

Active Server Pages

动态网页 设计精粹

张兴渊等 编著

兵器工业出版社

Active Server Pages

动态网页设计精粹

张兴渊 殷 镖 编著
门清毅 李 谦

兵器工业出版社

内容简介

本书介绍了目前最流行的在 Internet 上创建网站的技术——Active Server Pages(ASP)。现在很多成功的商业网站都是用 Active Server Pages 创建的,它的功能越来越强大,并被许多公司认可。本书共分八章和 4 个附录来介绍这一技术在开发 Web 应用程序上的使用。第一章从 ASP 基础入手,介绍了 ASP 的组成和运行环境配置,第二章讲述了 ASP 的脚本语言 VBScript,第三章深入地讲述了怎样与用户进行数据交换,第四章介绍了 ASP 组件的使用,第五章和第六章重点讲解了利用 ADO 来实现对数据库的访问以及怎样在 InterDev 中开发 Web 应用程序,第七章描述了用 VB 开发 ASP 组件的方法,最后一章讨论了怎样保护 ASP 程序的安全等内容。本书既可以作为高级开发人员的参考手册,也是一般开发人员的好帮手,对于初学者更是一本不可多得入门指南。

图书在版编目(CIP)数据

Active Server Pages 动态网页设计精粹 / 张兴渊等
编著. -北京:兵器工业出版社, 2000.6
ISBN 7-80132-806-X

I. V… II. 张… III. 主页-设计-应用软件 Active Server Pages
IV. TP393. 092

中国版本图书馆 CIP 数据核字(2000)第 62144 号

75482 / 13

责任编辑:宋丽华

责任技编:李秋生

社址:100089 北京市海淀区车道沟 10 号

经销:各地新华书店

印刷:北京市密云县印刷厂

版次:2000 年 6 月第 1 版第 1 次印刷

印数:1~2000

封面设计:学坤

责任校对:李小舟

责任印制:王京华

开本:787×1092 1/16

印张:27.625

字数:687 千字

定价:39.00 元

(版权所有 翻印必究 印装有误 负责调换)

目 录

第1章	ASP 基础	1
1.1	ASP 概述	1
1.1.1	ASP 的特点	2
1.1.2	ASP 的强大效益	2
1.2	ASP 的组成	3
1.2.1	ASP 内置对象	4
1.2.2	ASP 文件结构	5
1.2.3	ASP 的基本语法	5
1.2.4	ASP 的脚本编程语言	6
1.2.5	创建可重用代码 SSI (Server Side Include)	8
1.3	ASP 的运行环境和配置	9
1.3.1	运行 ASP 的需求	9
1.4	Internet Information Server 介绍	17
1.4.1	选择开发平台	17
1.4.2	开始创建 Web 站点	18
1.4.3	ASP 与其他技术的比较	22
第2章	ASP 的脚本语言—VBScript	26
2.1	VBScript 的编程基础	26
2.1.1	什么是 VBScript	26
2.1.2	VBScript 的历史	28
2.1.3	扩充网络客户及服务器的功能	28
2.1.4	VBScript 的变量数据类型	37
2.1.5	使用变量	39
2.2	使用运算符	43
2.2.1	算术运算符	43
2.2.2	关系运算符	48
2.2.3	逻辑运算符	51
2.2.4	连接运算符	56
2.2.5	运算符优先级	57
2.3	VBScript 控制程序流程	58
2.3.1	使用判定结构	58
2.3.2	实现循环逻辑	64
2.3.3	程序流程中的中断退出	69
2.3.4	使用 Err 对象进行错误排除	72
2.4	使用函数、子程序和对话框	76
2.4.1	事件驱动程序简介	77

