

相关主题

RECOMMEND ARTICLE

- 在ASP.NET 2.0中建立站点导航层次
- JSP和JSF双剑合并 打造完美Web应用
- 用ASP.NET2.0在数据库中存储二进制文件
- ASP.NET中上传文件到数据库
- ASP.NET定制简单的错误处理页面
- 基于JSF技术的WEB应用开发研究
- ASP.NET实现投票结果的图片进度条显示
- 圣殿骑士PHP 2007年Web开发技术预言

MORE

推荐文章

RECOMMEND ARTICLE

- 数据广播方案的优化
- 网络游戏的位置同步
- 游戏音乐制作案例之《战火 红色警戒》音效制作揭秘
- 英雄连Online 原画
- 游戏音乐制作案例之《乱武天下》
- 游戏音乐制作案例之《诛仙》
- 《鹿鼎记》最新原画
- MIDP2.1规范的新特性

MORE

热门文章

HOT ARTICLE

- [电子书下载]游戏设计 — 原理与实践
- [电子书下载]网络游戏开发
- 游戏设计全过程
- [电子书下载]游戏设计技术
- [电子书下载]游戏设计理论
- CS游戏人物模型制作教程
- CG人物插画基本流程
- [转贴]MAX高级人头教程

MORE

您的位置: WEB技术



文章标题	在ASP.NET 2.0中建立站点导航层次		
来源:	[陶刚编译]	浏览:	[450]

站点导航提供程序--ASP.NET 2.0中的站点导航提供程序暴露了应用程序中的页面的导航信息,它允许你单独地定义站点的结构,而不用考虑页面的实际物理布局。默认的站点导航提供程序是基于XML的,但是你也可以通过编写自定义的提供程序,从任何后端位置暴露这些信息。

站点导航API--站点导航API用于在应用程序的代码中访问站点导航信息,它摘录了导航信息存储的细节。你可以使用API来编程访问应用程序的导航节点。

导航控件--导航控件为页面之间的导航提供了通用的UI,例如树视图、菜单和breadcrumb("面包屑",一种显示当前所在页面的控件)。这些控件利用ASP.NET 2.0中的站点导航服务来检索你给站点定义的结构。SiteMapDataSource控件还允许你把其它UI控件绑定到站点导航数据。

网站经常需要显示导航数据,来指导用户如何使用站点。ASP.NET中的导航特性允许开发者简单地定义导航数据,并根据这些信息来显示UI。

站点导航API是一种用于访问站点导航数据的基于提供程序(provider)的编程内容。该API把导航数据存储在XML文件中,并通过一组SiteMapNode类来暴露这些数据。应用程序和控件开发者可以构建SiteMapNode实例并使用这些信息来显示导航界面。

面向导航的服务器控件包括Menu、TreeView、SiteMapPath和SiteMapDataSource控件。这些控件都是建立在站点导航类的顶端的,它们使用和显示导航数据的时候都是不考虑数据存储的特定细节问题的。Menu和TreeView控件还可以使用XML文件的数据和XMLDataSource控件的数据。

Url映射特性允许开发者为不同URL请求的重映射(re-mapping)定义简单的规则。

使用站点导航控件

Menu、TreeView、SiteMapPath和SiteMapDataSource控件根据导航数据生成导航界面。导航数据可以存储在XML文件中,或者利用站点导航特性的基于提供程序的能力来保存。下面的例子演示了如何组合使用站点导航特性的不同控件。

建立应用程序站点地图

示例的导航结构存放在Web.sitemap文件中,在下面你可以看到站点地图文件。Web.sitemap文件包含一个顶层的<siteMap>元素。在<siteMap>元素内至少嵌套一个<siteMapNode>元素。在一个站点地图内必须有一个顶层的<siteMapNode>。站点导航特性需要一个根<siteMapNode>来确保沿着节点层次的访问最终汇聚到一个已知的节点。你可以在根<siteMapNode>元素下嵌套多个<siteMapNode>元素。此外,嵌套<siteMapNode>元素的深度是没有限制的。

