

Mastering Active Server Pages 3

Active Server Pages 3

从入门到

精通

〔美〕 A.Russell Jones 著

邱仲潘 等译

- 使用ASP3建立动态的Web站点
- 为将来的Web开发提供基础



电子工业出版社

Publishing House of Electronics Industry

URL: <http://www.phei.com.cn>

431

TP393.409

G55

Mastering Active Server Pages 3

Active Server Pages 3 从入门到精通

〔美〕 A. Russell Jones 著

邱仲潘 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 提 要

ASP是建立动态、个性化Web应用程序的新技术,在Internet和Intranet站点中广泛应用。本书从服务器设置、HTML与XML标志、脚本语言、内部对象等各个方面详细介绍了ASP相关技术,并列出了大量例子,只要稍作修改,即可把书中提供的许多代码用于自己的站点。本书作者具有丰富的ASP使用经验和教学经验,行文风趣流畅,论述深入浅出,使读者不需要太多预备知识和编程经验即可学会ASP。

本书适合作为Internet和Intranet编程人员学习这门最新技术的自学材料,也可以作为高等学校教材。



Copyright©2000 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

图书在版编目(CIP)数据

Active Server Pages 3从入门到精通/(美)乔(Jones, A. R.)著:邱仲潘等译.-北京:电子工业出版社,2000.8

书名原文:Mastering Active Server Pages 3

ISBN 7-5053-6118-X

I. A... II. ①乔... ②邱... III. 万维网-主页-软件开发 IV. TP393.4

中国版本图书馆CIP数据核字(2000)第67744号

书 名: Active Server Pages 3从入门到精通

著 作 者: [美] A. Russell Jones

译 者: 邱仲潘 等

责任编辑: 王 军

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河金马印装有限公司

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036 电话: 68279077

北京市海淀区翠微东里甲2号 邮编: 100036 电话: 68207419

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 40.25 字数: 1000 千字

版 次: 2000年8月第1版 2000年8月第1次印刷

书 号: ISBN 7-5053-6118-X

TP·3261

定 价: 66.00元

版权贸易合同登记号 图字: 01-2000-1251

凡购买电子工业出版社的图书,如有缺页、倒页、脱页请向购买书店调换,若书店售缺,请与本社发行部联系。

前 言

本书介绍如何用ASP建立动态、交互式Web站点和应用程序。Microsoft公司的Active Server Pages (ASP) 技术使Web建立产生了革命，将几种不同类型的代码混合成一个无缝的整体。ASP本身不是语言，而是个熔炉。利用HTML，利用内置Web功能，加上一些脚本技术，再加上ActiveX组件或Java小程序，就可以建立基于ASP的Web站点。

预备知识

你要有阅读能力，要构思一个项目。书中提供了大量例子和小项目，但要真正学习一门计算机语言或学习ASP，就要练习项目，吝啬的老板和紧张的交货期常常是学习的促进剂。实际上，我也不知道你要预先知道什么，但我知道：

- 你不必是HTML专家，事实上你根本不必知道HTML（知道点当然也好）。
- 你不必知道Visual Basic、VBA、VBScript、Java、JavaScript、Perl或任何其他编程语言（知道点当然也好）。
- 你不必知道Web服务器或数据库知识（知道点当然也好）。
- 你不必知道ASP——书中会详细介绍。

选择何种项目呢？不要选不喜欢的项目，学习新东西别压力太大。应当选择喜欢并可模仿的项目。用净室法，不要复制站点进行扩展，而要用自己的思想重建站点。如果已经有Web应用程序的想法，则你很快就会成为一位ASP程序员。

读者对象

如果你买了这本书，我们就会有许多时间进行文字交流了。因此，你一定想知道我对你的假设。每个作者都有一个假想读者，我也是。如果你不符合这个条件，则应重新考虑是否买这本书。我假设：

- 你想用ASP建立动态Web站点或应用程序，或想提高现有水平。
- 你喜欢探索和了解技术，而不只是从光盘中复制代码。
- 你想知道ASP除了有一些内置功能之外，还能干什么。
- 你愿意花时间学习新技术。

我不指望你一下子读完书中的每一个字，而希望你循序渐进，学习后面的内容时已经掌握了前面的内容。本书不是参考手册，每一章并不是独立的。本书不能跳着读，而要顺序阅读。

本书内容

本书综合介绍ASP技术和几个相关技术。书中会介绍HTML和Cascading Style Sheets (CSS)，但不是HTML和CSS的启蒙教材或参考手册。书中深入介绍ASP对象模型和使用ASP对象与ASP本身的最佳方法，并大量介绍Microsoft的VBScript脚本语言，同时简单介绍了JScript，这是Microsoft的JavaScript变形。我们介绍如何用脚本语言从文件和数据库中读取和写入数据，如何建立Web站点，如何用Common Object Model (COM) 对象扩展ASP。