2.4.2	子程序和函数	79
2.1.3	使用对话框与用户交流	84
2.5	数据类型转换	90
2.5.1	标识数据类型	90
2.5.2	格式化函数	95
2.5.3	数据类型转换	99
第3章	与用户进行数据交互	104
3.1	向用户发送数据	104
3.1.1	如何结合 VBScript 和 HTML	104
3.1.2	缓冲输出	109
3.1.3	设置内容类型	111
3.1.4	重定向用户	114
3.1.5	格式化本地特定的数据	114
3.2	检索用户信息	116
3.2.1	检索环境变量	117
3.2.2	创建和发送 web 表单	120
3.2.3	验证数据的有效性	127
3.3	管理用户会话和应用程序	129
3.3.1	Web 应用程序	130
3.3.2	用户会话管理	133
3.3.3	使用 global.asa 指定事件脚本	135
3.4	永久保存用户信息	142
3.4.1	cookie 基本知识	142
3.4.2	设置 cookie	144
3.4.3	检索 cookie	147
3.4.4	检查客户的 cookie 启用状态	148
3.5	内部 ASP 对象的高级用法	153
3.5.1	编码字符串	153
3.5.2	代码页和字符集	159
3.5.3	映射服务器路径	161
3.5.4	限制脚本超时	162
3.5.5	高级标题	163
第4章	ASP 组件	165
4.1	使用组件	165
4.1.1	创建组件实例	165
4.1.2	使用字典对象	171
4.2	使用 IIS 的可安装组件	176
4.2.1	自动链接内容页	176

4.2.2	检测浏览器的能力	180
4.2.3	给站点添加横幅	182
4.3	发送和接收电子函件	186
4.3.1	SMTP 服务	186
4.3.2	发送电子函件	190
4.3.3	检索电子函件	196
第5章	利用 ADO 来实现对数据库的访问	200
5.1	ADO 编程基础	200
5.1.1	ADO 的原理	200
5.1.2	用 ADO 进行数据库编程概述	202
5.1.3	SQL 语言简介	203
5.2	开始使用 ADO	215
5.2.1	用开放式数据库互联 ODBC	215
5.2.2	创建一个系统 DSN	216
5.2.3	ADOVBS. INC 文件	218
5.3	Connection 对象	225
5.3.1	创建 Connection 对象	226
5.3.2	Connection 对象的方法	226
5.3.3	Connection 对象的属性	231
5.3.4	获取 DSN 服务器的详细信息	233
5.3.5	Errors 集合和 Error 对象	235
5.4	Command 对象	236
5.4.1	创建 Command 对象	236
5.4.2	Command 对象的属性	237
5.4.3	Command 对象的方法	238
5.4.4	Parameters 集合	241
5.4.5	Parameter 对象	242
5.5	Recordset 对象	243
5.5.1	Recordset 的简单介绍	243
5.5.2	RecordSet 对象的属性	246
5.5.3	RecordSet 对象的方法	253
5.5.4	Recordset 的应用	258
5.5.5	Fields 集合	264
5.5.6	Field 对象	265
5.5.7	利用 GetString 函数来优化数据库编程	267
第6章	用 InterDev 开发 Web 应用程序	268
6.1	可视化数据库工具	268
6.1.1	Visual Database 工具简介	268