一个<siteMapNode>元素通常包含Url(链接)、Title(标题)和Description(描述)属性。Url属性指明与应用程序中的页面对应的路径。它也可以包含其它应用程序中的页面的路径,或者指向完全不同的网站的多个URL。在下面的例子中,所有的Url属性都使用应用程序相对语法来引用路径。Title属性用于显示导航数据UI的文本内容。例如,SiteMapPath控件把Title属性作为控件的超链接文本显示。如果提供了Description属性,服务器控件就把显示为工具条提示或ALT文本。开发者也可以给<siteMapNode>添加自定义属性,利用SiteMapNode类的默认索引器(indexer)就可以检索这些属性了。你可以查阅.NET框架组件文档找到更多的关于<siteMapNode>元素其它一些属性的信息。

Web.sitemap的内容

```
<siteMap>
<siteMapNode title="Home" url="-/default.aspx" >
<siteMapNode title="Introduction to ASP.NET" url="-/introduction/default.aspx">
<siteMapNode title="What's New in Whidbey?" url="-/introduction/whatsnew.aspx"/>
<siteMapNode title="Sample Applications (Starter Kits)" url="-/introduction/starterkits.aspx"/>
<siteMapNode title="Introduction to Visual Web Developer" url="-/introduction/vwd.aspx"/>
</siteMapNode>
<siteMapNode title="Building A Web Application" url="-/development/default.aspx">
<siteMapNode title="Building a Simple Application" url="-/development/simple/default.aspx">
<siteMapNode title="Introduction to ASP.NET pages" url="-/development/simpl e/pages.aspx"/>
.....
</siteMapNode>
</siteMapNode>
</siteMap>
```

使用站点导航控件

在Web页面上提供站点导航的最简单办法就是使用图形化的站点导航控件(SiteMapPath、TreeView和Menu)。

- SiteMapPath--一个breadcrumb控件,它检索用户的当前页面并显示页面的层次结构。这让用户可以导航回层次中其它页面。SiteMapPath只能与SiteMapProvider一起使用,需要设置控件的SiteMapProvider属性。

- TreeView--在Web页面上提供垂直的用户界面,它可以展开和收缩选中的节点,也可以提供检查框功能以供用户选择数据项。TreeView控件支持宣告式的或编程设置数据元素和数据源控件。如果你使用SiteMapDataSource控件,那么数据绑

定就是自动进行的。

· Menu--提供水平的或垂直的用户界面，当用户把鼠标放在一项上的时候会弹出子菜单。Menu控件支持宣告式的或编程设置数据元素或数据源控件。如果你使用Si teMapDataSource控件，数据绑定就是自动进行的。

请注意，TreeView和Menu控件都可以用于非导航的情形。

下表描述了TreeView和Menu之间的差别，以便于你能根据需要做出适当的选择。

特性	Menu	TreeView
扩展	弹出	位置扩展
根据需要下载	No	Yes
检查框	No	Yes
模板	Yes	No
布局	水平 & 垂直	垂直
Style选项	Yes	Yes
选择模式	静态的、	动态的层次
	层次或父/根/叶，	或按数据项

在下面的例子中，你看到的TreeView和Menu控件为不同的区域和操作（例如NodeStyle和HoverNodeStyle）配置了一些样式属性。例子中的TreeView和Menu控件都连接到页面上的同一个Si teMapDataSource控件。Si teMapPath控件通过Web.config文件中设置的默认的Si teMapProvider来访问相同的数据。

```
<asp:treeview ID="TreeView1" ForeColor="White" DataSourceId="Si teMapDataSource1" NodeIndent="0" NodeStyle-ChildNodesPadding="10" runat="server">
<LevelStyles>
<asp:TreeNodeStyle Font-Bold="true"/>
<asp:TreeNodeStyle />
<asp:TreeNodeStyle Font-Size="x-small"/>
</LevelStyles>
<nodestyle forecolor="White" HorizontalPadding="5"/>
<SelectedNodeStyle backcolor="LightBlue" forecolor="Blue" />
<HoverNodeStyle Font-Underline="true" />
</asp:treeview>
<asp:sitemapath id="Si teMapPath1" runat="server" />
<asp:Menu ID="Menu1" DataSourceId="Si teMapDataSource1" runat="server">
<DynamicSelectedStyle BackColor="LightBlue" ForeColor="Blue" />
<DynamicHoverStyle Font-Underline="true" />
<StaticHoverStyle Font-Underline="true" />
</asp:Menu>
<asp:sitemapdatasource id="Si teMapDataSource1" runat="server" />
```

