

大师讲堂

ASP.NET 3.5

AJAX 开发范例 精讲精析

[服务器篇·VB版]

两岸三地公认最优秀的程序设计图书作家章立民先生，
带领最专业的技术团队，全面剖析ASP.NET 3.5的技术精粹
以实例的形式教您开发专业级的Web应用系统

中国台湾微软开发工具产品营销经理 **胡德氏** 专业推荐！

本书完整范例程序及相关数据库文件
章立民老师亲自录制的127分钟高清视频技术教程
——书本之外的增值部分，同样精彩



章立民（中国台湾） 编著
六度获选微软最有价值专家（MVP）

Microsoft® 微软专家 热点技术书库
Microsoft® 微软技术专家 章立民 作品系列

大师讲堂

ASP.NET 3.5

AJAX 开发范例 精讲精析

[服务器篇·VB版]

章立民（中国台湾） 编著

六度获选的微软最有价值专家（MVP）

内 容 提 要

本书是两岸三地公认最优秀的 IT 图书作家章立民先生的最新力作,全面剖析了 ASP.NET 3.5 AJAX 在 Web 服务器开发中的应用。

全书共 7 章。第 1~2 章对 AJAX 基础知识和 ASP.NET AJAX 技术进行了概述。第 3~7 章结合嵌套购物车、书籍订购界面、动态调整照片显示比例、个性化的网页相簿、异步平行输入输出界面、动态绘制的 2D 和 3D 统计图表等大量实例,深度探讨了 UpdatePanel、ScriptManager、UpdateProgress 和 Timer 等 ASP.NET 3.5 常用服务器控件的使用技巧,并完整阐述了 PageRequestManager 类的异步回发技巧。通过学习,读者可以树立正确而扎实的异步网页开发理念,能够编写出专业级别的 Web 应用系统。

本书实例丰富,好学实用,不仅适合初学 ASP.NET AJAX 技术的开发人员,对具有一定 ASP.NET 网络应用程序开发经验的开发人员,也会有有益的帮助。

图书在版编目(CIP)数据

ASP.NET 3.5 AJAX 开发范例精讲精析. 服务器篇: VB 版
/章立民编著. — 北京: 科学出版社, 2009
(大师讲堂)

ISBN 978-7-03-023834-4

I. A… II. 章… III. 主页制作—程序设计
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2008) 第 211502 号

责任编辑: 刘秀青 / 责任校对: 周 勤

责任印制: 科 海 / 封面设计: 林 陶

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京艺辉印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2009 年 2 月 第 一 版

开本: 16 开

2009 年 2 月 第一次印刷

印张: 23.00

印数: 0 001~4 000

字数: 631 千字

定价: 46.00 元 (含 1CD 价格)

(如有印装质量问题, 我社负责调换)

推荐序

章公立民最痛恨别人总是说：“我都是看你的书长大的”。首次为章公作序，当然也让我回想起十几年前所浸淫的 FoxPro 系列书籍——这十余年来章公在程序开发领域用心用力的笔耕，也堪称是著作等身。章公的书有个不易的特色，就是范例特多，每完成一本书就像是写了个大系统。没有其他哗众取宠的噱头，就是好学实用。

这次章公的大作着力在 ASP.NET 3.5 AJAX，这是微软 2008 年最新的网页开发技术。下一代 Web (NextWeb) 的应用潮流已经随着 Web 2.0 的商业化方向而更趋蓬勃发展，网页设计人员与开发人员无不花费心思在网站界面上争奇斗艳，提供崭新的用户体验 (User Experience)，此时 AJAX 与 CSS 两种互相关联的网页开发技术成为主流。然而，纯粹的 JavaScript 是难以操作、编写与维护的。Microsoft Visual Studio 2008 搭配 .NET Framework 3.5，增加了许多互动的功能，让 AJAX 网页的开发就像是 ASP.NET 的开发经验一般，更首度提供了 JavaScript 的调试功能，让程序开发人员事半功倍。

