

庖丁解牛

纵向切入

ASP.NET 3.5

控件和组件开发技术

郑健 著

不是诚心想成为杰出ASP.NET程序专家者，请勿浪费时间翻阅本书。

——胡德民

相信您阅读过此书后，会像庖丁一样，有一种提书而立，为之四顾，为之踌躇满志的感觉。

——王洪超



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

微软MVP作品系列
—Developer



庖丁解牛

纵向切入

ASP.NET 3.5

控件和组件开发技术

郑健 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要介绍ASP.NET的控件开发,书中通过70多个例子讲解了ASP.NET控件开发技术的各个方面,而且剖析了很多控件中系统基类源代码,读者从这些系统源代码可以体会设计模式思想。如果扎实地掌握了ASP.NET控件的运行机制,开发一个页面级的ASP.NET应用程序会变得非常简单。本书宗旨就是让开发人员真正理解ASP.NET技术,帮助开发人员提高ASP.NET开发的技术水平。学完本书后您不仅能够掌握控件开发各个方面的技术,而且深晓ASP.NET的工作原理。对大部分使用ASP.NET技术开发两年左右的开发人员来说,本书75%以上的内容可能没有接触过,本书将通过笔者精心准备的一些生动的例子来阐述这些内容。

本书适合从事ASP.NET技术多年想往深度理解ASP.NET工作机制的开发人员阅读,对于刚刚入门的初级开发人员,本书的前面几章也介绍了几个入门性的应用例子,本书还可以作为在校学生的就业预备教材。针对读者安装的开发环境不同,笔者分别为VS 2005和VS 2008建立了解决方案,即无论您是使用VS 2005还是VS 2008都可以成功打开本书随书光盘的代码,只是使用VS 2005打开的解决方案不包含.NET 3.5对应章节内容的示例代码。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

庖丁解牛:纵向切入ASP.NET 3.5控件和组件开发技术 / 郑健著. —北京:电子工业出版社, 2009.4
(微软MVP作品系列)

ISBN 978-7-121-07979-5

I. 庖… II. 郑… III. 主页制作—程序 IV. TP395.092

中国版本图书馆CIP数据核字(2008)第196518号

责任编辑:许 艳

印 刷:北京天宇星印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

开 本:787×1092 1/16

印张:41.5

字数:1012.5千字

彩插:4

印 次:2009年4月第1次印刷

印 数:4000册 定价:85.00元(含光盘2张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

推荐序



中学的时候，在语文课堂上学习过庖丁解牛的寓言故事。生活在战国时期的著名思想家庄子，以其神来之笔向我们介绍了一位古代的“牛人”庖丁，通过长期的实践，在掌握了“牛”的生理结构后，“解牛”时可以得心应手，运用自如，尤其是最后的描述“提刀而立，为之四顾，为之踌躇满志，善刀而藏之”，让吾辈神往不已，一代“解牛”大侠，颇有独孤求败的气势！

后来我又学过一篇寓言故事，即宋朝欧阳修所著的《卖油翁》，同样描述了一位经过反复实践，在酌油时，油可以“自钱孔入，而钱不湿”的卖油翁，这其实也是一位神人。而且卖油翁非常谦虚，在总结自己的成功经验时，只说了一句“无他，但手熟尔”。

在很长的一段时间内，我一直认为这两篇故事讲述的是同一个道理，所有的技能通过反复的磨炼，最后在实践中均可以做到得心应手，运用自如，均为“无它，但手熟尔”。

三十六行，行行出状元。我也一直将各种程序界的“牛”人比喻为庖丁或者卖油翁，认为他们可以非常熟练地掌握各种相关知识，在编写程序时信手拈来，拖拖拽拽之间，最终的应用程序成型出炉，最后提“鼠标”而立，为之四顾，为之踌躇满志。当有后辈请教时，非常谦虚地面带微笑曰：我亦无它，惟手熟尔。让周围手不熟的同事也为之神往不已。

但后来我发现，其实这些神人之间还是有很大差异的，这种差异其实正好体现了卖油翁与庖丁的不同境界。

回顾一下，在ASP.NET之后，Web编程开始变得异常简单，因为它将更多的实现细节隐藏在框架内部。大部分需求场景甚至仅需要拖曳控件就可以了，不需要编写任何代码。更有甚者，从ASP.NET 2.0开始，包括用户管理（注册、找回密码、更改密码等）、导航（站点地图、菜单等）在内的场景都不需要再编写代码，只需要将相应的控件拖到表单上即可以完成编码。

这在一定程度上使得生产力得以大大提升，开发人员不必了解具体的开发细节，只需要将这些积木式的“控件”所完成的功能了解清楚即可。在工作中，随心所欲地玩转这些控件，似乎已经成为了ASP.NET编程高手的象征。而在遇到问题时，大部分ASP.NET开发人员的第一个想法就是去互联网上寻找现成的控件。

这样，很多开发人员可以在经过短时间的磨炼后，就可以面带微笑得向其新来同事笑曰：我亦无它，惟手熟尔。但相信在这些“谦虚”的“卖油翁”当中，有很多只是“知其然，而不知其所以然”。

所以，与ASP.NET程序员的增长速度一样，在论坛上，一些类似的ASP.NET 开发问题也在呈几何级数的增加：

“服务器控件与客户端控件有什么区别？”

“动态创建的Button的单击事件怎么不执行？”

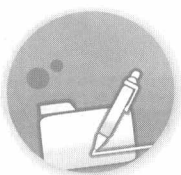
“为什么点击Button时，总是先触发PageLoad事件？”

所以作为一位开发人员，我们不应该只梦想着成为卖油翁，还要立志成为“知其然，更知其所以然”的庖丁。

郑健的这本《庖丁解牛—纵向切入ASP.NET 3.5控件和组件开发技术》恰恰是为那些已经可以熟练使用ASP.NET控件，但却渴望了解其内在运行机制的朋友们所准备的。本书以极为详尽的笔墨，向大家介绍了ASP.NET 3.5控件和组件的相关机制，并且提供了详细的示例代码，让你在实践中逐步找到共通的原理，达到举一反三的目的。相信您阅读过此书后，会有一种提书而立，为之四顾，为之踌躇满志的感觉。

