



万水计算机编程技术与应用系列



ASP

编程技巧与 实例分析

介绍了ASP技术的背景资料和工作平台

介绍了ASP的基本知识及控件的使用技巧和方法

介绍了HTML语言、VBScript语言和JScript语言

通过实例讲解了ASP的

ASP

Active Server Pages

苏 鹏 邵谦谦 等编著



中国水利水电出版社
www.waterpub.com.cn

ASP 网络编程案例解密与解密案例



ASP

编程技巧与 实例分析



ASP

ASP 网络编程案例解密与解密案例

ASP 网络编程案例解密与解密案例

ASP 网络编程案例解密与解密案例

万水计算机编程技术与应用系列

ASP 编程技巧与实例分析

苏鹏 邵谦谦 等编著

中国水利水电出版社

内 容 提 要

ASP 是由软件界的巨头、著名的微软公司开发的新一代网络程序开发工具，一直以来受到计算机编程爱好者和各机构专业人士的广泛青睐。为满足广大读者的愿望，迅速掌握 Web 环境下的 ASP 编程方法，作者编写了本书。书中融合了利用 ASP 开发 Web 下应用程序的理论和实践，全面和深入地介绍了利用 ASP 开发网络应用程序的常用方法和技巧，并附带了很多程序开发实例，所以实用性很强。

全书共分为八章，主要包括 ASP 中的基本部件和 ActiveX 控件的用途和使用方法，文件的打开与操作，以及 ASP 中引人入胜的网络与数据库开发应用。

本书资料丰富、内容翔实，具有易学易用、简洁明了的特点，适合于对 Web 编程有一定基础的软件开发人员使用，也可以作为广大计算机爱好者的参考读物。

书 名	ASP 编程技巧与实例分析
作 者	苏鹏 邵谦谦 等编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号 100044） 网址：www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）68359286（万水）、63202266（总机）、68331835（发行部）
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 33.5 印张 741 千字
版 次	2001 年 6 月第一版 2001 年 6 月北京第一次印刷
印 数	0001—5000 册
定 价	65.00 元（1CD，含配套书）

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

前 言

随着计算机技术的发展, 计算机软件产业也随之迅速地崛起, 网络程序设计成为近几年以来的热点。早期的 CGI (通用网关接口) 程序通常是应用 C, Perl 等语言开发, 这种开发本身难度较大, 而且编码的可读性不是很强, 令广大爱好者望而却步。

微软公司推出了 ASP 技术, 这个技术使这一切都变得相当简单了。ASP 的基础语言为 VBScript 程序设计语言, 这种语言本身就是非常简单的描述性语言, 简单易学, 同时它继承了 ActiveX 技术的特点, 而且借助于微软 COM 技术帮助, 使得 ASP 本身显示出了异常强大的功能。ASP 技术早已超越了一般描述性语言的功能。

由于有微软的软件基础, 如今 ASP 程序已经获得了广泛的技术支持, 如数据库的体系结构、ActiveX 控件的使用、Web 应用程序和安全线程等, 从而使得 ASP 的功能更加强大, 使用起来也更加方便和灵活, 提高了应用程序的开发效率。总之, ASP 是开发 Web 程序最为强大的工具之一, 无论是 Web 程序设计的初学者, 还是有经验的 Web 程序员, 利用 ASP 都可以迅速地开发出自己满意的应用程序。

为了满足广大读者的愿望, 迅速地掌握 Web 环境下的 ASP 编程方法, 作者编写了本书, 并把本书分为九部分。各部分的结构安排如下所示:

- ◇ 第一章主要介绍 ASP 技术的背景资料和发展前景
- ◇ 第二章介绍 ASP 运行环境的建立
- ◇ 第三章介绍 HTML 语言
- ◇ 第四章介绍 VBScript 语言和 JScript 语言
- ◇ 第五章介绍 ASP 的基本对象和使用
- ◇ 第六章介绍 ASP 与 ActiveX 的连接
- ◇ 第七章通过多个具体而典型的示例程序讲解 ASP 的一些控件的综合使用技巧和使用方法
- ◇ 第八章给出了一个完整的 ASP 站点实例, 希望读者可以通过示例更好地理解 ASP 的使用
- ◇ 本书最后为附录, 分别给出了 ASP 的常用部件, 建设网站的策略和 SQL 语言的简明介绍