许多朋友会问，ASP.NET AJAX 会不会只是过渡技术，而终将完全被 Silverlight 所取代？在目前可以预见的未来数年，ASP.NET AJAX 在微软的开发技术蓝图上还是会占有一席之地的。RIA 类型的网站技术，无可避免地必须在浏览器上下载安装一个插件 (Runtime)，才能呈现影音或高互动的用户界面。所以，两者间的市场定位区别是清楚的：如果要追求 HD 高画质影音的呈现、动画形态的互动界面与创意的设计空间，请采用 Silverlight 技术开发。但 ASP.NET AJAX 的特点也非常明显，无论是在运行速度还是在浏览器相容性，都有较 RIA 技术更为杰出的表现，但又可以获得较以往 ASP 开发技术更多的互动式设计灵活，不失为程序员迈向“设计”领域的一个好跳板。

因此，如果您对于 ASP.NET 的开发技术早已熟悉，延伸到 AJAX 领域一定是最直接的选择，这也是我们的建议。相信本书一定会对您大有助益，初学者可以从第 1 章开始了解来龙去脉，而有基础的朋友也可以直接切入范例，快速了解 AJAX 技术的优势。章公有十余年的声誉，绝非浪得虚名，祝福您也会像小时候的我一样，有个愉快的学习体验！

台湾微软开发工具暨平台推广处

产品行销经理

胡德民

2008 年 12 月

笑谈我的学习之路

读者在博客的留言板上留言，希望我能够分享 .NET 的学习过程、心态以及方法。
于是写下本文，与大家分享。

在信息软件界讨生活，要放轻松是谈何容易。说“笑谈”是有点故作轻松了，“心路历程”听起来又有点恶心，讲到“传承”吗，那更是不敢当，总之，大家就是放轻松，当作生活经验的交流。将我近 20 年来的经历与心得，毫不保留但简短地跟大家分享。更重要的是，我希望大家在读完本文后，都能够更清楚以及坚定地选择你未来所要走的路。

技术快速演进的不变定律

纵览人类文明的演进，大抵都建立在“今日会比昨日好、明日要比今日强”的基本史观之上。回顾过往的信息技术洪流，20 世纪 80 年代末还在钻研 DOS 系统、汇编语言，90 年代初却得踏进 Windows 3.1、Windows SDK、DDE 等新架构所延伸的密林，大气还没喘上一口，却又在 90 年代中期碰上了 32 位 Windows、MFC 与 OWL/VCL、OLE2 所构成的技术风暴，而 Web 也在此时开始火热，主从式架构也快速增加为三层式、甚至多层式的 Windows DNA，正当庆幸千禧年的 Windows 架构还在掌控之中，.NET 技术的跃进又像一波波不断来袭的大浪。

身处速度快、变化大的信息技术洪流，的确必须随时准备好面对更快、更好、更强的明日世界。但是，技术的本质是什么？学习的瓶颈该如何突破？如果只是一味盲目乱冲，而不适时让自己缓下来思考，很容易被洪流吞噬。面对技术洪流，你的生存之道是什么？你是否也思考了技术的本质以及学习的方法？更重要的是，你真的准备好面对这永无止尽的磨练（或者说折磨）吗？

你为什么进这一行

正所谓男怕入错行，女怕嫁错郎（我必须说，这观念其实是迂腐的）。所以在分享所谓的经验之前，应该先回到源头，问问自己，为什么入这一行。在两岸，常常会有一些年轻的程序员感到彷徨而来求助于我。我通常第一个问题就是问对方，你为什么进这一行。是因为你在学校原本就读信息相关的专业而自然进了这一行，还是当时自己或家人觉得电子与信息相关产业非常热门，因此进了这一行。如果你是因为觉得产业热门才进了这一行，而且是在 1999 年之后才入行的，那我只能送你一句

话,那就是“夕阳无限好,只是近黄昏”。所以,如果你在午夜梦回的时候常常会对自己的工作有所质疑,那真的应该好好思考思考。

我所要强调的是,你对信息软件真的有兴趣吗?你真的喜欢这样的工作吗?你顶得住压力吗?更重要的是,你快乐吗?能够把一件事情做好,并不表示你真的喜欢,也不表示你真的快乐,这只代表你有足够的能力。如果这真的是你的兴趣,你真的热爱,就算辛苦,也会乐在其中。

身处这一行,工作日与假日加班是常态。我常常看到许多年轻人,不过就是假日加个班,就觉得自己倍感辛苦,备受委屈。如果你也常有这种心态,赶快转行吧!更可怕的是,自己辛苦也就算了,家人与亲密爱人常常无法理解与谅解,同样地,他们的抱怨也成了常态。干这行,还有许多后遗症,比如身体变形,秃头,满头白发,心情很难放松,等等。

