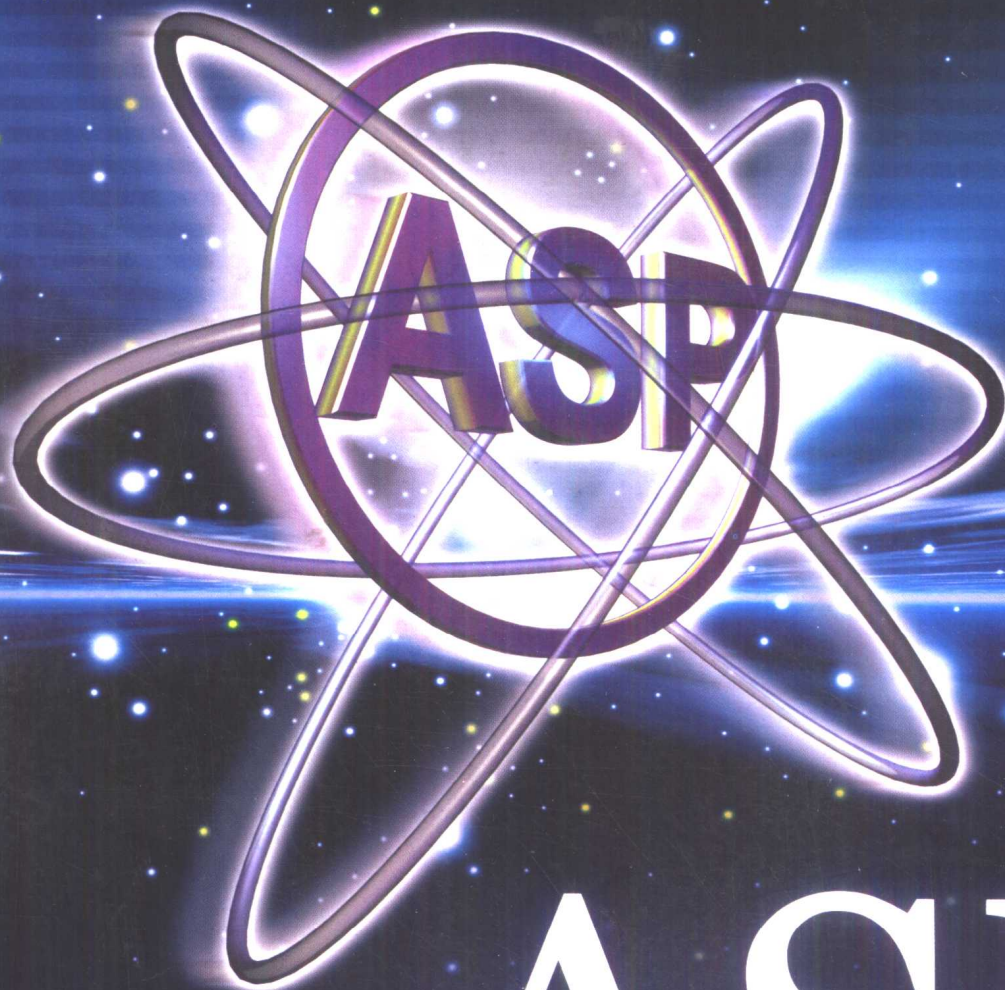




附光盘
CD-ROM



ASP

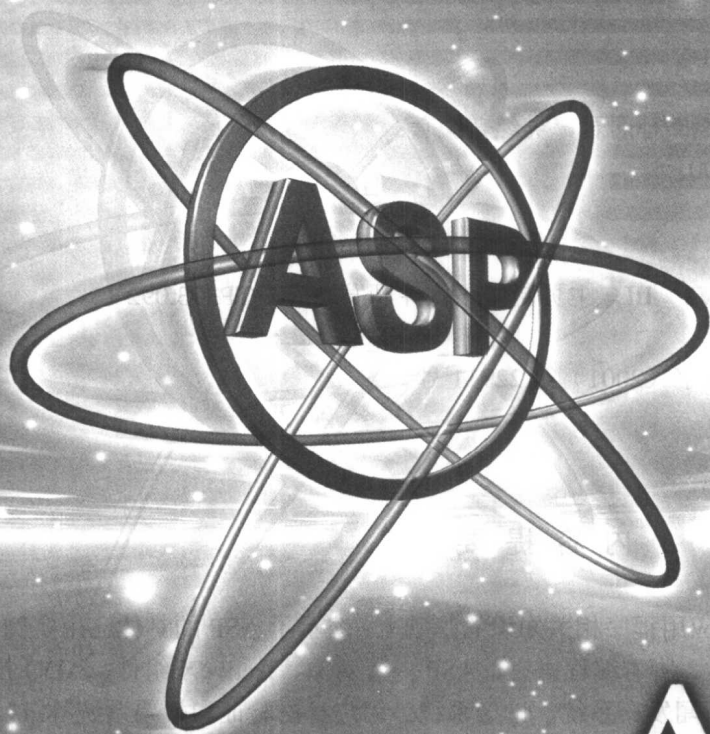
编程基础及 应用实例集锦

■ 孙丽君 吴红普 编著

人民邮电出版社
www.pptph.com.cn

00129845

TP393.4



ASP

编程基础及 应用实例集锦

■ 孙丽君 吴红普 编著



北航 C0548307

人民邮电出版社

J5580/18

图书在版编目 (CIP)

ASP 编程基础及应用实例集锦/孙丽君, 吴红普编著.

北京: 人民邮电出版社. 2001.5

ISBN 7-115-09292-3

I. A... II. ①孙... ②吴... III. 主页制作—软件开发 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2001) 第 22911 号

内 容 提 要

本书在介绍开发 ASP Web 应用程序相关知识的基础上, 演示了 ASP 的具体应用实例。全书分为 11 章, 包括 ASP 概述、脚本编程语言、ASP 内置对象、ActiveX 组件、ADO 技术、访问计数器、邮件列表、新闻发布系统、论坛系统、用户身份验证和投票系统等。

本书着重于将理论与实践相结合, 内容安排上由浅入深、阐述详尽。本书适合利用 ASP 进行 Web 开发的初学者阅读, 也可供具有一定 Web 开发经验的程序员参考。

ASP 编程基础及应用实例集锦

- ◆ 编 著 孙丽君 吴红普
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 21