本书注重开发实例、开发经验、开发技巧和 Web 高级特性开发, 适合于各个层次的 ASP 用户。对初学者来说, 可迅速加入 ASP 高级用户的行列; 对有一定使用经验的读者, 也可通过本书掌握 ASP 深层次的开发方法, 学会用更巧妙的办法开发出高水平的 ASP 应用。

书中融合了利用 ASP 开发 Web 下应用程序的理论和实践, 全面和深入地介绍了利用 ASP

开发应用程序的常用方法和技巧。在本书的编著过程中，我们尽量注意减少冗长无味的说明，代之以具体实用的例题演示。通过例题，引导读者把握 ASP 的精髓所在。

参加本书编写的除了苏鹏、邵谦谦外，还有刘荣华、张勉、李军、朱自强、赵华刚、朱峰、李光龙、许小荣、张浩、郝春荣、祁云峰、潘世军、朱红昌、刘阳、王建明、孙明、李大宇、姜海、赵斌、王征、王剑、柏崇剑、李进军、戴丽华、张颖、魏云、肖光、马良宇、邵丹、朱丽华、孙赵林、王大印、付东国、裴世聪等。

限于作者水平，书中的错误和不足之处在所难免，竭诚欢迎广大读者对本书提出批评和建议。

最后，书中示例中含有大量的程序代码、例子名称、图形图片、数据信息等，内容如有雷同，纯属巧合。

编者

2001 年 3 月

目 录

前言

第一章 ASP 技术基础	1
1.1 ASP 技术概述.....	1
1.2 ASP 技术的基本对象和方法.....	10
第二章 ASP 的工作平台	13
2.1 Windows NT 的安装.....	13
2.2 管理工具介绍.....	21
2.3 安装 IIS 服务器.....	24
2.4 IIS 的后台操作和设置.....	33
2.5 SQL 的安装和配置.....	45
2.6 建立数据库的连接.....	49
2.7 了解 SQL Server 的安全模式.....	53
2.8 设置用户组、用户和权限.....	56
2.9 SQL Security Manager.....	57
2.10 维护 SQL Server 数据库.....	58
第三章 HTML 基本讲解	62
3.1 什么是 HTML 语言.....	62
3.2 HTML 基本知识讲解.....	66
3.3 更深入的学习使用 HTML.....	88
3.4 HTML 设计的高级技巧.....	121
3.5 小结.....	154
第四章 ASP 语法讲解	155
4.1 VBScript 概述.....	155
4.2 在 Active Server Pages 中使用 VBScript.....	156
4.3 总结.....	198
4.4 介绍 JScript.....	198
4.5 在 Active Server Pages 中使用 JScript.....	199
4.6 小结.....	233
第五章 Active Server Pages 的基本对象	234
5.1 Request 和 Response 对象.....	234

5.2	利用缓存输出	235
5.3	处理运行时间长的脚本以及内容大的主页	238
5.4	处理主页的 Header	240
5.5	对 Header 的授权操作	244
5.6	REQUEST 对象的应用	247
5.7	Redirect 方法介绍	257
5.8	包含其他文件	258
5.9	Session 的介绍	260
5.10	Session 的使用和处理	261
5.11	Cookie	267
5.12	不利用 Cookie 来保存信息	272
5.13	什么是 Application	275
5.14	使用 Application 对象	276
5.15	全部显示 Application 对象	279
5.16	Application 事件	279
5.17	聊天室实例	281
5.18	谁在站上的应用	284
5.19	小结	288
第六章	ActiveX 对象的讲解	289
6.1	在 Active Server Pages 中使用组件	289
6.2	浏览器能力检测组件	292
6.3	使用浏览器能力检测组件的一个实例	297
6.4	简介对文件操作的组件	299
6.5	在一个文件中读写数据	299
6.6	对文件进行处理	305
6.7	对文件夹和驱动器进行操作	310
6.8	一个远程硬盘管理的程序实例	314
6.9	Ad Rotator 组件	319
6.10	Ad Rotator 对象的属性	323
6.11	统计访问者	326
6.12	广告控件的实例	328
6.13	ActiveX 数据对象概述	330
6.14	使用 ActiveX 数据对象	330
6.15	使用连接对象	334
6.16	数据对象使用实例	339

