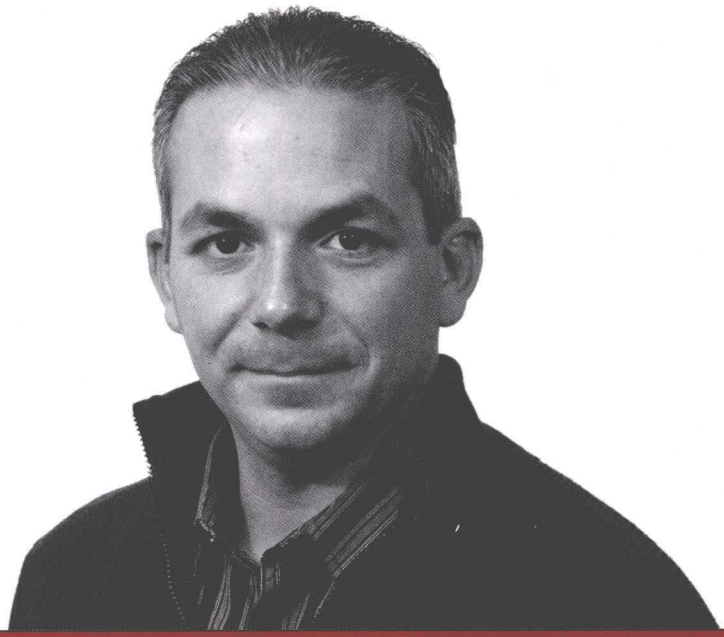
PROGRAMMER TO PROGRAMMER™



ASP.NET 3.5 Enterprise Application Development with Visual Studio 2008 Problem-Design-Solution

ASP.NET 3.5 商用开发架构精解

(美) Vince Varallo 著
刘建宁 张敏 常洁 译

wrox™
WILEY
清华大学出版社

ASP.NET 3.5 商用开发架构精解

(美) Vince Varallo 著
刘建宁 张 敏 译
常 洁

清华大学出版社
北 京

Vince Varallo
ASP.NET 3.5 Enterprise Application Development with Visual Studio 2008：Problem-Design-Solution
EISBN: 978-0-470-39686-5

图书在版编目(CIP)数据

ASP.NET 3.5 商用开发架构精解/(美) 瓦拉洛(Varallo, V.) 著；刘建宁，张敏，常洁 译.
—北京：清华大学出版社，2010.12
书名原文：ASP.NET 3.5 Enterprise Application Development with Visual Studio 2008：Problem-Design-Solution
ISBN 978-7-302-24040-2
Ⅰ.A… Ⅱ.①瓦… ②刘… ③张… ④常… Ⅲ.主页制作—程序设计 Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字(2010)第 207138 号

责任编辑：王　军　赵利通
装帧设计：孔祥丰
责任校对：胡雁翎
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**：29.75　**字　数**：724 千字
版　　次：2010 年 12 月第 1 版　　**印　　次**：2010 年 12 月第1 次印刷
印　　数：1～3500
定　　价：59.80 元

产品编号：034127-01

作 者 简 介

Vince Varallo拥有超过12年的Microsoft平台开发经验，目前在MTI Information Technologies担任程序开发总监。他使用 ASP.NET 和 SQL Server 开发医药行业的销售软件。在加入 MTI 之前，他曾经从事医药行业和金融行业，先后开发了临床数据管理系统、内部门户网站以及金融终端软件。

Vince 热衷于探索新技术，喜欢打高尔夫和骑自行车，并且是 Phillies 球队的球迷。他曾与人合著了 *Professional Visual Basic 6: The 2003 Programmer's Resource* 一书。

前　言

本书循序渐进地介绍了如何使用 Visual Studio 2008 中的新功能来开发 ASP.NET 3.5 应用程序，对每一个新功能都进行了详尽地介绍，并最终生成了一个解决方案，您可以将该解决方案作为起点来构建自己的应用程序。

