

第四篇 应用程序

第一章 什么是应用程序

在 asp.net 中，可以这样来定义一个 Application：能够在 web 应用服务器的子目录或者虚拟目录上运行的所有的文件、页面、操作、模块或者能被执行的代码。比方说，在一个 web 服务器上，一个“order”应用程序将会在“/order”这个目录下被发布。

Web 服务器上的 asp.net 应用程序在一个被称作应用程序域运行空间(AppDomain)环境中被执行，以保证类的隔离（没有版本、名称上的冲突）、安全屏蔽（防止有权访问某些机器/网络的资源）、静态变量的隔离。

在一个 web 应用程序的生命周期中，asp.net 维护一个 HttpApplication 实例池。Asp.net 对一个 Http 的请求会自动分配一个来处理，这个特别的 HttpApplication 实例对管理这个在全部的生命周期里的请求是可靠的，并且在处理完成后可以被重用。

在应用程序环境下，ASP.NET 并发处理客户端的请求，所以可能存在多线程对 Application 对象的同时存取。在这种情况下，对 Application 对象的草率处理，可能会导致不可预知的错误。例如以下代码：

```
<% Application("counter") = CType(Application("counter") + 1, Int32) %>
```

原本希望对实例进行计数，但如果同时到达两个以上请求时，则有可能产生漏计。正确的方法应该是在操作以前，对 Application 对象上锁，操作完成以后，再对 Application 对象解锁。代码如下：

```
<%  
Application.Lock()  
Application("counter") = CType(Application("counter") + 1, Int32)  
Application.UnLock()  
%>
```

4.1.1 配置应用程序的步骤

4.1.1.1 设置应用程序的目录结构

一个 WEB 站点可以有多个应用程序运行，而每一个应用程序可以用唯一的 URL 来访问，所以首先应利用 IIS 开放应用程序的目录为“虚拟目录”。各个应用程序的“虚拟目录”可

以不存在任何物理上的关系。例如：

应用 URL:	物理路径：
http://www.my.com	c:\inetpub\wwwroot
http://www.my.com/myapp	c:\myapp
http://www.my.com/myapp/myapp1	\\computer2\test\myapp

从“虚拟目录”上看来，http://www.my.com/myapp 和 http://www.my.com/myapp/myapp1 似乎存在某种联系，但实际情况却是，我们看到两者完全分布于不同的机器上，更不用说物理目录了。

4.1.1.2 . 设置相应的配置文件

根据应用的具体需要，可以拷入相应的 global.asax 和 config.web 配置文件，并且设置相应的选项。（配置文件的设置具体见相关章节）

global.asax 主要配置 application_start、applicatoin_end、session_start、session_end 等事件。

4.1.1. 3 . 把应用所涉及的各种文件放入“虚拟目录”中

把.aspx 文件、.ascx 文件以及各种资源文件分门别类放入应用目录中，把类引用所涉及的集合放入应用目录下的 bin 目录中。

4.1.2 应用程序框架

```
<%@ Application attribute="value" [attribute=value ... ]%>
```

```
<%@ Import namespace="value" %>...
```

```
<%@ Assembly Name="assemblyname" %>
```

```
<script language="vb" runsat=server>
```

```
...
```

```
</script>
```

```
<body>
```

```
<form runat=server>
```

```
...
```

```
</form>
```

</body>

</html>

说明：

1 . <%@ Application attribute="value" [attribute=value ...]%>

让 ASP.NET 运行环境动态从另一个应用中动态编译出一个类来继承使用。

例如：

<%@ Application Inherits="MyApp.Object" Description="Ourapp" %>

指定应用环境从 Myapp 应用中动态编译一个 MyApp.Object 的类以供使用 ,它的说明为“ Ourapp”。

2 . <%@ Import namespace="value" %>...

显视导入一个命名空间到应用程序 ,这样应用程序就可以使用命名空间中定义的各种类和接口来完成特定的功能 ,大大加快了程序的开发速度。

例如：

<%@ Import Namespace="System.IO" %>

<%@ Import Namespace="System.NET" %>

就可以利用系统为我们提供的大量文件和网络对象 ,快速的开发自己的文件和网络应用程序。

3 . <%@ Assembly Name="assemblyname" %>

在页面编译时产生到 assemblyname 的连接 ,这样就可以使用集合中类及接口。缺省情况下 ,应用会把应用程序目录下 bin 中的集合都动态载入。该项功能也可以在应用程序的 config.web 中配置 ,缺省情况下 ,config.web 中有如下形式：

<assembly>

<add assembly="*" />

</assembly>

即缺省情况下 ,加载 bin 下的所有集合。

又如：

<%@ Assembly Name="myassembly.dll" %>

加载 bin 下 myassembly.dll 集合

4 . 其他

<script>、</script>对之间的代码通常是各种事件的定义 ,诸如页面开始时、某个按钮被触发时所要做的事情。<body>、</body>和<form>、</form>之间通常是页面的界面要素 ,为显示给客户端的可视界面。

4.1.3 创建应用程序的典型步骤

4.1.3.1 配置 config.web

主要定义为 gb2312 字符集，以利于中文显示

```
<configuration>  
<globalization requestencoding="gb2312" responseencoding="gb2312" />  
</configuration>
```

4.1.3.2 配置 global.asax

主要定义应用初始化、结束，会话开始、结束，请求开始、结束等事件发生时，应用要做的事情。

```
<script language="VB" runat="server">  
    Sub Application_Start(Sender As Object, E As EventArgs)  
    End Sub  
  
    Sub Application_End(Sender As Object, E As EventArgs)  
    End Sub  
  
    Sub Session_Start(Sender As Object, E As EventArgs)  
    End Sub  
  
    Sub Session_End(Sender As Object, E As EventArgs)  
    End Sub  
  
    Sub Application_BeginRequest(Sender As Object, E As EventArgs)  
    End Sub  
  
    Sub Application_EndRequest(Sender As Object, E As EventArgs)  
    End Sub  
  
</script>
```

4.1.3.3 主程序

创建一个应用程序我们可以先在 web 服务器上创建一个虚拟目录或者在发布目录下创

建一个新的目录。装过 Windows 2000 Advance Server 的读者会知道，安装完成后，会有一个 c:/inetpub/wwwroot 的目录，你可以通过 IIS 管理工具来创建一个新的目录或者虚拟目录。一个应用程序可能含有大量的.aspx 文件、.ascx 文件、由其他工具产生的 assembly 集合以及页面中用到的各种资源文件（声音、图片、动画等等），这里就不再一一介绍了。

下面我们就创建一个简单的 aspx 页面来说明一个 Application 的应用，它只含有一个.aspx 文件，在用户浏览时显示“hello world”，可谓最简单的 web 应用了。

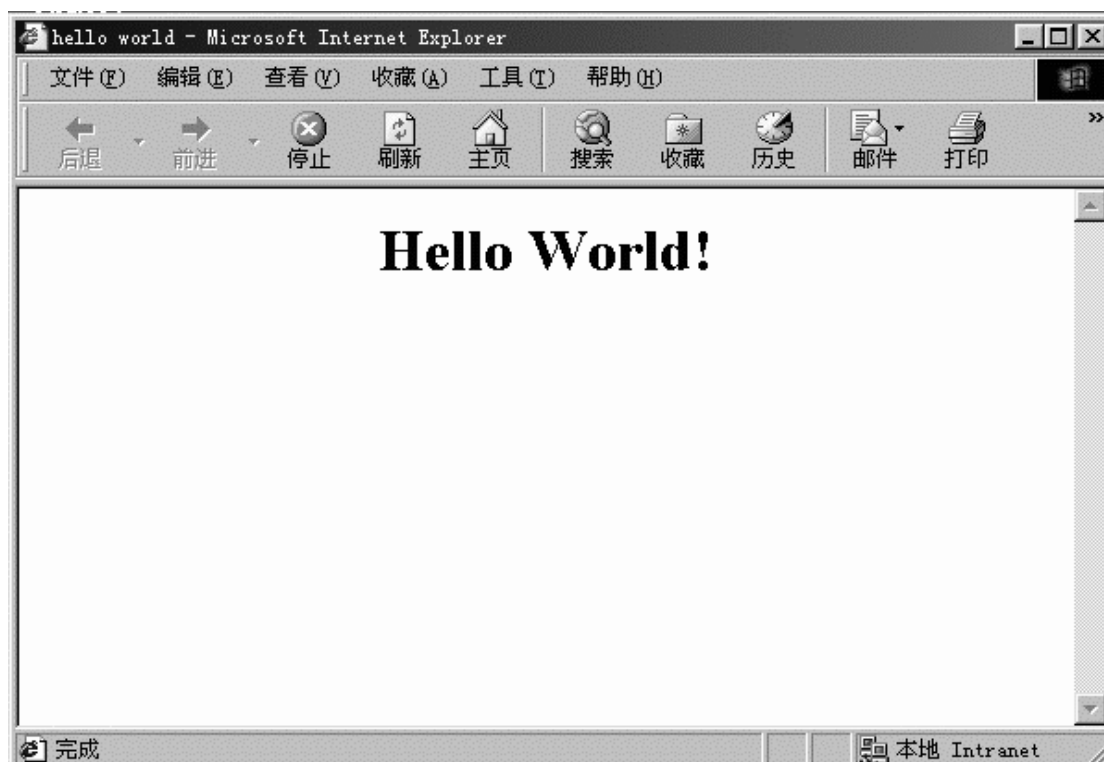
```
<!-- 文件名：application/FormAppHellp.aspx -->
<% @Page Language="VB"%>
<html>

<head>
  <title>
    hello world
  </title>
</head>

<body>
  <center>
    <h1><% Response.Write("Hello World!") %></h1>
  </center>
</body>

</html>
```

下面就是我们的运行效果：



4.1.4 小结

asp.net 平台的应用，通过指定虚拟目录，使得一个主机多个地址多个应用成为可能。采用 asp.net 开发应用程序带来的好处是：程序集中可方便打包，配置的层次结构更加灵活方便，应用独立运行于自身的应用环境中更加安全可靠。

配置一个应用的过程大致为：1) 指定应用目录为 IIS 的虚拟目录 2) 为应用设置适当的配置权限（配置 global.asax 和 config.web 文件）3) 在自己的应用目录下放置事先编好的程序。

从页面应用所支持的 Application、Import、Assembly 等标识看来，asp.net 对对象重用的支持大大加强了，ASP.NET 的“通用语言运行库”概念的提出，为实现各种开发语言的合作编程奠定了基础。

第二章 配置 Config.web

4.2.1 ASP.NET 配置简介

ASP.NET 提供了一个丰富而可行的配置系统，以帮助管理人员轻松快速的建立自己的 WEB 应用环境。ASP.NET 提供的是一个层次配置架构，可以帮助 WEB 应用、站点、机器分别配置自己的扩展配置数据。ASP.NET 的配置系统具有以下优点：

ASP.NET 允许配置内容可以和静态内容、动态页面和商业对象放置在同一应用

的目录结构下。当管理人员需要安装新的 ASP.NET 应用时，只需要将应用目录拷贝到新的机器上即可。

ASP.NET 的配置内容以纯文本方式保存，可以以任意标准的文本编辑器、XML 解析器和脚本语言解释、修改配置内容。

ASP.NET 提供了扩展配置内容的架构，以支持第三方开发者配置自己的内容。

ASP.NET 配置文件的更修被系统自动监控，无须管理人员手工干预。

4.2.2 配置文件的规则

ASP.NET 的配置文件是基于 XML 格式的纯文本文件，存在于应用的各个目录下，统一命名为“config.web”。它决定了所在目录及其子目录的配置信息，并且子目录下的配置信息覆盖其父目录的配置。

WINNT\Microsoft.NET\Framework\版本号\下的 config.web 为整个机器的根配置文件，它定义了整个环境下的缺省配置。

缺省情况下，浏览器是不能够直接访问目录下的 config.web 文件。

在运行状态下，ASP.NET 会根据远程 URL 请求，把访问路径下的各个 config.web 配置文件叠加，产生一个唯一的配置集合。举例来说，一个对 URL: http://localhost/webapp\owndir\test.aspx 的访问，ASP.NET 会根据以下顺序来决定最终的配置情况：

- 1 . \Microsoft.NET\Framework\v.1.00\config.web (缺省配置文件)
- 2 . \webapp\config.web (应用的配置)
- 3 . \webapp\owndir\config.web (自己的配置)

4.2.3 配置文件的语法规则

1) 标识

配置内容被置于 config.web 文件中的标记<configuration>和</configuration>之间。

格式：

```
<configuration>  
    配置内容  
    ...  
</configuration>
```

2) 配置段句柄说明

ASP.NET 的配置文件架构并未指定任何文件格式或者是支持的配置属性。相反的，它提出了“配置段句柄申明”的概念来支持任意的用户定义配置段。

格式：

```
<configsections>  
    <add name=欲定义配置段名 type=处理的句柄函数 />  
</configsections>
```

