



# 预览 ASP+

Richard Anderson Alex Homer  
Rob Howard Dave Sussman

著

高波 唐莉 王琰 等译

人民邮电出版社  
www.pptph.com.cn



乐思

# 预览 ASP+

Richard Anderson Alex Homer 著  
Rob Howard Dave Sussman

高波 唐莉 王琰 等译

人民邮电出版社

## 图书在版编目 (CIP) 数据

预览 ASP+ / [美] Anderson, R. 等著; 唐莉等译. —北京: 人民邮电出版社, 2001.4  
ISBN 7-115-09130-7

I. 预… II. ①安… ②唐… III. 主页制作—应用软件, ASP+ IV. TP393.092

中国版本图书馆 CIP 数据核字 (2001) 第 11227 号

---

### 预览 ASP+

---

- ◆ 著 Richard Anderson Alex Homer  
Rob Howard Dave Sussman  
译 高波 唐莉 王琰 等  
责任编辑 靳文娟 陈冀康
- ◆ 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  
北京顺义振华印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 800×1000 1/16  
印张: 22.5  
字数: 495 千字 2001 年 4 月第 1 版  
印数: 1-5 000 册 2001 年 4 月北京第 1 次印刷  
著作权合同登记 图字: 01-2000-3568 号  
ISBN 7-115-09130-7/TP·2087

---

定价: 38.00 元

## 内 容 提 要

本书全面介绍了 ASP+ 的概念及应用技术。首先解释 ASP+ 的总体概念，结合实例讲解如何创建 ASP+ 应用，其中涉及服务器端包含控件和新的数据存取技术 ADO+ 等；接着讲述了 ASP+ 一些高级编程技巧；然后介绍了另一种新技术——ASP+Web 服务，它演示了如何创建可以通过 Internet 获取的 XML 服务，使得开发 B2B 应用程序更加轻松。本书最后就 ASP+ 应用程序框架一些更深入的技术进行了讨论并给出了 ASP+ 应用示例，使读者对如何创建完整的应用程序有一个更深刻的认识。

本书语言简洁，实例丰富，特别适合有 ASP 开发经验的 Web 技术人员阅读，也适合对 Web 应用程序和服务开发感兴趣的读者学习参考。

## 版 权 声 明

Copyright © 2000 by Wrox Press.

All rights reserved.

Authorized translation from the English language edition published by wrox Press.

本书为 Wrox Press（乐思出版社）授权出版的中文译本。本书的专有出版权属人民邮电出版社。未经原版出版者和本书出版者书面许可，任何单位和个人不得擅自摘抄、复制（包括资料和出版物）本书的部分或全部，以任何形式进行传播。

版权所有，侵权必究。

## 作者简介

### **Richard Anderson**

Richard Anderson 是一位微软技术的专业研究员、软件工程师及作家。尽管已过人生最佳年龄，他仍然具有非凡的才智。目前 Richard 作为因特网/产品专家，与他人一起用一些极“酷”的技术帮助 BMS 开发企业级 Internet 薪水册系统。Richard 希望对 Matt Odhner（及 NASA）、Mark Anders、Scott Guthrie、Rob Howard、Sam、Alex、Tina 以及 Dave 表示特别的感谢，是他们使他能够充分享受西雅图之行。

### **Alex Homer**

Alex 是一位软件开发人员和技术作者，居住和工作在英格兰的 Derbyshire Dales 的田园诗般的乡村。他接触计算机很晚——实际上，当他在学校时，还在对发光二极管（LED）手表感兴趣。之后，他获得了数学学士学位，并且开始以计算机设计为职业。但他并不只忙于为 Stonebroom Software (<http://www.stonebroom.com>) 开发 ASP 组件，他还喜欢安装和编写最新的测试代码。Alex 的联系方式为：[alex@stonebroom.com](mailto:alex@stonebroom.com) 或 [alex@stonebroom.co.uk](mailto:alex@stonebroom.co.uk)。

