

- ◆ 兼顾基础，重在提高与技巧
- ◆ 专家解惑答疑，让你无师自通



ASP.NET 3.5

系统开发精髓

李彦 唐鑫 唐继强 崔英志 等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

代码的力量——

ASP.NET 3.5 系统开发精髓

李彦 唐鑫 唐继强 崔英志 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

对于 ASP.NET 3.5 初学者和 PHP/ASP/Java 平台的 Web 编程人员而言,在进一步学习 ASP.NET 3.5 的过程中,最希望能够有这样一本书:既能够对深入学习 ASP.NET 3.5 起到“提纲挈领”式的向导作用,又能够由点到线、轻重结合地系统讲解关键知识点;既能够了解 ASP.NET 3.5 系统开发的总体架构,又能够通过完整案例的精细讲解,理解到具体开发过程中的经验和技巧。本书正是为此而作。

全书分为三大部分:第一部分是基础整合篇,讲解了学习 ASP.NET 3.5 必备的基础知识;第二部分是应用提高篇,详细讲解了 Web.config 配置、HttpModule 和 HttpHandle、统一认证、单点登录、成员资格(MemberShip)、权限角色、WebPart、母版、Ajax 等 ASP.NET 3.5 高级技术。在讲解过程中均辅之以实例练习。第三部分是系统开发篇,在详细阐述了数据库访问层、数据表封装层、业务逻辑层、LINQ & LINQ to SQL 的基础之上,通过两个完整实例的精细化讲解,让读者彻底掌握 ASP.NET 3.5 系统开发的经验与技巧。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ASP.NET 3.5 系统开发精髓 / 李彦等编著. —北京:电子工业出版社, 2009.5

(代码的力量)

ISBN 978-7-121-08572-7

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

中国版本图书馆 CIP 数据核字(2009)第 044873 号

责任编辑: 祁玉芹 刘娴庆

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 28 字数: 717 千字

印 次: 2009 年 5 月第 1 次印刷

定 价: 45.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

为什么要写这本书？

对于许多初学者而言，学习 ASP.NET 的感觉是“上手容易精通难”。比如一个典型的 ASP.NET 操作学习过程：非常轻松安装完成 Visual Studio，进入 Visual Studio 后，在“工具箱”中拖入几个服务器控件，很快就知道如何在 aspx.cs 文件中编写控件事件执行的程序；单击“运行”按钮就可以在浏览器中查看 Web 页面的运行效果了，这符合微软公司产品的一贯特点，上手极为容易。但如果要使用 ASP.NET 技术开发一个 Web 系统，就会遇到非常多的问题了。例如，页面与控件的事件执行顺序是谁先谁后？什么时候需要修改 Web.config 配置文件？Web.config 文件中的各配置项有什么作用？为什么页面运行时，首先要执行 Page_Load 方法？如何自定义一个控件？为什么页面提交后，控件的数据会保留？为什么单击按钮提交后，页面会执行 Click 事件中的程序？总之，当初学者需要更进一步掌握 ASP.NET 时，却发现 ASP.NET 像一个巨大的迷宫，到处都是岔道，每一个知识点都不是独立的，都需要以其他知识点为基础。

对于熟悉其他 Web 编程（如 PHP、ASP）技术的程序员来说，在希望学习 ASP.NET 时，这些问题就显得尤为突出，因为 ASP.NET 中对 Web 工作模式进行了“友好”的封装，反而使得习惯于 PHP 或 ASP 编程的程序员对 ASP.NET 的工作模式有似懂非懂的感觉。正是如此，所以经常在网上技术论坛中看到，一些朋友在请求前辈们推荐一本升级学习 ASP.NET 的书。笔者正是基于大量的这种需求，结合自身长期从事项目开发实践与本科教学的优势，有针对性地组织了此书。

本书的特点

如果说 ASP.NET 技术是一张网，各种知识点错综复杂，互为支撑，难以深入学习，那就需要将这张网变成一根线，顺着这根线逐步深入到 ASP.NET 的“腹地”，对 ASP.NET 这个“庞然大物”有一个清晰的认识。一旦对 ASP.NET 有了正确、清晰、全面的认识，学习起来就不再是难事，甚至是游刃有余。

本书正是出于此目的，力图在 ASP.NET 技术这张“网”中归纳出一根学习“路线”，以引导读者走向 ASP.NET 的“腹地”。为了让读者能够有足够的兴趣和激情顺着这根“线”走下去，本书在选取所讲解的知识点时，首先注重的是实用价值，尽量选择在实际系统开发过程中常用的、操作性较强的技术进行讲解，以最大限度地满足本书读者对象的需求，并不过份追求技术点的深度。

