

用经典案例学通.NET 技术

ASP.NET 企业级开发案例精解（C#编程篇）

刘庆国 聂元铭 编著

人民邮电出版社

图书在版编目 (CIP) 数据

ASP.NET 企业级开发案例精解. C#编程篇 / 刘庆国, 聂元铭编著.

—北京: 人民邮电出版社, 2006.2

ISBN 7-115-13707-2

I. A... II. ①刘... ②聂... III. ①主页制作—程序设计 ②C 语言—程序设计
IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 008919 号

内 容 提 要

ASP.NET Starter Kit 是微软公司提供的免费下载 ASP.NET 入门指南, 其中包括学习资料和示例解决方案。ASP.NET Starter Kit 中的企业级解决方案功能实用、代码编写规范, 是学习和进行 ASP.NET 开发可借鉴的理想范例。但是它又非常复杂, 在没有指导的情况下, 初学者很难将其读懂并且应用到实际当中。本书从应用的角度出发, 按照功能模块对 ASP.NET Starter Kit 中的简单电子商务系统和项目进度管理系统做了详尽的解析, 并讲述了其中的设计思想和开发技巧。本书的范例采用的编程语言是 C#。

本书从实用的角度出发, 结合 ASP.NET Starter Kit 中范例讲解 ASP.NET 技术, 适合正在从事和希望学习 ASP.NET 开发的人员阅读。

用经典案例学通 .NET 技术

ASP.NET 企业级开发案例精解 (C#编程篇)

◆ 编 著 刘庆国 聂元铭

责任编辑 屈艳莲

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京密云春雷印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 22.25

字数: 670 千字

印数: 1—5 000 册

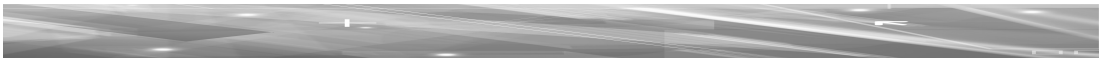
2006 年 2 月第 1 版

2006 年 2 月北京第 1 次印刷

ISBN 7-115-13707-2/TP · 4827

定价: 38.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223



前言

ASP.NET Starter Kit 是一组开放源代码的 ASP.NET Web 应用程序,ASP.NET Starter Kit 的主要目的是为 ASP.NET 的初学者提供一个学习 ASP.NET 开发的示例性“教科书”。无论学习哪一门开发语言,最有效的方式就是在学习过简单的语法之后,研读示例程序。只有通过对实际的示例研究和分析,才能学到实用的开发技巧。ASP.NET 的初学者很幸运有 ASP.NET Starter Kit 这样一组优秀的示例作为学习 ASP.NET 的突破点,ASP.NET Starter Kit 的每一个示例无论在结构设计还是在代码实现方面都堪称经典,它有灵活的设计模式、严谨的编程风格、精练的代码语句,值得初学者反复研究。

本书将以 ASP.NET Starter Kit 中的两个示例应用程序“Commerce Starter Kit——电子商务系统”和“Time Tracker Starter Kit——项目与工作管理系统”为基础,讲解 ASP.NET 开发的技术要点、开发思路以及编程技巧。

Commerce Starter Kit 是一个电子商务应用程序,实现了电子商务应用程序中关键的部分,对准备使用 ASP.NET 开发电子商务应用程序的开发者有很大的帮助。

Time Tracker Starter Kit 是一个项目管理和工作进程管理的示例程序,该示例以企业或组织结构内部的项目和工作管理为主要应用背景,对准备使用 ASP.NET 开发企业信息化的开发者有很大帮助。

编写本书的目的是希望读者既学习到两个示例中具体的开发技巧,也学习到 ASP.NET 的开发中的主要技术点以及针对不同项目的设计理念,希望读者通过对本书中两个示例的分析和研究,形成一套属于自己的 ASP.NET 开发方法。

本书的主要作者是刘庆国。同时,要感谢为本书付出辛勤劳动的技术人员。首先感谢参与本书校对工作的刘哲,他认真细致的修改保证了本书的质量。其次要感谢对本书部分章节提出过意见的张英男。还要感谢我的同事刘英鹏、谢丽玲、陈建新等人给予我的支持。最后要特别感谢我的合作伙伴聂元铭。

由于时间仓促加之水平有限,书中错漏之处在所难免,敬请读者批评指正。本书作者的联系方式是 wfconquer@163.com,欢迎来信交流。读者也可以通过发送 E-mail 至 quyanlian2@ptpress.com.cn 与本书责任编辑交流。

