



开发人员专业技术丛书

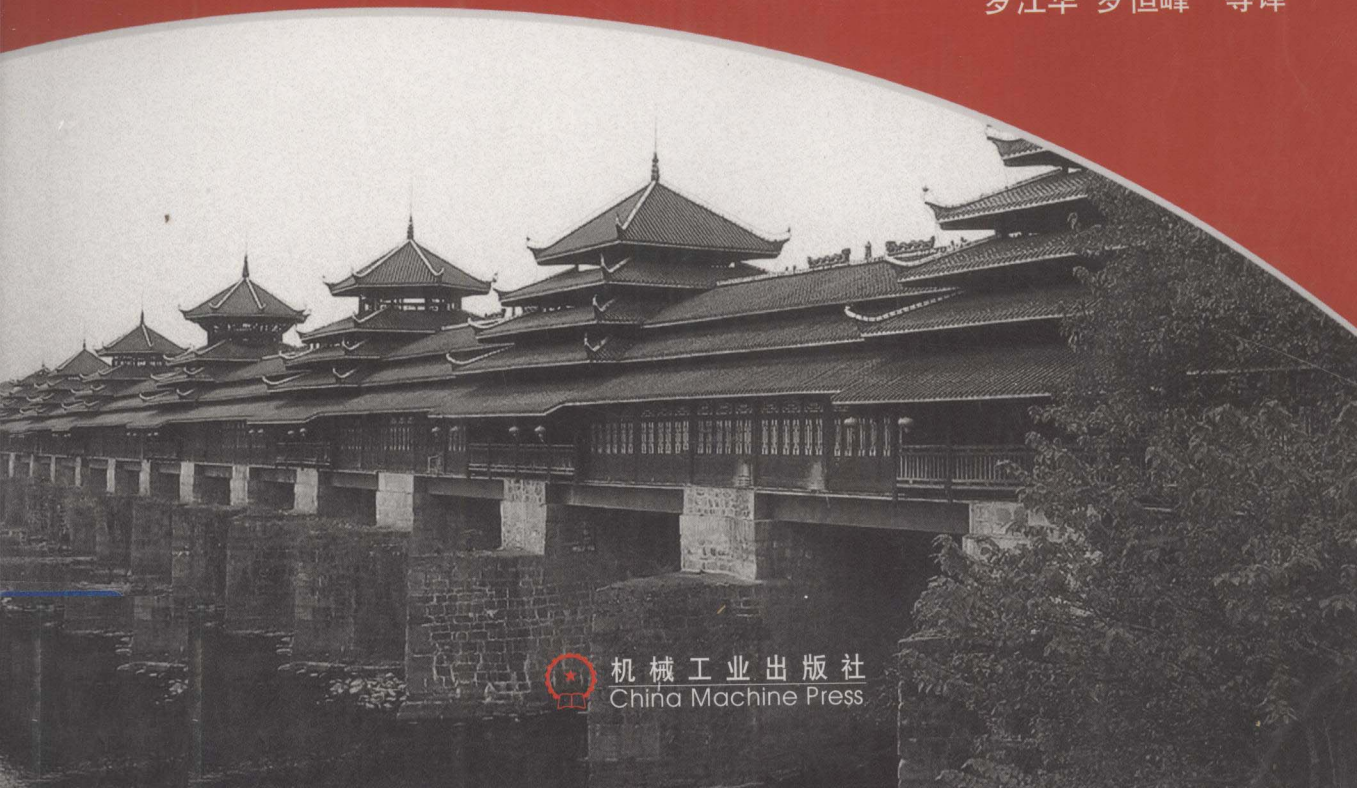
Advanced ASP.NET AJAX Server Controls
For .NET Framework 3.5

ASP.NET AJAX 服务器控件高级编程

(.NET 3.5版)

(美) Adam Calderon 著
Joel Rumerman

罗江华 罗恒峰 等译



机械工业出版社
China Machine Press

开发人员专业技术丛书

Advanced ASP.NET AJAX Server Controls

For .NET Framework 3.5

ASP.NET AJAX 服务器控件高级编程

(.NET 3.5版)

(美) Adam Calderon 著
Joel Rumerman

罗江华 罗恒峰 等译



本书介绍了如何使用 ASP.NET AJAX 来创建具有 AJAX 功能的服务器端控件, 通篇都结合 ASP.NET AJAX 的三个方: Microsoft AJAX 库、ASP.NET 2.0 AJAX 扩展以及 ASP.NET AJAX 控件工具箱来进行探讨。本书被分成了四个主要部分。第一部分主要聚焦 Microsoft AJAX 库和 JavaScript 语言。第二部分主要讲述创建分布式的启用 AJAX 的控件。第三部分介绍通信, 阐述了客户端控件与服务器端交流时的不同之处。最后, 在第四部分, 重点介绍了 ASP.NET AJAX 控件工具箱的相关知识。

本书内容详实, 结构清晰, 图文并茂, 可作为相关技术人员的参考书籍。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Advanced ASP.NET AJAX Server Controls: For .NET Framework 3.5* (ISBN 978-0-321-51444-8) by Adam Calderon, Joel Rurmerman, Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison Wesley.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2009-1334