我谈这些,并不是要你离开这个行业,我的目的是希望大家再次肯定自己是喜欢这个行业的,不论你是开发人员或者 IT Pro,都能够更加清楚自己的选择,然后坚定地走下去。

技术路线的选择

在你确定自己适合走技术这条路,也决定要走技术这条路之后,接下来面临的的就是技术路线的选择,而这通常也是学生或初入行者最大的抉择。为什么呢?因为这关系到他未来的就业,他希望自己押对宝。然而为什么会难以抉择呢?因为目前看不出何者拥有绝对的优势。据说是 .NET 小赢。但是我们必须了解,任何一种技术架构都有其特性与优劣所在,然而难就难在 .NET 与 J2EE 都非常庞大,我们很难两者都精通。因此我说成功没有一定的规则可言,失败却有,那就是“两面讨好者”。

我常常看到有许多人喜欢比较 .NET 与 J2EE。说句实在话,这两种技术路线都有极深的内涵,单单看技术架构,稍微用点其中的技术进行开发,就足以评断两者的优劣吗?这未免太自以为是,大放厥词。我投注了极大的心力研究 .NET,才认为能够稍微随心所欲地掌握其内涵,也才能够真正体会其中的优点与缺点。我相信 J2EE 也是一样的道理。问题是,我时间有限,精力有限,能力有限,不可能同时研究这两者,既然如此,何德何能去评论呢?

我要说的是,评论诸项技术时,最忌讳的就是“成见”以及那股莫名的“情绪”,或者说“情结”。当你有了这样的心态,如何以客观的心情去评量各种技术呢?奉劝大家,在选择技术的时候,不需要有太多的情绪与爱恨情愁。我知道有很多人就是看微软不爽,看微软赚那么多钱就觉得很讨厌,因为我讨厌微软,所以我不用微软的技术,所以我选择 J2EE、Java、Oracle。微软的技术真的不好吗?同样地,你也不需要因为非常喜欢微软,所以就讨厌 J2EE、Java、Oracle。身为一位技术人员,一定要抛开这些成见与情绪,因为一旦有这些成见,就会影响你的判断力,也因此很可能与好的技术失之交臂。所以我的态度是,不论技术是谁推出的,一定要用开放与正面的态度去看待。正所谓“知己知彼,百战百胜”,不论是自我学习,还是在市场上竞争,切忌闭门造车。

其实只要你真正投入,掌握技术的精要,自然能彻底发挥其特性,进而开发出优质的应用程序。如果你学东西总是半吊子,那即使你用的是伟大的技术,写出来的东西也是平淡无奇,甚至上不了台面。

我选择技术的原则非常简单，第一看技术的进入障碍；第二看技术有没有未来性。所谓进入障碍，当然就是跟我目前的技术是不是有延续性，是不是能够有良好的衔接，这是为了确保我过去投入的心血不会轻易付诸流水。就未来性而言，就是看技术有没有前景，说得露骨点，应该说有没有“钱”景，毕竟技术必须叫好，也要叫座，才是好技术，那种孤芳自赏者，通常没有好下场。没错，这两个原则有时候是彼此相冲突的。放眼技术的演进史，有时候会因为某种革命性架构与装置的推出，导致引发大跃进式的演进。在这种时候，未来性的考虑当然更胜于进入障碍。

其实在许多时候，技术（或者说技术路线）的选择也没有那么难，因为市场可能已经帮你做了决定。如果真的无法自行判断，就跟着潮流走吧！

不需要跟技术“谈恋爱”

别让“执着”害了你。就好比我们劝人在投资理财的时候不要跟股票“谈恋爱”一样，你也不需要跟技术“谈恋爱”。没错，你当初看好某一家公司的前景，买了该公司的股票准备长期投资，可是当你发现公司营运状况不如预期或者出现更大的危机时，难道还要抱着股票不放吗？你希望你的股票变成废纸吗？没错，你现在抛售股票已经受伤，已经赔钱，可是如果不当机立断、壮士断腕，损失将会更惨重。投资理财如此，技术的选择也是如此。