使用站点导航API

站点导航API是使用可配置的提供程序访问导航数据的编程抽象内容。站点导航提供程序把导航数据的存储细节信息与API的其它部分隔离开来。站点导航API通过Si teMap和Si teMapNode类来暴露导航数据。Si teMap类返回与当前页面对应的Si teMapNode实例。它还可以访问那些为站点导航特性配置的提供程序。Si teMapProvider为执行下面一些事务提供了丰富的API：

- 依据当前的HttpContext或任意URL检索Si teMapNode实例。
- 检索Si teMapNode的父或子节点。
- 访问当前页面的Si teMapNode，以及整个导航层次中的根Si teMapNode。
- 执行授权规则，这样就保证了提供程序只返回用户可以看见的节点。

Si teMapNode实例暴露的基本导航信息和功能包括：

- URL、Title和description属性，以及开发者给Si teMapNode添加的自定义属性。
- 获取某个节点的父和子节点。
- 在某个节点的前后节点之间进行导航。
- 获取Si teMapProvider实例的指针，它返回一个节点。

ASP.NET发布的时候带有Xml Si teMapProvider提供程序。该提供程序使用XML文件（web.sitemap）中的数据，并根据数据返回Si teMapNode实例。Xml Si teMapProvider有下面一些功能：

- 多个站点地图（sitemap）文件可以链接在一起构建一个“虚拟的”导航数据集合。
- 多个Xml Si teMapProvider实例可以链接到一起构建一个“虚拟的”导航数据集合。
- 提供程序可以根据站点当前的文件授权和URL授权规则来返回过滤后的节点。

有了Si teMapProvider指针之后，你就可以根据URL来检索特定节点的站点导航数据。它会让你获取站点导航数据中的Si teMapNode实例指针。可以检索任意Si teMapNode实例的能力和从任何Si teMapNode开始进行站点导航的能力组合在一起，使你能够轻易地遍历站点的导航数据。

作为一名开发者，你也可以把导航数据用其它的格式进行存储（例如作为关系数据存储在数据库中）。接着你应该构建一个衍生自Si teMapProvider的自定义提供程序。

使用站点导航类编程

你可以在代码中编程获取导航数据。编程获取站点导航数据的出发点是Si teMap类。在这个类中有大量的静态方法，其中最重要的一个是CurrentNode属性。在网站上的任何页面中，你都可以调用Si teMap.CurrentNode来引用与当前的执行页面相匹配的导航数据片断。导航数据是用Si teMapNode实例的形式返回的--当你调用Si teMap.CurrentNode的时候，该属性返回与当前页面对应的Si teMapNode实例。站点导航特性根据存储在XML文件中的导航数据返回正确的节点。

下面的例子演示了一个带有简单的分页功能的用户控件。在显示的页面中，用户控件位于页面的底部中间。最初该链接的内容是“下一个主题”。当你点击这个链接的时候，用户控件调用Si teMapNode对象来检测当前页面的附近是否存在页面。代码检测Si teMap.CurrentNode属性，看它的前面是否有页面（Si teMap.CurrentNode.PreviousSibling）、它的后面是否有页面（Si teMap.CurrentNode.NextSibling）。如果存在页面，用户控件就显示超链接，并把超链接的NavigateUrl属性设置