刘庆国
2006 年 2 月

	前 言	1
--	-----	---



目 录

第一部分 Commerce Starter Kit——电子商务系统

第 1 章	Commerce 简介	3
1.1	Commerce 的概况	3
1.1.1	Commerce 的开发背景	3
1.1.2	Commerce 的版本	4
1.2	Commerce 的安装	5
1.2.1	下载安装 Commerce	5
1.2.2	配置运行 Commerce	6
1.3	开发启示	6
1.3.1	Commerce 技术要点概述	6
1.3.2	Commerce 开发启示	6
第 2 章	需求分析与系统架构	8
2.1	系统功能	8
2.1.1	需求分析	8
2.1.2	系统功能	10
2.2	UML 图	11
2.2.1	用例图	11
2.2.2	类图	11
2.2.3	活动图	13
2.3	系统架构	13
2.4	数据库设计	14
2.5	设计启示	14
第 3 章	技术点解析	16
3.1	服务器控件	16
3.1.1	服务器控件的特点	16
3.1.2	服务器控件的简单应用	17
3.1.3	ASP.NET 可利用的服务器控件	18
3.2	多层结构	18
3.2.1	多层结构的概念	18
3.2.2	ASP.NET 中的多层结构	19
3.3	配置 Web.config	20
3.3.1	Web.config 的功能和特点	20
3.3.2	配置 Web.config	21
3.4	Web Service	24
3.4.1	Web Service 简述	24
3.4.2	构建 ASP.NET 的 Web Service	25
3.4.3	在 ASP.NET 中调用 Web Service	27
3.5	技术点总结	28
第 4 章	系统设计	29
4.1	页面样式基础	29

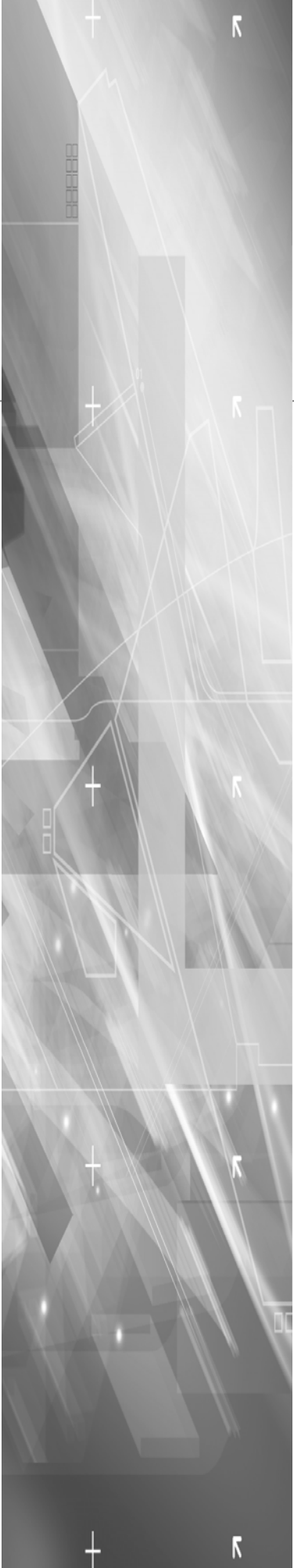
用经典案例学通.NET 技术		ASP.NET 企业级开发案例精解（C#编程篇）
4.1.1	使用级联样式表文件·····	29
4.1.2	ASPNETCommerce.css·····	30
4.2	Global.asax·····	31
4.2.1	Global.asax 的结构和作用·····	31
4.2.2	使用 Global.asax·····	32
4.3	用户控件·····	33
4.3.1	_Header.ascx·····	33
4.3.2	_Menu.ascx·····	35
4.3.3	在页面中调用用户控件·····	38
4.4	小结·····	38
第 5 章	用户体系·····	39
5.1	系统设计·····	39
5.1.1	需求分析·····	39
5.1.2	功能设计·····	40
5.1.3	数据库设计·····	41
5.2	用户注册·····	41
5.2.1	实现效果·····	41
5.2.2	关键技术点·····	43
5.2.3	存储过程·····	43
5.2.4	数据访问层·····	44
5.2.5	业务逻辑层·····	45
5.2.6	用户表示层·····	48
5.3	用户登录·····	49
5.3.1	实现效果·····	49
5.3.2	关键技术点·····	50
5.3.3	存储过程·····	50
5.3.4	数据访问层·····	51
5.3.5	业务逻辑层·····	53
5.3.6	用户表示层·····	56
5.4	开发启示·····	56
第 6 章	Web 商店·····	58
6.1	系统设计·····	58
6.1.1	需求分析·····	58
6.1.2	功能设计·····	59
6.1.3	数据库设计·····	60
6.2	关键技术点·····	62
6.2.1	ASP.NET 中的数据访问技术·····	62
6.2.2	ASP.NET 中的数据展现技术·····	67
6.3	商品浏览·····	69
6.3.1	实现效果·····	69
6.3.2	存储过程·····	70
6.3.3	数据访问层·····	73
6.3.4	业务逻辑层·····	77
2	目 录	