写作目的

通常，书本中是枯燥的专业知识，但我想介绍一下自己，介绍本书的写作目的。

我的背景

我是从事音乐、计算机培训 (CBT)、交互式视频培训 (IVT)、Web培训 (WBT)、数据库应用和通用人力资源Web应用的。我从1979年开始与计算机打交道，当时是田纳西大学钢琴专业本科生，发现Dr. Donald Pederson的音乐理论计算机实验室有许多全新的Apple II 微机，主内存为8K。当时不仅是美丽的，而且是不可抗拒。我的第一个应用程序是教人们学习如何识谱和写谱，音乐家们称其为听觉训练。

从这时起，我就迷上了计算机。获得音乐理论的硕士学位之后，我到依利诺斯大学攻读辅助教育的博士学位。这所大学是第一个重要计算机教学系统PLATO的所在地。作为研究助理，我和《Teaching Computers to Teach》的作者Dr. Esther Steinberg一起研究不同界面特性对初学者和熟练计算机用户的相对重要性。

我的工作

获得博士学位不久，我就到InterCom Training公司工作（我的老板很好，让我在1990年补充了缺课）。在InterCom Training公司的11年里，我是个编程人员、教学设计人员和项目与小组带头人。我从v1.0开始就使用ASP。1995年，我开始用Visual Basic生成交互式Web站点。1996年下半年Microsoft公司推出ASP时，我发现其比当时的Web技术进了一大步。Web显然正在改变公司对数据和应用程序的看法。尽管许多公司已经有或正在建立Web站点，但交互式Web站点的思想刚刚引起公司的注意。

从此以后，我建立了几十个大大小小的Web应用程序，从小的交互Web站点到公司数据库搜索引擎、数据密集的大型机模拟以及全球访问的人力资源应用程序。

最近，我加盟VF Services公司，它是拥有Lee Jeans、Healthtex、Wrangler和Russell Athletic等品牌的服装公司，我在公司从事大型图形数据库、系统设计和技术教学工作。

我的写作经历

近年来，我每隔一月要为Visual Basic Programmer's Journal杂志的Interactive Developer栏供稿。文章与书不同，除了篇幅不同之外，文章可能比书更难写。文章要求简明扼要，而书则要求完整。

好在这是我的第二本书，我的上一本书是为Sybex公司编写的《Visual Basic Programmer's Guide to ASP and IIS Applications》。其实这本书与ASP没有什么直接关系，主要是介绍用Visual Basic (VB) WebClass技术编写Web应用程序，通过写作，使我获益不浅，并在本书创作过程中受益于这些经验。

我的教学经验

我很荣幸多次在不同领域当过老师，教过钢琴、乐理、听训、计算机编程和阅读，还教过如何处理毒蛇、如何使用大型机、如何经营炼油设备、如何建立变压器、如何进行职业规划。但我总是乐教乐学。因此，我在每次教学中都记住这一点，不仅信息很重要，事实上，信息通常需要自动化之后才真正有用。重要的是能用信息干什么。对我来说，学习的最妙之处在于突然认识到你又扩展了连接知识模式的能力。

任何艺术工作的最佳部分在于我妻子Janet和我所称为的“境界”，即进行创造性劳动时的体验。教学中能够得到这种体验，因为教学是一门艺术。学习也能够得到这种体验，因为过程会突然自动化，思路会豁然开朗。编程也一定能够得到这种体验，因为编程和音乐、绘画一样，也是一门艺术。本书中我达到了这种境界，但愿你也能达到。

学习什么

到本书结束时，你将有ASP技术的全面基础，能够处理从简单Web站点到完整Web应用程序的各种项目。要在ASP中有所收获，就要多学VBScript。ASP本身只是进行决策的方法：返回什么HTML，启动什么对象，客户机运行什么浏览器，等等。因此，本书花很大篇幅介绍ASP所需的外围技术。

我们将介绍Internet如何工作，如何影响人们对Web站点和Web应用程序的看法，将介绍HTML基础。当然，还要介绍ASP脚本，主要是VBScript，这是ASP的缺省脚本语言，同时也介绍一些JScript。我们将介绍如何用ASP处理文件、数据库和外部COM对象，还将介绍用Visual Basic生成COM对象，扩展ASP功能。最后，我们要介绍改变了Web应用程序工作方式的XML技术。

书中有大量练习。作为原先的职业音乐人，我认为实践至关重要。我知道并非人人都喜欢编写代码，但我认为编写代码是学习的最佳方法。对于不爱练习的读者，我在本书选配光盘中提供了代码，其中有所有命名清单和完整项目。我希望你在读完本书之后再去寻找本书选配光盘中的代码。