除了单个的知识点外，本书的特色之处就是详细展示了分层开发的思想和方法。分层开发是一个人人尽知的概念，但在实际开发过程中如何规划和实现，这是一般的初级开发人员难以触及的知识。尤其是没有参加过大型项目开发的编程人员，对于系统级构架和分层开发

的具体实现方法就更加陌生了。本书不仅详细讲解了系统开发过程中如何分层，而且通过实例，阐述了如何以最直观、最简单的方式实现分层，其目的就是让读者更容易理解分层的概念和实现方法。

本书最后通过两个具有代表性的综合开发案例的实现，让读者进一步加深了对全书知识的认识。读者通过这些综合项目的分析，自然就会举一反三，灵活应用，很快就能够上手进行系统开发。而读者编写的 ASP.NET 程序也会变得更加有血有肉、有思想有灵魂。

本书的阅读对象

本书是专门为以下三类读者量身定做的。

1. 对 ASP.NET 有了一个大致概念认识，或者使用 ASP.NET 编写过简单的系统，但希望进一步捋清 ASP.NET 的技术思路，深入了解 ASP.NET 的工作原理，熟练使用 ASP.NET 构架并开发大型系统的读者。

2. 熟悉 ASP、PHP 或 Java，希望以最高效的方式了解和掌握 ASP.NET 技术思路、工作原理，并能够将自己积累的 Web 开发经验快速地移植到 ASP.NET 开发中的读者。

3. 熟悉 C/C++/Delphi 等语言的客户端编程模式，希望能够在现有的编程经验基础上，尽快了解和掌握 Web 的编程规律、掌握 ASP.NET 技术开发的读者。

无论哪一类读者，在阅读本书之前，都需要对 C# 语言有一个大致的了解。

本书的知识安排

本书的第 1 章至第 3 章是 Web 编程基础和 ASP.NET 基础，本书安排了极少的基础知识供 2、3 类读者快速掌握本书必备的基础知识，第 1 类读者可以选择性地阅读这部分知识。这里包括：如何开始建立一个 ASP.NET 的网站、ASP.NET 的知识结构图、Get/Post 在 ASP.NET 中的表现方式、页面与控件的基本概念、服务器控件的简介、服务器控件的工作原理、ASPX 页面与 Page 类的联系等。

第 4 章至第 8 章是对 ASP.NET 知识的进阶学习，其目的是为了学习 ASP.NET 的工作机制，了解 ASP.NET 一些特性解决方案，通过这些特性解决方案从侧面了解 ASP.NET 的技术特点。具体来说，第 4、5 两章讲解了 config 配置文件的组织方式和从 IIS 开始 ASP.NET 的工作流程与实现机制。第 6 章详细讲解了 ASP.NET 提供的对用户个性化信息存储与操作的解决方案，如统一认证、单点登录、成员资格(Membership)、权限角色等。第 7 章讲解了 ASP.NET 中的界面处理技术，并对工作机制进行了大量分析。第 8 章详细讲解了在 ASP.NET 中使用 Ajax 的技术细节。

第 9 章开始讲解分层系统开发，第 9、10 章讲解的是与数据访问相关的技术；第 11、12 章实现了一个完整的数据库访问层，并为数据表封装层提供基类，有了这两章的基础，便为业务逻辑层、界面逻辑层的实现打下了良好的基础。第 13 章通过精细讲解一个实例，来展示使用 ASP.NET 进行系统开发时的步骤和经验。第 14 章讲解了 LINQ 和 LINQ to SQL，LINQ to SQL 是第 11、12 章所讲技术的一种较为完美的替代方案。第 15 章又通过一个完整的实例来讲解使用 LINQ to SQL 开发系统的经验和架构方式。

在本书的写作过程中，难免有些错漏之处，敬请读者指正，请读者朋友通过 <http://www.WriteBook.com.cn> 网站与笔者交流。

笔者于 2009 年 1 月

目 录

第 1 部分 基础整合篇

第 1 章 ASP.NET 基础	1
1.1 Visual Studio 2008	1
1.1.1 新建一个 ASP.NET 网站	2
1.1.2 解决方案	4
1.1.3 网站与项目	6
1.1.4 编辑区	6
1.2 ASP.NET 知识结构图	8
1.3 C# 语言	9
1.3.1 C# 速览	9
1.3.2 C# 语言的数据类型及运算符	10
1.3.3 流程控制	14
第 2 章 页面、控件与事件	17
2.1 一个简单计算器的实现	17
2.2 Get 和 Post 方法	21
2.2.1 Get 方法	21
2.2.2 Post 方法	22
2.2.3 ViewState	23
2.3 页面的属性和事件	24
2.3.1 Page_Load()	24
2.3.2 页面的常用属性	25
2.4 控件的属性和事件	28
2.4.1 控件事件的定义和执行	28
2.4.2 AutoPostBack 属性	30
2.4.3 控件事件的实现过程分析	31