6.3.5	用户表示层	80
6.4	购物车	83
6.4.1	实现效果	83
6.4.2	存储过程	84
6.4.3	数据访问层	89
6.4.4	业务逻辑层	96
6.4.5	用户表示层	102
6.5	商品评论	103
6.5.1	实现效果	103
6.5.2	存储过程	104
6.5.3	数据访问层	105
6.5.4	业务逻辑层	107
6.5.5	用户表示层	109
6.6	开发启示	110
第 7 章	购物订单	112
7.1	系统设计	112
7.1.1	需求分析	112
7.1.2	功能设计	113
7.1.3	数据库设计	113
7.2	购物订单	113
7.2.1	实现效果	113
7.2.2	关键技术点	114
7.2.3	存储过程	116
7.2.4	数据访问层	118
7.2.5	业务逻辑层	122
7.2.6	用户表示层	123
7.3	开发启示	124
第 8 章	构建 Web Service	125
8.1	系统设计	125
8.1.1	需求分析	125
8.1.2	功能设计	126
8.2	实现 Web Service	127
8.2.1	实现效果	127
8.2.2	实现细节	130
8.2.3	第三方调用 Web Service	132
8.3	开发启示	133
第 9 章	扩展 Commerce	134
9.1	订单管理	134
9.1.1	需求分析	134
9.1.2	数据层	136
9.1.3	逻辑层	140
9.1.4	表示层	142
9.2	用户数据	147

用经典案例学通.NET 技术		ASP.NET 企业级开发案例精解 (C#编程篇)
9.2.1	需求分析	147
9.2.2	数据层	149
9.2.3	逻辑层	150
9.2.4	表示层	151
9.3	扩展 Web Service	156
9.3.1	需求分析	157
9.3.2	数据层	157
9.3.3	逻辑层	159
9.4	第三方调用	164
9.4.1	需求分析	164
9.4.2	开发 InvokeCommerce	164
9.5	开发启示	178
第二部分 Time Tracker Starter Kit——项目与工作管理系统		
第 10 章	TimeTracker 简介	181
10.1	TimeTracker 概况	181
10.1.1	TimeTracker 的开发背景	181
10.1.2	系统功能	182
10.1.3	版本信息	182
10.2	TimeTracker 的安装	183
10.2.1	推荐环境	183
10.2.2	安装步骤	183
10.2.3	配置说明	188
10.3	技术点概述	189
第 11 章	需求分析与系统架构	190
11.1	系统功能	190
11.1.1	需求分析	190
11.1.2	功能设计	191
11.2	UML 图	192
11.2.1	类图	192
11.2.2	活动图	194
11.3	系统架构	194
11.4	数据库设计	194
11.5	设计启示	196
第 12 章	技术点解析	197
12.1	DataGrid 的高级应用	197
12.1.1	DataGrid 的事件	197
12.1.2	常用的 DataGrid 事件	198
12.2	对象集合类的使用	200
12.3	GDI+技术	202
12.3.1	GDI+简介	202
12.3.2	GDI+技术	202
12.3.3	GDI+技术简单应用	203
4	目 录	

12.4	移动 Web 应用	205
12.4.1	移动 Web 应用基础知识	205
12.4.2	ASP.NET 移动 Web 应用程序	205
12.5	技术点总结	205
第 13 章	系统底层开发	207
13.1	页面样式	207
13.2	Web.config	207
13.2.1	系统级常量的定义	207
13.2.2	用户身份验证	208
13.3	Global.asax	210
13.4	用户控件 Banner	211
13.4.1	逻辑层	212
13.4.2	后台编码类	212
13.4.3	前台页面	214
13.5	用户控件 AdminTabs	214
13.5.1	后台编码类	215
13.5.2	前台页面	216
13.6	开发启示	216
第 14 章	用户体系	217
14.1	系统设计	217
14.1.1	功能设计	217
14.1.2	数据结构	219
14.2	关键技术点	220
14.2.1	ASP.NET 运行时用户信息的存储	220
14.2.2	集合类的使用	220
14.3	用户体系的实现	220
14.3.1	数据库设计	220
14.3.2	逻辑层	224
14.3.3	应用层	238
14.3.4	表示层	247
14.4	开发启示	249
第 15 章	项目管理与工作进程管理	250
15.1	系统设计	250
15.1.1	功能设计	250
15.1.2	数据结构	252
15.2	关键技术点	253
15.2.1	DataGrid 的高级应用	253
15.2.2	使用 ViewState 存储页面内部变量	254
15.3	项目管理	254
15.3.1	数据库设计	254
15.3.2	逻辑层	265
15.3.3	应用层	277
15.3.4	表示层	288