3) 配置段

具体定义配置的内容，供应用使用。

以下例子定义了一个“httpmodules”配置段，设置了系统 http 相关的处理模块

```
<configuration>

  <configsections>
    <add name="httpmodules" type="System.Web.Configuration.HttpModules
ConfigurationHandler" />
  </configsections>

  <httpmodules>
    <add type="System.Web.SessionState.CookielessSessionModule" />
    <add type="System.Web.Caching.OutputCacheModule" />
    <add type="System.Web.SessionState.SessionStateModule" />
    <add type="System.Web.Security.WindowsAuthenticationModule" />
    <add type="System.Web.Security.CookieAuthenticationModule" />
    <add type="System.Web.Security.PassportAuthenticationModule" />
    <add type="System.Web.Security.CustomAuthenticationModule" />
    <add type="System.Web.Security.UrlAuthorizationModule" />
    <add type="System.Web.Security.FileAuthorizationModule" />
  </httpmodules>

</configuration>
```

4.2.4 ASP.NET 定义的标准配置段

- 1) httpmodule 段： 定义了应用的 http 请求的处理模块以及诸如安全、日志之类的应用方式
- 2) httphandlers 段： 负责映射 URLs 到 IHttpHandler 类
- 3) sessionstat 段： 负责配置 http 模块的会话状态
- 4) globalization 段： 配置应用的公用设置
- 5) compilation 段： 配置 ASP.NET 的编译环境
- 6) trace 段： 配置 ASP.NET 的跟踪服务
- 7) security 段： ASP.NET 的安全配置
- 8) iisprocessmodel 段： 在 IIS 上配置 ASP.NET 的处理模式
- 9) browsercaps 段： 配置浏览器的兼容部件

4.2.5 一个配置读出的例子

- 1) config.web 配置文件

```
<!--config.web 请放入 FormCfg.aspx 所在目录-->
<configuration>
<!--申明一个 test 配置段-->
```



```

<configsections>
    <add name="test" type="System.Web.Configuration.DictionarySectionHandler" />
</configsections>

<test>
<!--配置一个键 key,其内容为 just a configure test-->

    <add key="key" value="just a configure test" />
</test>

</configuration>

```

2) 读出其内容

```

<!--文件名：Application/FormCfg.aspx-->
<html>
<head>
<script language="VB" runat=server>
sub page_load(s as object ,e as eventargs)
'取出 test 配置段的 key 键的值
    Dim CfgSection As Hashtable = Context.GetConfig("test")
    Dim Msg As String = CStr(CfgSection("key"))

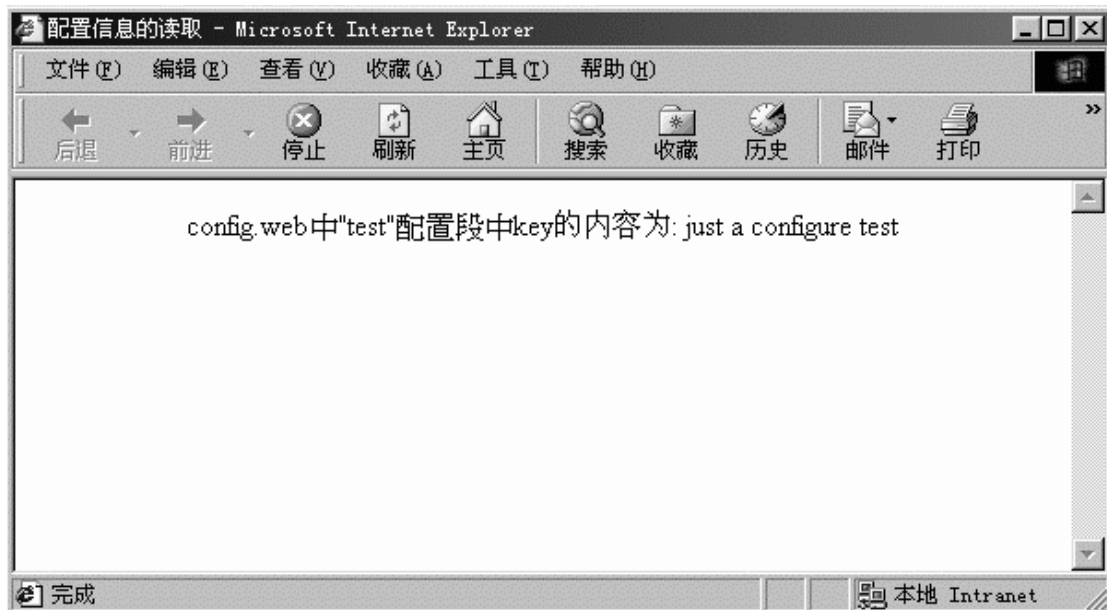
    lblMsg.text=Msg
end sub
</script>
<title>
配置信息的读取
</title>
</head>

<body>
<center>
config.web 中"test"配置段中 key 的内容为:
<asp:label id=lblmsg runat=server />
</center>
</body>

</html>

```

3) 运行结果



4.2.6 Config.web 配置实例

```
<configuration>
```

```
<!--定义用户应用的公用设置，如 SQL 的 sql 连接串等等-->
```

```
<appsettings>
```

```
</appsettings>
```

```
<!--设置浏览器的兼容性部件-->
```

```
<browsercaps>
```

```
</browsercaps>
```

```
<!--编译环境设置，非调试模式-->
```

```
<compilation debugmode="false">
```

```
<!--缺省编译语言为 vb，以后可以不再在 Page 中定义脚本语言-->
```

```
<compilers defaultlanguage="vb">
```

```
<!--以 MSVSA.dll 编译.vb 为后缀的 VB 文件-->
```

```
<compiler language="VB" extension=".vb" type="MSVSA.dll#Microsoft.VB.Compiler"/>
```

```
</compilers>
```

```
<assemblies>
```

```
<!--加入对 System.Data 的引用-->
```

```
<add assembly="System.Data" />
```

```
<!--去掉对 System.Data 的引用-->
```

```
<remove assembly="System.IO" />
```

```
<!--去掉 config.web 中包含或继承来的引用-->
```

```
<clear />
```

```
</assemblies>
```

```

</compilation>

<!--设置应用全局环境-->
<!--文件、请求、返回以 gb2312 编码，以保证浏览器正确显示中文-->
<globalization fileencoding="gb2312" requestencoding="gb2312"
responseencoding="gb2312"/>

<!--定义用户出错的处理-->
<!--出错缺省显示 defaultredirect 指定的页面,mode 为 on 时，遵循 customerrors 配置段-->
<!--mode 为 off 时,忽略用户出错，mode 为 remoteonly 时，本地才显示真正的出错原因-->
<customerrors defaultredirect="AnErrorHasOccured.aspx?ErrNum=-1" mode="remote">
<!--当出错码为 500 时,显示 redirect 指定的页面-->
<error statuscode="500" redirect="AnErrorHasOccured.aspx?ErrNum=500"/>
</customerrors>

<!--指定目录 webapp 的访问权限-->
<location path="webapp" >
<!--非授权用户不能进入 webapp 目录-->
<security>
<authorization>
<deny users="?" />
</authorization>
</security>
</location>

<!--定义安全属性-->
<security>
<authorization>
<!--角色为 Administrators 和所有的用户访问其指定的资源-->
<allow roles="Administrators"/>
<allow users="*" />
</authorization>
</security>

</configuration>

```

4.2.7 小结

Config.web 是 aspx 区别于 asp 的一个方面，我们可以用这个文件配置我们的很多信息。

第三章 编写 global . asax

为了编写用户界面的应用程序,开发者可以把应用程序标准的逻辑和时间处理的代码加到 Web Application 里面。这些代码不产生用户界面,也不想英单个得页面的请求。事实上,这些代码处理更高水平的事件,如 Application_Start, Application_End, Session_Start, Session_End,等等。开发者通过放在 web 应用程序根目录下面的 Global.asax 来响应这些事件。

Asp.net 通过一个动态的.NET Framework 类自动解析和编译这个文件,这个类就是 HttpApplication 基类,在第一时间里面,在这个文件里面的应用程序的资源将会被响应。

首先,在包含有请求的应用程序名字空间中被访问之前,Global.asax 将被解析和编译成.NET Framework 的一个类。这个文件本身有拒绝被访问的配置。

下面我们来看看这个文件里面的具体内容,首先我们声明这个文件的使用语言、运行环境:

```
<script language="VB" runat=server>  
‘相关方法  
</script>
```

然后我们就可以定义各种方法了,

```
Sub Application_Start()  
    ‘方法的属性等  
End Sub
```

如果事件处理代码需要用到名字空间,我们可以这样来引用它:

```
<%@ Import Namespace="System.Data.SQL"%>
```

下面我们来看看这个文件的具体应用,首先我们在我们的 Web Server 上建立一个 Global.asax 文件,我们在里面加上我们的代码:

```
<script language="VB" runat=server>  
‘相关方法
```

```
Sub Application_Start()  
    ‘方法的属性等  
End Sub
```

```
Sub Application_Start(Sender As Object, E As EventArgs)  
    Application.Lock()  
    Application("counter") = CType(Application("counter") + 1, Int32)
```

```

        Application.Unlock()
    End Sub

    Sub Application_End(Sender As Object, E As EventArgs)
        ' Clean up application resources here
    End Sub

    Sub Session_Start(Sender As Object, E As EventArgs)
        Response.Write("Session 正在启动...<br>")
    End Sub

    Sub Session_End(Sender As Object, E As EventArgs)
        ' Clean up session resources here
    End Sub
</script>

```

当然，我们还要配置 Config.web，用来指定出错信息的打印页面。根据上面我们配置 Config.web 的经验，我们很容易的就可以对这个文件进行配置：

```

<configuration>
  <customerrors mode="on" defaultredirect="error.htm" />
  <globalization requestencoding="gb2312" responseencoding="gb2312" />
</configuration>

```

第二句话就是配置我们指定的出错页面语句。我们写两个页面来实现它，一个为出错页面，一个为实现这个功能的 aspx 页面。出错页面很简单的，就是报告程序出错时显示的信息，我们就写“在 config.web 里面配置的连接！”，是经过 aspx 页面甩出来的。

在 aspx 页面，我们用下面的语句来响应出错按钮点击事件：

```

Sub Error_Click(Sender As Object, E As EventArgs)
    ' 甩出异常！
    throw New Exception()
End Sub

```

以外我们的响应 Session 的方法用下面的语句来说明：

```

Sub Session_Click(Sender As Object, E As EventArgs)
    Session.Abandon()
    Response.Redirect("global.aspx")
End Sub

```

下面是完整的代码：

```

<html>

```

```

<script language="VB" runat="server">
    '页面导入
    Sub Page_Load(Sender As Object, E As EventArgs)
        Response.Write("正在装入页面...<br>")
    End Sub

    'Session 事件
    Sub ssaidy(Sender As Object, E As EventArgs)
        Session.Abandon()
        Response.Redirect("global.aspx")
    End Sub

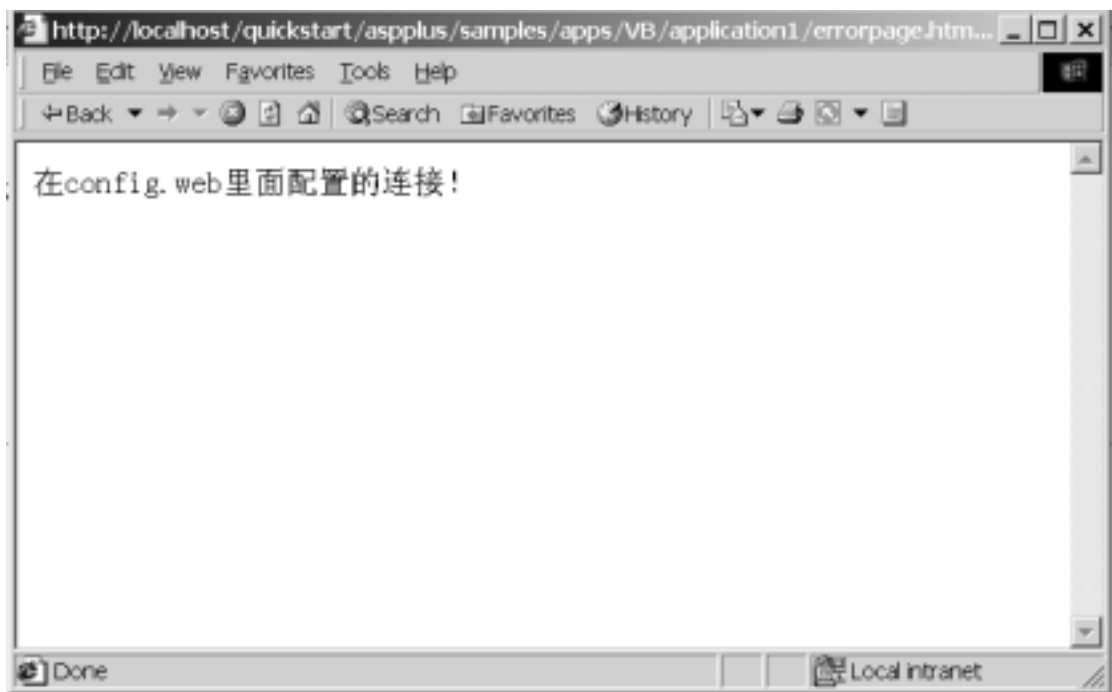
    '响应错误方法
    Sub esaidy(Sender As Object, E As EventArgs)
        '抛出异常
        throw New Exception()
    End Sub
</script>
<body>
    <br><br><br>
    <center>
        <form runat="server">
            <input type="submit" Value="刷新这个页面" runat="server"/>
            <input type="submit" OnServerClick="ssaidy" Value="结束这个 Session"
                runat="server"/>
            <input type="submit" OnServerClick="esaidy" Value="错误表示" runat="server"/><p>
            <hr>
        </form>
    </center>
    <br><br>
</body>
</html>

