


```

string selectedValue = Request.Form[Request[PostBackEventTarget]]. ToString();
{
    case "0": SetCulture("hi-IN", "hi-IN");
    break;
    case "1": SetCulture("en-US", "en-US");
    break;
    case "2": SetCulture("en-GB", "en-GB");
    break;
    case "3": SetCulture("fr-FR", "fr-FR");
    break;
    default: break;
}
}
}
///
///从会话中取得文件, 如果控制给导航到同一程序中的一个新页面。
///
if (Session["MyUI Culture"] != null && Session["My Culture"] != null)
{
    Thread.CurrentThread.CurrentUI Culture = (CultureInfo)Session["MyUI Culture"];
    Thread.CurrentThread.CurrentCulture = (CultureInfo)Session["My Culture"];
}
base.InitializeCulture();
}
///
///使用参数设置当前的UI Culture和CurrentCulture
///
///
///
protected void SetCulture(string name, string locale) {
    Thread.CurrentThread.CurrentUI Culture = new CultureInfo(name);
    Thread.CurrentThread.CurrentCulture = new CultureInfo(locale);
    ///
    ///由用户把当前线程的文化集保存在会话中
    ///, 以便它能够在当前应用程序中跨页面应用。
    ///
    Session["MyUI Culture"] = Thread.CurrentThread.CurrentUI Culture;
    Session["My Culture"] = Thread.CurrentThread.CurrentCulture;
}

```

因此, 用户在他/她选择的语言中会看到此内容。我们需要把该文件选择保存到一个会话或一个Cookie变量中, 因为如果用户移动到同一应用程序中的其它一些页面, 那么, 当新的页面类一开始被实例化时, 该线程的文化信息将会丢失(HTTP是无状态的!)。注意, 在用户的会话到期时, 如果你不想丢失当前线程的文化信息, 那么你可以使用Cookies。

一旦我们从web应用程序中提取了所有的内容并且基于用户选择和使用Resources.TestWebsite.XXXPropertyName设置了Culture和UI Culture, 那么, 我们就已经为我们的全球化框架作好了准备。现在, 剩下的唯一事情是把资源特定的数据添加到相应的资源文件中。针对每一种文件类型, 我们需要有一个单独的(和适当命名的)资源文件。这个过程称为本地化。在我的web.config文件中, 我使用了下列属性:

注意, 这里使用了编码属性-utf-8(8位Unicode转换格式), 因为它是可变长度字符编码; 并且, 除了它是ASCII兼容的之外, 还能够代表例如Greek, Arabic等语言。有关UTF-8编码的更多信息, 请参考下面这个链接:

<http://en.wikipedia.org/wiki/UTF-8>

另外, 特别值得注意的是, 尽管我们能够在发布服务器上拥有原始XML形式的资源文件(这样, 用户可以方便地编辑它们而不必重新编译整个站点), 但是, 如果我们对资源文件作出任何修改的话, 应用程序将重新开始运行。这有可能妨碍此发布的应用程序的性能。

Column Name	Data Type	Allow Nulls
FeedbackID	int	<input type="checkbox"/>
Comments	text	<input type="checkbox"/>
CreatedOn	datetime	<input type="checkbox"/>
EmailID	varchar(30)	<input checked="" type="checkbox"/>
DepartmentID	int	<input checked="" type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

二、设置语言方向相应的dir属性

许多时候, 我们还需要设置本地化文本的方向(这是使用或标签的dir属性设置的)。这是必需的, 因为有些语言从右到左(RTL)读取, 例如Arabic, 不同于象Hindi和English这样语言的标准的从左到右(LTR)的读取方式。这可以通过把.resx文件中的dir属性设置为适当的值来实现。

首先, 你可以在所有资源文件中创建一个Direction(你可以使用任何名)域, 并基于单个资源文件把它的属性设置为RTL或LTR。对于Arabic, 这个域的值是RTL, 而对于Hindi则是LTR。然后, 把标签的dir属性设置为如下:

这样就可以设置正确的方向, 因为该值来自于资源文件(基于当前线程文化)。

三、使用数据库实现本地化

我们已经看到了如何本地化控件的文本和UI描述。但是, 存储在数据库中的内容会怎么办呢? 其实, 这一部分内容也需要本地化, 但是由于它存储在一个DB中, 所以我们不能使用资源文件来实现相同目的。为此, 我们需要创建新的表格。

假定我有一个存储用户评价的表格。该表格结构如下所示:

Column Name	Data Type	Allow Nulls
FeedbackID	int	<input type="checkbox"/>
Comments	text	<input type="checkbox"/>
CreatedOn	datetime	<input type="checkbox"/>
EmailID	varchar(30)	<input checked="" type="checkbox"/>
DepartmentID	int	<input checked="" type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>

现在，我们想实现以本地化的文本来显示Comments和Name字段，但是，我们不可能把所有这些域的不同语言版本都存储在一个表格中(既然存在不需要被本地化但却重复的其它域)。因此，我们需要重新组织该表格结构并且创建另一个表格来存储这两个域的本地化版本。首先，我们需要从这个表格中删除这两个域并创建一个如下所示的新表格：

Column Name	Data Type	Allow Nulls
FeedbackID	int	<input checked="" type="checkbox"/>
CultureID	int	<input checked="" type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
Comments	text	<input checked="" type="checkbox"/>

在此，我们添加了一个新域：CultureID，它等价于LCID（或Locale标识符）。我们能够按如下所示添加文化特定的本地化数据：

FeedbackID	CultureID	Name	Comments
1	1033	US Name	US Comments
2	1036	Nom Français	Nom Français
NULL	NULL	NULL	NULL

现在，我们可以使用以CultureID(LCID)作为参数的SQL查询来取得本地化内容。我们还能够提供一个用户接口来把本地化数据输入到这样的表格以便能够以一种交互方式创建相应的内容。

四、总结

在本文中，我们讨论了在ASP.NET 2.0开发中有关实现全球化的一些重要方面，并且看到，这是非常容易实现的事情；但是，也存在许多值得注意的重要问题：

1. 不要依赖于web浏览器的设置。你可以在应用程序中显示一个链接(可以在头部位置)以便用户能够通过点击它来选择他们的语言。
2. 使用资源文件来把GUI中与描述相关的数据分离开来。资源fallback是ASP.NET使用的方法-当它不能找到相应于一种特定文化的资源文件时。它将首先试用中立资源文件，然后是缺省的资源文件或fallback资源文件(TestSisterResource.resx)。
3. 使用数据库表格把数据存储到一个DB中。为此，你需要创建单独的表格来存储本地化内容。
4. 如果你使用sn.exe来为你的主应用程序程序集创建一个强类型名，那么，你需要使用同一对密钥对(由sn.exe生成)中的私钥来签名你的小程序集；因为，强类型名字的程序集要求小程序集也应该是强类型名字。

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

发表评论：

昵称：

联系EMAIL：