祝愿每位朋友都在工作中可以成为程序界的“庖丁”。

王洪超
微软(中国)有限公司
互联网架构首席顾问
2008/12/16



推荐序

技术是为市场服务的，当下的互联网络局已经发生了巨大的改变。网络服务的崛起和高速发展已经说明了其市场需求和客户意愿。然而网络服务需要以技术作为驱动力，Web开发技术作为其重要的实现部分正在走向新的阶段。

大家通常认为ASP.NET仅是一种语言，但准确地讲，ASP.NET是Web开发技术中的一种，ASP.NET是一种技术而不是一种语言，了解一门技术的前提是要理解其工作原理。相对来说，学习ASP.NET技术要比学习一门语言要难许多。而本书的宗旨就是让开发人员真正的理解ASP.NET技术，帮助开发人员提高ASP.NET开发技能。书中内容主要以控件开发为主，如果扎实地掌握了ASP.NET控件的运行机制，则开发一个页面级的ASP.NET应用程序会变得非常简单，学完本书后您不仅能够掌握控件开发各个方面的技术，而且能够知道ASP.NET的工作原理。

此外，从事控件开发工作的好处是有更多机会进行面向对象的程序开发，比如控件中的接口非常多，而一个好的控件不仅在于功能强大，性能好，也在于是否易重用，是否易扩展，是否能用尽量少的代码实现我们的必要功能。看完本书，您将能够理解ASP.NET页面以及其中每个元素的生成过程。ASP.NET开发实质上就是应用控件的过程，我们在页面中使用的每个控件，甚至页面都是一个派生自System.Web.UI.Page控件的类，只不过Page控件是一个特殊的容器控件，但是它的生命周期与一般控件生命周期是一样的，在书中也有一些讲解Page控件运行机制的知识点。当知道了控件的工作原理再来开发一个页面就会非常容易，也可以说学习控件开发是迅速掌握ASP.NET技术的有效途径。

本书注重理论实践结合，每个ASP.NET控件开发技术知识点，基本上都会有对应的应用示例。在章节方面，按照技术点类别进行分类，对每个控件开发技术点，都进行了详尽的阐述，内容比较全面，因此本书也可以作为查找手册使用。

本书由微软中国区Visual C#技术微软最有价值专家（Most Valuable Professionals, MVP）郑健撰写。郑健先生作为微软MVP有多年的微软技术项目开发经验，他撰写过多部关于.NET/ASP.NET等方面的专著和文章，在参与微软相关的开发项目中收集了很多技术反馈和用户体验，这也为本书的创作打下了良好的基础。

微软最有价值专家（Microsoft MVP）是一群令人敬佩的人。他们都是具备一种或多种微软技术专业知识和积极参与在线或离线的社群活动，经常与其他专业人士分享知识和专业技能，受人尊敬、信任，而且平易近人的专家。通过分享其知识和经验并提供客观反馈，MVP帮助人们解决问题和发现新的潜能。

微软认为，技术社区为用户提供了技术交流的机会，这些交流促进变革和创新，从而提升大众的生活品质并促进行业的成功。郑健身为社区专家，积极活跃在线上和线下社区，不仅与广大专业人士分享微软的最新技术和自身的技术经验，还帮助用户更快和更有效地采用新的技术。郑健及其他MVP在社区所做贡献的同时也在帮助微软产品开发人员把握用户的“脉搏”和更好地满足客户的需要。

微软公司的ASP.NET技术经过了三个版本的大演变，已经变为比较成熟的技术，目前在软件企业应用中也广泛被采纳。通过阅读本书，我相信能够使您重新认识ASP.NET技术，并把它运用到实际的工作中。本书适合从事ASP.NET技术多年、想往深入理解ASP.NET工作机制的工作人员，考虑到对ASP.NET刚刚入门的初级开发人员，在本书前面几章也介绍了几个帮助他们入门的应用例子，所以本书也可以作为在校学生的就业预备教材。本书可以让您理解最流行的基于Web架构的应用程序呈现过程，阅读完本书之后，您会发现这是一本好书，它会成为您工作中的良师益友。

技术社区支援服务中国区最有价值专家主管

林思綺

很多开发人员都有这样的体会，使用ASP.NET开发很多年，但觉得水平总是提高不了，甚至我发现有超过五年工作经验的资深开发人员对ASP.NET技术掌握得也不是很深。其主要原因是在开发过程只注重往广度（横向）去学习各种技术，包括各种最新技术，而没有在掌握一个技术后往深度（纵向）去深入研究。还有很多开发人员虽然具有多年工作经验，但他们只是使用ASP.NET技术开发页面级的应用程序（页面开发人员），而没有或者说没有机会接触到ASP.NET的底层技术，比如Web应用程序的页面呈现过程等，这里所说的技术并不仅限于ASP.NET，而是指基于Web平台的B/S架构下必须要掌握的东西。比如一头牛，看上去似乎浑身都是肉，但当我们拿着刀去宰一头牛时，就会知道没那么容易，如果无规律地硬砍不但会费很大的力气，而且很可能会最终把牛砍得一片狼藉，原因在于对牛的身体组织结构不熟悉而无法正确地入刀；而《庄子·养生主》中为文惠君宰牛的庖丁则不然，其刀入牛身若“无厚入有间”而游刃有余，因为他对牛的结构了如指掌，这就是所谓的深度（纵向深入研究）。学习ASP.NET技术也是这样，如果在工作中只是对技术浅尝辄止，则很难提高自身的技术水平，本书主旨正是用“庖丁解牛”之术解析ASP.NET，从深度讲解ASP.NET的工作机制。

