



Beginning ASP.NET 2.0 AJAX

ASP.NET 2.0 AJAX 入门经典

Wallace B. McClure
(美) Paul Glavich
Steve C. Orr
吴宏泉

著
译



清华大学出版社



TP393.092/1064

2008

ASP.NET 2.0 AJAX 入门经典

Wallace B. McClure

(美) Paul Glavich

等著

Steve C. Orr

吴宏泉

译

清华大学出版社

北京

Wallace B. McClure, Paul Glavich, Steve C. Orr, Craig Shoemaker, Steven A. Smith, Jim Zimmerman
Beginning ASP.NET 2.0 AJAX
EISBN: 978-0-470-11283-0
Copyright © 2007 by Wiley Publishing, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2008-1930

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET 2.0 AJAX 入门经典/(美)麦克卢尔(McClure, W. B.), (美)格拉维奇(Glavich, P.), (美)欧尔(Orr, S. C.)等著; 吴宏泉 译. —北京: 清华大学出版社, 2008.6

书名原文: Beginning ASP.NET 2.0 AJAX

ISBN 978-7-302-17728-9

I. A… II. ①麦…②格…③欧…④吴… III. ①主页制作—程序设计②计算机网络—程序设计
IV. TP393.092 TP393.09

中国版本图书馆 CIP 数据核字(2008)第 075230 号

责任编辑: 王 军 郑雪梅

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 清华大学印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 21 字 数: 511 千字

版 次: 2008 年 6 月第 1 版 印 次: 2008 年 6 月第 1 次印刷

印 数: 1~4000

定 价: 39.99 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 025683-01

关于作者

Wallace B. Wally McClure 于 1990 年毕业于 Georgia Institute of Technology 学院, 并获得了电气工程科学学士学位。之后他继续在那里深造, 并于 1991 年获得了同一领域的硕士学位。之后, 他先后在几个公司中担任了顾问和开发的职位, 这些公司包括 The United States Department of Education、Coca-Cola、Magnatron 和 Lucent Technologies, 其中还包括一个 Web 搜索引擎公司和一个 Web 2.0 公司。他参与的产品和服务所涉及的技术包括 ASP、ADO、XML 和 SQL Server, 以及各种架构在 Microsoft .NET Framework 上的应用程序。自 2000 年夏天, Wally 就一直工作于 .NET Framework。他是构建应用程序方面的专家, 具有大量的用户及大量的数据。他是 Microsoft 的 MVP、ASPInsider、INETA Speaker's Bureau 的成员, 以及 Scalable Development, Inc. 公司的合伙人。Wally 开通了自己的博客, 网址为 <http://weblogs.asp.net/wallym/> 和 www.morewally.com。他和合著者 Paul Glavich 一起共同维护 ASP.NET Podcast 站点, 在 www.aspnetpodcast.com 上有他们的音频广播。除此之外, Wally 还穿梭于美国的东南部对用户进行指导, 并参与了各种 CodeCamps 活动。

当不工作时, Wally 会在家里陪伴他的妻子 Ronda 和两个孩子, Kirsten 和 Bradley。偶尔, 他也会打高尔夫, 并且在 2005 年 7 月 30 日, 他平生第一次在一个真实的高尔夫比赛中打破了标准杆数, 他以前从没有得过这么好的成绩, 使他觉得难以置信。

Paul Glavich 目前是 ASP.NET 的 MVP, 并且是 Readify 公司的资深顾问。以前, 他是 EDS Australia 的一个技术设计师, 他具有 15 年的业界经验, 所涉及的技术从 PICK、C、C++、Delphi 和 Visual Basic 3/4/5/6 到目前的 .NET——C#、COM+ 和 ASP.NET。

Paul 从 .NET 最初处于 Beta 测试时就开始使用这种技术进行开发, 并且是世界上第一个使用 .NET 技术实现 Internet 银行解决方案的技术设计师。

在各种 .NET 相关的新闻组中都可以看到 Paul 的文章, 他还出席悉尼的 .NET 用户组 (www.sdnug.org) 和 TechEd, 他也是 ASPInsiders (www.aspinsiders.com) 的理事会成员。Paul 还在一些开发团体站点上发表一些技术文章, 如 ASPAlliance.com (www.aspalliance.com)。他还撰写了一本名为 *Beginning AJAX in ASP.NET* 的图书, 并和别人合作撰写了 *Microsoft ASP.NET AJAX* 一书, 他现在致力于 Microsoft ASP.NET AJAX 和 Windows Communication Foundation 技术。

Steven C. Orr 是 ASP Insider、Microsoft Certified Solutions Developer (MCSD)、Certified ScrumMaster (CSM) 和 ASP.NET 方面的 Microsoft MVP。他是 Microsoft 技术专家, 这些技术包括 ASP.NET、Visual Basic .NET、C# 和 SQL Server。他闻名于他在 *ASP.NET Pro Magazine* 月刊上的 Control Freak 栏目, 而且 20 年来已经为西雅图区域的许多顶尖公司开发了一些软件解决方案。当不忙于设计软件系统或者撰写技术文章时, Steve 经常在本地的用户组中闲逛, 并且经常在 ASP.NET 的论坛中进行浏览和评论。