```

运行结果如下：



点击“错误表示”按钮，显示如下：



4.3.1 小结

讲述了配置文件 Global.asax 的配置问题，Global.asax 文件对一个.NET 技术构建的 WEB 站点来讲，是非常必须的，本章我们的内容就是针对它讲的。

第四章 Application 和 Session

4.4.1 Application 对象

在讲述 ASP.NET 的 Application 对象之前,我们先来回顾一下 ASP 的 Application 对象。我们知道由于变量的生命周期受限于网页,所以每当.asp 文件被解释执行完毕之后时,变量就会被释放,它的内容将不存在。而在编程过程中,我们有时又需要在页面之间传递变量的内容。例如,我们在一个登录页面中输入了用户的名字,为了使页面个性化,在后面的页面显示中,我们希望知道前面输入的用户名,以便于更好的人机交互。这就要求有一种变量传递的机制。人们最常用的保存变量的内容的方法是使用文件,但是毕竟对文件的操作是比较麻烦的事情,有没有更简单的方法呢?其中一种比较简单的方法就是使用 Application 对象来保存我们希望传递的变量。由于在整个应用程序生存周期中,Application 对象都是有效的,所以在不同的页面中都可以对它进行存取,就像使用全局变量一样方便。在 asp.net 环境下,Application 对象来自 HttpApplicationStat 类。它可以在多个请求、连接之间共享公用信息,也可以在各个请求连接之间充当信息传递的管道。

4.4.1.1 使用 Application 对象

Application 对象重要的属性：

- **All 属性**, 返回应用中保存的所有的公用对象数组

例如：

```
dim MyObjects() as object
MyObjects=Application.All
```

表示用 myobjects 取得了当前应用保存的所有对象

- **AllKeys 属性**, 返回应用中保存的公用对象的名字数组 (标识数组)

例如：

```
dim MyVars() as String
MyVars=Application.AllKeys
```

即取得了所有保存的公用对象的标识名字到 myvars 数组

- **Contents 属性**, 返回 this 指针, 主要是为了和 asp 兼容而保留

- **Count 属性**, 返回当前应用中保存的公用对象的数目

例如：

```
dim VarNum as integer
VarNum=Application.count
```

- **Item 属性**, 返回当前应用中保存的公用对象集合中的指定对象, 这是最常用的属性。

例如我们前面讨论的,记录变量内容的问题,就是通过 item 属性来保存的。

Application.Item(变量名)=要保存对象

但是通常我们都会省去 item 属性写成：

Application(变量名)=要保存对象

这里需要注意的是，Application 保存的对象为应用程序所共享，而.net 平台又是一个多用户多线程的环境，因而 Application 保存的对象在使用时，要注意避免冲突。

例如：

```
Application ( "counter" ) = Application ( "counter" ) +1
```

它使开始用户保存的数值加 1，我们可以利用它来统计页面浏览的次数。但是有一个问题发生了，那就是如果另外一个页面也使用了上述语句，那么混乱就产生了。设想一下如下情况，用户 a 对页面 a 访问，使 counter+1,然后用户 b 对页面 b 的访问，counter 又增加了 1，实际上无论对页面 a 还是页面 b，访问都只有一次，counter 却增加了 2 次，由于记数变量的相同使得我们统计页面的努力化为泡影。

· **StaticObjects 属性**，返回在应用程序文件中如<object runat=server></object>定义的对象集合。

下面，我们对上面学到的各种属性进行应用。

例子：

我们首先产生 6 个 Application 变量，然后分别用 item 属性和 all 属性去逐一取出各个 Application 变量的内容显示出来。注意为了避免其他公用 Application 变量的干扰，我们在页面加载时，调用了 removeall 方法，清空应用的所有公用变量。

1. 程序源代码

```
<!-- 文件名：Application\FormApplication.aspx -->
```

```
<html>
```

```
<script language="vb" runat=server>
```

```
Sub Page_Load(o as object,e as eventargs)
```

```
dim i as integer
```

```
dim tStr as String
```

```
dim sStr as String
```

```
dim strArray() as String
```

```
dim tObject() as Object
```

```
dim ObCol as HttpStaticObjectsCollection
```

```
If Not IsPostBack
```

```
Application.removeall
```

```
'为防止其他变量干扰，使用前清掉所有的保存变量
```

```
'保存六个变量
```

```
for i=1 to 6
```

```
tStr="变量名" & i
```

```
sStr="内容" & i
```

```
Application(tStr)=sStr
```




3. 当按下显示按钮后，显示 application 内容的画面：



Application 对象重要的方法调用：

- Add 方法，加入一个对象到 Application 对象的 Stat 集合中

例如：

```
Application.Add("string1","test")
```

表示向 Application 的 stat 集合中加入一个名为 string1 的值为"test"的字符串，其实它的效果和

```
Application("string1")="test"
```

以及 Application.item("string1")="test" 是一样的。

·**Remove 方法**,根据给出的标识从 Application 对象的 Stat 集合中删去

例如：

```
Application.Remove("string1")
```

表示把标识为 string1 的共享对象 string1 从 Application 对象的 Stat 集合中删去。使用它可以清除用 Add 方法添加的对象。

·**RemoveAll 方法**，把 Application 对象 Stat 集合中的所有对象清除，在对我们属性的使用举例中，我们已经见到了它的用法，但是值得小心，我们不提倡使用它，因为在编程中你并不清楚，是否有其他页面要依赖于某个 Application 的公用变量，一旦清除将造成不可预知的错误。

·**Clear 方法**，作用和 RemoveAll 方法一样。

·**Get 方法**，允许使用名字标识或者是下标，来取得 Application 对象 stat 集合中的对象元素。

例如：

```
dim tmp as object
```

```
tmp=Application.Get("string1")
```

```
或者 tmp=Application.Get(0)
```

表示从 Application 对象的 Stat 集合中取得标识为 string1 或者下标为 0 的对象

它等价于：

```
tmp=Application("string1")
```

```
tmp=Application(0)
```

或者是

```
tmp=Application.item("string1")
```

```
tmp=Application.item(0)
```

·**Set 方法**，修改 Application 对象 stat 集合中指定标识所对应的对象的值。

例如：

```
Application.Set("string1","try")
```

就把我们开始为 string1 变量设置的值"test"改为"try"了，它和下边的形式也是一样的：

```
Application("string1")=try
```

·**GetKey 方法**，根据给定的下标取得 Application 对象的 stat 集合中相应对象的标识名。

例如：

```
dim nameStr as String
```

```
nameStr=Application.GetKey(0)
```

表示取得 Application 对象中 Stat 集合的第一个对象的标识名

·**Lock 方法**，锁住其他线程对 Application 对象中 stat 集合的访问权限。这个方法主要是用来防止对 Application 的变量操作过程中，其他并发程序可能造成的影响。比如在记数过程中，如果不进行上锁操作，就有可能发生脏读脏写。例如，开始从变量中取得记数值 1，如果在记数并写回到变量之间，另一页面对它发生了一次记数，并先行写回变量，那么最终写回到变量中的值为 2，而并不是实际的 3。如果采用了上锁机制，在页面读出变量到记数并写回变量的过程中，即使发生了另一次记数，由于变量被锁住，它也不可能在变量被写回以前取得成功，只有等待变量释放，从而形成两者对变量操作的串行性，避免了数据的脏读和脏写。

·**Unlock 方法**，对 Application 对象 Stat 集合锁定的解锁操作，释放资源以供其他页面使用。

下面我们就上边学到的方法做一个例子，为了强调 lock 方法和 unlock 方法，我们将单独举一个例子。例子是这样的，开始我们创建 6 个 Application 变量，赋以数值序号，页面有 3 个按钮，分别是加 1，减 1 和清除。当点击“加 1”按钮后，我们会看到变量的值都会增加 1，当点击“减 1”按钮后，变量值都减 1，当按下清除后，变量都消失了。在清除功能中，我们为了同时演示 remove 和 clear 方法，采用最后三个用 clear 清除，其他的逐一用 remove 清除。

1. 源程序

```
<!-- 文件名：Application\FormApplication01.aspx -->
```

```
<html>
```

```
<script language="vb" runat=server>
```

```
Sub Page_Load(o as object,e as eventargs)
```

```
dim i as integer
```

```
If Not IsPostBack
```

```
for i=1 to 6
```

```
application.add("item"&i,i)
```

```
next
```

```
End If
```

```
response.clear
```

```
for i=0 to application.count-1
```

```
response.write(application.GetKey(i) & "=" & application.Get(i) & "<br>")
```

```
next
```

End Sub

Sub AddOne(s as object,e as eventargs)

'变量值加 1

dim i as integer

dim j as integer

dim t as string

for i=0 to Application.count-1

j=Application.Get(i)+1

t=Application.GetKey(i)

Application.Set(t,j)

next

Page_Load(s,e)

'刷新画面

End Sub

Sub SubOne(s as object,e as eventargs)

'变量减 1

dim i as integer

dim j as integer

dim t as string

for i=0 to Application.count-1

j=Application.Get(i)-1

t=Application.GetKey(i)

Application.Set(t,j)

next

Page_Load(s,e)

'刷新画面

End Sub

Sub Gone(s as object,e as eventargs)

'清空所有变量

dim i as integer

for i=0 to Application.count-3

Application.Remove(i)

next

'演示 remove 方法

Application.clear

'演示 clear 方法

```
Page_Load(s,e)
'刷新画面
End Sub
</script>
<head>
<title>
Application 方法试验
</title>
</head>
<body bgcolor=#ccccff>
<center>
<h2>Application 方法试验</h2>
<hr>
<form runat=server>
<asp.button type="submit" text="+1" OnClick="AddOne" runat=server />
<asp.button type="submit" text="-1" OnClick="SubOne" runat=server />
<asp.button type="submit" text="清空" OnClick="Gone" runat=server />
</form>
</center>
</body>
</html>
```

2. 开始时，输出画面:



3. 当两次点击+1后，输出的变量值：



4. 当点击“-1”后，变量的值为：



5. 当点击清空后，输出的画面



接下来，我们看一个使用 lock 和 unlock 方法制作计数器的例子：Application 对象对不同的连接者是共用的，因此适合制作计数器。

程序代码如下：

```
<!-- 文件名：application\FormAppLock.aspx -->
<html>
  <script language="vb" runat=server>
    sub Page_Load(o as object,e as eventargs)
      Application.Lock()
      Application("counter") = CType(Application("counter") + 1, Int32)
      Application.UnLock()
    end sub
  </script>

  <head>
    <title>
      Application 对象方法试验
    </title>
  </head>

  <body>
    <center>
      <h2>Application 对象 Lock 方法试验</h2>
      <hr>
```

```
你是第<%=Application("counter")%>位访问者！  
</center>  
</html>
```

效果图：



关于计数器非同步更新的问题

假设有两个上网者同时启动网页（这种情况很常见），同时执行了上面的步骤，他们读出的值是相同的，但结果都是相同的。很明显，访问量少了一次。如何处理呢？我们在这里举一个例子，假设我们都在使用一个数据库管理系统，有可能访问者同时对一张表进行操作，如果一人在修改这张表的数据，而另一个人在读数据，很明显此时数据前后就不一致。通过我们需要对表进行锁定。那么，我们可以锁定对象，程序修改为：

```
Application.lock  
Application("counter") = CType(Application("counter") + 1, Int32)  
Application.unlock
```

这样，当某一程序执行了 Application.lock 之后，其他应用程序就暂时不能使用 Application 对象，直到 Application unlock。

4.4.1.2 Application 的事件

当 Application 对象的生命周期开始时 Application_onstart 事件会被启动，当 Application 对象的生命周期结束时 Application_onend 事件会被启动。通常我们会在 global.asax 中定义 Application_onstart 事件。这和 ASP 程序相类似，有一点差别是 Application_onstart 事件是在 global.asa 中定义。还有就是除了以前的四个事件又增加了两个事件 Application_BeginRequest 事件和 Application_EndRequest 事件。

在上一个例子中我们实现了用计数器来对页面进行统计，但是这样的程序有这样的一个问题，就是只能统计单个的页面，我们在 asp+ 中可以很轻松的实现对整个站点页面的统计。为了达到这个目的，我们使用将使用 Application_BeginRequest 事件和 Application_EndRequest 事件。这两个事件在站点的任意一个文件被请求的时候都会被激发，因此我们便利用这个事件实现对站点的访问统计。

我们首先来看看这个 global.asax 文件

```
<script language="VB" runat="server">
Sub Application_End(Sender As Object, E As EventArgs)
我们捎带实现了 站点的当前在线人数
dim intOnlineNumber as integer
intOnlineNumber=cInt(Application("ONLINENUMBER"))-1
Application("ONLINENUMBER")=intOnlineNumber
End Sub

Sub Session_Start(Sender As Object, E As EventArgs)
Application.Lock
intOnlineNumber=cInt(Application("ONLINENUMBER"))+1
Application("ONLINENUMBER")=intOnlineNumber+1
Application.UnLock
End Sub

Sub Application_BeginRequest(Sender As Object, E As EventArgs)
response.write("当前访问的页面是 " + Request.FilePath + "<br>")
既然我们可以得到 FilePAth 则我们只要把这个参数进行详细的各种各样的统计就可以了
End Sub
</script>
```

好了一切完结之后，我们访问站点的任意一个 aspx 文件，都会在最上方发现一行文字：当前访问的页面是

怎么样，还不赶快尝试一下??