拿得起，也要放得下。当发现所选择的技术已经失去优势，应该毫不犹豫地转换。没错，你已经投入了许多的时间和精力学习该项技术，要放弃，谈何容易。不过 IT 这个行业就是如此，过去被奉为圣经、归镍的技术架构可能在短短的时间就被淘汰出局。所以我们说，你也必须保持一个开放的态度，随时注意趋势的变化，不要恃才而骄，自以为了不起。就好像戏水溺毙的人多半是游泳高手一样，因为他自认功夫过人一等，不会有任何危险，完全没有任何的警戒心，因此反而比别人死得更快。

我为什么要特别提到“不要跟技术‘谈恋爱’”，因为我之所以会在 1999 年急转直下，陷入低潮，就是犯了这个错误。在很久之前，我靠 FoxPro 成名，我的存款、房子、车子都是靠 FoxPro 这只“狐狸”赚来的。可是当时已经有一些明显的迹象显示 FoxPro 未来将不会是微软的发展重点。这些迹象包括：FoxPro 在美日两大市场卖得非常不好，Tech-ED 的 FoxPro 场次人数递减等。甚至连微软内部的高层人员都要我小心，可是我却不信邪。最后的结果就是，承受痛苦。

我是如何学习的

正所谓万丈高楼平地起，要把功夫练好，马步一定要蹲得稳。我想很多人最好奇的，就是我如何学习新技术。说真的，我用的都是笨方法。学习技术，一定要先把架构与原理彻底了解，然后才能晋升到技巧的活用。而这中间的窍门，就是不停且大量的实践，并将结论有组织地整理起来。

每当我阅读一段原理与说明时，都会尽可能实际动手操作一下或编写程序验证，一方面可以确认自己的理解是正确的，一方面也可以更加熟悉并体会其内涵，了解其应用层面。当然，人的记忆力是有限的，最终的验证结果与结论必须要进行系统地整理，以方便未来能够快速查阅。

这中间的过程不仅枯燥,更是乏味,而且很寂寞,然而却是必经的过程。就我的了解,业界的开发人员,可能最缺乏的就是这一段学习过程,毕竟大家都急于对老板与客户负责,交差结题,因此常常会流于追逐实用的技巧。但是大家别忘了,彻底了解技术架构,可能会比追逐新工具来得更长久,毕竟许多原理都是相通的。就.NET而言,从2002年推出到目前的.NET 3.5,基本的大架构并没有什么不同,只是一直在既有的基础上累加并扩展,如果你当初就能够秉持正确的学习原则,或许现在会感觉更加如鱼得水。

如何让自己与众不同

如果要在这个产业长久发展,你必须让自己与众不同,而不同之处不仅仅是技术。

我很喜欢问开发人员一个问题:“除了高人一等的技术能力之外,你还有什么?”我常跟别人说,你再会写程序,你也只是个程序员。这句话听起来或许很伤人。不过就像过去很多人问我写书要写到几岁一样,我以前从来不觉得这是个问题,但是当年纪大了,我开始思考这个问题。因此我现在问你,你要写程序写到几岁呢?难道你要一辈子写程序吗?你以为每一位会技术的人都可以像Bill Gates一样吗?你以为Bill Gates就只是会技术吗?他没有其他的特质吗?

也不是说你不能一辈子写程序,不能一辈子当技术人员。因为有些人真的不适合当主管,不适合开公司做生意。如果真的想要长久地做技术研究,你就需要一个比较稳定的环境。可能是在学校或研究单位,或者大型的私人企业。大家要早点想这个问题,毕竟中年转业不容易啊。

我也常告诫人,不要让高人一等的技术能力成为反动力。正所谓“工作的金字塔,一人抵三人。”你技术能力强,以一敌三,老板为什么要给你升职,因为要用三个人补你的缺,而且你看起来真的不像个主管。为什么升任主管的常常是你认为技术不太行的,只会“耍嘴皮子”的。结果是,你花了很长的时间与很大的努力来证明自己,可是等你年纪大了,公司可能觉得你太老,不想给你机会。唉!真是残酷啊!