6.17	使用命令对象	349
6.18	取出参数信息	357
6.19	高级反馈页应用程序实例	358
6.20	创建密码保护系统应用程序实例	363
6.21	小结	368
第七章	综合实例讲解	369
7.1	留言板实例	369
7.2	网上投票的技巧与实现	383
7.3	新闻讨论组实例	390
7.4	用 ASP 发送邮件的实例	396
7.5	邮件列表的实例	398
7.6	小结	407
第八章	一个网上职业介绍的完整实例	408
8.1	ASP Job 站点的简介	408
8.2	首页的编写	410
8.3	New Resume 部分	412
8.4	新工作部分的编写	421
8.5	升级工作部分	423
8.6	用户查找部分	424
8.7	找工作部分	435
8.8	小结	435
附录 1	ASP 对象和组件的快速参考	436
1.1	内置的 Active Server Pages 对象	436
1.2	可安装的 ASP 组件	442
附录 2	建设网站的策略	447
2.1	硬件配置	447
2.2	软件配置	451
2.3	安全配置	451
2.4	域的配置	460
附录 3	简明 SQL 数据库介绍	462
3.1	SQL 介绍	462
3.2	使用 SELECT 语句	463
3.3	创建新表	472
3.4	字段属性	477
3.5	向表中添加数据	479

3.6 删除和修改表	480
3.7 建立索引	481
3.8 SQL 核心语句	484
3.9 集合函数	488
3.10 其他常用的 SQL 表达式、函数和过程	491
3.11 使用 SQL Server Web 助理	503
3.12 小结	514
附录 4 ASP 网站的安全	515

第一章 ASP 技术基础

1.1 ASP 技术概述

1.1.1 Active Server Pages 是什么

Active Server Pages 实际上是将标准的 HTML 文件拓展了一些附加特征。Active Server Pages 像标准的 HTML 文件一样包含 HTML 对象且在一个浏览器上解释并显示。任何可以放在 HTML 中的东西，如 Java Applets、闪烁字符串、客户端脚本、客户端 ActiveX 控件等都可以放在 Active Server Pages 中。因此，Active Server Pages 有如下五个重要特征使之具备很强的通用性。

- Active Server Pages 可以包括服务端脚本，在本书中，可以学会如何利用 VBScript 和 JScript 来创建 Active Server Pages。利用 Active Server Pages 中的服务器端脚本，可以创建动态内容的网页，举一个最简单的例子，可以让网页在一天的不同时间显示不同内容。
- Active Server Pages 提供了一些 Built-in 对象。利用这些 Active Server Pages 中提供的 Built-in 对象，可以使的脚本更加强大；这些对象会允许从浏览器中接受和发送信息。例如，利用 Request 对象，可以将用户在 HTML FORM 中的信息接受到并在一个 Script 中来响应它。
- Active Server Pages 可以用另外的元素来扩展。Active Server Pages 本身就是来源于相当数量的标准服务端 ActiveX 元素，这些组件允许做诸如依据浏览器能力进行不同显示以及在浏览器内包含计数器的的工作。
- 标准的 ActiveX 组件非常有用，不过，并不会局限于此，可以轻松创建属于自己的附加 ActiveX 组件。这意味着可以不加限制地拓展 ActiveX 组件。
- Active Server Pages 可以和诸如 ACCESS 这样的数据库进行挂接，利用一些特别的对象集合，如 Active Data Object (ADO)，就可以在 Active Server Pages 中使用 SQL 语言。

再次重申，ADO 是 Active Server Pages 一个非常强大的功能所在，在线商务以及在线沙龙等各种非常高级的，动态更新的站点都需要数据库的支持，而且需要随数据库内容的更新而自动更新，这样，就可以利用 Active Server Pages 建立很多类似的高级站点。