字数: 504 千字 2001 年 5 月第 1 版
印数: 1-6 000 册 2001 年 5 月北京第 1 次印刷

ISBN 7-115-09292-3/TP·2215

定价: 38.00 元(附光盘)

编 者 的 话

ASP (Active Server Pages) 是目前最流行的 Web 应用程序开发工具之一。它是一种功能强大、灵活易学的服务器端脚本编程环境, 它的源代码均可在服务器端运行, 运行的结果以 HTML 代码的形式输出到客户端。利用 ASP 不仅可以快速创建交互性的动态网页, 而且程序代码完全保密, 更重要的是它不用考虑客户端使用何种浏览器。

内容和特点

本书由理论篇和实例篇两部分组成。理论篇由 ASP 概述、脚本编程语言、ASP 内置对象、ActiveX 组件和 ADO 技术 5 部分构成, 通过这几章的学习, 读者将对 ASP 的强大功能有一个更深刻的认识, 同时为学习后面的实例作铺垫; 实例篇由访问计数器、邮件列表、新闻发布系统、论坛系统、用户身份验证和投票系统 6 个精选的实例构成。这部分内容主要是通过一个个实例来讲述如何利用 ASP 开发 Web 应用程序, 其主要目的是传授编写程序的经验和讲解思考问题的方法。

本书的特点是理论与实践相结合, 在介绍相关的编程理论和知识的前提下, 利用丰富、实用的例子来演示 ASP 的魅力。

读者对象

本书适合于利用 ASP 编制 Web 应用程序的初中级读者。从最基础的编程讲起, 通过提供一些可实际运行的应用实例带领读者迅速入门, 并通过一些完整的实例的设计、分析, 使读者轻松地掌握 ASP 网页开发工具。