6.1.2	使用数据库工程	272
6.1.3	用 Visual InterDev 管理远程数据库	275
6.1.4	创建表	277
6.1.5	编辑数据库对象	283
6.1.6	处理其他数据库对象	285
6.2	使用数据库绑定控件	289
6.2.1	设计时控件和新的数据环境	290
6.2.2	使用 Recordset 控件	295
6.2.3	使用 Grid 控件	300
6.2.4	与其他控件一起使用 RecordsetNavBar 控件	304
6.2.5	其他数据绑定设计时 ActiveX 控件	306
6.2.6	Visual InterDev 6 附带的其他设计时控件	307
6.3	在对 Web 应用程序中显示数据	308
6.3.1	选择数据访问提供者	308
6.3.2	ActiveX 数据对象与数据环境对象模型	312
6.3.3	直接编码与 DTC 和 ActiveX 控件	315
6.3.4	服务器端对客户端数据绑定	318
6.3.5	在 HTML 中嵌入数据	318
6.3.6	直接访问数据库	323
6.3.7	使用 SQL Server 存储过程	323
第7章	用 Visual Basic 开发 ASP 组件	328
7.1	建立自己的组件的原因	328
7.1.1	决定建立组件的因素	328
7.1.2	使用组件	330
7.2	使用 Visual Basic 开发 ASP 控件	330
7.2.1	进程内与进程外	331
7.2.2	在 Visual Basic 中设定组件选项	331
7.2.3	开发进程的快速指南	333
7.3	深入组件开发	335
7.3.1	使用 Task List Viewer 和 Task Killing Utility	338
7.3.2	组件开发循环回顾	338
7.4	在组件中访问数据库	339
7.4.1	剖析组件 ASP 的相互作用	340
7.4.2	使用错误检测特性	341
7.4.3	一个更复杂的例子	342
7.5	在远程 Web 服务器上注册组件	345
7.6	使用 Microsoft Transaction Server	346
7.6.1	生活在事务世界中	347
7.6.2	移植已有组件到 MTS 中	347

7.7	Visual C++和 Visual J++控件创建	351
7.7.1	Visual C++	351
7.7.2	Visual J++	351
第8章	保护 ASP 应用程序安全	352
8.1	IIS 安全过程	352
8.1.1	IP 地址限制	352
8.1.2	用户身份验证	355
8.1.3	IIS 目录权限	356
8.1.4	NTFS 目录和文件权限	357
8.2	用 SSL 保护连接安全	358
8.2.1	生成证书签名请求	359
8.2.2	应用测试证书	362
8.2.3	安装证书	362
附录		364
附录 A	VBScript 2.0 参考	364
附录 B	Active Serve Pages 对象	394
附录 C	ADO 参考	416
附录 D	Active Server 组件参考	428
参考文献		435

第1章 ASP 基础

1.1 ASP 概述

Active Server Pages(ASP)是 Microsoft 推出的一种服务器端命令执行环境,它可以让用户轻松地结合 HTML Web 页面、脚本程序 (Script) 和 ActiveX 组件创建可靠的、功能强大的、与平台无关的 Web 应用系统。正是 ASP 技术能够创建易于维护、与平台无关的应用系统,才可以使你充分利用全球网所提供的广泛资源。

ASP 技术对于 Internet 的发展可以说具有里程碑的意义。在所有 Internet 新技术之中,ASP 是最令人激动的,这是因为在 ASP 之前没有一种可以支持所有浏览器的 Web 页面制作技术。而使用 ASP 技术可以创建为任何浏览器使用的 Web 页面。因此,ASP 技术被称做为“网络聚变的核动力”。

你可能已经开发过许多的网络程序,完成过许多项目,但或许都是利用基于浏览器的脚本语言,如 Javascript 或 VBscript,不能否认它们的功能是很强大的,能让你的网页丰富多彩,活灵活现。但是还是有很多的缺陷,尤其是当面对大量、广泛的用户时,采用客户端脚本语言就有许多弊端。例如,虽然有许多浏览器可以运行 Javascript 或 VBscript,但有些浏览器还是不能认识的。再者,如有的浏览器能利用功能强大的 ActiveX 的优势,如 Microsoft Internet Explorer,但是 Netscape Navigator 目前的版本并不支持此项功能,因此有许多 Netscape Navigator 的用户被迫转向 Microsoft 的 Internet Explorer。

ASP 根据开发人员的需求和浏览器的能力可动态地创建 HTML,消除了某些创建 Web 应用程序的不确定性。通过使用 ASP,不必再担心在客户机一端安装的是什么样的浏览器,数据的计算,页面的生成都由服务器来完成,它能及时准确地确定所输出的内容,准确地与所使用的客户浏览器相适应。