用经典案例学通.NET 技术		ASP.NET 企业级开发案例精解（C#编程篇）
15.4	工作进程管理	290
15.4.1	数据库设计	290
15.4.2	逻辑层	292
15.4.3	应用层	295
15.4.4	表示层	301
15.5	开发启示	303
第 16 章	数据报表	304
16.1	系统设计	304
16.1.1	功能设计	304
16.1.2	数据库设计	306
16.2	页面形式的报表	309
16.2.1	逻辑层	309
16.2.2	应用层	314
16.2.3	表示层	318
16.3	图片形式的报表	324
16.3.1	应用层	324
16.3.2	逻辑层	325
16.4	开发启示	331
第 17 章	移动 Web 应用	332
17.1	系统设计	332
17.2	应用详解	333
17.2.1	用户登录	333
17.2.2	管理工作进程	336
17.3	开发启示	345



第一部分

Commerce Starter Kit——电子商务系统

案例简介

Commerce Starter Kit 是一个简单的电子商务应用程序，其中包括了电子商务应用程序中最主要的模块。作为一个教程式的应用程序范例，Commerce Starter Kit 展示了电子商务应用程序的基本设计思路和实现方法，开发者不仅可以把它作为一个学习的示例，也可以将其作为电子商务应用程序的开发基础，根据实际情况对其进行扩展和补充。Commerce Starter Kit 中的技术点相对来讲难度较低，但是其设计思路和代码风格对初学者很具有指导意义。

知识点提要

Commerce Starter Kit 的重点知识点如下。

- 服务器控件
- 多层结构
- Web.Config 的配置
- Web Service

资源下载

读者可到微软 ASP.NET 官方网站下载源代码，地址是：

<http://www.asp.net/StarterKits/DownloadCommerce.aspx?tabindex=0&tabid=1>

第 1 章

Commerce 简介

1.1 Commerce 的概况

ASP.NET Commerce Starter Kit 是 ASP.NET Starter Kit 中提供的一套虚拟网络商店模型 Web 应用程序,浏览者和系统用户可以浏览包括商品名称、图片、简介和价格在内的商品信息,因为这是一套虚拟的示例应用程序,所以这些被展示的商品和商品的信息都是虚构的。浏览者和系统用户可以根据需要把想要购买的商品放入购物车,从而完成一笔订单。用户还可以对自己感兴趣但又暂时不想购买的商品进行收藏和跟踪。

1.1.1 Commerce 的开发背景

ASP.NET Commerce Starter Kit 所提供的虚拟网络商店模型是一个展示 ASP.NET 在电子商务应用程序开发方面的模型与示例。在实际使用当中还要添加一些功能,比如说商品的添加、订单支付方面的管理等。运行界面如图 1-1 所示。

以下是对 ASP.NET Commerce Starter Kit 所涵盖功能的简单概述。

- 商品浏览。未注册的浏览者和系统用户都可以浏览商品的列表,用户可以通过商品的分类和快速搜索两种方式检索到商品。用户可以通过商品列表进入商品详细信息页面,商品的详细信息包括商品名称、商品价格、商品详细说明、商品型号、商品分类、商品的图片以及购买此商品的用户通常还购买过什么商品这些信息。由于 ASP.NET Commerce Starter Kit 只作为一个示例程序,因此并没有商品信息管理的功能模块。
- 购物车。当系统的用户决定购买某个商品的时候,就会将该商品放入自己的购物车里面,这个过程和现实购物中的程序是一样的。当用户完成一次购物的时候,会把目前购

购物车内的所有商品作为这条订单的商品进行结账。当然，在用户结账之前可以对购物车内的商品进行任意的修改。

- 用户信息。普通的浏览者是不能通过本系统订购商品的，因为系统并不能得到普通浏览者的相关信息。只有系统的注册用户在登录之后才能订购商品，普通的浏览者可以通过系统提供的注册程序成为系统用户。由于 ASP.NET Commerce Starter Kit 只作为一个示例程序，因此并没有提供相应的用户管理的功能模块。