配套光盘内容简介

下面是本书配套光盘内容的详细说明。

ASP 文件

书中讲述的各个实例的源文件。可以通过查看 index.txt 文件, 获取每个文件的简单说明和相应源代码文件名。

ASP 辅助软件

主要提供几个 ASP 编辑软件。可以通过查看 index.txt 文件, 获取每个软件的简单说明和相应源代码文件名。

编者
2001 年 4 月

目 录

第 1 章 ASP 概述.....	1
1.1 什么是 ASP.....	1
1.2 ASP 的特点.....	1
1.3 脚本环境.....	2
1.4 在 ASP 中使用脚本语言.....	3
1.5 使用 ASP 指令.....	4
1.6 包含文件.....	5
1.7 Global.asa 文件.....	7
1.7.1 什么是 Global.asa 文件.....	7
1.7.2 应用实例.....	9
1.8 创建 ASP 页.....	13
1.9 创建事务性脚本.....	15
1.9.1 事务的内涵.....	15
1.9.2 声明事务性脚本.....	16
1.9.3 提交或终止事务脚本.....	16
1.9.4 编写事务事件.....	17
1.9.5 登记组件.....	18
1.9.6 事务排队.....	18
1.10 范例.....	19
1.11 ASP 脚本调试.....	22
1.11.1 Microsoft 脚本调试工具.....	22
1.11.2 VBScript Stop 调试器.....	24
1.11.3 用 JavaScript Debugger 调试.....	24
1.12 维护 ASP 应用程序的安全.....	25
第 2 章 脚本编程语言.....	28
2.1 VBScript.....	28
2.1.1 VBScript 简介.....	28
2.1.2 数据类型.....	29
2.1.3 运算符.....	33
2.1.4 控制语句.....	34
2.1.5 过程和函数.....	41
2.1.6 编码约定.....	43
2.1.7 VBScript 与窗体.....	46
2.1.8 在 VBScript 中使用对象.....	49

2.1.9 在 HTML 页面中使用 VBScript 代码	50
2.1.10 VBScript 页面的简单样例	52
2.2 JavaScript	54
2.2.1 JavaScript 的来历	54
2.2.2 JavaScript 的功能特点	54
2.2.3 与 Java 的比较	56
2.2.4 JavaScript 程序运行环境	58
2.2.5 JavaScript 的语法描述	58
2.2.6 JavaScript 编程	61
第 3 章 ASP 内置对象	65
3.1 ASP 内置对象	65
3.1.1 Request 对象	65
3.1.2 Response 对象	71
3.1.3 Application 对象	74
3.1.4 Session 对象	77
3.1.5 Server 对象	79
3.2 会话管理	82
3.2.1 SessionID 和 Cookie	82
3.2.2 启动和结束会话	85
3.2.3 存储数据	85
3.2.4 管理 Web Farm 的会话	86
3.2.5 无会话的 ASP 页	87
3.3 读取 HTML 表格数据	88
3.3.1 HTML 表格	88
3.3.2 用 Request.Form 读取数据	88
3.3.3 用 Request.QueryString 读取数据	90
3.3.4 验证表格输入	91
3.4 向浏览器发送内容	93
3.4.1 发送内容	93
3.4.2 设置内容的类型	94
3.4.3 重定向浏览器	94
3.4.4 内容缓冲	94
3.4.5 代理服务器缓存页	95
3.4.6 浏览器缓存页	96
3.4.7 动态通道	96
3.4.8 向服务器发送文件	97
3.4.9 向浏览器传送脚本	97

第 4 章 ActiveX 组件.....	99
4.1 基本概念	99
4.2 使用 ActiveX 组件	100
4.2.1 Ad Rotator 组件	101
4.2.2 DatabaseAccess 组件	105
4.2.3 ContentLinking 组件	107
4.2.4 BrowserCapabilities 组件	113
4.2.5 FileAccess 组件	118
4.2.6 Content Rotator 组件	122
4.2.7 Page Counter 组件	124
4.2.8 Permission Checker 组件	125
4.2.9 Counters 组件	125
4.3 使用第三方的组件	128
4.4 编制自己的组件	132
4.4.1 创建组件	132
4.4.2 调用组件	135
第 5 章 ADO 技术	136
5.1 ADO 概述	136
5.2 使用 ADO 数据对象	137
5.2.1 创建数据库源名	137
5.2.2 创建数据库连接	140
5.2.3 创建数据对象	141
5.2.4 操作数据库	141
5.2.5 关闭数据对象和链接	141
5.2.6 一个简单的对数据库操作示例	142
5.2.7 循环检索记录	146
5.2.8 优化查询	147
5.3 使用记录集	148
5.3.1 记录集游标和锁定类型	148
5.3.2 用记录集显示记录	150
5.3.3 操作记录集的高级方法	153
5.3.4 遍历记录集	154
5.3.5 返回记录数目	155
5.3.6 把记录取到数组中	156
5.3.7 对记录集进行分页显示	157
5.4 使用命令对象	163
5.5 管理数据库连接	165
5.5.1 连接超时	165
5.5.2 共享连接	166