读者一定急于想知道 ASP 到底是如何实现的。通俗地说,ASP 并不是一种语言,它所用的语言仍然是读者朋友们非常熟悉的 Javascript 或 VBscript,或者是它们两者的结合。唯一和通常的 Javascript 或 VBscript 应用程序不同的是它们是在服务器端运行的,而不是在浏览器或者说是客户端执行的。ASP 内含于 IIS3.0 和 4.0 之中,通过 ASP,大家可以结合 HTML 网页、ASP 指令和 ActiveX 控件建立动态、交互且高效的 Web 服务器应用程序。有了 ASP 你就不必担心客户的浏览器是否能运行你所编写的代码,因为所有的程序都将在服务器端执行,包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后,服务器仅将执行的结果返回给客户浏览器,这样也就减轻了客户端浏览器的负担,大大提高了交互的速度。

1.1.1 ASP 的特点

ASP 具有以下所特有的特点:

(1) 使用 VBScript、JScript 等简单易懂的脚本语言, 结合 HTML 代码, 可快速地完成网站的应用程序。

(2) 无须编译, 容易编写, 可在服务器端直接执行。

(3) 使用普通的文本编辑器, 如 Windows 的记事本也可进行编辑设计。

(4) 与浏览器无关 (Browse Independence), 用户端只要使用可执行 HTML 代码的浏览器, 即可浏览 ASP 所设计的网页内容。ASP 所使用的脚本语言 VBScript 和 Jscript 均在 Web 服务器端执行, 用户端的浏览器不需要能够执行这些脚本语言。

(5) ASP 能与任何 ActiveX scripting 语言相容。除了可使用 VBScript 或 JScript 语言来设计外, 通过 plug-in 的方式, 还可使用由第三方所提供的其他脚本语言, 譬如 REXX、Perl、Tcl 等。脚本引擎是处理脚本程序的 COM (Component Object Model) 部件。

(6) ASP 的源程序不会被传到客户浏览器, 因而可以避免所写的源程序被他人剽窃, 从而提高了程序的安全性。

(7) 可使用服务器端的脚本来产生客户端的脚本。

(8) ActiveX Server Components (ActiveX 服务器控件) 具有无限可扩充性。可以使用 Visual Basic、Java、Visual C++ 等编程语言来编写你所需要的 ActiveX Server Component。

1.1.2 ASP 的强大效益

采用 ASP 给系统所带来的强大效益可以归为以下四类:

1. 在前端方面

- 统一标准易用的浏览器界面。
- 用户只需学习简单的 Web 页面操作。
- 大大降低了学习障碍和培训成本。

2. 在 Web 开发者方面

- 用户端不需要特殊设置与软件安装。
- 应用软件集中在服务器端开发管理。
- 减少构建维护成本, 加快联机过程。

3. 系统环境

- 前端可使用任何浏览器(IE、Netscape)。
- 后端可存取任何数据库(SQL、Access、Oracle 等)。
- 可使用任何脚本语言(VBScript、JavaScript、Perl 等)开发。

ASP 应用程序以 .asp 为后缀名的, 存盘后无需编译便能由浏览器查看结果(当然不能在客户端执行, 只能在服务器执行后才能浏览)。ASP 程序的控制部分, 是使用 VBScript、JScript 等脚本语言来设计的, 当执行 ASP 程序时, 脚本程序将一整套命令发送给脚本解释器(即脚本引擎), 由脚本解释器进行翻译并将其转换成服务器所能执行的命令。当然, 同其他编程语言一样, ASP 程序的编写也遵循一定的规则, 如果你想使用你所喜爱的脚本语言编写 ASP 程序, 那么你的服务器上必须要有能解释这种脚本语言的脚本解释器。当你安装 ASP 时, 系统提供了两种脚本语言: VBScript 和 JScript, 而 VBScript 则被作为系统默认的脚本语言。你也可以根据自己的喜好改变系统默认的脚本语言。

