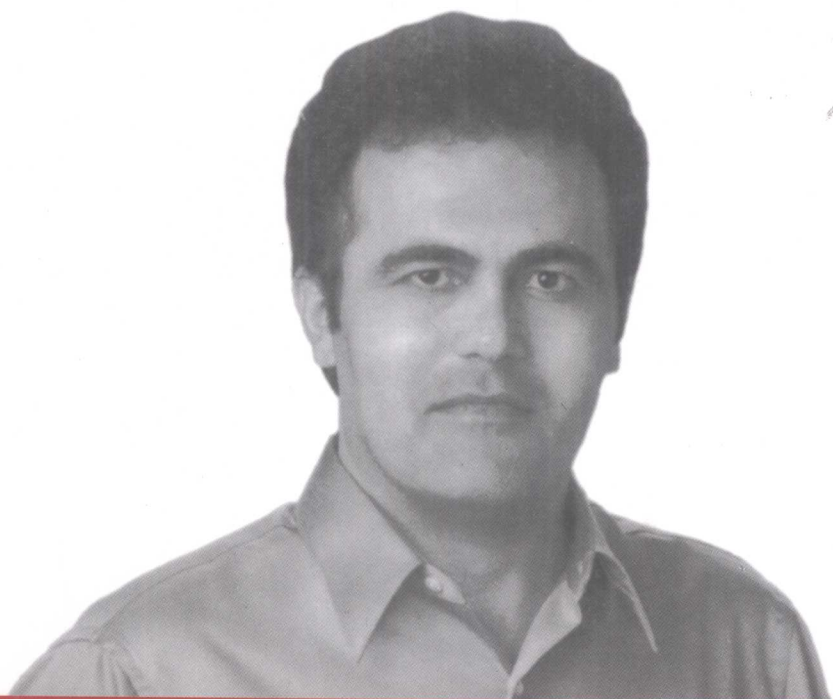


PROGRAMMER TO PROGRAMMER™



ASP.NET AJAX Programmer's Reference

# ASP.NET AJAX

# 编程参考手册

(涵盖 ASP.NET 3.5 及 2.0)

(美) Shahram Khosravi 著  
刘志忠 译



清华大学出版社

# ASP.NET AJAX

## 编程参考手册

(涵盖 ASP.NET 3.5 及 2.0)

(美) Shahram Khosravi 著

刘志忠 译

清华大学出版社

北 京

Shahram Khosravi

ASP.NET AJAX Programmer's Reference: with ASP.NET 2.0 or ASP.NET 3.5

EISBN: 978-0-470-10998-4

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: 01-2008-2280

本书封面贴有 Wiley 公司防伪标签, 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 AJAX 编程参考手册(涵盖 ASP.NET 3.5 及 2.0)/(美)霍斯拉维(Khosravi, S.)著; 刘志忠译. —北京: 清华大学出版社, 2009.3

书名原文: ASP.NET AJAX Programmer's Reference: with ASP.NET 2.0 or ASP.NET 3.5

ISBN 978-7-302-19482-8

I. A… II. ①霍…②刘… III. ①主页制作—程序设计—手册②计算机网络—程序设计—手册 IV. TP393.09-62

中国版本图书馆 CIP 数据核字(2009)第 016291 号

责任编辑: 王 军 李楷平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

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

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 78.25 字 数: 2504 千字

版 次: 2009 年 3 月第 1 版 印 次: 2009 年 3 月第 1 次印刷

印 数: 1~3000

定 价: 168.00 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: 027690-01

# 译者序

欢迎来到 AJAX 的奇妙世界！

随着技术的发展，人们对 Web 应用的要求不再仅仅是简单的信息浏览，而是要求能够和 Web 应用进行实时交互。但是，在几年前，为了和 Web 进行交互，您可能需要坐在浏览器前“浏览”大量的白屏。如今，您完全可以像使用桌面应用一样来使用 Web 应用。这一切都应归功于 AJAX(Asynchronous JavaScript and XML)这项新的技术。

五年前，如果不知道 XML，您就是一只无人重视的丑小鸭。今天，如果想跟上最新的技术时尚，那您的目标就是 AJAX。准确说来，AJAX 并不是一项新的技术，它是多项已有技术的一个组合，包括 JavaScript、XML、DOM 和 XHTML 等。这些技术的组合使得笨拙的 Web 界面转成交互式 AJAX 应用程序。支持 AJAX 的应用程序使用异步方式来和服务器进行通信，不中断终端用户和应用的交互，从而能够提高用户体验。另一方面，支持 AJAX 的应用实现“按需”更新，从而大大减少了网络传输的流量。因此，AJAX 将是未来 Web 应用发展的趋势。

本书以 ASP.NET 为工具，通过大量的示例，详细分析了 AJAX 的内部机制。本书首先介绍了为了支持 AJAX 而对 JavaScript 所进行的扩展，包括基本类型扩展、异常扩展、事件模型扩展以及 DOM 扩展等，之后本书就 AJAX 访问远程应用而展开了讨论，并就支持 AJAX 的底层实现机制展开了介绍，包括异步通信、UpdatePanel、ScriptManager 以及局部页面呈现技术。

