

第6章 活动服务器组件

在前几章中，我们已经研究了两类不同的对象：一类是 ASP对象模型的内置部分，另一类是脚本运行期库提供的对象。本章将介绍第三类能用于 ASP的对象，取名为活动服务器组件(或者就称为服务器组件)。

这些组件在很多场合下就如同在浏览器里或在像 Visual Basic编程环境中用于客户端脚本中的传统ActiveX 控件一样，但是这些组件可在服务器上运行，而不是一个只能运行在客户机上的对象。

现在的问题是：这些组件来自哪里？其中一部分是由常规的 ASP/IIS安装程序提供的，而另一部分可以从Web上免费或有偿获得。本章讨论 ASP的各种组件的主要用途，然后介绍可得到的其他组件的几个范例。一旦感觉到使用这些活动服务器组件得心应手，你就会毫不费力将其他供应商的产品嵌入到自己的 ASP页面中。本章后面有一个第三方组件的列表，本书后面还将介绍如何创建自己的组件。

还有一些用于ASP的组件，但本章不准备讨论。动态 Web网站技术的基本用途之一是发布直接从数据库管理系统和其他类型的数据存储获得的信息，这一用途和从数据库或其他数据存储中收集数据和存入数据的需求，促使程序员开发服务器端编程。为了使用 ASP完成这些任务，我们可以充分利用称为 ActiveX数据对象(ActiveX Data Objects, ADO)库的组件集。由于它涉及的内容很多，很难在本章中阐述清楚，还要考虑的其他的组件，本书从第 8章开始讲述相关的内容。

本章要讨论的内容有：

- 服务器组件与在ASP中使用的其他对象的区别。
- 简要说明如何实例化对象，并得到服务器组件的引用。
- ASP 3.0和IIS提供的可安装组件，例如 AD Rotator、Content Linking、Page Counter和 Tools等组件。
- 一些第三方提供的组件。

6.1 脚本对象与服务器组件的比较

前一章已说明了各种脚本对象，这些对象对于 ASP/IIS来说都是从脚本运行期库(scrun.dll)中得到的，重要的是不要把这些对象与服务器组件(即本章的主题)混淆一起。

服务器组件通常在其自己的 DLL或可执行文件中实现，例如本章将要讨论的“Content Linking”组件，就是由 nextlink.dll实现的。一旦在服务器安装和注册这个 DLL文件，那么其提供的对象就可以在ASP所支持的任何脚本语言中使用。组件与系统环境的关系如图 6-1所示。

大多数ASP/IIS标准组件针对的是特定任务，也就是要在 Web网站或局域网上完成的任务，当然，最终想借助 ASP达到的任务必须依赖于要想发布的信息和网站的目标。使用预装组件可以提供一个良好的开端。

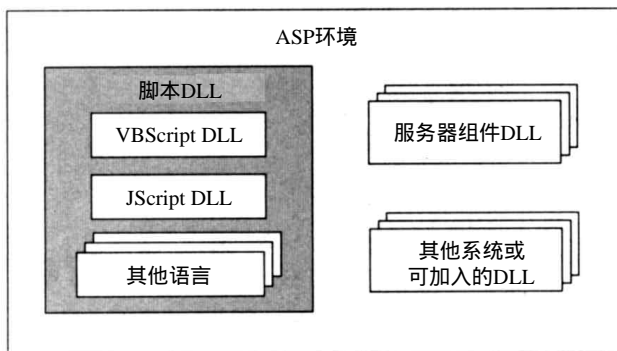


图6-1 组件与系统环境的关系

使用服务器组件

前一章已经介绍了对象和组件在页面中通用的实例化方法，使用服务器组件和使用脚本运行期库对象的方法完全一致，唯一的区别是，不像脚本对象，服务器组件通常作为单独的DLL文件实现，而且可能要求在使用前先运行一个安装程序，或者手工注册该组件（如果这些组件在ASP缺省安装时未注册）。

1. CreateObject方法

一般说来，可以在一个ASP页面中使用ASP内置的Server对象的CreateObject方法创建一个对象，例如：

```
Set objThis = Server.CreateObject("this.object") ' in VBScript
```

或：

```
var objThis = Server.CreateObject('this.object'); // in JScript
```

这样就在变量objThis中创建了该对象的引用，然后可以在脚本中使用该对象；换句话说，可以在代码中根据需要控制其属性及调用其方法。

2. 使用<OBJECT>元素

同样可以使用常规的<OBJECT>元素创建一个对象，该方法同样适用于在Web浏览器页面中创建对象。ASP支持HTML<OBJECT>元素的特殊实现，而且我可以使用这一方法在页面中放置一个对象。要在普通的.asp文件中定义一个组件或脚本对象的实例，使用下面语句：

```
<OBJECT RUNAT="SERVER" SCOPE="PAGE" ID="objThis" PROGID="this.object">
</OBJECT>
```

另一方面，可以使用对象的CLASSID取代“友好名”或PROGID：

```
<OBJECT RUNAT="SERVER" SCOPE="PAGE" ID="objThis"
CLASSID="CLSID:OACE4881-8305-11CF-9427-444553540000">
</OBJECT>
```

我们使用SCOPE属性来设定对象的范围，选项是“SESSION”、“APPLICATION”和“PAGE”。如果在普通的.asp文件中使用<OBJECT>元素，那么必须同时使用“PAGE”属性，并忽略SCOPE属性，因为创建的对象只能在它被实例化的页面中使用。如果在global.asp文件中使用<OBJECT>元素，则所创建的对象可在当前会话或整个应用程序中使用，因此SCOPE属性可设置为“SESSION”和“APPLICATION”。

3. 检测对象实例的存在

经常出现的错误是企图创建一个未安装 (或未注册) 的对象的实例, 或者在 CreateObject 方法或 <OBJECT> 元素中使用错误的对象 ProgID。缺省状态下, ASP 会出现错误并停止页面的执行。

给页面添加一些程序代码, 在访问对象之前进行检测是必要的。在 VBScript 中, 可以做到这一点, 方法是: 关闭缺省的错误处理, 然后使用 IsObject 函数查看对象变量是否真的引用了一个对象。

```
'in VBScript:
...
On Error Resume Next 'turn off default error handling
Set objThis = Server.CreateObject("this.object")

If IsObject(objThis) Then
    'the CreateObject method succeeded
    On Error Goto 0 'turn the default error handling back on
    ...
    'rest of the script goes here
...
Else
    Response.Write "Sorry, this page cannot be accessed at present"
    Response.Flush
    Response.End
End If
...
```

在 JScript 语言中, 可以使用 JScript 5.0 的脚本引擎中新的错误处理特性, 这在第 1 章中介绍过。

```
// in JScript:
...
try {
    var objThis = Server.CreateObject('this.object');
    ...
    // rest of the script goes here
    ...
}

catch (exception) {
    Response.Write('Sorry, this page cannot be accessed at present');
    Response.Flush();
    Response.End();
}
...
```

6.2 ASP/IIS 可安装组件

ASP/IIS 的标准安装包括一系列服务器组件, 这些组件在 IIS 文档中被描述为可安装组件 (installable component), 尽管缺省状态下它们是完全安装。但也有可能这个名称仅仅是沿用 ASP 的前几个版本中的名称。在以前版本中, 这些组件需要人工安装。现在, 如果要给它们起个合适的名称, 可称为“可实例化的组件”。

需要注意的是, 在第 8 章将介绍一个特定的可安装组件, 即 ADO 组件。后面的章节中使用的其他组件都是作为特定的服务或服务器应用程序的一部分而安装的。例如协作数据对象

(Collaborative Data Objects, 简称CDO和CDONTS)组件, 用于邮件服务器和电子邮件消息。本章的其余部分将讨论, 如何使用其他通用性较强的组件为 Web网站或Intranet提供增强功能。

在Windows 2000中的WinNT\System32\intesrv\目录下可找到这些 ASP可安装组件, 该目录也放置一些组件要求的配置文件。下面按字母顺序列出了一些组件。

- Ad Rotator 组件用于控制可点击图像的出现, 如广告或标题。使用存储在配置文件中的信息, Ad Rotator能够依据配置文件规定的比率显示不同的图像, 同时也具备把可点击的广告重新定位到相应的URL以及记录图像点击次数的特性。
- Browser Capabilities组件使用UserAgent字符串, 该字符串由浏览器发往服务器, 用来识别相应的浏览器。它提供了反映特定浏览器功能的一系列属性, 例如是否支持脚本、cookie或Java小程序。
- Content Linking组件用于提供一系列方便导航的页面, 这些页面可被改变或更新而不必编辑各个页面。该组件采用一个文本文件(含有页面URL和显示顺序), 使创建按指定顺序链接起来的一些页面很容易。
- Content Rotator组件采用一种配置文件来指定插入到其他页面中的 HTML、文本或程序代码, 根据配置文件指定的情况, 显示内容的任何部分。
- Counters组件能够创建、递增和永久地存储各个在脚本代码中使用的整数计数器的数值。
- Logging Utility组件提供对Web服务器日志文件的访问, 并且允许在一个ASP页面内读取和更新这些数据。这是Windows 2000中随IIS 5.0和ASP 3.0新增的特性。
- MyInfo组件为可在整个网站内访问的名称/值对的存储提供了一个存储空间, 这些数据在服务器上以XML格式的文本文件存储(不同于应用程序作用域, 这些数据在Web网站范围之外仍然保留)。
- Page Counter组件记录访问一个页面的次数, 可以用于跟踪页面的点击次数。
- Permission Checker组件有助于防止当一个用户试图访问没有正确的权限设置的资源时引起的脚本错误。
- Tools组件提供了一些在页面中可以采用的方法, 这些方法用来检验一个文件是否存在、处理HTML窗体、生成一个随机整数。另外还有一些用在Macintosh苹果计算机上的方法, 可以检查是否存在某个服务器插件程序, 检查当前用户是否为网站拥有者。

另外一个由IIS/ASP提供的组件, 即Status组件, 只能用于Macintosh计算机的个人Web服务器上。本书没有涉及该组件, 若要详细了解该组件, 可通过HTTP://localhost/iis/htm/asp/comp1qt0.htm查阅ASP/IIS提供的IIS 5.0文档。

6.2.1 使用示例页面

本书已经提供了一系列的示例页面来说明所有这些组件的用法。

可以从Wrox Web网站上下载这些实例页面, 并在自己的计算机上运行。在下载的文件Chapter06子目录中会找到本章所用的程序代码。

在运行示例页面之前, 确保MyInfo和Counter组件的“全局”实例化代码已包含在缺省Web站点的global.asa文件中。在示例文件中提供了global.asa文件中所需要的代码, 可以将这些代码加到现有的global.asa文件中, 不需要更换整个文件。

```

<!-- declare instance of the ASP Counters component with application-level scope -
-->
<OBJECT ID="objCounters" RUNAT="Server" SCOPE="Application"
PROGID="MSWC.Counters">
</OBJECT>

<!-- declare instance of the ASP MyInfo component with application-level scope -->
<OBJECT ID="objMyInfo" RUNAT="Server" SCOPE="Application" PROGID="MSWC.MyInfo">
</OBJECT>

```

示例文件的主菜单页面 (Default.asp) 使用 Content Linking 组件创建到每一个页面示例的链接。因此，我们将首先研究这个组件，也涉及了 Ad Rotator 和 Counters 组件的有关问题。

6.2.2 Content Linking 组件

Content Linking 组件对网站是非常有用的工具，它提供了内容页面，或者包含对同一站点上其他页面的链接列表的页面。它自动把当前显示页面的 URL 与存储在服务器里的文本文件的页面列表匹配起来，并且能允许用户通过页面列表按正反顺序进行浏览。也就是说，即使访问者已经在内容页面中点击了一个链接并且正在查看列表中的某一个页面，该组件仍然会辨认出所访问的页面在列表中的位置。

因为所有详细资料均在一个文本文件中，维护网站以及页面间的链接仅仅需要编辑这个文本文件。例如可以通过在内容链接列表文件中重新排列页面的顺序，来改变它们的显示顺序。

1. 内容链接列表文件

内容链接列表文件包括一个简单的按显示顺序排列的页面 URL 列表，同时也提供了相应的描述，用于在内容页面中显示链接文本。如果需要，还可以增加对每个页面的注释，注释可以帮助识别链接，但对访问者来说注释是不可见的。

列表文件对于每个页面包含一个文本行。每一行由 URL、描述和注释所组成，各部分由 Tab 字符(不是空格，否则文件将不能工作)分隔，最后用回车结束。例如：

```

newpages.htm      New additions to our site      we update this weekly
offers.htm        Special Offers for this week  we only update this monthly
register.htm       Registration for new users
main.htm          The main forum and chat area      must be registered first
index.htm         Back to the contents page

```

请注意目标 URL 必须由一个相对的虚拟或物理的路径来说明，诸如 forum\enter.htm。URL 不能使用“http:”、“//”或者“\”作为开头，如果要创建上述这样含有绝对 URL 的菜单，可以使用 ASP 建立一个重新定向页面，而且在内容链接列表文件中对其进行指定，例如，建立一个名为 redirect.asp 的页面，包含如下代码：

```

<%
'redirect the request to the site specified in the query string
Response.Clear
Response.Redirect Request.QueryString("url")
%>

```

可以在内容链接列表文件中按下述方式使用 redirect.asp 文件：

```

redirect.asp?url=http://www.cyscape.com/      CyScape BrowserHawk Web Pages
redirect.asp?url=http://www.softartisans.com/ SA fileUp Component Pages
redirect.asp?url=http://www.stonebroom.com/swindex.htm  Stonebroom RegEx Pages

```

2. Content Linking组件的成员

Content Linking组件提供了八种方法，如表 6-1所示。这些方法能够检索来自内容链接列表文件的条目，既能相对于当前页面检索条目，也可以使用索引号检索绝对条目。第一个条目的索引号是1。