如果您负责设计或开发企业级应用程序、部门级应用程序、门户网站或者各行业应用程序，那么本书将适合您。许多应用程序都会有一组相似的功能，本书中构建的应用程序就包含了一些企业应用程序的通用功能。其实每个应用程序都有一套相同的功能，只是各自有不同的实现方式。它们都采用后台数据库，而开发人员则负责使用户可以添加、更新、选择和删除记录。但实际情况并非这么简单。

实际的开发工作是从需求分析开始的，这时您要与用户在一起交谈，在第一时间了解他们的业务流程以及他们为何需要一个新的系统，或者为何要对系统进行改进。许多公司都有这样的部门，他们使用 Excel 和 Access 向导来创建小型应用程序，但最终一部分业务就要依靠这些应用程序来执行。由于使用工具的原因，经常会出现一些问题，召集高级管理人员开会、雇用项目经理和程序员、求助项目管理办公室(Project Management Office，PMO)，这样的事情时有发生。突然之间，松散定义的业务成了被优先考虑的对象，人们希望有标准的操作过程文档，希望能得到审计报表，希望能够少投入多产出，当然更希望能有一个系统可以完成所有这些工作，这也是您阅读本书的目的所在。但仔细考虑这个问题，您就会发现这是一个非常艰巨的任务。您要在业务流程、业务缺陷等各方面都成为专家，并且要能够创建一个公司赖以生存的系统。或许我有点夸大其词了，但当您想得到职务上的提升时可能也会有同样的言论。

本书将教会您如何构建一个可扩展的应用程序框架，您可以使用该框架来创建一个解决方案，以解决企业所面临的问题。设计模式采用了标准的三层结构，即用户界面层、业务逻辑层和数据访问层，还在每一层中创建了通用的业务逻辑类，将基于角色的安全模型、工作流、报表、动态菜单、数据输入、动态查询、通知、异常处理和审计等通用的业务封装起来。本书在推出整个解决方案的过程中，详细地对每一个业务需求进行了定义，并通过 ASP.NET 3.5 和 Visual Studio 2008 的最新功能，在一个可重用的框架中实现了它们。

企业应用程序通常是很复杂的，开发小组中的人员形形色色。有项目发起人、项目经理、业务分析师、架构师、UI 开发人员、中间层开发人员、数据库开发人员，或者还有测试人员。要记住：用户不是测试人员。如果您曾与专业的测试人员共事过，就会认识到他们在整个开发过程中有多么重要，他们是软件质量真正的保证。许多公司不情愿聘用专业测试人员，所以用户或者开发人员就扮演了测试人员的角色。本书的读者主要是架构师和开发人员，但对测试人员也具有参考价值，可以帮助他们了解企业级应用程序的整个设计开发过程。

我非常幸运，能够经常与用户或企业主来讨论需求，并负责开发解决方案来满足他们的需要。无论是与 HR 讨论部门级的应用程序，还是与副总裁研究企业级的应用程序，当把他们的需求分解成可以进行开发的实体对象时，就会发现他们需要的应用程序或多或少

有些类似。他们从不同的地方收集数据，但是都有自己的业务流程，需要 E-mail 通知、报表打印以及基于角色的安全体系。因此，本书中构建的企业应用程序框架为您提供了一个基础架构，您可以对它进行扩展来实现企业特定的业务需求。

读者对象

本书适合中高级开发人员或系统架构师。若使用过 Visual Studio、.NET Framework、ASP.NET 以及 C#，则更有帮助。因为书中的示例是基于 ASP.NET 的，但其设计模式适用于任何编程语言。本书关注企业级应用程序，但采用的模式适用于任何一种拥有 Web 前端并连接后台数据库的应用程序。