2.5	ASP.NET 服务器控件	32
2.5.1	服务器控件的工作原理	32
2.5.2	基本服务器控件简介	32
2.6	验证控件	40
2.6.1	RequiredFieldValidator 控件	40
2.6.2	ValidationSummary	43
2.6.3	其他验证控件	44
2.7	Web 用户控件	46
2.7.1	使用方法	47
2.7.2	添加属性、方法和事件	48
第 3 章	页面和自定义控件	53
3.1	Page 类	53
3.1.1	Page 类的作用	54
3.1.2	Page 类的生命周期中程序的执行顺序	54
3.1.3	继承 Page 类	58
3.2	aspx 文件	62
3.2.1	单页面代码模型	62
3.2.2	代码隐藏模型	63
3.2.3	<% %>嵌入式代码块	64
3.3	用户自定义服务器控件	67
3.3.1	建立项目	67
3.3.2	控件的生命周期	68
3.3.3	映射回传事件到服务器端事件	74
3.3.4	处理回传数据	77
第 2 部分 应用提高篇		
第 4 章	web.config 文件	81
4.1	ASP.NET 的配置文件	81
4.2	自定义配置节	84
4.3	常用的配置设置	89
4.4	配置节的加密	94
第 5 章	HTTP 运行时	97
5.1	基本结构	97

5.2	HttpModule.....	98
5.2.1	IHttpModule 接口.....	98
5.2.2	HttpApplication 类.....	99
5.2.3	使用示例.....	99
5.2.4	Global.asax.....	103
5.3	HttpHandle.....	104
5.3.1	IHttpHandle 接口.....	105
5.3.2	<httpModules>配置节.....	105
5.3.3	使用示例.....	106
5.3.4	IHttpHandlerFactory.....	107
第 6 章	用户信息管理.....	111
6.1	ASP.NET 身份验证方式.....	111
6.1.1	Windows 身份验证.....	111
6.1.2	Passport 身份验证.....	113
6.1.3	Forms 身份验证.....	113
6.1.4	<authorization>配置节.....	115
6.2	单点登录 SSO.....	118
6.2.1	子域.....	119
6.2.2	<machineKey>配置节.....	120
6.3	membership 成员资格.....	120
6.3.1	Aspnet_regsql.exe.....	121
6.3.2	<membership>配置节.....	122
6.3.3	登录控件.....	124
6.3.4	自定义 Provider.....	126
6.3.5	配置文件中验证用户.....	128
6.4	Profile 用户配置.....	129
6.4.1	<profile>配置节.....	129
6.4.2	设置用户属性.....	130
6.4.3	匿名用户使用 Profile.....	133
6.5	Role 用户角色.....	134
6.5.1	<roleManager>配置节.....	135
6.5.2	Role 的使用.....	135
第 7 章	界面处理.....	137
7.1	Web Part 技术.....	137
7.1.1	WebPartManager.....	138

89	7.1.2 WebPartZone.....	139
89	7.1.3 Catalog Zone.....	144
99	7.1.4 EditorZone.....	149
99	7.1.5 ConnectionZone.....	154
101	7.2 母版 Master.....	159
101	7.2.1 使用方法.....	160
101	7.2.2 运行机制.....	165
101	7.2.3 引用母版中的对象.....	166
106	7.3 主题 (Theme).....	168
107	7.3.1 CSS.....	168
111	7.3.2 主题与皮肤.....	170

第 8 章 Ajax.....173

111	8.1 工作原理.....	173
113	8.2 Ajax 扩展控件.....	176
113	8.3 ScriptManager.....	177
113	8.3.1 基本属性.....	178
118	8.3.2 调用 Web Service.....	179
119	8.3.3 ScriptManagerProxy.....	182
120	8.4 UpdatePanel.....	183
120	8.4.1 ContentTemplate.....	183
121	8.4.2 Triggers.....	184
123	8.4.3 在母版中使用 UpdatePanel.....	188
124	8.4.4 UpdatePanel 编程注意事项.....	190
126	8.5 UpdateProgress.....	192
128	8.5.1 使用实例.....	192
129	8.5.2 属性.....	193
129	8.6 Timer.....	195

第 3 部分 系统开发篇

第 9 章 数据库访问.....197

137	9.1 分层开发框架.....	197
137	9.2 SQL Server 2005 Express.....	198
137	9.2.1 数据库设置与管理.....	198
138	9.2.2 NorthWind 示范库.....	203