Craig Shoemaker 在教授软件开发人员有关面向对象开发、系统架构和在 .NET 中的最

佳实践技术方面的知识。除此之外，他是 Polymorphic Podcast(polymorphicpodcast.com)的主管，而且是.NET 开发人员团体的积极分子。Craig 是 *Beginning Ajax with ASP.NET* (Wrox) 一书的合著者，他是 *Ajax Design Patterns*(O'Reilly)一书的主要撰写人，并是 *CoDe Magazine*、*ASP Alliance* 和 PDSA eBook 系列的主要作者。Craig 的个人活动包括在 VSLive! 上演讲，参与 Southern California Code Camp、Southern California .NET Architecture User Group 和 Podcast and Portable Media Expo 的活动。作为一位全职的 PDSA, Inc.(pdsa.com) 的软件工程师，Craig 的开发经验涉及的范围包括娱乐和金融部分，与 PDSA 的客户一起工作给了他为一一些一流的组织提供咨询服务的机会，这些组织包括 Twentieth Century Fox Film Corporation 和 City of Hope Cancer Research Center。

Steven A. Smith 是 ASPAlliance.com 和 DevAdvice.com 的主席，他是 Microsoft Regional Developer、Microsoft ASP.NET MVP 以及 ASPInsiders 的理事会成员。他还是 International .NET Association (INETA) Speaker Bureau 的成员，并且著作了两本有关 ASP.NET 的图书。Steve 曾经是一个军方工程官员。他和他的妻子及商务伙伴 Michelle 和他们的女儿 Ilyana 住在俄亥俄州的 Kent。当不坐在电脑前时，Steven 喜欢和他的家庭去徒步旅行、骑自行车和做游戏。

Jim Zimmerman 目前是 Visual Developer——ASP/ASP.NET MVP。他在佛罗里达州的 Code Camps 和 .NET 用户组中做和 .NET 有关的演讲，内容包括 AJAX 和 Code Generation。Jim 是 Ajax Control Toolkit (www.codeplex.com/AtlasControlToolkit)团队的一员，并且会在孩子睡觉时撰写自己的博客，网址为 www.jimzimmerman.com/blog。他有一个软件顾问公司，致力于几个在线 Web 应用，包括他拥有股份的 CarCentral(www.carcentral.com)。他具有 10 年以上的 Web 开发经验，以及使用各种语言的经验，包括 Perl、PHP、Java 和 Visual Basic。在过去的 3 年中，Jim 使用 C#编写了大部分 Web 应用程序，并且目前正在专注于可升级的 Web 应用程序开发，使用的技术包括 ASP.NET 2.0、SQL Server 2005、C#、AJAX 和 Team Foundation Server。

在空闲时，Jim 喜欢和妻子及孩子在佛罗里达州的 Tampa 的沙滩上玩，并喜欢弹吉他。

前言

感谢购买《ASP.NET 2.0 AJAX 入门经典》一书，我们知道在购买一本编程图书时是有很多选择的，很高兴您能选择我们的书籍。我们保证您会对本书中的高质量内容满意，就像对 Wrox 出版的其他书籍所期望的一样。

ASP.NET 2.0 AJAX 框架是一套基于浏览器的 Web 技术，这会为 Web 应用程序的设计带来一场革命。它会为应用程序所使用的方法带来根本性改变，还可以给用户提供了反应迅速的应用程序，并为开发人员构建应用程序提供了另一种选择。本书致力于 ASP.NET 2.0 AJAX 框架的研究，并且介绍了基于浏览器的 .NET 框架，这个框架可以集成 ASP.NET 服务器端编程方法，并同时提供客户端服务。我们坚信本书可以满足您有关在 ASP.NET 平台上使用 ASP.NET 2.0 AJAX 编程的需求。

本书读者对象

本书适用于正在使用 ASP.NET 开发环境的开发人员，而且此开发人员正在寻求使用 AJAX 技术创建响应更快和更现代的应用程序的方法，这种应用程序的工作方式和传统的桌面应用程序的工作方式是很类似的。如果开发人员想要改善他们已有应用程序的用户体验，开发新的应用程序，开发内部的商务营运应用程序，以及想要使用全世界的开发人员正在谈论的热门技术时，那么本书可以提供很大的帮助。

本书面向使用 ASP.NET，并且正在开始使用 ASP.NET 2.0 AJAX 框架技术的开发人员。本书会帮助使用基于 ASP.NET 应用程序的开发人员来改善他们的应用程序和技术，可以为他们提供 ASP.NET 2.0 AJAX 框架的背景知识，并且引导他们如何把 ASP.NET 2.0 AJAX 框架应用到他们的应用程序中。