表6-1 Content Linking组件的方法及说明

方 法	说 明
GetListCount(links_file)	返回文件links_file中条目的数量
GetListIndex(links_file)	返回文件links_file中当前页面的索引号
GetNextURL(links_file)	返回文件links_file中下个页面的URL
GetNextDescription(links_file)	返回文件links_file中下个页面的描述
GetPreviousURL(links_file)	返回文件links_file中前一个页面的URL
GetPreviousDescription(links_file)	返回文件links_file中前一个页面的描述
GetNthURL(links_file, n)	返回文件links_file中第n个页面的URL
GetNthDescription(links_file, n)	返回文件links_file中第n个页面的描述

如果当前页面不在内容链接列表文件中，则：

- GetListIndex返回0。
- GetNextURL和GetNextDescription返回列表文件中最后一个页面的URL和描述。
- GetPreviousURL和 GetPreviousDescription返回列表文件中第一个页面的URL和描述。

3. 使用Content Linking组件

一旦创建了自己的内容链接文件，就可以把该组件添加到页面中。这里有一个典型的示例，它遍历内容链接文件中的所有条目，并且创建一个页面列表（以超链接表示页面）。

```
<% 'in VBScript:
Set objNextLink = Server.CreateObject("MSWC.Nextlink")
%>

<UL>
<%
'get the number of entries in the menu file
intCount = objNextLink.GetListCount("contlink.txt")

'loop through the entries
For intLoop = 1 To intCount %>
  <LI>
  <A HREF="<% = objNextLink.GetNthURL("contlink.txt", intLoop) %>">
  <% = objNextLink.GetNthDescription("contlink.txt", intLoop) %>
  </A>
<%
Next
%>
</UL>
```

这段代码首先创建了一个 Content Linking对象。在普通的和标记之间，使用Content Linking对象的GetListCount方法查找在内容链接列表文件有多少条目。然后再遍历这些条目，对于每一个条目在页面中放入一个标记，后面跟着一个<A>标记。HREF属性的值从列表文件中使用 GetNthURL方法检索得到，用作链接文本的描述使用 GetNthDescription方法得到。

注意，内容列表文件和使用它的 ASP页面应处在同一文件夹中。否则，应提供一个相对物理路径或一个完整的虚拟路径，如下所示：


```
intCount = objNextLink.GetListCount("links\contlink.txt") 'physical path
intCount = objNextLink.GetListCount("/demo/contlink.txt") 'virtual path
```

(1) 创建一个内容页面

我们已使用 Content Linking 组件为本章提供的示例创建了菜单页面。Chapter06 文件夹中的页面 Default.htm 采用上述代码创建菜单，如图 6-2 所示。



图6-2 使用Content Linking组件创建的菜单页面

可以看到内容链接列表文件的内容显示在链接的下面，我们使用 FileSystemObject 对象完成这一工作。

```
<%
'create an instance of a FileSystemObject object
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")

'open the text file as a TextStream object
Set objTStream = objFSO.OpenTextFile(Server.MapPath("contlink.txt"), ForReading)
Response.Write objTStream.ReadAll 'read the whole file and put into page
objTStream.Close
%>
```

如果你编辑了 contlink.txt 文件，下次再次调用这个页面时，会看到菜单的条目发生了变化。注意文件中的最后三个条目，这三个条目使用前面介绍的 redirect.asp 技术，把绝对（而不是相对）的 URL 插入到列表中。

(2) 浏览这些页面

创建一个 Content Linking 组件实例并且使用它的一个方法时，将把当前页面的 URL 与指定的内容链接列表文件中的条目相匹配。不仅能用该组件创建一个内容列表（就像刚看到的那样），而且当在浏览器上打开其中一个页面时，可以用该组件对列表中的各个页面进行导航。

这意味着可以用超链接或按钮从这些页面中的一个移到另一个。例如，可以给页面添加

Next和Back按钮,通过使用GetNextURL和GetPreviousURL方法能知道列表中的哪一个页面是前一个,哪一个页面是下一个。另一方面,能用 GetNthURL方法跳到列表中的任何页面,用GetListIndex方法能知道当前页面在列表中的位置。

下面是一段给页面添加 Next和Back按钮的代码,我们要做的只是把这段程序放在内容链接文件所列出的每一页中,或者用 SSI 的#include指令将它插入其中。

```
<!-- need a form to force Navigator to display the buttons -->
<FORM ACTION="">

<% 'in VBScript:
'create an instance of the Content Linking component
Set objNextLink = Server.CreateObject("MSWC.NextLink")
'set the content linking list file path and name
strListFile = "contlink.txt"
'get the index of the current page in the list
intThisPage = objNextLink.GetListIndex(strListFile)

If intThisPage > 1 Then 'can go back
%>
  <INPUT TYPE=BUTTON VALUE="&lt; Back"
    ONCLICK="location.href='<% = objNextLink.GetPreviousURL(strListFile) %>';"
    TITLE="Go to '<% = objNextLink.GetPreviousDescription(strListFile) %>'">
  &nbsp;
<%
End If
%>

<INPUT TYPE=BUTTON VALUE=" Menu " ONCLICK="location.href='Default.asp';"
  TITLE="Return to the main 'Installable Components' menu">

<%
If intThisPage < objNextLink.GetListCount(strListFile) Then 'can go forward
%>
  &nbsp;
  <INPUT TYPE=BUTTON VALUE="Next &gt;";
    ONCLICK="location.href='<% = objNextLink.GetNextURL(strListFile) %>';"
    TITLE="Go to '<% = objNextLink.GetNextDescription(strListFile) %>'">

<%
End If
%>
</FORM>
```

第一步是创建Content Linking组件,然后把列表文件名存入一个变量中,以便后面在创建页面里的客户端JScript代码时能够使用这个列表文件。另外这样做也便于页面的维护,因为如果想使用一个不同文件名仅需在一个地方改变就可以了。

现在让我们了解一下在链接列表中所处的位置。GetListIndex方法提供当前页面的索引号,从列表中的第一页开始。如果当前页面的索引号比1大,那么可以后退,所以我们在页面中创建Back按钮。如果当前页面的索引号比列表中的项数少,创建Next按钮。页面中总是包括一个Home按钮,以便访问者在任何时候都能轻易地返回到内容页面。

如果打开列在主菜单中的页面之一的话,将会看到使用这项技术在每个页面上创建的一系列导航按钮,创建按钮的代码在文件contlinkbuttons.inc中,可用#include指令将它插入每个页面中。图6-3所示为带有按钮的界面。

在图6-3中有三个按钮,使用GetPreviousDescription方法获得上个页面的链接文本,并把这个文本放入Back按钮的TITLE属性中,作为弹出的提示。

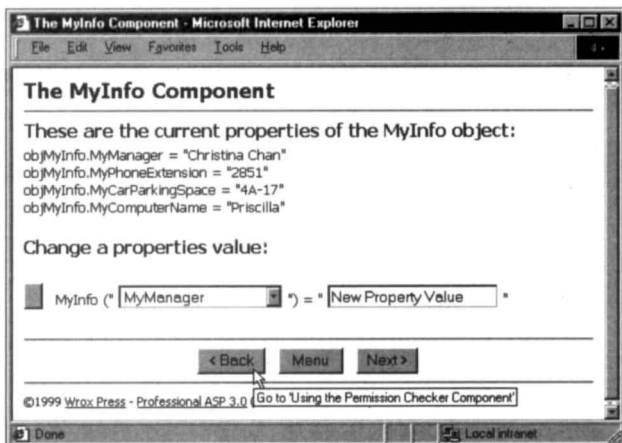


图6-3 带有按钮的界面

```
<INPUT TYPE=BUTTON VALUE="&lt; Back"
ONCLICK="location.href='<% = objNextLink.GetPreviousURL(strListFile) %>';"
TITLE="Go to '<% = objNextLink.GetPreviousDescription(strListFile) %>'>
```

Next按钮采用GetNextDescription方法完成类似的工作。

(3) 用JavaScript跳转

这些导航按钮使用JavaScript (而不是VBScript)装载新页面, 这样就能够和大多数现代浏览器兼容。例如Back按钮包含下面的属性:

```
ONCLICK="location.href='<% = objNextLink.GetPreviousURL(strListFile) %>';"
```

当页面的HTML版本由ASP创建时, 在<%...%>标记中的表达式用其数值取代, 所以发往浏览器的网页看起来如下所示:

```
ONCLICK="location.href='permissionchecker.asp';"
```

为什么选择使用变量来表示内容链接列表文件的名字? 其原因是在单个语句中包括一个三层的嵌套引用是非常麻烦的。

注意浏览器的缺省语言是JavaScript(或IE中的JScript), 而不是VBScript, 通过使用JavaScript, 可以避免在OnClick代码中必须提供LANGUAGE参数, 同时对非IE浏览器提供兼容性。

6.2.3 Ad Rotator组件

Ad Rotator组件允许浏览器在引用ASP页面时每次显示不同的图像, 这项技术经常用在显示广告的站点上。每次打开或重新载入页面时, ASP采用“旋转调度文件”(rotator schedule file)中的信息选择一副图像, 并插入网页中。然而Ad Rotator组件能做更复杂的工作, 它可以把图像设置成超级链接而不是一副静止的图像, 并能记录用户对每个广告的点击次数。

在本章所介绍的组件中, Ad Rotator组件可能是最难使用的, 因为它涉及几个不同的文件。在查看每个文件之前, 图6-4所示的工作过程总览可能有助于了解它们之间的组合方式。

访问者可以浏览页面中由旋转调度文件指定的广告。通常这是个超级连接, 并且对其点击会载入一个重定向文件。这是个常规的ASP页面, 能记录用户对文件的操作或点击计数, 然后重新定向到相应的广告主的网站。

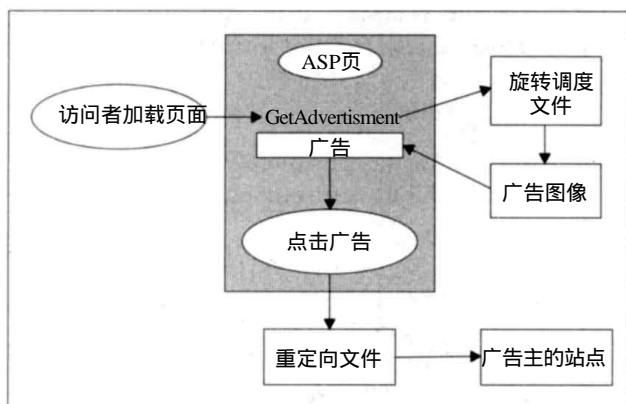


图6-4 Ad Rotator组件的工作过程总览

1. 旋转调度文件

Ad Rotator组件依赖于该文件来指定显示的广告或图像。这个文件包括每个图像文件的名字、显示的尺寸大小以及显示时间的相对百分数。该文件分为两部分，由一个仅含有一个星号(*)的行分隔。

文件格式如下：

```

REDIRECT URL
WIDTH width
HEIGHT height
BORDER border
*
adURL
adHomeURL
text
impressions
  
```

其中：

- URL是执行重新定向的ASP文件或程序的虚拟路径及名称。
- width和height定义了页面中广告条的大小(以像素为单位)，缺省值为440 × 60。
- border规定了广告条边界的宽度(以像素为单位)，缺省值为1，无边界时为0。

第一部分是可选的，设置应用于该文件中所有广告条的缺省值。这给设置所有图像的参数提供了一种方法，边界宽度可以使用Ad Rotator组件中的border属性来设置。如果忽略了一项或多项此类任选参数，而且没有明确设置对象属性，那么参数就使用其本身的缺省值。如果忽略所有这些参数，仍需包含“*”作为文件的第一行。

第二部分，即“*”后的部分，必须存在于该文件中，它提供了各个广告条的细节。对于每个广告条，下面四个参数要重复使用。这些参数是：

- adURL，是广告图像文件的虚拟路径和文件名。
- adHomeURL，是广告跳转到的URL，连字符(-)表明该广告条没有链接。
- text，浏览器不支持图形文件时显示的文本。
- impressions，是一个整数，指明了此广告条显示的相对时间或时间比率，例如一个文件包含三个广告条，而且它们的impressions值分别设置为4、6和10，那么表明返回的页面中有20%可能出现第一个广告条，30%出现第二个广告条，50%出现第三个广告条。这

没有指明广告条在浏览器中显示的实际时间。注意每一个广告条的实际出现是随机的，并被修改以达到需要的比率。

下面看一个重新定向文件的例子：

```
REDIRECT AdRotFiles/ad_redirect.asp
WIDTH 400
HEIGHT 50
*
AdRotFiles/wrox.gif
http://www.wrox.com/
Better Books from Wrox Press
3
AdRotFiles/lunar.gif
http://www.going-to-the-moon.com/store.asp
Acme Lunar Boost Supplies
1
```

2. Ad Rotator组件的成员

Ad Rotator组件含有一个方法和三个属性，方法如表 6-2所示。

表6-2 Ad Rotator组件的方法及说明

方 法	说 明
GetAdvertisement(<i>schedule_file</i>)	使用指定的 <i>schedule_file</i> 中的信息创建并返回一个字符串，这个字符串含有把下一个合适的图像插入到该页面所需的 HTML

属性如表6-3所示。

表6-3 Ad Rotator组件的属性及说明

属 性	说 明
Border	整数型，是指广告条边界的宽度(以像素为单位)，若未设定，则采用文件第一部分设定的值
Clickable	布尔型，规定广告栏图像文件是否显示为一个超级链接，缺省值为 True
TargetFram	字符串型，目标框(frame)的名字。如果用户点击图像文件的话，这个框用于显示广告主的页面。如果忽略的话，页面装载到当前浏览器的目标框或窗口内，取代含有广告栏图像的页面，这个属性也能设置成一个标准的 HTML框标识：_top、_new、_child、_self、_parent和_blank