本书主要侧重于对支持 AJAX 的应用的底层运行机制的分析与介绍，对应用并未进行深入探讨。以本书所介绍的应用为基础，您可以根据需求开发出各种各样的应用。

本书由刘志忠翻译，由肖国尊负责本书的翻译质量和进度。尽管我们已做了最大努力，但错误在所难免，敬请广大读者提供反馈意见，读者可以将意见发送到 [wkservice@163.vip.com](mailto:wkservice@163.vip.com)，我们会仔细查阅读者发来的每一封邮件，以求进一步提高今后译著的质量。

译者

2008 年 4 月

# 作者简介

**Shahram Khosravi** 博士是高级软件工程师、顾问、作家和导师，擅长于 ASP.NET、Windows 通信基础(Windows Communication Foundation, WCF)、ASP.NET AJAX、Windows 工作流基础(Windows Workflow Foundation, WF)、IIS7 和 ASP.NET 集成编程、ADO.NET、Web 服务、.NET 及 XML 技术——如 XSD、XSLT、XPath、SOAP 和 WSDL 等。他还拥有多年的面向对象分析、设计以及编程经验，熟悉软件体系结构、设计模式，同时精通面向服务的分析、设计与编程，还熟悉 3D 计算机图像编程、用户界面设计和可用性分析等。

Shahram 共著有如下 4 本著作：*Professional ASP.NET 3.5 and .NET 3.5 Programming (ASP.NET Internals plus ASP.NET AJAX, IIS 7.0, Enterprise Library Application Blocks, Windows Workflow Foundation, and Windows Communication Foundation)*、*ASP.NET AJAX Programmer's Reference with ASP.NET 2.0 or ASP.NET 3.5*、*Professional IIS7 and ASP.NET Integrated Programming* 及 *Professional ASP.NET Server Control and Component Development*。另外，他还在业界顶尖杂志上发表了与 ASP.NET、ADO.NET、.NET 及 XML 技术相关的文章多篇，如 *Dr. Dobb's Journal*、*asp.netPRO* 等杂志以及微软的 MSDN 在线文档等。

# 前 言

欢迎使用本书。ASP.NET AJAX 框架包括两个框架：ASP.NET AJAX 客户端框架和 ASP.NET AJAX 服务器端框架。

大家都知道，客户端编程和服务器端编程存在很大的差别。产生差别的主要原因是由于客户端编程缺乏像 ASP.NET/.NET Framework 一样功能丰富的编程框架。如果能够在具有像 ASP.NET/.NET 框架一样编程风格和能力的框架下编写您的客户端程序，那一定会得到不错的体验。使用 ASP.NET AJAX 客户端框架吧！它使得在客户端编程具有尽可能多的和 ASP.NET/.NET 一样丰富的编程特点。这些能力仅受限于客户端技术的基础限制，如 JavaScript 的限制。

ASP.NET AJAX 服务器端框架扩展了 ASP.NET 框架，从而使得服务器端能够支持那些支持 AJAX 的 Web 应用。ASP.NET AJAX 客户端框架和服务器端框架一同携手，来满足支持 AJAX 应用的客户端和服务器的需求。也就是说，ASP.NET AJAX 客户端技术可以和其他服务器端技术一起使用而不仅仅是 ASP.NET 框架。

本书使用独特的方法深入分析了 ASP.NET AJAX 客户端和服务器端框架，该方法具有以下特征。

- **实用真实例：**本书的讨论将基于大量的实际示例展开，这些示例都可以在您自己的 ASP.NET AJAX 应用中使用。
- **底层机制分析：**本书将深入分析 ASP.NET AJAX 客户端和服务端框架的底层机制，从而使您可以深入了解它们的工作原理，并对其进行扩展以满足您的应用需求。
- **代码逐行分析：**本书使用了大量的代码逐行分析，使得您可以在自己的 ASP.NET AJAX 应用中实现一些类似的功能。

## 本书读者对象

本书主要针对初次学习 ASP.NET AJAX 的 ASP.NET 开发人员。学习本书，并不需要具备 ASP.NET AJAX 的预备知识。

## 本书主要内容

本书共包括 24 章和 6 个附录，分别如下：

- **第 1 章 AJAX 相关技术** 本章通过示例对在支持 AJAX 的 Web 应用中所使用的一些主要技术进行一个简单综述，包括 XMLHttpRequest、XML 及 JSON 等。
- **第 2 章 JavaScript 基本类型扩展** 本章主要给出了 JavaScript 的基本类型扩展。在其中，您将了解到这些扩展类型如何改进了 JavaScript 的基本类型，包括 Array、Boolean、Date、Error、Object 及 String，并让您尽可能地像在 .NET 框架中一样应用这些类型。