4.4.2 Session

4.4.2.1 Asp.net 里的 Session

对 Session 的应用，我们通过在 Global.asax 中设置，然后在 aspx 页面中调用来进行。下面我们就一个例子来说明这个问题。

4.4.2.2 一个 Session 例子：

我们在 Global.asax 中加上一个 Session_Start 方法，在里面定义响应的属性：

```
Sub Session_Start(Sender As Object, E As EventArgs)
```

```
        Session("name") = "saidy"  
        Session("email") = "saidychan@sina.com"  
        Session("tel") = "130000121553"  
    End Sub
```

赋予属性一个初始值，这样当我们在调用页面（ session.aspx ）装入时直接就用这些默认值：

```
<script language="VB" runat="server">  
    返回 Session 值方法：  
    Function getinfo(Key As String) As String  
        Return Session(Key).ToString()  
    End Function  
</script>
```

注意 Key 的定义，我们用这个语句获得方法的具体返回值,如 “ name “ 值：

```
<%=getinfo("name")%>
```

在另外一个文件（ info.aspx ）中，我们用下面的语句来获得数据：

```
Sub sc(Sender As Object, E As EventArgs)  
    Session("name") = name.Value  
    Session("email") = email.Value  
    Session("tel") = tel.Value  
    Response.Redirect(State("Referer").ToString())  
End Sub
```

保存在 Session 中。

下面是我们具体的代码：

application\Global.asax 文件：

```
<script language="VB" runat="server">  
  
    Sub Session_Start(Sender As Object, E As EventArgs)  
  
        Session("name") = "global_saidy"  
        Session("email") = "global_saidychan@sina.com"  
        Session("tel") = "global_130000121553"  
    End Sub  
  
</script>
```

application\session.aspx 文件：

```

<html>
<script language="VB" runat="server">
    返回 Session 值方法 :
    Function getinfo(Key As String) As String
        Return Session(Key).ToString()
    End Function
</script>

<body >
    <br><br><br>
    <center>
        <h3><font face="Verdana">.NET->Session</font></h3>
    </center>
    <br><br>
    <center>
    <b><a href="info.aspx">员工档案 ! </a></b><p>
        <div align="center">
    <center>
    <table border="0" width="35%" cellspacing="0" cellpadding="0">
        <tr>
            <td width="50%">名称 : </td>
            <td width="50%"><%=getinfo("name")%></td>
        </tr>
        <tr>
            <td width="50%">邮箱 : </td>
            <td width="50%"><%=getinfo("email")%></td>
        </tr>
        <tr>
            <td width="50%">电话 : </td>
            <td width="50%"><%=getinfo("tel")%></td>
        </tr>
    </table>
    </center>
</div>
</center>
</body>
</html>

```

application\info.aspx 文件 :

```

<html>
<script language="VB" runat="server">

```

'取得上一页的信息

```
Sub Page_Load(Sender As Object, E As EventArgs)
```

```
    If Not (Page.IsPostBack)
```

```
        State("Referer") = Request.Headers("Referer")
```

```
    End If
```

```
End Sub
```

'按钮事件，把数据保存在 Session 中，并返回上一页

```
Sub sc(Sender As Object, E As EventArgs)
```

```
    Session("name") = name.Value
```

```
    Session("email") = email.Value
```

```
    Session("tel") = tel.Value
```

```
    Response.Redirect(State("Referer").ToString())
```

```
End Sub
```

'按钮事件，返回上一页

```
Sub cc(Sender As Object, E As EventArgs)
```

```
    Response.Redirect(State("Referer").ToString())
```

```
End Sub
```

```
</script>
```

```
<body >
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<center>
```

```
    <form runat="server">
```

```
    <h3><font face="Verdana">.NET->Session!</font></h3>
```

```
</center>
```

```
<br>
```

```
<br>
```

```
<center>
```

```
<b>选择你的信息:</b><p>
```

```
<table bgcolor="#ccccff">
```

```
<tr>
```

```
<td>名称:</td>
```

```
<td>
```

```
<select id="name" runat="server">
```

```
<option>saidy chan</option>
```

```
        <option>贺禧</option>
        <option>吕兵</option>
    </select>
</td>
</tr>

<tr>
<td>邮箱:</td>
<td>
    <select id="email" runat="server">
        <option>saidychan@sina.com</option>
        <option>chenyx@staff.yesky.com</option>
        <option>hexi1224@sina.com</option>
    </select>
</td>
</tr>

<tr>
<td>电话:</td>
<td>
    <select id="tel" runat="server">
        <option>023-77398202</option>
        <option>013011235488</option>
        <option>1398855566588</option>
    </select>
</td>
</tr>

</table>

<p>
<input type="submit" OnServerClick="cc" Value="取消" runat="server"/>
<input type="submit" OnServerClick="sc" Value="提交" runat="server"/>

</form>

</center>

</body>

</html>
```



我们看到页面把我们储存在 Global.asax 文件中的数据取出来了，我们点击“员工档案连接”：



点击“提交”按钮，我们看到下面的结果：



我们看到提交的结果是 Session 中保存的数据被替代了。

4.4.3 小结

利用几个例子讲述再网上非常有用的两个属性 Application 和 Session，这两个属性非常重要，凡是做个一些网站的读者都知道的。

第五章 安全访问控制

asp.net 中讨论安全性，首先要解决 2 个问题。那就是谁有权力进入系统？他进入系统以后能进行何种操作？在解决谁能进入系统的问题中，我们通常会维护一张允许进入系统的用户的名单，当用户要求进入的时候，我们判断他是否是合法用户。这样一来，问题就转化为如何有效地判别一个用户是否是系统的有效用户，我们称之为“验证”过程。一个常见的验证过程是，当我们进入某些系统时，被要求输入用户和口令。当用户进入以后，我们只允许他访问事先指定给他的资源，这一过程称为“授权”。只有通过授权检查后，用户才能够对相应资源进行操作。在 asp.net 环境中，asp.net 和 IIS 结合在一起为用户提供验证和授权服务。

Asp.net 的应用程序还可以根据进入用户的不同标识，执行相应不同的应用代码。这种方式被称为“角色扮演”(impersonation)。

在一些应用中，对于不同的用户所能够看到和执行的功是不尽相同的，这就要求 asp.net 提供“角色”(Role)和“用户”(User)的区分方法。

4.5.1 验证和授权(Authentication And Authorization)

asp.net 和 IIS 一起为用户提供验证服务,用户验证方式有 3 种,即基本验证方式(basic)、摘要验证方式(digest)、窗口验证方式(Windows)。同时 asp.net 支持微软的“护照”(passport)验证服务,它单方面提供签到服务和用户描述服务。

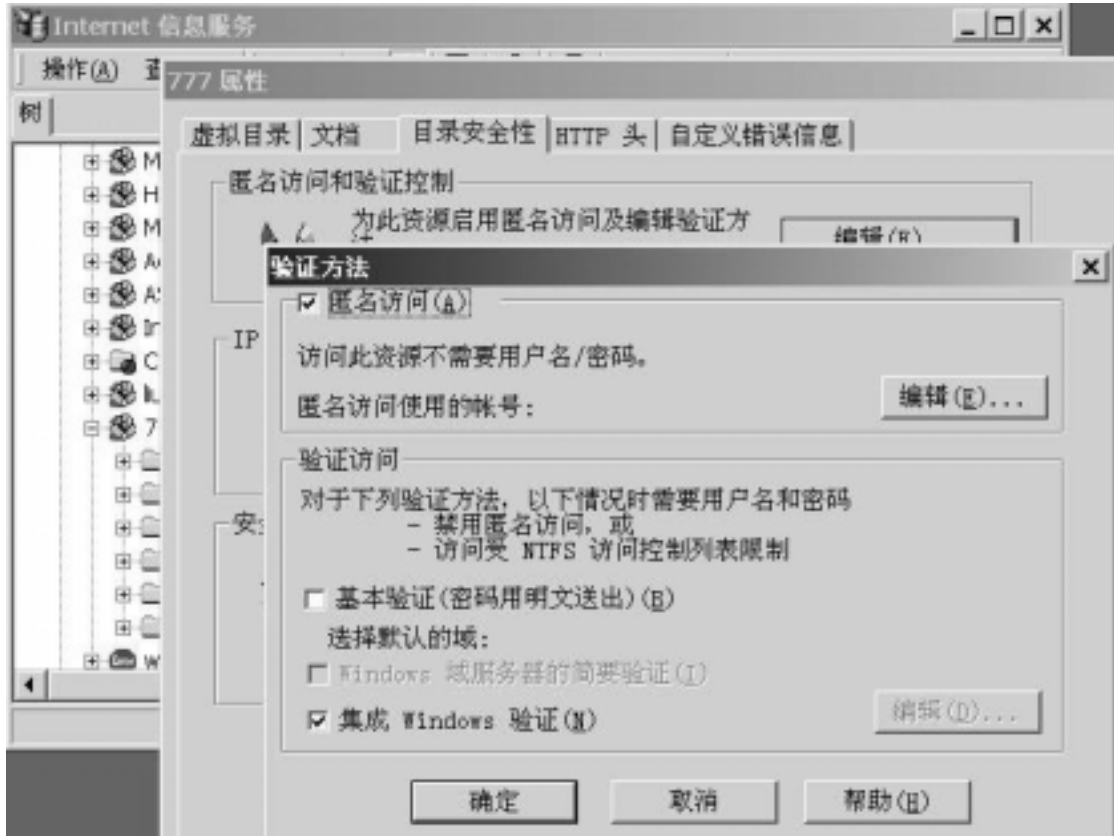
此外,asp.net 还提供了 cookies,帮助建立一种基于用户 Form 的验证方式,通过 cookies 用户的应用程序可以用自己的代码和逻辑实现用户定义的可信性验证。

由于 asp.net 的验证服务是建立在 IIS 的验证服务之上的,因此在设立自己的应用服务的时候有时需要在 IIS 中进行相应的设置。例如:为实现 asp.net 的基本验证服务,就必须利用 Internet Service Tool 工具把该应用设置成基本鉴别服务方式。

具体方法为:

选择“开始”->“程序”->“管理工具”->“Internet 信息服务”
打开“默认 Web 站点”,选择你的应用站点,比如:此处为“777”
右击鼠标,选择“属性”,再选择“目录安全性”标签
在“匿名访问和验证控制”框中,点击“编辑”按钮
然后在出现的“验证方法”窗口中,把“验证方法”框中,“基本验证”前面的选项打勾
最后确认,就完成了设置基本验证方法

操作的画面如下:



要使 Asp.Net 的验证机制生效，我们需要对应用的配置文件 config.web 进行设置。设置验证需要包含在<security>和</security>标识之间。它使用<authentication mode="...">语法形式，其中 mode 属性指定了验证采用的方式，具体的验证模式如下表格所示：

模式	描述
None	没有任何 Asp.Net 的验证服务被激活。
Windows	Asp.Net 和 Windows 的机制一起配合使用,可以授权限制 windows Nt 的用户和工作组的访问。
Cookie	Asp.Net 验证服务可以管理 cookies，可以引导使一个未经授权的用户去登录网页。这种模式经常和 IIS 一起配合，可以让一个匿名用户去访问应用程序。
Passport	Asp.Net 验证服务通过使用护照 (passport) 服务软件包，附属在服务的外层,更加强了验证服务,这种软件包必须安装到系统中才能使用。

举例说来,使用 Cookie 验证方式对某个应用进行验证,那么在其应用的配置文件 config.web 中,应该含有一下几行：

```
<!--config.web-->
...
<configuration>
```

```

...
<security>
...
    <authentication mode="Cookie"/>
...
</security>
...
</configuration>
...