几乎每一章都包含代码，通常包含不少代码。我希望你亲自输入这些代码。我相信实际输入的代码量与编程水平成正比。不输入一定数量的代码，就不可能成为熟练编程人员。

不输入一定数量的代码，也很难成为熟练的设计人员。

编程书通常分为三类：不包含大项目的主要用于参考的书、包含一个或两个大项目以贯穿全书的书、用小项目演示特定语言特性的书（还有一种，就是透视编程内幕的书）。

本书属于小项目类型。我读过许多编程书，没见过哪个大项目真让我感兴趣到愿意实际建立这个项目。许多章节都包含小项目，利用这些项目可以构造你要建的应用程序。有时可以直接修改代码以在你的应用程序中借用，有时则要利用这些项目包含的思想。我将它们编写成不需运行项目就能了解其主要特性，但愿对你有用。

需要什么

要使用本书的项目，需要几个程序，也许大部分已经拥有。

一个Web服务器 需要有Internet Information Server (IIS) 或Personal Web Server (PWS) 并安装ASP扩展，对本书的第一部分，也可以用运行Chili!Soft ASP的Unix服务器（本书选配光盘中带有Chili!Soft ASP的测试版）。

一个浏览器 我推荐Internet Explorer (IE) 5。本书的例子是用IE 5开发的，但也可以用任何其它浏览器，包括Netscape浏览器。ASP是个服务器端技术，能向任何浏览器客户机提供内容。注意，本书的有些例子无法在Netscape浏览器中运行，因为Netscape浏览器对World Wide Web Consortium (W3C) 的Document Object Model (DOM) 规范的支持不如IE。有些例子也无法在IE 4.x以前版本中运行。如果使用3.x以前的浏览器，则应先升级之后再利用本书。

一个编辑器 建议用Microsoft InterDev所带的编辑器，因为它能彩色编码HTML和脚本，但也可以用其它编辑器。只要Windows NotePad就可以编写本书的任何ASP例子和项目，ASP并不依赖于任何特定开发环境。

Visual Basic 对第26章，需要Microsoft Visual Basic (VB) 5带有service pack 2 (sp2) 或VB 6才能建立组件。如果没有VB或不准备建立VB例子，别担心，我已经提供了预编译组件，可以在服务器中运行。

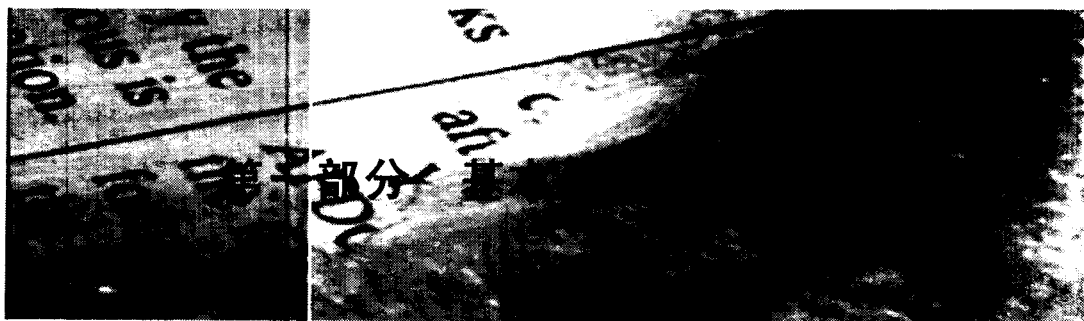
本书选配光盘内容

本书选配光盘中包含书中的所有代码例子，加上已编译的组件、数据库和SQL代码。预先写好的代码能更快运行，但不利于学习编写ASP代码，最好的方法是亲自编写。当然，复制代码很省事，但亲自编码能学到更多东西。

本书代码组成一个MasteringASP站点。要使用这些代码，就要在服务器中建立MasteringASP虚拟目录。首先将MasteringASP目录和所有子目录复制到服务器中，然后按第2章中的指令设置站点。每章的代码放在Chapter1, Chapter2等子目录中。对有些章节，要有Web服务器帐号的其他权限以便生成和写入文件。需要其他权限时，书中会明确指出。

除了IIS管理之外，还要有足够权限以管理Microsoft Transaction Server (MTS)，Windows 2000中更名为COM+。书中生成和安装Common Object Model (COM) 组件时需要这些权限。

光盘中所带的Chili!Soft ASP Developer Edition是个跨平台ASP，可以在NT和Linux中运行。有关Chili!Soft ASP的详细内容，请见光盘中的Chili!Soft ASP白皮书。



第1章 内幕——活动服务器页面如何工作

- ASP简介
- ASP的好处
- 发布动态内容的其他方法
- ASP的竞争对手
- ASP与其竞争对手的比较
- 如何使用ASP
- 为何不用ASP
- ASP的作用
- Web请求
- ASP请求与HTM请求的差别

Active Server Pages (ASP, 活动服务器页面) 技术实际上很简单。如果能把HTML与代码配合起来, 则Web站点能成为强大的工具。ASP正是起这种作用的。ASP可以利用Web服务器的功能处理用户请求和基于逻辑文件和数据库数据提供动态的个性化内容, 以及处理用户的个人数据。换句话说, ASP使多个用户可以同时运行Web服务器的程序。实际上, 你可以得到大型机的功能和本地工作站的图形处理功能, 这一切都封装在跨平台的浏览程序——浏览器中, 并封装在全球标准化的通信机制——Web中。本书介绍如何利用这个功能来发布灵活而个性化的应用程序。

