
ASP.NET 4.5
Lesson 4: Site Navigation
Capgemini

Lesson Objectives

- The objectives of this lesson are to:
  - Understand Site Navigation in ASP.NET
  - Understand various Navigation Controls in ASP.NET

4.1: Why Site Navigation?

Overview

- Web Sites with more than one page and hierarchical in nature
- Users may be lost
- Previously, Navigations were made possible with hyperlinks
  - User controls utilized
  - Difficult to manage
- ASP.NET 2.0 provides Site Navigation System

All of the web applications developed usually have more than one page and they are usually interconnected via some mechanism. One of the major issues faced by Web site users is figuring out how to get around the site effectively. Web sites are often hierarchical in nature, and pages are sometimes nested several layers deep. Users may often find themselves asking questions like “Where am I now?” and “Where can I go from here?”.

Previously in ASP.Net, navigation was made possible by using hyperlinks with the help of include file or user controls. They are pretty handy but not so much when the pages are moved around or their names change. In order to overcome this drawback, ASP.Net 2.0 introduces the new Site Navigation System. Site navigation uses the layered architecture.
Controls such as Menu and TreeView provide the navigation UIs.

4.2: ASP.NET Navigation Support

Overview

- ASP.NET Navigation Support
  - Navigation Controls
    - Map Human-Readable Display Names to URLs
  - Site Map Data Source
    - Stores the information about a site's hierarchical organization
  - Site Map Provider
    - Interprets Physical data (often in the form of an XML file) and implements a kind of database cursor representing the current position within a site's hierarchy

ASP.NET's navigation support comes in three parts: the navigation controls, the site map data source, and the site map provider architecture.

The Navigation controls have the capability to resolve human-readable display names to real URLs to which HTTP requests may be sent.

The SiteMap Datasource stores information about a site's hierarchical organization.

The SiteMap Provider interprets Physical data (often in the form of an XML file) and implements a kind of database cursor representing the current position within a site's hierarchy.

4.3: Site Maps

Overview

- XML files that describe the logical structure of your web application
- Defines layout of all the pages and their relation to each other
- Ta add a Site Map add a new item of type SiteMap to the web application
  - Default name is Web.SiteMap

Site maps are XML files which describes the logical structure of your web application. We can use it to define the layout of all pages in application and how they relate to one another. One can add or remove links or pages to your site map there by managing navigation of website effectively.
Site maps are defined with .sitemap extension.

Use new SiteMapDataSource control to help bind the information in the site map file to new navigation server controls provided by ASP.NET 2.0.

The example below shows a sample Web.SiteMap file:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode title="Home" description="Home" url="~/home.aspx" >
    <siteMapNode title="Trainings" description="Our
    Trainings“ url="~/Trainings.aspx">
      <siteMapNode title="Technical Trainings“ description="Technical
    Trainings we offer“ url="~/TTrainings.aspx" />
      <siteMapNode title="Behavioral Trainings“ description="Behavioral
    Trainings we offer“ url="~/BTrainings.aspx" />
      <siteMapNode title="Tool Trainings“ description="Tool Trainings we
    offer“ url="~/ToTrainings.aspx" />
    </siteMapNode>
    <siteMapNode title="Services" description="Services we
    offer“ url="~/Services.aspx">
      <siteMapNode title="Express Learning Capsule"
    description="ELC“ url="~/elc.aspx" />
      <siteMapNode title="Senior Capsule" description="Senior
    Capsule“ url="~/sc.aspx" />
      <siteMapNode title="Facilitation"
    description="Facilitation“ url="~/fact.aspx" />
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

4.4: Navigation Controls

Overview

- Menu
  - Interprets information contained in the sitemap XML file and presents it in a menu format
- TreeView
  - Interprets information contained in the sitemap XML file and presents it in a tree format
- SiteMapPath
  - Interprets information contained in the sitemap XML file and presents it in a breadcrumb format

ASP.NET includes three server-side controls devoted specifically to site navigation—the SiteMapPath, the Menu, and the TreeView control.
The Menu and the TreeView ,both maintain collections of display name/URL mappings.
These collections may be edited by hand. In addition, these controls can build hierarchical collections of display name or URL mappings based on information in a site map data source.
The SiteMapPath builds its collection of display name or URL mappings solely through a site map data source.

ASP.NET Navigation controls are summarized as follows:
Menu

It interprets the site navigational information contained in the sitemap XML file and presents it in a menu format.
Top level XML nodes become top level menu items, with child XML nodes becoming child menu items.