3. 使用Ad Rotator组件

下面的代码说明了 Ad Rotator组件的使用。创建了组件实例之后，设置属性参数并调用 GetAdvertisement方法，指定调度文件的相对物理路径。由组件返回的 HTML代码通过使用 Response.Write方法插入到页面中。

```
<% 'VBScript example
QUOT = Chr(34)
Set objAdRot = Server.CreateObject("MSWC.AdRotator")
objAdRot.Border=0 'no border
objAdRot.Clickable=TRUE 'is a hyperlink
objAdRot.TargetFrame="TARGET=" & QUOT & "_blank" & QUOT 'load into new window named fraAdFrame
strHTML = objAdRot.GetAdvertisement("AdRotFiles\ad_schedule.txt")
Response.Write(strHTML) 'put the HTML into the page
%>
```

GetAdvertisement方法的参数是相对于当前目录或 Web网站根目录的调度文件物理路径和文件名。在上面的代码中，此文件取名为 AdRotFiles，位于包含正在运行的 ASP页面的当前目录下的子目录中。如果这个文件存入 Web网站根目录下的 AdRotFiles子目录，应使用“\AdRotFiles\MyAdFile.txt”。

调用GetAdvertisement方法时，返回能插入网页中用于创建广告栏的 HTML代码，对于上例中的旋转调度文件中的第一个广告栏，可以得到下列代码：

```
<A HREF="/AdRotFiles/AdRedirect.asp?
    url=http://www.wrox.com/&image=/AdRotFiles/AdPics/wrox.gif">
<IMG SRC="/AdRotFiles/AdPics/wrox.gif"
    ALT="Better Books from Wrox Press" WIDTH=440 HEIGHT=60 BORDER=0></A>
```

可以看到在常规的 <A>标记中放置了图像文件，HREF属性设置为重定向文件的名称 AdRedirect.asp，同时广告主的主页 URL和所用的图像添加到查询字符串中，因此当用户点击广告栏时，重定向文件将在服务器上载入并运行。

4. 重定向文件

重定向文件可以是一个 ASP页面、一个 ISAPI DDL 或者是一个 CGI 应用程序，无论是哪一种，它必须能够接受和辨别广告主的主页 URL 以及图像名参数。它检查这些值并决定下一步该做什么。例如通常会把用户重新定向到与此广告栏有关的 URL，这一点使用下面的 VBScript 代码很容易实现。

```
Response.Redirect Request.QueryString("url")
```

然而，重新定向文件提供给我们的不仅仅是这些。例如可以统计点击每个广告的用户数量，如果根据点击数收取广告费用的话，这将显得尤其有用。也可以用本章后面介绍的 Counters 组件来完成这项工作。

```
...
'get the URL of the advertiser's target page
strAdvertiserURL = Request.QueryString("url")
If Instr(strAdvertiserURL, "wrox.com") Then
    'this is an advert for the Wrox Press site
    objCount rs.Increment("wrox")
    Response.Clear
    Response.Redirect strAdvertiserURL
End If

If Instr(strAdvertiserURL, "going-to-the-moon") Then
    'this is an advert for the Lunar Boost store
    objCounters.Increment("lunar")
    Response.Clear
    Response.Redirect strAdvertiserURL
End If
...
..'same for other advertisers
...
```

查询字符串包含图像文件名时，在脚本中也要考虑这一点，如果在调度文件中，同一广告主有三种不同的图像，可以对每个图像单独地跟踪访问次数。

```
...
If Instr(strAdvertiserURL, "wrox.com") Then
    'this is an advert for the Wrox Press site
    Select Case Request.QueryString("image")
        Case "books.gif"
```

```

objCounters.Increment("wrox_books")
Case "website.gif"
objCounters.Increment("wrox_site")
Case Else
objCounters.Increment("wrox_other")
End Select
Response.Clear
Response.Redirect strAdvertiserURL
End If
...

```

5. 一个Ad Rotater示例页面

本章提供的示例文件包括一个 Ad Rotator 组件演示程序，它能够从前面见过的 ASP Installable Components 主菜单中打开。页面的上部显示的是一个由名为 ad_schedule.txt 的旋转调度文件定义的广告(可在本章示例目录下的 Ad RotFiles 子目录中找到 ad_shedule.txt)，如图6-5所示。

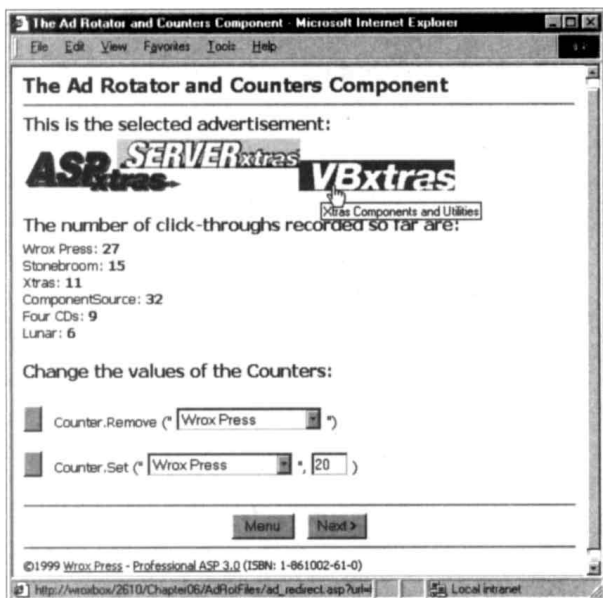


图6-5 AD Rotating 页面示例

使用的代码前面已经讨论过，这里在调度文件中仅增加了另外一些条目，以便更清楚地显示载入页面时的广告的变化情况。如果点击一个广告，将运行名为 ad_redirect.asp 的重定向文件中的代码，并且相应的广告主的页面在新的窗口里打开。完成该工作的代码和前面见到的相同。

在重定向文件中，可使用计数器(由Counters组件创建)实例存储每个广告的点击次数。

```

...
If Instr(strAdvertiserURL, "wrox") Then
objCounters.Increment("wrox")
Response.Clear
Response.Redirect strAdvertiserURL
End If
...

```

此页面的其余部分演示了 Counters 组件的作用，下面介绍该组件。

6.2.4 Counters组件

Counters组件能用于创建、存储、递增和检索每个计数器的值。不要把它和本章后面将要介绍的Page Counter组件混淆，Counters组件能用于支持任何种类数据的统计。

一个计数器含有一个整数值，能通过 Counters组件的方法进行运算。使用 Set方法设置计数器的指定值，用 Get方法检索计数器中的值，使用 Increment方法使计数器的值加 1，使用 Remove方法删除一个计数器。所有的计数器的值存储在一个名为 Counters.txt的文本文件中，可在Counters.dll组件所在的目录中找到该文件。

1. Counters组件的成员

Counters组件提供了用于维护每个计数器组件中数值的四个方法，如表 6-4所示。

表6-4 Counters组件方法及说明

方 法	说 明
Get(counter_name)	返回指定计数器的当前值，如果此计数器先前没有创建，首先创建并设置成 0，其返回值为 0
Increment(counter_name)	增加指定计数器的当前值，如果此计数器先前没有创建，首先创建并设置为 1
Remove(counter_name)	删除指定的计数器
Set(counter_name, value)	把指定计数器的值设置成参数 value提供的整数值，如果此计数器先前没有创建，先创建并设定为指定值

2. 使用Counters组件

由于Counters.txt文件仅有一个所有组件实例都能访问的拷贝。因此，应该只创建单个的Counters组件实例，并且使之对 Web网站的所有页面都是可用的，实现这一点的常用方法是在缺省Web站点根目录下的global.asa文件中创建一个应用程序范围的实例。

采用下面程序：

```
<!-- declare instance of the Counters component with application-level scope -->
<OBJECT ID="objCounters" RUNAT="Server" SCOPE="Application"
        PROGID="MSWC.Counters">
</OBJECT>
```

可以使用Counters组件对需要完成的任务创建一个新的计数器。在下面的程序中，给出了有三项选择的调查问题，并对每一种选择的回答次数进行了统计，当使用者提交包含三项选择的窗体后，将调入这个页面。假设选项通过点击 SUBMIT按钮的 cmdYes、cmdNo和 cmdMaybe来选择，其对应值分别是“是”、“否”和“可能”。

```
<% 'in VBScript:
If Request.Form("cmdYes") = "Yes" Then objCounter.Increment("Response_Yes")
If Request.Form("cmdNo") = "No" Then objCounter.Increment("Response_No")
If Request.Form("cmdMaybe") = "Maybe" Then
    objCounter.Increment("Response_Maybe")
%>
```

如果这是第一次收到一个指定的响应，程序将创建一个新的计数器并自动初始化为 1。

计数器在使用范围上没有限制，因为 Counters对象创建在文件 global.asa中，这意味着在虚拟应用程序或 Web网站中创建的任何一个页面中都是可用的，所以这个“调查计数器”可用在应用程序的任何页面上，记住单个的 Counters对象能提供所需的许多独立计数器，不需要创建很多Counters对象实例。


```
Wrox Press: <B><% = objCounters.Get("wrox") %></B><BR>
Stonebroom: <B><% = objCounters.Get("stonebroom") %></B><BR>
Xtras: <B><% = objCounters.Get("xtras") %></B><BR>
ComponentSource: <B><% = objCounters.Get("compsrc") %></B><BR>
Four CDs: <B><% = objCounters.Get("fourcds") %></B><BR>
Lunar: <B><% = objCounters.Get("lunar") %></B><BR>
```

这些控件在一个<Form>上，点击任何一个小的空白按钮时，此窗体便提交给同一个页面，方法与本章中的所有页面所用的几乎一样。下面的程序是为Remove方法创建控件的HTML代码。



当载入页面时，通过检查 Request.Form 集合查看点击的按钮，如果找到了某个按钮，将运行代码的相应部分。在点击 Remove 按钮的情况下，相应的代码是：

对于Set方法，情况类似，但不仅仅需要从文本框中读取新值，而且在调用 Set方法之前，检查文本框中的值是否是有效值。

```
If Len(Request.Form("cmdSet")) Then
    strCounterName = Request.Form("1stSet")    'get the counter name
    strNewValue = Request.Form("txtSet")        'get the new value
    If IsNumeric(strNewValue) Then              'if it can be converted to a number
        intNewValue = CInt(strNewValue)        '... then convert it
        objCounters.Set strCounterName, intNewValue
        Response.Write "Set counter '<B>' & strCounterName & _
            "</B>' to <B>' & strNewValue & "</B>.<P>"
    Else
```

```
Response.Write "<B>" & strNewValue & "</B> is not a valid number.<P>"  
End If  
End If
```

使用页面中的按钮调用 Counters 对象的方法时，重新载入时会在页面顶端看到一段信息和计数器中的新值，如图 6-7 所示。

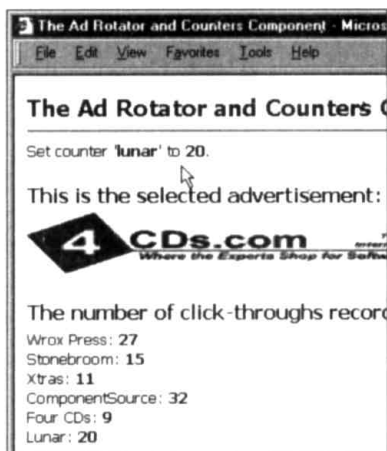


图6-7 调用Counters对象的方法重新载入页面

6.2.5 Browser Capabilities组件

创建各种 Web 网页时面临的问题之一是，不仅仅使用 ASP 技术创建动态网页，而且能够使用 HTML 元素和其他客户端技术，像 Java Applets、ActiveX 控件以及最近出现的 HTML 元素。需要意识到的是，一些访问者若使用了恰好不支持它们的浏览器，那么对于精心编制的网页，访问者看到的可能是文本、图像的杂乱组合，甚至更糟糕的还有相应工作的脚本程序代码。

这里不讨论应当如何设计支持各种不同浏览器的网页(如果了解这方面更多的内容，可以查阅 Alex Homer 编写的，Wrox 出版的《Professional ASP Techniques for Webmasters》一书，书号是 ISN 1-861001-79-7)。然而，确实要引用某个页面时，ASP 和 IIS 提供的 Browser Capabilities 服务器组件可以用来检测浏览器所支持的相关特征。

用户请求来自服务器的页面时，浏览器传送的 HTTP 报头包含了正在使用的浏览器的细节。在 HTTP-speak(它被称为用户代理字符串)中，定义了浏览器的名称、版本、操作系统及其兼容性。Browser Capabilities 组件在自己的配置文件中查找这个字符串，并采用许多与浏览器特征等同的特性。因此，在网页运行的任何时候，Browser Capabilities 组件能够提供支持或不支持某个特性的细节。

在 ASP 3.0 版本中，Browser Capabilities 组件增加了一个新特性。在 ASP 页面中包含 METADATA 指令，指示组件从浏览器中取出一个 cookie，并把其包含的任意值添加到当前的组件实例中作为新属性。这提供了一种方法，从浏览器收集更多的用户特定的信息，而不仅仅是通常从 browscap.ini 文件中得到的浏览器指定的信息。了解了现有的浏览器检测特性如何工作后，再回头介绍新的 METADATA 技术。

1. browscap.ini 文件

Browser Capabilities 组件使用一个基于服务器的 browscap.ini 文本文件，该文本文件必须

和browscap.dll组件文件处于同一目录中。browscap.ini文件包含大多数关于以前和当前浏览器的信息，并且当浏览器的用户代理字符串与文件中的指定字符串都不匹配时，将使用browscap.ini文件中的缺省部分。所以添加关于浏览器的新信息或者更新现有的信息，只需编辑browscap.ini文件。

