

Active 动感网络技术应用系列

ASP+和JSP

网络技术应用



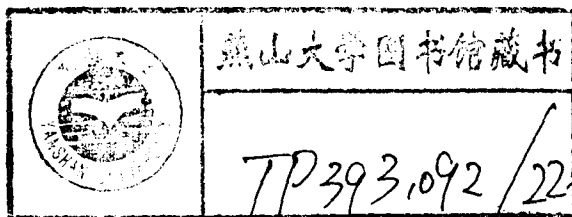
董欣 编著

国防工业出版社

Active 动感网络技术应用系列

ASP+和JSP网络技术应用

董欣 编著



国防工业出版社

·北京·



0350427

~ 31

05
102

图书在版编目(CIP)数据

ASP+ 和 JSP 网络技术应用/董欣编著. —北京:国防工业出版社,2003.1

(Active 动感网络技术应用系列)

ISBN 7-118-02945-9

I.A... II.董... III.①主页制作—应用软件,ASP+
②JAVA语言—主页制作—程序设计 IV.TP393.092

中国版本图书馆 CIP 数据核字(2002)第 069899 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 25 576 千字

2003 年 1 月第 1 版 2003 年 1 月北京第 1 次印刷

印数:1—4000 册 定价:35.00 元

(本书如有印装错误,我社负责调换)

前 言

Microsoft Active Server Pages (ASP, 动态服务器主页)和 Sun Java Server Pages (JSP, Java 服务器主页)是两种脚本式 Web 网页技术,这两种网页可以根据 Web 浏览器的请求显示动态内容。这两项技术都使用 HTML 来决定网页的版面。在生成内容和查询数据库或其他应用时,ASP 依赖于用嵌入式 Microsoft 脚本语言编写的程序,而 JSP 则使用 Java 程序。

在 Internet 上冲浪,阅读具有交互式内容的 Web 网页时,这些网页很可能是利用微软公司的 ASP 技术或 Sun 公司的 JSP 技术制作的。例如,包含天气预报内容的 Web 主页一般根据 Web 浏览器的具体请求提供动态信息。不管用户是否想获得丹佛或西雅图的天气预报,该主页的图形显示不会发生变化,而发生变化的是通常由数据库提取的精确天气信息。

Java Server Pages (JSP)是一种实现普通静态 HTML 和动态 HTML 混合编码的技术,有关 JSP 基础概念的说明请参见《JSP 技术简介》。

许多由 CGI 程序生成的页面大部分仍旧是静态 HTML,动态内容只在页面中有限的几个部分出现。但是包括 Servlet 在内的大多数 CGI 技术及其变种,总是通过程序生成整个页面。JSP 使得我们可以分别创建这两个部分。

近年来,ASP+和 JSP 在我国也得到了广泛的应用,本书主要介绍了这两个技术在网络设计中的应用以及所用到的一些基本的和较深的方法。本书同时还参考了好多关于这两种技术的资料和相关文章,所涉及的范围可能比较大,所以对初学者可能有一定的难度,不过书中仍然讲解了一些比较简单的基础知识,在学了这些知识后,学习以后的内容就比较容易了。

由于时间的仓促和知识的有限,本书难免会有一些错误和不当之处,敬请读者批评指正。

作 者
2002 年 8 月

内 容 简 介

ASP 和 JSP 是两种脚本式 Web 网页技术,可以根据 Web 浏览器的请求显示动态内容。本书主要介绍这两种网页技术在网络设计中的应用以及所用到的一些基本的和较深层次的方法。