为附近节点的Url属性。

如果你点击页面左部的Treeview链接,可以看到用户控件是如何自动地显示适当的“前一个主题”和“后一个主题”链接的。用户控件还显示了另外一个超链接,你可以点击它返回主页。如果你查看这个超链接如何工作就会发现,该控件利用了主页<siteMapNode>元素中的自定义属性“customAttribute”。该控件演示了如何使用SiteMapNode的默认索引器来检索自定义属性的值。

```
<script language="VB" runat="server">
Sub Page_Load()
If (Not SiteMap.CurrentNode.NextSibling Is Nothing) Then
NextTopic.NavigateUrl = SiteMap.CurrentNode.NextSibling.Url
Else
NextTopic.Visible = false
Separator.Visible = false
End If

If (Not SiteMap.CurrentNode.PreviousSibling Is Nothing)
PrevTopic.NavigateUrl = SiteMap.CurrentNode.PreviousSibling.Url
Else
PrevTopic.Visible = false
Separator.Visible = false
End If

'使用FindSiteMapNode查找URL中的节点并提取一些信息
Dim rootNode as SiteMapNode = SiteMap.Provider.FindSiteMapNode("~/Home.aspx")
GoHome.NavigateUrl = rootNode.Url
GoHome.ToolTip = rootNode.Description
GoHome.Text = rootNode("customAttribute")
End Sub
</script>
```

站点导航的安全性

站点导航特性可以根据授权规则过滤提供程序所返回的SiteMapNode实例。XmlSiteMapProvider可以根据当前网站使用的文件和URL授权规则过滤节点。

下面的例子使用了窗体授权规则,预定义的用户凭证存储在web.config中。在global.asax中,根据用户名,用户的角色都被附加到当前的请求上。在web.config中,嵌套在<siteMap>元素之下的站点地图提供程序使用的<add>元素的securityTrimmingEnabled属性被设置为真。同时,web.config文件的末尾定义了一组URL授权规则。当你运行示例并登陆之后,XmlSiteMapProvider会自动地依据用户所属的角色和web.config中定义的授权规则来对SiteMapNode执行授权检测。

请使用下面三个帐户之一运行示例:

- Userid: SectionOne Password: SectionOne
- Userid: SectionTwo Password: SectionTwo
- Userid: AllSections Password: AllSections

在页面的右上角有一个“退出”链接,因此你可以用不同的帐号登陆和退出。请注意,根据你登陆所使用的帐号不同,导航UI显示的Treeview和Menu控件会自动地反映该用户所获得的访问权限。提供程序自动地过滤了返回的节点--实现这种功能不需要额外的代码。用“SectionOne”帐号登陆的时候,只在左边的Treeview控件中显示“SectionOne”链接和外部链接。用“SectionTwo”帐号登陆的时候,只在左边的Treeview控件中显示“SectionTwo”链接和外部链接。用“AllSections”帐号登陆的时候,Treeview控件中显示了所有的链接。web.config中的授权规则配置为给“SectionOne”和“SectionTwo”层次授予了部分访问权力。

这个示例还演示了如何处理应用程序目录范围之外的URL安全性。在web.sitemap文件中,外部链接的节点使用了roles属性。语法roles="*"授予所有用户访问和查看导航控件中的节点的权力。语法roles="Administrators,RegularUsers"只允许这些角色的用户检索和查看导航控件中的节点。由于在示例中global.asax文件把用户分成了这两种角色,所以你可以一直可以看到外部链接。

开发者可以选择同时使用文件/URL授权规则和roles属性来控制用户对SiteMapNode实例的访问权。如果两者的设置信息都是正确的,站点导航提供程序就会根据文件/URL授权规则和roles属性中的角色来对当前用户进行认证。如果当前用户通过了任何一种授权检查,就可以访问节点。

如果默认的安全性操作不适用于你的应用程序,开发者还可以从XmlSiteMapProvider衍生类,并用自定义的节点授权实现来重载IsAccessibleToUser方法。

Web.config的内容

```
<system.web>
<authentication mode="Forms">
<forms name="." ASPXAUTH="Login.aspx" loginUrl="Login.aspx" protection="All" timeout="30" path="/" requireSSL="false"
slidingExpiration="true" defaultUrl="Home.aspx" cookieless="UseCookies" enableCrossAppRedirects="false">
<credentials passwordFormat="Clear">
<user name="SectionOne" password="SectionOne"/>
<user name="SectionTwo" password="SectionTwo"/>
<user name="AllSections" password="AllSections"/>
</credentials>
</forms>
</authentication>
<authorization>
<deny users="?"/>
</authorization>
</system.web>
<location path="SectionOne.aspx">
<system.web>
<authorization>
<allow users="SectionOne" roles="Administrators" />
<deny users="*/>
</authorization>
</system.web>
</location>
<location path="SectionOne">
<system.web>
```

```

<authorization>
<allow users="SectionOne" roles="Administrators"/>
<deny users="*/>
</authorization>
</system.web>
</location>
<location path="SectionTwo.aspx">
<system.web>
<authorization>
<allow users="SectionTwo" roles="Administrators"/>
<deny users="*/>
</authorization>
</system.web>
</location>
<location path="SectionTwo">
<system.web>
<authorization>
<allow users="SectionTwo" roles="Administrators"/>
<deny users="*/>
</authorization>
</system.web>

