Join the discussion! p2p.wrox.com

Wrox Programmer to Programmer™

Real World .NET 4, C#, and Silverlight:Indispensible Experiences from 15 MVPs

.NET、C#与Silverlight开发圣典

——分享15位[redacted]佳实践经验

Bill Evjen
[美] Dominick Baier 等著
György Balássy
王净 范园芳 李卉 译

华大学出版社

.NET、C#与 Silverlight 开发圣典

——分享 15 位 MVP 的最佳实践经验

Bill Evjen
[美] Dominick Baier 等著
György Balássy
王净 范园芳 李卉 译

清华大学出版社

北 京

Bill Evjen，Dominick Baier，György Balássy，et al
Real World .NET 4，C#，and Silverlight：Indispensible Experiences from 15 MVPs
EISBN：978-1-118-02196-5

图书在版编目(CIP)数据

.NET、C#与 Silverlight 开发圣典——分享 15 位 MVP 的最佳实践经验 / (美)依维恩(Evjen,B.)，(美)贝尔(Baier,D.)，(美)贝拉思(Balássy,G.)等著；王净，范园芳，李卉译. 一北京：清华大学出版社，2012.10

书名原名：Real World .NET 4，C#，and Silverlight：Indispensible Experiences from 15 MVPs
ISBN 978-7-302-29995-0

Ⅰ. ①N… Ⅱ. ①依… ②贝… ③贝… ④王…⑤范…⑥李… Ⅲ. ①网页制作工具—程序设计②C 语言—程序设计 Ⅳ. ①TP393.092②TP312

中国版本图书馆 CIP 数据核字(2012)第 211378 号

责任编辑：王　军　于　平
装帧设计：牛艳敏
责任校对：蔡　娟
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　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：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：36.5　　**字　　数**：820 千字
版　　次：2012 年 10 月第 1 版　　**印　　次**：2012 年 10 月第1 次印刷
印　　数：1～4000
定　　价：68.00 元

产品编号：043161-01

技术编辑简介

James Miller 是高级架构师和技术传播者，他利用最新的 Microsoft 平台和技术开发具有高扩展性、可维护性及可用性的企业应用程序。他曾在多个行业工作过，在公共和私人部门任过职，并且有近 30 年的程序设计经验，其经验几乎覆盖了软件生命周期的方方面面。他尤其精通架构和开发用于快速应用程序开发和敏捷实践的框架和工具。

Miller 多次荣获 Microsoft Certified Professional Developer 证书以及 Web、WF(Workflow Foundation，工作流基础)和 Silverlight 开发方面的 Technology Specialist 证书。他拥有密歇根大学的电气工程学士学位，专攻计算机和数字系统，并且辅修会计和金融。他和他的妻子、3 个儿子、两个女儿生活在密歇根州 Ann Arbor 外的乡村。

致　谢

你现在所看的这本书凝结了来自世界各地很多人的共同努力。感谢 Wrox 出版社的工作人员使本书最终出版。

—— Gill Cleeren

感谢 Proaction Mentors 为此所花费的时间和精力。感谢 Jef、Todd、Tony、Tim、Dave、Ken 以及 Improving Enterprises 的其他朋友为我灌输了敏捷的思想方式，并让我领会真正的 TDD。感谢 Craig Walls 教我 DI 的相关知识，感谢 Raymond Lewallen 向我介绍 BDD。特别要感谢 Microsoft 和所有的 MVP 程序。感谢本书中的所有 MVP 作者，以及 Paul 和 Wiley 的全体职员——感谢他们促成了本书的完成！最后，将我所参与的这一章献给全世界的开发者——那些努力磨练技能和改善技巧的开发人员。

—— Caleb Jenkins

特别感谢 Stephen Toub、Microsoft Patterns 和 Practices 所提供的文档以及对我所提问题的及时回答。

—— Jeffrey Juday

感谢 Steve Michelotti 和 Sajad Deyargaroo 给出了有益的反馈。

—— Vishwas Lele

首先，感谢 Lyns 在我写第 14 章时所提供的支持。感谢你的耐心和辛勤的工作，并且像“家庭成员”那样照顾我。非常感谢你。

—— Scott Millett

前　言

本书是由多位作者共同完成的。在一开始计划编写本书时，就想编写一本不同风格的书。如今市面上的很多计算机图书都是对计算机技术的某一特定领域进行详细的讲述，由一位或少数几位作者来编写。不管你感兴趣的是 C#、ASP.NET、XML(Extensible Markup Language，可扩展标记语言)还是 WCF(Windows Communication Foundation，Windows 通信基础)，都可以找到从入门到精通各个阶段的相关图书。如今在市面上可以找到很多相关的参考书目。