图书在版编目 (CIP) 数据

ASP.NET AJAX 服务器控件高级编程 (.NET 3.5 版)/(美) 卡尔德诺 (Calderon, A.) 等著; 罗江华等译. —北京: 机械工业出版社, 2009.9

(开发人员专业技术丛书)

书名原文: *Advanced ASP.NET AJAX Server Controls: For .NET Framework 3.5*

ISBN 978-7-111-27096-6

I. A… II. ①卡… ②罗… III. ①主页制作 - 程序设计 ②计算机网络 - 程序设计
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2009) 第 074314 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 陈佳媛

北京京北印刷有限公司印刷

2009 年 9 月第 1 版第 1 次印刷

186mm × 240mm · 26.25 印张

标准书号: ISBN 978-7-111-27096-6

定价: 65.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010) 68326294

译者序

AJAX 无疑是现在最为热门的一项技术。但是, Ajax 本身并不是一项新技术, AJAX 是由 JavaScript、XML、XSLT、CSS、DOM 和 XMLHttpRequest 等多种技术组成的。微软顺应潮流最先推出了 Atlas, 它是 ASP.NET AJAX 的前身, 微软最终将其命名为 ASP.NET AJAX。

还记得上网提交注册信息给网站吗? 用户不得不将大量的时间花费在等待上。即使客户端和服务器的连接状况非常好, 网络延迟可以忽略不计, 在加载新页面时浏览器中仍然会有一个显示空白页面的过程, 也就是一次“闪烁”。这是因为浏览器需要清除原有的内容并重新写入新的内容。Ajax 应用程序则是借助 XMLHttpRequest 对象的帮助, 只从服务器端取得必要的表示这一条评论的数据, 而不是新页面的所有内容, 避免了在网络中传输不变内容所造成的浪费, 也就自然减少了用户等待的时间。

AJAX 应用程序通常很重视用户体验, 它们一般都会关注目前用户的操作, 并及时地给出下一步的提示。即使是一个很小的动作, 客户端都可以将其回送至服务器, 然后服务器即可充分利用它掌握的大量信息以及强大的计算能力迅速向用户给出最恰当的提示。例如, 某个系统中用户的所在地信息可以由省份和城市两部分组成。在 Web 页面上通常使用两个下拉列表让用户选择。当用户选择了省份后, Ajax 应用程序就可以通过 XMLHttpRequest 对象取得该省内所有城市的列表, 并填充到城市列表中供用户进行下一步选择。

AJAX 将许多不同的技术结合在了一起, 而它们之间的协调者就是运行于浏览器中的 JavaScript。在 AJAX 中, JavaScript 的功能已经不只是在页面上显示漂浮广告或修改浏览器状态栏。通过充分挖掘 JavaScript 的潜力, 我们能够把许多原本只有桌面应用程序才能够实现的丰富功能(例如, 鼠标拖动、右键菜单、自动完成、拼写检查、键盘快捷键等)搬到 Web 页面中, 这些都不是问题! Windows Live Mail 就是典范之作, 它提供的完善功能直逼 Office Outlook, 甚至让用户无法感受到这是一个基于 Web 的应用程序, 本书的第一部分: 客户端代码部分会向你阐述 JavaScript 的方方面面。

AJAX 之所以能够实现如此强大的功能, 就是因为开发者广泛使用了现有的各种成熟技术, 充分挖掘了它们的潜力, 并把它们有机地组合到一起。

ASP.NET AJAX 可以相应地分为服务器端和客户端两个部分, 前者包含 ASP.NET 2.0 AJAX Extensions 以及 ASP.NET AJAX 控件工具箱, 后者包含微软 AJAX 库。ASP.NET AJAX 控件工具箱是由开发爱好者支持的, 微软并不提供对它的支持, 在本书的第四部分会对其进行详细阐述。

本书的读者对象十分广泛对读者具有指导和参考意义。本书适用于 ASP.NET 2.0 或 3.5

IV

的一般开发人员，以及新技术追随者，但不适合初级读者。

本书在翻译过程中得到了很多朋友的支持和帮助，再次特别感谢你们。她们是：振雅丹、施丽鹃。

由于本书的翻译进度要求高，涉及技术面广，翻译难度较大，加之译者水平有限，书中难免有疏漏或错误，欢迎广大读者批评指正。

罗江华

2009年5月于成都

对本书的赞誉

“如果你是一位 ASP.NET 开发人员，并且正在寻找使用 AJAX 技术来开发基于 Web 2.0 应用程序的捷径，那么本书正是你需要的。使用你了解到的工具和方法来构建一个具有超强交互的 AJAX 功能正是当代 Web 应用程序所需要的。”