```

本地化站点地图数据

存储在si temap文件中的导航数据可能需要进行本地化（localize）。<si temapNode>元素中的URL、Title和Description属性也可以本地化。此外，开发者放置在<si temapNode>元素中的任何自定义属性也可以本地化。

下面的示例包含了英语和法语的本地化文本。它的web.sitemap文件使用两种类型（隐式的和显式的）的本地化表达式来实现这种功能。Sitemap文件在根sitemap元素中使用了enableLocalization=true就表明它使用了本地化数据。

站点地图文件的隐式表达式让开发者能够轻易地用查找键（lookup key）标记每个<si temapNode>元素，而查找键是用于从资源文件检索资源的。在示例的web.sitemap中，除了第一个节点之外，所有的节点都有隐式的资源表达式。它的语法类似resourceKey="Autos"。当XmlSitemapProvider根据web.sitemap文件中的信息检索SitemapNode的时候，它根据SitemapNode属性的名称、resourceKey和为提供程序配置的si temapFile属性的值来检索字符串资源。使用示例中的"Autos"节点的时候，提供程序（provider）会根据当前的文化来查找以"web.sitemap"开头的资源文件。这意味着，对于一个发送法语头信息的浏览器来说，提供程序会查找名称为web.sitemap.fr.resx的资源文件。在这个资源文件中，提供程序会依据resourceKey + "." + [SitemapNode属性名]来查找资源键。例如，把"Autos"节点的Title属性当作例子，提供程序会在web.sitemap.fr.resx资源文件中查找键为Autos.Title的资源。

显式表达式使开发者对包含本地资源的文件和资源键（resource key）的名称有更强的控制能力。在示例web.sitemap中，第一个<si temapNode>元素使用了显式资源表达式。显式表达式在每个属性上指定。第一个<si temapNode>元素的Title属性使用了显式表达式。显式表达式必须以\$resource:开头。在这个标识符之后，开发者必须提供资源文件的根名称和资源键。开发者可以选择提供一个默认值。在例子中，表达式\$resources: Title, MyTitle, Home表明提供程序应该查看以"Title"开头的资源文件。对于发送法语头信息的浏览器开说，提供程序会查找Title.fr.resx资源文件。接下来提供程序查看键为MyTitle的资源。如果提供程序无法找到这种资源，它会把字符串"Home"作为默认值。

你可以运行示例来查看站点地图本地化的效果。把英语作为默认语言的浏览器会显示英语文本。如果使用IE，你可以通过点击"工具->Internet选项"，并在"通用"选项卡点击"语言"按钮，点击"添加"按钮并选择添加"法语"。如果需要，还需要选中法语并点击"向上移动"按钮，使它成为IE的默认请求语言。把默认的语言改成法语之后，刷新示例页面。请注意，Menu、Treeview和SitemapPath控件中的文本自动地显式为App_GlobalResources目录中存放的法语资源文件中的法语文本。

Web.sitemap的内容

```

<?xml version="1.0" encoding="utf-8" ?>
<sitemap xmlns="http://schemas.microsoft.com/AspNet/Sitemap-File-1.0" enableLocalization="true">
<sitemapNode url="-/Default.aspx" title=" $resources: Title, MyTitle, Home" description="Default page
description when no localized value exists." >
<sitemapNode url="-/Category.aspx" resourceKey="Category">
<sitemapNode title="Autos" description="Autos" url="-/Autos.aspx" resourceKey="Autos" />
<sitemapNode title="Games" description="Games" url="-/Games.aspx" resourceKey="Games" />
<sitemapNode title="Health" description="Health" url="-/Health.aspx" resourceKey="Health" />
<sitemapNode title="News" description="News" url="-/News.aspx" resourceKey="News" />
</sitemapNode>
</sitemap>