主要内容

本书中的示例采用 Visual Studio 2008、ASP.NET 3.5、C#以及 SQL Server 2005 构建。每一章都详细地介绍了一个模块，提供了丰富的示例代码，并使用了 Visual Studio 2008 中的新功能以及.NET Framework 3.5 中增强的语言特性。解决方案中包括诸如 LINQ to SQL、母版页、自定义控件、GridView、业务对象、数据对象、Crystal Reports 等新技术的示例。新的语言特性包括 LINQ、扩展方法、部分方法、自动属性、匿名类型、lambda 表达式以及新的对象初始化方法。

当然，代码才是大多数开发人员的兴趣所在，所以每一章中都提供了很多示例。

组织结构

本书的每一章都分为三个部分，首先提出要解决的问题，然后针对问题考虑解决方法，最后给出问题的解决方案。解决方案中包含了大量的代码。由于每一章都以前一章的内容为基础，所以建议您按照顺序来阅读。在前几章中介绍的基类对于理解本书其他部分的内容至关重要。后面各章以这些基类为基础，在三层结构的每个层中扩展它们的功能。

使用要求

- Visual Studio 2008 标准版或专业版
- SQL Server 2005 或 SQL Server 2005 Express
- Windows Vista、Windows XP、Windows 2003 或 Windows 2000

源代码

在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 http://www.tupwk.com.cn/downpage 上下载。登录到站点 http://www.wrox.com/，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

提示：

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 978-0-470-39686-5。

在下载了代码后，只需对其执行解压缩操作。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正文和代码中不出现错误，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，同时也有助于我们提供更高质量的信息。

请给 wkservice@vip.163.com 发送电子邮件，我们将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 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 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您发布与 Wrox 图书相关的消息和相关技术，以及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，它可以给您发送感兴趣的主题。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，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，但要发布自己的消息，就必须加入该论坛。

加入论坛后，就可以发布新消息和回复其他用户发布的消息。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

企业应用程序框架

本书中要实现的解决方案来源于一个虚构的公司，主要用于人力资源部门对员工的请假及休假申请进行处理并制作报表。该解决方案将使用 Visual Studio 2008 进行开发，基于 ASP.NET 3.5，采用 C#进行编程，后台数据库为 SQL Server 2005。

概念虽然简单，但我们要将解决方案设计得足够灵活，使它能方便地进行扩展，从而可以满足您自己的业务需求。本章将对项目需求进行定义，并介绍它是如何进行架构的。每一章都专门针对特定的需求采用三层结构实现一种解决方案，这三层结构分别是：用户界面(User Interface，UI)、业务逻辑层(Business Logic Layer，BLL)以及数据访问层(Data Access Layer，DAL)。

1.1 提出问题

Sue 是一位人力资源副总裁，她采用 Excel 与 Word 模板相结合的方式来处理员工的休假和请假申请。她注意到，在使用这种方式时很多请求无法说明原因，并且难于进行跟踪。她想为整个公司构建一个系统，用一个企业级应用程序来代替当前的模板，她的想法得到了认可。作为项目的发起人，她任命 Mary 为 IT 部门的关键联系人，以便于收集需求。下面的对话可能发生在该项目的初始阶段，您或许会发现任何项目都有这么一个相似之处。

Mary：经理让我负责这个项目，但我对计算机和如何构建系统一点也不了解。他们几年前就想做个类似的系统，但没什么进展，开发人员也被解雇了。我先对我们要完成的工作简单说明一下。在 Z 盘上有一个 Word 模板，每个人要申请休假或请假的话就必须填写它。去年 12 月我们把记录了员工假期均衡表的 Excel 文件发送给经理后，发现了很多不一致的地方。经理有时忘记把休假申请发送给我们，或者员工有时候会取消申请，但却没有通知我们。现在需要用一个数据库来替代这些 Word 模板。