● 本书主旨及本书特色

准确地讲，ASP.NET是一种技术而不是一种语言，掌握一门技术的前提是要理解其工作原理。相对来说，学习ASP.NET技术要比学习一门语言要难许多。而本书的宗旨就是让开发人员真正地理解ASP.NET技术，帮助开发人员提高ASP.NET开发能力。书中内容主要以控件开发为主，如果扎实地掌握ASP.NET控件的运行机制，则开发一个页面级的ASP.NET应用程序会变得非常简单，学完本书后您不仅能够掌握控件开发各个方面的技术，而且能够透彻理解ASP.NET的工作原理。

本书不是讲解怎么使用ASP.NET控件，而是实实在在地讲解怎么开发控件。在讲解过程中剖析了很多控件系统基类源代码，因为多接触些.NET Framework的源代码对设计模式思想的理解也会更深刻，相对于书中的理论这其实就是实践。

控件开发与设计模式有何相干？在此简单谈谈设计模式。在继承系统基类开发自定义控件的过程中，细心的开发人员可能会发现基本上系统的所有的基类都有一个最基本的设计模式——模板方法模式，这个设计模式的特点是：由基类定义骨架，由基类驱动（比如控件的生命周期各个阶段方法的执行就是由基类驱动的），由基类提供最普遍的默认实现，由基类抽取出不变代码实现代码重用。这样的好处是，在大多数情况下，我们在开发自己的自定义控件时，写很少的代码（或者什么都不写）就可以实现一个控件，举个例子，请看一下如下代码：

```
Public KingTextBox : TextBox
```

```
{ }
```

以上代码中没有写一句实现代码，但它仍然是一个完整的控件，重用了基类中的所有功能，当我们需要功能与基类的功能不同时，仅需要重写（override）基类个别方法，即可实现必要的功能，其余的功能仍然使用基类的默认实现，这里要说明的是，实现自定义控件类不像实现我们自己的业务类一样对类中的每个方法都要亲自编码实现。此外，在控件开发技术中，使用这种设计模式可以让烦琐的事情由系统基类来统一完成，在一定程度上可以避免开发人员出错的可能性。

前言

设计模式是面向对象的公式，我个人也看过不少设计模式方面的书，也对设计模式比较感兴趣。个人觉得学习设计模式最好的办法是先找几本书看一遍，吃透，再看大量的源代码，经常会有顿悟的感觉；否则可能看懂了一段经典设计模式的代码，也不知道是什么设计模式，只是读懂了代码。在控件开发中，如果你想了解得更多，必须要看.NET Framework中的系统控件代码，在不知不觉中就会提高自己的开发能力和领悟面向对象的思想。

由于本书内容主旨不是讲设计模式，这里就谈这么多，如果想了解更多面向对象的话，还要看一下系统基类之前的关系，想一下它们的类为什么要这样设计，我仍然要提醒的是在看系统源代码时不要局限于仅看懂代码。

书中有些重要的知识点一般开发人员可能永远不会在工作中用到，但这些知识可以让您对ASP.NET技术理解得更透彻、更深刻。

控件开发工作的好处是有更多机会实战面向对象，比如控件中的接口非常多，一个好的控件不仅在于功能强大，性能好，也在于是否易重用，易扩展，是否能用尽量少的代码实现所需的功能，掌握了控件开发，您也就修炼成一位面向对象编程的高手。

看完本书，您将能够理解ASP.NET页面以及其中每个元素的生成过程。ASP.NET开发实质上就是应用控件的过程，我们在页面中使用的每个控件，甚至页面都是一个派生自System.Web.UI.Page控件的类，只不过Page控件是一个特殊的容器控件，但是它的生命周期与一般控件生命周期是一样的，在书中也讲解了一些Page控件运行机制的知识。当了解了控件的工作原理后开发一个页面就会非常容易，也可以说学习控件开发是加速掌握ASP.NET技术的有效途径。

本书注重理论与实践结合，每个ASP.NET控件开发技术知识点，基本上都会有对应的示例应用。在章节方面，按照技术点类别进行分类，内容比较全面，对每个控件开发技术点，都进行了详尽的阐述，因此本书也可以作为查找手册使用。

● 光盘使用说明

本书含两张光盘。第一张光盘内容与本书内容直接相关，由两大部分组成。

(1) “ControlExample”文件夹下面是本书所有的示例控件代码。对于每个示例控件，笔者特意做了两个方案（分别放在VS 2005和VS 2008对应文件夹内），以便使用不同VS版本的读者可以在自己的IDE环境下可以运行书中的代码。您可以根据自己当前的环境选择对应代码版本。

(2) “附赠 学习教程系列”文件夹中的是赠送给读者的.NET 3.5的3个学习教程：

- ASP.NET Ajax框架教程
- SilverLight入门教程
- JavaScript组件之jQuery (A~Z) 框架教程

本书的另一张光盘为微软正版Visual Studio 2008学习版DVD光盘，其中包含了Visual Basic 2008、Visual C# 2008、Visual Web Developer 2008，以及Visual C++ 2008的安装程序。在本书的第644页提供了Visual Studio 2008的注册说明。