```

asp.net 提供了两种授权方式：基于 ACL、资源权限的授权方式和 URL 授权。基于 ACL 和资源权限的授权方式有点类似于 Unix 下的文件权限检查，不过它更加严格和完备，当用户请求某个页面时，asp.net 检查该页面的 ACL(访问控制列表)和该文件的权限，看该用户是否有权限读取该页面。有，则该页面称作“已授权”。这种授权方式主要通过系统管理员对文件的权限的设定来实现。

而 URL 授权，对于某个用户的页面请求，并不是从文件权限出发，而是根据系统的配置情况，来决定用户的请求是否是经过授权的。URL 授权的方式的实现通常是通过设置应用配置文件 config.web 中关于授权和角色的配置节来实现的。

4.5.2 基于 WINDOWS 的验证

当采用基于 windows 方式的验证以后(<authentication mode="windows" />), asp.net 对每一个页面请求产生一个 WindowsPrincipal 对象 User，这个对象可以被 URL 授权方式用来进行授权，或者是被用户程序用来判断是否是某种角色。

例如，判断当前用户是否是管理员，可以采用以下的代码：

```

If User.IsInRole("Administrator")
    '如果是管理员组成员，...
...
Else
    '如果不是管理员组成员，...
...
End If

```

可能用户希望对验证过程加入一些自己的控制代码，那么就需要对 WindowsPrincipal 对象的 WindowsAuthentication_OnAuthenticate 事件作出自己的处理，编写自己的消息处理函数。通常用户通过实现 System.Security.Principal.Iprincipal 类来完成。

4.5.3 基于 FORM 的验证

基于 FORM 的验证，实际上是允许应用程序定义自己的验证画面和可信性验证。当用户进入时，出现事先定义的画面并请求输入验证要素，输入完毕，用用户逻辑对输入验证，通过，则进入应用，否则，返回起始输入画面。基于 FORM 的验证，通常都采用 cookies

技术来实现验证任务。启用基于 FORM 的验证，是在 config.web 中设置<authentication mode="Cookie" />来实现的。

例如：采用基于 Form 的验证方式，拒绝匿名用户进入的配置段如下：

```
<!--config.web-->
...
<configuration>
  <security>
    <authentication mode="Cookie"/>
    <authorization>
      <deny users="?" />
    </authorization>
  </security>
</configuration>
...
```

其中，<authorization>一节中，用 deny 标识表示禁止某种用户，“？”代表匿名用户，“*”代表所有用户，当然在 users 后也可以跟指定的用户，表示只拒绝指定的用户。

此外，在配置节中还有一个 cookie 标识，它规定了 Cookie 的行为，也较为重要。Cookie 标识有三个属性：

- **decryptionkey** 用于指定对 cookie 加解密的密钥。如果不指定或指定为 autogenerate，密钥将采用 Crypto API 产生的系统密钥。当指定为 autogenerate 时，产生的密钥和机器相关，因而不能跨机器和平台，除非显示给出密钥。
- **loginurl** 验证页面。当用户验证失败以后，被定向到的页面，它可以是在本地，也可以是其他机器上。
- **cookie** 指定用于验证任务的 cookie 的名字。由于在同一台机器上，可能存在多个应用，而同一应用使用一个 cookie 名，所以同一机器上的 cookie 名字不能相同。

例如，在 config.web 中有如下定义：

```
<!--config.web -->
...
<security>
  <authentication mode="Cookie">
    <cookie decryptionkey="autogenerate" loginurl="login.aspx"
      cookie=".ASPXCOOKIEDEMO" />
  </authentication>
</security>
...
```

从这段配置，我们可以知道 cookie 的密钥由系统产生，当验证失败以后，页面将跳转至 login.aspx，cook 名为 “.ASPXCOOKIEDEMO”

在基于 FORM 的验证中，一个常用到的对象是 CookAuthentication。

CookAuthentication 有 4 个比较重要的方法：

- **RedirectFromLoginPage** 方法，它通常在验证成功以后，从用户的验证画面返回用户

开始请求的页面。它带有两个参数，第一个为 cookie 要记录的用户名，第二个表明是否记录到永久性 cookie 中。

例如：

```
CookieAuthentication.RedirectFromLoginPage( " Chen",True)
```

表示把用户“ Chen”记录到硬盘上的 cookie 记录中 ,然后跳转到用户原先请求的页面上。

GetAuthCookie 从 cookie 中取得指定用户名的值。它的参数和 RedirectFromLoginPage 方法是一样的。

例如：

```
CookieAuthentication.GetAuthCookie( " Chen",Flase)
```

表示从当前连接的 cookie 中取出用户名为 chen 的值

· **SetAuthCookie** 把指定用户名存入 cookie 中。参数和 GetAuthCookie 方法一致

例如：

```
CookieAuthentication.SetAuthCookie( " Chen",Flase)
```

表示把用户 “ Chen”记录到连接过程中存在的 cookie 中

· **SignOut** 删除当前用户的验证 Cookie ，它不会理会 Cookie 到底是在内存，还是在硬盘上。

下面我将介绍一个完整的例子：

application\delault.aspx 文件：

```
<%@ Import Namespace="System.Web.Security " %>
```

```
<html>
```

```
<script language="VB" runat=server>
```

```
Sub Page_Load(Src As Object, E As EventArgs)
    Welcome.Text = "Hello, " + User.Identity.Name
End Sub
```

```
Sub Signout_Click(Src As Object, E As EventArgs)
    CookieAuthentication.SignOut()
    ‘删去 Cookies
    Response.Redirect("login.aspx")
    ‘返回验证画面
End Sub
```

```
</script>
```

```
<body>
```

```
<h3><font face="Verdana">使用 Cookie 验证方式</font></h3>
```

```
<form runat=server>
```

```
<h3><asp:label id="Welcome" runat=server/></h3>
```

```
<asp:button text="Signout" OnClick="Signout_Click" runat=server/>
```

```
</form>
```

```
</body>
```

```
</html>
```

application\login.aspx 文件 :

```
<% @ Import Namespace="System.Web.Security " %>
```

```
<html>
```

```
<script language="VB" runat=server>
```

```
Sub Login_Click(Src As Object, E As EventArgs)
```

```
    If (UserEmail.Value = "test@263.net" Or UserEmail.Value = "try@163.net") And  
    UserPass.Value = "password"
```

```
        ‘如果用户为 test@263.net 或者是 try@163.net 而且口令为 password,就认为通过验证
```

```
            CookieAuthentication.RedirectFromLoginPage(UserEmail.Value,  
PersistCookie.Checked)
```

```
            ‘记录 Cookie 并且跳转到请求的页面
```

```
            Else
```

```
                Msg.Text = "非法用户或口令: 请重试"
```

```
            End If
```

```
End Sub
```

```
</script>
```

```
<body>
```

```
<form runat=server>
```

```
<h3><font face="Verdana">登录窗口</font></h3>
```

```
<table>
```

```

<tr>
  <td>Email 地址:</td>
  <td><input id="UserEmail" type="text" runat=server/></td>
  <td><ASP:RequiredFieldValidator ControlToValidate="UserEmail" Display="Static"
ErrorMessage="*" runat=server/></td>
</tr>
<tr>
  <td>口令:</td>
  <td><input id="UserPass" type=password runat=server/></td>
  <td><ASP:RequiredFieldValidator ControlToValidate="UserPass" Display="Static"
ErrorMessage="*" runat=server/></td>
</tr>
<tr>
  <td>永久性 Cookie:</td>
  <td><ASP:CheckBox id=PersistCookie runat="server" /> </td>
  <td></td>
</tr>
</table>

```

```
<asp:button text="Login" OnClick="Login_Click" runat=server/>
```

```
<p>
```

```
<asp:Label id="Msg" ForeColor="red" Font-Name="Verdana" Font-Size="10"
runat=server />
```

```
</form>
```

```
</body>
```

```
</html>
```

config.web 文件 :

```
<configuration>
```

```
<security>
```

```
<authentication mode="Cookie">
```

```
<!--采用 cookie 验证方式-->
```

```
<cookie decryptionkey = "autogenerate" loginurl = "login.aspx" cookie =
".ASPXUSERDEMO" />
```

```
<!--密钥系统产生，登录页面为 login.aspx,记录的 Cookie 名为.aspxuserdemo-->
```

```
</authentication>
```

```
<authorization>
```



```

    <deny users="test@263.net" />
    <!-- 虽然程序中有两个用户 test@263 和 try@163.net 可以进入，但此处封掉
test@263.net,观察 deny 标识是否有效-->
    <deny users="?" />
    <!--拒绝匿名用户 anonymous-->
</authorization>
</security>
<globalization requestencoding="UTF-8" responseencoding="UTF-8" />
</configuration>

```

我们还可以在 config.web 中设置用户和密码，当用户在验证窗口输入用户和密码后，将其带入 CookieAuthenticationManager.Authenticate，有它来验证是否是合法用户。

在 config.web 中设置用户和密码使用 credentials 标识，它有一个属性 passwordformat，决定口令存储的方式。Passwordformat 可以为以下值：

- **Clear** 口令以纯文本方式存储。
- **SHA1** 口令以 SHA1 方式存储。
- **MD5** 口令以 MD5 方式存储。

例如：在一个配置文件中，

```

<!--config.web-->
...
<configuration>
...
<security>
    <authentication>
        <credentials passwordformat= " SHA1 " >
            <user name="Li" password= " GASDFSA9823598ASDBAD " />
            <user name="Chen" password= " ZASDFADSFASD23483142 " />
        </credentials>
    </authentication>
</security>
...
</configuration>

```

从这个例子，我们看到 credentials 标识中的 passwordformat 为 SHA1，所以下面的用户的密码是以 SHA1 方式加密以后的格式。在<credentials>和</credentials>之间我们定义了两个用户“Chen”和“Li”，其密码不可识别。

4.5.4 授权用户和角色

URL 授权控制对资源的访问权限。它可以使一些用户和角色对资源有存取权限，也可以拒绝某些用户和角色对资源的存取。甚至它还可以决定能够存取资源的 HTTP 方法(例如：不允许 get，允许 POST 等等)。

对于授权用户和角色的控制，asp.net 通过配置文件 config.web 中的<authorization>标识

段来实现。<allow>标识表示允许对资源的访问，<deny>标识表示拒绝对资源的访问。

它们都有两个属性，users 和 roles 分别表示用户和角色。

我们来看一个实例：

```
<!--config.web-->
...
<configuration>
<security>
<authorization>
    <allow users="nobody@163.net" />
    <allow roles="Admins" />
    <deny users="*" />
</authorization>
</security>
</configuration>
...
```

它表明了这样一个事实，用户“ nobody@163.net“ 和角色 Admins 有访问本站点的权力，其他用户对本站点的访问将被拒绝。也就是用户 nobody@163.net 和角色 Admins 分别是授权用户和授权角色。

同样，我们可以定义多个用户或者角色被授权或禁止，它们之间以“，”分隔。

例如：

```
...
<allow users="Chen,Li,Wang" />
<deny roles="Admins,Everyone" />
...
```

表示用户“ Chen”、“ Li”、“ Wang”是授权用户，但是角色为“ Admins”或者是“ Everyone”的被排除在外。

它的效果和分开写是一样的，如上例也可以写为：

```
...
<allow users="Chen" />
<allow users="Li" />
<allow users="Wang" />
<deny roles="Admins" />
<deny roles="Everyone" />
...
```

此外，还可以决定用户的某种 HTTP 方法是否可以被允许，方法是使用 verb 属性来表明对那种 HTTP 方法操作。

例如：

```
...
    <allow verb=post users="Chen,Li" />
    <deny verb=get roles="everyone" />
...
```

表示允许用户“ Chen”，“ Li”采用 post 方法访问资源，而拒绝角色 everyone 的 get 方式

对资源的访问。

符号“*”和“?”在<allow>和<deny>标识中有特殊的含义。“*”表示任何用户，“?”表示匿名用户。

例如：

```
...
<authorization>
  <allow users="*" />
  <deny users="?" />
</authorization>
```

表示除了匿名用户以外的所有用户都被允许访问本站点。

由前面的学习我们知道，在 asp.net 中应用是树型分层的，所以其配置文件也是呈层次结构，这也就导致了用户和角色的授权不是单一的结果，它要取决于沿树型结构上所有配置文件指定的结果的合集，而且越接近叶节点的配置越是有效。

例如：访问 <http://www.my.com/MyApp/a.aspx>

在 <http://www.my.com> 的根目录下的配置文件 config.web 有如下内容：

```
...
<security>
  <authorization>
    <allow users="*" />
  </authorization>
</security>
```

而在 MyApp 目录下有配置文件 config.web 内容如下：

```
...
<security>
  <authorization>
    <allow users="Chen" />
    <deny users="*" />
  </authorization>
</security>
```

那么，授权用户的集合到底是怎样的呢？asp.net 首先取得站点根目录下的配置，即所有用户都被允许访问，然后 asp.net 进入 Myapp 的目录取得其下的配置，除用户“Chen”以外所有的用户被拒绝，最后合并两个授权集合，并且如果两者之间有冲突，以后者为准。所以最终的授权集合为用户“Chen”，其他用户被拒绝。