- **第 3 章 内置的和自定义异常类型** 本章首先深入介绍了 ASP.NET AJAX 的一些内置的异常类型,包括 `ArgumentException`、`ArgumentNullException`、`ArgumentOutOfRangeException`、`ArgumentTypeException`、`ArgumentUndefinedException`、`InvalidOperationException`、`NotImplementedException` 及 `ParameterCountException`, 然后介绍了一种开发自定义异常类型的方法, 使用该方法实现了一个异常 `DuplicateItemException`, 然后通过一个页面来使用这个自定义的异常类型。
- **第 4 章 JavaScript 面向对象编程和类型反射扩展** 本章首先对 ASP.NET AJAX 面向对象编程(OOP)和类型反射扩展机制中一些在底层使用的 JavaScript 技术进行研究。这些技术使得 JavaScript 具有 OOP 属性并支持类型反射扩展。然后本章详细分析了 Type 类型及其方法。在此, 您可以通过学习大量的实例来了解如何定义名称空间、接口、类及枚举类型, 如何实现一个实现了一个或多个接口的类, 以及如何实现一个从其他类派生出来的类。
- **第 5 章 事件编程模型扩展** 本章将循序渐进地为您介绍一种实现自定义 ASP.NET 客户端类, 并为其添加事件的方法, 从而使得类的使用者能够扩展它们的功能来执行与应用相关的逻辑。然后提供了一个具体实例来使用这种方法。本章还讨论了 `EventArgs`、`CancelEventArgs` 和 `EventHandlerList` 等几个类及其方法和属性。
- **第 6 章 DOM 扩展** 本章主要解释了 ASP.NET AJAX DOM 扩展如何扩展传统的 DOM 编程技术, 从而使其支持 .NET 的方法和属性, 以及如何在自己的 DOM 编程中使用这些扩展。本章主要介绍 ASP.NET AJAX 委托、`DomElement` 和 `DomEvent` 客户端类及其方法和属性。
- **第 7 章 组件开发基础设施** 本章主要介绍 ASP.NET AJAX 组件开发的基础设施及其主要接口, 包括 `IDisposable`、`INotifyDisposing`、`INotifyPropertyChanged` 和 `IContainer` 以及两个重要的类 `Component` 和 `Application`。此外, 您还可以通过大量的实例学习到如何实现这些接口以及如何实现一个从 `Component` 基类派生出来的自定义组件。本章还包括组件的应用、组件的生命周期以及应用级事件等。
- **第 8 章 开发客户端控件** 本章描述了 `Control`、`Label`、`Image` 及 `HyperLink` 等客户端控件及其方法和属性, 并给出了使用这些自定义控件的相应示例。还给出并讨论了一个自定义 `Image` 控件的实现。该控件扩展了 ASP.NET AJAX 客户端 `Image` 控件的功能。
- **第 9 章 事件冒泡机制及客户端 Button 控件** 本章首先介绍 `CommandEventArgs` 事件数据类和客户端 `Button` 控件。然后讨论了 ASP.NET AJAX 事件冒泡机制, 并讨论了如何实现一个自定义的控件, 使其事件冒泡给它的父控件, 以及如何实现一个自定义控件来捕获它的子控件所冒泡的事件。本章还实现了一个自定义的客户端控件 `GridView`, 该控件使用了 ASP.NET AJAX 的事件冒泡机制。
- **第 10 章 类型描述扩展** 本章详细介绍了 `TypeDescriptor` 类和 `ICustomTypeDescriptor` 接口。基于此, 您将了解到 ASP.NET 类型描述能力为您的应用所带来的好处: 它使得您的客户端代码可以独立于客户端代码所处理的特定的对象类型。这将使得用户代码可以不经修改而与不同的对象类型进行交互。本章实现了 3 个 Web 页面, 在这些页面中您可以检测任何 ASP.NET AJAX 类型的属性、事件和方法。还实现一个自定义的