章 名	主 要 内 容
第1章 服务器控件概述 及开发环境部署	本章介绍ASP.NET服务器控件定义以及服务器控件在软件开发过程中的作用;讲述服务器控件的类型和ASP.NET提供的控件开发的现有基类结构,理解.NET Framework为创建控件提供的一些常用基类;通过一个非常简单的示例,说明一个控件是怎么生成的,并讲解怎样调试,以及把它部署到Web Site中去运行;用示例演示控件生命周期。
第2章 服务器控件开发 中的调试技术	本章主要介绍开发ASP.NET服务器控件时所需的调试技术,内容包括:客户端脚本调试技术,服务端控件运行模式和设计模式下的各种调试方法。
第3章 从零开始开发 服务器控件	本章主要介绍开发ASP.NET服务器控件的过程,内容包括在ASP.NET环境下生成一个控件的过程,以及.NET Framework提供的一些开发控件帮助类;还有笔者在开发控件过程中积累的些许技巧和经验。
第4章 服务器控件属性	本章主要介绍属性的分类,以及设计时特性功能;各种复杂属性标记的实现方式,包括如何实现自定义的复杂属性;讲解系统常用编辑器及使用方法,自定义集合编辑器、自定义弹出式模态编辑器、自定义下拉编辑器、实现辅助计算功能的模态计算器编辑器;讲解系统常用类型转换器及使用方法,介绍两个自定义的类型转换器;讲解如何实现自定义的属性。
第5章 事件和数据 回发机制	本章主要介绍控件的事件和数据回发处理机制,内容包括:怎样定义控件事件;客户端触发事件及常用的几种形式;捕捉触发的事件;处理回发的数据及触发回发数据事件机制;组件控件事件的两种典型事件机制;最后用示例讲解通过控件对AjaxPro组件封装,实现自动完成控件。
第6章 页面状态机制	本章主要内容包括:页面状态存储原理;视图状态机制,控件状态机制及应用方案;视图状态和控件的关系;对页面状态进行加密以及页面状态的清理机制;动态增加控件使用视图时的注意事项;对页面状态的保存和装载应用高效率的类型转换器;页面状态的性能优化策略;视图状态和控件状态各自的优缺点及其适用场景。
第7章 定制自定义控件 主题样式	本章主要内容包括:.NET框架提供的Style类的工作原理以及用法;WebControl基类对样式的支持功能及工作原理;HtmlTextWriter类对控件样式的支持功能;定义控件属性样式时几种可遵守的规则;几个通用的CSS→Style的转换方法;使用Style属性定义简单复合控件样式方案,通过CSS样式定义控件样式方案,以及与ASP.NET主题的无缝集成。
第8章 控件客户端 编程模型	本章主要内容包括:ASP.NET服务器控件对客户端的支持;ClientScriptManager类功能详解;客户端回调原理及示例;使用JavaScript封装客户对象的参照控件示例;使用HTC组件技术封装客户对象的下拉数据选择控件;在线签名控件;常用JavaScript框架。

前言

章 名	主 要 内 容
第9章 定制用户控件	本章主要内容包括：用户控件功能概述；用户控件基类介绍；简单用户控件示例；通过复杂数据绑定用户控件示例讲解怎样在用户控件中利用基类Control支持动态创建控件，以及实现对五种数据源的支持；用户控件缓存及其使用方法，以及用户控件缓存与页面缓存集成使用时的规则。
第10章 开发验证控件	本章主要内容包括：ASP.NET内置控件介绍；验证控件可继承基类(接口)以及各自的应用场景；验证控件的工作原理；开发自定义验证控件的示例。
第11章 在控件中使用 嵌入资源	本章主要内容包括：控件嵌入资源的工作原理；在控件中应用设置和应用嵌入资源的步骤；通过示例讲解应用程序常用的各种类型资源文件的方法；几个浏览器常用工具介绍。
第12章 开发模板数据 绑定控件	本章主要内容包括：模板控件介绍；ASP.NET对模板属性控件的支持，包括ITemplate接口和模板容器支持功能介绍；Repeater模板控件解析；自定义模板数据绑定控件应用示例。
第13章 数据绑定和 数据源控件	本章主要内容包括：数据绑定控件概述；数据绑定控件几个基类解析；数据绑定控件应用示例；数据源控件概述；ASP.NET对数据源控件支持的接口和基类解析；数据源控件应用示例。
第14章 扩展控件方案 ——GridView	本章主要内容包括：控件概述；示例应用一，增加GridView实现多级HeaderRow功能；示例应用二，增加GridView对滚动条的支持，以及实现固定HeaderRow功能。
第15章 自定义控件设计 模式下编程汇总	本章主要内容包括：.NET框架对设计时行为的支持；设计时元数据属性支持；设计时的复杂属性序列化；类型转换器和类型编辑器总结；设计模式可用基类介绍；设计时呈现支持；自动套用格式；模板UI设计器；智能操作列表；控件设计器区域。
第16章 ASP.NET Ajax控件 开发——客户端	本章主要内容包括：ASP.NET Ajax框架概述；Ajax Library对客户端面向对象功能支持；Ajax Library中的常用类型介绍，以及Sys.Stringbuilder类介绍；Ajax Library请求服务和WebServices；Ajax Library对控件客户端类开发的支持；控件客户端类的创建用法，以及Ajax Library中新增的Sys.Debug调试类的简单用法。
第17章 ASP.NET Ajax控件 开发——服务端	本章主要内容包括：ASP.NET Ajax服务器控件概述；.NET 3.5框架中增加的一些重要Ajax服务器控件用法，以及扩展程序控件介绍；ASP.NET Ajax对服务器控件支持基类解析；最后通过两个具体应用分别说明ScriptControl和ExtenderControl这两种类型控件的创建方法。
附录A	本附录主要内容包括：对.NET提供的集合类进行优缺点和适用场景对比，ASP.NET Ajax框架教程；SilverLight技术教程；JQuery框架教程。限于篇幅，这三个教程在书中只列出了目录大纲，它们的具体内容和示例应用代码您可以在本书随书光盘中找到。

● 本书适合的读者对象

本书通过讲解ASP.NET控件技术来帮助开发人员提高自身ASP.NET开发水平，适合从事ASP.NET技术开发多年，想往深度理解ASP.NET工作机制的开发人员，考虑到对ASP.NET刚刚入门的初级开发人员，在本书前几章也介绍了几个帮助初级人员入门的应用例子。本书也可以作为在校学生的就业预备教材。

● 是什么原因让我有写这本书的意愿

开始我仅仅是把控件开发经验写在Blog上，后来发现这样写在Blog上比较琐碎，最大的缺点是不能保证知识点前后连贯性，也不能系统地由始到终全面地讲解一种技术。慢慢地随着开发经验的增多，渐渐就有了写这本书的打算。这本书对我来说也是对自己近几年工作的一个系统的总结，在写书的过程中自己能力又提高了许多。