实际上, Web应用程序包括Web服务器与浏览器之间的一系列简短对话。记住, 《乱世佳人》中斯佳丽在舞会中被十二个小伙子团团围住时, 每个小伙子都想与斯佳丽约会。小伙子们发出邀请, 斯佳丽作出应答。用户/服务器之间的关系也差不多。每个用户(浏览器)向服务器发出请求, 就像小伙子们发出邀请, 服务器像斯佳丽一样迅速作出应答。每个用户与你的Web页面约会时, 好像你是与他单独交流。Web应用程序就要起到斯佳丽的作用, 你要吸引每个用户的注意, 响应每个用户, 而不能有所偏颇。

虽然所有用户都在同一机器(你的服务器)上运行同一程序, 但ASP能够让你一一处理。显然, 在一台计算机上对多个用户运行复杂程序需要占用大量资源。好在, Microsoft保证了这些资源能供你任意使用。ASP提供:

- 对每个用户保存个性化数据
- 访问文件系统
- 访问数据库
- 启动与控制任何COM (Component Object Model, 组件对象模型) 组件

简而言之, ASP可用来建立多用户应用程序, 可以利用ASP提供应用程序的伸缩性。如

果认真规划ASP应用程序，则可以实现处理30~100个同时用户或每天几千次访问的单服务器应用程序和处理几千个同时用户或每天几百万次访问的多服务器应用程序。

ASP不是单一的技术。有效利用ASP要求学习一系列工具、语言、技术和方法。这一切不是突然冒出来的，而是各有自己的演变历程。介绍ASP技术之前，先要介绍这些技术的历史。

ASP简介

要介绍ASP，最好的方法是先介绍ASP之前的一些技术，为什么需要新功能，ASP如何实现这些新功能。

HTML简介

HTML (HyperText Markup Language, 超文本标记语言) 是用于标记文本文件格式的特殊语言，用于格式化Web显示内容，通常在浏览器中显示。HTML是只有几个命令的简单语言，这是HTML得以迅速普及的原因之一。HTML出现之前，非编程人员要显示文本和图形，唯一的方法是利用WYSIWYG (What-You-See-Is-What-You-Get, 所见即所得) 字处理程序。但是，要浏览文档，就必须有相同操作系统、相同字处理程序和相同的字处理程序版本。HTML改变了这个现实，使整个世界焕然一新。如今，HTML也比原先丰富了许多，但只要几个小时的简单介绍，就能使用户生成相当精彩的页面了。

第4章“HTML基础”中将更详细地介绍HTML，作为HTML有多么简单的例证，下面举几个例子：

```
<p>This is a paragraph</p>
<b>This text is bold</b>
<ol>
  <li>This is an ordered list</li>
  <li>This is an ordered list</li>
  <li>This is an ordered list</li>
  <li>This is an ordered list</li>
</ol>
```

三角括号 (<>) 将内容与布局分开。三角括号之间的文本是布局或标记。标记控制浏览器如何显示内容，内容就是三角括号之外的文本。

HTML的简单性是其得以迅速普及的原因之一，另一个同样重要的原因是文件之间易于漫游。只要利用鼠标指向和单击，用户就可以在文件之间移动，而不必知道文件的实际位置。这种导航称为超文本，是30年前由Ted Nelson发明的，这时还没有HTML。

由于HTML的简单性，使其迅速成为World Wide Web (WWW, 万维网) 显示语言的首选。人们常说Web是向全球读者及时提供信息的新媒介，但其实不太准确。当然，人们把几千万个文本与图形页面放在Web上，但读者很快就会发现，这些信息缺乏针对性。他们需要个性化的页面，就如他们喜欢的Windows程序一样。他们还需要站点记住他们的个人选项、通信、交互，简而言之，读者需要应用程序，而不是单纯的信息。

进入CGI

HTML需要一些帮助手段。作为简单的布局语言，HTML无与伦比，但作为应用程序引擎，则它显然有许多不足。很快，CGI（Common Gateway Interface，通用网关接口）应用程序应运而生。CGI应用程序完全在服务器上运行。浏览器与服务器连接并作出文件请求时，CGI应用程序就如HTML文件一样返回HTML。差别在于，CGI应用程序可以处理浏览器发送的信息，返回响应不同条件的不同HTML。CGI应用程序可以向一个用户发送一种类型的信息，向另一个用户发送另一种类型的信息，或根据用户浏览器的类型和版本发送不同文件。

尽管CGI应用程序可以用各种语言编写，但大多数早期CGI编程人员使用Perl，这是个具有强大文字处理功能的语言。由于Perl是个脚本语言，许多CGI应用程序都称为CGI脚本。脚本语言相对较小，通常是解释性语言，而不是编译性语言。Web站点设计人员很快就用预先编好的CGI脚本进行简单计算，如计算访问Web页面的用户数。有些CGI应用程序支持Web搜索引擎以检索内容和将索引放进强大的数据库中。