客户端控件 CustomTable。该控件使用 ASP.NET AJAX 类型描述能力来显示任意类型的数据记录。最后,本章介绍了动态注入元数据信息的方法。

- **第 11 章 数据类** 本章首先讨论了 IData 接口,然后深入介绍了 ASP.NET AJAX 的数据类 DataColumn、DataRow 和 DataTable。还实现一个自定义控件。该控件实现了 IData 接口并可以显示任何数据源(如 DataTable)的数据。
- **第 12 章 客户-服务器通信** 本章主要涉及到 ASP.NET AJAX 框架客户-服务器通信层及其主要的类,详细讨论了 WebRequest、WebRequestExecutor、WebRequestManager、NetworkRequestEventArgs 和 XMLHttpRequest,并结合一些实例来讨论如何在您自己的 ASP.NET AJAX 应用中使用这些类。
- **第 13 章 通过 SOAP 消息使用 Web 服务** 本章首先详细讨论了 WSDL 文档和 SOAP 消息,然后结合一个实例给出了使用 ASP.NET AJAX 框架的客户/服务器通信层的类来和 Web 服务交互 SOAP 消息的方法。
- **第 14 章 通过 JSON 消息使用 Web 服务** 本章深入讨论了 WebServiceProxy 和 WebServiceError 类,并给了 3 种不同的途径来实现客户端调用服务器端的方法:调用页面方法、Web 服务方法及 Web 服务桥。还详细分析了 .asbx 文件,给出并实现了 ASP.NET AJAX REST 方法调用-请求-处理基础设施的主要组件,包括 ScriptHandlerFactory、RestHandlerFactory、RestHandler、HandlerWrapper 和 ScriptModule 类,并实现了一个实例来使用这些方法。本章还使用这些方法分析了页面方法调用和 Web 服务桥。
- **第 15 章 代理类** 本章详细介绍了与页面方法、Web 服务桥及 Web 服务方法相关的代理类。接着,本章还讨论了服务器端 ScriptManager 和 ScriptManagerProxy 控件,以及服务器端 ScriptManagerProxy 控件在父/子页面方案中的作用。然后还实现了 ASP.NET AJAX 自动代理-类-产生器基础设施主要组件的完整功能流程,包括 ScriptManager、ServiceReferenceCollection、ServiceReference、ClientProxyGenerator 和 RestClientProxyHandler。您可以看到在应用中该基础设施如何产生与页面方法、Web 服务桥、Web 服务方法相关的代理类。最后本章给出了一个使用这些方法的实例。
- **第 16 章 行为** 本章详细介绍了 Behavior 基类及其方法和属性,并介绍了如何从该基类派生出新的类来实现自定义的行为。然后讨论了 ASP.NET AJAX 控件工具包中的行为基类 BehaviorBase,并讨论了如何从 BehaviorBase 类派生新的类来实现自定义的工具包行为。
- **第 17 章 服务器端 script 控件和 extender 控件** 本章深入介绍了 ASP.NET AJAX 服务器端框架主要组件的完整功能流程,该流程涉及两类重要服务器端控件的内部功能,即 script 控件和 extender 控件。分析这两类控件有助于您理解服务器端控件如何和相应的客户端组件进行交互,它们之间的异同以及如何实现自定义的 script 控件和 extender 控件。本章所实现的 ASP.NET AJAX 服务器端框架的组件包括: IExtenderControl、ExtenderControl、IScriptControl、ScriptControl、ScriptDescriptor、ScriptComponentDescriptor、ScriptBehaviorDescriptor、ScriptControlDescriptor、ScriptReference、ResolveScriptReference、ScriptReferenceCollection 和 ScriptManager。本章最后实现了自定义的服务



器端 script 控件和 extender 控件，从而使得您可以开发自己的服务器端 script 控件和 extender 控件。

- **第 18 章 Web 服务桥与转换器** 本章首先给出了一个使用 Amazon Web 服务的、支持 Web 服务桥的服务器端 script 控件，然后详细讨论了 ASP.NET AJAX 转换器，包括 XmlBridgeTransformer 和 XsltBridgeTransformer。接着，本章使用这些转换器改进了支持 Web 服务桥的服务端 script 控件。还介绍了如何实现自定义的转换器。
- **第 19 章 UpdatePanel 和 ScriptManager** 本章使用了大量实例来学习如何实现异步局部页面呈现，如何实现触发器以及通过多种不同的方法来实现有条件地更新服务器端 UpdatePanel 控件，包括：指定 ChildrenAsTrigger 属性，直接将控件添加到其他服务器端 UpdatePanel 控件，通过内容网页间接将控件添加到其他服务器端 UpdatePanel 控件，通过用户控件间接将控件添加到其他服务器端 UpdatePanel 控件，以及在代码中显式调用它的 Update 方法。最后，本章实现了两个自定义的、支持局部页面呈现的服务器端控件 BaseMasterDetailControl 和 BaseMasterDetailControl2。
- **第 20 章 在用户控件和自定义控件中使用 UpdatePanel** 本章实现了 3 个自定义的支持局部页面呈现的服务器端控件 MasterDetailControl、MasterDetailControl2 和 MasterDetailControl3，以及一个自定义的支持局部页面呈现的数据控件域 MasterDetailField 和一个支持局部页面呈现的线索化(threaded)讨论论坛用户控件。该章还用这些支持局部页面呈现的自定义服务器端控件、数据控件域和用户控件实现了一些页面。
- **第 21 章 页面生命周期和异步的局部页面呈现** 本章跟踪了 Page 对象处理第一次针对支持局部页面呈现的 Web 页面的请求的整个生命周期，从而使得您对 ASP.NET AJAX 异步局部页面呈现基础设施及其主要组件有一个深刻的认识，包括 ScriptManager 类和服务器端 PageRequestManager、UpdatePanel、UpdatePanelTrigger、UpdatePanelControlTrigger 及 AsyncPostBackTrigger 等类。本章还实现了一个自定义的服务器端 UpdatePanel 控件 CustomUpdatePanel 和一个定义的触发器 AsyncMultiPostBackTrigger。
- **第 22 章 ASP.NET AJAX 客户端 PageRequestManager** 本章首先详细介绍了当前客户端 PageRequestManager 实例的实例化和初始化过程。在此过程中您可以了解到该实例的 pageLoaded 事件及其对应的 PageLoadedEventArgs 事件数据类型。另外，还介绍了该事件的一个使用示例。接着，深入分析了当前客户端 PageRequestManager 实例向服务器发送一个异步页面回传请求的过程，其中您可以了解到 PageRequestManager 实例的 initializeRequest 和 BeginRequest 事件。同样，本章也给出了一个示例，解析如何在您的 ASP.NET AJAX 应用中使用这些事件。
- **第 23 章 异步局部页面呈现：服务器端处理** 本章主要跟踪了 Page 对象处理异步页面回传请求的整个生命周期。在此过程中，您可以了解到产生最终服务器响应文本时服务器端 PageRequestManager、RetrievePostData、ScriptManager、UpdatePanel、ScriptRegistrationManager 以及触发器的作用。本章还实现了一个页面，使得您可以深入分析服务器端响应文本。
- **第 24 章 异步局部页面呈现：客户端处理** 本章跟踪了客户端 PageRequestManager 实例处理服务器针对某异步页面回传的响应的生命周期。在此过程中，您将可以看到现