而本书并非如此，相反，我们的出发点是“汇集当今业界最好的从业人员(Microsoft MVP 和 Microsoft Regional Director)，让他们就自己最熟悉的领域单独编写一章”。

当然，这些作者对.NET 都有全面的了解，但是却只专门编写自己喜欢和感兴趣的部分(也是他们最熟悉的领域)。总的来说，本书由通用领域中的一系列大文章组合而成，覆盖了.NET Framework 的大部分领域。

目前，.NET Framework 包含的内容非常庞大，所以在使用时不可能了解所有的内容。这也就是很多软件开发人员组成开发团队的原因。他们注重将个人组织在一起成为一个整体，以便可以全面了解将.NET Framework 作为工作基础所产生的强大力量。

当开始处理.NET Framework 各个领域中的工作时，你将会发现本书是非常好的资源。当需要完成一些.NET Framework 中陌生领域的工作而又未曾花费时间来完全了解该领域时，本书可以作为技术顾问。

读者对象

本书适合专门使用.NET Framework 构建解决方案的中高级开发人员使用。在本书中可以找到从 Web 开发直至终端开发的所有相关内容。

主要内容

本书覆盖了许多.NET Framework 的核心领域。在讲授 Silverlight 之前，首先通过重点讲授 ASP.NET 来介绍客户端的相关内容。而在讲授 ASP.NET 时，还介绍了在构建 ASP.NET 应用程序后如何使用这些应用程序以及如何使用 jQuery(jQuery 是目前开发 Web 应用程序最为流行的方法)。除了 ASP.NET，还介绍了如何在 Silverlight 中应用 MVVM

(Model-View- ViewModel，模型-视图-视图模型)等模式。然后，将 Silverlight 的覆盖范围由 PC 客户端扩展到手机客户端。而当处理客户端时，有一章讨论了如何在设计者和开发者之间搭建桥梁。

介绍完客户端开发工作后，介绍了 WCF 等通信技术以及一些保证通信安全的方法(如使用 WIF，即 Windows Identity Foundation，Windows 标识基础)。其中介绍了一些特殊的通信协议(如 REST 和 OData)以及.NET Task Parallel Library(.NET 任务并行库)。

之后的几章介绍了一个关键主题，包括使用 Windows 工作流和 WPF 数据绑定。而最后几章则介绍了开发生命周期的各个方面，其中包括使用用户故事及利用单元测试进行开发。

总之，本书内容丰富，每章为成功完成手头的工作提供了有针对性的内容。

使用本书的条件

.NET Framework 4.0 可以在 Windows XP、Windows 2003、Windows 7 及最新的 Windows Server 2008 R2 中运行。要使用.NET Framework 编写代码，需要安装.NET 4 SDK。

此外，除非打算使用文本编辑器或其他一些第三方开发环境编写 C#代码，否则必定需要使用 Visual Studio 2010。运行托管代码并不需要完整的 SDK，但需要.NET 运行库。

同时，虽然本书中用 C#来显示所有的代码示例，但如果愿意，可以对这些示例进行转换，并在 Visual Basic 中完成相同的功能。

源代码

在练习书中的示例时，可以选择手动输入代码或者使用本书附带的源代码文件。书中用到的所有源代码都可以从 www.wrox.com 下载。进入站点 http://www.wrox.com 后，只需要找到本书的书名(使用 Search 搜索框或书名列表)，单击本书详细信息页面上的 Download Code 链接，就可以得到本书所有的源代码。

注意：因为很多书的书名都相似，所以用 ISBN 搜索更为容易。本书英文版的 ISBN 是 978-1-118-02196-5。

下载完代码后，用您喜欢的压缩工具把它解压缩。此外，也可以去 Wrox 的主下载页面 www.wrox.com/dynamic/books/download.aspx 找到本书或 Wrox 出版的其他书籍的代码。

勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误，但人无完人，错误难免会发生。如果您在 Wrox 出版的书中发现了错误(例如拼写错误或代码错误)，我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间，同时也会帮助我们提供更高质量的信息。

要找到本书的勘误页面，可以进入 www.wrox.com，使用 Search 搜索框或书名列表定位本书，然后在本书的详细信息页面上单击 Book Errata 链接。在这个页面上可以查看为本书提交的、Wrox 编辑粘贴上去的所有错误。完整的书名列表(包括每本书的勘误表)也可以从 www.wrox.com/misc-pages/booklist.shtml 上获得。