TreeView

It interprets the site navigational information contained in the sitemap XML file and presents it in a tree format.
The top level sitemap XML nodes in this case become higher-level branches in the tree, with child nodes represented as child tree nodes.

SiteMapPath

Interprets the site navigational information contained in the sitemap XML file and presents it in a “breadcrumb” format.
In this case, only the current XML node’s path is displayed (from the root node to the current child node).

4.4.1: Site Map Datasource Properties

Site Map Datasource Properties

- ShowStartingNode
  - Use this property to show or hide the first (top-level) node
  - The default is true
- StartingNodeUrl
  - Use this property to change the starting node
  - Set value to the URL of the node to be the first node in the navigation tree
- StartFromCurrentNode
  - Use this property to set the current page as the starting node
- StartingNodeOffset
  - Use this property to shift the starting node up or down the hierarchy
  - Set a positive number to move down or a negative number to move up

ShowStartingNode
Set this property to false to hide the first (top-level) node that would otherwise appear in the navigation tree.

StartingNodeUrl
Set this value to the URL of the node that should be the first node in the navigation tree. This value must match the url attribute in the site map file exactly. For example, if you specify a StartingNodeUrl of "~/home.aspx", then the first node in the tree is the Home node, and you will see nodes only underneath that node.

StartFromCurrentNode
Set this property to true to set the current page as the starting node. The navigation tree will show only pages beneath the current page (which allows the user to move down the hierarchy). If the current page doesn't exist in the site map file, this approach won't work.

StartingNodeOffset
It takes an integer that instructs the SiteMapDataSource to move from the starting node down the tree (if the number is positive) or up the tree (if the number is negative).
The actual effect depends on how you combine this property with other SiteMapDataSource properties.
For example, if StartFromCurrentNode is false,you'll use a positive number to move down the tree, from the starting node toward the current node.
If StartFromCurrentNode is true, you'll use a negative number to move up the tree, away from the current node and toward the starting node

4.4.2: Menu Control

How to Use a Menu Control?

- Drag and Drop a SiteMapDataSource control on the form
  - It automatically binds it self to web.sitemap file
- Drop a Menu control on the form and set the DataSource id to SiteMapDataSource1

The Menu control can be bind to hierarchical data source that implements IHierarchicalDataSource or IHierarchicalEnumerable. Although It is tailor-made to support site maps, they work with other data sources.
Figure above shows the Menu control in action.

Menu Properties:-
StaticMenuStyle / DynamicMenuStyle :- Sets the appearance of the overall “box” in which all the menu items appear.
In the case of StaticMenuStyle, this box appears on the page, and
with DynamicMenuStyle, it appears as a pop-up.

StaticMenuItemStyle / DynamicMenuItemStyle:- Sets the appearance of individual menu items.

StaticSelectedStyle / DynamicSelectedStyle:- Sets the appearance of the selected item. Note that the selected item isn’t the item that’s currently being hovered over; it’s the item that was previously clicked(and that triggered the last postback).

StaticHoverStyle / DynamicHoverStyle :-Sets the appearance of the item that the user is hovering over with the mouse.

4.4.2: Menu Control

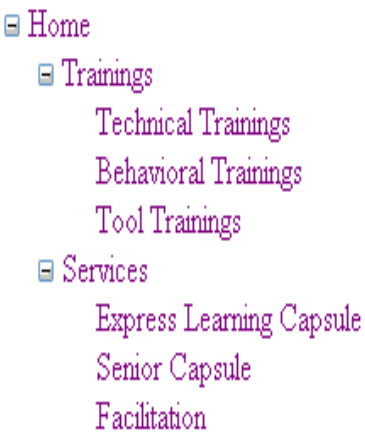
Demo

- Creating and using Menu Control for Navigation between the web pages

4.4.3: TreeView Control

How to Use a TreeView Control?

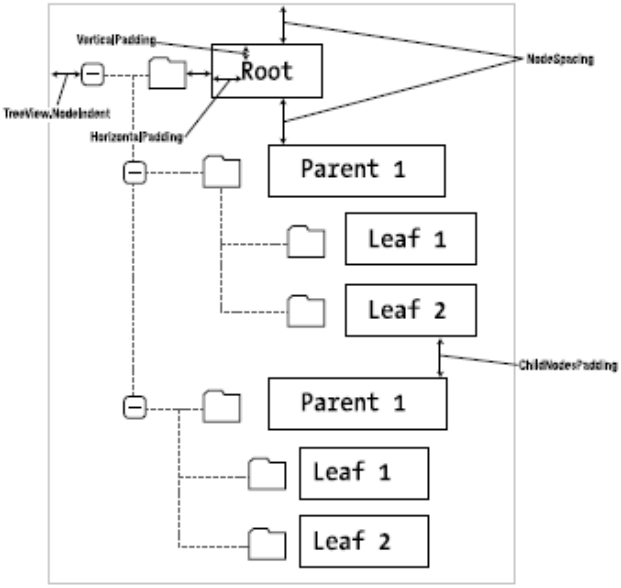
- Drag and drop a SiteMapDataSource control on the form.
- It automatically binds it self to web.sitemap file
- Drop a treeView control on the form and set the DataSource id to SiteMapDataSource1