本书主要内容

这是一本有关 ASP.NET 2.0 AJAX 框架的入门级教材，您可能已经很熟悉 ASP.NET 了，但是对于 AJAX 框架可能还不是很了解。本书包括两个主要部分：纸制图书和在线内容。

我们觉得在任何时候都需要使用纸制图书，作为一个读者，您在购买本书时可能希望书中的代码在本书出版的第一天，以及 18 个月后都可以正常运行。我们会尽量保证本书中的代码尽可能长时间地保持有效。下面列出的是本书的主要内容：

- ASP.NET 2.0 AJAX 框架的体系结构。
- 如何把 ASP.NET 2.0 AJAX 框架应用到一个已有的应用程序中。

- Microsoft AJAX 框架的基本使用方法。本部分会介绍如何使用 Web 服务进行基本的 AJAX 风格的操作，并介绍如何使用 UpdatePanel、AJAX Control Toolkit 和其他在 ASP.NET 2.0 AJAX 框架中对于开发人员来说是全新的特性。
- Microsoft AJAX 框架的高级应用。本部分包括安全性、与 ASP.NET 服务集成，以及调试功能。

对于想要使用在 CTP(Community Technical Preview)中包含的 ASP.NET 2.0 AJAX 框架特性的开发人员来说该怎么办呢？他们的运气并不坏。本书的作者们觉得介绍在 12 个月以后也许变得不支持的特性是不合适的。但是，还有人期望应该介绍和这些特性有关材料，很多开发人员对此比较感兴趣。因此，我们提供了一些在线内容，这些内容会包括下面所列的材料：

- XML-Script
- 客户端数据绑定
- 拖放功能
- 桥接

除了提供在线内容外，我们还会基于产品的升级相应地保持在线内容的更新。

本书内容结构

本书分为下面几个章节：

- 第 1 章 “ASP.NET AJAX 综述”——本章主要介绍什么是 ASP.NET 2.0 AJAX 框架、这个产品的内容、在业界的开发背景，以及本书的组织结构。
- 第 2 章 “创建 ASP.NET AJAX 应用程序”——本章讨论的是如何创建一个 ASP.NET AJAX 应用程序，以及如何在一个已有的应用程序中应用 ASP.NET AJAX 功能。
- 第 3 章 “ASP.NET AJAX 体系结构”——本章介绍的是 ASP.NET AJAX 框架的各种特性是如何协同工作的。
- 第 4 章 “Web 服务调用”——调用 Web 服务器上的逻辑而不用回送到 Web 服务器是 ASP.NET AJAX 框架所提供的核心内容。本章介绍的是如何设置 Web 服务，调用 Web 服务，客户端代理的工作方式，客户端的调用顺序，以及在服务器端和客户端的数据交换。
- 第 5 章 “UpdatePanel 控件”——UpdatePanel 控件提供了一个简单的机制，可以用来在 ASP.NET 服务器控件和页面生命周期中集成 AJAX 功能。这个控件允许在一个应用程序中添加 AJAX 功能，而不用大规模地修改应用程序。它为应用程序提供了易于实现的 AJAX 功能，同时提供了 ASP.NET 开发人员已经熟悉的服务器端编程模型。
- 第 6 章 “Control Toolkit”——虽然 ASP.NET AJAX 框架包含了一套图形用户界面控件，但其中还包括了一套二级控件。这套控件是单独下载的，并被认为是 ASP.NET AJAX 的一部分，因此介绍一下这些控件的使用方式还是有必要的。

- 第7章“控件扩展器”——控件扩展器可以在客户端 Web 浏览器运行时扩展服务器端控件的功能。开发人员可以使用这种技术在客户端控件中添加新功能，并可以集成服务器端的数据。
- 第8章“JavaScript 增强”——当使用 ASP.NET AJAX 进行编程时，理解 JavaScript 语言以及 ASP.NET AJAX 为 JavaScript 程序员提供的新功能是很重要的。
- 第9章“Microsoft AJAX Library”——ASP.NET AJAX 提供了构建在已有的 JavaScript 对象基础之上的一套扩展对象，这些对象提供了对许多 .NET Framework 特性的支持，例如事件调用机制和低级别的 HTTP 调用次序。
- 第10章“用户界面设计”——ASP.NET AJAX 可以异步发出调用，因此在一个 ASP.NET AJAX 操作发生时，用户可以继续工作，在此过程中为用户提供一些反馈信息是很重要的。本章介绍了几种策略来为用户提供有关后台所发生的操作的反馈信息。
- 第11章“安全性和集成”——安全性对任何人来说都很重要，ASP.NET 提供了一套服务(如配置服务、成员资格服务和身份验证服务)。本章介绍的是如何使用 ASP.NET 2.0 AJAX 框架安全地使用这些服务。
- 第12章“调试”——没有代码是完美无缺的，对所有开发人员来说，了解应用程序的运行情况以及查找软件漏洞的能力是很关键的。
- 第13章“ASP.NET AJAX Futures CTP: 在线内容”——本章提供了在线内容的简短描述。