5.5.3 跨页连接.....	166
5.5.4 关闭连接.....	167
5.6 其他问题.....	167
5.6.1 运用 ADO 进行数据库表数据互导.....	167
5.6.2 指定记录集的最大容量.....	169
5.6.3 使用命令对象调用存储过程.....	170
5.6.4 使用返回状态值.....	171
5.6.5 使用输出参数.....	172
5.6.6 使用输入参数.....	173
5.6.7 取出参数信息.....	175
第 6 章 应用实例一—— 访问计数器.....	177
6.1 简单的计数器.....	177
6.2 与页面分离的计数器.....	179
6.3 与页面分离的图像计数器.....	180
第 7 章 应用实例二—— 邮件列表.....	183
7.1 概述.....	183
7.2 数据库的建立.....	184
7.3 面向用户功能的实现.....	185
7.4 用户注册功能的实现.....	189
7.5 邮件的编辑及发送.....	194
第 8 章 应用实例三—— 新闻发布系统.....	197
8.1 概述.....	197
8.2 数据库的建立.....	198
8.3 主控模块的实现.....	198
8.4 管理模块的实现.....	205
8.5 检索模块的实现.....	211
8.6 新闻代码的使用.....	213
第 9 章 应用实例四—— 论坛系统.....	219
9.1 需求分析.....	219
9.2 数据库的建立.....	219
9.3 注册模块的实现.....	221
9.4 显示模块的实现.....	223
9.5 发表文章模块的实现.....	228
9.6 文章审阅模块的实现.....	230
第 10 章 应用实例五—— 用户身份验证.....	240
10.1 设计目标.....	240

10.2 初始化及输入合法性检查	241
10.3 用户身份验证	243
10.4 在安全页面中检查用户身份	245
10.5 会话终止	247
10.6 完整的代码	249
10.6.1 登录模块的实现	249
10.6.2 退出模块的实现	252
10.6.3 注册模块的实现	254
10.6.4 显示模块的实现	258
10.6.5 默认页面的实现	260
10.6.6 启动和退出功能的实现	262
10.6.7 显示当前用户功能的实现	264
10.6.8 公用函数和过程	265
第 11 章 应用实例六—— 投票系统	272
11.1 系统概述	272
11.2 数据库的建立	274
11.3 创建和编辑投票主题	275
11.4 投票主题的创建和维护	276
11.5 投票表单的生成与结果分析	286
11.6 完整的代码	292
11.6.1 投票系统主页面的实现	292
11.6.2 创建或删除投票主题功能的实现	293
11.6.3 编辑投票主题模块的实现	298
11.6.4 投票表单的实现	312
11.6.5 显示投票结果页面的实现	317

第 1 章 ASP 概述

ASP 是服务器端脚本编程环境。使用 ASP 可以将 HTML 页、脚本命令和 ActiveX 组件组合起来, 创建动态的、交互的 Web 网页和基于 Web 的功能强大的应用程序。本章主要就 ASP 的概念、语法以及 ASP 的编程思路等问题进行论述。

1.1 什么是 ASP

ASP (Active Server Pages) 是微软公司推出的一种用以取代 CGI (Comman Gateway Interface) 的技术。简单讲, ASP 是位于服务器端的脚本运行环境, 通过这种环境, 用户可以创建和运行动态的、交互式的 Web 应用程序, 如交互式的动态网页, 包括使用 HTML 表单收集和处理信息、上传与下载文件等。

ASP 内含于 IIS (Microsoft Internet Information Server) 3.0 和 4.0 之中。有了 ASP, 就不必担心客户端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客户端浏览器, 这样就减轻了客户端浏览器的负担, 大大提高了信息交互速度。

从字面上说, ASP 包含下面 3 方面的含义:

- (1) **Active:** ASP 使用了微软公司的 ActiveX 技术。ActiveX 技术是现在微软公司软件的重要基础。它采用封装对象, 程序调用对象的技术, 简化编程, 加强程序间的合作。ASP 本身就封装了一些常用组件, 很多公司也开发了许多实用组件。只要在服务器上安装这些组件, 通过访问组件, 可以快速、简易地建立自己的 Web 应用程序。
- (2) **Server:** ASP 运行在服务器端。这样就不必担心浏览器是否支持 ASP 所使用的编程语言。ASP 的编程语言可以是 VBScript 和 JavaScript。VBScript 是 Visual Basic 的一个简集, 会 Visual Basic 的人可以很快学会 VBScript。然而 Netscape 浏览器不支持客户端的 VBScript, 所以最好不要在客户端使用 VBScript。而在服务器端, 则无需考虑浏览器的支持问题, Netscape 浏览器也可以正常显示 ASP 页面。
- (3) **Pages:** ASP 返回标准的 HTML 页面, 可以在常用的浏览器中正常显示。浏览者查看页面源文件时, 看到的是 ASP 生成的 HTML 代码, 而不是 ASP 程序代码, 从而防止源程序被抄袭。