ASP.NET技术可以说是技术的一次伟大变革，但也为开发人员带来一些隐患，因为它使程序员变得比较“傻”，Web服务器技术是一个比较复杂的技术，程序员如果一开始从事开发就遇到ASP.NET，就会认为开发Web程序像CS架构那么简单，导致程序员不仅不了解客户端技术，而且由于ASP.NET的易用性（为了实现易用性而把一些Web技术封装到底层）使程序员对服务器端技术也不甚了解。我对这个是比较愤慨的，这也是写这本书的另一个重要原因。我发现好多开发人员对ASP.NET技术的理解甚微，与我一起工作的甚至有很多年开发经验的朋友对一些ASP.NET基本性的原理还不了解，也可以说终究是因为ASP.NET技术的出现以及开发环境的集成性让很多开发人员变成“傻瓜式”的编程，使开发人员没有了机会去了解那些东西。

以ASP.NET为例，举几个开发人员常犯的基础性认识错误：

◎ 分不清楚控件的服务端和客户端ID

页面中有一个ID为TextBox1的服务端控件，很多开发人员在客户端访问该控件的DOM元素时，用如下语句：

```
var txtbox = document.getElementById("TextBox1");
```

之后有一天，他突然发现代码出问题了，通过以上代码再不能够检索到TextBox1控件的客户端元素了。这是因为对服务端控件来说这种写法是错误的，正确写法是：

```
var txtbox = document.getElementById('<%= TextBox1.ClientID %>');
```

出错原因是他没有理解服务端控件不像客户端控件只有一个ID。服务端控件有三个属性标示控件的唯一名称，这三个属性是：ID，UniqueID和ClientID。其中ID表示我们给它命名的ID，而不管在服务端还是客户端程序中都不会使用这个ID；UniqueID表示控件的服务端ID，在服务端标志控件的唯一名称；ClientID表示控件的客户端ID，在客户端标记页面中元素唯一名称。从使用角度讲，如果继承了INamingContainer接口，当我们为子控件设定一个ID后，它的UniqueID和ClientID会自动加上父控件名称和分隔符作为前缀。一般来说，在服务端分隔符默认使用“\$”进行分隔（UniqueID），但是到了客户端会自动将这些“\$”转换为下划线“_”（ClientID），即客户端ID和服务端ID名称是一样的，只是分隔符不同。

如果采用前面的错误写法，某天该程序员把TextBox1控件放到了母版页容器控件中，就会发生状况，

前言

在增加母版页之前，TextBox1控件由于没有父容器，所以它的ID，UniqueID以及ClientID都是“TextBox1”，只是巧合；而一旦TextBox1放到了容器控件中，它的UniqueID和ClientID都会发生改变，要增加父容器的名称作为前缀，比如它的ClientID可能变为Panel1_TextBox1，服务端UniqueID变为Panel1\$TextBox1。这里限于篇幅不再深入阐述，本书中会有详解。

◎ 滥用ASP.NET提供的系统级对象

由于一些开发人员对ViewState（还包括Cache，Session等）机制不了解，而导致其在开发过程中滥用。比如误认为ViewState是存储在客户端（表面看是在页面中）而不是服务端，而把ViewState当作一个海量容器来用，甚至把页面上的业务列表数据都缓存到ViewState中，最终导致页面流量庞大，打开页面速度非常慢却找不到原因。

学习一种技术与学习一种语言是有区别的。首先要了解ASP.NET各个对象的机制，简单地说ViewState的工作流程是这样的：每次提交时，它都会以客户端到服务端的形式来回传递一次，当处理完成后，最后会以处理后的新结果作为新的ViewState存储到页面中的隐藏字段，并与页面内容一起返回到客户端。

上面导致页面非常慢的原因是，开发人员不了解ASP.NET的页面呈现过程，误把业务级数据的序列化字符串存储到页面上的隐藏控件中，从而导致页面字节数变得庞大，使传输速度成为瓶颈。ViewState最喜欢存储数据量少的键/值串。有关页面状态（视图状态和控件状态）的保存在第6章有专门讲解。

还有朋友问我，在页面提交时要在服务器上运行一段JavaScript代码，而不是在客户端运行，应怎么做？或者有的朋友问，在两个服务端代码段中间执行一段JavaScript脚本该怎么做？

提出这些问题的最根本原因是他们没有了解清楚ASP.NET服务端控件生命周期，不知道客户端脚本是必须在服务端执行完成后才在客户端执行的。

使用面向对象思想构建软件可以带来很多好处，比如封装性、扩展性，等等，好处多多，这里就不一一列举。这里我要说的是面向对象给开发人员带来的隐患。很多开发人员进行面向对象开发时，都是在不了解基类的前提下进行派生并开发的，如果使用面向对象技术，不了解基类真的是非常危险的事情。这里的基类指的是系统基类、三方组件基类或自己开发的基类，其中系统基类是最应该了解，却也是最容易被忽略的。

比如我们在派生类中经常会使用一些类似base.方法名（参数）语句格式的方法，尤其是在控件开发过程中，基本上都是派生自系统控件基类开发自己控件的。比如一个简单的base.Render(e)方法的使用，我在开发过程中就遇到这些应用场景：有时候不调用该语句；有时候必须调用；有时候需要调用它但要屏蔽基类base.Render(e)方法体中的一些功能或修改部分功能；有时候不调用但要执行base.Render(e)方法中的部分代码语句，等等。要实现这么多场景功能，前提是必须要了解base.Render(e)方法，更确切地说应该是了解整个基类。也有许多开发人员在不知道基类中方法的功能时，通过例子去做猜测性的尝试，根据调用的结果来了解它的功能，但在很多时候这种方式得出的结论是不准确的。最好的办法是彻底地看一下基类的源代码。