学习本书所需要的准备工作

下面所列的是运行本书示例的系统要求：

- Visual Studio .NET 2005 或者免费的 Visual Web Developer。
- Windows Vista、Windows XP 或者 Windows 2003 Server。
- 一个现代的 Web 浏览器，如 Internet Explorer 的最新版本、Mozilla Firefox 或者 Apple 的 Safari。
- ASP.NET 2.0 AJAX 框架插件——访问 <http://ajax.asp.net> 可以得到有关此插件的详细信息，包括如何下载。

本书中的示例是按照下面条件编写的：

- 所有的服务器端代码都是以 C# 语言编写的。
- 所有的客户端代码都是以 JavaScript 语言编写的。
- 在 Visual Web Developer for .NET 2.0 中可以打开所有 ASP.NET 示例。

如何下载本书的示例代码

在读者学习本书中的示例时，可以手动输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书的合作站点 <http://www.wrox.com/> 或

www.tupwk.com.cn/downloadpage 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

许多图书的书名都很相似，所以通过 ISBN 查找本书是最简单的，本书的 ISBN 是 978-0-470-11283-0。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@tup.tsinghua.edu.cn 发电子邮件，我们会检查您的信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

p2p.wrox.com

P2P 邮件列表是为作者和读者之间的讨论而建立的。读者可以在 p2p.wrox.com 上加入 P2P 论坛。该论坛是一个基于 Web 的系统，用于传送与 Wrox 图书相关的信息和相关技术，与其他读者和技术用户交流。该论坛提供了订阅功能，当论坛上有新贴子时，会给您发送您选择的主题。Wrox 作者、编辑和其他业界专家和读者都会在这个论坛上进行讨论。

在 <http://p2p.wrox.com> 上有许多不同的论坛，帮助读者阅读本书，在读者开发自己的应用程序时，也可以从这个论坛中获益。要加入这个论坛，需执行下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读其内容，单击 Agree 按钮。
- (3) 提供加入论坛所需的信息及愿意提供的可选信息，单击 Submit 按钮。

然后就可以收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

提示：

不加入 P2P 也可以阅读论坛上的信息，但只有加入论坛后，才能发送自己的信息。

加入论坛后，就可以发送新信息，回应其他用户的贴子。可以随时在 Web 上阅读信息。如果希望某个论坛给自己发送新信息，可以在论坛列表中单击该论坛对应的 **Subscribe to this Forum** 图标。

对于如何使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作原理，以及许多针对 P2P 和 Wrox 图书的常见问题解答。要阅读 FAQ，可以单击任意 P2P 页面上的 FAQ 链接。

目 录

第 1 章 ASP.NET AJAX 综述	1
1.1 开发趋势	1
1.2 问题所在	4
1.3 改善用户体验	4
1.4 现状	5
1.5 什么是 AJAX	5
1.5.1 AJAX 的优势	6
1.5.2 AJAX 发展历史	6
1.5.3 组成 AJAX 的技术	6
1.6 什么是 ASP.NET 2.0 AJAX	7
1.7 AJAX 的使用者	8
1.7.1 现状	9
1.7.2 程序包	9
1.7.3 AJAX 的未来	9
1.8 本章小结	10
第 2 章 创建 ASP.NET AJAX 应用程序	11
2.1 安装 ASP.NET AJAX	11
2.2 创建 ASP.NET AJAX 应用程序	13
2.2.1 创建一个管理用户和角色的数据库	15
2.2.2 创建一个数据访问层	16
2.2.3 创建一个角色管理页面	18
2.2.4 添加一个 UpdateProgress 控件	21
2.3 在已有应用程序中添加 ASP.NET AJAX 功能	22
2.3.1 添加 ScriptManager 控件	28
2.3.2 添加 UpdatePanel 控件	29

2.3.3 添加 AutoCompleteExtender 控件	29
2.4 本章小结	31
第 3 章 ASP.NET AJAX 体系结构	33
3.1 ASP.NET AJAX 基础	33
3.2 理解代码库	34
3.3 实现跨浏览器兼容	35
3.4 探索 Microsoft AJAX Library	36
3.5 客户端事件生命周期	37
3.6 服务器端控件和组件	38
3.7 检测 JavaScript 文件	41
3.8 操控 HTTP 处理程序	42
3.9 使用 web.config 文件	43
3.10 本章小结	50
第 4 章 Web 服务调用	51
4.1 AJAX 介绍	51
4.1.1 使用 ASP.NET AJAX 开发的优势	52
4.1.2 使用 ASP.NET AJAX 开发的劣势	52
4.1.3 各种 AJAX 库	53
4.2 客户端 JavaScript 概述	53
4.2.1 什么是 JavaScript	54
4.2.2 为什么需要 JavaScript	54
4.2.3 常见控件引用	54
4.2.4 浏览器支持	56
4.3 客户端内容	56
4.3.1 ScriptManager	57
4.3.2 快捷方式	61
4.3.3 ScriptManagerProxy	61
4.3.4 XML-Script	62