ASP 程序其实是以扩展名为 .asp 的纯文本形式存在于 Web 服务器上的, 你可以用任何文本编辑器打开它, ASP 程序中可以包含纯文本、HTML 标记以及脚本命令。你只需将 .asp 程序放在 Web 服务器的虚拟目录下(该目录必须要有可执行权限), 就可以通过 WWW 的方式访问 ASP 程序了。要学好 ASP 程序的设计, 必须掌握脚本的编写。那么究竟什么是脚本呢? 其实脚本是由一系列的脚本命令所组成的, 如同一般的程序, 脚本可以将一个值赋给一个变量, 可以命令 Web 服务器发送一个值到客户浏览器, 还可以将一系列命令定义成一个过程。要编写脚本, 你必须要熟悉至少一门脚本语言, 如 VBScript。脚本语言是一种介于 HTML 和诸如 Java、Visual Basic、C++等编程语言之间的一种特殊的语言, 尽管它更接近后者, 但它却不具有编程语言复杂、严谨的语法和规则。如前所述 ASP 所提供的脚本运行环境可支持多种脚本语言, 譬如: Vbscript、Perl 等等, 这无疑给 ASP 程序设计者提供了广泛的发挥余地。ASP 的出现使得广大 Web 设计者不必在为客户端浏览器是否支持而担心, 实际上就算你在同一个 .asp 文件中使用不同的脚本语言, 你都无须为此担忧, 因为所有的一切都将在服务器端进行, 客户浏览器得到的只是一个程序执行的结果, 而你只需在 .asp 中声明使用不同的脚本语言即可。

1.2 ASP 的组成

在前一节中, 已经介绍了 ASP 的一些基本情况, 现在将介绍如何处理 ASP 页的更多细节。为此, 笔者提供了一个熟悉的图表, 它描述了 ASP 是如何处理一次请求的。如图 1-1 所示。

ASP 的一个要点是脚本完全在服务器上运行, 这样, 可以保护你的程序不被复制, 并且, 可以由于浏览器不同而带来影响, 这一影响在写客户代码时必须要考虑。整个服

务器端代码都要处理，只有简单的 HTML 或选择生成的任意种类的内容发送到客户。

当 Web 服务器接收到对 xyz.asp 页的请求时，它会调用 ASP 引擎，这样又依次调用适当的脚本编程语言来执行添加到 xyz.asp 的服务器端代码。

ASP 有一个很大的优点，即它是与语言无关的，这意味着程序语言将不限于 Microsoft 提供的标准脚本编程语言，如 VBScript。它可由你选择的脚本编程语言来扩展，只要该脚本语言能支持 ActiveX Scripting 模型即可。