### **Rob Howard**

Rob 是一位微软早期 Adopter 组的开发人员和技术作者。他与一些重要的网络公司一起工作，帮助他们更好地理解、采用微软技术并付诸于实践。当他不写程序（或书）时，通常飞往华盛顿河垂钓。你可以通过 [rhoward@microsoft.com](mailto:rhoward@microsoft.com) 与他联系。

### **Dave Sussman**

在意识到写作是件更有趣的事之前，Dave 大部分时间都花在程序开发上。尽管第一版和测试版代码同样令他兴奋，但他专门从事 Internet 和数据存取技术。如 Alex 一样，Dave 也住在乡村，工作之余，常常凝望窗外的小鸟和树木。业余时间，他喜欢听昂贵的 Hi-Fi 和观看魔术。

# 前 言

## ASP+的故事

很久以前，还是在 1997 年 11 月下旬的时候，我们一些来自 Microsoft 的 Internet 信息服务器组的人员开始讨论创建网络应用软件的新方法。那时候 Active Server Pages (ASP) 才发布了一年，但令人难以置信的是，它正成为一项很流行的技术。但是，我们刚刚将 IIS 4.0 和 NT Option Pack 投入使用，那时才是假日的开始，人们刚刚从长期的劳苦中解脱出来，办公室里格外宁静。因此，Scott Guthrie 和我利用这个时候开始讨论我们对 ASP 满意和不满意的地方。当然我们喜欢 ASP 开发应用程序的简便性和它所提供的灵活性，但是还有很多问题需要思考。

我们达成的第一个共识就是，ASP 太复杂了，你必须编写非常多的代码。当我们试图告诉人们这些情况的时候，他们通常都无法理解我们在说些什么。“你们是什么意思？ASP 并不复杂，它使用起来非常简单！”他们是对的——ASP 很简单。我们看到过一些非常庞大的 ASP 页面，许多的服务器脚本代码与客户脚本代码混杂在一起，不久以后就很难判断发生了什么事情。是的，代码是简单而且很直接，但是太多了！

除此之外，用 ASP 做任何事情你都必须编写代码——做任何事情都没有纯粹的说明性方法。例如，如果你要编写一个广告 rotator 组件，就不能只是将它放在页面上就让它运行。你必须编写代码。原因就是 ASP 没有实际的组件模型。实际上由于 ASP 页面定义的都是单个的操作，制作一个组件模型就更加困难了。你必须从页面的顶部开始执行数据库查询和商务逻辑直到底部从而生成 HTML。因此，一个组件在 HTML 中的位置决定了它什么时候开始运行。

因此，如果一个位于页面底部的组件需要在页面顶部组件之前执行某些操作（假定是要生成其它组件使用的数据），你就必须重新构建代码来实现这个目的。另一方面，如果你想一下 VB Form 是如何工作的，就会发现它拥有一套控件，该 Form 通过调用其程序和引发某些事件与它们进行通话。所有这些控件都可以人为创建，可以更新内容、进行数据连接并读取数据等等。

工具支持是另外一个问题。缺乏组件支持和事件驱动模型会使 ASP 工具的开发变得非常困难。我们相信为了使 Web 应用程序的开发真正变得简单，我们需要更强大的工具支持。我们希望能对 Web 做出点贡献，就像 VB 为 Windows 所提供的——这样使任何人都可以编写具有丰富内容的应用程序。Visual InterDev 6 为开发者提供了这样一个模型，但是为了实

现此目的，必须使用大量的引用并生成大量的脚本代码。诸如 Visual Basic 的工具在使用 ASP 时面临着另外一个困难，就是 ASP 只支持解释性的脚本语言。当然，VB 的用户希望能够使用真正的 VB 代码！

在与客户的对话中，我们发现了很多问题。例如，代码与说明混杂在一起对很多人来说都是一个真正的问题，特别是当设计者和开发者进行同一个项目的时候。这使得支持多用户类型和应用程序国际化这样的方案很难实现。