4.4	服务器端操作	62	5.7.5	pageLoaded	96
4.5	客户端和服务端协同工作	64	5.7.6	endRequest	96
4.5.1	使用 ASP.NET 调用约定	64	5.7.7	init	97
4.5.2	执行异步操作	66	5.7.8	pageUnload	97
4.6	AJAX 应用	68	5.8	添加控件	99
4.7	返回数据	68	5.9	编制控件	105
4.7.1	简单数据类型	69	5.9.1	注册脚本	105
4.7.2	使用定制业务对象	70	5.9.2	初始化脚本	106
4.8	基于页面的 Web 服务	73	5.9.3	实施客户端处置功能	107
4.9	JSON 数据格式	74	5.10	其他控件	108
4.9.1	什么是 JSON	75	5.10.1	UpdateProgress 控件	108
4.9.2	为什么使用 JSON	76	5.10.2	Timer 控件	108
4.10	问题所在	76	5.11	本章小结	110
4.10.1	无效回送或回调参数	76	第 6 章	Control Toolkit	111
4.10.2	同步 ViewState	78	6.1	开发团体项目	111
4.10.3	出错处理	79	6.2	安装工具集	112
4.11	Futures CTP	82	6.3	测试工具集控件	116
4.12	本章小结	82	6.3.1	扩展器控件	116
第 5 章	UpdatePanel 控件	83	6.3.2	控件基本使用方法	117
5.1	什么是 UpdatePanel	83	6.4	测试控件	119
5.2	理解异步回送	84	6.4.1	控件列表	119
5.3	UpdatePanel 的浏览器支持	85	6.4.2	应用控件	126
5.4	配置 UpdatePanel	86	6.5	创建扩展器控件	149
5.4.1	ScriptManager	87	6.6	为 AJAX Control Toolkit 开放 源代码项目做出贡献	149
5.4.2	ScriptManager 的方法和属性	88	6.7	本章小结	150
5.4.3	触发 UpdatePanel	88	第 7 章	控件扩展器	151
5.4.4	<asp:UpdateProgress>	90	7.1	核心框架	151
5.4.5	终止按钮	90	7.2	创建一个控件	152
5.4.6	ChildrenAsTriggers	91	7.2.1	创建项目	153
5.4.7	UpdateMode	91	7.2.2	嵌入脚本资源	155
5.4.8	RenderMode	91	7.2.3	实施抽象方法	157
5.5	服务器端调试	91	7.2.4	提供动态行为实现	160
5.6	动态添加 UpdatePanel	92	7.2.5	完成的扩展器	167
5.7	客户端页面生命周期	94	7.2.6	ASP.NET AJAX Control Toolkit	167
5.7.1	PageRequestManager	95	7.2.7	使用 AJAX Control Toolkit 增强扩展器	171
5.7.2	initializeRequest	95			
5.7.3	beginRequest	95			
5.7.4	pageLoading	96			

7.2.8	AJAX Control Toolkit 效果	174	10.4	在操作进行中提供反馈	236
7.3	使用 AJAX Control Toolkit 的优缺点	175	10.5	中断操作	239
7.4	本章小结	176	10.6	操作异常	243
第 8 章	JavaScript 增强	179	10.7	本章小结	246
8.1	数据类型	179	第 11 章	安全和集成	249
8.1.1	Object() 对象	180	11.1	集成工作方式	249
8.1.2	Boolean() 对象	181	11.2	保护 Web 应用程序	249
8.1.3	Number() 对象	182	11.3	ASP.NET 成员资格服务	251
8.1.4	String() 对象	183	11.4	使用 Microsoft ASP.NET 2.0 AJAX	253
8.1.5	Array() 对象	184	11.5	身份验证服务	253
8.1.6	StringBuilder() 类	186	11.5.1	Login 函数	253
8.1.7	Date() 对象	188	11.5.2	Logout 函数	254
8.1.8	Error() 对象	190	11.5.3	启用身份验证服务	254
8.2	代码管理	191	11.5.4	小结	260
8.2.1	命名空间	191	11.6	配置服务	261
8.2.2	类	193	11.6.1	属性字段	261
8.2.3	继承	194	11.6.2	Load 函数	261
8.2.4	接口	196	11.6.3	Save 函数	261
8.2.5	集成语言特征	196	11.6.4	启用配置服务	262
8.2.6	枚举	198	11.7	本章小结	271
8.3	本章小结	199	第 12 章	调试	273
第 9 章	Microsoft AJAX Library	201	12.1	服务器端调试	273
9.1	客户端命名空间	201	12.2	使用断点	275
9.2	Sys 命名空间	202	12.3	JavaScript 调试	276
9.2.1	类型	202	12.3.1	其他调用调试器的方法	284
9.2.2	Sys.Debug	203	12.3.2	使用 debugger 关键字	285
9.2.3	Sys.StringBuilder	204	12.3.3	其他检测变量值的方法	287
9.3	Sys.Net 命名空间	211	12.4	ASP.NET AJAX 调试支持	291
9.4	Sys.Serialization 命名空间	219	12.4.1	使用 Sys.Debug	291
9.5	Microsoft AJAX Library 事件处理	224	12.4.2	断言和失败	297
9.6	本章小结	225	12.4.3	使用 Sys.Debug.assert	297
第 10 章	用户界面设计	227	12.4.4	使用 Sys.Debug.fail	299
10.1	异步和同步回送	227	12.4.5	ScriptManager 调试支持	299
10.2	用户处理通知	228	12.5	中间人	300
10.3	动画显示异步回送	233	12.6	使用 Web Development Helper	305
			12.7	在 Firefox 中调试	307