本书的作用在于帮助复习者深入理解教材的内容,巩固基本概念,达到培养良好程序设计能力和习惯的目的,进而帮助读者顺利通过各种类型的考试。

书中所选取的习题能够为读者顺利考取计算机专业的研究生提供较大的参考价值。

本书可作为高等院校计算机类或非计算机类的学生使用的教材,也可以作为感兴趣的读者的自学指南,还可以作为其他书籍的参考书。

目 录

第 1 章 ASP+ 和 JSP 简介	1
1.1 ASP 和 JSP 简介	1
1.1.1 ASP 的基础知识	1
1.1.2 Java Servlet 及其特点	2
1.1.3 JSP 及其特点	3
1.2 ASP 和 ASP+	4
1.2.1 ASP+ 的介绍	4
1.2.2 ASP+ 的环境配置	5
1.2.3 ASP+ 服务器端控制(ASP+ Server Control)	7
1.3 JSP、ASP 与 ASP+ 的比较	11
1.3.1 JSP 与 ASP 的比较	11
1.3.2 ASP 和 ASP+ 的区别	12
1.3.3 ASP 与 JSP 的优势	19
第 2 章 ASP+ 的语法	21
2.1 ASP 的语法简介	21
2.2 ASP+ 的基本语法	22
2.3 书写第一个 ASP+ 文件	29
2.4 ASP+ 的服务器端编程初步	31
2.5 ASP+ 的服务器端编程控件	33
2.6 数据库	42
2.7 ASP+ 的服务器端控件语法	45
2.7.1 ASP+ HTML 服务器端控件语法	46
2.7.2 数据捆绑(Databinding)语法 <%# %>	46
2.7.3 HTML 控件	47
2.7.4 Web 控件	49
2.7.5 Data Bind	52
第 3 章 ASP+ 的数据库应用	57
3.1 ASP+ 中的数据库访问技术	57
3.1.1 ADO 概述	57
3.1.2 Connection 对象	57
3.1.3 Recordset 对象	59
3.1.4 Field 对象	61

3.1.5 应用实例	61
3.2 利用 ASP 存取各种常用类型数据库	62
3.2.1 ASP 的对象存取数据库方法	62
3.2.2 链接数据库和打开数据表	63
3.2.3 程序清单	63
3.2.4 实例(存取 Excel 数据表)	66
3.3 关于 Namespace(命名空间)的使用	68
3.3.1 操作数据库的需要	68
3.4 数据库的应用	71
3.5 运用 ASP+ 创建日志文件	74
第 4 章 从 ASP 迁移至 ASP+	76
4.1 最初的考虑(一)	76
4.2 最初的考虑(二)	77
4.3 进入 DataSet	79
4.4 Custom Validator 控件	79
4.5 从用户收集数据	80
4.6 处理会话变量	82
第 5 章 ASP+ 的应用实例	84
5.1 系统概述	84
5.2 系统设计	84
5.2.1 数据库设计	84
5.2.2 登录页面	88
5.2.3 注册页面	90
5.2.4 修改页面	97
5.2.5 密码提示页面	104
5.2.6 链接数据库组件方法	107
5.2.7 操作组件方法	108
5.2.8 配置文件	115
5.3 运行结果	116
第 6 章 JSP 简介和配置环境	118
6.1 JSP 简介	118
6.1.1 JSP 与 ASP 的简单比较	118
6.1.2 运行环境	119
6.1.3 JSP 的开发背景	119
6.1.4 Web 应用开发的 JSP 技术方法	120
6.1.5 JSP 的技术知识	121
6.1.6 JSP 的页面	122
6.1.7 JSP 页面示例	123
6.1.8 JSP 页面应用	128