或许你可以思考如何进一步融入“热情”、“人文素养”与非技术的“专业素养”,或许这样可以让你更发光发热。我在2007年之前的好几年都不讲研讨会,因为我不觉得我可以讲得比以前更好,我觉得难以再超越过去的成绩,我有疲惫感,也就是说我没有热情。与其勉强上台,还不如不上台。假设我自己都不热爱这样的工作,如何让听众去热爱呢。不要忘记,任何人比你都敏感。千万不要以为可以浑水摸鱼,蒙混过关。

我还想问你,你的特质是什么。为什么同一首歌,由不同的人演唱,会有完全不同的效果。因为他捉住了其中的感觉,能做最佳的诠释。反过来说,量身订制,因此你应该寻找最适合你的表现方式。

千万记得,在演讲、讲课或做简报时,是不可以忘我的,要随时与台下观众保持互动。所谓的互动不见得是言语的互动,而是通过你的眼神、话语和肢体语言来传达信息,而从听众的表现、坐姿等,都可以看出他们的反应,并随时动态调整。因此我较不喜欢列出太细的项目,保持动态调整的弹性,而且准备的内容应该要比实际的会议时间来得长和丰富。

最后建议科技人应当走出技术的象牙塔,培养良好的人际关系,展现不凡的热情与活力。

抓住机会，秀出自己

技术人员一般来说都比较内向，比较沉默，不喜欢表现。可是我真的鼓励大家，抓住机会，秀出自己（Present yourself）。当初我第一次到微软并且帮微软讲研讨会，根本就是毛遂自荐的。重点是，当别人给我机会时，我充分抓住了。

成功是性格、努力与机会的结合。我们常说性格决定命运，努力要有方法和策略，最后还需要天时、地利以及人和。你说路遥知马力，日久见人心，问题是有机会有表现时，你闷不吭声，好像不存在。其结果是，人家还以为你能力不够。

当然，如果你是金玉其外，败絮其中，也撑不了太久。但是如果你真的有能力，给你一个良心的建议：“不要只做沉默的工蜂”。毕竟机会稍纵即逝，逝者不可追。

在这里给大家提供一组真实的统计数据，当你说话的时候，人们接收到的信息是什么？请注意，有 93% 来自肢体语言和语调。

- 55%：外型与肢体语言。
- 38%：语调。
- 7%：话语（内容）。

为什么我要特别强调上述数据，因为我听很多技术人员讲话的时候，真的很想扁人。不是说话的音调没有高低起伏，就是喃喃自语不知所云，不然就是自顾自地讲，完全沉醉在自己的话语和情绪中。唉！真是要昏倒。

另外还有一组重要的数据提供给大家。一场演讲后，人们还记得什么：

- 过了 1 小时，人们会忘掉 50% 的内容。
- 过了 1 天，人们会忘掉 80% 的内容。
- 过了 1 周，人们会忘掉 95% 的内容，剩下的记忆是：故事、例子、亲身经历。

这也就是为什么我们即使谈的是技术性的研讨会，也要举许多的实例与成功案例，因为这样印象才会深刻。

反省的能力

策略一定要动态调整。人们往往会活过去的成功模式中，然而今非昔比，策略应时时调整。

千金难买早知道，千金难买回头望。也许我们无法抓住机会，但是藉由不停地反省与修正，也能让自己朝对的方向前进。

健康是最大的财富

大家都听过这句话吧：“健康的身体是1，没有这个1，后面的0将毫无意义”。美国前总统里根曾经说过：“It is true hard work never killed anyone, but I figure, why take the chance?”（努力工作确实不会致人死命，但我想，何必冒险？）

当然，我们之前一直提到要努力，加班也不要抱怨，问题是没人要你熬夜与晚起，你就不能早睡早起吗？时间一样长得很，更重要的是，这样对身体有益。很多东西在失去了之后还可以找回，但是健康不能。刻骨铭心的恋情可能会永远占驻你心中的角落，不时还会隐隐作痛，问题是它不会夺去你的生命；错过的电影，还有下一场，健康可能没有；这个男生不爱你，没关系，下一个男人会更好，只要你有健康的身体。

过去六七年来，我一直保持运动的习惯，饮食也是清淡得不行，有人说我过得是教徒的生活。运动不仅让身体更舒畅，也让大脑能够休息与清醒，为下一次的吸收做好准备。建议大家也尝试看看吧！