有的客户端 `PageRequestManager` 实例如何实现服务器响应文本的解析，下载需要的脚本，并更新客户端所需要的客户端 `UpdatePanel` 控件。通过本章，您可以了解到当前客户端 `PageRequestManager` 实例的 `pageLoading` 和 `endRequest` 方法及其相应的 `PageLoadingEventArgs` 和 `EndRequestEventArgs` 事件数据类。本章给出了一个示例来说明这些事件和相应的事件数据类如何使用，包括当前客户端 `PageRequestManager` 实例所产生的异常：`PageRequestManagerTimeoutException`、`PageRequestManagerServerErrorException`、`PageRequestManagerParserErrorException`、`InvalidOperationException` 等。最后，本章还实现了一个自定义的错误处理器和一个使用该错误处理器的页面。

- **附录 A XML 脚本** 该附录提供了详细的 ASP.NET AJAX XML 脚本，从而使得用户可以通过声明式的形式来编写 JavaScript 代码而只需要很少的甚至不需要命令式或者程序式的 JavaScript 代码。该附录涵盖了 ASP.NET AJAX XML 脚本解析基础设施的主要组件，如 `MarkupContext` 和 `MarkupParser`。在本章，您可以通过大量的示例来学习如何让您的客户端类的客户端能够声明式地实例化和初始化您在 XML 脚本中定义的类，而不需要写一些过程化的 JavaScript 代码。您还可以学习如何扩展 ASP.NET AJAX XML 脚本解析基础设施来增加对客户端类的自定义解析。
- **附录 B 绑定** 主要涉及 ASP.NET AJAX 绑定。该部分深入地讨论了客户端 `BindingBase` 类、内置的和自定义的转换器以及客户端的 `Binding` 类。
- **附录 C 动作** 讨论了 ASP.NET AJAX 动作相关的主要内容，包括客户端 `IAction` 接口、`Action` 基类，在 XML 脚本中的行为以及内置行为，如 `InvokeMethodAction`、`SetPropertyAction` 和 `PostBackAction`。
- **附录 D 数据控件** 该部分首先介绍了 ASP.NET AJAX `DataControl` 基类及其相应的方法、属性和事件。然后实现了一个自定义的数据控件 `CustomTable`。该控件派生于 `DataControl` 基类，并使用了 ASP.NET AJAX 类型描述能力来显示任意类型的数据记录。
- **附录 E 模板化控件** 该部分首先详细介绍了客户端 `ITemplate` 接口和客户端 `TemplateInstance` 和 `Template` 类。然后开发了一个自定义模板 `TemplateField`，该类派生于 `Template` 类并支持自己的 `parseFromMarkup` 静态方法。该方法限定 ASP.NET AJAX XML 脚本解析基础设施如何解析一个在 XML 脚本中声明的 `TemplateField` 类的实例。最后，该附录开发了一个自定义的模板化数据控件，使得用户可以在 XML 脚本中使用 `TemplateField` 实例来限定不同类型的 HTML 标记文本显示不同类型的数据区域。
- **附录 F ListView** 本附录首先简单介绍了 ASP.NET AJAX `ListView` 客户端控件及其方法、属性和事件。接着给出了一个示例来说明该客户端控件如何显示从后台 Web 服务中下载得到的数据记录。然后深入分析了客户端 `ListView` 控件及其方法、属性和事件，以及与此相关的类和接口，如 `ITask`、`_TaskManager` 和 `ListViewRenderTask`。在此您还将学会如何开发一个和客户端 `ListView` 控件一样复杂的自定义模板化数据控件。

## 使用本书所需要的准备工作

运行本书的代码，需要如下的运行环境：

- ASP.NET AJAX Extensions 1.0
- ASP.NET Futures
- Windows Server 2003、Windows 2000、Windows XP 或者 Windows Vista
- Visual Studio 2005、Visual Studio 2005 Express Edition、Visual Studio 2008 或者 Visual Studio 2008 Express Edition
- SQL Server 2005 或 SQL Server 2005 Express Edition