6.1.9 JSP 技术的未来	129
6.2 JSP 的配置环境	129
6.2.1 安装 Servlet 和 JSP 开发工具	129
6.2.2 安装支持 Servlet 的 Web 服务器	130
6.3 Resin + Apache 的安装	131
6.3.1 Resin 的下载和安装	131
6.3.2 软件的配置	131
6.3.3 测试	132
6.4 Orion App Server 的安装	132
6.4.1 Orion App Server 的下载和安装	132
6.4.2 Orion App Server 系统配置	132
6.4.3 Orion App Server 功能测试	133
6.5 JDBC - ODBC 翻页的例子	133
6.5.1 JDBC - ODBC 运行前准备	133
6.5.2 代码设置	133
6.5.3 程序运行	136
6.6 MySQL 翻页的例子	136
6.6.1 运行前的准备	136
6.6.2 代码设置	136
6.6.3 程序运行	138
6.7 一个分页的 JavaBean 和调用实例	138
6.7.1 运行前的准备	139
6.7.2 代码设置	139
6.7.3 程序运行	145
6.8 产品目录/论坛 树型结构的实现	145
6.8.1 运行前的准备	145
6.8.2 代码设置	146
6.8.3 程序运行	152
6.9 E-mail 发送程序	156
6.9.1 运行前的准备	156
6.9.2 代码设置	156
6.9.3 程序运行	160
6.10 抓取网址的程序	160
6.10.1 运行前的准备	160
6.10.2 代码设置	160
6.10.3 程序运行	164
第 7 章 JSP 语法和基本组件	165
7.1 JSP 介绍	165
7.1.1 概述	165

7.1.2 JSP 语法概要表	165
7.1.3 关于模板文本(静态 HTML)	167
7.1.4 JSP 的语法	167
7.1.5 JSP 的特点	180
7.1.6 JSP 编译器	184
7.2 JSP 的基本组件	185
7.2.1 内置组件	185
7.2.2 Session 对象	185
7.3 JSP 的简单实例	187
7.3.1 日期显示	187
7.3.2 使用 ErrorPage	188
7.3.3 通过 Session 传递选项	190
7.3.4 网页调色板	193
7.3.5 网络备忘录	196
第 8 章 ASP+ 和 JSP 的简单网络技术应用	207
8.1 ASP 和 JSP 的简单应用	207
8.2 Java Bean 原理与应用	209
8.2.1 什么是 JavaBean	209
8.2.2 非可视化的 JavaBean	210
8.2.3 JSP 的网络应用实例	212
8.3 ASP+ 简介及其应用	216
8.3.1 运行 ASP+ 的环境	216
8.3.2 ASP+ 的语法规则	217
8.4 ASP+ 和 JSP 的脚本语言	220
8.5 ASP+ 和 JSP 的脚本编写	223
8.6 ASP+ 和 JSP 的一些基本技巧	229
8.6.1 ASP+ 的运行环境	230
8.6.2 ASP+ 编写的 Web 动态功能	230
8.6.3 JSP 的 HTML 表单	234
8.7 ASP+ 的调试	240
8.8 Servlet 基本结构	241
8.8.1 JSP 与 Servlet 的区别	242
8.8.2 Applet 与 Servlet 通信的四种方法及其比较	243
8.9 Servlet 的简单介绍和示例	248
8.9.1 Servlet 实例演示	248
8.9.2 使用 Servlet 的注意事项	256
第 9 章 ASP+ 和 JSP 的深入网络技术	259
9.1 处理表单数据	259
9.1.1 表单数据概述	259

9.1.2 运用实例	260
9.1.3 JSP 和 JavaBeans 如何对数据库进行操作	264
9.1.4 表单的编辑	267
9.2 读取 HTTP 请求头	267
9.2.1 HTTP 请求头概述	267
9.2.2 在 Servlet 中读取请求头	268
9.2.3 实例分析——输出所有的请求头	269
9.3 ASP+ 内建对象	270
9.3.1 ASP+ 的一些基本概念	270
9.3.2 ASP+ 的内建对象	271
9.4 HTTP 应答状态	283
9.4.1 状态代码概述	283
9.4.2 设置状态代码	284
9.4.3 HTTP 1.1 状态代码及其含义	284
9.4.4 应用实例——访问多个搜索引擎	287
9.5 HTTP 应答头设置	290
9.5.1 HTTP 应答头概述	290
9.5.2 常见应答头及其含义	290
9.5.3 应用实例——内容改变时自动刷新页面	291
9.6 访问 CGI 变量	294
9.6.1 CGI 变量概述	294
9.6.2 标准 CGI 变量的 Servlet 等价表示	294
9.6.3 应用实例——读取 CGI 变量	295
9.7 处理 Cookie	297
9.7.1 Cookie 概述	297
9.7.2 Servlet 的 Cookie API	297
9.7.3 应用实例——定制的搜索引擎界面	299
9.8 会话状态	303
9.8.1 会话状态概述	303
9.8.2 会话状态跟踪 API	303
9.8.3 应用实例——显示会话信息	305
9.9 adduser.java 和 Global.asa 文件	308
9.9.1 adduser.java 文件	308
9.9.2 ASP+ 的内建对象——Global.asa 文件	310
9.10 使用 JSP 和 ASP+ 技术生成动态 XML	316
9.10.1 概述	316
9.10.2 XML 与关系型数据库	316
9.10.3 应用实例——在线像册	317
9.10.4 使用 JSP 技术生成动态 XML	318