```

修改提供程序（Provider）返回的站点导航数据

存储在web.sitemap中、供XmlSitemapProvider使用的导航数据是静态的--这些数据被载入内存中并作为只读数据存储。但是，很多站点的导航结构是根据查询字符串的值来参数化的。例如，新闻组（newsgroup）站点可能拥有良好定义的页面结构（例如，主页、新闻类别页面和新闻内容页面），但是实际的内容可能会有很大的不同，这依赖于查询字符串中的标识符。尽管把每种可能的查询字符串值都存储在<si temapNode>元素中也是可能的，但是即使是中等数量的查询字符串值，也要求sitemap文件包含数百个<si temapNode>元素。

站点导航特性在SitemapProvider基类中暴露了SitemapResolve事件。可以使用Sitemap.SitemapResolve或直接使用提供程序SitemapProvider.SitemapResolve来执行这个事件。这个事件的返回值是一个SitemapNode实例。你可以在自己的事件处理程序中编写自定义逻辑来建立SitemapNode实例的层次结构。这个逻辑可以修改每个SitemapNode的属性，因此URL和Title等属性会反映查询字符串带有的数据信息。

下面的例子在global.asax中注册了一个事件处理程序。这个事件处理程序的代码是App_Code目录中的一个类。这个自定义的类复制与当前页面对应的SitemapNode实例。XmlSitemapProvider返回的节点都是只读的，而调用SitemapNode上的Clone方法返回的是可写入的节点。在实例中，如果给Clone传递了true值，将导致当前的SitemapNode和它的所有父节点都是可写入的。这个类的代码的其它部分检查当前的页面和当前页面的查询字符串，确定当前页面位于站点层次结构的什么位置。代码修改了URL和Title属性，包含一些额外的信息，这样SitemapPath控件显示的导航UI就反映了网站用户为到达当前页面的实际点击路径。

运行示例的时候，你开始位于站点的主页。SitemapPath控件也反映了这一点。点击任何链接都会带你进入分类页面，它显示相关新闻类别中的新闻链接。请注意，如果你把鼠标停留在SitemapPath控件的最后一个链接上，浏览器状态栏中显示的URL包含了查询字符串信息（它指定了新闻类别）。点击任何一个发布链接都会把你带回到新闻发布页面。如果你把鼠标停留在SitemapPath控件的链接上，可以注意到控件中的最后两个链接带有的URL和Title包含了点击路径的正确查询字符串和描述信息。如果你导航到站点的主页，并点击其它的新闻组和内容链接，SitemapPath控件会被更新并反映第二次点击的链接。


```

Public Class PathExpansionHandler
Public Shared Function ExpandPath(ByVal sender As Object, ByVal e As SiteMapResolveEventArgs) As SiteMapNode
' 获取当前和之前节点的引用
Dim nodeCopy As SiteMapNode = SiteMap.CurrentNode.Clone(True)
Dim tempNode As SiteMapNode = nodeCopy

' Check if there is a newsgroup type in the query string
Dim typeId As String = Nothing
Dim typeIdUrlEncoded As String = Nothing
If Not String.IsNullOrEmpty(e.Context.Request.QueryString("type")) Then
typeId = e.Context.Server.HtmlEncode(e.Context.Request.QueryString("type"))
typeIdUrlEncoded = e.Context.Server.UrlEncode(e.Context.Request.QueryString("type"))
End If

' 首先执行发布页面URL的固定
' 如果查询字符串中包含发布ID, 我们就知道当前节点式发布页面
If Not String.IsNullOrEmpty(e.Context.Request.QueryString("postingID")) Then
Dim postingID as string = _
e.Context.Server.HtmlEncode(e.Context.Request.QueryString("postingID"))
Dim postingIDUrlEncoded as string = _
e.Context.Server.UrlEncode(e.Context.Request.QueryString("postingID"))
Dim NewUrl As String = tempNode.Url + "?type=" + typeIdUrlEncoded + "&postingID=" + postingIDUrlEncoded
Dim NewTitle As String = tempNode.Title + ": " + postingID
tempNode.Url = NewUrl
tempNode.Title = NewTitle

tempNode = tempNode.ParentNode
End If

' 然后, 对新闻组页面进行固定
' 这时候nodeCopy 变量知悉了新闻组节点
If Not String.IsNullOrEmpty(e.Context.Request.QueryString("type")) Then
Dim NewUrl As String = tempNode.Url + "?type=" + typeIdUrlEncoded
Dim NewTitle As String = tempNode.Title + ": " + typeId
tempNode.Url = NewUrl
tempNode.Title = NewTitle
End If

' 最后返回当前节点
Return nodeCopy
End Function
End Class