我：所以您想将这个过程自动化。我用过这种模板，所以对这个过程有所了解，但不清楚当经理签字后会发生什么，您能解释一下这个流程吗？

Mary：当员工想休假时，需要填写表单，并在打印后签字，然后交给他们的主管经理来批准和签字。主管经理签字后会通过办公室之间的邮件发送到人力资源部，我就对照着

Excel 电子表格进行检查，看该员工是否有足够的天数来休假，然后在表中减去相应的天数。另外，如果一次性休假超过 2 周，就需要由经理和副总裁签字。这就会造成问题，因为副总裁很忙，经常忘记把表单发送给我们。

我：那么您需要一个用户休假请求管理系统，在系统中构建一个工作流，让经理、副总裁和您可以对休假申请进行审批，员工请假的天数决定了所需要的审批级别。

Mary：对的，是这样。我认为还应该有驳回休假申请的功能。我们通常只是把它们否定掉，但是保留下来或许是个好主意，这样就能参考一下。

我：您提到过一个问题，就是员工取消申请后不通知您。那么，系统中是否要允许用户取消休假申请？如果需要，我可以让程序在请求被取消时向您发送一个邮件通知。此外，在请求到达流程的每一点时都可以向相应的审批者发送一个邮件。

Mary：是的，那就更好了。然后我就可以在 Excel 电子表格上相应地进行调整了。

我：您能把那个 Excel 电子表格发送给我吗？我可以在系统中以报表的方式重新生成电子表格的内容，然后可以把它导出为 Excel。这样，所有的请求和取消请求都可以在系统中追踪到，以后不需要再维护 Excel 电子表格了。

Mary：好，我可以发送给您，但是如果不用 Excel 电子表格的话，我应该在某个地方能对休假情况进行追踪，我猜是通过数据库。您能实现这种功能吗？

我：我们肯定能实现。您输入一个初始值，然后若是请假则减去相应的天数，若是取消，则把相应的天数加回来。是这样的吗？

Mary：对！另外，员工最多可以将 5 天假期累积到下一年中。每年从 1 月 1 日开始进行计算。

我：这个我们也可以实现。现在谈谈安全性吧。您是否想让一些人只能访问报表，而其他人可以访问所有功能？

Mary：我还没考虑过这一点，但可以肯定的是我不希望 Bob 能够输入初始值。他只需要打印报表，不需要做别的。

我：好，您需要的是基于角色的安全设置，允许通过分组来设置只读权限、编辑权限、无权访问以及报表的执行权限。

Mary：我不知道什么是基于角色的安全设置，但听起来是这样的。

我：那么追踪审计(Audit Trail)呢？您是否需要查看系统中的休假请求或安全设置上发生的任何更改？举个例子，如果有人进入到应用程序中，授予某人报表访问权限，那么系统就可以捕获到这个操作，您可以把这些操作打印成一个报表。您需要这种跟踪功能吗？

Mary：我认为是这样，我没有考虑过这些，但我的经理可能会喜欢这个功能，QA 部门中那些对我们的工作进行阶段审查的人可能也希望看到它。您能实现吗？

我：是的，可以实现。那么如何查看那些请求呢？是否需要提供不同的方式？例如查看重要的请求，取消的请求，经理的请求或者某个部门的请求。您现在不需要了解这些不同的休假请求，我构建的系统能让您用不同的方式对它们进行浏览，您不需要向技术支持人员求助就可以生成一个自定义的报表。甚至可以把这些查看方式放到主页中，这样一进入系统就能看到等待您处理的请求。

Mary：放到主页上吗？我喜欢这个想法！

我：好吧，现在我已经了解到足够的信息了。以后我还会到您这来把其余的需求了解

清楚。您需要这样一个系统：有工作流来处理请求，有报表功能、基于角色的安全设置、主页、电子邮件通知、查询功能以及追踪审计功能。您赞同吗？

Mary：我对查询功能不是很清楚，但我相信您。另外，系统必须好用，并且保持一致。我不想有的页面是绿色，有的却是蓝色，字体也是千奇百怪。

我：我们会注意的。过不了多久我会回来向您了解更多关于处理上的问题的，不过现在了解的已经够多了，很期待与您合作。

Mary：谢谢您能抽时间过来，希望您能比上一个开发人员做得好。

除了上面这些通过与 Mary 讨论而得出的需求外，对公司中所有的软件还有一些 IT 需求。IT 部门要求不能在用户的桌面上安装任何程序，并且所有的公司内部程序要使用单点登录。IT 部门不参与到应用程序的系统管理中去，只是在购买服务器与提供 SLA(Service-Level Agreement，服务等级协议)时提供参考。