12 月的月末总是一个反思的时候，也是为新一年做出某些决定并发誓改变一些坏习惯的时候。Scott 和我也发了一个誓。我们讨论过很多次，但是我们还没有写出任何代码！因此，我们互相许诺要建立一个模型，用来将我们正在研究的理论演示出来。Scott 在新年的周末回了家，1998 年 1 月回来并带回了一个模型。

这个模型以及它所包含的内容都是非常有用的，它允许我们验证我们的想法，并与其他人交流看法。当然，我们将它第一个演示给我们的经理：David Treadwell 和他的老板，J Allard。现在每个人都已经回来开始工作并且准备投入其中，对我们来说重要的是要表明“嘿，我们已经有了非常好的原材料，我们要继续工作下去！”他们对此表现出令人惊讶的热情，鼓励我们继续干下去。最终他们甚至给了我们财力支持使我们能够真正建立起来我们的系统。

当我们真正开始构建 ASP+ 的时候，我们要做的第一个也是最重要的一个决定就是如何构建。在 Microsoft 我们是第一个完全在 NGWS Runtime 上开始这项工作的小组。我们选择它是因为它可以提供最好的开发环境，不仅仅是为我们，而且是为那些使用 ASP+ 创建应用程序的用户。事实证明，在此 Runtime 中进行开发是一个非常好的选择。我们的效率很高，基础系统的广泛功能是 ASP+ 功能完备的一个重要原因。如果没有它我们真的无法编写 ASP+。另外 ASP+ 完全是用 C# 编写的，事实证明它是一个创建高效的，基于组件系统的非常好的语言。

从第一个模型开始的两年半时间，这个小组——现在已经是一个非常大的组织了——一直在致力于为 Web 应用程序开发一个广阔坚固的平台。除了我们前面列出的问题，我还想到了许多有关 ASP+ 和它的开发过程的问题，如下所示：

□ 分解设计。我们最好的一个设计决策就是将 ASP+ 完全分解为一套组件。这使我们很容易对这个模型进行扩展，使它能够执行以下功能：低级 HTTP 处理程序，Web 服务，插件认证组件以及可扩展状态服务。

□ 可升级性。我们努力创建一个高度可升级的模型。关键问题就是要创建可升级状态服务，能够在网络农场和网络公园里使用会话状态定位问题，这是一项使用多处理器运行一个给定应用程序的技术，每一个处理器都与一个多进程单元中的 CPU 相连。

□ 可用性。我们尽力使 ASP+ 具有高度的可用性。这包括一个新的和高性能的处理模型，这个模型能够检测到失效事件，死锁、内存泄漏并且进行修复。这样 ASP+ 应用程序永远都不会失效！

□ 使用工具。为了真正做到“就像 VB 提供给 Windows 的那样为 Web 做点什么”，最



重要的就是我们拥有强大的以 ASP+为目标的工具。幸运的是，Visual Studio 工作组有同样的想法，并投入了相当多的精力来创建 ASP+工具并帮助创建 ASP+。我们将 ASP+的设计视为与 VS 工作组的协作项目，ASP+当然也因此而成为一个更强大的产品。最终的结果就是 Visual Studio 7 成为创建 Web 应用程序和使用 ASP+的服务的一个非常好的工具。

最后，所有这些特点中最突出的一点就是——最终使开发者能够得心应手地使用 ASP+。为了真正掌握 ASP+，有一些介绍它的书是很重要的。那么这些书是怎么来的呢？我们于 1999 年 9 月开始这项工作。在华盛顿的 Wrox 会议上 Scott Guthrie 和我首次公开演示 ASP+。就是在那里我们第一次向从 Wrox 来的 John Franklin 提起要写一本书。我们认为 Wrox 是最好的选择，因为他们有过编写 ASP 和 Windows DNA 的历史。当我后来在 1999 年 11 月他们的伦敦会议上再次提出 ASP+的时候，我有机会与所有的作者共进午餐并就书的事情进行了讨论。他们都非常有热情。当然在那个时候，他们对 ASP+还所知甚少。他们不知道 NGWS Runtime, Visual Studio, C#或者我们正在开发的其它技术。因此，实际上他们并不知道他们要做的是做什么。