12.8 本章小结	310
-----------	-----

第 13 章 ASP.NET AJAX Futures CTP: 在线内容

13.1 XML-Script	313
13.2 数据绑定	314
13.3 拖放功能	315
13.4 桥接	316
13.5 Silverlight	317

附录 参考资料

11.1 本章小结	271
11.2 本章小结	271
11.3 本章小结	271
11.4 本章小结	271
11.5 本章小结	271
11.6 本章小结	271
11.7 本章小结	271
11.8 本章小结	271
11.9 本章小结	271
11.10 本章小结	271
11.11 本章小结	271
11.12 本章小结	271
11.13 本章小结	271
11.14 本章小结	271
11.15 本章小结	271
11.16 本章小结	271
11.17 本章小结	271
11.18 本章小结	271
11.19 本章小结	271
11.20 本章小结	271
11.21 本章小结	271
11.22 本章小结	271
11.23 本章小结	271
11.24 本章小结	271
11.25 本章小结	271
11.26 本章小结	271
11.27 本章小结	271
11.28 本章小结	271
11.29 本章小结	271
11.30 本章小结	271
11.31 本章小结	271
11.32 本章小结	271
11.33 本章小结	271
11.34 本章小结	271
11.35 本章小结	271
11.36 本章小结	271
11.37 本章小结	271
11.38 本章小结	271
11.39 本章小结	271
11.40 本章小结	271
11.41 本章小结	271
11.42 本章小结	271
11.43 本章小结	271
11.44 本章小结	271
11.45 本章小结	271
11.46 本章小结	271
11.47 本章小结	271
11.48 本章小结	271
11.49 本章小结	271
11.50 本章小结	271
11.51 本章小结	271
11.52 本章小结	271
11.53 本章小结	271
11.54 本章小结	271
11.55 本章小结	271
11.56 本章小结	271
11.57 本章小结	271
11.58 本章小结	271
11.59 本章小结	271
11.60 本章小结	271
11.61 本章小结	271
11.62 本章小结	271
11.63 本章小结	271
11.64 本章小结	271
11.65 本章小结	271
11.66 本章小结	271
11.67 本章小结	271
11.68 本章小结	271
11.69 本章小结	271
11.70 本章小结	271
11.71 本章小结	271
11.72 本章小结	271
11.73 本章小结	271
11.74 本章小结	271
11.75 本章小结	271
11.76 本章小结	271
11.77 本章小结	271
11.78 本章小结	271
11.79 本章小结	271
11.80 本章小结	271
11.81 本章小结	271
11.82 本章小结	271
11.83 本章小结	271
11.84 本章小结	271
11.85 本章小结	271
11.86 本章小结	271
11.87 本章小结	271
11.88 本章小结	271
11.89 本章小结	271
11.90 本章小结	271
11.91 本章小结	271
11.92 本章小结	271
11.93 本章小结	271
11.94 本章小结	271
11.95 本章小结	271
11.96 本章小结	271
11.97 本章小结	271
11.98 本章小结	271
11.99 本章小结	271
12.0 本章小结	271