第六章 会员系统例子：

4.6.1 数据库：

我们建立两个表 City 和 Customer，在建立几个存储过程来操作数据库（member.sql）：

```
CREATE TABLE [dbo].[City] (
  [ID] [int] NOT NULL ,
  [Name] [nvarchar] (20) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[customer] (
  [ID] [int] IDENTITY (1, 1) NOT NULL ,
  [Name] [nvarchar] (30) NULL ,
  [sex] [bit] NOT NULL ,
  [Birth] [smalldatetime] NULL ,
  [City] [int] NULL ,
  [Zip] [nvarchar] (6) NULL ,
  [Address] [nvarchar] (50) NULL ,
  [Telephone] [nvarchar] (30) NULL ,
  [PIN] [nvarchar] (30) NULL ,
  [Password] [nvarchar] (30) NULL ,
  [EMail] [nvarchar] (30) NULL ,
  [Question] [nvarchar] (50) NULL ,
  [Answer] [nvarchar] (50) NULL
) ON [PRIMARY]
GO

SET QUOTED_IDENTIFIER OFF      SET ANSI_NULLS ON
GO

CREATE PROCEDURE [GetAllCities] AS
  Select [ID],[Name] From City
GO

SET QUOTED_IDENTIFIER OFF      SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF      SET ANSI_NULLS ON
GO

CREATE PROCEDURE [GetDetailByID]
@ID Int
```

```
AS
Select PIN,Password,[Name],Birth,Sex,City,Zip,EMail,Telephone,Address,Question,Answer,[ID]
From Customer
Where [ID]=@ID
```

```
GO
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
CREATE PROCEDURE [GetDetailByPIN]
@PIN NVarChar(30)
AS
Select PIN,Password,[Name],Birth,Sex,City,Zip,EMail,Telephone,Address,Question,Answer,[ID]
From Customer
Where [PIN]=@PIN
```

```
GO
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON SET ANSI_NULLS ON
GO
```

```
CREATE PROCEDURE [IsValidUser]
@PIN NVarChar(30),
@Password NVarChar(30),
@ID Int Output
AS
Select @ID=[ID] From Customer Where PIN=@PIN and Password=@Password
```

```
GO
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
CREATE PROCEDURE [Modify]
```

```
@ID Int,
```

```
@PIN NVarChar(30),
```

```
@Password NVarChar(30),
```

```
@Name NVarChar(30),
```

```
@Birth DateTime,
```

```
@Sex bit,
```

```
@City int,
```

```
@Zip NVarChar(6),
```

```
@EMail NVarChar(30),
```

```
@Telephone NVarChar(30),
```

```
@Address NVarChar(30),
```

```
@Question NVarChar(50),
```

```
@Answer NVarChar(50)
```

```
AS
```

```
Update Customer
```

```
Set
```

```
Password=@Password,[Name]=@Name,Birth=@Birth,Sex=@Sex,City=@City,Zip=@Zip,EMail
```

```
=@EMail,Telephone=@Telephone,Address=@Address,Question=@Question,Answer=@Answer
```

```
Where ID=@ID And PIN=@PIN
```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON SET ANSI_NULLS ON
```

```
GO
```

```
CREATE PROCEDURE [PromptPassword]
```

```
@PIN nvarchar(30),
```

```
@Answer nvarchar(50),
```

```
@Password nvarchar(30) output
```

```
AS
```

```
select @Password=Password From Customer Where PIN=@PIN And Answer=@Answer
```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
```

GO

CREATE PROCEDURE [PromptQuestion]

@PIN nvarchar(30),

@Question nvarchar(50) output

AS

select @Question=Question From Customer Where PIN=@PIN

GO

SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON

GO

CREATE PROCEDURE [Register]

@ID Int OUTPUT,

@PIN NVarChar(30),

@Password NVarChar(30),

@Name NVarChar(30),

@Birth DateTime,

@Sex bit,

@City int,

@Zip NVarChar(6),

@EMail NVarChar(30),

@Telephone NVarChar(30),

@Address NVarChar(30),

@Question NVarChar(50),

@Answer NVarChar(50)

AS

Insert Into Customer

(PIN,Password,[Name],Birth,Sex,City,Zip,EMail,Telephone,Address,Question,Answer)

Values

(@PIN,@Password,@Name,@Birth,@Sex,@City,@Zip,@EMail,@Telephone,@Address,@Question,@Answer)

SELECT @ID = @@IDENTITY

GO

SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON

GO

4.6.2 登录页面：

判断用户的 Pin 和密码是否正确，并且是注册和密码提示的入口：

(csbook\appsoft\member\login.aspx)

```
<%@ Page EnableSessionState="False" MaintainState="false" Debug="True"%>
<%@ Import Namespace="DarkMan" %>

<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        RegisterLink.NavigateUrl="register.aspx?ReturnUrl="+Request.Params("ReturnUrl")

        PromptPasswordLink.NavigateUrl="prompt.aspx?ReturnUrl="+Request.Params("ReturnUrl")
    )
    End Sub

    Sub OnLogin(sender As Object, e As EventArgs)
        Dim theDB As New MemberDB

        Try
            Dim theID As Integer = theDB.IsValidUser(CStr(pin.Text),CStr(password1.Text))
            CookieAuthentication.SetAuthCookie(theID, false)
            Page.Navigate("modify.aspx")
        Catch e1 As Exception
            Prompt.Text="登录失败！"
        End Try
    End Sub
</script>

<HTML>
<BODY>
<DIV Align="Center">
<H1>会员登录</H1>
<hr wifth=600>
</DIV>

<FORM RunAt="Server">
<TABLE WIDTH=600 BORDER=0 CELLSPACING=1 CELLPADDING=1 Align="Center">
    <TR>
        <TD COLSPAN="2">
```



```

        <asp:RegularExpressionValidator id="RegPin" runat="server"
            ControlToValidate="pin"
            ValidationExpression="^[S^]{6,30}$"
            Display="Dynamic">
            帐号中：某些特殊字符禁用！另外，必须至少 6 个字符！
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>
    <TD COLSPAN="2">
        <asp:RegularExpressionValidator id="RegPassword" runat="server"
            ControlToValidate="password1"
            ValidationExpression="^[S^]{6,30}$"
            Display="Dynamic">
            密码中：某些特殊字符禁用！另外，必须至少 6 个字符！
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>
    <TD>帐号</TD>
    <TD><asp:textbox id="pin" RunAt="server"/>
        <asp:RequiredFieldValidator id="RFpin" runat="server"
            ControlToValidate="pin"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
</TR>
<TR>
    <TD>密码</TD>
    <TD><asp:textbox id="password1" RunAt="server" TextMode="Password"/>
        <asp:RequiredFieldValidator id="RFpass" runat="server"
            ControlToValidate="password1"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
</TR>
<TR>
    <TD COLSPAN="2"><asp:label id="Prompt" RunAt="Server"/>

```

```

        </TD>
    </TR>
    <TR>
        <TD COLSPAN="2">
            <input type="submit" OnServerClick="OnLogin" Value=" 马上登录 "
runat="server"/>
            <asp:Hyperlink id="RegisterLink" runat="server"
                Text="我要注册"
            />
            <asp:Hyperlink id="PromptPasswordLink" runat="server"
                Text="密码提示"
            />
        </TD>
    </TR>
</TABLE>
</FORM>

</BODY>
</HTML>

```

4.6.3 注册页面：

(csbook\appsoft\member\register.aspx)：

```

<%@ Page EnableSessionState="False" MaintainState="false" Debug="True"%>
<%@ Import Namespace="DarkMan" %>

<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not Page.IsPostBack Then
            Dim result As SqlDataReaderResult
            Dim obj As New City
            result = obj.GetAll()

            city.DataSource=result.dr
            city.DataBind

            result.Close
        End If
    End Sub

    Sub OnRegister(sender As Object, e As EventArgs)

```

```

If Page.IsValid Then
    Dim theUser As New MemberDetail
    theUser.PIN = pin.Text
    theUser.Password = password1.Text
    theUser.Name = name.Text
    theUser.Birth = DateTime.Parse(birth.Text)
    theUser.Sex = sex.SelectedItem.Value
    If city.SelectedItem <> DBNull Then
        theUser.City = city.SelectedItem.Value
    End If
    theUser.Zip = zip.Text
    theUser.EMail = email.Text
    theUser.Telephone = telephone.Text
    theUser.Address = address.Text
    theUser.Question = question.Text
    theUser.Answer = answer.Text

    Try
        Dim theDB As New MemberDB
        Dim theID As Integer = theDB.Register(theUser)

        Page.Navigate( Request.Params("ReturnUrl") )
    Catch e1 As Exception
        ErrMsg.Text="注册失败"
    End Try
End If
End Sub

```

```
</script>
```

```
<HTML>
```

```
<BODY>
```

```
<DIV Align="Center">
```

```
<H1>会员注册</H1>
```

```
<hr wifth=600>
```

```
</DIV>
```

```
<FORM RunAt="Server">
```

```
<TABLE WIDTH=600 BORDER=0 CELLSPACING=1 CELLPADDING=1 Align="Center">
```

```
<TR>
```

```
<TD COLSPAN="4">
```

```
<asp:RegularExpressionValidator id="RegPin" runat="server"
```

```
ControlToValidate="pin"
```

```
ValidationExpression="^[S^]{6,30}$"
```

```

        Display="Dynamic">
        帐号中：某些特殊字符禁用！另外，必须至少 6 个字符！
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>
    <TD COLSPAN="4">
        <asp:RegularExpressionValidator id="RegName" runat="server"
            ControlToValidate="name"
            ValidationExpression="^[S^]{6,30}$"
            Display="Dynamic">
            姓名中：某些特殊字符禁用！另外，必须至少 6 个字符！
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>
    <TD COLSPAN="4">
        <asp:RegularExpressionValidator id="RegPassword" runat="server"
            ControlToValidate="password1"
            ValidationExpression="^[S^]{6,30}$"
            Display="Dynamic">
            密码中：某些特殊字符禁用！另外，必须至少 6 个字符！
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>
    <TD COLSPAN="4">
        <asp:RegularExpressionValidator id="RegBirth" runat="server"
            ControlToValidate="birth"
            ValidationExpression="^\d{4}-\d{1,2}-\d{1,2}$"
            Display="Dynamic">
            日期格式：2001-7-01
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>
    <TD COLSPAN="4">
        <asp:RegularExpressionValidator id="Regzip" runat="server"
            ControlToValidate="zip"

```

```

        ValidationExpression="\d{6}$"
        Display="Dynamic">
            邮政编码不对
        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>

<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="Regtel" runat="server"
        ControlToValidate="telephone"
        ValidationExpression="^[d-\(\)]+$"
        Display="Dynamic">
            电话号码不对
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>
<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="RegEMail" runat="server"
        ControlToValidate="email"
        ValidationExpression="^[w-_.]*[w-_.]@[w].+[w]+[w]$"
        Display="Dynamic">
            EMail 格式：xxx@222.com
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>
<TD COLSPAN="4">
    <asp:CompareValidator id="CompPassword12"
        ControlToValidate="password2"
        ControlToCompare = "password1"
        Display="Dynamic"
        Type="String" runat="server">
            密码校验不正确！
    </asp:CompareValidator>
</TD>
</TR>

<TR>
<TD>账号</TD>
<TD><asp:textbox id="pin" RunAt="server"/>

```

```

        <asp:RequiredFieldValidator id="RFpin" runat="server"
            ControlToValidate="pin"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
    <TD>姓名</TD>
    <TD><asp:textbox id="name" RunAt="server"/>
        <asp:RequiredFieldValidator id="RFname" runat="server"
            ControlToValidate="name"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
</TR>
<TR>
    <TD>密码</TD>
    <TD><asp:textbox id="password1" RunAt="server" TextMode="Password"/>
        <asp:RequiredFieldValidator id="RFpass" runat="server"
            ControlToValidate="password1"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
    <TD>密码校验</TD>
    <TD><asp:textbox id="password2" RunAt="server" TextMode="Password"/></TD>
</TR>
<TR>
    <TD>生日</TD>
    <TD><asp:textbox id="birth" RunAt="server"/></TD>
    <TD>性别</TD>
    <TD>
        <asp:RadioButtonList RepeatColumns="2" id="sex" runat="server">
            <asp:ListItem text="男" value="0" Selected/>
            <asp:ListItem text="女" value="1"/>
        </asp:RadioButtonList></TD>
</TR>
<TR>
    <TD>城市</TD>
    <TD>
        <asp:DropDownList id="city" runat="server"
            DataTextField="Name"
            DataValueField="ID"
        />
    </TD>

```

```

</TD>
<TD>邮政编码</TD>
<TD><asp:textbox id="zip" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFzip" runat="server"
        ControlToValidate="zip"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD>EMail</TD>
<TD><asp:textbox id="email" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFemail" runat="server"
        ControlToValidate="email"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
<TD>电话</TD>
<TD><asp:textbox id="telephone" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFtel" runat="server"
        ControlToValidate="telephone"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD COLSPAN="1">地址</TD>
<TD COLSPAN="3"><asp:textbox id="address" RunAt="server" Columns="50"/>
    <asp:RequiredFieldValidator id="RFAdress" runat="server"
        ControlToValidate="address"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD COLSPAN="1">密码提示问题</TD>
<TD COLSPAN="3"><asp:textbox id="question" RunAt="server" Columns="50"/>
    <asp:RequiredFieldValidator id="RFQ" runat="server"
        ControlToValidate="question"
        ErrorMessage="*"

