

## <<ASP.NET 3.5商用开发架构>>

### 图书基本信息

书名：<<ASP.NET 3.5商用开发架构精解>>

13位ISBN编号：9787302240402

10位ISBN编号：730224040X

出版时间：2010-12

出版时间：清华大学

作者：瓦拉洛

页数：462

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 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ASENET3.5 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，希望能得到审计报告，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ASENET3.5和VisualStudio2008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。

## <<ASP.NET 3.5商用开发架构>>

### 内容概要

如果您在开发业务应用程序时面临许多挑战，vince varallo可以助您一臂之力，他是一位经验丰富的作者，也是一位资深的asp.net专家。

通过阅读《asp.net 3.5商用开发架构精解》，您将轻松掌握使用visual studio 2008开发asp.net应用程序所需要的工具，并学会如何开发易于维护和扩展的应用程序，从而能为特定问题创建解决方案。

本书采用“提出问题—设计方案—解决方案”方式，每一章都关注一个单独的场景，提出要解决的问题，设计解决方案，最后实现问题的解决方案。

在开发整个解决方案的过程中，认真研究了每一个业务需求，并使用asp.net 3.5及visual studio 2008提供的最新功能构建了一个可以重用的框架。

本书包含的大量代码为您提供了一个良好的学习基础，您可以扩展这些代码，以实现公司特定的业务需求。

主要内容 展示了.net framework 3.5中的新功能，如lambda表达式、扩展方法、部分方法、linq、asp.net ajax控件 在数据访问层中使用linq to sql的设计模式 数据输入、审计、工作流及事件通知功能的最佳实践 如何开发和自定义服务器端控件 如何用asp.net web部件控件创建仪表盘 实现了microsoft exception handling application block 读者对象 本书适合负责开发企业级应用程序、部门级应用程序、门户网站或者各类业务应用程序的开发人员与系统架构师

具有visualstudio、.netframework、asp.net和c#的使用经验对阅读本书大有裨益，但这并非必要条件。

## 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。

## 书籍目录

第1章 企业应用程序框架 1.1 提出问题 1.2 设计方案 1.2.1 第2章：数据访问层 1.2.2 第3章：业务逻辑层 1.2.3 第4章：用户界面层 1.2.4 第5章：异常处理 1.2.5 第6章：基于角色的安全体系 1.2.6 第7章：工作流引擎 1.2.7 第8章：通知 1.2.8 第9章：报表 1.2.9 第10章：查询生成器控件 1.2.10 第11章：仪表盘 1.2.11 第12章：追踪审计 1.2.12 第13章：代码生成器 1.3 解决方案 1.4 小结 第2章 数据访问层 2.1 提出问题 2.2 设计方案 2.2.1 ado.net连接 2.2.2 动作查询 2.2.3 行返回查询 2.2.4 标量查询 2.2.5 sqlhelper 2.2.6 在bll与dal之间传递数据 2.2.7 ltnq to sql 2.2.8 创建datacontext 2.2.9 添加记录 2.2.10 更新记录 2.2.11 删除记录 2.2.12 选择记录 2.2.13 存储过程 2.3 解决方案 2.4 小结 第3章 业务逻辑层 3.1 提出问题 3.2 设计方案 3.2.1 entbasebo 3.2.2 entbasebolist 3.2.3 entbaseeo 3.2.4 entbaseeolist 3.3 解决方案 3.3.1 首个编辑对象类 3.3.2 首个编辑对象列表对象 3.4 小结 第4章 用户界面层 4.1 提出问题 4.2 设计方案 4.2.1 paidtimeoff样式表 4.2.2 父母版页 4.2.3 表格编辑页面的母版页 4.2.4 编辑页面的母版页 4.3 解决方案 4.3.1 administration页面 4.3.2 用户列表页面 4.3.3 用户编辑页面 4.4 小结 第5章 异常处理 5.1 提出问题 5.2 设计方案 5.2.1 应用程序级的异常处理 5.2.2 页面级的异常处理 5.2.3 方法级的异常处理 5.2.4 web.config 5.2.5 运行状况监视 5.2.6 enterprise application blocks 5.3 解决方案 5.4 小结 第6章 基于角色的安全体系 6.1 提出问题 6.2 设计方案 6.3 解决方案 6.3.1 实体对象与datacontext 6.3.2 数据类 6.3.3 业务类 6.3.4 用户界面 6.4 小结 第7章 工作流引擎 7.1 提出问题 7.2 设计方案 7.2.1 工作流数据表设计 7.2.2 存储过程 7.2.3 实体对象与datacontext 7.2.4 业务类 7.2.5 用户界面 7.2.6 transitions 7.3 解决方案 7.3.1 pto表的设计 7.3.2 实体对象与datacontext对象 7.3.3 业务类 7.3.4 用户界面 7.4 小结 第8章 通知 8.1 提出问题 8.2 设计方案 8.2.1 smtpclient类 8.2.2 设计数据库表 8.3 解决方案 8.3.1 定义通知服务所用到的数据表 8.3.2 用于e-mail通知的业务对象 8.3.3 创建通知注册页面 8.4 小结 第9章 报表 9.1 提出问题 9.2 设计方案 9.2.1 直接连接到数据库来生成报表 9.2.2 报表查看器 9.2.3 基于三层结构的报表 9.3 解决方案 9.4 小结 第10章 查询生成器控件 10.1 提出问题 10.2 设计方案 10.2.1 entbasequerydata 10.2.2 entbasequerybo 10.2.3 自定义查询生成器控件 10.3 解决方案 10.4 小结 第11章 仪表盘 11.1 提出问题 11.2 设计方案 11.3 解决方案 11.4 小结 第12章 追踪审计 12.1 提出问题 12.2 设计方案 12.2.1 自定义追踪审计 12.2.2 创建用户界面 12.3 解决方案 12.4 小结 第13章 代码生成器 13.1 提出问题 13.2 设计方案 13.3 解决方案 13.4 小结

## 章节摘录

插图：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VisualStudio2008进行开发，基于ASRNET 3.5，采用C#进行编程，后台数据库为SQLServer2005。

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？

## <<ASP.NET 3.5商用开发架构>>

### 编辑推荐

《ASP.NET 3.5商用开发架构精解》由清华大学出版社出版。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>