这些CGI应用程序用HTML表单接收用户请求，搜索数据库，并提供定制的反应。CGI应用程序是新的活动Web的先驱者。简单的存储页面足以提供信息显示，但现在人们用Web应用程序来寻找这些页面。Web站点编程人员开始用CGI程序提供动态内容，而不是静态HTML。

但是，CGI程序有几个缺点。Web是小事务的世界。每个事务包括浏览器的一个请求或一系列简单请求。当你请求一个页面时，Web服务器提供该页面，然后就会忘记你。对于较忙的站点，这种请求可能以每秒100次请求发生。第一代CGI程序用执行程序装入每个请求到内存中，然后CGI程序处理这个请求并终止。所有这些装入和卸载都要占用服务器时间和资源。显然，装入和执行程序要比返回HTML文件内容需要更多内存和Central Processing Unit（CPU，中央处理单元）时间。Web服务器设计人员开始寻找发送动态内容的更有效方法。

ISAPI与NSAPI

自CGI编程以来，Web服务器软件厂家开始考虑Web服务器本身的挂接。他们标准化ISAPI（Internet Server Application Programming Interface，Internet服务器应用程序接口）或NSAPI（Netscape Server Application Programming Interface，Netscape服务器应用程序接口，对Netscape服务器）的概念。利用这些挂钩，使用C语言和其他通用编程语言的编程人员可以编写像CGI程序一样处理请求与返回动态内容的程序。与CGI程序不同的是，这些新的ISAPI与NSAPI程序保持装入服务器地址空间的内存中。这样，ISAPI与NSAPI程序可以执行与CGI程序相同的任务，而没有对每个请求装入与终止程序的开销。

Microsoft公司推出了两种不同的ISAPI程序：过滤器（filters）在服务器看到Web请求之前截获Web请求，而应用程序（applications）处理Web请求。严格说来，ASP是ISAPI应用程序。不要把应用程序（applications）与用ASP建立的应用程序混起来。本书中，我把ASP应用程序称为ASP引擎。ISAPI ASP应用程序是个DLL（Dynamic Link Library，动态链接库）（asp.dll），它解析文件并执行其中包含的代码。

ASP的好处

ASP的好处很多，将在本章多处介绍。下面先介绍ASP的两个主要的好处。

ASP是独立于语言的

ASP引擎是独立于语言的。事实上，ASP引擎并不实际执行你所编写的代码。相反，它是个与语言无关的脚本宿主。ASP引擎可以使用任何与Microsoft Scripting Host（Microsoft脚本宿主）要求相符的脚本语言，甚至可以处理同一页面上使用多个脚本语言所编写的代码。

ASP引擎先分开HTML与脚本代码，然后请求相应的脚本引擎来执行代码。本书的大多数代码使用VBScript，这是缺省ASP脚本语言，但也可以看到许多JavaScript和一些PerlScript。

ASP是针对非编程人员的

计算世界的一个理想境界是寻找一种语言，让普通人（非编程人员）能够进行编程工作，其思路是，编程人员生成可复用的通用实体（块），任何人都可以轻轻松松地将这些块组合成复杂程序。不要担心，编程的简单性是个发展的目标。

在计算世界的早期，编程人员使用机器语言，通过拨动开关操作位和字节。汇编语言改变了这一切，汇编器使非编程人员能方便地操作位和字节。随着汇编器变得越来越复杂，出现了Fortran和C之类的语言，用于连接汇编语言模块。在Windows之前的DOS（Disk Operating System，磁盘操作系统）时代，批处理文件中使用简单的脚本语言进行连接。将简单批处理文件连接起来，可以使其他程序自动化并完成复杂任务。但是，最初的简单批处理语言不断复杂化，最后只有编程人员能够使用了。几年前，Visual Basic还是个连接语言，如今Visual Basic已经是能够生成构件的语言了。如今的连接语言是ASP，如今的ASP更适合作为连接语言而不是进行主流处理，但今后也可能改变。

显然，必须了解ASP不是低级编程语言，也不是数据库语言，但可以从ASP访问数据库。如果你要把ASP作为数据仓库/数据净化（data warehouse/data-scrubbing）功能的前端，则应三思而行。ASP不是组件建立语言。如果要用ASP生成下一代字处理器程序或电子报表，则显然找错对象了。ASP不是数学建模语言，不是图形操作语言，也不是GUI（Graphical User Interface，图形用户接口）建立程序。

ASP的优势在于提供简单决策功能，使静态HTML页面与后端组件协调起来，对不同客户返回快速HTML响应。换句话说，ASP是胶水。

但ASP又不是单纯的胶水。尽管你不一定要有很高编程水平就可以建立简单ASP页面，但只有编程人员才能建立ASP应用程序。编程主要靠头脑、靠天赋和靠实践，而不是靠教育。换句话说，只要有一定的天赋和实践，你就能成为编程人员。