结论

人在好的时候要想到不好的时候，人在不好的时候不要去想好的时候，而是应该静下心来思考方法与策略，然后迅速脱离谷底。当然，我自己对未来也有很多期许，希望能再创高峰。

有人说：“发现一瓶好水，令人心旷神怡；发现一位好作者，令人眉开眼笑。”这么多年来，我写书一向注重质量，期许自己能够成为让人眉开眼笑的好作者。过去如此，现在如此，未来也将如此。近20年来，经过起起伏伏，风风雨雨，今天还站在这里，我是心存感激的。下次再见，谢谢大家。

章立民

2008.6.23

目录

Chapter 01	万事开头难，从了解 AJAX 开始学起	1
1-1	AJAX 的用户观点	2
1-2	AJAX 的开发人员观点	6
1-3	使用 AJAX 的思考	10
1-4	AJAX 应用现状	15
	结语	18
Chapter 02	迈向解密之路，ASP.NET 3.5 AJAX 技术综述	19
2-1	什么是 ASP.NET 3.5 AJAX	20
2-2	探讨 ASP.NET 3.5 AJAX 的架构	21
2-3	探讨 ASP.NET 3.5 AJAX 服务器端功能	21
2-4	探讨 ASP.NET 3.5 AJAX 的客户端功能	24
2-5	关于 AJAX Control Toolkit	25
	结语	26
Chapter 03	实战的第一步，使用 UpdatePanel 控件异步局部更新页面	27
3-1	如何异步局部更新页面	28
3-1-1	整页更新 vs 异步局部更新	28
3-1-2	实现异步局部更新	32
	页面范例1 异步更新的日期选择界面	37
	页面范例2 异步更新的日期选择界面	40
	页面范例3 异步的步进式界面	40
	页面范例4 异步的产品列表查看界面	42
	页面范例5 异步的订货金额求和查看界面	45
	页面范例6 异步的2D饼图查看界面（ObjectDataSource+LINQ+GridView+.ashx）	48
	页面范例7 异步的3D饼图查看界面（ObjectDataSource + LINQ +GridView + .ashx）	53

页面范例8	异步地添加、修改与删除数据界面 (DetailsView + LinqDataSource)	58
页面范例9	异步地导航、添加、修改与删除数据界面 (ListView + LinqDataSource)	64
页面范例10	异步分组数据查看界面 (LinqDataSource分组实现+嵌套ListView)	66
页面范例11	异步的分组数据查看界面 (LinqDataSource分组实现+将GridView嵌套在 ListView当中)	72
页面范例12	异步的可折叠式分组数据查看界面 (JavaScript + LinqDataSource分组实现+ 嵌套ListView控件)	74
3-2	探讨 UpdatePanel 控件的运作方式	82
3-3	探讨 UpdatePanel 控件的使用模式	83
页面范例	示范如何在模板中使用UpdatePanel控件	83
3-4	探讨 UpdatePanel 控件内容的更新时机	85
页面范例	异步的简易购物车界面 (ListView + ListView)	88
3-5	探讨 UpdatePanel 控件的触发器 (Triggers)	91
页面范例1	示范定义UpdatePanel控件的Triggers属性	95
页面范例2	示范定义UpdatePanel控件的Triggers属性	97
页面范例3	异步男女平均工资2D饼图查看界面	98
页面范例4	异步男女平均工资3D饼图查看界面	100
页面范例5	异步的一对多界面	101
页面范例6	异步的书籍目录查看界面 (ListView + ListView)	103
页面范例7	异步的产品数据搜索界面	107
页面范例8	异步的自行车产品查看界面	110
页面范例9	示范如何协调两个UpdatePanel控件之间的互动	116
页面范例10	异步的格式化与数据列求和 (加总) 界面	120
页面范例11	异步地添加、修改与删除数据界面 (含删除确认功能)	122
页面范例12	异步地添加、修改与删除数据界面 (含删除确认功能)	125
页面范例13	结合GridView与DetailsView的异步添加、修改与删除数据界面 (含删除确认功能)	125
页面范例14	示范如何更新其他UpdatePanel控件的内容而不更新本身的内容——非常重要!	126
页面范例15	示范何时需要将UpdatePanel的触发器设置成由其内部的控件事件来引发 (ListView + ListView)	130
3-6	何时需要使用触发器 PostBackTrigger	132
页面范例1	示范如何在UpdatePanel中使用FileUpload控件	134
页面范例2	搭配使用FileUpload与UpdatePanel控件来制作具备照片查看和编辑功能的添加、修改与 删除数据界面 (ADO.NET或LINQ)	135
3-7	如何以嵌套方式使用 UpdatePanel 控件	144