9.11 用 JSP 和 ASP+ 技术构建动态网站	324
9.11.1 JavaBean	326
9.11.2 数据库连接	327
9.11.3 技术分析	329
9.12 程序实例	330
第 10 章 ASP+ 和 JSP 的常见问题及编程技巧	341
10.1 ASP+ 常见问题及编程技巧	341
10.1.1 避免使用 DSN 连接数据库	341
10.1.2 记录集中记录总数的计算	341
10.1.3 数据库搜索技巧	343
10.1.4 数值显示格式化	343
10.1.5 SQL 命令中的引号处理	344
10.1.6 处理数据库日期值	344
10.1.7 检查 Cookie 是否启用	345
10.1.8 检查文件是否存在	346
10.1.9 如何更新被锁定的 DLL	346
10.1.10 记录文件下载日志	347
10.1.11 路径问题	347
10.1.12 小数转分数	348
10.1.13 将 VB 中的 Class 结合到 VBScript 中来	349
10.2 JSP 的常见问题	350
10.2.1 在 Windows 98 下安装 JSP 环境	351
10.2.2 在 Windows NT4.0 下安装 Apache + Servlet + JSP	352
10.2.3 JSP 在 Windows 2000 下的安装	354
10.2.4 安装配置 IIS + resin	354
10.2.5 安装配置 WebSphere	355
10.2.6 安装配置 apache + resin	359
10.2.7 同时安装并支持 PHP 和 JSP	360
10.3 JSP 编程技巧	363
10.3.1 关于 Java 中文问题的解释	363
10.3.2 常用基础函数	364
10.3.3 在 JSP 中调用 Servlet	375
10.3.4 JSP 连接数据库大全	379
10.3.5 使用 JSP/Servlet 上载文件	384
10.3.6 跨平台应该注意的问题	386

第 1 章 ASP+和 JSP 简介

1.1 ASP 和 JSP 简介

Microsoft Active Server Pages (ASP, 动态服务器主页)和 Sun Java Server Pages (JSP, Java 服务器主页)是两种脚本式 Web 网页技术,它们可以根据 Web 浏览器的请求显示动态内容。这两项技术都使用 HTML 来决定网页的版面。在生成内容、查询数据库或其他应用时,ASP 依赖于用嵌入式 Microsoft 脚本语言编写的程序,而 JSP 则使用 Java 程序。

在 Internet 上冲浪,阅读具有交互式内容的 Web 网页时,这些网页很可能是利用 Microsoft 公司的 ASP 技术或 Sun 公司的 JSP 技术制作的。例如,包含天气预报内容的 Web 主页一般根据 Web 浏览器的具体请求提供动态信息。不管用户是否想获得美国丹佛或西雅图的天气预报,该主页的图形显示不会发生变化,而发生变化的通常是由数据库提取的精确天气信息。

1.1.1 ASP 的基础知识

Microsoft Active Server Pages (ASP)是一套微软开发的服务器端脚本环境。ASP 内含于 IIS 之中,最新版的 ASP 3.0 内含于 IIS 5.0 之中。通过 ASP 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 就不必担心用户的浏览器是否能运行所编写的代码,因为所有的程序都将在服务器端执行,包括嵌在普通 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用户浏览器,这样就减轻了用户端浏览器的负担,大大提高了交互的速度。