首先看一下browscap.ini文件的格式，该文件中的所有条目都是可选的。但包括缺省部分是非常重要的。如果使用的浏览器与browscap.ini文件中的任何一个都不匹配，并且没有指定缺省浏览器设置，那么所有的特性将设置成“UNKNOWN”。

下面是browscap.ini文件的格式：

```
; we can add comments anywhere, prefaced by a semicolon like this

; entry for a specific browser
[HTTPUserAgentHeader]
parent = browserDefinition
property1 = value1
property2 = value2
...

[Default Browser Capability Settings]
defaultProperty1 = defaultValue1
defaultProperty1 = defaultValue1
...
```

[HTTPUserAgentHeader]行定义了特定浏览器的起始段，并且Parent行指明了包含浏览器更多信息的另外一个定义。下面的各行定义了我们想通过Browser Capabilities组件可获得的属性以及对于该浏览器的相应值。如果浏览器没有列在所属段中，或者尽管列出了但没有列出所有的属性，将采用Default部分所列出的属性和相应的值。

例如，这个文件包含以[IE5.0]开头的段，这个段包含有Internet Explorer 5.0的相应值，这里没有Parent行，显示的(除了那些在Default部分定义的)仅是我们显式定义的属性。

```
[IE 5.0]
browser=IE
Version=5.0
majorver=5
minorver=0
frames=TRUE
tables=TRUE
cookies=TRUE
backgroundsounds=TRUE
vbscript=TRUE
javascript=TRUE
javaapplets=TRUE

ActiveXControls=TRUE
Win16=False
beta=False
AK=False
SK=False
AOL=False
Update=False
```

此段描述不和任何一个浏览器相匹配，因为，HTTPUserAgentHeader行仅仅是[IE 5.0]。然而，如果把[IE 5.0]作为父代，可以对浏览器添加针对IE5的定义：

```
[Mozilla/4.0 (compatible; MSIE 5.*; Windows 95*)]
parent=IE 5.0
```

```
version=4.0
minorver=0
platform=Win95
```

这样我们把[IE 5.0]指定为浏览器的父代，则显式提供的属性将代替或增加给父代定义的相应的值，但这里也假定任何其他的属性值没有显式地列在其所属段中。

为了识别非常相似的浏览器版本，在 HTTPUserAgentHeader 行可以使用星号通配符，如：

```
[Mozilla/4.0 (compatible; MSIE 5.*; Windows 95*)]
```

将和下面的语句相匹配：

```
[Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)]
[Mozilla/4.0 (compatible; MSIE 5.5; Windows 95 AOL)]
...
```

然而，只有在浏览器发送的用户代理字符串和不含“*”的 HTTPUserAgentHeader 不完全匹配的情况下，才采用通配符匹配。也只有当这种测试失败了，字符串才会企图和含通配符的 HTTPUserAgentHeader 相匹配，并且使用文件中所找到的确实匹配的第一个值。

最后，加上缺省浏览器段：

```
[Default Browser Capability Settings]
browser=Default
Version=0.0
majorver=#0
minorver=#0
frames=False
tables=True
cookies=False
backgroundsounds=False
vbscript=False
javascript=False
javaapplets=False
activexcontrols=False
...
```

这里假设一种最坏的情况，浏览器几乎什么都不支持。应在此基础上定义我们实际想要使用的值。但是，如果定义了一些缺省值为 True，在一个 UNIX 终端上使用纯文本浏览器浏览页面时，可能达不到我们所希望的效果。

维护 browscap.ini 文件

关闭浏览器时，更新 Browscap.ini 文件中相应值使其与浏览器的特性保持一致，增加一些旧的或我们关注的专用的值显然也是非常重要的。为了给用户方便，通常可从 Microsoft Web 网站下载支持 ASP 的一个相当全面的 browscap.ini 版本或其升级版本，而 CrScape Inc 公司提供的 browscap.ini 版本经常比 Microsoft Web 网站上的版本更新一些。

可以在 <http://www.cyscape.com/browscap/> 上找到最新的 browscap.ini 版本，并且订阅一份邮件列表就可自动地收到该文件的最新版本。CrScape 公司也制作一个与 Microsoft Browser Capabilities 组件竞争的组件，称为 browserHank (本章后面将介绍)，新版的 browscap.ini 文件也可从 <http://www.asptracker.com> 上获得。

2. 使用 Browser Capabilities 组件

我们已经掌握了 browscap.ini 文件如何提供包含有关特定浏览器信息的可定制属性，下面介绍如何使用 browser Capabilities 组件。相对而言，使用 Browser Capabilities 组件简单一些，下面创建组件的一个实例并说明其属性。

```
<% 'in VBScript:
Set objBCap = Server.CreateObject("MSWC.BrowserType")
blnVBScriptOK = objBCap.vbscript    'save the result in a variable
If blnVBScriptOK Then
    Response.Write "This browser supports VBScript"
Else
    Response.Write "This browser doesn't support VBScript"
End If
%>
```

上面代码程序检查浏览器是否支持 VBScript并显示一个信息，可以想象这段代码根据浏览器给出的不同响应修改相应的网页，引导用户到不同的页面。

当然，使用 Browser Capabilities 组件的属性可做比这更复杂的工作，一个让人喜爱的技术是根据浏览器支持的属性为网站载入不同的索引网页。如果网站有一套使用帧 (frame) 的页面和一套不使用帧的页面，当用户第一次访问网站时，能够检查浏览器显示帧的能力，并将其重新定位到合适的索引网页上。

3. 使用 Browser Capabilities 的 cookie 特性

新版 Browser Capabilities 组件增加的特性之一是提供了一种方式，以获得更多的有关调用网页的特定客户的信息。

browscap.ini 文件的信息只适用于特定类型的所有浏览器，所以组件仅能报告所安装的浏览器的共同特性，例如能知道浏览器是否支持 cookie，但不能知道用户是否已在浏览器“选项”对话框中关闭了 cookie。

同样，使用复杂的页面设计时，最好了解用户使用的连接类型，以便能选择大小适当的图像文件传送给他们，例如用户通过局域网 (而不是调制解调器) 连接，则允许我们提供更加丰富的环境。如果能知道用户所采用的屏幕分辨率、浏览器所用的语言、操作系统和处理器类型等参数，对于我们的设计是有帮助的。

IE 5 通过使用缺省行为提供这种信息，这是客户端网页的一个元素。IE 5 中的行为是新增加的，其他的浏览器不支持，这是一种对网页中的元素添加特殊功能的方法，通过 STYLE 属性 (或 CSS 风格表项) 和元素联系起来。特别是，IE 5 中提供的 clientCaps 行为能用于提供有关客户机和浏览器设置以及当前选项的信息。

通过创建一种元素和与之相连的 clientCaps 行为，能通过该元素查询到有关客户的信息。下列页面来自我们自提供的示例文件 (browscap_cookie.htm) 正是这样做的。它首先定义了应用于 <IE:clientcaps> 类型的所有元素的包含 clientCaps 行为的风格。这是 XML 语法，在 <HTML> 标记的 XMLNS 属性中使用为当前网页定义的名称空间。

然后，页面 browscap_cookie.htm 从应用 clientCaps 行为的元素得到一系列值，并且建立一个包含这些值的 cookie，最后，把这个 cookie 分配给文档的 cookie 属性，以便有对这个特定服务器目录的页面请求时，将它传送给服务器。

```
<HTML XMLNS:IE>
<HEAD>
<STYLE>
    IE\:clientcaps {behavior:url(#default#clientcaps)}
</STYLE>
</HEAD>
<BODY ONLOAD="createCookie();">
<IE:clientcaps ID="objCCaps" />

<SCRIPT LANGUAGE="JavaScript">
```

```

function stopAllErrors() {
    return true; // prevent display of any errors
}

function createCookie() {
    window.onerror = stopAllErrors;
    var strCookie = new String();
    strCookie = 'width=' + objCCaps.width
        + '&height=' + objCCaps.height
        + '&availWidth=' + objCCaps.availWidth
        + '&availHeight=' + objCCaps.availHeight
        + '&bufferDepth=' + objCCaps.bufferDepth
        + '&colorDepth=' + objCCaps.colorDepth
        + '&javaEnabled=' + objCCaps.javaEnabled
        + '&cookieEnabled=' + objCCaps.cookieEnabled
        + '&connectionType=' + objCCaps.connectionType
        + '&platform=' + objCCaps.platform
        + '&cpuClass=' + objCCaps.cpuClass
        + '&systemLanguage=' + objCCaps.systemLanguage
        + '&userLanguage=' + objCCaps.userLanguage;
    document.cookie = 'BrowsCap=' + strCookie;
}
</SCRIPT>

</BODY>
</HTML>

```

为了使用这个cookie，只需把特定的METADATA指令插入到ASP页面中。如下所示：

```

<!--METADATA TYPE="Cookie" NAME="BrowsCap"
    SRC="browserCapabilities/browscap_cookie.htm"-->

```

现在，运行这个ASP网页时，会自动把页面 browscap_cookie.htm发送给客户机，客户机便运行这个行为特性，然后返回 cookie，随后 Browser Capabilities组件把cookie的内容添加到组件实例的可用属性的列表中，查询方法与查询 browscap.ini文件创建的属性所用的方法相同。

```

width: <B><% = objBCap.width %></B><BR>
height: <B><% = objBCap.height %></B><BR>
...

```

Browser Capabilities示例网页显示两类系列数值，一类是从由 browscap.ini文件决定的属性中收集的数据，另一类来自客户端 cookie页面。当然，不限于仅仅收集来自客户端页面中的 clientcaps行为的值，使用动态HTML技术可以查询浏览器的任何属性或者是像 navigator.appName这样的传统对象属性。Browser Capabilities组件示例页面如图6-8所示。

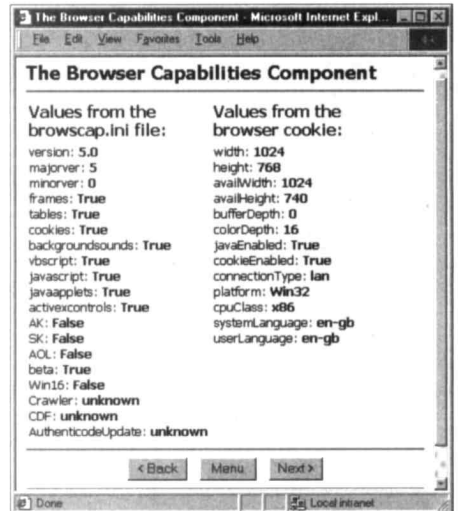


图6-8 Browser Capabilities组件示例页面

6.2.6 Content Rotator组件

Content Rotator组件像一个简化的 Ad Rotator组件，需要提供一种内容调度文件 (Content

Schedule File)。该文件仅是一种包含文本段和 HTML 代码段的文本文件，Content Rotator 组件可在页面中自动显示其中的一个。调度表文件可以包括任何数量的文本内容条目，并可规定比率，用以控制在返回网页中每个条目的显示频率。

1. 内容调度文件

内容进度文件的结构比 Ad Rotator 的调度文件简单，仅需提供要使用的各个文本字符串的列表，由前面加两个百分号 (%%) 的语句行说明。为了设置每个条目的比率 (决定它出现在返回页面中的频率)，在双百分号的后面添加了一个数字，其后还可跟一对前向斜线符 (//) 添加注释。如：

```
%% 3 // This is the first entry in the schedule text file
For more information, mail us at
<A HREF="mailto:feedback@wrox.com">Wrox Press</A>

%% 4 // This is a multi-line text string
<H4>Wrox Press</H4>
<UL>
  <LI> Language Primers
  <LI> Advanced Programming
  <LI> Internet Applications
</UL>

%% 2
Visit us on the <A HREF="http://www.wrox.com">World Wide Web</A>
```

这个例子说明了可以用于网页中的三个文本字符串，比率分别是 3、4 和 2，所以条目将以 3/9、4/9 和 2/9 的频率出现。注意 (和 Ad Rotator 组件一样) 每个字符串的实际出现是随机的，然后经修正后获得相应的比率，如果调入网页九次，可能不会正好得到这个结果。

2. Content Rotator 组件的成员

Content Rotator 组件仅提供两个方法，用于从内容调度文件中检索内容，如表 6-5 所示。

表6-5 Content Rotator 组件的方法及说明

方 法	说 明
GetAllContent(schdule_file)	检索和显示 schdule_file 文件中的所有字符串
ChooseContent(schdule_file)	检索 (但不显示) 来自 schdule_file 文件的下一个合适的内容字符串

3. 使用 Content Rotator 组件

为了从调度文件中获得指定段的文本和 HTML，可使用对象的 ChooseContent 方法检索一个条目，按照内容调度文件中规定的比率。可以使用 Response.Write 方法将其插入到发送给客户的输出流中。

```
<% 'in VBScript:
Set objMyContent = Server.CreateObject("MSWC.ContentRotator")
strContent = objMyContent.ChooseContent("ContentRotator/content_schedule.txt")
Response.Write strContent
%>
```

这个代码使用了和页面在同一目录中的调度文件，如果这段代码存放在其他地方，必须注明调度文件的路径，这个路径可以是一个相对于 Web 网站根目录的物理路径或是一个完整的虚拟路径。

```
objMyContent.ChooseContent("\content\mycontent.txt") 'relative physical path
objMyContent.ChooseContent("/demo/mycontent.txt") 'full virtual path
```

如果想显示内容调度文件中的所有内容字符串，可以使用对象的 GetAllContent 方法，如 ChooseContent 方法的情况一样，必须提供调度文件的路径。例如下面的代码得到了内容调度文件中的所有内容并显示在网页里。注意这个方法把内容自动地插入到网页中，所以不必使用 Response.Write 方法，这个方法也自动地在每个条目之间加了一个水平线。

```
objMyContent.GetAllContent("mycontent.txt")
```