您可以从 <http://msdn.microsoft.com/vstudio/express/> 上下载免费的 Visual Studio 2005 Express 版或者 Visual Studio 2008 Express 版，从 <http://msdn.microsoft.com/vstudio/express/> 下载免费的 SQL Server 2005 Express 版，并从 <http://ajax.asp.net/downloads/> 下载 ASP.NET AJAX Extensions 1.0 和 ASP.NET Futures。

## 源代码

在阅读学习本书中的示例时，可以手工输入所有的代码，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合作站点 <http://www.wrox.com/> 或 [www.tupwk.com.cn/downpage](http://www.tupwk.com.cn/downpage) 上下载。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-10998-4。

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@vip.163.com](mailto:wkservice@vip.163.com) 发电子邮件，我们就会检查您的反馈信息，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英文版的勘误表，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](http://www.wrox.com/misc-pages/booklist.shtml)。

## P2P.WROX.COM

要与作者和同行讨论，请加入 [p2p.wrox.com](http://p2p.wrox.com) 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 [p2p.wrox.com](http://p2p.wrox.com)，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

### 注释：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

# 目 录

第 1 章	AJAX 相关技术	1
1.1	Google Suggest	1
1.2	XMLHttpRequest	3
1.3	XML	13
1.4	JSON	17
1.4.1	对象	17
1.4.2	数组	17
1.4.3	字符串	17
1.4.4	数字	18
1.4.5	null、true 和 false	18
1.5	ASP.NET AJAX	21
1.6	安装 ASP.NET AJAX Extensions 和 ASP.NET Futures	22
1.7	小结	22
第 2 章	JavaScript 基本类型扩展	23
2.1	ASP.NET AJAX Array 类型扩展	23
2.1.1	add	24
2.1.2	addRange	24
2.1.3	clear	25
2.1.4	clone	25
2.1.5	contains	26
2.1.6	enqueue 和 dequeue	26
2.1.7	forEach	28
2.1.8	indexOf	29
2.1.9	insert	30
2.1.10	parse	30
2.1.11	remove	31
2.1.12	removeAt	31
2.2	ASP.NET AJAX Boolean 类型 扩展	32
2.3	ASP.NET AJAX Date 类型扩展	32
2.4	ASP.NET AJAX Object 类型 扩展	33
2.5	ASP.NET AJAX String 类型 扩展	33
2.5.1	endsWith	33
2.5.2	startsWith	34
2.5.3	trim	34
2.5.4	格式化	35
2.6	ASP.NET AJAX Error 类型扩展	35
2.6.1	create	36
2.6.2	popStackFrame	38
2.6.3	popStackFrame 的内部机理	41
2.7	小结	44
第 3 章	内置的和自定义异常类型	45
3.1	ASP.NET AJAX 内置异常类型	45
3.1.1	ArgumentException	45
3.1.2	ArgumentNullException	47
3.1.3	ArgumentOutOfRangeException	49
3.1.4	ArgumentTypeException	51
3.1.5	ArgumentUndefinedException	53
3.1.6	InvalidOperationException	55
3.1.7	NotImplementedException	56
3.1.8	ParameterCountException	57
3.2	实现自定义异常类型	59
3.2.1	实现自定义异常类型的流程	59
3.2.2	使用流程	60
3.2.3	clickCallback	63
3.2.4	validateInput	64
3.3	小结	64
第 4 章	JavaScript 面向对象编程和 类型反射扩展	65
4.1	JavaScript 函数	65
4.2	JavaScript 类	66
4.3	Type	67
4.4	registerClass	68

4.5	getName	70
4.6	isClass	71
4.7	registerNamespace	72
4.8	isNamespace	74
4.9	registerInterface	75
4.10	getInterfaces	78
4.11	isInterface	80
4.12	继承	81
4.13	getBaseType	84
4.14	initializeBase	87
4.15	resolveInheritance	87
4.16	callBaseMethod	88
4.17	getBaseMethod	93
4.18	implementsInterface	96
4.19	inheritsFrom	100
4.20	isImplementedBy	102
4.21	getRootNamespaces	104
4.22	parse	106
4.23	registerEnum	108
4.24	isEnum	109
4.25	isFlags	110
4.26	小结	110
<b>第 5 章</b>	<b>事件编程模型扩展</b>	<b>111</b>
5.1	事件编程模型	111
5.1.1	Sys.EventArgs	112
5.1.2	Sys.CancelEventArgs	112
5.1.3	EventHandlerList	112
5.2	使用事件编程	116
5.2.1	基本类	117
5.2.2	事件	122
5.2.3	ShoppingCart	130
5.2.4	添加一个事件处理程序	134
5.2.5	移除一个事件处理程序	134
5.3	小结	135
<b>第 6 章</b>	<b>DOM 扩展</b>	<b>136</b>
6.1	DomElement	136
6.1.1	getElementById	136
6.1.2	addClass	141
6.1.3	containsCssClass	141
6.1.4	removeCssClass	141
6.1.5	toggleCssClass	143
6.1.6	getLocation	144
6.1.7	setLocation	145
6.1.8	getBounds	147
6.2	MouseButton	148
6.3	Key	149
6.4	委托	149
6.4.1	Namespace	155
6.4.2	Mover	155
6.4.3	TextProvider	155
6.4.4	ImageProvider	156
6.5	DomEvent	156
6.5.1	构造函数	157
6.5.2	静态方法	159
6.5.3	实例方法	166
6.6	使用 DomEvent 类	170
6.6.1	Mover	176
6.6.2	TableProvider	180
6.7	小结	181
<b>第 7 章</b>	<b>组件开发基础设施</b>	<b>182</b>
7.1	接口	182
7.1.1	IDisposable	183
7.1.2	INotifyDisposing	186
7.1.3	INotifyPropertyChanged	189
7.2	组件	194
7.3	IContainer	196
7.4	应用	197
7.4.1	addComponent	198
7.4.2	removeComponent	199
7.4.3	getComponents	199
7.4.4	findComponent	200
7.5	应用生命周期	201
7.6	组件生命周期	205
7.7	继续应用生命周期之旅	209
7.7.1	endCreateComponents	209
7.7.2	raiseLoad	211