页面范例1 以嵌套方式查看产品类别与产品数据	148
页面范例2 嵌套的购物车	152
页面范例3 动态更改照片显示比例	158
3-8 如何在母版页面中使用 UpdatePanel 控件	160
页面范例1 示范如何在母版页面中使用ScriptManager控件	161
页面范例2 示范使用母版页面上的控件来局部更新内容页面内的UpdatePanel控件中的内容	164
页面范例3 示范使用母版页面上的控件来局部更新内容页面内的UpdatePanel控件中的内容	168
页面范例4 示范如何在内容页面中使用ScriptManagerProxy控件	170
3-9 哪些控件不兼容 UpdatePanel 控件	173
页面范例 搭配使用TreeView与UpdatePanel控件	174
结语	176

Chapter 04 一切随心所欲，探讨 UpdatePanel 的可程序化功能..... 177

4-1 如何以程序控制方式更新一个 UpdatePanel 控件	178
页面范例1 以程序代码实现更新UpdatePanel控件	178
页面范例2 以程序代码实现更新UpdatePanel控件	181
页面范例3 以程序控制方式局部更新工资数据与2D饼图	182
页面范例4 以程序控制方式局部更新产品类别照片与产品数据	187
页面范例5 异步的自行车产品目录——重要！	190
页面范例6 异步的照片相簿——重要！！	199
页面范例7 异步的水平滚动相簿——非常重要！	207
页面范例8 示范通过调用Update方法来灵活地显示与更新信息	216
4-2 如何以程序控制方式创建一个 UpdatePanel 控件	219
页面范例1 示范如何动态为页面添加一个UpdatePanel控件	219
页面范例2 示范如何整齐排列动态添加页面中的UpdatePanel控件的内容	220
页面范例3 示范如何动态为页面添加多个UpdatePanel控件	223
页面范例4 示范如何动态为页面添加多个UpdatePanel控件	229
页面范例5 示范如何动态为页面添加包含数据绑定控件的UpdatePanel控件	230
4-3 如何在自定义控件中启用局部更新功能	234
页面范例 示范创建一个启用局部更新功能的自定义控件	235
4-4 自定义 UpdatePanel 控件的错误处理	241
页面范例 示范如何自定义局部更新期间的错误处理	241
结语	246

Chapter 05 漫步于异步之间，探索 PageRequestManager..... 247

5-1 什么是 PageRequestManager	248
5-2 如何取得 PageRequestManager 的实例.....	248
5-3 客户端页面生命周期事件	249
5-4 如何取消异步回发	252
页面范例1 示范如何取消一个正在执行的异步回发	252
页面范例2 示范如何让一个异步回发拥有较高的优先级.....	260
页面范例3 示范采用默认的“后者为赢”的优先级处理模式	270
5-5 如何让 UpdatePanel 控件动起来.....	272
页面范例1 示范如何更新UpdatePanel控件并让它晃动	272
页面范例2 示范如何只让特定的UpdatePanel控件产生晃动效果	276
5-6 自定义局部更新的错误处理.....	284
页面范例1 示范如何自定义局部更新的错误处理.....	285
页面范例2 示范如何自定义局部更新的错误处理.....	289
结语	290

Chapter 06 别让用户干着急，善用 UpdateProgress 控件..... 291

6-1 自定义 UpdateProgress 控件的内容	292
6-2 了解 UpdateProgress 控件的解释方式.....	292
6-3 探讨 UpdateProgress 控件的显示时机.....	293
页面范例1 异步数据与照片查看界面（含进度提示）	294
页面范例2 异步邮件发送界面（含进度提示）	301
页面范例3 各部门男女平均工资的3D饼图查看界面（含进度提示）	304
页面范例4 异步平行输入输出界面（含进度提示）	309
6-4 用 JavaScript 程序代码来显示与隐藏 UpdateProgress 控件	315
页面范例 异步的章节目录提取界面（含进度提示与取消查询功能）	316
6-5 如何动态添加 UpdateProgress 控件	321
页面范例 示范动态添加UpdateProgress控件	321
结语	322