——Shawn Burke

微软 AJAX 控件工具箱主管

“你知道什么令人沮丧？”当我阅读或听到一种技术，并开始使用它，然后因为文档提供的示例概念过于简单或者一本仅对文档进行简单介绍的参考书籍时，就是这样的心情。Adam Calderon 和 Joel Rurmerman 将使你构建 ASP.NET AJAX 服务器控件的技能更上一层楼。它们不仅为你展示应该如何使用这项技术，而且还深入地探讨了其内部机制。本书适合那些想要通过自定义方式来封装一些比 UpdatePanel 控件更复杂的控件的读者。如果你正在寻找这样一本资料，那么这本书是最好的了。”

——Thomas Lewis

微软 Web 用户体验宣传经理

“本书是讲述 AJAX 框架方面最完整、最权威的一本指南，作者对该项技术的灵活性、扩展性进行了深入浅出的剖析，本书适合初级、高级开发人员。”

——Jason Schmitt

Steelbox Networks 公司产品副总裁

众望所归 “Adam Calderon 和 Joel Rurmerman 推出了这本以构建 AJAX 服务器控件为主题的书籍。本书是专门针对 ASP.NET AJAX 开发人员的“红药丸”（参看电影《黑客帝国》）。自定义控件和组件是所有 .NET 开发人员进行代码重用的最好机会，本书将控件开发引入了 AJAX 时代。”

——Joe Stagner

微软开发工具和平台高级程序经理

序 言

如今 ASP.NET 平台上已经架构着数以百万计的网站，这或许是 Web 开发方面最有生产力的平台之一。在过去将近 10 年的开发和使用过程中，ASP.NET 已经在它的周围形成了一个强大的社区，以及一个由开发者构成的并且蓬勃发展的生态系统。

页面框架和其相关的服务器控件是 ASP.NET 的成功、开发者经验、编程模式，以及可扩展性的精髓所在。写这个序言，把我带回到了早期的 ASP.NET 时代，也更提醒了我 Web 平台的框架在不断演变。

在 20 世纪 90 年代后期，Web 还是一个非常初期的应用程序平台。浏览器带来了新级别通信的潜力，但是几乎没有提供兼容性的能力（记得 HTML 3.2 吗？），而且诸如“无状态编程模型”的概念也成为了一种奇怪的范例。服务器控件提供了一系列类似的抽象，然后为 Web（Visual Basic 风靡一时）创建了一个基于快速应用程序开发（RAD）编程体验的组件，并允许开发者像它们最初接触 Web 那样自然地构建下一代数据驱动应用程序。

在 2006 年，AJAX 的追随者们重新创造了一种作为应用程序平台的 Web 应用。今天，AJAX 成为了主流，而且毫不夸张地说，它无处不在。它允许构建有交互性的用户体验，这也是用户所期待的。当然，它带来了一些新而又彼此相同的挑战：变化的浏览器 API 函数和不熟悉的基于脚本的编程模型。ASP.NET（特别是服务器控件）又一次提供了创造一种具有生产力的开发模型，为基于 AJAX 的技术合作，也将服务器和客户端行为封装为一种熟悉而统一的组件模型。

ASP.NET 创造了一个完美的 AJAX 故事。传统的服务器控件创建一个简单的以服务器为中心的 AJAX 编程模型，但是它们仅是故事的一部分。新一代的服务器控件更是支撑起了一种 AJAX 脚本框架，可以独立地构建一种客户端为中心的 AJAX 编程模型。核心的框架是由 AJAX 工具箱来组成的，它提供了一组引人注目的“即插即用”的组件和一个为了更好地开发 ASP.NET AJAX 栈的开源项目。我很兴奋地看到这个完整的故事在本书中得到了展示。

在本书中，Adam 和 Joel 致力于深入 ASP.NET AJAX 框架的内部工作和扩展性，从而提供超越基础的知识。他们通过介绍由脚本框架、架构以及创建支持 AJAX 服务器控件的技术所创建的编程模式来做到这一点。它们也讲到了一些诸如本地化和错误处理这样很有关联的主题。通过提供一个概念性的指导来理解和展开这个框架，这本书肯定可以为任何正在寻找解决 ASP.NET AJAX 真实潜力的应用程序和组件开发者提供帮助。

Nikhil Kothari

微软 .NET 开发者平台软件架构师

前言

介绍

服务器控件是我们所构建的 ASP.NET 应用程序不可或缺的一方面。它们将浏览器外观和服务器功能封装于一个可重用的对象之中。它们可以在一个单独的 ASP.NET 应用程序中跨多个页面使用,同时还可以跨多个 ASP.NET 应用程序使用。ASP.NET 有许多预构建的服务器控件。我们有一些简单的控件(例如,标签),还有一些类似 GridView 这样的复杂控件。我们可以创建自己的服务器控件来实现没有一个现有控件可以满足的需求,只需要通过继承合适的基类然后重载所需的方法,即可实现。

这种使用服务器控件的方式封装了浏览器的外观和服务器功能的模式,从起初的 ASP.NET 1.0 开始就很好地为我们的需求服务,但是我们对服务器控件的需求一直在改变。