1.2 ASP 的特点

作为 IIS 环境下开发 Web 应用的一种简单、方便的编程工具, ASP 具有如下特点:

- (1) 使用 VBScript、JavaScript 等简单易懂的脚本语言，结合 HTML 代码，可快速完成网站应用程序的开发；
- (2) 容易编写，无须编译，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，即可进行编辑；
- (4) 与浏览器无关，只要客户端浏览器能够执行 HTML 代码，就可浏览利用 ASP 设计的网页内容。ASP 所使用的脚本语言均在 Web 服务器端执行，客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 语言相兼容，除了可以使用 VBScript 或 JavaScript 脚本语言来设计外，还可以通过 plug-in 的方式，使用由第三方所提供的其他脚本语言，例如 REXX、Perl、Tcl 等，脚本引擎是处理脚本程序的 COM 组件；
- (6) ASP 的源程序不会被传到客户浏览器，因而可以避免源程序被剽窃，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 服务器组件具有无限可扩充性，可以使用 VisualBasic、Java、Visual C++、COBOL 等编程语言来编写所需的 ActiveX 服务器组件。

1.3 脚本环境

ASP 应用完全是基于 IIS 之上的，用于 Windows NT Server 和 WorkStation 的 IIS 版本不同，但其功能几乎完全相同，不同的只是安装过程。一般来说，我们使用的都是基于 NT 服务器上的 IIS 版本。在 NT Server 环境中，发布信息和管理站点的工作一般都是通过 IIS 来完成的。通常，我们在 NT 4.0 版本中运行的是 IIS 2.0 版，但它不具有支持 ASP 的功能。ASP 必须在安装完 IIS 后单独安装。微软公司发布的 ASP 安装包可以从微软公司的网站下载。当 IIS 2.0 中增添了支持 ASP 的功能后，它会自动升级为 3.0 版本。对于 Web 服务器来说，这并没有多大改动，只是单纯地增加了运行 ASP 的能力。

IIS 3.0 中提供 3 种服务：WWW、Gopher、FTP。WWW 服务为客户浏览器提交 Web 页，并允许客户访问 ASP 文件。最新的 IIS 4.0 版本具有更强、更高的 Web 管理机能和安全性，可直接把它安装在服务器中。在 IIS 4.0 中，IIS 的管理界面发生了根本性的改变，以前的 Internet 服务管理器变为 Microsoft 管理控制台（简称 MMC）。

在 NT 4.0 上安装 IIS 4.0 时，要求系统必须已安装了 NT ASP 3 和 Internet Explorer 4.01。

IIS 支持虚拟目录，通过在“服务器属性”对话框中的“目录”标签可以管理虚拟目录。建立虚拟目录对于管理 Web 站点来说具有非常重要的意义。首先，虚拟目录隐藏着有关站点目录结构的重要信息。因为在浏览器中，客户通过选择“查看源代码”，很容易就能获取到页面的文件路径信息，如果在 Web 页中使用物理路径，将暴露有关站点目录的重要信息，这容易导致系统受到攻击。其次，只要两台计算机具有相同的虚拟目录，就可以在不对页面代码做任何改动的情况下，将 Web 页面从一台机器上移到另一台机器上。最后，将 Web 页面放置于虚拟目录下，就可以对目录属性进行设置了，如 Read（读访问）、Excute（执行访问）、

Script（脚本访问）。读访问表示将目录内容从 IIS 传递到浏览器。而执行访问则可以在该目录内执行可执行文件。在需要使用 ASP 时，就必须将存放 ASP 文件的目录设置为“Execute（执行）”。比较好的做法是在设置 Web 站点时，将 HTML 文件同 ASP 文件分开放置在不同的目录下，然后将 HTML 子目录设置为“读”，将 ASP 子目录设置为“执行”，这样不仅方便了对 Web 站点的管理，而且能提高 ASP 程序的安全性，防止源程序的内容被客户访问。

与一般的程序不同，ASP 程序无须编译，ASP 程序的控制部分是使用 VBScript、JavaScript 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，如果想使用自己喜爱的脚本语言编写 ASP 程序，那么服务器上必须要有能解释这种脚本语言的脚本解释器。安装 ASP 时，系统提供两种脚本语言：VBScript 和 JavaScript，其中，VBScript 是系统默认的脚本语言。用户可根据自己的喜好改变系统默认的脚本语言。