7.7.3 应用生命周期小结	212	8.3.2 htmlEncode	239
7.8 应用层事件	213	8.3.3 text	240
7.8.1 Init	213	8.3.4 prototype	241
7.8.2 load	214	8.3.5 descriptor	241
7.8.3 unload	214	8.4 使用 Label 客户端控件	242
7.9 可丢弃对象	214	8.5 客户端 Image 控件	244
7.10 使用 Application 对象和 Component 基类	217	8.5.1 构造函数	244
7.10.1 dispose	217	8.5.2 prototype	244
7.10.2 initialize	220	8.5.3 imageURL	245
7.10.3 id	222	8.5.4 width	245
7.10.4 raisePropertyChanged	222	8.5.5 height	246
7.10.5 get_events	223	8.5.6 alternateText	246
7.10.6 INotifyPropertyChanged	224	8.6 使用 Image 客户端控件	246
7.10.7 INotifyDisposing	224	8.7 扩展客户端 Image 控件	248
7.11 小结	230	8.7.1 构造函数	251
<b>第 8 章 开发客户端控件</b>	<b>231</b>	8.7.2 prototype	252
8.1 Control	231	8.7.3 imageURL	252
8.1.1 定义	231	8.7.4 mouseOverImageURL	253
8.1.2 get_element	232	8.7.5 mouseOverCallback	253
8.1.3 get_id	233	8.7.6 mouseOutCallback	254
8.1.4 set_id	233	8.7.7 Duration	254
8.1.5 set_parent	233	8.7.8 transition	255
8.1.6 get_parent	234	8.7.9 Transition 枚举类型	255
8.1.7 get_visibilityMode	234	8.7.10 initialize	256
8.1.8 get_visible	235	8.7.11 dispose	256
8.1.9 set_visibilityMode	235	8.8 使用客户端 Image2 控件	257
8.1.10 set_visible	236	8.9 HyperLink 客户端控件	259
8.1.11 addCssClass	236	8.9.1 构造函数	259
8.1.12 removeCssClass	236	8.9.2 prototype	259
8.1.13 toggleCssClass	237	8.9.3 navigateURL	259
8.1.14 dispose	237	8.9.4 initialize	260
8.1.15 onBubbleEvent	237	8.9.5 add_click	260
8.1.16 raiseBubbleEvent	238	8.9.6 remove_click	261
8.2 开发自定义客户端控件	238	8.9.7 _onClick	261
8.3 客户端 Label 控件	239	8.9.8 dispose	261
8.3.1 构造函数	239	8.9.9 descriptor	262
		8.10 使用 HyperLink 客户端控件	262
		8.11 小结	264