最近刚出现的新服务器控件需求,就是与 AJAX 功能的结合能够直接放在服务器控件的能力。

这种需求出现的原因是,我们的 Web 应用程序需要更立即的响应以及比传统 ASP.NET 控件提供的重画整个屏幕的那种模式更直观的交互性。这种需求已经浮现出来,因为用户正在使用类似 Gmail、Live.com、Yahoo! Mail 以及其他一些不需要每次点击一个按钮或需要接收新数据时重画屏幕这样的站点。所以,它们选择了 AJAX 来获取新的数据,然后基于所取得的新数据来更新或添加屏幕的一部分。因为这些站点的大量应用,用户在使用这些站点的过程中已经习惯了这种体验,他们期望其他站点也能够像他们所喜好的这些站点一样,表现得同样优雅。

因为我们的用户正在要求一种本质上使用了 AJAX 的站点体验,我们使用服务器控件来构建我们的 ASP.NET 站点,我们需要一个简单的方式来创建服务器控件,不仅需要封装浏览器外观和服务器功能,而且还要包含 AJAX 的功能,这样服务器控件才启用了 AJAX 功能。

退一步看,不像其他一些你可能已经读到过的技术,ASP.NET AJAX 服务器控件没有提供任何不能实现的东西。我们总是能够将 AJAX 功能嵌入服务器控件中……这还真苦恼。

我们曾有好几种方法在服务器控件中包含 JavaScript,比如将其作为一个资源嵌入,但是我们最终其实还是要做相同的三个任务。为了让我们的服务器控件具有客户端的兼容性,我们总是需要连接几个字符串来构成一个 JavaScript 语句和功能,写探测浏览器的语句来确保 JavaScript 有跨浏览器的兼容性,然后添加属性或呈现出附着 JavaScript 功能到服务器控件的客户端版本的 HTML。这并非不可能,但是这通常有某种错误倾向,而且总是有这种服务器

代码和 JavaScript 的混合，而变得很难维护，甚至是更难以阅读。

另外，如果你有几个拥有客户端能力的服务器控件，要确保每个服务器控件要求的客户端功能在呈现到浏览器时没有相互覆盖，这是很困难（并非不可能）的事情。跟踪到那样的问题也总是要大约有趣的一个小时吧。

如果我们想要包含一个用户能在服务器控件中点击按钮时与服务器异步通信的机制，那么困难度就会呈指数上升。就算是有一个通信库的帮助器工具，让控件正确地和服务端通信还是需要一些诀窍。

这些妨碍很有问题，会导致一些不好的编程习惯和代码，会吓跑一些程序员，让它们甚至不敢尝试在它们的服务器控件中包含进 AJAX 功能。

这些问题正是微软 ASP.NET AJAX 所解决掉的问题。

在本书中，我们教你如何使用 ASP.NET AJAX 来创建封装了 AJAX 功能的服务器控件。ASP.NET AJAX 提供了服务器和客户端编程的构建模型，用来轻易地添加 AJAX 能力到服务器控件。但是 ASP.NET AJAX 减少了给服务器控件添加 AJAX 能力的复杂性，我们可以创建那些有 AJAX 能力的服务器控件，只会受到自己的创造力的限制。如果我们想要一个能够自己根据新数据更新的列表框，如果我们想要一个能动态从服务器生成预先键入（type-ahead）的文本框，或者我们想要一个按钮来提交地址去验证，我们都可以轻易地通过 ASP.NET AJAX 来完成这些事情。

ASP.NET AJAX 组件

这本书主要探讨 ASP.NET AJAX 的三个方面：微软 AJAX 库、ASP.NET 2.0 AJAX 扩展以及 ASP.NET AJAX 控件工具箱。这里对这些不同组件给出一个快速概览。

微软 AJAX 库

微软 AJAX 库是 ASP.NET AJAX 的 JavaScript 编程框架。它提供了全部的客户端编程结构，你可以使用其创建新的客户端对象和组件。它包含于 MicrosoftAjax.js JavaScript 文件中，而后者已经嵌入到了 System.Web.Extensions.DLL 中。

ASP.NET 2.0 AJAX 扩展

ASP.NET 2.0 AJAX Extension 是服务器对象，例如 ScriptManager、ScriptControl 和 ScriptDescription，它提供了一个在微软 AJAX 库和 ASP.NET 开发之间的连接。这些服务器对象展示了 ASP.NET AJAX 和其他 AJAX 框架的一个重要的区别，因为它们为处理客户端代码（同时也允许我们创建基于 AJAX 的服务器控件！）提供了服务器编程模型。与微软 AJAX 库一样，它们也包含在 System.Web.Extensions.DLL 中。

ASP.NET AJAX 控件工具箱

ASP.NET AJAX 控件工具箱是一个构建于 ASP.NET AJAX 之上的一个共享代码项目。这是 Microsoft 和 ASP.NET AJAX 社区共享的成果，目标在于开发强大和可重用的 ASP.NET

AJAX 扩展和控件。

它其实并不是 ASP.NET AJAX 的一部分，但是它提供了不错的服务器控件和扩展控件，它对 ASP.NET AJAX 社区而言是非常宝贵的。通过它来创建新的扩展控件也是我们会全面介绍的一个主题。