```

URL映射

URL映射特性利用web.config中存储的配置信息把收到的请求重映射（remap）到不同的URL。重映射发生在对请求的所有其它处理操作之前。下面的例子演示的是重映射一个页面请求，实际上任意文件类型都可以把请求重映射到不同的URL。

定义重映射URL

URL映射的配置信息存储在web.config中。<urlMappings>元素中嵌套的每个<add>元素为重映射进入站点的（inbound）url定义了一条规则。url属性定义了进入站点的url的exact（原样）属性，URL映射特性会试图用它进行匹配操作。如果exact匹配操作发生了，就会给进入站点的URL重新写入mappedUrl属性值。请注意，这个特性不支持更高级的规则（例如基于通配符和正则表达式的匹配）。

示例web.config为大量的url定义了映射规则。示例使用的web.sitemap文件定义的大量带有URL值的节点都会被重映射。其结果是，URL映射和站点导航的组合使用，使得开发者可以用友好的url来定义导航结构，并使用URL映射把请求重新写到不同的页面来执行实际的处理过程。

当你运行示例的时候，请注意Menu和Treeview控件是如何根据web.sitemap文件中定义的站点结构来显示导航数据的。如果你把鼠标停留在Treeview控件或右上角的SiteMapPath控件的链接上方，状态栏中显示的url是一个友好的url链接。当你点击任何导航链接的时候，实际运行的页面是Default.aspx。但是，Menu、Treeview和SiteMapPath控件中的导航信息仍然反映为友好的URL结构。

在页面的底部你还可以看到Request.Path、Request.QueryString["category"]和Request.RawUrl返回的值。Request.Path和Request.QueryString["category"]返回的值一直反映重映射进入站点的url的结果。但是，Request.RawUrl的值反映了重映射之前的友好的url。当站点导航特性试图把url信息与sitemap文件包含的数据进行匹配的时候，它会使用Request.RawUrl。如果匹配的值没有找到，XmlSiteMapProvider就把Request.Path作为替代者。在例子中，所有的友好url在web.sitemap文件中都有条目，因此使用站点导航的控件一直根据友好的url来显示和引用节点。

```

<?xml version="1.0" ?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
<system.web>
<urlMappings enabled="true">
<add url="-/Category.aspx" mappedUrl="-/Default.aspx?category=default" />
<add url="-/Autos.aspx" mappedUrl="-/Default.aspx?category=autos" />
<add url="-/Games.aspx" mappedUrl="-/Default.aspx?category=games" />
<add url="-/Heal th.aspx" mappedUrl="-/Default.aspx?category=heal th" />
<add url="-/News.aspx" mappedUrl="-/Default.aspx?category=news" />
</urlMappings>
</system.web>
</configuration>

```

本栏目登载此文出于传递信息之目的，如有任何的问题请及时和我们联系！

无任何评论！

请您注意：

尊重网上道德，遵守《全国人大常委会关于维护互联网安全的决定》及中华人民共和国其他各项有关





发表评论：

昵 称：

GO

- 法律法规
- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
 - 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
 - 中国网游研发中心新闻留言板管理人员有权保留或删除其管辖留言中的任意内容
 - 您在中国网游研发中心留言板发表的作品，中国网游研发中心有权在网站内转载或引用
 - 参与本留言即表明您已经阅读并接受上述条款

联系EMAIL:

j<  j>  j<  j>  j< 