和任何艺术一样，ASP应用程序编程也是有窍门可循的，本书正是要提供这种窍门。我们的目标是让基础代码自动生成，使更多的精力放在大问题上。如果要巧妙使用ASP，就要编写大量小段程序，将其不断完善，然后再组合成更大更复杂的程序。你会发现，ASP本身不是完整的编程方案，还可以用其他同样有趣的工具帮助建立有用而美观的Web应用程序。

发布动态内容的其他方法

ASP不是发布动态内容的唯一方法，只是最新的方法而已。下面介绍发布动态内容的其他方法。在90年代中期，热门的Web技术是CGI程序，使Web服务器能够不仅提供静态HTML页面，而且提供其他功能。

何谓CGI程序

CGI程序是服务器响应浏览器请求时执行的小执行程序。CGI程序处理代码中的请求并向浏览器返回HTML。CGI程序与静态HTML页面的差别在于向浏览器返回的HTML可以随请求者的不同而不同，随请求时间的不同而不同，等等。Microsoft发现CGI程序并不是处理活动内容请求的最有效方式，因为服务器对每个请求都必须装入CGI程序。

这就涉及到HyperText Transfer Protocol (HTTP, 超文本传输协议)的工作方式。HTTP请求是客户机与服务器之间的简单事务。一旦事务完成之后，客户机与服务器之间不再有任何关系。

假设舞会上两个人相遇，每次见面时都要相互介绍，进行简单对话，然后分手，再也记不住对方。也可以想象一下电影《Groundhog Day》中的情形，每一天都是重复的，电影中每个人每天说相同的话，做相同的事，只有主角Bill Murray知道他们的生活是雷同的。Web请求也是这样，只是没有主角Bill Murray而已。

由于客户机与服务器都把每个HTTP事务当作全新的实体，因此要为每个请求进行一系列复杂的握手。更糟的是，由于服务器在每个事务之后忘记客户机，因此也删除处理该请求时调用的任何CGI程序。然后服务器要为下一次请求重新装入CGI程序。装入执行程序通常比执行这个程序占用更多时间。对Web站点而言，装入程序是个非生产性活动，我们的目的是尽量有效地利用服务器。

何谓ISAPI

要消除与CGI应用程序相关的开销，Microsoft公司引入了ISAPI (Internet Server Application Programming Interface)。装入ISAPI应用程序与装入CGI程序需要同样长的时间，但服务器不在每次请求结束时删除ISAPI应用程序，而是使其保持装入，直到服务器关闭或到达预定时间。ISAPI程序有两种：ISAPI应用程序与ISAPI过滤器。

ISAPI应用程序

ISAPI应用程序在请求Web服务器之后运行。ISAPI应用程序可以向Web服务器增加功能，同时还可利用Web服务器提供的所有其他特性。ASP引擎是个ISAPI应用程序。IIS (Internet Information Server, Internet信息服务器)向ASP引擎发送ASP文件请求，而ASP引擎处理提供动态内容的请求。

ISAPI过滤器

ISAPI过滤器在请求Web服务器之前运行。ISAPI过滤器可以监视或截获进行特殊处理的请求。例如，将请求分类并根据用户IP地址分配服务器带宽的程序即可采用ISAPI过滤器。

程序截获请求并过滤之后再访问Web服务器。也可以将ISAPI过滤器作为防火墙，阻止不信任来源的请求。

ASP比HTML的优势

HTML是简单而灵活的格式化与布局语言，但没有编程结构。如果需要显示静态文本与图形，则可用HTML；如果需要显示经常改变的内容，或需要对每个用户显示不同内容，则不能用HTML，因为HTML没有任何决策功能。

例如，假设要显示基于投票访谈的全国投票结果，则可以建立将结果放进数据库中的基础结构。HTML前端显示一个全国地图。客户选择一个或几个州时，可以按该州的投票结果显示各个县的投票情况。

如果在选举结果出来之后再建立站点，则用HTML就可以了。但要显示正在进行的投票结果，则要显示实时数据，即需要编程语言，而HTML并不是一个编程语言。

ASP脚本提供了读取数据库、取得该州最新统计结果、按县进行分组和根据最新数据生成定制HTML响应的功能。

每次请求时改变的内容称为动态内容。只提供静态内容的站点很快会让人觉得乏味，如今大多数Web站点都是动态站点。提供动态内容的站点，特别是用户与内容交互而不仅是通过指向与单击改变页面的站点即称为Web应用程序。

ASP的竞争对手

Web是全球可访问的，是发布应用程序的全新模式，允许中央管理，即时、单点升级和用相同代码库向不同操作系统上的多个客户发布应用程序。人们都看好这个新模式，因此ASP有许多竞争对手。当然，这要看你对竞争对手的定义。不同厂家都想标新立异，使自己的产品超越一般的Web开发环境。