为了研究 Content Rotator 组件的作用，从 “ASP Installable Components” 主菜单打开提供的示例页面，这个页面使用前面讲过的内容调度文件。调用 ChooseContent 方法，并把一个条目放在页面的顶部，然后使用 GetAllContent 方法显示调度文件中所有条目的内容。Content Rotator 组件示例页面如图 6-9 所示。



图6-9 Content Rotator组件示例页面

6.2.7 Page Counter 组件

Page Counter 组件用于统计每个网页被访问的次数，该组件定期把统计的数据存入服务器磁盘上的一个文本文件——访问次数统计数据文件 (Hit Count Data File) 中，所以在停机或出现错误信息，当前数据也不会丢失。

1. Page Counter 组件的成员

Page Counter 组件提供了向该文件增加访问次数的方法以及读取和重新设置访问计数总和的方法，如表 6-6 所示。

表6-6 Page Counter 组件的方法及说明

方 法	说 明
Hits([Page_Path])	返回由 Page_Path 指定的网页的访问次数，如果选项 Page_Path 省略的话，返回的是当前网页的访问次数
PageHit()	增加当前网页的访问次数
Reset([Page_Path])	设置由 Page_Path 指定的网页的访问次数为 0，如果选项 Page_Path 省略的话，设置当前网页的访问次数为 0

2. 使用 Page Counter 组件

下面例子使用 JScript 语言来说明 Page Counter 组件的使用，先用 Server.CreateObject 方法创建对象实例，然后调用 Hits 方法得到对当前网页的访问次数。如果访问次数达到 5 次，则把计数器重新设置为 0，并且在网页中放置一个链接，以便向访问者通知成绩。

```
<% // in JScript:
var objPageCount = Server.CreateObject('MSWC.PageCounter');

// increment the counter and display the current hit count
objPageCount.PageHit();
Response.Write('You are visitor number ' + objPageCount.Hits() + '<P>');
```

```
// we treat every fifth visitor as a winner
if (objPageCount.Hits() == 5) {
    objPageCount.Reset();    // reset the counter
    Response.Write('You\'re a lucky winner!<BR>');
    Response.Write('<A HREF="winner.asp">Go to our winners page</A><BR>');
}
%>
```

本书提供了一个使用 Page Counter 组件的示例网页。然而，由于在每个示例页面里使用一个 VBScript 包含文件，因此采用 VBScript 为这个组件示例编写程序。尽管在一个 ASP 网页里，混用脚本语言是可能的，但在 SSI 的 #include 文件中使用脚本语言能使网页缓存出现问题，并会引起不可预测的结果。虽然脚本语言不同，但示例网页的工作方式与上述相同。Page Counter 组件示例页面如图 6-10 所示。



图6-10 Page Counter组件示例页面

该网页中包含的<FORM>段中含有两个按钮，两个按钮都把窗体提交以重新调入网页。

[illegible]

在网页的开头，与其他的示例网页一样，在请求中查找相关的按钮名字。在本页面中，我们仅对第二个名为 `cmdReset` 的按钮感兴趣，如果点击了该按钮，运行 `Page Counter` 组件的 `Reset` 方法，重新设置本页面的访问次数为 0。

```
'look for a command sent from the FORM section buttons
If Len(Request.Form("cmdReset")) Then objPageCount.Reset 'reset the counter
```

前面看到的代码将调用PageHit方法把计数器设置为1，并显示在网页里。注意已经使用了Hits和Reset方法而没有给可选参数提供数值，所以它们都在当前网页计数器上操作，当然能够在另一个网页中使用，并可以对一些网页的统计结果汇总。

6.2.8 Permission Checker组件

Permission Checker 组件用来检测用户当前正在访问网站的帐号是否允许读取某个特定文件或者访问某个资源 (仅在 Windows NT/2000 而不是在 Windows 9X)。可以用它来定制页面, 这些网页中仅包含当前用户允许访问的资源的链接, 这项技术对于隐藏限制访问的网页和资源是非常有用的, 它可以使被限制的访问者不知道这些资源的存在, 不会产生强行进入其中的企图。

1. Permission Checker组件的成员

Permission Checker组件只有一个方法，如表 6-7所示。

表6-7 Permission Checker组件的方法及说明

方 法	说 明
HasAccess(file_path)	检查当前执行页面的用户帐号是否允许访问 file_path指定的页面。该路径是一个网页、文件或资源的物理的或虚拟的路径如果当前帐号允许访问，返回值为“ True ”，否则；返回值为“ False ”

2. Permission Checker组件的工作过程

通常，除非特别规定，否则用户是匿名访问网页的，因此 IIS本身会使用自己的 Windows 帐号访问网页，缺省值是 IUSR_machinename(例如IUSR_WROXBOX)，Permission Checker组件检查IUSR_machinename帐号是否能够访问指定的网页或资源。

然而，通过Internet Services Manager的Allow Anonymouse Access选项关闭对Web网站或者对某一指定目录的匿名访问，可以强迫 IIS提示用户输入一个有效用户帐号的用户名和口令，然后使用该帐号“非匿名访问”资源。

对一个网站或目录，在 Internet Services Manager上的Properties对话框中的 Directory Security选项卡中设定访问控制。在对话框中的 Anonymous access和Authentication control中，点击Edit按钮打开 Authentication Methods对话框，关闭Anonymous Access选项。设置界面如图 6-11所示。

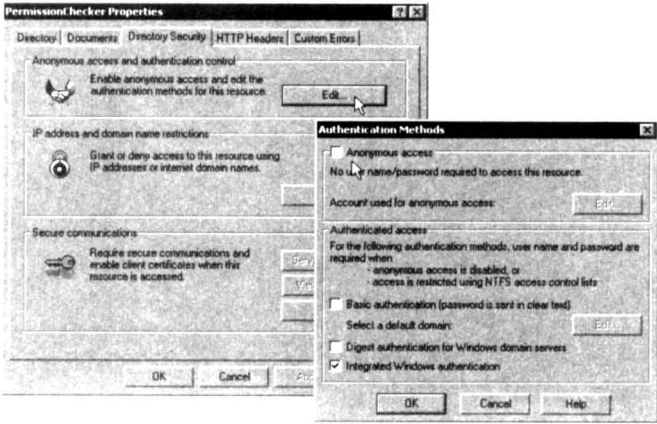


图6-11 访问权限的设置界面

当Permission Checker组件在用户用其自己的帐号(而不是IUSR_machinename)访问的网页中被实例化时，将检查其帐号是否可以访问这个指定的资源，如果不能使用匿名访问，那么所有的用户必须提供帐号细节。可以使用 Windows Explorer对每个文件或资源设置指定的权限(在相应的Properties对话框中的Security选项卡中)。设置界面如图 6-12所示。

3. 使用Permission Checker组件

可以使用Permission Checker组件检查当前用户是否有使用 HasAccess方法访问其他页面(网站上的任何页面)的权限。本书提供了一系列实例页面，可用来试验 Permission Checker组件，打开此示例(从ASP Installable Components主菜单里打开)，会看到网页里的三项链接，如图6-13所示。

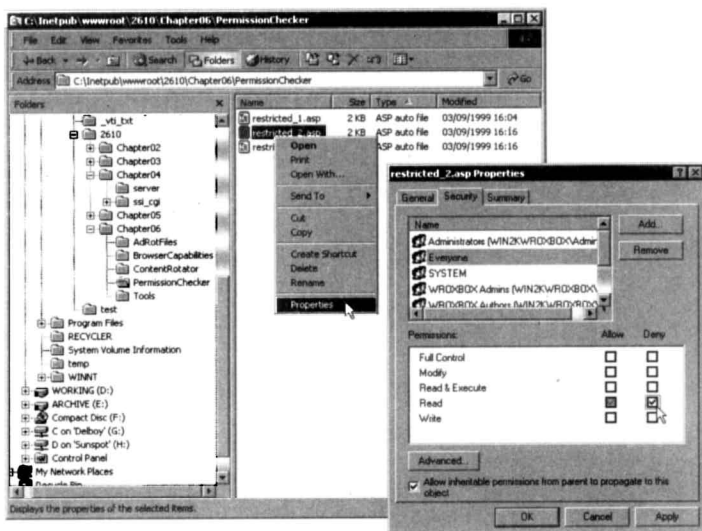


图6-12 权限设置界面

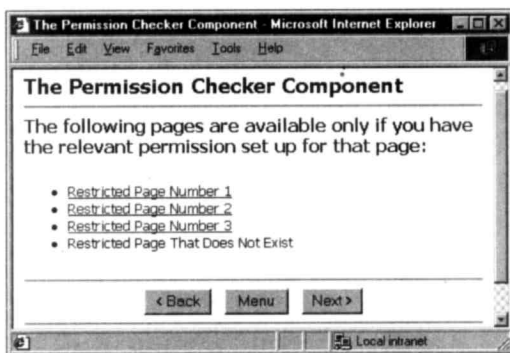


图6-13 使用Permission Checker组件的示例页面

链接所指向的每一个页面都位于 Chapter06示例目录下的 Permission Checker子目录中。Permission Checker组件检查当前用户是否有访问各个网页的权限。如果用户确实有访问这个网页的权限的话，这个网页的条目将是一个超级链接。

'create an instance of the component'

```
Set objPermit = Server.CreateObject("MSWC.PermissionChecker")
```

```
%>
```

```
<UL>
```

```
<LI>
```

```
<% If objPermit.HasAccess("PermissionChecker/restricted_1.asp") Then %>
```

```
<A HREF="PermissionChecker/restricted_1.asp">
```

```
<% End If %>
```

```
Restricted Page Number 1
```

```
<% If objPermit.HasAccess("PermissionChecker/restricted_1.asp") Then %>
```

```
</A>
```

```
<% End If %>
```

```
</LI>
```



```
...
.. 'other page links here
...
</UL>
```

现在我们有访问三个有访问限制的网页的权限，所以这三个条目都是超级链接。然而这里不存在第四个链接（它可以被移去或是删除），因为组件不能使用当前用户（也就是 IUSR_machinename）的帐号来访问它，因此不显示为超级链接。

注意这里选择仅移去超级链接但留下了链接文本，在实际情况中应删除不可访问的网页的整个条目。

如果从 PermissionChecker 子目录中的 restricted_2.asp 文件上删除 Read（和任何其他的）权限，并重新调入网页，对应的条目将不显示为一个超级链接，如图 6-14 所示。

- Restricted Page Number 1
- Restricted Page Number 2
- Restricted Page Number 3
- Restricted Page That Does Not Exist

图6-14 删除Read权限后的页面显示

6.2.9 MyInfo 组件

MyInfo 组件最初是与个人 Web 服务器一起引入的，用于存储服务器管理员提供的个人信息。它现在可以用在 ASP 3.0 中，为名称/数值对，或者任何其他类型的可以表示为文本字符串的信息提供永久存储。该信息以 XML 格式在文本文件 myinfo.xml 中保存，此文件位于 Web 服务器的 inetsrv 目录下。在 Windows 2000 中，其路径为 WinNT\system32\inetsrv\。

和 Counter 组件一样，只需创建一个简单的 MyInfo 组件的实例就可以使得 Web 网站上的所有页面均能获得其信息，确保缺省 Web 站点上的 global.asa 文件中有下面的代码：

```
<!-- declare instance of the MyInfo component with application-level scope -->
<OBJECT ID="objMyInfo" RUNAT="Server" SCOPE="Application" PROGID="MSWC.MyInfo">
</OBJECT>
```

1. 使用 MyInfo 组件

MyInfo 组件缺省时没有属性和方法。然而可给其添加属性，简单地说，也就是赋一个名称和数值给这些属性。比如，可以添加有关工作环境和合作伙伴的信息。

```
<% 'in VBScript:
objMyInfo.MyManager = "Christina Chan"
objMyInfo.MyPhoneExtension = "2851"
objMyInfo.MyCarParkingSpace = "4A-17"
objMyInfo.MyComputerName = "Priscilla"
%>
```

同别的组件属性一样，可以使用下面代码检索数值。

```
<% 'in VBScript:
strManagerName = objMyInfo.MyManager
strPhoneExtension = objMyInfo.MyPhoneExtension
strParkingSpace = objMyInfo.MyCarParkingSpace
strComputerName = objMyInfo.MyComputerName
%>
```

这是一种在两个页面请求之间存储值的有用方法，不需要用户的 Session 对象。这也为将来可能改变的数值提供了一个存储区域，采用这些值的页面将会在下次运行时自动地采集这些变化的值，从而避免必须编辑大量的其他页面。

本书提供了一个使用 MyInfo 组件的简单示例，这个示例从 ASP Installable Components 主

菜单中运行，使用刚才讨论的设置组件属性的代码，并可改变属性值，如图 6-15所示。

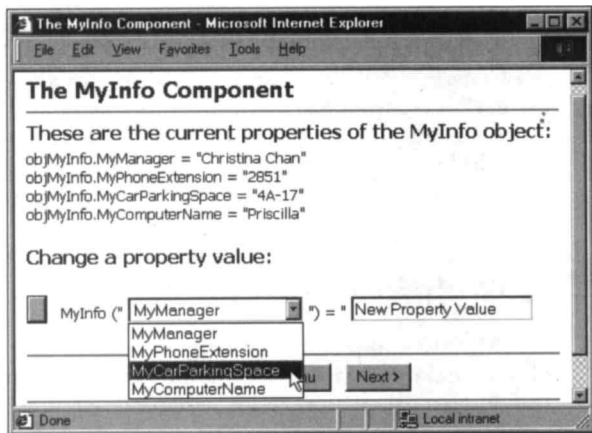


图6-15 使用MyInfo组件的页面

首次打开页面时，创建属性时使用了一个标志变量 SetDefaultValues，该变量存放在用户的 Session对象中。当页面在每个会话里首次打开时，把属性参数设置成“缺省”值，之后，把此标志变量设置成 Yes。在当前会话中，重新调入页面时，这个数值将不再重新设置成缺省值。

```
'an instance of the component is already created in global.asa
If Not Session("SetDefaultValues") = "Yes" Then
    'set the properties if they're not set to the default values,
    'i.e. if this is the first time that the page has been run
    'during the current user session
    objMyInfo.MyManager = "Christina Chan"
    objMyInfo.MyPhoneExtension = "2851"
    objMyInfo.MyCarParkingSpace = "4A-17"
    objMyInfo.MyComputerName = "Priscilla"
    'set a session flag to show that the default values hve been set
    Session("SetDefaultValues") = "Yes"
End If
```