ASP 本身并不是一种脚本语言，它只是提供一种使嵌在 HTML 页面中的脚本程序得以运行的环境。

ASP 程序是存在于 Web 服务器上的扩展名为 .asp 的纯文本文件，可以使用任何文本编辑器对它进行编辑，其中可以包含纯文本、HTML 标记以及脚本命令。只需将 ASP 程序放在 Web 服务器的虚拟目录下，用户就可以通过 WWW 的方式访问 ASP 程序。但是，该目录必须具有可执行权限。

1.4 在 ASP 中使用脚本语言

在服务器端使用脚本语言是 ASP 的特色，那么什么是脚本语言呢？它和 HTML 以及编程语言又有什么不同呢？

脚本语言介于 HTML 和 Java、C++ 以及 VisualBasic 等编程语言之间。HTML 通常用于格式化和链接文本，而编程语言通常用于向机器发出一系列复杂的指令。脚本语言介于两者之间，它的函数与编程语言更为相似。脚本语言与编程语言之间最大的区别是后者的语法和规则更为严格和复杂。

在服务器端使用脚本语言，需要在服务器端安装脚本引擎。脚本引擎用于处理脚本的 COM（组件对象模型）对象。ASP 为脚本引擎提供主机环境，并把 ASP 文件中的脚本交给脚本引擎处理。对于 ASP 文件中使用的每种脚本语言，都要将它们相应的脚本引擎安装在 Web 服务器上。ASP 的默认脚本语言是 VBScript，在 ASP 安装完毕时，它就已经存在于系统中了，所以不必安装 VBScript 的脚本引擎。同样的，若要使用 JavaScript，也不必安装 JavaScript 的脚本引擎。如果要使用不太常用的脚本语言，就需要安装相应的脚本引擎。

ASP 使 Web 开发者可以用各种脚本语言编写程序，在一个 ASP 文件中可以使用多种脚本语言，只要在每段脚本程序开始处用 HTML 标签标明所用的脚本语言即可。

前面讲过，VBScript 是 ASP 默认的主脚本语言。主脚本语言用在定界符<%和%>之内，在定界符之内可以使用任何有效的 VBScript 命令，ASP 会按 VBScript 的要求处理这些命令。同时，ASP 允许将任意一种脚本语言设为主脚本语言，可以将某一种脚本语言设置为 Web 服务器中所有页面的主脚本语言，也可以逐个页面地进行设置。要在某一页中将一种脚本语言设为主脚本语言，需在 ASP 文件开始处添加如下命令：

```
<%@LANGUAGE=ScriptingLanguage%>
```

其中，ScriptingLanguage 是被设置为主脚本语言的语言。若要将某一种脚本语言设为 Web 服务器中所有页的主脚本语言，需要利用功能强大的注册编辑器 regedit.exe，ASP 注册参数的注册路径如下：

```
HKEY_LOCAL_MACHINE\SYSTEM
\CurrentControlSet
\Services
\W3SVC
\ASP
\Parameters
```

将注册表项 DefaultScriptLanguage 的默认值“VBScript”改为要设置的脚本语言即可实现主脚本语言的设置。例如，若想将 JavaScript 设为主脚本语言，则只需要将注册表项 DefaultScriptLanguage 的值改为“JavaScript”即可。在同一个 ASP 文件中混合使用不同的脚本语言是 ASP 的特色之一。这样，就可以利用不同的脚本语言编写不同的脚本程序来完成不同的工作。

脚本程序实际上就是一组执行特定任务的脚本命令。如果经常需要执行某一组特定任务，可以将之定义为程序，以便重复调用。程序要按设置的脚本语言的规则写在标签<SCRIPT>和</SCRIPT>之间。当然，如果用主脚本语言编写程序的话，写在定界符<%和%>之间也可以。如果程序只在某一页内被调用，把它放在这一页内即可；如果在多页中都要调用某些程序，应把这些程序放在一个单独的 ASP 文件中，然后在需要调用这些程序的 ASP 页中将这个 ASP 文件包含进来即可。在 ASP 页中包含其他文件的具体方法请参见 1.7 节。

要调用主脚本语言是 VBScript 的程序，只需在关键字 Call 后加上程序名即可。如果要调用的程序中含有参数，则需要把参数放在括号内。如果要省略关键字 Call，则也要省略括号。