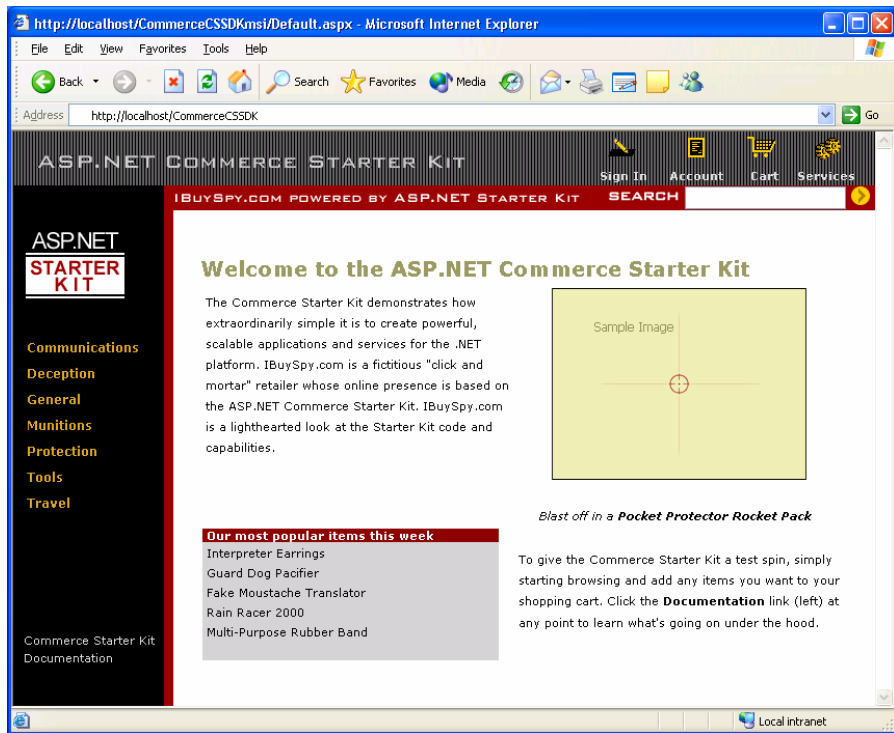


图 1-1 ASP.NET Commerce Starter Kit 运行界面

- 订单管理。ASP.NET Commerce Starter Kit 提供的订单管理的功能主要是用户可以对目前的订单和历史订单进行查看，包括每一条订单的总体情况和详细信息。但是 ASP.NET Commerce Starter Kit 并没有提供订单处理的功能模块。
- Web Service。为了实现系统和第三方应用程序的通信，ASP.NET Commerce Starter Kit 提供了一个 Web Service，该 Web Service 提供订购一笔订单和查看订单状态两个方法。通过这两个方法，第三方应用程序可以很方便地实现与系统的通信。

由于 ASP.NET Commerce Starter Kit 仅仅是提供了一个简单的电子商务模型，设计者在构建应用程序的时候充分考虑到基于现有的程序进行其他功能模块和现有模块的二次开发，在系统的设计上留下了可供扩展的接口，从而使得以 ASP.NET Commerce Starter Kit 为基础迅速开发出一套完整强大的电子商务应用程序变得相对简单。

1.1.2 Commerce 的版本

ASP.NET Commerce Starter Kit 一共有 6 个版本，如表 1-1 所示。

4	第 1 章	Commerce 简介
---	-------	-------------

表 1-1 ASP.NET Commerce Starter Kit 的不同版本信息

版 本 号	使 用 语 言	推荐 Web 服务器	备 注
Commerce Starter Kit (VB SDK)	VB.NET	Web Matrix	
Commerce Starter Kit (CS SDK)	C#	Web Matrix	
Commerce Starter Kit (JS SDK)	J#	Web Matrix	需要 .NET Framework 1.1 支持
Commerce Starter Kit (VB VS)	VB.NET	IIS 6.0	
Commerce Starter Kit (CS VS)	C#	IIS 6.0	
Commerce Starter Kit (JS VS)	J#	IIS 6.0	需要 .NET Framework 1.1 支持

本书讲解的是 Commerce Starter Kit (CS VS)版本。后面章节提到 ASP.NET Commerce Starter Kit 如不做特殊说明，均为该版本。

1.2 Commerce 的安装

1.2.1 下载安装 Commerce

登录 <http://www.asp.net/Default.aspx?tabindex=8&tabid=47>，这里是 ASP.NET Starter Kit 的主页。在这里可以免费下载到 ASP.NET Commerce Starter Kit 的最新版本。

安装 ASP.NET Commerce Starter Kit 推荐使用如下环境：

- .NET Framework 1.1；
- Internet 信息服务器（Internet Information Server，IIS）；
- Microsoft SQL Server 2000 或 Microsoft SQL Server Desktop Engine 2000。

注意：以上列出的只是推荐使用的开发环境，并不是必需的开发环境。比如对 Web 服务器有特殊要求的话，可以使用 Cassini，因为目前最新版本的 Commerce Starter Kit 已经可以在 Cassini 下运行。