12.1 本章小结	271
12.2 本章小结	271
12.3 本章小结	271
12.4 本章小结	271
12.5 本章小结	271
12.6 本章小结	271
12.7 本章小结	271
12.8 本章小结	271
12.9 本章小结	271
12.10 本章小结	271
12.11 本章小结	271
12.12 本章小结	271
12.13 本章小结	271
12.14 本章小结	271
12.15 本章小结	271
12.16 本章小结	271
12.17 本章小结	271
12.18 本章小结	271
12.19 本章小结	271
12.20 本章小结	271
12.21 本章小结	271
12.22 本章小结	271
12.23 本章小结	271
12.24 本章小结	271
12.25 本章小结	271
12.26 本章小结	271
12.27 本章小结	271
12.28 本章小结	271
12.29 本章小结	271
12.30 本章小结	271
12.31 本章小结	271
12.32 本章小结	271
12.33 本章小结	271
12.34 本章小结	271
12.35 本章小结	271
12.36 本章小结	271
12.37 本章小结	271
12.38 本章小结	271
12.39 本章小结	271
12.40 本章小结	271
12.41 本章小结	271
12.42 本章小结	271
12.43 本章小结	271
12.44 本章小结	271
12.45 本章小结	271
12.46 本章小结	271
12.47 本章小结	271
12.48 本章小结	271
12.49 本章小结	271
12.50 本章小结	271
12.51 本章小结	271
12.52 本章小结	271
12.53 本章小结	271
12.54 本章小结	271
12.55 本章小结	271
12.56 本章小结	271
12.57 本章小结	271
12.58 本章小结	271
12.59 本章小结	271
12.60 本章小结	271
12.61 本章小结	271
12.62 本章小结	271
12.63 本章小结	271
12.64 本章小结	271
12.65 本章小结	271
12.66 本章小结	271
12.67 本章小结	271
12.68 本章小结	271
12.69 本章小结	271
12.70 本章小结	271
12.71 本章小结	271
12.72 本章小结	271
12.73 本章小结	271
12.74 本章小结	271
12.75 本章小结	271
12.76 本章小结	271
12.77 本章小结	271
12.78 本章小结	271
12.79 本章小结	271
12.80 本章小结	271
12.81 本章小结	271
12.82 本章小结	271
12.83 本章小结	271
12.84 本章小结	271
12.85 本章小结	271
12.86 本章小结	271
12.87 本章小结	271
12.88 本章小结	271
12.89 本章小结	271
12.90 本章小结	271
12.91 本章小结	271
12.92 本章小结	271
12.93 本章小结	271
12.94 本章小结	271
12.95 本章小结	271
12.96 本章小结	271
12.97 本章小结	271
12.98 本章小结	271
12.99 本章小结	271
13.0 本章小结	271

第 1 章

ASP.NET AJAX 综述

近些年来,开发人员经历了很多开发方面的变化,从基于终端的编程,到基于 PC 的编程,到基于 Windows 的编程,再到 Web 编程。现在,我们已经到了另一个编程革命的边缘——就是为 Web 应用程序带来更多交互的用户界面。这个编程的革命为开发人员带来了一套称为 AJAX(Asynchronous JavaScript And XML, 异步 JavaScript 和 XML)的技术。在单击一个按钮来提交数据时,用户可以不再看到令人反感的屏幕闪烁,用户也不再被抛到了页面的顶部而丢失上下文关联。通过使用 AJAX,开发人员可以构建克服了传统 Web 回送所带来的缺陷的应用程序,改善了用户界面,并且允许开发人员创建用户友好的应用程序。

本章的主要内容:

- ASP.NET 开发以及如何与 AJAX 集成。
- AJAX 的工作原理,以及一些基本技术的高层次概述。
- AJAX 的优势。
- ASP.NET AJAX 是什么。
- 什么情况不适合使用 AJAX。

1.1 开发趋势

如果您已经有了一些开发经验,就像我们这些老家伙,您可能已经经历了几个开发阶段——从连接到伺服主机或小型机的终端设备到个人电脑,然后发展到客户端/服务器的开发。客户端/服务器开发可以允许后端资源、网络资源和前端 PC 的最小化,因为在前后端之间只需传递一些必需的数据。智能的客户端/服务器的开发允许构建响应迅速的应用程序,并且使得网络 and 后端资源的应用更有效。随着 20 世纪 90 年代后期 Web 开发的出现,我们好像又不幸地返回到了终端模式的开发阶段。在这种方法中,任何在客户端和服务器端之间的主要操作都要求发送所有的数据,这个过程被称为一个往返行程(round trip)。在往返行程过程中,表单上的所有数据会从客户端发送到 Web 服务器,Web 服务器会处理这些数据并发送回客户端。往返行程的结果是大量的数据会在客户端和服务器端之间来回传递,这会造成额外的数据传输和 CPU 占用,而且会使得 Web 应用程序和服务器不堪重负。

ASP.NET 开发

ASP.NET 是一套由 Microsoft 出品的 Web 开发技术，用于构建动态的 Web 站点、Web 应用程序以及基于 XML 的 Web 应用程序。ASP.NET 是 .NET Framework 的一部分，并允许开发人员使用多种语言构建应用程序，如 Visual Basic .NET、JScript .NET 和 C#。

1. 设计方法

ASP.NET 使得 Web 开发方法和 GUI 开发方法很类似，并允许开发人员像 GUI 那样通过控件组成网络页面。在 ASP.NET 功能中的服务器端控件和其他环境中的 GUI 控件是类似的。按钮、文字框、标签和数据网格等都具有一些可以修改的属性，并且可以提供可处理的事件。ASP.NET 服务器控件知道如何通过 HTML 页面来显示它们的内容，就像基于 GUI 的控件知道如何在它们的 GUI 环境中显示它们自己一样。ASP.NET 的一个额外的好处是 Web 服务器控件的属性和方法是类似的，在某种情况下与 Windows GUI/Windows 窗体环境中的对应控件是相同的。

2. ASP.NET 能够解决的问题

自从在 Windows 中发布了 IIS 后，Microsoft 已经出品了各种 Web 应用程序的开发方法。为什么开发人员需要 ASP.NET？ASP.NET 可以解决哪些以前的开发方法不能解决的问题？