```

```

        ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
</TR>
<TR>
    <TD COLSPAN="1">密码提示问题答案</TD>
    <TD COLSPAN="3"><asp:textbox id="answer" RunAt="server" Columns="50"/>
        <asp:RequiredFieldValidator id="RFA" runat="server"
            ControlToValidate="answer"
            ErrorMessage="*"
            ForeColor="Red">
        </asp:RequiredFieldValidator>
    </TD>
</TR>
<TR>
    <TD COLSPAN="4">
        <input type="submit" OnServerClick="OnRegister" Value="马上注册"
runat="server"/>
        <br>
        <font color="red"><asp:label id="ErrMsg" runat="server"/></font>
    </TD>
</TR>
</TABLE>
</FORM>

</BODY>
</HTML>

```

4.6.4 修改页面：

(csbook\appsoft\member\modify.aspx)：

```

<% @ Page EnableSessionState="False" MaintainState="false" Debug="True"%>
<% @ Import Namespace="DarkMan" %>

<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not Page.IsPostBack Then
            Dim theDB As New MemberDB
            Dim m_PIN As String=Request.Params("PIN")
            Dim m_ID As Integer=CInt( Request.Params("ID") )
            Dim theUser As MemberDetail

```



```

If m_PIN=DBNull And m_ID=DBNull Then
    CookieAuthentication.RedirectFromLoginPage(theID, false)
Else If m_PIN<>DBNull Then
    theUser=theDB.GetDetailByPIN(m_PIN)
    userid.Text = theUser.ID
    pin.Text=theUser.PIN
    password1.Text=theUser.Password
    name.Text=theUser.Name
    birth.Text=theUser.Birth
    zip.Text=theUser.Zip
    email.Text=theUser.EMail
    telephone.Text=theUser.Telephone
    address.Text=theUser.Address
    question.Text=theUser.Question
    answer.Text=theUser.Answer
Else
    theUser=theDB.GetDetailByID(m_ID)
    userid.Text = theUser.ID
    pin.Text=theUser.PIN
    password1.Text=theUser.Password
    name.Text=theUser.Name
    birth.Text=theUser.Birth
    zip.Text=theUser.Zip
    email.Text=theUser.EMail
    telephone.Text=theUser.Telephone
    address.Text=theUser.Address
    question.Text=theUser.Question
    answer.Text=theUser.Answer
End If

Dim result As SqlDataReaderResult
Dim obj As New City
result = obj.GetAll()

city.DataSource=result.dr
city.DataBind

result.Close

End If
End Sub

Sub OnModify(sender As Object, e As EventArgs)

```

```

If Page.IsValid Then
    Dim theUser As New MemberDetail
    theUser.ID = userid.Text
    theUser.PIN = pin.Text
    theUser.Password = password1.Text
    theUser.Name = name.Text
    theUser.Birth = DateTime.Parse(birth.Text)
    theUser.Sex = sex.SelectedItem.Value
    If city.SelectedItem <> DBNull Then
        theUser.City = city.SelectedItem.Value
    End If
    theUser.Zip = zip.Text
    theUser.EMail = email.Text
    theUser.Telephone = telephone.Text
    theUser.Address = address.Text
    theUser.Question = question.Text
    theUser.Answer = answer.Text

    Dim theDB As New MemberDB
    Dim theID As Integer = theDB.Modify(theUser)

    CookieAuthentication.RedirectFromLoginPage(theID, false)
    Page.Navigate("login.aspx")
End If
End Sub

```

```
</script>
```

```
<HTML>
```

```
<BODY>
```

```
<DIV Align="Center">
```

```
<H1>会员信息修改</H1>
```

```
<hr wifth=600>
```

```
</DIV>
```

```
<FORM RunAt="Server">
```

```
<TABLE WIDTH=600 BORDER=0 CELSPACING=1 CELLPADDING=1 Align="Center">
```

```
<TR>
```

```
<TD COLSPAN="4">
```

```
<asp:RegularExpressionValidator id="RegPin" runat="server"
```

```
ControlToValidate="pin"
```

```
ValidationExpression="^[S^]{6,30}$"
```

```
Display="Dynamic">
```

帐号中：某些特殊字符禁用！另外，必须至少 6 个字符！

```

        </asp:RegularExpressionValidator>
    </TD>
</TR>

<TR>

<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="RegName" runat="server"
        ControlToValidate="name"
        ValidationExpression="^[S^]{6,30}$"
        Display="Dynamic">
        姓名中：某些特殊字符禁用！另外，必须至少 6 个字符！
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>

<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="RegPassword" runat="server"
        ControlToValidate="password1"
        ValidationExpression="^[S^]{6,30}$"
        Display="Dynamic">
        密码中：某些特殊字符禁用！另外，必须至少 6 个字符！
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>

<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="RegBirth" runat="server"
        ControlToValidate="birth"
        ValidationExpression="^\d{4}-\d{1,2}-\d{1,2}$"
        Display="Dynamic">
        日期格式：2001-7-01
    </asp:RegularExpressionValidator>
</TD>
</TR>

<TR>

<TD COLSPAN="4">
    <asp:RegularExpressionValidator id="Regzip" runat="server"
        ControlToValidate="zip"
        ValidationExpression="^\d{6}$"
        Display="Dynamic">

```

邮政编码不对

```
</asp:RegularExpressionValidator>
```

```
</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD COLSPAN="4">
```

```
<asp:RegularExpressionValidator id="Regtel" runat="server"
```

```
ControlToValidate="telephone"
```

```
ValidationExpression="^[d-\(\)]+ $"
```

```
Display="Dynamic">
```

电话号码不对

```
</asp:RegularExpressionValidator>
```

```
</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD COLSPAN="4">
```

```
<asp:RegularExpressionValidator id="RegEMail" runat="server"
```

```
ControlToValidate="email"
```

```
ValidationExpression="^[w-_.]*[w-_.]@[w].+[w]+[w] $"
```

```
Display="Dynamic">
```

E Mail 格式 : xxx@222.com

```
</asp:RegularExpressionValidator>
```

```
</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD COLSPAN="4">
```

```
<asp:CompareValidator id="CompPassword12"
```

```
ControlToValidate="password2"
```

```
ControlToCompare = "password1"
```

```
Display="Dynamic"
```

```
Type="String" runat="server">
```

密码校验不正确 !

```
</asp:CompareValidator>
```

```
</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>账号</TD>
```

```
<TD><asp:textbox id="pin" RunAt="server"/>
```

```
<asp:RequiredFieldValidator id="RFpin" runat="server"
```

```
ControlToValidate="pin"
```

```

                ErrorMessage="*"
                ForeColor="Red">
            </asp:RequiredFieldValidator>
        </TD>
        <TD>姓名</TD>
        <TD><asp:textbox id="name" RunAt="server"/>
            <asp:RequiredFieldValidator id="RFname" runat="server"
                ControlToValidate="name"
                ErrorMessage="*"
                ForeColor="Red">
            </asp:RequiredFieldValidator>
        </TD>
    </TR>
    <TR>
        <TD>密码</TD>
        <TD><asp:textbox id="password1" RunAt="server" TextMode="Password"/>
            <asp:RequiredFieldValidator id="RFpass" runat="server"
                ControlToValidate="password1"
                ErrorMessage="*"
                ForeColor="Red">
            </asp:RequiredFieldValidator>
        </TD>
        <TD>密码校验</TD>
        <TD><asp:textbox id="password2" RunAt="server" TextMode="Password"/></TD>
    </TR>
    <TR>
        <TD>生日</TD>
        <TD><asp:textbox id="birth" RunAt="server"/></TD>
        <TD>性别</TD>
        <TD>
            <asp:RadioButtonList RepeatColumns="2" id="sex" runat="server">
                <asp:ListItem text="男" value="0" Selected/>
                <asp:ListItem text="女" value="1"/>
            </asp:RadioButtonList></TD>
    </TR>
    <TR>
        <TD>城市</TD>
        <TD>
            <asp:DropDownList id="city" runat="server"
                DataTextField="Name"
                DataValueField="ID"
            />
        </TD>
        <TD>邮政编码</TD>

```

```

<TD><asp:textbox id="zip" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFzip" runat="server"
        ControlToValidate="zip"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD>EMail</TD>
<TD><asp:textbox id="email" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFemail" runat="server"
        ControlToValidate="email"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
<TD>电话</TD>
<TD><asp:textbox id="telephone" RunAt="server"/>
    <asp:RequiredFieldValidator id="RFtel" runat="server"
        ControlToValidate="telephone"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD COLSPAN="1">地址</TD>
<TD COLSPAN="3"><asp:textbox id="address" RunAt="server" Columns="50"/>
    <asp:RequiredFieldValidator id="RFAddress" runat="server"
        ControlToValidate="address"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>
</TD>
</TR>
<TR>
<TD COLSPAN="1">密码提示问题</TD>
<TD COLSPAN="3"><asp:textbox id="question" RunAt="server" Columns="50"/>
    <asp:RequiredFieldValidator id="RFQ" runat="server"
        ControlToValidate="question"
        ErrorMessage="*"
        ForeColor="Red">
    </asp:RequiredFieldValidator>

```

```

        </TD>
    </TR>
    <TR>
        <TD COLSPAN="1">密码提示问题答案</TD>
        <TD COLSPAN="3"><asp:textbox id="answer" RunAt="server" Columns="50"/>
            <asp:RequiredFieldValidator id="RFA" runat="server"
                ControlToValidate="answer"
                ErrorMessage="*"
                ForeColor="Red">
            </asp:RequiredFieldValidator>
        </TD>
    </TR>
    <TR>
        <TD COLSPAN="4">
            <input type="submit" OnServerClick="OnModify" Value=" 马上提交 "
runat="server"/>
        </TD>
    </TR>
</TABLE>

<asp:textbox id="userid" RunAt="server" Columns="0" width="0" height="0"/>

</FORM>

</BODY>
</HTML>

```

4.6.5 密码提示页面：

当我们输入用户的 Pin 时，程序自动把用户的提问问题显示出来，如果没有这个用户，则提示没有这个用户（csbook\appsoft\member\prompt.aspx）：

```

<% @ Page EnableSessionState="False" MaintainState="false" Debug="True"%>
<% @ Import Namespace="DarkMan" %>

<script language="VB" runat="server">
    Sub OnPromptQuestion(sender As Object, e As EventArgs)
        If Page.IsValid Then
            Try
                Dim theDB As New MemberDB
                question.Text = theDB.PromptQuestion(pin.Text)
            Catch
            End Try
        End If
    End Sub

```

```

        Dim obj As New RequiredFieldValidator()

        obj.ControlToValidate="answer"
        obj.ErrorMessage = "需要输入密码问题答案！"
        Panel.Controls.Add(obj)
    Catch e1 As Exception
        password.Text = "不存在此账号！"
    End Try
End If
End Sub

Sub OnPromptPassword(sender As Object, e As EventArgs)
    m_Answer = answer.Text
    If Not Page.IsValid Or m_Answer=DBNull Then Exit Sub

    Dim theDB As New MemberDB
    Dim m_Password As String

    Try
        m_Password = theDB.PromptPassword(pin.Text,answer.Text)
        If m_Password <> DBNull Then
            password.Text = " 你的 密 码 是  : "+m_Password+"<a
href="+Request.Params("ReturnUrl")+">现在返回</A>"
        Else
            password.Text = "无法提示你的密码。可能你没有正确地回答问题！"
        End If
    Catch e1 As Exception
        password.Text = "无法提示你的密码。可能你没有正确地回答问题！"
        Exit Sub
    End Try

End Sub

```

```
</script>
```

```
<HTML>
```

```
<BODY>
```

```
<DIV Align="Center">
```

```
<H1>会员注册</H1>
```

```
<hr wifth=600>
```

```
</DIV>
```

```
<FORM RunAt="Server">
```

```
<TABLE WIDTH=600 BORDER=0 CELLSPACING=1 CELLPADDING=1 Align="Center">
```



```

<TR>
  <TD COLSPAN="4">
    <asp:RegularExpressionValidator id="RegPin" runat="server"
      ControlToValidate="pin"
      ValidationExpression="^[\\S^]{6,30}$"
      Display="Dynamic">
      帐号中：某些特殊字符禁用！另外，必须至少 6 个字符！
    </asp:RegularExpressionValidator>
  </TD>
</TR>

<TR>
  <TD>帐号</TD>
  <TD><asp:textbox id="pin" RunAt="server" OnTextChanged="OnPromptQuestion"
AutoPostBack="True"/>
    <asp:RequiredFieldValidator id="RFpin" runat="server"
      ControlToValidate="pin"
      ErrorMessage="*"
      ForeColor="Red">
    </asp:RequiredFieldValidator>
  </TD>

<TR>
  <TD COLSPAN="1">密码提示问题</TD>
  <TD COLSPAN="3"><asp:textbox id="question" RunAt="server" Columns="50"
ReadOnly/>
  </TD>
</TR>
<TR>
  <TD COLSPAN="1">密码提示问题答案</TD>
  <TD COLSPAN="3"><asp:textbox id="answer" RunAt="server" Columns="50"/>
    <asp:Panel id="Panel" runat="server"/>
  </TD>
</TR>
<TR>
  <TD COLSPAN="4"><font color='red'>
    <asp:label id="password" runat="server"/></font>
  </TD>
</TR>
<TR>
  <TD COLSPAN="4">
    <input type="submit" OnServerClick="OnPromptPassword" Value="请马上