本书解析

这本书分成了四个主要部分。在第一部分，我们聚焦于 Microsoft AJAX 库及其底层的 JavaScript 语言的基础。我们称这部分为客户端代码。在第二部分，我们聚焦于创建分布式的基于 AJAX 的控件，我们称这部分为控件。第三部分为通信，我们聚焦于客户端控件与服务器通信的不同方式。最后，在第四部分，我们聚焦于 AJAX 控件工具箱，一个稍微高级的创建基于 AJAX 的服务器控件的模型。最后一部分为“AJAX 控件工具箱”。

客户端代码

第 1 章，JavaScript 编程，主要是关注 JavaScript 这个强化微软 AJAX 库的编程语言。我们在 JavaScript 上面花了一整章，是因为非常多的开发人员（也包括我们自己）在和这个语言工作的时候隐藏了太多的关键细节；而且因为你正准备为基于 AJAX 的服务器控件编写如此多的 JavaScript 代码，所以扎实的基本功是很重要的。

第 2 章，微软 AJAX 库编程，我们从第 1 章停下来的地方继续。我们解析微软 AJAX 库是如何构建在 JavaScript 上而提供一个 .NET 开发人员所熟悉的一个编程平台。

控件

从第 3 章开始，组件，我们通过学习如何使用和从三个关键客户端类型（组件、控件以及 behaviors）来开始创建完全封装好的基于 AJAX 的控件。我们会讲述理论，并且也提供几个练习例子。

在第 4 章，Sys. Application，我们会讨论 Sys. Application 以及它如何可以表现得像我们能够与之交互的客户端运行时那样，这可能是微软 AJAX 库里面最重要的部分。

第 5 章，为服务器控件添加客户端能力，当我们学习到如何创建那种可以自动创建相应组件的服务器组件时，我们就会把服务器混合起来。

在第 6 章，ASP.NET AJAX 本地化，我们继续为控件添加在 ASP.NET AJAX 中本地化的深度检测能力。

最后，在第 7 章，局部回传环境中的控件开发，我们基于 UpdatePanel 会如何影响控件开发的考虑，而把控件的封装进行隔离。

通信

在第 8 章，ASP.NET AJAX 通信架构，我们开始看看在 ASP.NET AJAX 中使用 WCF（Windows Communication Foundation）服务、页面方法和客户端 Web Service 代理的通信。

在第 9 章，应用程序服务，我们将覆盖到应用程序服务，并且包含一个教你如何构建你

自己的应用程序服务的演示。

AJAX 控件工具箱

从第 10 章开始, ASP.NET AJAX 控件工具箱架构, 我们来看看 AJAX 控件工具箱。我们会探讨被工具箱控件使用的基础类, 以及提供额外特性的支持类 and 设计类。

最后, 我们在第 11 章总结全书, 使用 ASP.NET AJAX 控件工具箱来给服务器控件添加客户端能力, 这里我们将使用 AJAX 控件工具箱来给服务器控件添加客户端能力。这章包括如何构建一个新的扩展控件并为其提供设计时特性。

有什么还没讲

你也许会感觉看到这样一段文字来谈论我们没有讲到的事情会有点奇怪。但是我们是有点原因的。

第一, 本书相对于整个详尽的 ASP.NET AJAX 而言, 只是讨论了一个相当狭窄的主题。正因为如此, 我们介绍了 ASP.NET AJAX 的基础或历史, 而不是一般的入门教程。而我们也做了一个假设, 你已经或多或少地掌握了一些 ASP.NET AJAX 的知识。如果你没有, 也不要担心; 在你感觉容易而不需要花太长时间的地方多学习 ASP.NET AJAX 的知识, 会使后面学习本书其他部分也很容易。对于这类信息, 微软 ASP.NET AJAX 网站 <http://asp.net/ajax> 上面有非常精彩的资源。

第二, 省略了一个熟悉的 ASP.NET AJAX 主题, 我们想要找一个机会, 也想在来得及的时候告诉你而捍卫我们的决定。其实这也是我们不断反复争论的一些事情, 而我们也询问过很多同事并征求他们的意见, 这个决定对我们来说真是相当不容易。

没有介绍如何使用 UpdatePanel 服务器控件。

OK, 你还没有合上本书吧? 好的, 让我们来解释一下我们是如何以及为什么要做这个决定的。

简单而言, UpdatePanel 是一个服务器控件。它在 ASP.NET AJAX 里面提供一个快速但是并不干净的方式来刷新一个页面的局部, 这样页面就可以走过它自己一般的生命周期却不用在页面处理完成时刷新整个页面。如果使用它, 我们可以不必警惕我们自 ASP.NET 1.0 时代就开始的网页编程方式。这很好, 而且对 Microsoft 而言也是个快速的胜利。它允许 ASP.NET AJAX 能够被 ASP.NET 的开发人员快速使用, 同时也提供了一种独特的优势来对抗其他的 AJAX 框架。

然而, UpdatePanel 也仅仅是一个服务器控件, 它的开发方式有所不同, 它和我们所涉及的 ASP.NET AJAX 服务器控件类型开发没有太多可比较的特性。