以下是 ASP.NET Commerce Starter Kit 具体的安装步骤。

- （1）运行安装文件。打开“欢迎安装 ASP.NET Commerce Starter Kit”的页面。
- （2）单击 Next 按钮，显示出安装说明和许可协议。然后选择是否将程序运行在 IIS 下，如果是，直
接单击 Next 按钮。接下来选择应用程序安装在本机还是一个远程主机，如果选择安装在远程主机，那么
必须有远程主机的管理员权限。
- （3）选择程序的安装路径和程序的运行用户身份。在选择程序运行身份时，可选择本应用程序仅供安
装用户运行，还可以选择当前服务器上所有用户都具有运行权限。
- （4）程序开始自动进行安装，在应用程序安装完毕之后，会自动开始数据库的安装。在选择安装目标
数据库之后必须测试数据库连接，如图 1-2 所示，测试成功之后才能进行数据库的安装。
- （5）当数据库安装完毕之后，就会出现 ASP.NET Commerce Starter Kit 安装成功的画面。

以上讲述的是利用 ASP.NET Commerce Starter Kit 的安装程序进行安装，也可以通过手工部署应用
程序和导入数据库的方式安装 ASP.NET Commerce Starter Kit。在使用安装程序安装成功之后，在 ASP.NET
Commerce Starter Kit 的安装目录下的 Setup 目录中提供了导入数据库的 SQL 脚本。在 SQL Server 的查
询分析器中依次运行 Commerce_CreateDB.sql、Commerce_GrantPermissions_ForLocal.sql（如果选择数
据库服务器在远程主机，那么此处应运行 Commerce_GrantPermissions_ForRemote.sql）、
Commerce_LoadData.sql。

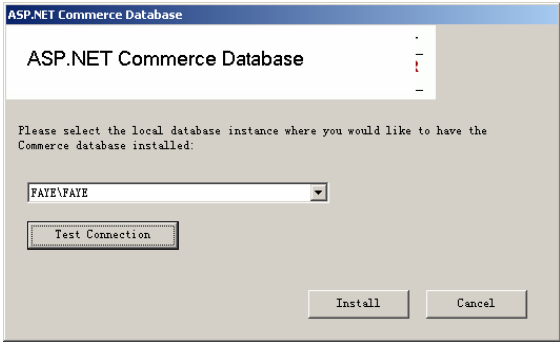


图 1-2 测试数据库连接页面

1.2.2 配置运行 Commerce

安装结束之后，ASP.NET Commerce Starter Kit 就可以正常地运行了。如果没有选择使用安装程序安装或是更改目前应用程序的运行环境，那么就需要更改 ASP.NET Commerce Starter Kit 的配置文件。

更改数据库连接应配置 ASP.NET Commerce Starter Kit 的 Web.config 文件，查找名为 ConnectionString 的配置字符串。假如数据库服务器名称为 CommerceServer，数据库名称为 Commerce，数据库的登录名为 sa，密码为空，那么更改 Web.config 文件中的 ConnectionString 为如下格式：

```
<configuration>
<appSettings>
<add key="ConnectionString" value="server=CommerceServer;database=Commerce;uid=sa;pwd=" />
.....
<appSettings>
.....
</configuration>
```

1.3 开发启示

1.3.1 Commerce 技术要点概述

ASP.NET Commerce Starter Kit 主要的技术要点如下：

- 服务器控件的使用，前台页面的高度 HTML 化；
- 使用页面高速缓存来提高性能；
- ADO.NET 三层架构的使用；
- 基于表单的用户身份认证；
- Web Service 的应用。

以后的章节将对这些技术要点在 ASP.NET Commerce Starter Kit 中的应用进行详细讲解。

1.3.2 Commerce 开发启示

ASP.NET Commerce Starter Kit 是一个简单的电子商务 Web 应用程序，其中包括商品浏览、购物车

和订单结算等电子商务应用程序中必不可少的模块。ASP.NET Commerce Starter Kit 的开发目的有以下两个。

（1）为学习 ASP.NET 的初学者提供一个教程式的经典案例

学习一门语言最快和最有效的途径莫过于对经典的开发案例进行反复的研究和揣摩，ASP.NET Starter Kit 的目的正是为 ASP.NET 的初学者提供一组教程式的案例，其中每一个案例都有具体的应用背景，而每一个案例给学习者的启示都是不同的。ASP.NET Commerce Starter Kit 作为 ASP.NET Starter Kit 中的一个案例，其技术难度并不是很高，设计框架也不是很复杂，尤其适用于 ASP.NET 的初学者的入门。在技术方面，ASP.NET Commerce Starter Kit 的设计者力求使用最简单实用的技术实现电子商务 Web 应用程序的设计，这种技术的实现思路和电子商务应用的背景有很大关系，因为在多数电子商务应用程序中，性能和程序的稳定性是被人们首要考虑的，繁琐的技术实现手段无论对于应用程序的性能还是稳定性都是有百害而无一利。在设计方面，ASP.NET Commerce Starter Kit 本着简洁通用的特点，对多数电子商务应用中的对象和模块进行了抽取和提炼，将其应用到 ASP.NET Commerce Starter Kit 的设计当中去。