```

```

提示" runat="server"/>
    </TD>
</TR>
</TABLE>
</FORM>

</BODY>
</HTML>

```

4.6.6 连接数据库组件方法：

通过这个组件，我们获得在配置文件 Config.web 中的数据库的连接串，
(csbook\appsoft\member\DBConn\DBConn.vb)：

```

Imports System
Imports System.Data
Imports System.Data.SQL
Imports System.Web
Imports System.Collections

Namespace DarkMan

    Public Class DBConn
        Shared m_ConnectionString As String

        Shared ReadOnly Property ConnectionString As String

        Get
            If m_ConnectionString = "" Then

                Dim appsetting As Hashtable =
                CType(HttpContext.Current.GetConfig("appsettings"), Hashtable)
                m_ConnectionString = CStr(appsetting("DSN"))

                If m_ConnectionString = "" Then
                    throw new Exception("iShop DSN Value not set in Config.web")
                End if

            End If

            return m_connectionString
        End Get
    End Class

```

```

        End Get

    End Property

End Class

Public Class SqlDataReaderResult

    Public conn As SqlConnection
    Public dr As SqlDataReader

    Public Sub Close()
        If conn.State = DBObjectState.Open then
            Try
                dr.Close()
                conn.Close()
            Catch e As Exception
                throw e
            End Try
        End If
    End Sub

End Class

End Namespace

```

4.6.7 操作组件方法：

这个组件里封装了对数据的插入、更新、密码提示方法、用户有效性的判断等方法。
csbook\appsoft\member\com\member.vb:

```

Imports System
Imports System.Data
Imports System.Data.SQL

Namespace DarkMan

    Public Class MemberDetail
        Public ID As Integer
        Public PIN As String
        Public Password As String
    End Class
End Namespace

```

```

Public Name As String
Public Birth As DateTime
Public Sex As Boolean
Public City As Integer
Public Zip As String
Public EMail As String
Public Telephone As String
Public Address As String
Public Question As String
Public Answer As String
End Class

```

```

Public Class City
    Public Shared Function GetAll() As SqlDataReaderResult
        Dim myConnection As SqlConnection = new
SqlConnection(DarkMan.DBConn.ConnectionString)
        Dim myCommand As SqlCommand=New SqlCommand("GetAllCities",
myConnection)

        myCommand.CommandType = CommandType.StoredProcedure

        Dim result As New SqlDataReaderResult
        result.Conn = myConnection

        Try
            myConnection.Open()
            myCommand.Execute(result.dr)
        Catch e As Exception
            throw e
        End Try

        return result
    End Function
End Class

```

```

Public Class MemberDB

    '登记信息
    Public Function Register(ByRef user As MemberDetail) As Integer
        Return RegNMod(0,user)
    End Function

    '密码提示
    Public Function PromptPassword(ByVal PIN As String,ByVal Answer As String) As String

```

```
Dim myConnection As SqlConnection = new
SqlConnection(DarkMan.DBConn.ConnectionString)
```

```
Dim myCommand As SqlCommand= new SqlCommand("PromptPassword",
myConnection)
```

```
myCommand.CommandType = CommandType.StoredProcedure
```

```
Dim param As SqlParameter
```

```
Param = new SqlParameter("@PIN", SqlDbType.NVarChar,30)
```

```
Param.Value = PIN
```

```
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Answer", SqlDbType.NVarChar,50)
```

```
Param.Value = Answer
```

```
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Password", SqlDbType.NVarChar,30)
```

```
Param.Direction = ParameterDirection.Output
```

```
myCommand.Parameters.Add(Param)
```

```
Try
```

```
myConnection.Open()
```

```
myCommand.Execute()
```

```
Catch e As Exception
```

```
throw e
```

```
Finally
```

```
If myConnection.State = DBObjectState.Open then
```

```
myConnection.Close()
```

```
End If
```

```
End Try
```

```
return CStr( Param.Value )
```

```
End Function
```

密码问题提示

```
Public Function PromptQuestion(ByVal PIN As String) As String
```

```
Dim myConnection As SqlConnection = new
SqlConnection(DarkMan.DBConn.ConnectionString)
```

```
Dim myCommand As SqlCommand= new SqlCommand("PromptQuestion",
myConnection)
```

```
myCommand.CommandType = CommandType.StoredProcedure
```

```
Dim param As SqlParameter
```

```
Param = new SqlParameter("@PIN", SqlDbType.NVarChar,30)
Param.Value = PIN
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Question", SqlDbType.NVarChar,50)
Param.Direction = ParameterDirection.Output
myCommand.Parameters.Add(Param)
```

```
Try
    myConnection.Open()
    myCommand.Execute()
Catch e As Exception
    throw e
Finally
    If myConnection.State = DBObjectState.Open then
        myConnection.Close()
    End If
End Try

return CStr( Param.Value )
```

End Function

'修改信息

```
Public Function Modify(ByRef user As MemberDetail) As Integer
    Return RegNMod(1,user)
End Function
```

```
Public Function GetDetailByID(ByVal ID As Integer) As MemberDetail
    return GetDetail(0,ID)
End Function
```

```
Public Function GetDetailByPIN(ByVal PIN As String) As MemberDetail
    return GetDetail(1, PIN)
End Function
```

'判断是否为合法用户

```
Public Function IsValidUser(ByVal PIN As String,ByVal Password As String) As Integer
    Dim myConnection As SqlConnection = new
SqlConnection(DarkMan.DBConn.ConnectionString)
```

```
Dim myCommand As SqlCommand= new SqlCommand("IsValidUser",
myConnection)
```

```
myCommand.CommandType = CommandType.StoredProcedure
Dim param As SqlParameter
```

```
Param = new SqlParameter("@PIN", SqlDbType.NVarChar,30)
Param.Value = PIN
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Password", SqlDbType.NVarChar,30)
Param.Value = Password
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@ID", SqlDbType.Int)
Param.Direction = ParameterDirection.Output
myCommand.Parameters.Add(Param)
```

```
Try
    myConnection.Open()
    myCommand.Execute()
Catch e As Exception
    throw e
Finally
    If myConnection.State = DBObjectState.Open then
        myConnection.Close()
    End If
End Try
```

```
return CInt( Param.Value )
```

```
End Function
```

```
Protected Function GetDetail(ByVal op As Integer,Optional ByVal ID As Integer=0,Optional
ByVal PIN As String="") As MemberDetail
```

```
Dim theUser As MemberDetail=New MemberDetail
```

```
Dim myConnection As SqlConnection = new
SqlConnection(DarkMan.DBConn.ConnectionString)
```

```
Dim myCommand As SqlCommand
```

```
Select Case op
```

```
Case 0 'By ID
```

```
myCommand = new SqlCommand("GetDetailByID", myConnection)
```

```
Dim param As SqlParameter
```

```

        Param = new SqlParameter("@ID", SqlDbType.Int)
        Param.Value = ID
        myCommand.Parameters.Add(Param)
    Case 1 'By PIN
        myCommand = new SqlCommand("GetDetailByPIN", myConnection)
        Dim param As SqlParameter
        Param = new SqlParameter("@PIN", SqlDbType.Int)
        Param.Value = PIN
        myCommand.Parameters.Add(Param)
    Case Else
        Exit Function
End Select

myCommand.CommandType = CommandType.StoredProcedure

Dim dr As SqlDataReader

Try
    myConnection.Open()
    myCommand.Execute(dr)
    If dr.Read Then
        theUser.PIN = dr.GetString(0)          'dr.GetValue(0,theUser.PIN)
        theUser.Password = dr.GetString(1)    'dr.GetValue(1,theUser.Password)
        theUser.Name = dr.GetString(2)       'dr.GetValue(2,theUser.Name)
        theUser.Birth = dr.GetDateTime(3)    'dr.GetValue(3,theUser.Birth)
        theUser.Sex = dr.GetSQLBit(4).BoolValue 'dr.GetValue(4,theUser.Sex)
        theUser.City = dr.GetInt32(5) 'dr.GetValue(5,theUser.City)
        theUser.Zip = dr.GetString(6) 'dr.GetValue(6,theUser.Zip)
        theUser.EMail = dr.GetString(7)     'dr.GetValue(7,theUser.EMail)
        theUser.Telephone = dr.GetString(8)  'dr.GetValue(8,theUser.Telephone)
        theUser.Address = dr.GetString(9) 'dr.GetValue(9,theUser.Address)
        theUser.Question = dr.GetString(10)  'dr.GetValue(10,theUser.Question)
        theUser.Answer = dr.GetString(11)    'dr.GetValue(11,theUser.Answer)
        theUser.ID = dr.GetInt32(12)         'dr.GetValue(0,theUser.ID)
    End If
Catch e As Exception
    throw e
Finally
    If myConnection.State = DBObjectState.Open then
        myConnection.Close()
    End If
End Try

Return theUser

```


End Function

```
Protected Function RegNMod( ByVal op As Integer,ByRef user As MemberDetail) As Integer
```

```
    Dim myConnection As SqlConnection = new SqlConnection(DarkMan.DBConn.ConnectionString)
```

```
    Dim myCommand As SqlCommand
```

```
    Dim ParamID As New SqlParameter("@ID", SqlDbType.Int)
```

```
    Select Case op
```

```
        Case 0 'Insert
```

```
            myCommand = new SqlCommand("Register", myConnection)
```

```
            ParamID.Direction = ParameterDirection.Output
```

```
            myCommand.Parameters.Add(ParamID)
```

```
        Case 1 'Modify
```

```
            myCommand = new SqlCommand("Modify", myConnection)
```

```
            ParamID.Value = user.ID
```

```
            myCommand.Parameters.Add(ParamID)
```

```
        Case Else
```

```
            Exit Function
```

```
    End Select
```

```
    myCommand.CommandType = CommandType.StoredProcedure
```

```
    Dim param As SqlParameter
```

```
    Param = new SqlParameter("@PIN", SqlDbType.NVarChar, 30)
```

```
    Param.Value = user.PIN
```

```
    myCommand.Parameters.Add(Param)
```

```
    Param = new SqlParameter("@Password", SqlDbType.NVarChar, 30)
```

```
    Param.Value = user.Password
```

```
    myCommand.Parameters.Add(Param)
```

```
    Param = new SqlParameter("@Name", SqlDbType.NVarChar, 30)
```

```
    Param.Value = user.Name
```

```
    myCommand.Parameters.Add(Param)
```

```
    Param = new SqlParameter("@Birth", SqlDbType.SmallDateTime)
```

```
    Param.Value = user.Birth
```

```
    myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Sex", SqlDbType.Bit)
Param.Value = user.Sex
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@City", SqlDbType.Int)
Param.Value = user.City
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Zip", SqlDbType.NVarChar, 6)
Param.Value = user.Zip
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@EMail", SqlDbType.NVarChar, 30)
Param.Value = user.EMail
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Telephone", SqlDbType.NVarChar, 30)
Param.Value = user.Telephone
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Address", SqlDbType.NVarChar, 50)
Param.Value = user.Address
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Question", SqlDbType.NVarChar, 50)
Param.Value = user.Question
myCommand.Parameters.Add(Param)
```

```
Param = new SqlParameter("@Answer", SqlDbType.NVarChar, 50)
Param.Value = user.Answer
myCommand.Parameters.Add(Param)
```

```
Try
    myConnection.Open()
    myCommand.Execute()
Catch e As Exception
    throw e
Finally
    If myConnection.State = DBObjectState.Open then
        myConnection.Close()
    End If
End Try
```

```
    If op=0 Then
        return CInt(ParamID.Value)
    Else
        return user.ID
    End If
```

```
End Function
```

```
End Class
```

```
End Namespace
```

4.6.8 配置文件：

我们把数据库的连接字符串方法在配置文件 Config.web 里面：
(csbook\appsoft\member\Config.web)：

```
<configuration>
  <!--数据库连接字符串 -->
  <appsettings>
    <add key="DSN" value="server=localhost;uid=sa;pwd=;database=darkman" />
  </appsettings>

  <globalization requestencoding="gb2312" responseencoding="gb2312" />
</configuration>
```

4.6.9 运行效果：

登录页面：



密码提示页面：



4.6.10 小结

本章利用一个会员系统来讲述了一个利用.NET 的 Application 的应用。