以上四个特性对 Active Server Pages 进行了限定，Active Server Pages 就是由 Server 端脚

本、对象以及组件拓展过的标准主页。利用它，主页就可以包含动态内容。

本书中对 Active Server Pages 的讲解前提是你使用的是微软的 Internet Information Server (IIS) 作为网站 Server 软件，当然，也可以利用许多其他的 Server，例如，针对于 Win 95, Win 97, Win 98 的 Personal Web Server (PWS) 以及针对于 NT 的 Peer Web Server。甚至微软家族之外的 NetScape Enterprise and FastTrack Server (这可不是浏览器)，Lotus Domino and Go Servers, O'Relly's Website 等等，当然这些需要你使用的是 Chili!ASP，可以参看 www.chilisoft.net

1.1.2 Active Server Pages 可以做什么

在利用 Active Server Pages 工作时几乎没有什么限制，只要网站存在并正常运行，以下是简单的应用示例列表：

- 在网站主页顶部添加一个滚动显示的广告栏
- 从 HTML 的 FORM 中接受信息并且存到数据库中
- 根据不同访问者显示不同内容，创建个性化主页
- 在主页中添加点击计数器
- 根据用户浏览器的版本、类型和能力显示不同的内容
- 连接多个主页使之容易网际导航
- 跟踪用户网站上的活动信息并且存入日志文件

以上只是简单介绍，看完下面章节，会对你所能够利用 Active Server Pages 做的事情有更加全面深入的了解，你会真正学会如何使用 Active Server Pages。

1.1.3 Active Server Pages 是怎样工作的

理解 Active Server Pages 怎样工作的最好方法是将支持它和不支持的 Web Server 进行对比。Microsoft 利用 Internet Information Server(IIS)4.0 版本和 Active Server Pages 结合在一起。在 IIS4.0 中引入了 Active Server Pages 就能够使站点内容从静态变成动态。这是什么意思呢？

在前面对 Active Server Pages 的介绍中，了解到 IIS 的主要功能是支持静态 HTML 主页，当用户向一个使用 IIS 的服务器提出请求后，服务器将会从自己的硬盘或者内存中提出一个静态主页发送到用户的浏览器上，IIS 的主要作用就是作为用户浏览器访问服务器硬盘的一些文件的一个有效的接口。

注意：

早期的 IIS 版本包括一个称做 Internet Database Connector(IDC)的工具，IDC 可以从数据库中接受和存储数据，你仍然可以利用 IDC 做这种工作，但是显然 Active Server Pages 可以做得更好。

从这个意义上来说，IIS 和其他的 Web Server 没有什么区别，任何 Web Server 的主要作用都是支持静态 HTML 主页，理解其对 HTML 的支持过程如下列步骤所示：

(1) 作为一个用户首先是在浏览器的地址栏中填上要访问的主页地址并回车触发这个申请。

(2) 浏览器将申请发送到诸如 IIS 的 Web Server 上。

(3) Web Server 接收这些申请并根据.htm 或.html 的后缀名识别到这是 HTML 文件。

(4) Web Server 从当前硬盘或内存中读取正确的 HTML 文件然后将它送回用户浏览器。

(5) HTML 文件将会被用户的浏览器解释并将结果显示在用户浏览器上。

当然, 有的过程可能会更加复杂一些 (例如, FORM 中的内容以及查询字串的传递)。在大多数情况下, 对于各种典型的 Web Server 来说, 这种过程是按部就班的, Web Server 接收文件申请并从内存或者硬盘中读取正确的文件。

Active Server Pages 改变了这一切, 在 IIS 仍然支持静态 HTML 主页的前提下, 利用 Active Server Pages 可以根据用户要求在 Web Server 上建立新的主页。

为了便于理解它和支持静态主页的区别, 将它分成以下步骤介绍:

