

Optimize SharePoint 2010 with Breadcrumb Navigation

When developing a website, designing efficient ways to navigate through the site is of great importance. Site maps, horizontal menus, vertical menus, and breadcrumb navigation bars are examples of navigational elements in web page design. In this article, we will show you how to configure and stylize breadcrumb navigation bars in SharePoint 2010.

Using a breadcrumb navigation bar has many benefits. It saves space because it displays a hierarchical path of hyperlinked page names, as shown in Figure 1. It also provides an easy way for users to return to a higher-level page from the current location. In internet-facing sites, it improves your search engine optimization (SEO) strategy by creating links (with keywords in anchor text) within your site that help rank you higher in search engine results.

Figure 1: An example of a breadcrumb navigation bar



In SharePoint, you can use the [SiteMapPath](#) control to set up a breadcrumb navigation bar. In SharePoint 2010, you can also use the new [ListSiteMapPath](#) control. Both controls use a [SiteMapProvider](#) object as an information source. The controls represent the hyperlinks and apply a style to them. Let's take a closer look at the SiteMapProvider object and the SiteMapPath and ListSiteMapPath controls.

The SiteMapProvider Object

The SiteMapProvider object represents a tree-based structure depicting website navigation. For example, the navigation in XYZ Company's website might be represented as follows:

XYZ Company Web Root Page

- Publishing
- Services
 - SQL Server
 - Business Intelligence
 - SharePoint Branding
- News
- ...

A SiteMapProvider object uses a node collection to store navigation information, which includes information about the current node and the root node. It is very important that this object always stores information about the current node, which is the web page that is currently being visited by a user. The root node is the root page of the site.

In SharePoint 2010, there are several out-of-the-box SiteMapProvider objects, but you should pay particular attention to six of them:

- **SPXmlContentMapProvider:** Is used to display layout and administration pages (e.g., /_layouts/settings.aspx).
- **SPContentMapProvider:** Is used to display form pages (e.g., AllItems.aspx, EditForm.aspx, DisplayForm.aspx).
- **CurrentNavSiteMapProviderNoEncode:** Is used to display publishing pages (e.g., /Pages/Home.aspx, /subsite1/Pages/Page1.aspx).
- **GlobalNavigation:** Is used to display global navigation, which corresponds to the top horizontal menu.
- **CurrentNavigation:** Is used to display current navigation, which corresponds to the left vertical menu.
- **CombinedNavSiteMapProvider:** Is used to display a solution that combines current and global navigation.

You can find more information about SiteMapProvider in the [MSDN "SiteMapProvider Class" web page](#).

The SiteMapPath Control

The SiteMapPath control is an ASP.NET object that is commonly used to implement breadcrumb navigation bars in SharePoint. To configure it, you need to pick a SiteMapProvider and optionally customize the look and feel with Cascading Style Sheets (CSS).

Picking a SiteMapProvider

Depending on your users' needs, you can use one of the out-of-the-box SiteMapProvider objects or you can create your own. For example, if you are designing a publishing site that clearly separates system and site master pages and that has all the user-facing pages in one site master page, then you need to use CurrentNavSiteMapProviderNoEncode. Listing 1 shows an example how to configure a breadcrumb navigation bar for a publishing site using CurrentNavSiteMapProviderNoEncode. You would place this code in your master page.

Listing 1: Configuring breadcrumb navigation using the SiteMapPath control

```
<div class="SolidBreadCrumb">
  <asp:SiteMapPath runat="server"
    SiteMapProvider="CurrentNavSiteMapProviderNoEncode"
    RenderCurrentNodeAsLink="false"
    NodeStyle-CssClass="SolidBreadCrumb"
    CurrentNodeStyle-CssClass="SolidBreadCrumbCurrent"
    RootNodeStyle-CssClass="SolidBreadCrumbRoot"
    SkipLinkText="" />
</div>
```

If you are designing an intranet site and you need a breadcrumb navigation bar for the administration pages, list pages, library pages, view pages, and so forth, then you need to use SPContentMapProvider or SPXmlContentMapProvider, depending on your requirements. You will

probably also need to implement a custom breadcrumb solution by making your own SiteMapProvider object or your own class that inherits from SiteMapPath.