Chapter 07 定时操作的最佳利器，善用 Timer 控件 323

7-1 Timer 控件的运行机制	324
7-2 Timer 控件的使用模式	324
7-3 Timer 控件的摆放位置与其影响	324
7-4 Timer 控件范例集锦	332
页面范例1 定时显示性能计数器	332
页面范例2 定时更新多个统计图	337
页面范例3 自动播放相簿	342
结语	348

Appendix A 范例光盘使用说明 349

A-1 安装 SQL Server 2005 Express Edition	350
A-2 安装范例数据库并复制范例网站	350
A-3 使用页面范例	350
A-4 视频讲座	351

Chapter 01



万事开头难， 从了解AJAX开始学起

- AJAX 的用户观点
- AJAX 的开发人员观点
- 使用 AJAX 的思考
- AJAX 应用现状

越来越多的应用系统为了减少部署成本,纷纷将应用程序的用户界面从传统的窗口界面(Windows Form 窗体)更改成 Web 页面(Web Form,也就是 ASP.NET 网页),以便能够直接在大家所熟悉的 IE 等浏览器中运行应用程序。此举最大的好处,就是可以拥有最优的学习曲线,并进而降低教育培训成本。然而有利就有弊,受限于传统的网页技术,网页界面在功能性的丰富程度上、操作的灵活上(例如:拖放功能)、响应速度上……都不如传统的窗口窗体应用程序。相信大家都有这样的体验,使用页面应用程序时,时常要等待网站服务器(Web 服务器)将页面内容处理完毕并发送回客户端的浏览器之后,才能进行后续操作。令人不禁感叹,难道生命就该浪费在无谓的等待中吗?

当然不是,生命不该虚度光阴。因此,本书将说明如何使用 ASP.NET 3.5 的 AJAX 服务器端功能来整合现有的技术,以便突破传统网页的瓶颈。不过,在学习 ASP.NET 3.5 AJAX 之前,大家应该先了解 AJAX 的一些基本概念,这也正是本章的重点。闲话不多说,现在就让我们一起开始学习吧!

1-1 AJAX 的用户观点

相信页面程序开发人员对于 JavaScript、CSS(Cascading Style Sheets,层叠样式表)、DOM(Document Object Model,文档对象模型)、DHTML(Dynamic HTML)与 XML 等技术都不会感到陌生。所谓的 AJAX(Asynchronous JavaScript And XML),其实就是上述这些技术综合之后的强化体。从英文的含意可以了解到, AJAX 技术就是异步的 JavaScript 与 XML,它打破了传统页面开发技术的藩篱,使得页面更具互动性,也顺势打开了 Web 2.0 时代之门。

在过去,即使只是页面中极小部分的内容需要更新,也必须将整个页面发送给 Web 服务器来处理,处理完毕之后还必须再将整个页面的内容发送回客户端。这样不分青红皂白地一股脑式的处理,效率当然低,响应时间更是不佳。而 AJAX 的重要和高明之处在于,它只会将页面中需要更新的部分发送给 Web 服务器来处理,并且将处理后的内容发送回客户端浏览器来进行局部更新。由于只需要发送并处理少量的数据,而不是将整个页面内容重新加载到浏览器中,自然在效率上大幅提升。这样的做法,扎扎实实地提升了页面的互动性、响应速度与便利性,使得页面界面的用户体验大幅提高,更贴近于 Windows Form 应用程序。

或许这样说,大家还无法真正体会 AJAX 所带来的好处与便利性。下面这几个真实世界中的 AJAX 实际应用案例,相信大家必定会印象深刻。

- 在 AJAX 问世之前,大部分的网站都缺乏与用户互动的设计,即便是号称全球最大的网络书店——亚马逊(<http://www.amazon.com>),也仅能根据以前购物行为给予新品推荐。再以门户网站为例,多数的门户网站不允许用户自己设计页面,这大大限制了用户根据自身需求个性化页面的能力。在有了 AJAX 之后,浏览器首页可以由不同信息源的区块所组成,或在不同区块中显示各式各样的对象与信息源。比如说,以 <http://home.live.com> 的个性化网页而言,如图 1-1 所示,可以通过个性化功能,在网页的不同区块中摆放特定的小工具;甚至可以如图 1-2 所示,直接选择网页区块,再用鼠标将区块拖放到需要的位置上。如此一来,浏览器首页将呈现出符合用户需求和具有个人特色