(1) 一个用户在浏览器的网址栏中添入 Active Server Pages 文件名称, 并回车触发这个 Active Server Pages 的申请。

(2) 浏览器将这个 Active Server Pages 的请求发送给 IIS。

(3) Web Server 接收这个申请要求并由于其.asp 的后缀识别到这是个 Active Server Pages 请求。

(4) Web Server 从硬盘或者内存中接收正确的 ASP 文件。

(5) Web Server 将这个文件发送到一个叫做 ASP.DLL 特定文件中。

(6) Active Server Pages 文件将会从头至尾被执行并根据命令要求生成响应的静态主页。

(7) HTML 主页将被送回浏览器。

(8) HTML 主页将会被用户浏览器解释执行并显示在用户浏览器上。

注意:

为了清晰地说明问题, 以上步骤已经进行了很大的简化, 一个 Active Server Pages 并不一定每一次都重新编译解释, 如果再次接受以前的那个请求而且没有任何变化, Active Server Pages 会从 Cache 中提出结果而不是再次运行。

对于 Web Server 来说, Active Server Pages 与 HTML 有着本质的区别, HTML 是不经任何处理送回给浏览器, 而 Active Server Pages 的每一个命令都首先被用来生成 HTML 文件, 因此 Active Server Pages 允许生成动态内容。

另一方面, 对于浏览器来说, Active Server Pages 和 HTML 几乎是没有什么区别的, 仅仅是后缀为.asp 和.htm 的区别, 当在客户端提出 Active Server Pages 的申请后, 浏览器接受的是 HTML 格式的文件。因此它适用于任何浏览器。

1.1.4 在 Active Server Pages 中结合 Scripts 脚本

Active Server Pages 主要就是 Scripts 脚本环境, 在学习本书以后的内容时, 就可以将

VBScript 和 JScript 结合到 Active Server Pages 中去,也可以同样在 Active Server Pages 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 Active Server Pages。

注意:

如果你熟悉 Perl,你就会很高兴地发现 Perl 和 Active Server Pages 是兼容的。如果希望更多了解,请看 <http://www.hip.activeware.com> 或访问 <http://www.aspsite.com>。

在 VBScript 和 JScript 的章节中(第四章),会学到怎样将 Script 脚本结合到 Active Server Pages 的具体实现。这部分主要是详细解释以及介绍一些可选择的方法来进行 Scripts 和 Active Server Pages 的结合。

最简单的结合 Scripts 的方法是利用<%和%>,任何在这个符号中包含的都被认为是 Scripts 脚本。下面是个例子:

```
<HTML>
<HEAD><TITLE>ASP Script 示例</TITLE></HEAD>
<BODY>
这是个
<% FOR I=1 TO 10 %>
非常,
<% NEXT %>
非常长的句子。
</BODY>
</HTML>
```

该段 Active Server Pages 执行后在浏览器上显示为:

这是个非常,非常,非常,非常,非常,非常,非常,非常,非常,非常,非常长的句子。

这段脚本利用 VBScript 的 FOR NEXT 循环生成了 10 份“非常”的拷贝。

Active Server Pages 默认为主要使用的是 VBScript,也就是说,在使用<%和%>时不需要作任何事情来说明使用何种语言。不过,如果需要限定的话,可以有三种方式。

首先,可以在 Internet Information Server 管理器中进行限定使得所有 Active Server Pages 的默认语言为你所限定的。这样做的步骤如下:

- (1) 从开始菜单 IIS(Internet Information Server)的程序组中执行网络服务管理器。
- (2) 右键你的网站名称(如果没有改变默认设置,名称将是 Default Web Site)。
- (3) 选择属性项。
- (4) 点击主目录表。
- (5) 点击设置按钮(要这样做,必须有一个已存在的 Application。如果没有的话,点击

Create 来创建一个)。

(6) 在 Application Configuration 的对话框中点击 App Option 表。

(7) 在默认的 ASP 语言文本框中输入所要的主要脚本语言，例如 VBScript 或者 JScript。当这些做完后，<%和%>中包含的程序就默认为所指定的那种语言。

