subSite.Navigation.QuickLaunchDim navNode As New SPNavigationNode("New Link", "http://www.msn.com", True)nodes.AddAsFirst(navNode) |
| C# |
| SPSite siteCollection = SPControl.GetContextSite(Context);SPWeb site = siteCollection.AllWebs["TestSite"];SPWeb subSite = site.Webs["SubTestSite"];SPNavigationNodeCollection nodes = subSite.Navigation.QuickLaunch;SPNavigationNode navNode = new SPNavigationNode("New Link", "http://www.msn.com", true);nodes.AddAsFirst(navNode); |

* The third parameter of the [SPNavigationNode](http://msdn.microsoft.com/en-us/library/ms415159.aspx) constructor is set to **true** because the URL for the new link is external to the SharePoint deployment.
* To run this example, you must add a [Microsoft.SharePoint.WebControls.FormDigest](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.webcontrols.formdigest.aspx) control to the page making the post. For information on how to add a **FormDigest** control, see [Security Validation and Making Posts to Update Data](http://msdn.microsoft.com/en-us/library/ms472879.aspx). The example also requires referencing and importing the [Microsoft.SharePoint](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.aspx), **Microsoft.SharePoint.Navigation** and [Microsoft.SharePoint.WebControls](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.webcontrols.aspx) namespaces. For basic information about how to create a Web application that runs in the context of Windows SharePoint Services

# **Additional resources and references**

* [Customizing](http://msdn.microsoft.com/en-us/library/aa973249.aspx) site navigation
* [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)