Now let's look at some examples. We'll use a default master page (nightandday.master) to display three SharePoint pages. In each page, we'll add three SiteMapPath controls. Each control will be configured with one out-of-the-box SiteMapProvider object (SPXmlContentMapProvider, SPContentMapProvider, or CurrentNavSiteMapProviderNoEncode). By looking at the resulting breadcrumb navigation bars, we can graphically observe how each SiteMapProvider performs in different pages. The three pages are as follows:

- Site settings page inside the URL http://publishingweb.xyz.com/subsite1/_layouts/settings.aspx. Figure 2 shows the resulting three breadcrumb navigation bars. Note that only the first breadcrumb navigation bar is showing the site settings page because only SPXmlContentMapProvider is storing it.
- Publishing page SubSite1-Page 1 inside the URL <http://publishingweb.xyz.com/subsite1/Pages/SubSite1-Page1.aspx>. Figure 3 shows the resulting breadcrumb navigation bars. Note that SPContentMapProvider shows the "Pages" hyperlink between "SubSite 1" and "SubSite1-Page1", but CurrentNavSiteMapProviderNoEncode is optimized to hide it. Therefore, CurrentNavSiteMapProviderNoEncode is the best option for publishing sites.
- Forms page showing SubFolder1 located in Custom List at SubSite 1. Figure 4 shows the resulting breadcrumb navigation bars. Note that only SPContentMapProvider is able to properly show all the list, folder, and item hyperlinks. SPXmlContentMapProvider shows nothing, and CurrentNavSiteMapProviderNoEncode shows the hyperlinks only through the subsite.

You can find more information about SiteMapPath in the [MSDN "SiteMapPath Class" web page](#).

Figure 2: Breadcrumb navigation in a site settings page using the SiteMapPath control

Site Settings

SiteMapPath with SiteMapProvider="SPXmlContentMapProvider" shows Layout and Administration Pages

Publishing Site > SubSite 1 > Site Settings

SiteMapPath with SiteMapProvider="SPContentMapProvider" shows Form Pages

Publishing Site > SubSite 1

SiteMapPath with SiteMapProvider="CurrentNavSiteMapProviderNoEncode" shows Publishing Pages

Publishing Site > SubSite 1

Figure 3: Breadcrumb navigation in a publishing page using the SiteMapPath control **SubSite 1-Page 1**

SiteMapPath with SiteMapProvider="SPXmlContentMapProvider" shows Layout and Administration Pages

SiteMapPath with SiteMapProvider="SPContentMapProvider" shows Form Pages

Publishing Site > SubSite 1 > Pages > **SubSite1-Page1**

SiteMapPath with SiteMapProvider="CurrentNavSiteMapProviderNoEncode" shows Publishing Pages

Publishing Site > SubSite 1 > **SubSite1-Page1**

Figure 4: Breadcrumb navigation in a forms page using the SiteMapPath control

Custom List > **SubFolder1** > All Items ▾

SiteMapPath with SiteMapProvider="SPXmlContentMapProvider" shows Layout and Administration Pages

SiteMapPath with SiteMapProvider="SPContentMapProvider" shows Form Pages

Publishing Site > SubSite 1 > **Custom List** > **SubFolder1**

SiteMapPath with SiteMapProvider="CurrentNavSiteMapProviderNoEncode" shows Publishing Pages

Publishing Site > **SubSite 1**

Customizing the Look and Feel with CSS

If you used out-of-the-box SiteMapProvider objects with the SiteMapPath control to create a breadcrumb navigation bar, you might want to customize the look and feel of it. To do so, you need configure all CSS properties and create CSS code. Listing 2 shows some sample CSS code that you would place in the .css file.

Listing 2: Sample CSS code

```
.SolidBreadCrumb{
    font-family: Arial, Segoe UI, Verdana, sans-serif;
    color: #003F50;
    clear:both;
    font-size:9pt;
    padding-left:7px;
    padding-bottom:2px;
    padding-top:4px;
    border-bottom:1px #003F50 solid;
    margin-bottom:4px;
}

.SolidBreadCrumb a, .SolidBreadCrumb:active, .SolidBreadCrumb a:active{
    padding-left:0px;
    padding-right: 0px;
    text-decoration:none;
    color: #003F50;
}

.SolidBreadCrumb a:hover{
    text-decoration:underline;
}

.SolidBreadCrumbCurrent{
    font-weight:bold;
}
```

The ListSiteMapPath Control

In SharePoint 2010, you can use the new ListSiteMapPath control to combine all navigation providers or SiteMapProvider objects. Instead of putting the breadcrumbs in a horizontal path, this control, by default, puts them in a vertical hierarchy.

Listing 3 shows an example of how to use the ListSiteMapPath control. As you can see, the control uses the SiteMapProviders property to specify two SiteMapProvider objects (SPXmlContentMapProvider and SPContentMapProvider). You can specify as many SiteMapProvider objects as needed, which means you can display breadcrumbs for all kinds of SharePoint pages using only one ListSiteMapPath control.