1.5 使用 ASP 指令

ASP 除提供使用脚本语言的指令外，还提供以下一些输出指令和处理指令。

1. 显示表达式值的输出指令

语法格式如下：

```
<%= expression %>
```

这个输出指令等同于使用 `Response.Write` 显示信息。

2. 处理指令

语法格式如下：

```
<%@ keyword %>
```

它主要是为 ASP 提供处理 ASP 文件所需的信息。例如，下面的指令可将 VBScript 设为该页的主脚本语言：

```
<%@ LANGUAGE=VBScript %>
```

处理指令必须出现在 ASP 文件的第 1 行。不要将处理指令放在包含 `#include` 语句的文件中。必须在“@”标记和关键字之间加入一个空格。处理指令中可用的关键字有：

- (1) LANGUAGE 关键字 设置页面的脚本语言；
- (2) CODEPAGE 关键字 设置页面的代码页（字符编码）；
- (3) LCID 关键字 设置页面的现场标识符；
- (4) TRANSACTION 关键字 指定将在事务处理环境下运行的页面；
- (5) ENABLESESSIONSTATE 关键字 指定 ASP 页是否使用会话状态。

可以在单个指令中包含多个关键字，但关键字/值对之间必须由空格分开。不要在等号(=)左右加入空格。例如，下面的脚本同时设置了脚本语言和代码页：

```
<%@ LANGUAGE=JavaScript CODEPAGE=932 %>
```

1.6 包含文件

服务器端包容（SSI）指令为用户提供在 Web 服务器处理之前，将一个文件的内容插入到另一个文件的方法。ASP 只使用这一机制的 `#include` 指令。要在 ASP 文件中插入另一个文件的语法格式如下：

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

其中：

- (1) filename 是被包含的文件；
- (2) virtual 表示要包含的文件在一个虚拟目录内，该目录为 Web 共享目录；
- (3) file 表示要包含的文件在一个当前文件的相关目录内。

假设有两个 Web 共享目录 Dir1 和 Dir2，目录 Dir1 内有文件 file11.asp 和 file12.asp，目录 Dir2 下有文件 file2.asp，若要在文件 file11.asp 中包含 file2.asp，则将如下脚本加入到文件 file11.asp 中：

```
<!--INCLUDE VIRTUAL="Dir2/file2.asp"-->
```

若要在 file11.asp 中包含 file12.asp，则将如下脚本加入到文件 file11.asp 中：


```
<!--INCLUDE FILE="file12.asp"-->
```

或者使用如下脚本:

```
<!--INCLUDE VIRTUAL="Dir1/file12.asp"-->
```

被包含的文件不要求有专门的文件扩展名,但要为被包含文件赋予.inc 扩展名,以便和其他类型文件相区分,这也是一个良好的编程习惯。

1. 使用 Virtual 关键字

使用 Virtual 关键字可以指出路径开始于虚拟目录。例如,如果一个被命名为 Footer.inc 的文件属于一个名为/Myapp 的虚拟目录,则下面的语句,将把 Footer.inc 的内容插入到包含该语句的文件中:

```
<!--#include virtual ="/myapp/footer.inc"-->
```

2. 使用 File 关键字

使用 File 关键字指示使用的是相对路径。相对路径开始于含有该包含文件的目录。例如,如果文件位于目录/Myapp 中,而文件 Header1.inc 在 Myapp\Headers 中,下面的语句可将 Header1.inc 插入到文件中:

```
<!--#include file ="headers/header1.inc"-->
```

注意:被包含文件的路径 Headers/header1.inc 是相对于包含文件的;如果包含#include 语句的脚本不在/Myapp 目录中,那么该语句将不起作用。

如果 Internet 服务管理器中的“启用上层目录”选项被选中,则可以使用 File 关键字和../ 语法包含父目录(即上一层目录)中的文件。

3. 被包含文件的位置

被包含文件可以在 Web 站点内的某个目录中,也可以在 Web 站点之外。通常情况下,应使被包含文件位于 Web 站点的目录中。如果一个被包含文件位于 Web 站点内,则当浏览器下次请求包含文件时,将显示对被包含文件的改变信息。但是,如果被包含文件位于 Web 站点之外,直到 ASP 应用程序或 Web 服务器重新启动时,被包含文件的改变才能体现出来。