ASP产品也是个难以分类的产品，它不是真正意义上的开发环境（可以用任何文本编辑器开发ASP页面），也不是个语言（可以用各个脚本语言开发ASP页面）。同样，ASP可以建立简单Web站点，也可以建立复杂Web应用程序。

ASP在应用领域的主要竞争对手有Application Server供应商，Sun的JavaServer Pages，Java本身，C++ ISAPI应用程序以及Visual Basic。Microsoft公司推荐的ASP站点建立工具是Visual InterDev，其在站点建立领域的主要竞争对手有FrontPage，HomeSite，DreamWeaver和各种共享软件与自由软件站点建立器/HTML编辑器。

Application Server供应商

ASP的高端竞争对手是Web应用程序开发环境，其中一些称为Application Server供应商，即表示将应用程序开发环境捆绑成企业级软件，通常有硬件和服务合同。Cold Fusion、Bluestone和NetDynamics就属于这类竞争对手。这三者都提供HTML标记扩展和定制命令，可以方便地访问数据库、格式化页面和发布定制内容，都是强大的解决方案，但比较昂贵。

Application Server的主要优点是能更方便地建立易于扩展的复杂页面，主要缺点是要求自定义服务器安装，通常由几个客户付费，可能隐藏低层细节（甚至隐藏代码），很难改变你的站点以区分用同一软件包建立的其他站点。

JavaServer Pages

最近, Sun公司推出了JavaServer Pages (JSP, Java服务器页面), 我认为它将在今后成为ASP的主要竞争对手。JSP已经可以在几种不同Web服务器上运行, 包括Apache (Unix)、NetScape和IIS服务器。ASP也能通过ChiliSoft的Chili! ASP产品在Unix上运行, 但会失去一些功能。

Java

Sun公司最初把Java作为Web上的通用客户端语言推出, 但很快发现其性能无法胜任。但是, 最近Java在服务器上派上了新的用场。Java与C++相似, 但具有一些特性 (如自动内存管理), 使其成为更好的快速开发环境。此外, 近年来Java开发环境已经大有改进, 有些已经可以像Microsoft的Visual Studio一样特性完整。

Java本身是个优秀的语言, 和Microsoft的Visual Basic一样可以解释或编译。Java编译成本机代码时, 可以和编译C语言代码一样快。

许多C和C++开发人员发现很容易转入Java, 因为这两种语言的语法有极大的相似性。Enterprise Java Beans的开发 (类似于Microsoft的COM技术) 加上Java在不同操作系统中运行的功能, 使Java成为开发业务连接Web应用程序的首选。

VB IIS应用程序

Microsoft本身提供了ASP技术的高端和低端竞争对手。在高端, Visual Basic 6增加了WebClasses类模块。WebClasses是在ASP技术上建立的, 但可以将代码编译成ActiveX DLL。

理论上, 编译代码能使应用程序运行速度更快。实际上, 应用程序的运行速度取决于多个因素, 大多数因素与代码编译或解释无关。编译应用程序的主要优点是, 销售代码时, 客户不能看到代码, 他们只能使用应用程序, 而不能改变应用程序。主要缺点是要在服务器上注册已编译的ActiveX DLL, 即发布应用程序之后不能那么方便地修改代码。

其他 (如C++)

可以用几乎任何语言编写Web应用程序, 但许多开发公司都选择C++, 因为它是访问Windows底层的最佳语言, 速度很快, 且C++代码能移植到其他一些操作系统上。尽管C++不是严格的ASP竞争对手, 但我把它放在这里, 因为它是商业性应用程序开发的最佳选择。如果准备用ASP开发商业性应用程序, 则应知道你的竞争对手可能使用更快的方法。

ASP与其竞争对手的比较

ASP比起大多数其他Web应用程序开发语言和环境有几大优点, 特别是针对Intranet开发:
ASP代码放在文本文件中 文本文件即使在部署之后也很容易修改。用文本编辑器远程解决程序问题是个巨大的优点。Web应用程序如果依赖于编译代码或注册过的ActiveX对象, 则会更难维护和升级。此外, 今天编写的ASP代码一定可以在明天用文本编辑器

修改，而文本文件与IDE（Integrated Development Environment，集成开发环境）的版本无关。大多数其他代码开发环境则不是这样，因为它们不断改进、不断改变。它们通常（但不总是）向下兼容旧版本，但通常无法用另一环境的编辑器进行编辑。利用ASP文件，则任何人均可用文本编辑器编辑代码。

ASP代码超时 缺省情况下，IIS在90秒钟之后不再执行ASP页面。可以调整这个值。因此，如果不慎编写了一个死循环，或允许别人请求几百万个记录，则只在超时间隔之内影响服务器。许多ISP（Internet Service Providers，Internet服务提供者）通常不允许放置编译应用程序而允许放置ASP应用程序，一个原因就是脚本能够过期。