这样，简单地说，你已经知道了 ASP+的故事。结束之前，我要感谢两组人——其中一组是因为他们已经做过的事情，另一组是因为他们将要做的事情。首先，我要感谢所有开发 ASP+的人。他们是一个了不起的团体，做了非常优秀的工作。我为他们所有的人而骄傲。最后，我要感谢（事先）所有使用 ASP+开发下一代 Web 应用程序和服务的开发者。我们已经花了很多时间来设计它，目标就是让你能够顺利并愉快地使用它。

**Mark Anders**  
产品部经理  
NGWS Frameworks

## 简介

Microsoft 的 **Active Server Pages** 技术仍然是一个创建动态网络站点和基于网络的分布式应用程序的相当新的方法。但是，在它短暂的生命旅途中，它已经发展成为面向 Windows 的 Web 程序员的工具箱中最重要的工具。这可能是因为使复杂网页和应用程序的创建变得更简单，并且可以通过**组件对象模型 (COM/COM+)** 结构来使用自定义组件和现有的 Microsoft 和第三方的商业组件。

但是，在计算机界有一些新的标准和技术。当然可能其中最具竞争性的就是**可扩展标识语言**，或者说 **XML**。它与 **XML 样式语言 (XSL)**、**XML Schemas** 和**简单对象存取协议 (SOAP)** 等相关的技术一起，迅速地改变了我们设计和构建各种类型的分布式应用程序的思维方式。

### 一种新的 ASP

在编写本书的时候，Microsoft 正在开发下一代的 ASP，暂时叫做 **ASP+下一代 Web 服务**。应用它可更容易创建动态和可升级的 Web 应用程序，特别是在用户交互、数据处理和异步服务等公用领域里。虽然它改变很大，不过你所使用的技术和你现有的专业知识还是非常有用的。这项新技术旨在：

- **简化开发**。虽然 ASP 非常容易上手，并且相当简明易懂，但是如果一个网页比较复杂的话，它就会很杂乱。而 ASP+甚至可以使最复杂的页面也容易编写。

- **语言的独立性**。虽然 ASP 允许使用不同的语言，但它仍然受到限制——只能使用脚本语言编写。而 ASP+允许使用编译语言，这提供了更好的性能和交叉语言的兼容性。

- **代码和内容的分离**。在许多环境下，开发一个 Web 应用程序要用到很广泛的专业知识——例如，需要程序员编写代码，需要设计师来使 HTML 变得更美观。代码和内容都混杂在一个两者都要在其中执行的文件里，这使得它们很难联系起来完成同一件事。而 ASP+允许代码和内容的分离。

- **简化使用**。目前我们在使用组件方面遇到了一些困难，这些组件都必须注册并且在使用的时候会被操作系统锁定。而且还必须通过特殊的应用软件进行配置。所有这些因素都使你难以将你的应用程序做为一个单独的实体使用、移动或者复制。

- **支持多用户类型**。要创建适用于所有使用网络的不同类型的用户的网页是很难的。ASP+提供了数量众多，内容丰富的服务器端组件，它可以自动生成输出，特别是可以针对每一种类型的用户生成不同的输出。

□ **支持“下一代” Web 服务。** ASP+提供了一个新的“Web 服务”特性，它允许对使用和生成 XML 的支持创建服务程序。

□ **提高可用性。** ASP+更能够容忍应用程序的失效、内存泄漏等问题。它可以主动重新启动无法提高可靠性和可用性的进程。

□ **提高可升级性。** 新的会话状态特性使得在网络农场（有多个服务器）和“网络花园”（有一个服务器和多个处理器）里创建应用程序更为容易。

□ **提高工具支持能力。** ASP+所拥有的功能强大的、基于组件的、事件驱动的编程模型使 Microsoft 提供的工具（例如 Visual Studio）和其它供应商提供的工具变得更容易进行集成。这使得开发过程更像是创建传统的应用程序，开发环境也可以更好地相互作用，这使工作变得更容易。