每次调用页面时，检查请求中的值是否是用于改变属性值的按钮的名称。这个按钮在一个包含下拉列表和文本框的 <FORM>上。如果发现点击了这个按钮，则从下拉列表和文本框里采集数值，然后用这些数值改变相关的属性值。

```
'look for a command sent from the FORM section buttons
If Len(Request.Form("cmdChange")) Then
    strPropertyName = Request.Form("lstName") 'get the name of the property
    strNewValue = Request.Form("txtValue") 'get the new value for the property
    objMyInfo(strPropertyName) = strNewValue 'set the component property value
End If
```

6.2.10 Tools组件

Tools组件提供了一些有用的方法，可在页面中检查文件是否存在、处理一个 HTML窗体、以及产生一个随机整数，还有用于 Macintosh计算机的一些方法，还可以检查是否存在某个服务器插件以及检查当前用户是否是网站的拥用者。

1. Tools组件的成员

Tools组件提供了五个方法，其中两个依赖于操作系统，如表 6-8所示。

表6-8 Tools组件的方法及说明

方 法	说 明
FileExists(<i>relative_url</i>)	如果 <i>relative_url</i> 指定的文件存在, 返回值为 True, 否则为 False。必须给出虚拟相对路径及文件名, 并且文件必须存在于发布的 Web网站目录中。
Random()	产生一个位于 -32768~32767之间的随机整数。使用 ABS函数(VBScript)或 Math.abs(JScript)得到在 0~32768之间的正整数。使用 Mod运算符(VBScript)或%运算符(JScript)得到指定范围内的一个数值。例如： <div>intRand = (objTools.Random Mod 76)+25</div> 得到一个在 25~100之间的整数。
ProcessForm(<i>output_url</i> , <i>template_url</i> , [<i>insertion_point</i>])	通过 <i>template_url</i> 指定的文件处理一个 HTML窗体, 并且插入来自窗体中已提交给当前页面的数值。结果写入 <i>output_url</i> 指定的文件, 如果指定了可选项 <i>insertion_point</i> (字符串)参数的话, 组件可在已存在的输出文件中找到这个字符串, 并在该位置插入新的内容。如果 <i>insertion_point</i> 参数没有指定, 任何已存在的 <i>output_url</i> 文件则被新的输出取代
Owner	仅适用于Macintosh机, 如果当前用户帐户是Web网站的拥有者, 返回值为 True, 否则返回值为 False
PluginExists(<i>plugin_name</i>)	仅适用于Macintosh机, 如果指定的服务器 <i>plugin_name</i> 安装在机器上, 返回值为 True, 否则为 False

2. 使用FileExists方法

在允许用户访问之前, 可以使用 FileExists方法检查某些文件是否存在于服务器中 (注意这个方法和FileSystemObject.FileExists 以同样的方式工作)。

下面的例子中, 用户提供了网页的相对 URL, 如果用户想通过在名为 txtURL的文本框中键入URL打开网页, 在重新定向之前可以检查其是否存在。

```
<% // in JScript:
var objTools = Server.CreateObject('MSWC.Tools');
var strURL = Request.Form('txtURL'); // collect the page URL they entered
if (objTools.FileExists(strURL)      // see if it exists
    Server.Transfer(strURL)           // if it does, transfer to it
else                                  // or if not display a message
    Response.Write('Sorry, the page you requested does not exist');
%>
```

这里提供了一个示例页面 (使用VBScript)来演示组件的三种方法 (非Macintosh), 可以从 ASP Installable Components主菜单中运行, 如图 6-16所示。

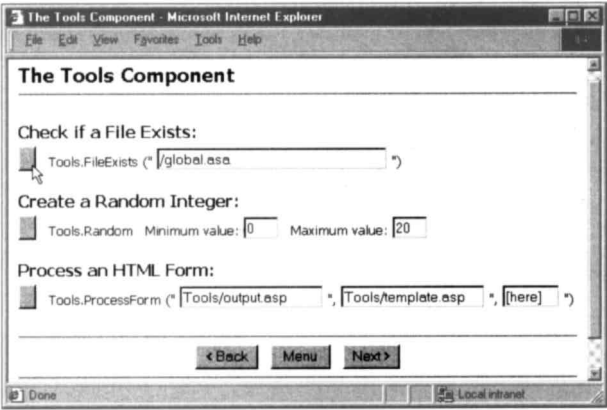


图6-16 运行Tools组件的方法的页面

网页的第一部分允许输入一个文件的相对 URL，并告诉用户该文件是否存在。示例提供的缺省值是查看在网站的根目录中是否有 global.asa 文件。点击按钮时，说明该文件是否找到的信息将放在页面的顶部，如图 6-17 所示。

把页面的所有控件放在 <FORM> 中，提交回本页面，这已经成为一种规范。在页面的开始，查看点击了哪个按钮。如果是 FileExists 的按钮，就调用组件的 FileExists 方法并显示合适的信息。

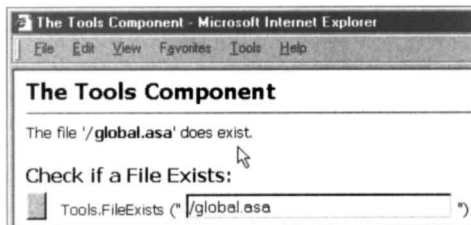


图 6-17 运行 FileExists 方法的结果

```
'look for a command sent from the FORM section buttons
If Len(Request.Form("cmdExists")) Then
    strFile = Request.Form("txtFile")
    If objTools.FileExists(strFile) Then
        Response.Write "The file '<B>' & strFile & "</B>' does exist.<P>"
    Else
        Response.Write "The file '' & strFile & '' <B>does not</B> exist.<P>"
    End If
End If
```

3. 使用 Tools.Random 方法

在 ASP 页面中，有时需要一个随机数来完成某些任务，例如，把用户重新定位到一个随机网页、选择颜色或显示每日提示。可以使用 VBScript 中的 Rnd 函数，但要把所得数值转变成指定范围内的整数。Tools 组件的 Random 方法更易于使用，因为能够直接提供整数值。

Random 方法的结果是一个在 -32768~32767 范围中的整数值，为了获得一个指定范围的整数，可以使用脚本语言中的 Abs 函数并对下一个最大的整数取模。例如为了用 VBScript 语言创建 0~20 的正整数，可以使用下列语句：

```
intRandom = Abs(objTools.Random) Mod 21
```

为了得到在 50~100 之间的数值，可以用：

```
intRandom = (Abs(objTools.Random) Mod 51) + 50
```

示例网页使用这项技术生成随机数时，首先需要检查由用户输入的数值，以保证这些数值既是有效正整数又有正确的相对关系。

```
If Len(Request.Form("cmdRandom")) Then
    intMin = -1 'preset to illegal values and then
    intMax = -1 'only set if a valid number is entered
    strMin = Request.Form("txtMinimum")
    strMax = Request.Form("txtMaximum")

    If IsNumeric(strMin) Then intMin = CStr(strMin)
    If IsNumeric(strMax) Then intMax = CStr(strMax)

    If (intMin >= 0) And (intMax > intMin) Then
        intRandom = (Abs(objTools.Random) Mod (intMax - intMin + 1)) + intMin
        Response.Write "Your random value is: <B>" & intRandom & "</B><P>"
    Else
        Response.Write "<B>The numbers you entered are not valid.</B><P>"
    End If
End If
```

End If

当页面重新调入时，结果显示在网页的顶部，如图 6-18 所示。

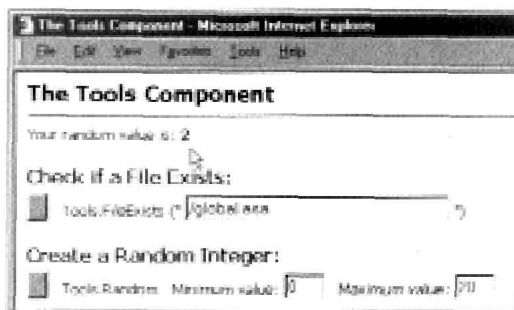


图6-18 运行Random方法的结果

4. 使用Tools.ProcessForm方法

Tools组件中最复杂的方法是ProcessForm，用来读取存在磁盘上的临时文件，并在其中插入创建的信息(可能来自当前页面的Request.Form集合的内容)，然后把结果作为一个文件输出到磁盘，这个方法的语法是：

```
ProcessForm (output_url, template_url, [insertion_point])
```

临时文件和输出文件相对于当前页面使用相对 URL来定义。输出文件可以是 ASP网页，如果这样，当其在浏览器中打开时，将正常处理。临时文件可以包含普通的 ASP代码，但不运行，仅简单地拷贝到输出文件。然而，如果把临时文件中的 ASP代码放在<%%...%%>限定符中，当临时文件调时代码将被执行，这允许动态生成的数值(诸如进行处理的时间和日期)插入到输出页面中。

下面是示例文件template.asp(在chapter06目录的Tools子目录中)：

```
This file was created by the ASP Tools component
-----
The content of the request was:

Output file name: <%% = Request.Form("txtOutput") %%>
Template file name: <%% = Request.Form("txtTemplate") %%>
Insertion point text: <%% = Request.Form("txtInsert") %%>

-----
Created <%% = Now() %%>
```

示例页面包含着预定使用这个临时文件的控件，这些控件创建一个和临时文件在同一个文件夹中的名为output.asp的输出文件，如图6-19所示。

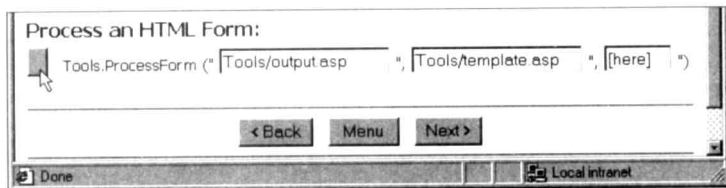


图6-19 运行Tools.ProcessForm方法的界面

点击按钮时，将运行一部分ASP代码，从文本框中采集数据并调用ProcessForm方法：

```
If Len(Request.Form("cmdProcess")) Then

    strTemplate = Request.Form("txtTemplate")
```

```

strOutput = Request.Form("txtOutput")
strInsertPoint = Request.Form("txtInsert")
...
' we display the template contents here
...

'process the form contents
objTools.ProcessForm strOutput, strTemplate, strInsertPoint

...
' we display the output file contents here
...

End If

```

(1) 设置输出文件的访问权限

如果得到一个“MSWC.Tools error 80004005 Couldn't open output file”错误信息，这意味着IIS不允许向指定的目录中写入输出文件。解决这个问题最快捷的方法是，在 Properties对话框中的Security选项卡中，将相应的Tools目录以及存放output.asp的目录的Everyone组的Full Control设置成允许，如图6-20所示。

(2) 查看文件内容

图6-21显示了上面代码用缺省值运行的结果，可以看到原来的临时文件内容，以及使用ProcessForm方法插入到输出文件中的内容。

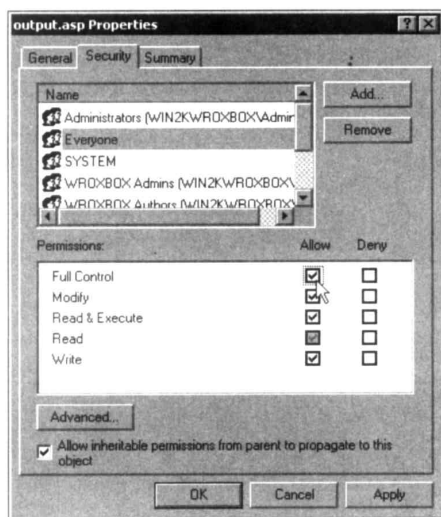


图6-20 设置输出文件访问权限的界面

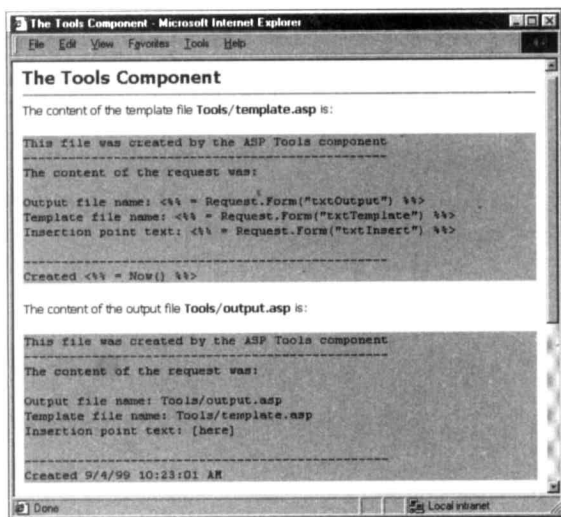


图6-21 输出文件的内容

这里省略了显示来自前面见过的文件内容的程序代码。显示内容的方法与 ASP Installable Components 主菜单页面中用于显示内容链接列表文件的方法相同。使用 FileSystemObject 和 TextStream 对象的实例把整个文件读入到一个字符串中，然后将其插入网页中(记住要使用 Server.HtmlEncode 方法，以便把字符“<”和“>”转换成可用于显示的字符)。

```

QUOT = Chr(34) 'double-quote character
...