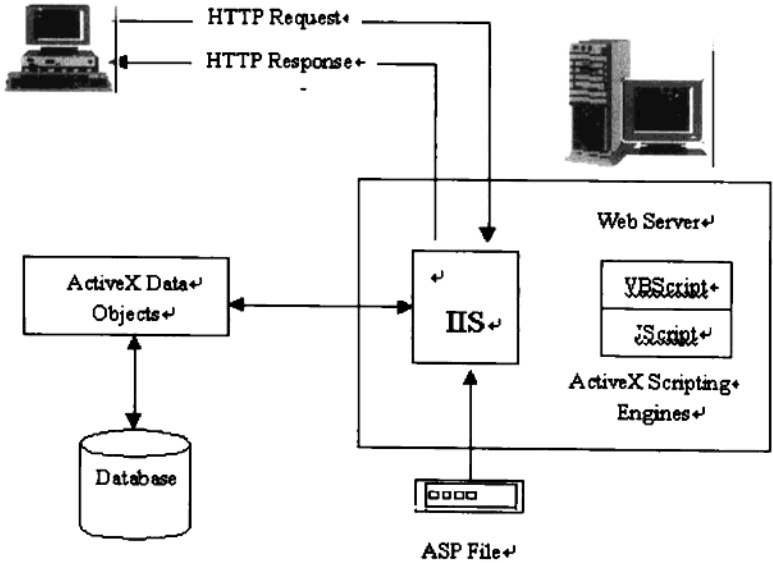


图 1-1 如何处理 ASP 请求

1.2.1 ASP 内置对象

ASP 除了可以创建和管理 Web 应用程序外，还对请求和响应处理提供如下 6 个内置对象：

- Application（应用）：保存程序范围的状态信息。
- Session（会话）：在此对象中维护每个用户个人的信息。
- Request（请求）：包含了从浏览器发送到服务器的所有信息，允许访问用表单发送的数据。
- Response（响应）：将 HTML 和各种其他信息包括 cookies 和报头返回给客户。
- Server（服务器）：为 ASP 中的应用提供服务器功能。
- Object Context（对象上下文）：允许你提交或忽略事务，它由 Transaction Server 管理。

新术语:

会话是访问站点的每个用户的个人存储区。

这些对象在所有的 ASP 页中都可立即使用, 所以使用前无需创建它们。如何访问这些对象呢? 可通过在脚本编程语言编写的 ASP 页中添加代码来实现。

1.2.2 ASP 文件结构

在一个 ASP 文件中可以包含以下内容:

- (1) HTML 标记 (tags)。
- (2) VBScript 或 Jscript 语言的程序码。
- (3) ASP 语法。

ASP 只处理服务器端脚本语言, 对于 ASP 文件中的其他内容, 如 IIS 等支持 ASP 的服务器会将其原封不动地发送到客户端, 由用户端的浏览器进行处理。

目前可以在 ASP 使用的脚本语言是: VBScript 和 JavaScript, 其中系统默认的是 VBScript。你可以同样在 ASP 中很好地利用其他的脚本语言。任何一个可以和 ActiveX Script 标准兼容的脚本语言引擎都可以用于 ASP。

注意: 如果你熟悉 Perl, 你就会发现 Perl 和 ASP 是兼容的。如果希望更多地了解, 请访问 <http://www.hip.activeware.com> 或 <http://www.aspsite.com> 站点。

1.2.3 ASP 的基本语法

要区分 HTML 命令标识与普通字符, 可以在 HTML 命令标识的两端分别加上 “<” 和 “>” 分隔符。

例如:

```
<B>Hello World!</B>
```

上例将用粗体显示字符 “Hello,World!”。

要区分服务器端的 ASP 脚本和其他字符, 可以用 “<%” 和 “%>” 包含 ASP 的命令加以区别。例如用以下命令给变量赋值:

```
<%sport="climbing"%>
```

在以后的大量程序中会看到都是使用这种标记来进行的, 它使在开发时很容易区分一个普通的脚本程序和 ASP 程序。因为在下面读者可以看到还有一种方法也能实现, 就是利用普通的脚本语言标志。

另一个标准的 HTML 的标识 <SCRIPT>, 在编写服务器端的 ASP 脚本可以使用它来

标识服务器端用户定义的函数、方法或模块。但须在<SCRIPT>标识内指定属性 RUNAT=SERVER。如果你创建了一个客户端的 VBScript 或者 Jscript 脚本,你就应当对<SCRIPT>更加熟悉了。微软的<Script>拓展可以被用来指定是在服务端还是在客户端执行。如果说,你忽略了 RUNAT=SERVER 这个限定,你的 Script 脚本将被认为是客户端的。这种场合下,Server 端将忽略这个 Script 并由浏览器尝试执行(然而结果肯定是以失败告终,因为这很可能是一个不合法的客户端 Script 脚本)。