出现这些变化是因为 ASP+的结构更加模块化。每一个页面都变成一个可通过编程得到的并且完全可以编译的运行时间对象，可以利用多种技术，例如面向对象的设计，及时编译，运行时间支持和动态缓存。同时，ASP+的向后兼容性意味着现有的页面和应用程序能够始终如一地使用，并且 ASP+会逐渐地增加新的特性。你可以在运行 ASP+页面的同时运行现有的 ASP 页面并按照你自己的情况转到 ASP+。

## 一个新的运行时间环境

在第 1 章中，我们对 ASP+的特征进行了概述，描述了它所包含的新的结构和概念。但是，ASP+只是一个更大的 Windows 系统服务的一部分。它依赖于一个新的运行时间系统，叫做下一代 **Web 服务运行时间**（NGWS Runtime）。这个运行时间为 ASP+和其它应用程序和服务程序提供了一个全新的、高效的和可升级的执行环境。

这个新的运行时间是构建 ASP+的基础，它提供了创建新一代的基于网络的分布式应用程序和动态网络站点所需要的所有功能。然而它们的紧密结合意味着你需要更多地了解一些当前的运行情况以用 ASP+获得最好的效果。而且，在第 1 章中，我们还将讨论它所包含的主要问题。

## 本书的内容

本书写于 Microsoft 将要发布 ASP+的第一个预览版本的时候。这个版本几乎具有全部的特性，并且足够稳定，使开发者可以开始学习并使用这项新技术。虽然我们不能保证最终的版本也与此相似，不过可以保证我们所介绍的所有的概念，示例和解释直到第一个 ASP+的完全版本都是准确无误的。

在本书中，我们试着解释 ASP+究竟是什么，如何使用它，以及你可以用它来做什么。我们在第 1 章中介绍了 ASP+的总体概念，为什么要有这些变化以及发生了什么样的变化，还有为了有效使用它所必须知道的东西。我们还简短地介绍了 NGWS 运行时间框架，并介

绍了它是如何和我们编写的 ASP+代码结合在一起的。

**第 2 章**将继续介绍我们如何以和 ASP 大致相同的方式用 ASP+ 创建动态网页。不过, ASP+ 的新结构允许我们用更少的代码完成更多的功能——这是程序设计员梦寐以求的事情。我们将向你演示 ASP+ 如何使代表 HTML 控件和组件的对象在服务器上执行, 所需的所有支持的 HTML 和代码都会自动生成。

为了对这些基于服务器的控件的功能进行扩展, 我们可以使用模板和服务器端数据连接以修改创建的代码。在**第 3 章**中, 我们介绍了如何将这些特性加入到你的页面中去。ASP+ 将使用一种新的数据存取技术叫做 ADO+, 它与数据连接过程相结合, 提供了真正的断开和多种记录集功能。

在**第 4 章**中, 我们介绍并描述了一些高级 ASP+ 编程技巧。这包括新的输入验证功能, 可重复使用的“pagelet”组件, 输出缓存支持, 以及错误处理和页面调试的技巧。

**第 5 章**介绍了另一种新技术叫做 ASP+Web 服务。它向你演示了如何创建可以通过 Internet 获取的 XML 服务。与 ASP+ 中许多新的服务器端组件一样, 这些技术可以使你无需使用我们过去使用的繁琐方法就可以创建出强大的 B2B 应用程序。

在**第 6 章**中, 我们将就 ASP+ 应用程序框架更深入的技术进行介绍。为了更充分地利用 ASP+, 你需要进一步了解请求和响应是如何被管理的, ASP+ 应用程序的配置与现有的技术有何不同。令人欣喜的就是它有一个更具可扩展性和延展性的管理系统, 从而变得比以前更易掌握了。