基于此，本书的另一大特色就是在讲解每一种控件开发类型时，都会先对系统提供的可继承控件基成员（基类或基接口）作详细解析。这样不但可以让开发人员开发时能够感觉得心应手，更重要的是能够帮助他们开发出更准确更精确的控件。

庖丁解牛

● 运行环境介绍

本书所述的开发环境主要以微软最新框架.NET Framework 3.5为基础,整合VS 2008开发环境进行讲解,本书随书光盘所有示例均在IE 7.0浏览器全部验证通过。

.NET Framework 3.5完全兼容之前的版本,也就是说.NET Framework 3.5之前版本的控件类库是.NET Framework 3.5的子集,基于此在随书光盘中我专门用VS 2005开发环境做了一套示例(只是缺少.NTE 3.5新功能特性,如Ajax控件)。这样可以让使用VS 2005或VS 2008的朋友都能够打开随书光盘中的控件工程。在书中的许多技术点都以VS 2005和VS 2008这两个开发环境进行对比分析。

● 我的承诺

如果您之前没有从事ASP.NET控件研究过ASP.NET开发,看了这本书您对ASP.NET技术的理解会提高三倍,对于已从事开发一两年的开发人员,对于本书75%以上的内容可能还没有接触到。本书就像“庖丁解牛”一样剖析ASP.NET,当然前提是您要有耐心把本书读完。由于本书介绍的是一些ASP.NET高级编程技术,可能有些内容初学者理解起来比较困难,遇到这种情况千万不要泄气,只要坚持把书看完就能把前后技术连贯起来。

另外,书中讲到的每个例子在随书光盘中对应章节都可以找到,每看完一个例子要运行一次实际的代码,这些例子不仅仅是演示一下每个控件开发技术,更重要的是它提供了一些控件开发解决方案经验。

● 不是我自己在奋斗

首先我要感谢开心大哥和林思绮为本书所作的推荐序,以及各位专家们(胡德民、杜勇、朱晔、王涛、李会军、高阳、孟宪会、苏鹏、张博、叶帆、李洪根)一针见血的书评,你们的书评为本书增色不少。

其次,对提供技术分享的博友们表示感谢,或许是我们并不相识,但你们对知识毫无保留的分享是我进步的源泉。特别要感谢杜勇先生创办的开发者的网上家园——博客园,可堪称技术专家基地,不管是我在写作过程中还是在工作中,每当遇到困难时博客园的朋友们总是能及时提供解决方案。

在工作中,每天与我最亲密的就是如兄弟姐妹般的同事们,多谢你们对我技术上的指导以及对我生活中的关怀,多谢!

对本书的各位编辑李冰、许艳、李玲以及电子工业出版社所有相关工作人员,表示由衷的感谢,本书的出版离不开你们的指导和辛勤的工作,尤其是在本书写作的过程中,李冰始终的耐心支持与鼓励,以及许艳对书稿编辑的细心和一丝不苟,还有李玲对本书封面极富创意的设计,再次对你们三位表示特别感谢。

我还要感谢我的家人。寸草春晖,父母生我养我,您们的恩情难以回报,您们对我无微不至的关心与照顾,使得我能够心无旁骛,专心写作。感谢新婚妻子赵敬对我写作本书期间的全力支持,在我忙碌写书的时候,总是忽略了默默守候在身边的你,即使是在阳光明媚的秋日,我也无法陪伴你去南山郊游,但你没有丝毫怨言,没有你的理解和鼓励,就不可能有本书的出版。家人的关怀始终是如此温暖,我希望能将本书献给我亲爱的家人。