（2）为使用 ASP.NET 开发电子商务 Web 应用程序提供一个基础框架

ASP.NET Commerce Starter Kit 实际上只实现了电子商务应用程序中最关键的一部分模块，而诸如商品管理、订单管理、配送管理以及财务统计等方面的模块，ASP.NET Commerce Starter Kit 并没有涉及。而在现有的应用程序中，商品对象以及用户对象也仅仅实现了该对象最基本的属性和方法。是 ASP.NET Commerce Starter Kit 的设计者粗心大意导致的么？实际上并不是这样。ASP.NET Commerce Starter Kit 的设计者的另外一个目的就是将 ASP.NET Commerce Starter Kit 设计成为一个电子商务 Web 应用程序的基础框架，其他的开发者可以基于这个框架，根据实际情况开发出属于可以被应用的电子商务 Web 应用程序。正是基于这个目的，ASP.NET Commerce Starter Kit 的设计者才没有实现电子商务应用中全部的模块，而 ASP.NET Commerce Starter Kit 中的主要对象也只具有基本的属性。这样可以留给开发者更广阔的空间，以 ASP.NET Commerce Starter Kit 为基础开发出功能完备、特点鲜明的电子商务应用程序。

对于学习一门新的程序设计语言来说，往往最需要学习的是如何实现常见的功能模块，例如在 ASP.NET 的学习中页面元素以及列表的展现，数据库的连接与操作，用户机制的实现等都是初学者所关心的，另外在新的程序设计语言中，新的概念以及思想也是初学者所关心的。在 ASP.NET Commerce Starter Kit 中，初学者可以找到这些问题的答案，对于 ASP.NET Web 应用程序中页面元素以及列表的展现，数据库的连接与操作，用户机制的实现等关键问题，ASP.NET Commerce Starter Kit 都有涉及，并且其实现方法和思路都十分经典。另外 ASP.NET Commerce Starter Kit 中使用了 Web Service 技术，为初学者提供了一个学习 Web Service 技术基础的范例。

初学者通过对 ASP.NET Commerce Starter Kit 的学习，不仅可以掌握 ASP.NET 开发的具体方法，也可以学习到如何利用面向对象思想分析和抽取应用程序中的对象，以及根据对象模型设计类库。在实际的应用程序开发中，开发者对于系统中对象的把握和设计的类库，犹如万丈高楼的根基直接决定系统的成功与否，因此在面向对象设计中科学的方法论是非常重要的。希望读者能够通过对本书的阅读以及对 ASP.NET Commerce Starter Kit 的学习真正做到举一反三，在其他 Web 应用程序的开发过程中应用科学的面向对象设计方法。

第 2 章

需求分析 与系统架构

ASP.NET Commerce Starter Kit 的设计者从根本上是想把项目设计成为一个模型，所以在系统设计方面，考虑的更多是通用性和可扩展性。通过对该项目的学习和研究，开发者能够深刻地理解 ASP.NET 的开发模式和开发技巧，同时 ASP.NET Commerce Starter Kit 也可以作为一个电子商务应用的基础模

型进行二次开发和扩展。本章中将从需求分析和系统架构方面分析 ASP.NET Commerce Starter Kit。

2.1 系统功能

2.1.1 需求分析

软件工程中包含需求、设计、编码和测试 4 个阶段，其中需求分析是软件工程第一个也是很重要的一个阶段，需求分析的主要任务是绘制关联图、创建开发原型、分析可行性、确定需求优先级、为需求建立模型、编写数据字典、应用质量功能调配。需求分析从总体上看是说明项目应该具有什么样的功能，而不考虑实现这些功能的具体技术。

ASP.NET Commerce Starter Kit 是一个电子商务应用站点，要想提出站点的需求，就需要从站点的目标和用户入手。在实际项目的开发中，需求分析这一阶段是面向商务应用的，往往是客户提出的，但是在 ASP.NET Commerce Starter Kit 的设计中则采用了虚拟需求的模式，因为并没有明确的项目应用背景，所以需求分析都是设计者根据大量的电子商务模型提取和抽象出来的。这些被提取出来的需求具有很强的通用性，电子商务必不可少的功能和多数电子商务都需要具备的一些基本功能成为 ASP.NET Commerce Starter Kit 的需求，例如用户管理和生成订单这样的功能模块。