ASP+ 所提供的服务器端控件无疑是强大的和有用的, 有时你需要创建自己的控件或者对它提供的控件进行扩展。这也许是为了得到新的功能, 或者是为了覆盖更广范围的不同类型的客户端用户代理。**第 7 章**将介绍如何建立我们自己的服务器端控件, 并且将其添加到 ASP+ 环境中去。

最后, 在**第 8 章**中, 我们介绍了一个有关 ASP+ 使用的例子, 它是 Microsoft 设计并创建的一个应用程序。它使用了许多 ASP+ 的新特性以及通用的运行时间。它会使你对如何创建完整应用程序有一个感性认识。

## 本书的读者对象

本书所介绍的产品还处于开发阶段, 因此它的对象是走在最前沿的高级 ASP 开发人员——而不是普通的 ASP 开发人员或是初学者。举例来说, 我们将不会介绍本书中使用的 COM、ASP 或编程语言和脚本语言的基础知识。

实际上(在编写本书的时候)此产品还是一个预览版本, 但这并不意味着你可以忽视它。ASP+ 将会使用新的 NGWS 运行时间软件包, 它可能也会成为下一个版本的 Visual Studio (它包括使用 ASP+ 时可能用到的工具)的一部分。

这里我们的目的就是对此产品进行一个总体概述——包括一些背景理论和对该产品开发思路的说明。然后将进一步深入研究开发者首先要使用的一些功能。我们还介绍了如何快

速正确地使用这些新特性。

如果你以前使用过 ASP，并且对这些概念相当了解，你就能够直接使用这本书而不需要任何其它参考资料（除了 SDK 和本产品提供的帮助文件）。你还需要充分掌握使用 COM 来例示以及使用组件和 Visual Basic（或者 VBScript）语言的基本原理。某些例子是用其它语言写的，例如 C++，JScript 和 C#（读做 C Sharp），通用的运行时间框架中都支持这些语言，但是使用本书不需要精通这些语言。

## 致谢

我们感谢软件制造商为我们编写本书提供的技术支持和几乎所有的信息，我们必须承认本书是在一个特定的情况下产生的。从这项技术诞生起 Wrox 就一直走在 ASP 出版的最前面，我们非常感谢 Microsoft 的开发者和产品经理为我们提供的长期支持。

本书诞生于 ASP+ 开发早期。当然如果没有 Microsoft 的 ASP 和 COM+ 工作组的许多成员提供的慷慨和及时的帮助，本书是不可能完成的，特别是 Mark Anders, Scott Guthrie, Brad Merrill, Brad Millington, Eric Andrae 和 ASP+ 小组的其它人员。

感谢你们，所有的人——希望你们喜欢这个成果。

## 使用这本书你需要什么

要能够运行所有的例子，你需要安装以下软件：

- Windows 2000
- Internet Explorer 5.5
- NGWS Runtime
- SQL Server 7

本书所有的源代码都可以从我们的 WWW 站点下载 <http://www.wrox.com>。本书的更新源代码可以从 <http://www.wrox.com/beta> 获得。

## 约定

为了帮助你更好地理解本书的内容并明白当前的进程，我们在本书中使用了一些约定。例如：

这些方框包含非常重要的与上下文直接相关的信息，需时刻记住。

*此背景样式用于当前的讨论。*

本文中的样式如下所示：

- 我们介绍时会将重要的词以**黑体显示**。
- 我们将键盘按键这样表示：***Ctrl-A***。
- 我们将本文中的文件名和代码这样表示：**doGet ()**。
- 用户界面（和 URL）的文本这样表示：**Menu**。