如果您在本书的勘误页面上没有看到您发现的错误，可以到 wkservice@vip.163.com 上填写表单，把您发现的错误发给我们。我们会检查这些信息，如果属实，就把它添加到本书的勘误页面上，并在本书随后的版本中更正错误。

p2p.wrox.com

如果想和作者或同行进行讨论，请加入 http://p2p.wrox.com 上的 P2P 论坛。该论坛是一个基于 Web 的系统，您可以发布有关 Wrox 图书及相关技术的消息，与其他读者或技术人员交流。该论坛提供了订阅功能，当您感兴趣的主题有新帖子发布时，系统会邮件通知。Wrox 的作者、编辑、其他业界专家和像您一样的读者都会出现在这些论坛中。

在 http://p2p.wrox.com 网站上，您会找到很多不同的论坛，它们不但有助于您阅读本书，还有助于您开发自己的应用程序。加入论坛的步骤如下：

(1) 进入 http://p2p.wrox.com，单击 Register 链接。

(2) 阅读使用条款，然后单击 Agree 按钮。

(3) 填写加入该论坛必需的信息和其他您愿意提供的信息，单击 Submit 按钮。

(4) 您将收到一封电子邮件，描述如何验证您的账户和完成加入过程。

注意：不加入 P2P 也可以阅读论坛里的消息。但是如果要发布自己的消息，就必须加入。

加入之后，就可以发布新的消息和回复其他用户发布的消息。可以随时在 Web 上阅读论坛里的消息。如果想让某个论坛的新消息以电子邮件的方式发给您，可以单击论坛列表中论坛名称旁边的 Subscribe to this Forum 图标。

要了解如何使用 Wrox P2P 的更多信息，请阅读 P2P FAQ，其中回答了论坛软件如何使用的问题，以及许多与 P2P 和 Wrox 图书相关的问题。要阅读 FAQ，单击任何 P2P 页面上的 FAQ 链接即可。

目　录

第 1 章

ASP.NET 和 jQuery

David Giard

大约在 10 年前，Microsoft 公司推出了 ASP.NET——一个使用新的.NET 平台构建 Web 应用程序的框架。设计 ASP.NET 的目的是帮助开发人员构建、部署和维护 Web 页面与网站。ASP.NET 集成了 IIS(Microsoft Internet Information Server，Microsoft Internet 信息服务器)，同时还为开发人员提供了大量用于开发动态 Web 应用程序的工具。

推出 ASP.NET 时，该框架主要侧重于 Web Forms。Web Forms 将很多 HTML 和 HTTP 低层次的复杂性抽象化，以便开发人员可以像构建 Windows Forms 那样构建 Web Forms。通过使用 Web Forms，即使开发人员对网络底层技术知之甚少也可以快速创建交互式的 Web 页面。这种开发体验与流行的 Visual Basic 语言类似，所以这就要求开发人员熟悉 Visual Basic 语言。

与其前身 ASP(Active Server Pages，动态服务器页面)一样，ASP.NET 为开发人员提供了使用代码与标记相结合的方式构建 Web 应用程序的能力。

但是与传统的 ASP 不同的是，ASP.NET 建立在 CLR(Common Language Runtime，公共语言运行库)基础之上，并且使用了.NET 的强大功能。正是由于这个原因，ASP.NET 充分利用了 CLR 的优势，如自动垃圾收集机制、丰富的库及健壮的安全模型。

ASP.NET 的最新版本包括了对 ASP.NET Web Forms 框架的增强及 ASP.NET MVC(Model-View-Controller，模型-视图-控制器)框架的更新版本。Visual Studio 2010 也是第一个附带 jQuery 库的版本。开发人员可以使用 jQuery 库为自己的网站构建丰富的客户端功能。

本章将重点介绍 ASP.NET 4 的新功能。首先，将介绍 ASP.NET 4 Web Forms 的新特性及增强功能；然后介绍 ASP.NET MVC；最后介绍如何有效地使用 jQuery 来增强应用程序的功能。

在学习过程中都会有示例应用程序来指导如何实现这些新功能。

1.1 了解 Web Forms

在 ASP.NET 1.0 中就引入了 Web Forms Framework。通过该平台已经成功地构建和部署了无数网站。

从本质上讲，Web Forms Framework 的设计目的是将 HTML(Hypertext Markup Language，超文本标记语言)和 HTTP(Hypertext Transfer Protocol，超文本传输协议)的复杂性抽象，从而使 Web 开发像 Visual Basic 窗体开发那样容易。下面列出了 Web Forms 应用程序开发的主要特征。

- 开发人员所看到的是类似于 Web 页面的设计界面。
- 可以将 Web 控件拖拽至设计界面。
- 可以在页面和控件上设置属性。
- 可以通过编写代码来操纵页面、控件及相关联的数据。