ASP 总结起来有以下几个特点:

(1) 使用 VBScript、JavaScript 等简单易懂的脚本语言,结合 HTML 代码,即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就可编写。为了更好地进行团队开发,微软特别开发一套开发工具 Visual InterDev,它是一个集成的 Web 应用软件开发系统,包括开发、发行以及管理数据库驱动的 Web 应用软件所需的所有功能。

(4) ASP 所使用的脚本 VBScript、JavaScript 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 的源程序不会被传到用户浏览器,因而可以避免所写的程序被他人剽窃,从而提高了网站的安全性。

(6) ActiveX Server Components 具有无限的可扩充性。可以使用 Visual Basic、Java、

Visual C++、CoBol 等编程语言来编写所需要的 ActiveX Server Component。

(7) ASP 程序中可以包含纯文本、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（该目录必须要有可执行的权限），就可以通过 WWW 的方式访问 ASP 程序。

学习 ASP 要掌握以下基础知识：

(1) 学会 MicroSoft Visual InterDev 软件的使用。

(2) ASP 服务器的安装和使用。ASP 内含于 IIS，如 Windows NT 4.0 的 IIS 4.0 含有 ASP2.0，Windows 2000 的 IIS 5.0 含有 ASP 3.0。如果是普通 Windows 98 用户也没关系，微软也开发了一个专为学习 ASP 的服务器 Personal Web Server，它一般包含在 Windows 98 正式版的 add-ins/pws 目录下，只要装了 Personal Web Server，Windows 98 用户照样可以学习 ASP 编程。

(3) 熟练掌握 HTTP 和 HTML。

(4) 熟练掌握 VBScript 或 JavaScript。

(5) 熟练掌握数据库知识，如 Microsoft SQL Server、ADO 和 ODBC；熟悉 Transact-SQL 语言。

1.1.2 Java Servlet 及其特点

Servlet 是 Java 技术对 CGI 编程的回答。Servlet 程序在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其他类似 CGI 的技术相比，Java Servlet 具有更高的效率，更容易使用，功能更强大，具有更好的可移植性，更节省投资。简单说来，具有以下几个特点：

1. 高效

在传统的 CGI 中，每个请求都要启动一个新的进程，如果 CGI 程序本身的执行时间较短，启动进程所需要的时间很可能超过实际执行时间。而在 Servlet 中，每个请求由一个轻量级的 Java 线程处理（而不是重量级的操作系统进程）。

在传统 CGI 中，如果有 N 个并发的对同一 CGI 程序的请求，则该 CGI 程序的代码在内存中重复装载了 N 次；而对于 Servlet，处理请求的是 N 个线程，只需要一份 Servlet 类代码。在性能优化方面，Servlet 也比 CGI 有着更多的选择，比如，缓冲以前的计算结果，保持数据库连接的活动，等等。

2. 方便

Servlet 提供了大量的实用工具例程，例如自动地解析和解码 HTML 表单数据、读取和设置 HTTP 头、处理 Cookie、跟踪会话状态等。

3. 功能强大

在 Servlet 中，许多使用传统 CGI 程序很难完成的任务都可以轻松地完成。例如，Servlet 能够直接和 Web 服务器交互，而普通的 CGI 程序不能。Servlet 还能够各个程序之间共享数据，使得数据库连接之类的功能很容易实现。

4. 可移植性好

Servlet 用 Java 编写。Servlet API 具有完善的标准，因此，为 I-Planet Enterprise Server 写的 Servlet 无需任何实质上的改动即可移植到 Apache、Microsoft IIS 或者 WebStar。几

乎所有的主流服务器都可直接或通过插件支持 Servlet。

5. 节省投资

不仅有许多廉价甚至免费的 Web 服务器可供个人或小规模网站使用，而且对于现有的服务器，如果它不支持 Servlet 的话，要加上这部分功能也往往是免费的（或只需要极少的投资）。