Mary 已经把主要的需求告诉了我们。我将向您展示如何把需求变成一个软件——从概念到设计再到开发。希望您能围绕着与 Mary 的其余对话进行下去，乐趣自然来！

1.2　设计方案

如前所述，解决方案是由使用了三层体系结构的 ASP.NET 3.5 Web 应用程序组成的。接下来的各章将在总体需求上进行扩展，最终开发出一个功能完整的、健壮的企业应用程序。下列各小节对每章涉及的主题进行简要介绍。

第 2～5 章将介绍整个框架的体系结构，它们是着手进行开发之前所要阅读的最关键的章节。每章都详细叙述并使用了 Visual Studio 2008 提供的新功能。第 6 章及随后的各章演示了如何在应用程序中实现 Mary 的需求，并提供了丰富的代码，您可以将这些代码应用到自己的应用程序中。最终的解决方案可以从 Wrox 的网站(www.wrox.com)或 http://www.tupwk.com.cn/downpage 上下载。

1.2.1　第 2 章：数据访问层

从概念上来说，传统的三层体系结构中最简单的就是数据访问层。简单地说，数据访问层调用数据库中的存储过程或对数据库执行动态 SQL。在.NET Framework 以前的版本中，作为开发人员，一定要清楚用何种 ADO.NET 对象来执行存储过程，以及将何种类型的对象传递给业务层。关于传递给业务层的对象，一直存在着激烈的争论，不过.NET Framework 提供了一个工具来解决这些纷争。

本章介绍了 LINQ to SQL 这个新功能，并利用内置的 ORM 设计器工具创建了一些实体类，这些类用来同数据库进行通信，并被传回业务层。在 LINQ to SQL 中仍然可以使用存储过程，我将展示如何使用它们。

1.2.2　第 3 章：业务逻辑层

业务逻辑层也称为中间层，是应用业务规则在数据库中对数据进行保存和删除的地

方。仅仅按照表中的字段来创建一个映射类是不够的。业务逻辑层必须通过实现业务规则来保护应用程序中的数据的完整性，例如必填字段、唯一字段、约束以及计算关系等。如果违反了业务规则，业务逻辑层必须将它们传回调用端，而不能传递到数据访问层。本章介绍基类的设计模式，这些基类封装了中间层所有业务对象的业务规则。本章还将介绍创建业务对象列表和基类的实现方式，它们是构造程序中的列表对象的基础。

1.2.3 第 4 章：用户界面层

用户界面层是程序中唯一的用户可见部分，保持其一致性和易于导航是很重要的。可以通过样式表和 ASP.NET 主题来控制应用程序中的字体及颜色，但开发人员要对应用程序的整体导航负全责。试想，在某个页面中您将一个 Save 按钮放在左下角，而在另一个页面中却将它放在右上角，那么使用样式表也无济于事。本章将介绍样式表、母版页(master page)、嵌套式母版页(nested master page)，以及整个程序中将要使用的 UI 框架。还将创建两个自定义服务器端控件，用于生成动态菜单，它们最终将与第 6 章中构建的基于角色的安全体系相集成。

1.2.4 第 5 章：异常处理

这是在构建应用程序时经常被忽略的一个方面，但它是整个应用程序最重要的部分之一。我看到一些新手有时完全忘记给应用程序添加异常处理；有时不能正确地进行异常处理；有时异常发生时没有报告给任何人。当异常发生时，依靠应用程序通知比依靠用户通知更好。它不但对调试过程有帮助，还能使您看起来更像一个具有前瞻性的开发人员。本章将介绍 Microsoft 异常处理应用程序块(Microsoft Exception Handling Application Block)，我所有的应用程序都使用它进行异常处理。它是一个类集，能根据异常的类型提供日志及电子邮件发送功能，并且是可配置的。