1.2.4 ASP 的脚本编程语言

看一下图 1-1 可发现,在 ASP 页中并不仅限于使用 VBScript。除此之外,还可使用 Jscript。但通过 IIS 的设计,你还可以使用 Perl、Rexx 或许多其他语言的脚本编程实现。

注意:

在 ASP 页中可用多种不同的脚本编程语言,但本书的所有例子均用 VBScript 编写。

如果想用特定的脚本编程语言编写代码,必须先告诉 ASP 你的选择。下面是 3 种可设置的级别:

- Web 站点级: 改变特定 Web 站点的所有页的默认脚本编程语言。
- 文件级: 定义单页的脚本编程语言。
- 函数级别: 仅对一页中的特定函数设置脚本编程语言。

1.2.4.1 更改 Web 站点级的脚本编程语言

默认情况下每个 Web 站点的脚本编程语言均为 VBScript。如果你打算使用另一个脚本编程语言做大部分工作,就应当更改默认设置。

举例来说,可按下面的步骤将默认设置改为 JScript(注意,该过程只应用于 Windows NT Workstation 上的 IIS 和 PWS)。

(1) 打开 Internet 服务管理器,打开 Hope Web 站点(自己创建的 Web 站点)的 Properties 对话框。切换到 Home Dictionary (主目录)选项卡。

(2) 在 Application Setting (应用程序设置)框架中单选 Configuration 按钮, Application Configuration 对话框被打开。

(3) 切换到 App Option 选项卡, 在 Default ASP Language (默认 ASP 语言) 区域中删除 VBScript, 并输入 JScript (参见图 1-2)。

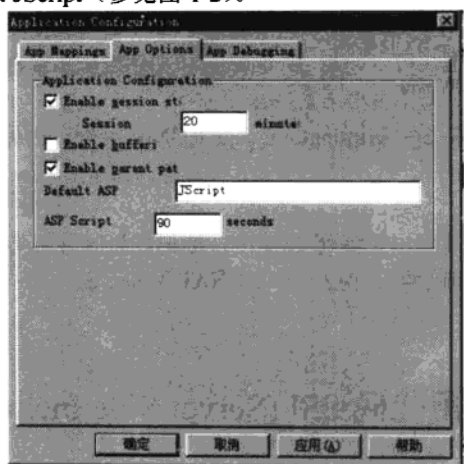


图 1-2 整个 Web 站点更改默认脚本编程语言

(4) 单击“确定”按钮关闭所有对话框, 并使更改生效。

从这时起, 在配置的站点上运行时所有脚本都将默认使用 JScript 引擎。

1.2.4.2 更改文件级的脚本编程语言

在已经对 Web 站点设置了默认的脚本编程语言后, 可能还想用另外的脚本编程语言来编写一些网页。例如整个站点以 VBScript 作为默认的脚本编程语言, 但用 Perl 来做系统级编程, 因为 Perl 特别适合这类编程。

提示:

在文件级上设置脚本编程语言还有一个更令人信服的理由: 要将文件分布到服务器上而服务器并不能保证所用的脚本编程语言为默认设置。

在文件级上设置脚本编程语言的语法如下:

```
<%@Language=ScriptingLanguage%>
```

要使用户这个网页作为默认设置的脚本编程语言的名称来替换 ScriptingLanguage 参数, 这一语句必须作为文件的开头。例如, 设置本页的默认脚本编程语言为 Jscript 的语法如下:

```
<%@Language=JScript%>
```

在本书以后的内容中, 可了解更多有关该技术的细节。

1.2.4.3 更改函数级脚本编程语言

在每个函数的基础上，可以超越 Web 站点和文件级脚本编程语言的设置来设置脚本编程语言。在函数级控制脚本编程语言的语法如下：

```
<SCRIPT LANGUAGE=ScriptingLanguage RUNAT=Server>...
```

```
...
```

```
</SCRIPT>
```

你也许已经从客户端脚本了解到 SCRIPT 标记。为区分服务器端和客户端代码，服务器端<SCRIPT>语句包含了伪指令 RUNAT = Server，应当用所选的语言替换 ScriptLanguage。