Listing 3: Configuring breadcrumb navigation using the ListSiteMapPath control

```

<SharePoint:ListSiteMapPath
  runat="server"
  SiteMapProviders="SPXmlContentMapProvider,SPContentMapProvider"
  RenderCurrentNodeAsLink="false"
  PathSeparator=""
  CssClass="s4-breadcrumb"
  NodeStyle-CssClass="s4-breadcrumbNode"
  CurrentNodeStyle-CssClass="s4-breadcrumbCurrentNode"
  RootNodeStyle-CssClass="s4-breadcrumbRootNode"
  NodeImageOffsetX=0
  NodeImageOffsetY=353
  NodeImageWidth=16
  NodeImageHeight=16
  NodeImageUrl="/_layouts/images/fgimg.png"
  RTLNodeImageOffsetX=0
  RTLNodeImageOffsetY=376
  RTLNodeImageWidth=16
  RTLNodeImageHeight=16
  RTLNodeImageUrl="/_layouts/images/fgimg.png"
  HideInteriorRootNodes="true"
  SkipLinkText=""/>
  
```

However, if you need only one master page and only one breadcrumb navigation bar in your site, you have two options:

1. Create one custom-coded control that combines all your providers (inheriting from SiteMapPath).
2. Use ListSiteMapPath with SiteMapProviders="SPXmlContentMapProvider,SPContentMapProvider" and later override the CSS styles in order to get your own design style.

Note that if you use the second method, you can remove the "Pages" link in the breadcrumb navigation bar using jQuery or JavaScript. However, if you don't mind the "Pages" link, you can leave it in and save some development time.

Like we did for the SiteMapPath control, let's observe how the breadcrumb navigation is displayed when the ListSiteMapPath control is used in three kinds of SharePoint pages: a site settings page (Figure 5), a publishing page in SubSite1 (Figure 6), and a forms page in SubFolder1 located in Custom List (Figure 7). When comparing the results, note that:

- The breadcrumb navigation worked well in all three SharePoint pages without changing the SiteMapProvider objects.
- By default, the breadcrumbs are shown vertically. You can, however, modify this behavior by using CSS, JavaScript code, or custom code.
- If you want to create a custom breadcrumb, you can inherit from SiteMapPath and customize the hyperlink generation.

For more information about ListSiteMapPath, see the [MDSN "ListSiteMapPath Class" web page](#).

Figure 5: Breadcrumb navigation in a site settings page using the ListSiteMapPath control

ListSiteMapPath with SiteMapProviders="SPXmlContentMapProvider,SPContentMapProvider".

↳ Publishing Site
 ↳ SubSite 1
 ↳ Site Settings
 ↳ SubSite 1

Figure 6: Breadcrumb navigation in a publishing page using the ListSiteMapPath control

ListSiteMapPath with SiteMapProviders="SPXmlContentMapProvider,SPContentMapProvider".

↳ Publishing Site
 ↳ SubSite 1
 ↳ Pages
 ↳ SubSite1-Page1

Figure 7: Breadcrumb navigation in a forms page using the ListSiteMapPath control

ListSiteMapPath with SiteMapProviders="SPXmlContentMapProvider,SPContentMapProvider".

↳ Publishing Site
 ↳ SubSite 1
 ↳ Custom List
 ↳ SubFolder1

Conclusion

If you are branding a publishing site or an internet-facing site, you probably can use the SiteMapPath control with some CSS customizations. However, if you need to manually make specific changes to breadcrumbs to meet users' needs, you probably need implement your own code to hide, add, or change some hyperlinks in the breadcrumb navigation bar.

If you are branding an intranet site, the ListSiteMapPath control will likely be the best choice. With it, you can use the same breadcrumb navigation code on all kinds of SharePoint pages (e.g., settings, publishing, forms).

Remember that there isn't one solution for all scenarios, so you need to find out what your users need and apply the best option based on those needs. Also remember that you need to take into account SEO for internet-facing sites. In these sites, you need to generate the best internal link structure to positively affect the search engine rankings.

About the Author



Jose Quinto Zamora ([Twitter](#)) is a SolidQ engineer. He earned his master's degree in Computer Science at University of Alicante. He is working toward his doctorate in the Contextual and Multimodal Search field at the University of Alicante, combining his doctoral thesis with his work at SolidQ. He has a broad experience with Microsoft SharePoint Server 2010 and Office SharePoint Server 2007, specializing in accessibility, location, development, and management. He delivered several sessions at the SQLU Summit in Madrid and several webcasts for Microsoft Iberica Spain. He is an MCP, MCTS, MCPD, and MCITP in SharePoint 2010.



Rely on the most trusted experts in the Microsoft
Data and Development space.

Consulting

www.solidq.com