我们并不是说它不是一个重要的服务器控件, 而其实它在 AJAX 世界里是有地位的。或者说, 它是一个极具价值的工具, 其复杂性和合理的用途都值得写一本小书, 然而不是这本。

最后, 虽然我们没有讲如何使用 UpdatePanel, 我们却谈及了如何创建服务器控件让其能够在 UpdatePanel 或更特定的局部回传环境中正常工作。我们期待你想让新服务器控件能够工作于任何 ASP.NET 环境, 即使是一个局部回传环境也不例外。虽然如此, 局部回传环境需

要我们使用一些不同的方法，新的 `ScriptManager.RegisterXXX` 方法开始变得很普遍，还需要注意一下我们是如何创建服务器控件的。所以，第 7 章将主要讲述这个主题。

什么是服务器控件

围绕服务器控件来写这样一本书，就让我们可以更深入地挖掘对于 Web 应用程序开发者十分重要的一个主题。包含 ASP.NET AJAX 技术方方面面的书籍，现在都有了。因为它们在于覆盖很广泛范围的主题，每件事都要尝试，但是在关于 ASP.NET AJAX 的某具体部分如何工作时就会遇到麻烦。它们尝试给予主题的那种泛泛而谈却恰是我们认为的创建服务器控件的关键。许多开发人员会快速地翻过那些很概略的书本，这是我们的经验，因为工作于一个真正的 Web 应用程序生命周期里，非基本状况就会迅速出现。

目标读者

这本书的主要目标人群还是那些有过自定义 Web 服务器控件开发经验的 ASP.NET 开发者。我们期待你能通过阅读本书来获得新的 ASP.NET AJAX 技能，进而强化你已有而精通的 ASP.NET 开发技能。你开发的应用程序要求优雅、专业、易于维护以及可扩展性，所以你为了获得优势，趋于在任何可能的时候使用服务器控件。

除了你在 ASP.NET 上的经验，我们还希望你已熟悉 JavaScript 和 ASP.NET AJAX 基础。因此，我们就不再覆盖如何建立一个新的支持 ASP.NET AJAX Web 应用程序方面的东西。尽管我们会覆盖一些 JavaScript 知识，但是我们的起始级别是假定一些已有知识的存在。

我们的目标是给你提供你所需要的工具，来构建可重用的 ASP.NET AJAX 服务器控件，或者 AJAX 控件工具箱扩展控件。我们的感觉是，有一定经验的 ASP.NET 开发者是有理由能够通过本书学会这些技能来创建新的 ASP.NET AJAX 服务器控件，然后再添加这个控件到它们的 ASP.NET 开发工具箱中。

必要条件

本书需要 ASP.NET 3.5 AJAX 和 Visual Studio 2008。我们大量覆盖了包含在 ASP.NET 3.5 AJAX 而不包含在 ASP.NET 2.0 AJAX 的特性，以及诸如自动属性和 JavaScript IntelliSense 这样在 C# 和 Visual Studio 2008 中的新功能。

源代码

本书的源代码例子可以在网站上找到：www.informit.com/title/9780321514448。

致 谢

我们非常惊讶有如此多的人员为这本书做了如此多的努力。从编辑到技术审阅者、文字编辑以及营销人员，有很多负责的人员来创建这本高质量的书，而不单是作者们。

我们首先想感谢我们所有的技术审阅者：Joe Stagner、Jason Schmitt、Milan Negovan 和 Russel Gauthier。没有你们在每章投入的大量的时间和精力，本书将无法达到这样完美。最初的版本很粗糙，真的非常粗糙。如果你们不是如此的真诚，我们真会认为我们所写的已经准备好出版了，那可真是是一个弥天大错。你们的才能让我们整理了凌乱的思绪，看到了我们真正想要说的东西，将我们领上了一条撰写容易理解而有用之书的道路，这可真是令人惊叹。缺少你们所提供的任何一丝帮助，我们都不可能完成。

我们还要感谢我们的项目编辑：Jovana San Nicolas-Shirley，为我们做了最后的修改，并回答了我们那些看似无穷无尽的问题（以及我们有些还要问的问题）。

还有，我们要感谢我们的文字编辑，Keith Cline，在我们写得不清楚的时候为我们阐述清晰、纠正我们错误的语法，让我们看起来像经验丰富的作者。（那时还完全不是!）

特别感谢 Curt Johnson、Nancy Valentine 以及 Andrea Bledsoe 所在的营销团队为我们出版本书的努力。

我们还想要感谢出色的编辑，Joan Murray，不仅指导了两个新作者完成了他们第一次写书经验，而且这个过程中还照管了一个小宝贝。再次祝贺，并感谢提供的所有支持以及我们写一本好书所需要的努力。

感谢在 Addison-Wesley 工作于这本书的每一个人，包括我们的临时编辑 Emily Frey；扮演早期开发编辑的 Kristy Hart 和 Sheri Cain；让我们进入美国国会图书馆的 Karen Opal；以及其他我们所遗忘和不知道的每个人！