1.2.5 创建可重用代码 SSI (Server Side Include)

在设计 Web 站点时，需要查明在整个站点上多个网页基础上建立的元素。这些元素可能是导航栏，或者是底部的一行，其中具有 Web 负责人的电子函件地址。其他元素可能是代码段，以显示日期、标准的金融计算以及许多其他标准的 HTML 或 ASP 代码段，如常量。

金融计算很可能会随时间而改变（导航栏也有可能这样）。然而，要对所有合并的网页进行相应的更改，而不丢失网页，这个问题还有待解决，你必须改变和验证许多文件。

有一个好办法，就是在所有页中用一个包含文件来包含导航栏。如果要更改这些栏，只需更改一次（在包含文件中）。这些改变将自动合并到包含它的所有网页中，从而大大降低了工作量。

如果用这一机制来建立代码库，便会有同样的效果。只要将含有当前页要用函数的文件包含进来，从这个代码库中调用这些函数，就与在当前页中已声明过它们一样。

现在，我们了解了包含方法的优点，可以从理论到实践将它们联系起来。将文件包含进来的一般语句为：

```
<!--#include file|virtual="filename"-->
```

可以将该包含伪指令放在网页的任意位置，但它必须驻留在所有代码块的外部。另外还有一个限制，即不能由脚本更改 filename 参数，因为包含的文件优先于脚本的执行。

1. Virtual 包含

如果在包含伪指令中用了 Virtual 关键字，文件将绝对包含到 Web 站点的根上。将文件包含于脚本的一个例子如下：

```
<!--# include virtual="/includes/myheader.inc" ->
```

这个语句使用与 Web 站点根相关的绝对路径将文件包含到当前文件中，该语句也位于当前文件中。

注意:

这里包含了一个具有.inc 扩展名的文件。如果该文件包含有效的 ASP 代码, 如果用户直接访问该文件将会发生什么情况呢? ASP 代码不会执行, 且将发送给用户, 因为 ASP 引擎在默认情况下不处理.inc 扩展名。因此, 为安全起见, 在包含文件时, 应当一直坚持使用.asp 扩展名。

2. File 包含

file 关键字包含了与包含文件驻留的目录相关的文件。你不用提供完全合乎要求的到服务器根的绝对路径, 提供包含文件的当前页路径的相对路径就足够了。如下所示:

```
<!--#include file "samples / codeo301.asp">
```

该语句将目录 samples 下的文件 codeo301.asp 包含到当前 ASP 页中。

提示:

你也可以使用../语法访问当前目录之上的目录。例如, file 参数../getme.asp 应当包含文件 getme.asp, getme.asp 驻留的目录比有包含伪指令的文件所驻留的目录要高一层。

1.3 ASP 的运行环境和配置

1.3.1 运行 ASP 的需求

ASP 可以在多种平台上使用。在本节提到的所有 Microsoft 平台中, 我们建议至少使用 Microsoft Personal Web Server (PWS) for Windows NT Workstation 平台来开发 Active Server Pages 应用程序, 因为它优于 Windows 98 的 PWS。但是, 若要使用其所有功能, 例如 ASP 文档的全文本搜索, 则应该在 Windows NT Server 上安装 Microsoft Internet Information Server (IIS) 4.0 版。

如果仅有一台小型计算机用于开发工作, 并且没有 Windows NT Server, 则可以用基于 Windows 98 或 Windows NT Workstation 的 PWS 开始学习 ASP。

1.3.1.1 Personal Web Server(PWS)

PWS 适用于需要规模较小的做内部网应用或开发的公司和个人。PWS 不同于 IIS, 它只能容纳一个 Web 站点。PWS 的优点在于, 它可以在廉价的操作系统如 Windows 95、Windows 98 或 Windows NT workstation 上运行。