TreeView controls can bind to hierarchical data source implementing IHierarchicalDataSource or IHierarchicalEnumerable. Although they are tailor-made to support site maps, they work with other data sources. The figure above shows TreeView Control in action.

4.4.3: TreeView Control

Node Spacing



Because a TreeView is rendered using an HTML table, you can set the padding of various elements to control the spacing around text, between nodes, and so on. One other property that comes into play is TreeView.NodeIndent, which sets the number of pixels of indentation (from the left) in each subsequent level of the tree Hierarchy

Clearly, styles give you a lot of control over how different nodes are displayed. To apply a simple TreeView makeover, and to use the same style settings for each node in the TreeView, you apply style settings through the TreeView.NodeStyle property. You can do this directly in the control tag or by using the Properties window.
For example, here's a TreeView that applies a custom font, font size, text color, padding, and spacing:

```
<asp:TreeView ID="TreeView1" runat="server"
DataSourceID="SiteMapDataSource1">
<NodeStyle Font-Names="Tahoma" Font-Size="10pt" ForeColor="Blue"
HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />
</asp:TreeView>
```

4.4.3: TreeView Control

Demo

- Creating and using TreeView Control for Navigation between the web pages

4.4.4: SiteMapPath Control

Using a SiteMapPath Control

- Shows a trail that indicates where the user is in the web page hierarchy
- Shows a path back to the top node – **Bread Crumbs**
- Useful within sites that maintain a very deep hierarchy

Home > Trainings > Behavioral Trainings

Home > Services > Facilitation

The SiteMapPath control shows a trail indicating where the user is within the Web page hierarchy and shows a path back to the top node (kind of like a trail of bread crumbs). The SiteMapPath is mostly useful within sites that maintain a very deep hierarchy for which a Menu or a TreeView control would be overwhelmed.

Although the SiteMapPath control is like the Menu and the TreeView (the SiteMapPath control reflects the state of the SiteMap object), it does deserve special attention. The SiteMapPath control and the site map data within the provider are tightly coupled.
For example, if you leave a page out of your site map and the user somehow ends up on the page, the user will not see the SiteMapPath control on the page.

4.4.4: SiteMapPath Control

Site Map Node Navigation Properties

- ParentNode
  - Returns the node one level up in the navigation hierarchy, which contains the current node
  - On the root node, this returns a null reference
- ChildNodes
  - Provides a collection of all the child nodes
  - Use the *HasChildNodes* property to check whether child nodes exist
- PreviousSibling
  - Returns the previous node that's at the same level
  - A null reference if no such node exists
- NextSibling
  - Returns the next node that's at the same level
  - A null reference if no such node exists

SiteMap path Properties :-

ShowToolTip:- Set this to false if you don't want the description text to appear when the user hovers over a part of the site map path.
ParentLevelsDisplayed :-This sets the maximum number of levels above the current page that will be
shown at once. By default, this setting is –1, which means all levels will be shown.
RenderCurrentNodeAsLink :-If true, the portion of the page that indicates the current page is turned into a clickable link. By default, this is false because the user is already at the current page.
PathDirection :-You have two choices: RootToCurrent (the default) and CurrentToRoot (which reverses the order of levels in the path).
PathSeparator :-This indicates the characters that will be placed between each level in the path.
The default is the greater-than symbol (>). Another common path separator is
the colon (:).

4.4.4: SiteMapPath Control

Demo

- Creating and using SiteMapPath
- Creating and Using Navigation Controls in a Master Page

Summary

- In this module, you have learnt:
  - Why Site Navigation is important
  - Different Site Navigation Controls in ASP.NET.0

Review Question

- Question 1: Which of the following Navigation Control presents data in "breadcrumb" format?
  - TreeView
  - Menu
  - SiteMapPath
  - Map
- Question 2: What is the default name for a Site Map?
  - Machine.SiteMap
  - Web.SiteMap
  - Web.xml
  - SiteMap.xml