来自 Adam

首先，我想感谢我的合著者，Joel，感谢它躬亲职业的对卓越无止尽的追求。我们关于技术和人生无尽的讨论，让撰写本书成为了一段真正严肃的经历。我也想感谢在 InterKnowlogy 出色的团队，提供了令人一直保持对技术追求热情的环境。InterKnowlogy 所持续尝试的重要工作，让在这个环境中工作起来非常有挑战且令人激动。

来自 Joel

虽然本书是献给我的妻子 Stacey 的，我还是想要在这里感谢她。我们在家里共享一个办

公室，她在为执照考试学习而我在写作，我们一起度过了大量的时光。如果你没有帮我打破千篇一律的生活，没有在我失落的时候牵引我出来，那么我将没有办法一直保持思维清醒的状态。你活泼的思维和逻辑是写这本书最有意思的方面之一。你真是一个令人惊叹的女人、一个贤惠的妻子、一个多才多艺的专家、一个梦幻般的办公室友。

虽然我的妻子在过去一年的每个晚上每个周末都是我的办公室友，但是没有我的合著者 Adam，这本书就不会出现，我也不会在我的职业生涯中进步。当我们在 InterKnowlogy 工作的时候，Adam 跟我提起要一起合著这本书，因为我所显示出来对于这项技术的兴趣已经远远超越了工作本身的需求。因为我真的是很投入这项技术，但更多是因为我很尊敬它，无论是从专业还是个人方面，我迅速地加入了。尽管在我们开始写书不久我就离开了 InterKnowlogy，我们的个人和专业联系却更紧密了；而且没有他的带领、奉献、技术知识和行业联系，这本书永远不可能像现在这般美好。谢谢你，Adam，感谢你接起我无数的电话、麻烦的问题，以及所有的煎熬。

我的双亲：爸爸和妈妈。谁会想到我其实是爸爸那里得到百分之五十，妈妈那里得到百分之五十？爸爸，你的“不要猜，要想”的建议让我在高中学 Pascal 编程时解决了一个问题。妈妈，你在我的报告、论文和卷子上奉献了无数的时间，造就了一个半路改行的像样作家，也没有被文字编辑上的红笔吓着。

Keri、Seth、Riley 和 Cameron。谢谢你们总是要求这本书，一直鼓励着我。记住，对小孩而言这是一个很好的睡前故事，而且我确信我也会让你们睡着的。

所有在 CoStar Group 的开发者们，特别是我的队友 Jason、Louise 和 John。你们帮我成为了一名开发者和一个团队成员，我们一起交付了首屈一指的映射解决方案。我确信接下来的一年会如同过去的一年那样有价值，因为我们又进入了新的未知领域。

最后，感谢在过去 14 个月里和我共享啤酒，并听我为完成里程碑欢呼或者抱怨即将来临的期限的所有朋友，谢谢你们。无论你们相信与否，你们从始至终的支持对我意义重大。

目 录

译者序	
对本书的赞誉	
序言	
前言	
致谢	

第一部分 客户端代码

第1章 JavaScript 编程	2
1.1 常规的 JavaScript	2
1.1.1 JavaScript 介绍	2
1.1.2 语言特性	3
1.1.3 简单数据类型	4
1.2 string	5
1.2.1 对象	6
1.2.2 变量和函数参数	11
1.2.3 错误处理	17
1.2.4 使用超时和间隔延迟 代码执行	21
1.3 面向对象 JavaScript 编程	27
1.3.1 抽象数据类型	27
1.3.2 继承	32
1.4 小结	35
第2章 微软 AJAX 库编程	36
2.1 扩展内置的 JavaScript 类型	36
2.1.1 布尔类型	36
2.1.2 日期类型和数据类型	37
2.1.3 字符串类型	37
2.1.4 数组类型	38
2.2 扩展微软 AJAX 库	41

2.2.1 类	41
2.2.2 接口	46
2.2.3 枚举	50
2.2.4 继承和接口实现	54
2.3 重要的新类型	61
2.3.1 Sys. EventHandlerList	61
2.3.2 Sys. StringBuilder	65
2.3.3 Sys. Debug	66
2.3.4 Sys. UI. DomElement	70
2.3.5 Sys. UI. DomEvent	74
2.4 维护范围	77
2.4.1 委托	78
2.4.2 回调	79
2.5 小结	80

第二部分 控 件

第3章 组件	82
3.1 组件定义	82
3.2 Sys. Component	84
3.2.1 定义新组件	86
3.2.2 创建组件	90
3.2.3 封装组件	102
3.3 控件	102
3.3.1 新概念	103
3.3.2 定义一个新控件	104
3.3.3 创建一个控件	106
3.3.4 封装控件	107
3.4 behavior	107
3.4.1 定义一个 behavior	108
3.4.2 创建一个 behavior	109
3.4.3 封装 behavior	113