还可以在特定的主页中指定主要脚本语言，这样做，要将这种语言名称直接放在 Active Server Pages 最靠上的一行，如下：

```
<%@ LANGUAGE=Jscript %>
<HTML>
<HEAD>
<TITLE>ASP Script示例</TITLE>
</HEAD>
<BODY>
这是个
<% for(i=1;i<11;i++){ %>
非常，
<%}%>
非常长的句子。
</BODY>
</HTML>
```

这种直接放在首行的脚本，指出这个文件中所有脚本都将利用 JScript 生成，当利用这种方式指定脚本语言时，请注意在@和保留字 LANGUAGE 之间空出一个空格。另外，非常重要的是这句指示要在任何一句命令之前（否则就要报错了）。

第三种在 Active Server Pages 包含 Scripts 脚本的选择是利用微软的 HTML<Script>拓展对象，如下所示：

```
<HTML>
<HEAD>
<TITLE>ASP Script示例</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="Jscript" RUNAT="server">
function sayhello()
{ response.write("hello! 朋友")
```

```
}  
</SCRIPT>  
<% sayhello() %>  
</BODY>  
</HTML>
```

这里, <SCRIPT>对象包含了一个 JScript 函数。<SCRIPT>的 LANGUAGE 属性限定了使用哪一种语言。RUNAT 属性是指出该脚本应当在 Server 端还是在 Client (浏览器上) 端实现。

这个函数在第一行中被命名为 sayhello()。在第二段脚本中, 利用<%和%>标记的对象是真正调用该函数的位置。Active Server Pages 会在显示器上显示出字符串:

“hello! 朋友”

注意:

response.write()语句是将一段文本显示在显示器上。你会在后面的内容中了解更多。如果创建了一个客户端的 VBScript 或者 JScript 脚本, 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。如果忽略了 RUNAT=server 这个限定, Script 脚本将被认为是客户端的。这种情况下, Server 端将忽略这个 Script 并由浏览器尝试执行 (然而会很悲惨地以失败告终, 因为这不是一个合法的客户端 Script 脚本)。

为什么直接使用<SCRIPT>对象而不是<%和%>来限定 Script 脚本语言? 通常不会使用<SCRIPT>对象。不过, 这两种限定 Script 脚本的方法还是有很明显的区别的。

首先, 使用<SCRIPT>包含的脚本会立即执行, 无论在 Active Server Pages 的任何位置。例如下面这个例子:

```
<HTML>  
<HEAD>  
<TITLE>ASP Script 示例</TITLE>  
</HEAD>  
<BODY>  
This is the first sentence.  
<SCRIPT LANGUAGE="JScript" RUNAT="server">  
response.write("This is the second sentence.")  
</SCRIPT>  
</BODY>  
</HTML>
```

看这个例子后，也许会认为，显示器上会首先显示 This is the first sentence，然后是 This is the second sentence，但是，这个 Active Server Pages 显示在浏览器上是相反的顺序（或者什么都不显示，那是因为生成了非法的 HTML 语法）。

为什么会这样？因为主页里面无论什么放在<SCRIPT>里面都会首先被执行。如果使用浏览器的 View Source，会看见主页被 Active Server Pages 解释为如下所示：

```
This is the second sentence.<HTML>
<HEAD><TITLE>ASP Script示例</TITLE></HEAD>
<BODY>
This is the first sentence.
</BODY>
</HTML>
```

<SCRIPT>对象的行为有两种含义。首先，能够在 Active Server Pages 的任何位置上的<SCRIPT>对象中放置 Script 脚本。其次，<SCRIPT>对象，如果用于更多的目的，将会用包含函数和子程序的方法来进行限制。任何不包含函数或者子程序的脚本都将被立即执行并显示为一个非法的 HTML 主页。

<SCRIPT>对象使用<%和%>限制有一个明显的优点。使用<SCRIPT>对象，可以在一个单独的主页中混合使用多种 Script 脚本。看下面这个例子：

```
<%@ LANGUAGE="VBScript"%>
<HTML>
<HEAD><TITLE>ASP Script示例</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript" RUNAT="server">
function sayhello()
{
    response.write("hello! 朋友")
}
</SCRIPT>
<% FOR I=1 TO 10
sayhello()
NEXT
%>
</BODY>
```