4. 包含文件提示和警告

一个被包含文件也可以包含其他文件。如果#include 指令没有导致循环,一个 ASP 文件可以多次包括同一个文件。例如,如果文件 First.asp 包含文件 Second.inc,则 Second.inc 一定不能再包含 First.asp。一个文件也不能包含自身。ASP 能够检测此类循环或嵌套错误,并在检测到错误时产生一个错误消息,同时停止处理被请求的 ASP 文件。

ASP 在执行脚本命令之前执行包含文件。因此,不能用脚本命令去创建被包含文件的名称。例如,下面的脚本将不能打开 Header1.inc,因为 ASP 会在给变量 name 分配一个文件名之前先执行#include 指令。

```
<!-- This script will fail -->
```

```
<% name=(header1 & ".inc") %>
```

```
<!--#include file="<%= name %>"-->
```

脚本命令和过程必须完全包含在脚本定界符<%和%>、HTML 标记<SCRIPT>和</SCRIPT>或<OBJECT>和</OBJECT>中。也就是说,不可在一个 ASP 包含文件中打开一个脚本定界符之后,在一个被包含的文件中才关闭该定界符。脚本或脚本命令必须是一个完整的单元,例如,下面的脚本将无法运行:

```
<!-- This script will fail -->

<%

For i = 1 To n

statements in main file

<!--#include file="header1.inc" -->

Next

%>
```

下面的脚本可正常运行:

```
<%

For i = 1 to n

statements in main file

%>

<!--#include file="header1.inc" -->

<% Next %>
```

1.7 Global.asa 文件

1.7.1 什么是 Global.asa 文件

Global.asa 文件是一个可选文件,程序编写者可以在该文件中指定事件脚本,并声明对象具有会话和应用程序作用域。每个应用程序只能有一个 Global.asa 文件。该文件的名称必须是 Global.asa,而且必须存放在应用程序的根目录中。

在 Global.asa 文件中,如果包含的脚本没有使用<SCRIPT>标记封装、定义的对象没有会话或应用程序作用域,则服务器将返回错误。使用任何支持脚本的语言都可以编写 Global.asa 文件中包含的脚本。如果多个事件使用同一种脚本语言,可以将它们封装在一组<SCRIPT>标记中。

在 Global.asa 文件中,声明的过程只能从与 Application_OnStart、Application_OnEnd、Session_OnStart 和 Session_OnEnd 事件相关的一个或多个脚本中调用。在基于 ASP 应用程序的 ASP 页中,这些脚本是不可用的。如果要在应用程序之间共享过程,可在单独的文件中声明这些过程,然后使用服务器端包容语句将该文件包含在调用该过程的 ASP 程序中。通常,

包含文件的扩展名应为.inc。

一个标准的 Global.asa 文件的内容如下所示：

```
<SCRIPT LANGUAGE="VBScript"RUNAT="Server">
'Session_OnStart 当客户首次运行 ASP 应用程序中的任何一个页面时运行
'Session_OnEnd 当一个客户的会话超时或退出应用程序时运行
'Application_OnStart 当任何客户首次访问该应用程序的首页时运行
'Application_OnEnd 当该站点的 Web 服务器关闭时运行
</SCRIPT>

<SCRIPT LANGUAGE="VBScript"RUNAT="Server">

SubApplication_OnStart

VisitorCountFilename=Server.MapPath("/ex2")+"\\VisitCount.txt"

SetFileObject=Server.CreateObject("Scripting.FileSystemObject")

SetOut=FileObject.OpenTextFile(VisitorCountFilename,1,FALSE,FALSE)

Application("visitors")=Out.ReadLine

Application("VisitorCountFilename")=VisitorCountFilename

EndSub

SUBApplication_OnEnd

SetFileOutObject=Server.CreateObject("Scripting.FileSystemObject")

SetOut=FileOutObject.CreateTextFile(Application("VisitorCountFilename"),TRUE,FALSE)

Out.WriteLine(application("visitors"))

EndSub

SubSession_OnStart

Session.Timeout=5

Application("visitors")=Application("visitors")+1

Session("ID")=Session.SessionID

EndSub

</SCRIPT>
```

在这个 Global.asa 程序中，涉及到了 ASP 的 FileAccess 组件，它可以提供用于访问文件系统的方法、属性和集合。关于 ASP 的组件的详细内容请参考第 4 章。在这里，它的作用是在服务器上创建新文件并对文件进行写操作。其实这是一个 ASP 页面访问计数器应用程序的 Global 文件。首先，当客户首次访问该应用程序的首页时，过程 Application_OnStart 定义在指定的服务器虚拟目录下新建一个 VisitCount.txt 文本文件，并将文件的路径和内容保存在应