1.1.3 JSP 及其特点

JSP 是一种实现普通静态 HTML 和动态 HTML 混合编码的技术。

许多由 CGI 程序生成的页面大部分仍旧是静态 HTML，动态内容只在页面中有限的几个部分出现。但是包括 Servlet 在内的大多数 CGI 技术及其变种，总是通过程序生成整个页面。JSP 使得我们可以分别创建这两个部分。例如，下面就是一个简单的 JSP 页面：

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0 Transitional//EN">
```

```
<HTML>
```

```
<HEAD><TITLE>欢迎访问网上商店</TITLE></HEAD>
```

```
<BODY>
```

```
<H1>欢迎</H1>
```

```
<SMALL>欢迎,
```

```
<!-- 首次访问的用户名字为"New User" -->
```

```
<% out.println(Utils.getUserNameFromCookie(request)); %>
```

要设置账号信息，请单击

```
<A HREF="Account-Settings.html">这里 </A></SMALL>
```

```
<P>
```

页面的其余内容

```
</BODY></HTML>
```

下面是 JSP 和其他类似或相关技术的一个简单比较：

(1) JSP 和纯 Servlet 相比

JSP 并没有增加任何本质上不能用 Servlet 实现的功能。但是，在 JSP 中编写静态 HTML 更加方便，不必再用 `println` 语句来输出每一行 HTML 代码。更重要的是，借助内容和外观的分离，页面制作中不同性质的任务可以方便地分开，比如，由页面设计专家进行 HTML 设计，同时留出供 Servlet 程序员插入动态内容的空间。

(2) JSP 和 SSI (Server-Side Include) 相比

SSI 是一种受到广泛支持的在静态 HTML 中引入外部代码的技术。JSP 在这方面的支持更为完善，因为它可以用 Servlet，而不是独立的程序来生成动态内容。另外，SSI 实际上只用于简单的包含，而不是面向那些能够处理表单数据、访问数据库的“真正的”程序。

(3) JSP 和 JavaScript 相比

JavaScript 能够在用户端动态地生成 HTML。虽然 JavaScript 很有用，但它只能处理以用户端环境为基础的动态信息。除了 Cookie 之外，HTTP 状态和表单提交数据对 JavaScript 来说都是不可用的。另外，由于是在用户端运行，JavaScript 不能访问服务器端

资源，比如数据库、目录信息等。

1.2 ASP 和 ASP+

1.2.1 ASP+的介绍

ASP+的运行需要以下的环境：

Windows 2000 Professional, Windows 2000 Server, Windows 2000 Advanced Server
NGWS IE 5.5

以上是支持 ASP+的需要的系统环境，目前只有 Windows 2000 才有安装 NGWS 的可能，Windows NT, Windows 98, Windows Me 暂时还没有办法支持 ASP+，而且还不是所有的 Windows 2000 都可以，Windows 2000 的版本号是 RC 3618 以上的才行。

ASP+的功能是无比强大的，几乎可以做在网络能想到的事情，举个简单的例子，在 ASP 中，文件的上传只能通过组件才行，但是在 ASP+中只需要以下的代码就可以了。下面是一个例子：

```
<html>
<head>
<script language="C#" runat="server">
void Button1_Click(object Source, EventArgs e) {
if (Text1.Value == "") {
Span1.InnerHtml = "Error: you must enter a file name";
return;
}
if (File1.PostedFile != null) {
try {
File1.PostedFile.Save As("c:\\temp\\"+Text1.Value);
Span1.InnerHtml = "File uploaded successfully to <b>c:\\temp\\"+Text1.Value+"</b> on
the web server";
}
catch (Exception exc) {
Span1.InnerHtml = "Error saving file <b>c:\\temp\\"+Text1.Value+"</b><br>" + exc.
ToString ();
}
}
}
</script>
</head>
<body>
<h3><font face="Verdana">HtmlInputFile Sample</font></h3>
```