最后,感谢具有非凡人般毅力、又有坚强的性格,能够为追求自己的理想而努力拼搏、永不放弃的我。我的座右铭:对一件觉得有意义的事情,要设法用最快的时间完成它。

郑健

2008年12月1日

网友评论

在本书正式出版之前，作者曾将第2~6章放在网上作为试读样章，以下是网友们对样章的评论。

好。讲得比较细致，不知道是否每章都是这么细致。

姚宏

各章贯穿得很好，循序渐进。读完很有感受。非常感谢作者的分享！

007

这篇比较容易懂，对初级开发人员也比较适合。

开拓者

控件属性就被你讲得这么透彻，太多了，收藏一下以后慢慢看。

路瑶

对事件阐述得非常深度，建议大家都看一下。感觉文章中的内容不仅仅用于开发控件，开发ASP.NET级的页程序，这些也都是适用的。

hgood

每一篇都好长，讲得很仔细，太好了，正适合我，多谢！调试也这么多方法啊，牛人！感谢！

Good

全部看完了，讲得太精辟了！很少能够遇到这种文章。

kank

感觉很好。以前写过一些自定义的控件，也翻阅过很多书，都觉得很一般，今天看到这个觉得还不错。

杜耀辉

深入浅出，浅显易懂，是一本不错的参考书。

dreammaster

非常期待。书中70多个例子讲解，必定详细之至。到时候直接拿来用到项目中，楼主不要“打击盗版”噢。

yanggaolou

70个示例啊，我一定要买一本。只是目录和内容简介我就看了N遍，封面也是相当帅的一本书。

usea99

这本书对于我们刚从事ASP.NET的程序员来说真的是受益匪浅，希望能快点看到这本书。

czwk

我已经看了三章了，本书对ASP.NET机制分析得非常透彻，我每天都来看的。

好人

文中随便几个例子都是我存在的问题，有时间去买本书看看。

支持

目 录

第 1 章 服务器控件概述及开发环境部署	1	第 3 章 从零开始开发服务器控件	27
1.1 自定义服务器控件	1	3.1 选择基类	27
1.2 服务器控件在软件开发过程中的作用	1	3.2 控件呈现顺序	28
1.3 在控件开发中提升自己	2	3.3 Render 呈现控件的几种方式	30
1.4 可继承控件基类介绍	2	3.3.1 使用 HtmlTextWriter 类输出	30
1.5 运行一个简单的控件	7	3.3.2 直接输出 HTML 标签	32
1.5.1 开发一个简单控件	7	3.3.3 使用服务器控件的 RenderControl 方法	33
1.5.2 部署和运行	11	3.4 AddAttributesToRender 方法	34
1.6 控件生命周期	12	3.5 CreateChildControls 方法	35
1.6.1 概述	12	3.6 INamingContainer 接口	36
1.6.2 用代码跟踪控件生命周期过程	14	3.7 实现复合控件	38
1.7 本章总结	17	3.7.1 CreateChildControls 方法	39
第 2 章 服务器控件开发中的调试技术	18	3.7.2 ChildControlsCreated 属性	39
2.1 预置代码示例	18	3.7.3 EnsureChildControls 方法	39
2.2 调试页面服务端代码	19	3.7.4 RecreateChildControls 方法	39
2.3 服务器控件的调试	20	3.7.5 Controls 属性	40
2.3.1 运行模式下的调试方法	20	3.7.6 HasControls 方法	40
2.3.2 设计模式下的调试方法	20	3.7.7 HasEvents 方法	40
2.4 JavaScript 脚本调试方法	22	3.7.8 FindControl 方法	41
2.4.1 调试页面中的 JavaScript 脚本 (方法一)	22	3.7.9 实现复合控件示例	41
2.4.2 调试页面中的 JavaScript 脚本 (方法二)	23	3.8 常用开发技巧	51
2.4.3 调试非嵌入式 JavaScript 脚本文件	23	3.8.1 DesignMode 属性	51
2.4.4 调试嵌入式 JavaScript 脚本资源文件	24	3.8.2 屏蔽基类控件中的属性	52
2.4.5 高效率调试的技巧	24	3.8.3 Page.Request.Browser 属性	53
2.5 Ajax Library 对 JavaScript 调试支持	26	3.8.4 设置控件 ID 规范	53
2.6 本章总结	26	3.8.5 增强 FindControl 功能	54
		3.8.6 映射服务端控件值到客户端	55
		3.8.7 禁止派生自 CompositeControl 的控件创建子控件	56
		3.8.8 使用 CreateChildControls 的注意事项	56
		3.8.9 不要误解设计元属性 DefaultValue	57

3.8.10 在 Render 方法中利用基类资源	57	5.4 本章总结	193
3.8.11 条件编译&条件属性	60	第 6 章 页面状态机制	194
3.9 本章总结	61	6.1 页面状态概述	194
第 4 章 服务器控件属性	62	6.2 视图状态机制	195
4.1 控件属性的作用	62	6.2.1 IStateManager 接口	195
4.1.1 系统属性	62	6.2.2 控件生命周期中的装载和保存 视图阶段	196
4.1.2 自定义属性	62	6.2.3 简单类型视图状态应用	197
4.2 简单属性	63	6.2.4 实现自定义类型视图状态	205
4.3 属性的设计时特性	64	6.3 控件状态机制	217
4.4 复杂属性	67	6.4 视图状态和控件状态的关系	221
4.4.1 概述	67	6.4.1 在禁用视图状态的情况下仍然 使用 ViewState 对象	221
4.4.2 复杂属性的几种标记形式	67	6.4.2 IStateManager 接口仍然可以在 控件状态中使用	223
4.4.3 深入研究——复杂属性分析器	79	6.4.3 视图状态和控件状态组合使用规则	225
4.5 深入研究——定制自己的属性 编辑器	85	6.5 加密页面状态	225
4.5.1 系统属性编辑器	85	6.6 清除页面状态	226
4.5.2 定制属性编辑器	91	6.7 对动态添加控件的视图状态分析	228
4.6 类型转换器	126	6.8 自定义类型转换器实现高效率 序列化	231
4.6.1 系统类型转换器	126	6.9 页面状态性能优化策略	238
4.6.2 定制自己的类型转换器	128	6.9.1 存储位置优化——把视图状态信息 保存在服务端而非客户端	238
4.7 实现自定义属性	138	6.9.2 体积优化——压缩视图状态数据	240
4.8 本章总结	145	6.9.3 分块存储视图状态数据	243
第 5 章 事件和数据回发机制	146	6.10 视图状态和控件状态的总结	243
5.1 控件事件和数据回发概述	146	6.11 本章总结	244
5.1.1 事件及其意义	146	第 7 章 定制自定义控件主题样式	245
5.1.2 数据回发机制	147	7.1 控件样式概述	245
5.2 事件和数据回发机制的实现	148	7.2 ASP.NET 对样式的支持	245
5.2.1 客户端回传事件接口 IPostBackEventHandler	148	7.2.1 Style 类的工作原理及使用	245
5.2.2 客户端回发/回调解密	150	7.2.2 WebControl 基类对样式的支持功能 及其工作原理	249
5.2.3 回传数据处理接口 IPostBackDataHandler	153	7.2.3 HtmlTextWriter 类对控件样式 的支持	252
5.2.4 正确处理继承基类中控件的事件	159	7.2.4 定义控件内部的样式属性规则	253
5.2.5 扩展 KingTextBox 控件功能 (King TextBoxCanPostBackEvent 控件)	161	7.2.5 两个常用转换方法	256
5.3 复合控件的事件处理机制	168	7.3 应用示例	260
5.3.1 高效率事件集合对象	168	7.3.1 复合控件样式及视图状态	260
5.3.2 定制自己的委托和事件参数类	170		
5.3.3 复合控件的事件处理	175		
5.3.4 自动完成功能控件 SearchControlIntelligent	185		