1.2.5 第 6 章：基于角色的安全体系

大多数商业应用程序都需要基于角色的安全体系，组织中的领导者通常希望特定的组(group)可以完全控制所有页面，而其他的组中，有的可以完全控制某些页面，而有的只能拥有某些页面的只读权限。本章将演示如何在系统中创建角色，以及为角色分配权限的一个模式。这些权限分为“无权限”、“只读”和“编辑”，可以选择将它们与应该或不应该向用户显示的菜单项关联起来。用户可以拥有单个或多个角色。此模式使组织中的领导可以自己管理应用程序，而不依赖于 IT 人员在将应用程序发布后建立用户和角色。

1.2.6 第 7 章：工作流引擎

在企业应用程序中，大部分工作流是类似于请假以及准假这样简单的流程。Microsoft 专门针对工作流开发了 Windows Workflow Foundation (WWF)，给开发人员提供了一个基础，帮助他们构建应用程序。要学习 WWF，在书中用一章的内容讲述它是远远不够的。

但在本章中您可以学到一个可以利用的模式，只需要通过几个表和类的协作就可以简单地实现工作流。任何基于状态机的工作流应用程序都可以使用此模式。例如，批准休假、批准旅游、许可网络访问甚至是文档追踪系统。

1.2.7　第 8 章：通知

自动通知功能会使应用程序增色不少，尤其是在工作流应用程序中。当提交了一个请求等待批准时，或者请求被批准、被否决或在一定时间内请求没有得到处理时，系统应该通知用户。本章将介绍如何构建 Windows 服务应用程序，用于监视系统中的活动，并且根据系统中的事件向相应的用户发送电子邮件。本章不仅展示了通知服务的设计模式，还介绍了如何构建、调试和安装 Windows 服务。

1.2.8　第 9 章：报表

Visual Studio 中自带的免费的用于.NET 对象的 Crystal Reports 很强大，可以给您的应用程序带来专业的视觉效果和感受。本章将展示如何利用 Crystal Reports 对象将数据显示为 HTML、PDF、Excel 或 Word 格式。这里还是以中间层作为核心，获取数据进行处理，然后传递给用户界面。在用户界面层，Crystal Reports 对象只需要获取传递的数据，并将它转化为恰当的格式即可。此设计模式可以使中间层对象在多个报表视图中重用，本章还展示了如何在报表中动态设置分组，或显示页眉、页脚，以及详细资料行。

1.2.9　第 10 章：查询生成器控件

给予用户访问数据报表或数据视图的功能后，他们就会希望能够根据日期、用户、状态等属性来过滤数据。大多数情况下，用户需要的是相同的数据，只是通过不同的方式进行过滤。本章将展示如何构建一个服务器端的查询生成器控件，该控件使用 AJAX 技术，允许用户自己动态创建过滤器，可以用于视图以及第 9 章设计的任何报表中。这使用户能够提取感兴趣的数据，并在 Excel 中操作它们或者简单地将它们显示在 PDF 文件中。

1.2.10　第 11 章：仪表板

仪表板现在已经成为了 Web 应用程序的一个标准，通过 Visual Studio 内置的 Web Part 控件，可以很轻松地创建出实用的主页，主页上可以显示图表、警报、文档等信息，并把最重要的数据直接展示在用户眼前。本章将介绍 Web Part 控件及其使用方式，并为 Paid Time Off 应用程序创建包含了仪表板的主页，包括对未能及时处理的休假申请进行提醒。

1.2.11　第 12 章：追踪审计

在大多数企业应用程序中，追踪审计是很重要的，有时候甚至是必要的。由于 2002