除了上述实际的应用需求之外，还要考虑到项目开发和项目设计中的需求，比如开发人员的协作、应用程序的整体性能、系统的可扩展性和可维护性等。特别是 ASP.NET Commerce Starter Kit 提供的是一个可以供二次开发的应用模型，所以对系统的通用性和可扩展性要求就更高了。

系统的需求分为物理需求、结构需求、逻辑需求。

物理需求的任务很明确,就是确定 Web 站点的物理服务器的最终架构和软硬件环境。例如应用程序是否需要分布式部署,数据库服务器和 Web 服务器是否必须集成在同一台服务器上,是否允许第三方应用程序进行远程调用等。根据 ASP.NET Commerce Starter Kit 设计者的初衷,物理需求应包括如下几方面。

(1) 支持可分布式部署的服务器群组

支持分布式的服务器群组是优秀的网络应用程序必须提供的一个物理功能,因为大型的网络应用程序不可能将所有的应用和操作运行于同一台服务器。支持分布式的服务器群组有利于降低服务器负荷,使服务器的功能更加具有针对性。

例如可以将数据库服务器和 Web 服务器脱离开,这样不仅能够提高系统的性能,而且也便于管理。

(2) 支持 .NET 的服务器操作平台

这是一个必须要满足的需求。ASP.NET 应用程序不可能脱离 .NET Framework 的支持,因此 Web 服务器必须支持 .NET。

(3) 支持第三方应用程序的远程调用

作为系统扩展的接口,支持第三方应用程序的远程调用是必须的。第三方应用程序可以通过远程调用的形式完成和 ASP.NET Commerce Starter Kit 的交互,无形间也将 ASP.NET Commerce Starter Kit 的应用渗透到第三方应用程序中。

(4) 仅限于 Microsoft SQL Server 的数据库管理系统

支持多种数据库类型是一个不错的构想,但是 ASP.NET Commerce Starter Kit 更多地展示的是 ASP.NET 以及 ADO.NET 中数据操作的新特性,而在 ADO.NET 中针对 Microsoft SQL Server 提供了很多具体的对象和方法。为了更多地介绍和展现 ADO.NET 中的对象和方法,ASP.NET Commerce Starter Kit 采用了 Microsoft SQL Server 作为系统的数据库管理系统 (DBMS)。

ASP.NET Commerce Starter Kit 的结构需求也是从设计者的初衷中抽象出来的,根据系统的目标和针对性,可以确定如下的结构需求。

(1) 站点的可维护性和可扩展性强

大多数的 Web 应用程序在实际应用中都需要不断地添加功能模块,ASP.NET Commerce Starter Kit 也是一样,在二次开发和实际应用中要根据项目的具体情况添加一些功能模块。因此项目在设计之初就要考虑到,当前的架构对系统的扩展工作会不会形成障碍。

使用 ASP.NET 中层的设计概念能够增强站点的维护性和扩展性,基于层的设计模式允许开发者以三层甚至多层的模式开发 ASP.NET 应用程序,将数据操作、业务逻辑、前台显示等单元分离开,每一层都有针对性,层是以一组序列分布在系统数据和用户之间的,不相邻的层在业务上没有耦合,每一层都是继承和调用上一层中的对象和方法。这种模式使得站点的功能分布更加合理化。例如扩展一部分业务逻辑,首先是要在业务逻辑层中建立相应的方法,然后才是在前台显示层中建立新的页面控件。关于层的详细应用,第 3 章将做详细的讲解。

(2) 站点的功能模块通用性强

由于 ASP.NET Commerce Starter Kit 是作为一个示例和应用程序框架被设计和开发的,因此其功能模块要具有较高的通用性,以保证功能模块中的业务逻辑、数据模型从多数的电子商务应用程序中提取。简单地说,ASP.NET Commerce Starter Kit 需要提供电子商务中最基本的对象和这些对象最基本的属性,只有这样才能使基于 ASP.NET Commerce Starter Kit 的二次开发具有更大的扩展性。例如商品信息只列出最基本的商品信息,至于一些具体应用中商品的特殊属性,并不应该出现在 ASP.NET Commerce Starter Kit 中。

模块化的构建同时也意味着模块之间尽量降低耦合度,这样做的好处是使得更改模块内部或新增其他模块对系统的稳定性影响不大。

物理需求和结构需求都是从设计框架上对系统提出了宏观的要求,而逻辑需求则是在实际项目的需求分析阶段对系统的业务逻辑提出的要求。以下是 ASP.NET Commerce Starter Kit 的逻辑需求。