```

```
'use a FileSystemObject object to display the template file contents
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")

Response.Write "The content of the template file <B>" & _
    strTemplate & "</B> is:"

Set objTStream = objFSO.OpenTextFile(Server.MapPath(strTemplate), ForReading)
strContent = Server.HtmlEncode(objTStream.ReadAll)    'read whole file
objTStream.Close

Response.Write "<DIV CLASS=" & QUOT & "showcode" & QUOT & "><PRE>" & _
    strContent & "</PRE></DIV>"

...

```

(3) 关于插入点参数

ProcessForm方法中可选的insertion_Point参数能用来在文件中的特定点插入文本，这个方法是先查找输出文件中特定字符串的第一个实例，然后在该位置插入创建的新内容，在输出文件中通过把星号(*)放在insertion_point字符串前面来放置新内容。如果省略了insertion_point参数，新的内容会取代整个输出文件。

注意insertion_point参数在Tools.ProcessForm方法的早期版本中不能使用。

6.2.11 Logging Utility组件

Logging Utility组件是ASP 3.0的新组件，提供了访问日志文件的方法，这个日志文件是由服务器支持的用于Web和FTP网站的IIS创建的。

1. Logging Utility组件的成员

Logging Utility组件提供了六种方法，能读取日志文件的内容、把新记录写进已存在的日志文件中。这些方法和上一章讨论的FileSystemObject对象的方法有相似性，如表6-9所示。

表6-9 Logging Utility组件的方法及说明

方 法	说 明
AtEndOfLog()	用ReadLogRecord方法遍历记录时，如果文件中的所有记录都已读出，返回值为True
CloseLogFiles(io_mode)	关闭打开的日志文件，io_mode的值指定要关闭的文件，常数ForReading(1)表示关闭打开供读出的所有文件，ForWriting(2)表示关闭打开供写入的所有文件，AllOpenFiles(32)表示关闭打开的所有文件
OpenLogFile(filename,io_mode, Service_name, service_instance, Output_format)	打开filename指定的日志文件或日志文件集，可选的io_mode参数可以设置成ForReading(缺省值，1)或ForWriting(2)。当读取到一个特定的IIS服务(例如“W3SVC”和“1”)时。可选的service_name和service_instance参数限制从文件中返回的记录。可选的output_format参数指明写入文件时使用的格式
ReadFilter(start,end)	当在规定的时间和日期范围内读取文件时，限制返回的记录，两个参数都是可选的，如果忽略的话，start是文件的第一个记录，end是最后一个记录
ReadLogRecord()	从用OpenLogFile方法打开的当前日志文件中读取下一个记录
WriteLogRecord (logging_object)	从打开供读出的文件中把记录写入一个打开供写入的文件中，参数是一个引用保存源记录的Logging Utility组件实例的对象变量

Logging Utility组件提供了二十种与IIS日志记录相对应的属性，如表6-10所示。

表6-10 Logging Utility组件的属性及说明

属 性	说 明
BytesReceived	请求时从浏览器收到的字节数
BytesSent	响应时送至浏览器的字节数
ClientIP	客户机或其主机(即代理服务器)的IP地址
Cookie	在请求中传送的cookie内容
CustumFields	加到请求中的定制报头的一个数组
DateTime	请求的日期和时间(GMT)
Method	操作类型,如“ Get ”和“ Post ”
ProtocolStatus	返回到客户机的状态消息,即“ 200 OK ”
ProtocolVersion	协议版本字符串,如“ HTTP/1.1 ”
Referer	包含初始化请求链接的网页的 URL
ServerIP	IIS机器的IP地址
ServerName	IIS服务器的机器名
ServerPort	请求到达的端口号,如“ 80 ”
ServiceName	服务器名称,如“ MSFTPSVC ”或“ W3SVC ”
TimeTaken	检索和创建返回网页的总处理时间
URIQuery	请求中加到URL的查询字符串中的所有参数
URIStem	请求的目标URL
UserAgent	由客户机发送的用户代理字符串
UserName	如果不是匿名地访问服务器, UserName是用户的登录
Win32Status	处理了请求后返回的Win32状态码

2. 使用Logging Utility 组件

Logging Utility 组件最普遍的应用是用于对日志文件的自定义查询。从现有文件中选择一些记录写入新文件的能力,意味着能够汇总某些类型的条目,或者有选择性地挑出一些记录供将来使用。

为了使用ForReading、ForWriting和AllOpenFiles常数,必须在网页的<HEAD>部分包含METADATA指令。

```
<!-- METADATA TYPE="typelib" FILE="C:\WINNT\system32\inetsrv\logscrt.dll" -->
```

为了遍历记录,仅须打开文件并且重复调用 ReadLogRecord,直到 AtEndOfLog的返回值 为True。在下面的例子中将过滤记录,仅获得最后 24小时以内的记录。

```
'create the component instance
Set objLogUtil = Server.CreateObject("MSWC.IISLog")

'open the log file for reading, for the W3SVC instance number 1
objLogUtil.OpenLogFile "extend#.log", ForReading, "W3SVC", 1

'set a filter for the last day's records only
objLogUtil.ReadFilter DateAdd("d", -1, Now), Now

'loop through the records
Do While Not objLogUtil.AtEndOfLog
    objLogUtil.ReadLogRecord      'read the next record
    Response.Write "Request received for page " & objLogUtil.URLStem & " on " _
        & objLogUtil.DateTime & " from IP address " _
        & objLogUtil.ClientIP & ".<BR>"
Loop
objLogUtil.CloseLogFiles(ForReading)  'close the file(s)
```

本书提供了演示使用 Logging Utility 组件的一个示例网页，这个网页可以从 ASP Installable Components主菜单运行。

确保在 Internet Services Manager中不允许匿名访问包含示例文件的目录或 loggingutility.asp文件，否则将不能访问日志文件。打开 loggingutility.asp文件或包含该文件的目录的Properties对话框的Directory Security选项卡，点击Anonymous access and authentication control部分的Edit按钮，打开 Authentication Method对话框，并关闭Anonymous access选项。

必须改变日志文件的文件名以适合你的计算机。在缺省 Web网站的 Properties对话框的 Web Site 选项卡中，打开W3C Extended Logging，如图6-22所示。

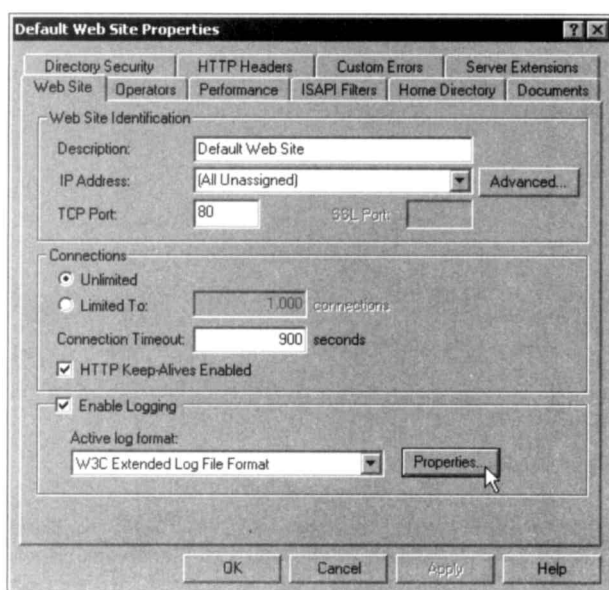


图6-22 使用Logging Utility组件的设置 1

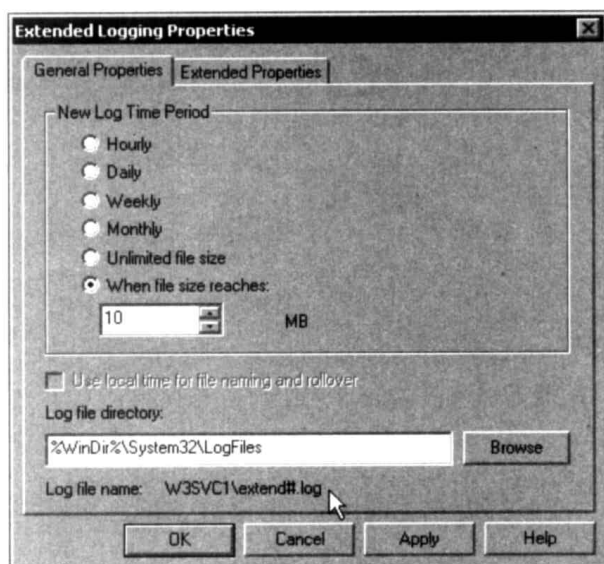


图6-23 使用Logging Utility组件的设置 2

在这个对话框中，点击 Properties 按钮，打开 Extended Logging Properties 对话框，在这里会找到要用的文件名。选择固定大小的日志文件选项，这样文件名将是 extend1.log 和 extend2.log 等等。如图 6-23 所示。

打开 Extended Properties 选项卡，确保选中想要的条目，如图 6-24 所示。

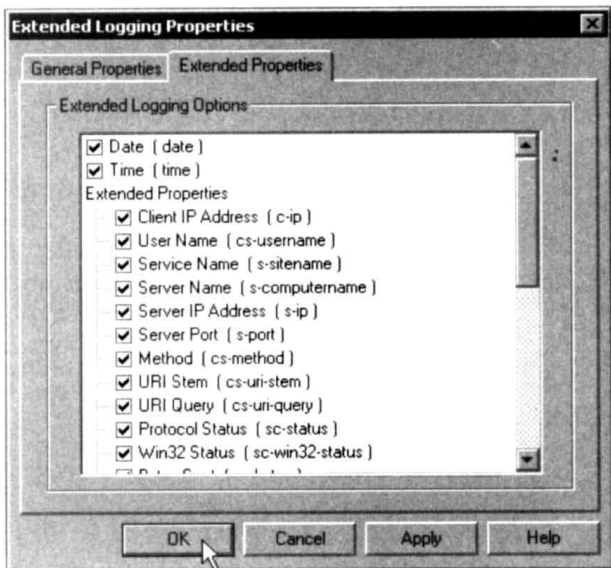


图6-24 使用Logging Utility组件的设置3

现在可以运行示例网页并检查结果，可以使用下列代码在 HTML 表中显示数值。

```
<%
Set objLogUtil = Server.CreateObject("MSWC.IISLog")
objLogUtil.OpenLogFile "extend#.log", ForReading, "W3SVC", 1, 0
objLogUtil.ReadFilter .DateAdd("d", -1, Now), Now
%>
<TABLE CELLSPACING="10">
<TR>
<TH>Date/Time</TH>
<TH>Client IP</TH>
<TH>Bytes Sent</TH>
<TH>Target URL</TH>
</TR>

<%
Do While Not objLogUtil.AtEndOfLog
    objLogUtil.ReadLogRecord 'read the next record
%>
<TR>
<TD><% = objLogUtil.DateTime %></TD>
<TD><% = objLogUtil.ClientIP %></TD>
<TD><% = objLogUtil.BytesSent %></TD>
<TD><% = objLogUtil.URLStem %></TD>
</TR>
<%
Loop
objLogUtil.CloseLogFiles(ForReading)
%>
```

其结果如图 6-25 所示。

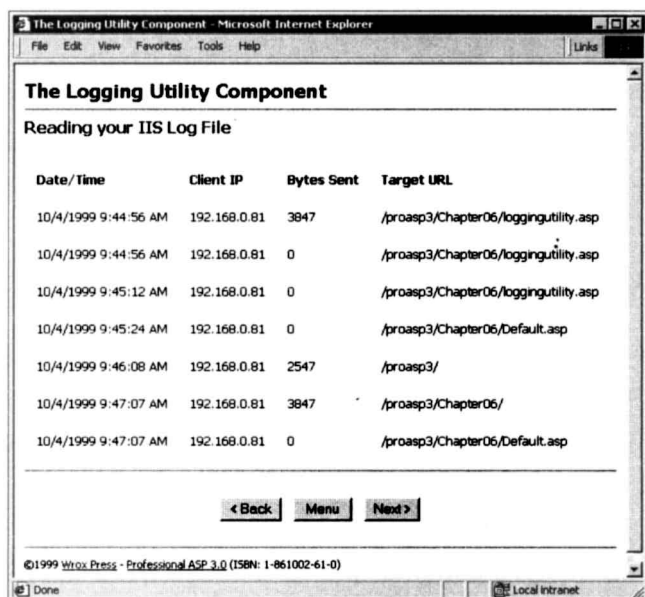


图6-25 使用Logging Utility组件的示例运行结果

6.3 第三方服务器组件

在本节中，简要概述ASP系统中要用到的一些常见的商用和免费的第三方服务器组件。

开发Web网站时，必须完成的两个任务是管理兼容性以及向服务器上载文件。下面将介绍的两个组件能有助于完成上述的任务，而且还介绍另外一个组件，可取代 Microsoft的 Registry Access组件(该组件一般是从相应的Web网站得到的)。

在附录G中，给出了一些最为有用的组件的清单。

6.3.1 BrowserHawk组件

很多人使用由IIS及ASP提供的Browser Capability组件时，都会遇到一个麻烦，浏览器的定义文件(browscap.ini)要求定期的维护或替换以跟得上新版浏览器的要求。

Cyscape公司提供了browscap.ini的升级版本，同时提供了他们自己的能在服务器端和客户端进行浏览器检测的组件，而且，还提供了便捷的图形界面用于查看及修改其自己的浏览器定义文件，如图6-26所示。

可以看到对于每个浏览器类型，都有许多属性，如动态HTML、样式表、JavaScript版本、文件上载能力、SSL(Secure Sockets Layer)支持、操作系统细节以及语言等。与 Microsoft的Browser Capability组件一样，也可以添加自己的内容。这里还有一个非常有用的具有向导风格特性，能帮助我们新的浏览器类型添加到定义文件中，如图6-27所示。