ASP代码是服务器完全的 ASP代码在有限空间内运行，例如，利用ASP不能自然读取和写入二进制文件。利用本机ASP脚本很难完全摧毁IIS服务器。因此许多ISP通常不允许放其他技术开发的应用程序而允许放ASP应用程序。

ASP代码不需要注册 IIS安装程序安装ASP运行DLL、脚本DLL、ADO（Microsoft ActiveX Data Objects）DLL和Microsoft Scripting Runtime DLL，运行ASP应用程序只需要这些代码就够了。大多数其他开发工具则要求其他服务器端安装和注册操作。有些要求安装特殊的服务器应用程序。在开发服务器上，这并不难，但在生产服务器中，则会造成问题，特别是在只要纠正拼写错误或进行小更改时。

ASP应用程序通常较小 由于服务器中已经安装所有DLL，因此只要发布代码文件、图形和支持文件即可运行ASP应用程序，这些文件通常较小，可以大幅度压缩。注意，应用程序中增加编译过的ActiveX组件之后，情况就不再这样了。

可以升级ASP应用程序而不停止IIS 尽管这好象很平常，其实是个很大的优点。如果只有这个应用程序在服务器上运行，则没什么了不起，但如果同一服务器中运行几十个应用程序（大公司通常如此），则可以升级ASP应用程序而不停止IIS真是太方便了。这种改变通常要进行计划，而且要提前计划。问题在于，谁都无法预料停止服务器对该服务器上运行的所有应用程序会产生怎样的影响。设身处地为用户想想，你肯定不希望应用程序仅仅为了某个人要升级而突然停止响应，甚至失去数据。

如何使用ASP

介绍ASP功能之后，如何用ASP建立应用程序呢？使用ASP的方法取决于应用程序的规模、预计的客户数、服务器的功能、应用程序是面向intranet还是面向Internet以及开发与资源。大多数可扩展站点用ASP的脚本功能作为MTS（Microsoft Transaction Server，Microsoft事务服务器）组件的前端，MTS组件进行所有数据库访问和大多数商务逻辑处理。但你不一定要这样使用ASP。只要牺牲一些速度和可扩展性，ASP完全可以处理几个用户请求而不需外部COM对象或MTS的帮助。

建议在学习时只用ASP。随着脚本和数据库功能的增加，就可以开始建立和使用COM对象。如果ASP固有的功能不能满足可扩展性要求，则要开始使用MTS。

简单文本处理

用ASP脚本进行简单文本处理。这是ASP的强项，通常涉及一系列如何格式化响应的决策。避免用ASP脚本进行复杂字符串操作（特别是在循环中），因为其速度很慢。

复杂决策

用ASP进行复杂决策。计算机在作出简单决策时的速度很快，ASP脚本虽然不是最快的语言，但足以进行这些决策。组合决策与文本处理即可建立生成个性化交互式应用程序的构件。

浏览器与COM组件的中介

随着决策与数据处理要求的增加，可以购买或建立COM组件。服务器COM组件的使用增加时，ASP即成为浏览器与COM组件的中介。前面曾经介绍过，ASP是浏览器与COM组件的中介胶水。

为何不用ASP

ASP有强项也有弱项：

- ASP是解释性的，而不是编译性语言，因此自然比其他编译性解决方案要慢。
- ASP没有强变量类型——所有变量类型都是Variant。Variant很方便，但比类型变量更大更慢。
- ASP要在每次使用时插入包括文件。
- ASP把所有对象变量作为后期绑定对象。它要对每个属性和方法访问请求对象信息，从而会减慢响应速度。

相反，编译语言通常有强变量类型，编程人员要指定每个变量的内容类型和取值范围。这样，编程人员的工作更多，而程序的工作则更少，因为程序不再需要决策。同样，编译程序中的包括文件通常只在编译时插入一次，对象引用可以早期绑定，使编译器可以在编译时而不是运行时检查属性和方法名与参数。

作为逻辑

ASP不适合实现业务规则，原因有两个。第一，因为这种规则通常涉及多个对象属性和数据库访问。VB和C++之类的语言用早期绑定来访问对象变量，即编译器在编译时知道对象显露的属性、方法、参数和返回值。相反，ASP要在运行时对每个属性和方法访问请求对象信息。这种对象访问称为后期绑定。可以想象，后期绑定要比前期绑定慢好几倍。后期绑定通常不成问题，除非要多次访问一个对象，如在循环中或实现复杂的业务规则时。

第二（也许更重要的是），理想程序模型将业务规则放在对象中，生成规则实现方法与接口之间的明确分界。将业务规则放在ASP页面中，与格式化和接口代码混在一起会破坏这种分界，而且代码可能在多处重复出现。如果要改变一个对象中已经分开的规则，则可以改变这个对象而对应用程序的其他部分不造成大的影响。如果业务规则代码与接口和格式代码混在一起，则很难改变。注意，分开业务规则还可以单独测试这些规则，可以在多个应用程序中重用相同的规则。