Microsoft 的第一个流行的 Web 开发技术是 Internet Database Connector(IDC)，IDC 只提供了数据库的访问，它不提供任何通过编程方式对其他资源的访问。例如，不可能通过编程的方式发送电子邮件，或者实现其他非数据库的操作。另一个问题是，这种方式和大多数开发人员使用的传统开发语言(例如，Visual Basic 和 C++)有些不同。除此之外，这种开发体验在 Microsoft FrontPage 内并不是很引人注目。另外，IDC 还不支持调试。总之，IDC 只是一个用来改善应用环境的权宜之计。

Microsoft 出品的下一个 Web 开发方法是 Active Server Pages(ASP)。ASP 是一个脚本环境，它允许开发人员使用类似 Visual Basic 或类似 JavaScript 的方式开发。遗憾的是，这种环境同样具有几个问题，如下所示：

- 意大利面条代码——ASP 代码不提供结构化的开发环境，经常会导致缠绕在一起的“意大利面条代码”。确切地说 ASP 代码就是一个在每个页面顶部包含了一些配置信息的文件，每个页面会按照从顶到底的顺序执行。虽然可以使用组件对象模型(Component Object Model, COM)对象来避免意大利面条代码的出现，但这会带来要使用另一种开发工具的复杂性。
- 缺少代码分隔——ASP 代码会和显示逻辑代码混杂在一起，这就要求网页的开发人员和设计人员要能很好地配合工作，实际上这种情况并不常见。例如，大家都知道各种可视化的开发工具可以打开工作正常的 ASP 网页，在重新安排了一些代码后，ASP 代码很可能就会出错。

- 缺少代码的可重用性——在 ASP 环境中，代码可重用的能力很差。在传统的 ASP 中，代码的可重用性就是以 COM 对象形式提供逻辑的函数，这和 ASP 环境中的某些内容是对立的。
- 缺少调试支持——调试 ASP 应用程序基本上就是使用 `Response.Write`，这和集成开发环境(IDE)是非常不同的。
- COM 的问题——ASP 是基于 COM 模型的，因此也具有 COM 的一些问题，下面列出两个主要的问题：
 - 第一个问题是，对一个 COM 对象的更新往往是使用一个新的对象覆盖一个老的对象。如果方法调用方式改变了，或者引入了其他新的行为，这很可能会成为一个问题。
 - COM 的第二个主要的问题是，它是一个二进制标准，这个二进制标准是基于一个 32 位编程模型的。因此，COM 对象不可以在本地调整为可以运行在基于 Intel 的 64 位环境中。虽然在设计和构建 COM 的 20 世纪 90 年代中期，这不会是一个大的问题，但是在 21 世纪早期，引入了并不昂贵的 64 位系统后，这就会成为一个瓶颈。
- 解释型语言问题——ASP 是解释型语言。每次加载一个 ASP 文件，ASP 环境首先要分析这个 ASP 文件，然后编译代码，最后执行此文件。每次调用一个 ASP 文件时都会重复此过程，结果就是在服务器上会浪费很多处理时间。
- 状态机的存在——ASP 应用程序通常在每个 ASP 页面的顶部保存状态机，这是为了处理用户的状态，并随后显示代码(在软件代码中，一个状态机是代码的一个部分，其中的内容取决于用户的直接输入和在前面调用中的输入)。对于大多数客户端应用程序来说，它们是构建在事件基础上的，这是与状态机类似的概念，而这种方式和使用 ASP 开发是不同的。

在得到开发人员的反馈后，Microsoft 开发了 ASP.NET，它大大地简化了 Web 开发：

- 开发人员不用再费心处理状态。通过使用 ASP.NET，可以在一系列事件中执行不同的动作，并提供了和状态机类似的功能。
- 通过使用后置/旁置代码模型，代码可以和显示逻辑分别存储，这样设计工具和开发工具彼此冲突的机会就会大大减少。
- 可以使用一致的开发工具来构建应用程序和业务逻辑。通过使用一个集成的开发套件，开发人员可以更容易地和应用程序逻辑交互，这就可以实现更多代码的重用，并减少出错机会。
- 通过使用 Visual Studio 2005 IDE，ASP.NET 可以支持多种方法来调试和跟踪 ASP.NET 应用程序的运行情况。
- 由于 ASP.NET 是基于公共语言运行库(Common Language Runtime, CLR)和 .NET Framework 的，因此 ASP.NET 就不会有 COM 的版本问题。.NET Framework 允许在一个系统中具有多个版本的组件，而它们之间不会互相干扰。
- ASP.NET 是可编译的。在第一次加载文件时，ASP.NET 会编译和处理此文件。编译的文件会被保存在一个临时的目录里，随后的对此 ASP.NET 文件的调用就会直接处理编译好的文件。执行编译文件比解释环境的传统 ASP 文件要快得多。