我们用以下两种方式表示代码：

在我们的代码示例中，阴影样式的代码表示新的、重要的、相关的代码

而没有阴影的代码表示在上下文中比较不重要的代码，或者前面已有的代码。

## 告诉我们你的想法

我们尽力使这本书对你尽可能地有用，因此我们希望知道你的想法。我们一直希望知道你想要什么以及你需要知道什么。

我们希望我们的努力得到反馈，并将批评和赞扬在我们将来的编辑工作中加以体现。如果你有什么想法，通过 E-mail 通知我们：

**feedback@wrox.com**

也可以使用我们站点上的反馈链接：

**http://www.wrox.com**

# 目 录

<b>第 1 章 ASP+简介</b> .....	1
ASP 的发展 .....	1
动态服务器端网络编程 .....	2
ASP 的版本 .....	3
下一代网络 Web 框架 .....	4
NGWS 框架是什么 .....	4
ASP+与 ASP 有什么不同 .....	7
我们为何需要一个新版 ASP+ .....	7
ASP+最大的优势 .....	8
新特性列表 .....	19
ASP+页面 .....	20
ASP+列表控件 .....	23
ASP+Web 服务 .....	25
ASP+配置和使用 .....	25
ASP+应用程序和会话状态 .....	26
ASP+的错误处理、调试和跟踪 .....	27
其它的 ASP+特性 .....	29
开始使用 ASP+ .....	29
安装 ASP+ .....	30
创建一个 ASP+应用程序 .....	30
检验安装 .....	31
关于 ASP+最终版本 .....	32
小结 .....	33
<b>第 2 章 ASP+页面</b> .....	35
编码问题 .....	35
旧的编码方法 .....	35
ASP+页面中的编码 .....	37
Web 控件 .....	43
HTML 控件或 ASP+控件 .....	44
使用 Web 控件 .....	44

控件家族 .....	46
内部控制 .....	46
增强控件 .....	60
移动通信控件 .....	67
自定义控件 .....	67
小结 .....	68
<b>第 3 章 数据和数据处理 .....</b>	<b>69</b>
ADO+是什么 .....	69
Web 应用程序中的数据 .....	70
ADO+ 的使用 .....	70
数据连接 .....	77
数组 .....	77
与 XML 数据相连接 .....	82
与数据库数据相连接 .....	84
数据网格控件 .....	89
自定义外观 .....	90
使用模板 .....	92
编辑数据 .....	96
高级数据连接 .....	100
控件 .....	102
小结 .....	103
<b>第 4 章 高级 ASP+ 页面技术 .....</b>	<b>104</b>
输入验证技术 .....	104
ASP+ 验证控件概述 .....	105
ASP+ 验证控件总结 .....	106
使用 ASP+ 验证控件 .....	112
编写“隐含代码” .....	123
使用“隐含代码” .....	124
隐含代码范例 .....	126
用小页面建立可重用控件 .....	127
建立小页面控件 .....	128
小页面示例 .....	128
使用 ASP+ 缓存和输出缓存 .....	134
输出缓存 .....	134



ASP+缓存 .....	137
ASP+缓存示例 .....	138
错误处理、跟踪与调试 .....	140
自定义错误页面 .....	140
页面级跟踪 .....	142
应用级跟踪 .....	145
ASP+调试器 .....	146
小结 .....	147
<b>第 5 章 Web 服务 .....</b>	<b>148</b>
什么是 Web 服务 .....	148
体系结构 .....	149
编写探寻代码 .....	150
建立 ASP+ Web 服务 .....	151
商务逻辑 .....	151
建立 .asmx 文件 .....	152
通过 Web 浏览器测试 .....	154
使用 DefaultSdlHelpGenerator.aspx .....	154
.asmx 文件语法 .....	155
Web 服务协议 .....	158
ASP+支持的协议 .....	158
数据类型 .....	161
Web 服务描述 .....	162
服务描述语言 (SDL) .....	162
建立 SDL 文档 .....	163
编程访问我们的服务 .....	164
建立代理类 .....	165
使用 WebServiceUtil.exe .....	166
从 ASP+ 页面调用 MathWeb 服务 .....	168
线上传输 .....	169
小结 .....	171
<b>第 6 章 应用框架与服务 .....</b>	<b>173</b>
使用全局应用文件: global.asax .....	173
全局应用文件是什么样的 .....	173
global.asax 存在哪里 .....	174