```
<form enctype="multipart/form-data" runat="server">
Select File to Upload: <input id="File1" type="file" runat="server">
<p>
Save as filename (no path): <input id="Text1" type="text" runat="server">
<p>
<span id=Span1 style="font: 8pt verdana;" runat="server" />
<p>
<input type=button id="Button1" value="Upload" OnServerClick="Button1_Click"
runat="server">
</form>
</body>
</html>
```

上面的程序不是太难。用 ASP+ 上传文件根本就不需要什么组件。ASP+ 的功能远不止如此, 这里只是举个小例子, 它的更多功能将在以后介绍。

编写 ASP+ 程序, 有点类似编写 VB 或者是 VC, 很多的思想都来自于 VB 或 VC。所以, 编写程序的思维需要改变, 感觉这是在写软件, 不是在写传统的 ASP 程序。

1.2.2 ASP+的环境配置

对于 ASP 来说, ASP+ 有以下突破: 运行机制不同, ASP 属于一种解释型的编程框架, 它的核心是 VBS 和 JS。受这两种脚本语言的限制, 决定了 ASP 先天不足, 无法进行像传统编程语言那样的底层操作, 所以如果需要进行一些诸如 socket 文件等的操作时, 不得不借助于用其他传统编程语言如 C++、VB、Java 等编写的组件, 由于 ASP 是解释执行的, 所以在运行效率上大打折扣。而 ASP+ 是一种编译型的编程框架, 它的核心是 NGWS runtime, 除了和 ASP 一样可以采用 VBScript 和 JavaScript 作为编程语言外, 还可以用 VB 和 C# 来编写, 这就决定了它功能的强大, 可以进行很多低层操作而不必借助于其他编程语言。另外由于是编译后运行的, 所以执行效率要比 ASP 高得多。

想要运行 ASP+ 程序, 需要 ASP+ 的支撑环境, 也就是 NGWS runtime, 可以从微软站点免费下载, 由于 NGWS 是计划在 Visual Studio 7 中出正式版本, 所以现在能下载的是 beta 版, 具体版本号是 1812.9。下载之前, 首先要看机器是否能够运行它, 具体要求如下:

- CPU: Intel Pentium II-class 300 MHz (Intel Pentium III-class 600 MHz recommended)
- RAM: 96 MB (128 MB recommended)
- Available hard disk space (for install): 250 MB
- Available hard disk space (post install): 155 MB
- Video: 800x600, 256 colors
- CD-ROM: required
- Operating System: Microsoft Windows 2000 and Microsoft Internet Explorer 5.5
- Other Software: MDAC 2.6 Beta 2

如果计算机符合以上要求, 就可以下载这个超过 80MB 的庞然大物, 具体下载地址是: <http://download.microsoft.com/download/platformsdk/Trial/1812.10full/NT5/EN-US/Setup.exe>

下载完后，只需要运行这个 `setup.exe` 程序，然后按照向导完成安装即可。现在就可以来运行第一个 ASP+ 程序了。看下面这个程序：

file : intro1.ASPx

```
<html>
<head>
<link rel="stylesheet"href="intro.css">
</head>
<body>
<center>
<form action="intro1.ASPx" method="post">
<h3> Name: <input id="Name" type="text">
Category: <select id="Category" size=1>
<option>psychology</option>
<option>business</option>
<option>popular_comp</option>
</select>
<input type="submit" value="Lookup">
</form>
</center>
</body>
</html>
```

这是一个标准 `htm` 文件，把它存为后缀为 `.ASPx` 的文件，然后在浏览器里运行一下，就是一个 ASP+ 文件。由于 ASP+ 兼容 ASP，所以任何一个 ASP 文件都可以作为 ASP+ 文件来运行，下面把上边这个文件稍微改动一下，把两个 `input` 前面的 `name` 和 `category` 分别改为姓名和分类，就像下面这个文件：

intro1c.ASPx

```
<html>
<head>
<link rel="stylesheet"href="intro.css">
</head>
<body>
<center>
<form action="intro1.ASPx" method="post">
<h3> 名字: <input id="Name" type="text">
分类: <select id="Category" size=1>
<option>psychology</option>
<option>business</option>
<option>popular_comp</option>
</select>
```