

# Windows 2000 中 ASP

## 实践两例

刘开南 武颖 (华北矿业高等专科学校电教中心 101601)

### 1 ASP 内置对象的升级

IIS 5.0 中的 Active Server Page 新版本包含了一些对 ASP 内置对象的实质性的改变。Active Server Page 新版本的方法可以更好地控制程序执行、错误处理并提高了 Session 和 Application 对象的性能。

在 IIS 5.0 以前, 你如何显示两个不同的 Active Server Pages 之一呢?

有两种办法: 其一用 Server 对象的 Redirect 方法, 把用户引导至另一个 ASP 页面。其二可以用 #INCLUDE 指令显示包含的两个 Active Server Pages 之一。

存在的问题。① Response 对象的 Redirect 方法把一个 HTTP 头传回网络浏览器, 控制浏览器请求另一个页面。所以浏览器必须请求两次才能到达一个页面, 获取页面的速度很慢。特别是, 有些浏览器会忽略自动取得新页面的指令, 而显示 Object Has Moved。② 用 # INCLUDE 指令把多个 ASP 放在一个 ASP 中, IIS 会在处理任何脚本以前插入所有被包含的页面。这样 ASP 会占去宝贵的内存。

IIS 5.0 提供了更好的页面定向解决办法。也是两种方法。其一使用 Server 对象新的 Transfer 方法引导至服务器而不是浏览器上的一个页面。其二用 Server 对象新的 Execute 方法, 在一个 ASP 中执行另一个 ASP 的脚本。下面进行详细的介绍。

#### 1.1 ASP 页面定向的方法

Server 对象的 Transfer 方法与 Redirect 方法的作用一样, 但没有 Redirect 的缺点。Transfer 的方法是把一个用户引导到另一个网页, 而不是命令浏览器请求另一个新 ASP 页面。作用都发生在网络服务器上, 从而提高了页面访问的速度。

例如, 进入聊天室前, 要求用户输入其用户名和密码。下面是一个 HTML 表单的例子, 提出上述要求:

```
<html>
<head><title>Form</title></head>
<body>
<form method="post" action="result.asp">
```

```
<br> 输入用户名:
<input name="username">
<br> 输入用户密码:
<input name="password">
<input type="submit" value="Okay">
</form>
</body>
```

```
</html>
```

这个表单被传递到一个名为 result.asp 的 ASP 页面。在 result.asp 中可以判断是否提交了用户名和密码。如果空白, 可以自动把用户引导到原来的页面, 重新输入。代码如下:

```
<%
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
if username = "" or password = "" then
Server.Transfer"form.asp"
end if
%>
<html>
<head><title>Thanks</title></head><body>
谢谢! 您的名字、密码正确, 正在进入请稍候.....
</body>
</html>
```

如果提交的用户名和密码是空白的, Server 对象的 Transfer 方法把用户带回原来的 HTML 表单。当调用 Transfer 方法时, 停止当前脚本的执行, 不再显示 ASP 的其余部分, 而显示新页面。Transfer 方法另一个显著的特点是在页面之间转换时, Request 集的内容不会丢失。即当表单信息被提交给 result.asp 页面, 用户被带回 form.asp 页面时, 用户原来提交的信息仍保存在页面 form.asp 中。并非 Response.Transfer 完全代替 Response.Redirect 方法, 有些情况用 Response.Redirect 方法比 Response.Transfer 更好一些。用 Response.Transfer 方法不能把用户引导到另

一个网站的网页上,只能是同一个网络服务器上的另一个页面。而且,不能用Transfer方法把用户带入另一个ASP应用程序。可以进入另一个ASP应用程序中存在的页面,但不能访问第二个应用程序中的应用程序变量。

### 1.2 执行ASP的方法

把脚本的执行从一个ASP转换到另一个上。可以用Server对象的Execute方法在一个ASP中执行另一个ASP中的脚本。

考虑如下名为my1.asp的ASP:

```
<!-- my1.asp -->
<html>
<head><title>Messages</title></head>
<body>
Executing another script:
<% Server.Execute "my2.asp" %>
<br>完成!
</body>
</html>
```

这个ASP执行第二个名为Active2.asp的ASP。执行my2.asp脚本时,把一条信息向浏览器窗口输出5次。执行完my2.asp后,显示my1.asp中的剩余脚本。如图1所示。

```
<!--my2.asp -->
<%
for i = 1 to 5 Response.Write "<br>你好!这是在执行my2!"
next
%>
```

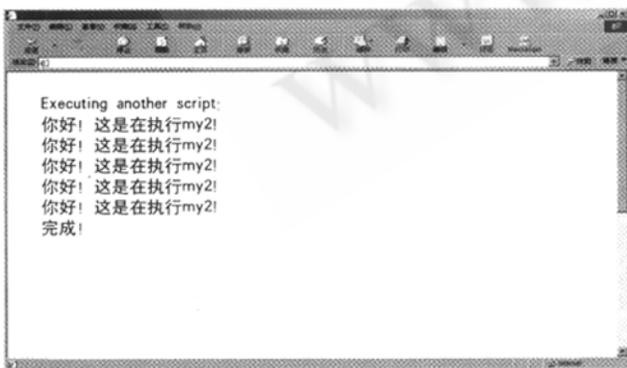


图 1

可以看出,脚本变量并没有在两个脚本之间传送,如果在my1.asp中一个变量被赋了一个值,则该变量的值不

能从my2.asp中获取。如果要在两个脚本之间传送一个变量,就必须使用应用程序变量或会话变量,或是在一个查询字符串中传送此值。下面的脚本把一条信息从outer.asp传送到inner.asp:

```
<!--outer.asp -->
<html>
<head><title>Messages</title></head>
<body>
Executing another script:
<%
Application("message") = "你好!"
Server.Execute "inner.asp" %>
<br>完成!
</body>
</html>
```

调用名为inner.asp的脚本时,传送应用程序变量中的信息"你好!".执行此脚本时,正确显示应用程序变量的值。

```
<!--inner.asp-->
<%=Application("message")%>
```

与以前的IIS不同,IIS 5.0中有两种方法可用把应用程序变量从内存中删除。可以用Application对象的Remove方法输出一个应用程序变量,或是用RemoveAll方法把所有的都删除。尽可能的减少了占用的内存。

## 2 ASP 的 Logging Utility 新组件

IIS 5.0中的ASP组件的性能已经显著提高,除了性能的升级,对组件只有很少的改动外。IIS加入了一个新组件Logging Utility。

在ASP脚本中使用Logging Utility组件访问网络服务器的日志文件。生成关于网站使用情况的定制统计数字,提高网站性能等是非常有用的。例如,你可以用该组件取得并显示某一天一个横幅广告、联结、新闻等被查看的次数。也可以获取关于网站上运行得最慢的ASP的统计数字。还可以收集网站被访问最频繁的页面的信息等功能。

Logging Utility组件在使用上应注意以下两点:首先,它只能处理ASP中的4种日志文件格式,而不能处理ODBC记录(记录到数据库)和第三方组件开发的定制日志文件格式。其次,必须在网络服务器上得到确认,才能使用Logging Utility组件。不能在匿名的用户帐户下用该组件访问ASP。■