<b>第 9 章 事件冒泡机制及客户端 Button 控件</b> .....265	<b>第 11 章 数据类</b> ..... 335
9.1 CommandEventArgs.....265	11.1 IData.....335
9.2 客户端 Button 控件.....266	11.2 DataColumn.....336
9.2.1 构造函数.....267	11.3 DataRow.....339
9.2.2 prototype 属性.....267	11.3.1 构造函数.....339
9.2.3 argument 属性.....267	11.3.2 descriptor.....340
9.2.4 command 属性.....268	11.3.3 ICustomTypeDescriptor.....341
9.2.5 initialize 方法.....268	11.3.4 Owner.....345
9.2.6 add_click 方法.....269	11.3.5 INotifyPropertyChanged.....346
9.2.7 _onClick.....269	11.4 DataTable.....347
9.2.8 dispose.....270	11.4.1 构造函数.....347
9.2.9 descriptor.....270	11.4.2 IData.....349
9.3 使用客户端 Button 控件.....271	11.4.3 Descriptor.....356
9.3.1 捕获一个冒泡的事件.....271	11.4.4 INotifyPropertyChanged.....357
9.3.2 冒泡一个事件.....276	11.4.5 InotifyCollectionChanged.....359
9.4 小结.....286	11.4.6 createRow.....362
<b>第 10 章 类型描述扩展</b> .....287	11.4.7 getChanges.....363
10.1 TypeDescriptor.....287	11.4.8 getColumn.....364
10.1.1 构造函数.....288	11.4.9 raiseRowChanged.....365
10.1.2 getTypeDescriptor.....304	11.4.10 parseFromJson.....365
10.1.3 getProperty.....308	11.5 使用 DataColumn、DataRow
10.1.4 getAttribute.....310	和 DataTable.....367
10.1.5 setProperty.....311	11.6 小结.....376
10.1.6 invokeMethod.....315	<b>第 12 章 客户-服务器通信</b> ..... 377
10.1.7 getPropertyType.....316	12.1 WebRequest.....377
10.2 使用 ASP.NET AJAX 类型	12.1.1 构造函数.....377
描述能力.....317	12.1.2 目标 URL.....378
10.2.1 StringBuilder.....318	12.1.3 HTTP 动作.....378
10.2.2 CustomTable.....321	12.1.4 请求体.....378
10.3 动态注入元数据信息.....329	12.1.5 超时.....379
10.3.1 addProperty.....329	12.1.6 Web 请求执行器.....379
10.3.2 addMethod.....330	12.1.7 请求头.....380
10.3.3 addEvent.....331	12.1.8 completed 事件.....380
10.3.4 addAttribute.....332	12.1.9 调用一个 Web 请求.....381
10.4 ICustomTypeDescriptor.....332	12.2 WebRequestExecutor.....382
10.5 小结.....334	12.2.1 构造函数.....382
	12.2.2 WebRequest.....382



12.2.3	get_started	383	12.8	小结	422
12.2.4	get_responseAvailable	383	<b>第 13 章</b>	<b>通过 SOAP 消息使用 Web 服务</b>	<b>423</b>
12.2.5	get_timedOut	383	13.1	构建 Web 服务	423
12.2.6	get_aborted	384	13.2	WSDL 文档	424
12.2.7	get_responseData	384	13.2.1	参数的名称、类型和顺序	426
12.2.8	get_statusCode	384	13.2.2	返回值的类型和顺序	426
12.2.9	get_statusText	384	13.2.3	方法描述	427
12.2.10	get_xml	384	13.2.4	描述访问方法的通信协议	427
12.2.11	get_object	385	13.2.5	指定访问方法的站点	428
12.2.12	executeRequest	385	13.2.6	指定方法类	429
12.2.13	abort	385	13.3	SOAP 消息	429
12.2.14	getResponseHeader	385	13.4	小结	441
12.2.15	getAllResponseHeaders	386	<b>第 14 章</b>	<b>通过 JSON 消息使用 Web 服务</b>	<b>442</b>
<b>12.3</b>	<b>WebRequestManager</b>	<b>386</b>	14.1	WebServiceProxy	442
12.3.1	构造函数	386	14.1.1	Timeout	442
12.3.2	默认超时	387	14.1.2	默认成功回调函数	442
12.3.3	默认执行器类型	387	14.1.3	默认失败回调函数	443
12.3.4	事件	387	14.1.4	路径	443
12.3.5	执行一个 Web 请求	389	14.1.5	调用一个 Web 方法	443
12.3.6	NetworkRequestEventArgs	390	14.2	使用 WebServiceProxy	454
<b>12.4</b>	<b>XMLHttpRequest</b>	<b>391</b>	14.3	WebServiceError	457
<b>12.5</b>	<b>XMLDOM</b>	<b>391</b>	14.4	使用 WebServiceError	460
<b>12.6</b>	<b>XMLHttpRequestExecutor</b>	<b>392</b>	14.5	调用页面方法	464
12.6.1	构造函数	392	14.6	调用自定义方法	466
12.6.2	get_timedOut	395	14.7	底层机制	472
12.6.3	get_started	395	14.7.1	ScriptHandlerFactory	472
12.6.4	get_responseAvailable	395	14.7.2	RestHandlerFactory	476
12.6.5	get_aborted	396	14.7.3	RestHandler	478
12.6.6	执行请求	396	14.7.4	HandlerWrapper	483
12.6.7	getResponseHeader	398	14.7.5	页面方法揭秘	484
12.6.8	getAllResponseHeaders	399	14.7.6	Web 服务桥揭秘	489
12.6.9	get_responseData	399	14.7.7	使用这些复制品	492
12.6.10	get_statusCode	400	14.8	小结	494
12.6.11	get_statusText	400	<b>第 15 章</b>	<b>代理类</b>	<b>495</b>
12.6.12	get_xml	400	15.1	什么是代理?	495
12.6.13	abort	401			
<b>12.7</b>	<b>使用 WebRequest、WebRequestManager 和 XMLHttpRequestExecutor</b>	<b>402</b>			