7.3.2 控件结构与样式分离方案	268	10.1.2 验证控件的工作机制解析	423
7.4 本章总结	295	10.2 实现自定义验证控件	443
第8章 控件客户端编程模型	297	10.3 本章总结	451
8.1 ASP.NET 服务端控件对客户端的支持	297	第11章 在控件中使用嵌入资源	452
8.2 ClientScriptManager 类功能详解	301	11.1 在控件中嵌入资源的原理	452
8.2.1 RegisterArrayDeclaration 方法	301	11.2 在控件中应用嵌入资源	453
8.2.2 RegisterClientScriptBlock 方法	302	11.2.1 在控件中使用嵌入资源	453
8.2.3 RegisterClientScriptInclude 方法	303	11.2.2 常用嵌入资源类型及用法	455
8.2.4 RegisterClientScriptResource 方法	304	11.3 浏览器常用工具	460
8.2.5 RegisterStartupScript 方法	305	11.3.1 HttpWatch	460
8.2.6 RegisterExpandAttribute 方法	305	11.3.2 IE Developer Toolbar	461
8.2.7 RegisterHiddenField 方法	306	11.3.3 Fiddler	461
8.2.8 GetCallbackEventReference 方法	307	11.4 本章总结	462
8.2.9 GetPostBackClientHyperlink 方法	307	第12章 开发模板数据绑定控件	463
8.2.10 GetPostBackEventReference 方法	308	12.1 ASP.NET 模板属性控件机制	463
8.2.11 GetWebResourceUrl 方法	308	12.1.1 模板属性介绍	463
8.2.12 RegisterOnSubmitStatement 方法	309	12.1.2 ASP.NET 对模板控件的支持	464
8.3 客户端回调	309	12.1.3 Repeater 控件模板实现机制解析	466
8.4 客户端控件应用示例	323	12.2 自定义控件应用示例	469
8.4.1 封装 JS 客户端对象	323	12.2.1 模板控件功能	469
8.4.2 封装 HTC 客户端对象	343	12.2.2 应用示例	470
8.4.3 在线签名控件示例应用	371	12.3 本章总结	489
8.5 常用的一些 JavaScript 框架	385	第13章 数据绑定和数据源控件	490
8.6 本章总结	387	13.1 数据绑定控件	490
第9章 定制用户控件	388	13.1.1 数据绑定控件概述	490
9.1 用户控件概述	388	13.1.2 ASP.NET 对数据绑定控件的支持	491
9.1.1 用户控件的作用	388	13.2 数据绑定控件示例应用	501
9.1.2 了解 System.Web.UI.UserControl 基类	388	13.2.1 控件实现功能	501
9.2 用户控件应用示例	389	13.2.2 应用示例	502
9.2.1 简单用户控件	389	13.3 数据源控件	508
9.2.2 复杂数据绑定用户控件	394	13.3.1 数据源控件概述	508
9.3 用户控件缓存	419	13.3.2 ASP.NET 对数据源控件的支持	508
9.3.1 ASP.NET 缓存介绍	419	13.4 数据源控件示例应用	515
9.3.2 用户控件缓存	420	13.4.1 控件实现功能	515
9.4 本章总结	421	13.4.2 应用示例	516
第10章 开发验证控件	422	13.5 本章总结	522
10.1 验证控件的工作原理	422	第14章 扩展控件方案——GridView	523
10.1.1 内置验证控件简述	422	14.1 扩展控件概述	523
		14.2 任意定制 GridView 多表头	523

14.2.1 功能概述	523	16.2.3 类类型以及面向对象继承特征	569
14.2.2 代码实现	524	16.2.4 枚举类型	570
14.3 固定 GridView 表头并增加滚动条支持	527	16.2.5 反射功能	570
14.3.1 功能概述	527	16.3 Ajax Library 中的常用类型	571
14.3.2 代码实现	528	16.3.1 Ajax Library 对 JavaScript 基类型的扩展	571
14.4 本章总结	533	16.3.2 Sys.StringBuilder 类	576
第 15 章 自定义控件设计模式下编程汇总	534	16.4 Ajax Library 请求服务端和 WebServices	577
15.1 .NET 框架对设计时支持	534	16.4.1 Sys.Net 命名空间中的类	577
15.2 设计时元数据支持	535	16.4.2 请求过程解析	578
15.3 复杂属性序列化	539	16.4.3 WebRequest 请求示例	579
15.4 类型转换器	539	16.4.4 使用代理实现 Ajax 调用服务端示例	581
15.5 属性编辑器	540	16.5 Ajax Library 对控件客户端类开发的支持	584
15.6 设计模式可用基类介绍	541	16.5.1 客户端控件基成员(类/接口)	584
15.7 设计时对控件呈现的支持	543	16.5.2 ASP.NET Ajax 控件客户端生命周期	589
15.7.1 概述	543	16.6 客户端控件示例应用	592
15.7.2 应用示例	543	16.7 调试和跟踪	603
15.8 自动套用格式	546	16.8 本章总结	605
15.8.1 概述	546	第 17 章 ASP.NET Ajax 控件开发——服务端	606
15.8.2 应用示例	547	17.1 ASP.NET Ajax 服务器控件概述	606
15.9 模板 UI 设计器	551	17.2 Ajax 服务器控件	607
15.10 智能操作列表	552	17.2.1 Ajax 扩展控件	607
15.10.1 概述	552	17.2.2 ASP.NET Ajax 扩展程序控件	612
15.10.2 应用示例	552	17.3 ASP.NET Ajax 对服务器控件支持的基类	612
15.11 控件设计器区域	558	17.3.1 Ajax 控件支持的基类	612
15.11.1 概述	558	17.3.2 Ajax 控件应用示例	623
15.11.2 应用示例	558	17.4 本章总结	636
15.12 本章总结	562	附录 A .NET 3.5 技术教程系列——Collection/ASP.NET Ajax/SilverLight/JQuery	637
第 16 章 ASP.NET Ajax 控件开发——客户端	563		
16.1 ASP.NET Ajax 框架概述	563		
16.2 Ajax Library 对客户端面向对象功能支持	564		
16.2.1 命名空间及 Type 和 Function 类	565		
16.2.2 接口类型	568		