3.5 小结	113
第4章 Sys. Application	114
4.1 背景资料	114
4.1.1 创建 Sys. Application	114
4.1.2 类型信息	116
4.1.3 方法信息	116
4.2 组件管理	117
4.2.1 添加组件	118
4.2.2 查找组件	120
4.2.3 删除组件	123
4.2.4 返回组件	124
4.3 初始化流程	124
4.3.1 组件创建过程	127
4.3.2 Load 事件	134
4.4 unload 流程	138
4.5 小结	140
第5章 为服务器控件添加 客户端能力	141
5.1 脚本生成体系结构	141
5.1.1 behavior 和控件的脚本生成	142
5.1.2 Script 资源	146
5.1.3 ScriptManager	147
5.2 使用 ExtenderControl 添加客户端 behavior	149
5.2.1 ExtenderControl 概述	149
5.2.2 创建一个 Extender 控件	150
5.3 使用 ScriptControl 添加 客户端功能	158
5.3.1 ScriptControl 概述	159
5.3.2 创建 ScriptControl	161
5.4 使用 IScriptControl 接口添加客户端 功能到复合控件中	168
5.4.1 复合控件概述	168
5.4.2 IScriptControl 接口	170
5.4.3 创建复合控件	170
5.5 小结	174
第6章 ASP.NET AJAX 本地化	175
6.1 ASP.NET 中的本地化	175
6.1.1 决定需要本地化的内容	176

6.1.2 在特定文化背景下运行程序	180
6.1.3 本地化所显示的值	183
6.2 ASP.NET AJAX 本地化	190
6.2.1 JavaScript 本地化功能	191
6.2.2 ASP.NET AJAX 本地化 功能	193
6.3 小结	217
第7章 在局部回传环境中进行 控件开发	219
7.1 UpdatePanel 行为	219
7.2 局部回传在客户端组件的效果	224
7.2.1 behavior 和控件的自动释放	227
7.2.2 组件的自动释放	234
7.2.3 组件、控件或 behavior 的 手动释放	236
7.3 JavaScript 语句和文件的加载	247
7.3.1 ScriptManager Registration 方法	247
7.3.2 Sys. Application. notify- ScriptLoaded ()	252
7.4 Sys. Application 事件	253
7.4.1 Init 事件	253
7.4.2 Load 事件	254
7.5 小结	256

第三部分 通 信

第8章 ASP.NET AJAX 通信架构	258
8.1 新的信道通信范例	258
8.2 ASP.NET AJAX 2.0 扩展 通信架构	260
8.2.1 Web 服务	260
8.2.2 网页方法	267
8.2.3 串行化	268
8.2.4 服务器框架组件	273
8.3 微软 AJAX 库通信架构	277
8.3.1 服务代理	278
8.3.2 串行化	287
8.3.3 Web 请求	288
8.3.4 Web 请求核心	292

8.4 小结	293
第9章 应用程序服务	294
9.1 ASP.NET 2.0 的 Membership、Role 和 User Profile 服务	294
9.1.1 Forms 认证	294
9.1.2 ASP.NET 2.0 Provider 模型	296
9.1.3 网站管理工具	298
9.1.4 成员资格	299
9.1.5 角色	301
9.1.6 配置	307
9.2 ASP.NET AJAX 应用服务	309
9.2.1 认证服务	309
9.2.2 角色服务	313
9.2.3 配置服务	314
9.3 自定义应用服务	317
9.3.1 HTTP Handler Factory 和 支持类	319
9.3.2 服务代理	327
9.3.3 配置	330
9.4 小结	331

第四部分 AJAX 控件工具箱

第10章 ASP.NET AJAX 控件

工具箱架构	334
10.1 工具箱概要	334
10.1.1 基于属性简化开发	334
10.1.2 丰富的 .NET 类	334
10.1.3 丰富的 JavaScript 类	335
10.1.4 支持动态化	335
10.2 工具箱的构成	335
10.2.1 安装	335
10.2.2 解决方案的结构	336
10.3 基于服务端架构	337
10.3.1 属性	337
10.3.2 扩展和脚本控件的基类	340
10.3.3 设计类	343
10.4 基于客户端的架构	346
10.4.1 BehaviorBase 类	346

10.4.2 ControlBase 类	347
10.5 Animations	347
10.5.1 animation 结构和类型	348
10.5.2 客户端架构	348
10.6 小结	355

第11章 使用 ASP.NET AJAX 工具箱给服务器控件添加客户端功能

11.1 使用 ExtenderControlBase 添加 客户端行为	356
11.1.1 Visual Studio 2008 扩展控件 模板库	356
11.1.2 继承自 ExtenderControl- Base 类	359
11.1.3 创建 AjaxControlToolkit. BehaviorBase 类	361
11.1.4 向控件添加 extender	363
11.1.5 最后的思考	363
11.2 添加设计时支持到你的 Extender 控件上	364
11.2.1 默认的设计时体验	364
11.2.2 给属性添加设计器和 编辑器	365
11.3 给 extender 控件添加 animation	370
11.3.1 使用 JavaScript API 的 animation	370
11.3.2 使用声明式方法的 animation	373
11.4 小结	377

第五部分 附 录

附录 A Visual Studio 2008 中的 JavaScript	380
附录 B 验证方法参数	386
附录 C ASP.NET handlers 和 modules	389
附录 D 客户端错误处理代码	396