BrowserHawk组件具有辨认出新的浏览器(采用一项特殊的模式匹配算法)以及那些缺省支持的一系列浏览器的能力，包括Opera和WebTV。使用过滤功能添加新属性也很容易。

当然，BrowserHawk组件的最大优点可能是定期对来自Cyscape公司的Web网站的浏览器定义文件新版本进行自动检测，同时把一些内容吸收合并到当前使用的定义文件里，这意味着没有必要不断地将定义文件拷贝到所有的Web服务器中。

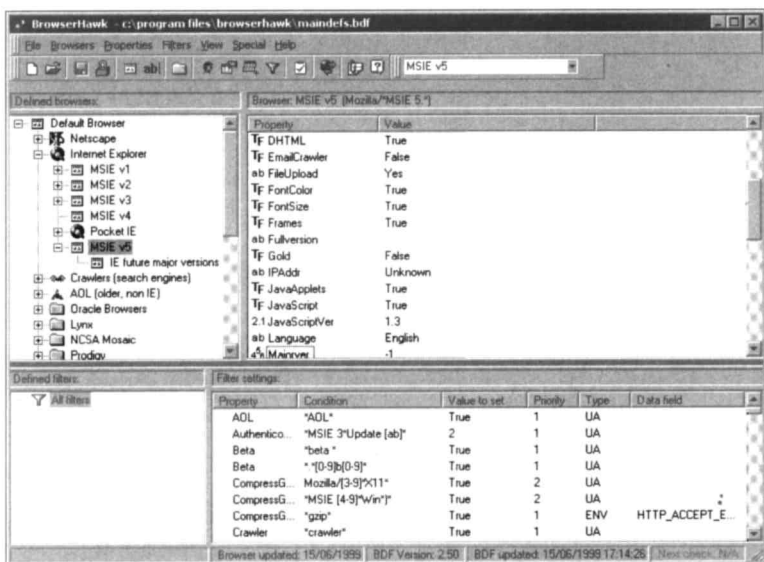


图6-26 BrowserHawk组件

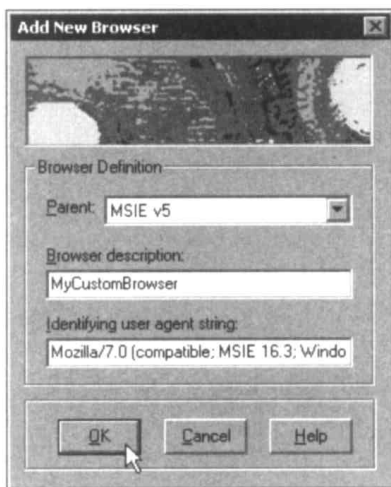


图6-27 向定义文件添加新浏览器的向导

BrowserHawk组件使用起来比较容易，与 Browser Capability组件是兼容的，需要做的是在创建组件实例的 `Server.CreateObject`方法或 `<OBJECT>`元素中改变 `ProgID`。把“`MSWC.BrowserType`”改为“`cyScape.browserObj`”。

```
Set objBCap = Server.CreateObject("cyScape.browserObj")
```

对可能出现的问题，应该阅读 BrowserHawk组件的文档。

可从Cyscape公司的Web网站(<http://www.cyscape.com>)下载BrowserHawk的拷贝。本章中已经包含了使用 BrowserHawk组件的示例网页，使读者能够了解如何使用 BrowserHawk组件，在 Chapter06目录下的 BrowserHawk子目录中可查找到这个示例页面。

6.3.2 SA-FileUp组件

在很多Web网站上，一个常规要求是，服务器接受客户机发送的文件的能力如何？在 IIS 4.0里，Microsoft提供了一个名为Posting Acceptor组件完成这项工作，IIS 5.0和ASP 3.0的标准安装中没有该组件，如果安装Site Server Express的话，则需要包含这一组件。可以从Microsoft Web网站http://www.microsoft.com/windows/software/webpost/post_accept.htm上下载该组件。

然而，很多人发现 Posting Acceptor组件在功能上有局限性且很难编程。现在一个流行的替代品是来自Software Artisans的SA-FileUp组件。与Posting Acceptor组件(仅是ISAPI型过滤器)不同，SA-FileUp组件是一个真正的ActiveX DLL服务器组件，容易集成到ASP网页中。这意味着可以使用别的ASP页面或组件上使用的相同的有关设定页面安全保护及限制用户访问的技术。该组件也可在MTS/COM+的环境中上运行。

使用SA-FileUp组件的教程可在<http://www.activeserverpages.com/upload/>站点上找到，并在这里只给出了结果。本章的示例中已经包含了使用该组件的示例页面(在Chapter06目录下的SAFileUp子目录中)，可用来了解如何使用该组件，有必要的可从 Software Artisans 网站<http://www.softartisans.com/>上下载SA-FileUp的拷贝。

使用SA-FileUp组件

SA-FileUp组件的示例页面创建了一个简单的页面，包含一个 FileUpload元素和一个按钮，并使用了HTML的INPUT元素，如图6-28所示。

```
<FORM ACTION="safileup_response.asp"
      ENCTYPE="MULTIPART/FORM-DATA" METHOD="POST">
...
<INPUT TYPE="FILE" NAME="FILE1">
...
<INPUT TYPE="SUBMIT" NAME="SUB1" VALUE="Upload File">
...
</FORM>
```

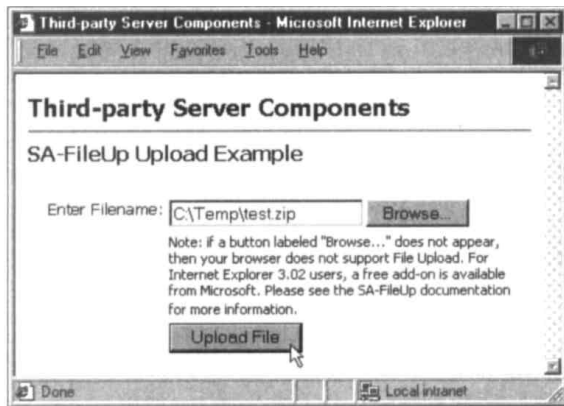


图6-28 使用SA-FileUp组件的示例页面

用户提交窗体时，其中的内容传送给服务器，并且 ASP页面创建了一个组件实例，并开始向服务器上传信息，以后组件返回一系列描述上传的文件的数据，如图6-29所示。

SA-FileUp组件读取文件并把它以标准的磁盘文件或二进制的数据库数据的形式暂存在服务器上。有了它，就能用ASP控制上传过程、限制上传文件的大小，甚至删除上传文件，也

能动态地改变其名字和上传文件的位置。

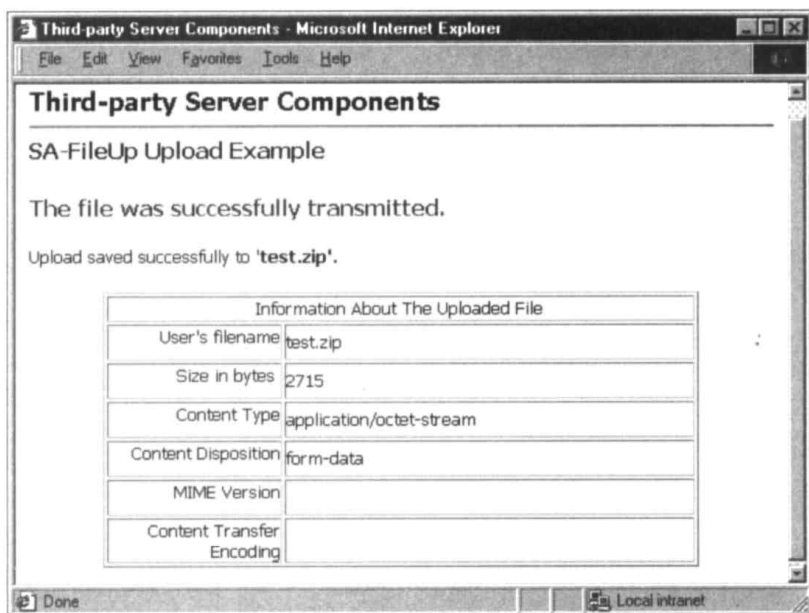


图6-29 使用SA-FileUp组件的结果

6.3.3 RegEx 注册表访问组件

Microsoft IIS 4.0提供的一个示例组件能够访问服务器上的系统注册表。这个功能有助于检查诸如安装的软件的版本，或者检查 IIS和操作系统使用的各种参数。

在IIS 5.0中取消了这个组件，但如果需要这个功能，可以从其他商用版本中获得。RegEx组件是其中之一，能够访问本地机器注册表的任何部分（假如允许当前用户访问）。在ASP页面的匿名访问中，当前用户是IUSR_machinename帐户。

1. RegEx组件的成员

RegEx组件提供了四个读写注册表的方法和一个访问内部组件版本信息的方法，如表 6-11所示。全部信息和文档随 RegEx组件提供，其测试版本可以从 <http://www.stonebroom.com/> 下载。其附带的示例页面包含在本书提供的示例中，然而你需要自己下载组件本身。

表6-11 RegEx 组件方法及说明

方 法	说 明
GetRegValue(RegistryHive, SubKeyString)	从注册表中返回给定的子键的缺省值，如果其值不能访问的话，则返回一个空字符串
SetRegValue(RegistryHive, SubKeyString, NewValue)	在给定的子键中，创建或更新“缺省”值，把数据类型设置成 Reg_SZ(字符串)。如果更新成功的话，返回 True；否则，返回 False
GetRegValueEx(RegistryHive, SubKeyString, ValueName, ValueType)	从特定的子键中返回被命名的值，用返回数值的数据类型，更新方法的最后一个参数所指定的变量。如果数值不能访问的话，则返回一个空字符串，并且 REG_ERROR作为数据类型

(续)

方 法	说 明
SetRegValueEx(RegistryHive, SubKeyString, ValueName, NewValue, ValueType)	在给定的子键中，创建或更新被命名的数值，把数据类型设置为方法的最后一个参数指定的类型 data_type。如果更新成功的话，返回 True；否则，返回 False
GetInternalVersionNumber()	返回组件的内部版本号，如果是一个测试版本，返回值之中也包含这些信息

2. 使用RegEx组件

RegEx组件提供了一个用 Visual Basic编写的示例程序，可用来试验其中包含的方法，实例程序使用RegEx组件，允许对系统注册表的任何部分以所支持的数据类型之一或以缺省的字符串数据类型读写数据，如图 6-30所示。在进行任何更新之前会显示一个警示以防止意外操作。

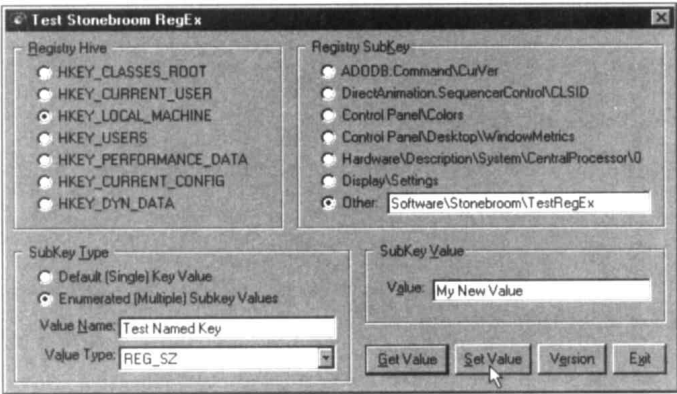


图6-30 使用RegEx组件的示例程序

另外还有一个比较简单的 ASP实例，可显示 Web服务器注册表中的一些数据，并且允许读和设置一个“测试”子键。可以修改这个页面，在你自己的应用程序中使用时无限制。图 6-31所示的是在 NT4中 IIS 4.0上运行的结果。

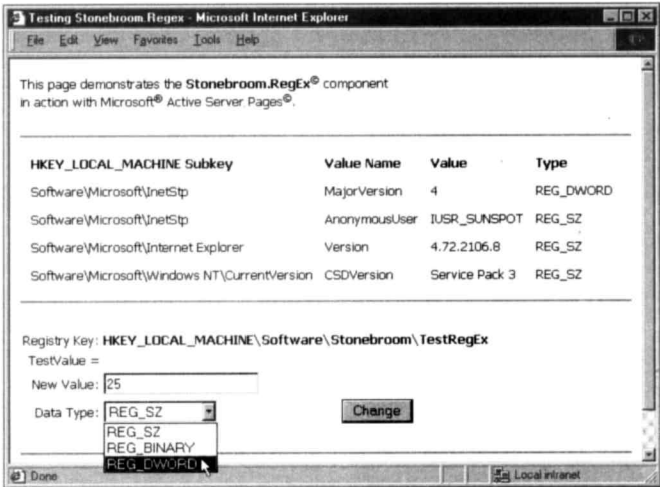


图6-31 使用RegEx组件的另一示例的结果

要记住，改变注册表中的数值可能导致系统不正常地工作，编辑注册表内容之前应该备份注册表文件。

6.4 小结

本章讨论了ASP如何利用服务器组件增加其他组件所不能提供的特性，也考虑了服务器组件和其他类型的ASP对象(如ASP内置的对象和在脚本运行期库中的对象)的区别。

在研究由ASP 3.0和IIS 5.0提供的可安装组件之后，介绍了三个提供常规要求的功能的商用组件，也列出了几个可以从Web上下载的通用组件，其中有一些是免费的或者是共享/测试版本。

本章主要研究了以下内容：

- 服务器组件和其他类型的ASP对象的区别。
- 概要说明如何实例化服务器组件并得到其引用。
- 由ASP 3.0和IIS 4.0提供的可安装组件，如Ad Rotator、Content Linking、Page Counter和Tools。
- 一些可获得的第三方组件。

然而，至今我们还有一个很重要的组件没有涉及到，它就是ADO。在讨论这个问题之前，先来讨论另一个基本问题：错误处理和程序调试。