开发人员可以使用 Visual Studio 快速构建 Web 应用程序。同时还可以使用丰富的内置控件及第三方控件来减少开发的工作量。

然而，快速应用程序开发(rapid application development，RAD)是需要付出相应代价的。Web Forms 和相关联的控件会自动生成 HTML 及 JavaScript 代码。有时所生成的代码并不能完全满足需求。同时为了增加难度，不同的 Web 浏览器会以不同的方式来显示相同的 HTML 和理解 JavaScript 代码——特别是 JavaScript 与 Web 页面的 DOM(Document Object Model，文档对象模型)交互时更是如此。这样就产生了一个常见的问题：在某种浏览器中可以正确显示的页面，在另一种浏览器中却不能显示。

使用 Web Forms 时，很难修改输出的 HTML 或 JavaScript。特别是想要在多种浏览器显示时，这个问题就更为突出。如果无法对 HTML 进行相应的修改，也就无法对某些浏览器提供支持。为此，Web 开发人员应该尽可能地了解 HTML、CSS(Cascading Style Sheets，级联样式表)和 HTTP。像 Web Forms 这样的抽象概念是非常好的，但是开发人员同样需要理解所抽象的技术。

Microsoft 在 ASP.NET 4 中引入了许多增强功能。下面列举了一些最重要的功能。

- 视图状态(View State)改进的处理方法使用户能更加灵活地控制视图状态中的控件。
- web.config 文件被大大简化。
- 新的 Web 项目(Web Project)模板提供了更强大的功能。
- web.config 转换使 Web 应用程序可以更容易地部署到新环境中。

现在具体介绍这些重要功能。

1.1.1 视图状态

在默认状态下，Web 及其底层的传输协议(HTTP)是无状态的。这也就意味着在默认情况下，每个请求对前一个请求一无所知。而 ASP.NET 引入了视图状态来保持 Web 应用程序中页面之间的状态。

视图状态通过在 HTML 表单中自动创建一个额外的、隐藏的输入项来保持状态。为了保存页面中所提交字段的状态，ASP.NET 将视图状态数据编码为单一的字符串，并将其设置为隐藏字段的值。该隐藏字段在服务器端创建，并随着 Web 页面的其他部分一起发送到客户端。

然而，这种便利也是需要付出代价的。每次提交表单时，也会提交编码数据。而每次响应请求时，编码数据又会返回到浏览器。如果视图状态中存储了大量的数据，那么编码数据也会非常庞大，就会使客户和服务器之间的通信变得缓慢。

在使用 ASP.NET 的早期版本时，开发人员可以在页面级别或控件级别启用或禁用视图状态。在 ASP.NET 4 之前，可以通过三个选项启用视图状态：

- 关闭整个页面的视图状态——如果选择了该选项，则不会体验到视图状态的任何好处。
- 启用整个页面的视图状态——如果选择了该选项，则可能会对那些不必要的项进行编码并将其添加到视图状态中，从而增加了服务器与浏览器之间传递的有效载荷，这样也将使应用程序运行变慢，影响用户的使用。
- 启用整个页面的视图状态，同时对那些不需要视图状态的控件禁用视图状态——如果希望为页面中的多数控件(少数控件除外)启用视图状态，那么选择该选项是非常合适的。但如果只是为少数控件启用视图状态，那么就不要选择该选项。

ASP.NET 4 提供了两个相关联的属性来控制视图状态：ViewStateMode 和 Enable ViewState。其中 ViewStateMode 是 ASP.NET 4 中新增加的，而这两个属性需要一起使用。

从 ASP.NET 1 开始就已经有 EnableViewState 属性了。可以为整个页面或者单个控件设置该属性，其属性值为 True 或 False。然而，EnableViewState 并不会为任何控件启用视图状态。更确切地说，该属性确定某一控件以及控件层次结构中该控件下面的控件是否可以启用视图状态。只有在 EnableViewSate 属性值为 True 时，才可以通过控件的 ViewStateMode 属性启用或禁用视图状态。可以显式地启用或禁用 ViewStateMode，也可以从父容器中继承该属性。

如果将页面或控件的 View State 属性设置为 False，则会禁用页面或控件(及其所包含的任何控件)的视图状态。

在 ASP.NET 的早期版本中，启用和禁用控件的视图状态的唯一方式是在页面中设置 EnableViewState= " True " ，然后对那些不想为其维护由于在客户端和服务器之间传递视图状态而产生的系统开销的控件，设置 EnableViewSate= " False " 。

可以将 ViewStateMode 设置为页面或控件的属性，也可以在代码中对其进行设置。可以设置的值如下：

- Enabled——启用某一控件的视图状态。
- Disabled——禁用某一控件的视图状态。
- Inherit——告知 ASP.NET 将某一控件的 ViewStateMode 设置为该控件父容器的 ViewStateMode 值。对于用户控件而言，Inherit 是默认值。

只有在某一控件或控件父容器的 EnableViewState 设置为 True 时，ViewStateMode 才会