9.3	连接数据库	205
9.3.1	连接字符串	206
9.3.2	执行 SQL 语句	207
9.3.3	连接池	208
9.3.4	事务处理	210
9.3.5	分布式事务处理	211
9.4	DBCommand 类	215
9.4.1	使用参数防止注入漏洞	215
9.4.2	执行存储过程	217
9.5	DataSet 与 DataTable	219
9.5.1	DataColumn 与 DataRow	220
9.5.2	DataGridView	223
9.5.3	DataTable 与 XML 交换数据	224
9.6	DataAdapter	225
第 10 章	GridView 使用技巧	229
10.1	普通列的使用技巧	229
10.2	命令列的使用技巧	233
10.2.1	编辑、更新和取消	233
10.2.2	DataKeys	235
10.2.3	JavaScript 实现选择行	236
10.2.4	“删除”按钮与 JavaScript 确认框	238
10.2.5	ButtonType	239
10.3	按钮列的使用技巧	240
10.3.1	CommandName	241
10.3.2	CommandArgument	241
10.3.3	命令列与按钮列的关系	242
10.4	模板列的使用技巧	243
10.4.1	FindControl	243
10.4.2	利用 Eval 绑定数据	245
10.4.3	利用“表中表”显示主从表数据	246
第 11 章	数据库访问层的实现	251
11.1	系统分析	251
11.1.1	需求分析	251
11.1.2	总体架构设计	252

205	11.2 基本功能的实现.....	253
205	11.2.1 封装 DbCommand 类.....	253
205	11.2.2 支持参数.....	256
205	11.2.3 支持事务.....	258
205	11.2.4 获取 DataTable 和更新数据库.....	260
215	11.3 分页获取数据.....	263
215	11.3.1 分页基本要素.....	263
215	11.3.2 Oracle 的分页技术.....	264
215	11.3.3 SQL Server 的分页技术.....	265
215	11.3.4 分页功能实现.....	267
220	11.4 获取序列值.....	270
223	11.4.1 获取 Oracle 的序列值.....	271
224	11.4.2 实现 SQL Server 的序列功能.....	271
225	11.5 完整代码.....	272
	11.5.1 DBHandler 类.....	272
225	11.5.2 DBHandlerSQLServer 类.....	279
225	11.5.3 DBHandlerOracle 类.....	282
225	11.6 DBHandler 类的实例化.....	284
225	11.6.1 DBHandlerFactory 类.....	284
225	11.6.2 使用示范.....	286
225	11.6.3 分页控件.....	288
225	第 12 章 数据表封装层.....	291
240	12.1 系统分析.....	291
241	12.1.1 需求分析.....	291
241	12.1.2 架构设计.....	292
245	12.2 基类的实现.....	293
245	12.2.1 获取记录.....	293
245	12.2.2 构造方法.....	295
245	12.2.3 实现公有方法.....	295
245	12.2.4 完整代码.....	296
245	12.3 子类的实现.....	299
245	12.3.1 子类的构造方法.....	299
245	12.3.2 “表字段”封装为“类属性”.....	300
245	12.3.3 完整代码.....	302
245	12.4 业务逻辑层.....	305

第 13 章	开发实例——《工作任务协作系统》	307
13.1	需求分析	307
13.2	总体设计	307
13.2.1	系统模块设计	307
13.2.2	数据表设计	308
13.3	App_Code 目录	309
13.3.1	代码组织方式	309
13.3.2	数据表封装代码	310
13.3.3	逻辑层功能实现	319
13.4	界面规划	321
13.4.1	文件规划与 Web.sitemap	321
13.4.2	界面布局与母版	323
13.4.3	页面基类	325
13.5	界面逻辑实现	326
13.5.1	Web 用户控件划分	327
13.5.2	Web 用户控件实现	330
13.6	小结	334
第 14 章	LINQ 和 LINQ to SQL	335
14.1	接触 LINQ	335
14.2	C#3.0 的新语法	336
14.3	LINQ 查询表达式	340
14.3.1	from 子句	340
14.3.2	group 子句	342
14.3.3	orderby 子句	343
14.3.4	into 子句	343
14.3.5	join 子句	344
14.3.6	Let 子句	345
14.4	LINQ 操作方法	345
14.5	LINQ to SQL	348
14.5.1	数据表封装类	349
14.5.2	DataContext 类	353
14.6	查询数据库	356
14.6.1	基本查询	356
14.6.2	Join 方法	358
14.6.3	SelectMany 方法	359

14.6.4	GroupJoin 方法	362
14.6.5	动态查询	363
14.7	操作数据记录	364
14.7.1	创建数据库	364
14.7.2	添加新记录	365
14.7.3	修改和删除记录	366
第 15 章	LINQ 开发实例——《网络项目交易(威客)系统》	367
15.1	需求分析	367
15.2	总体设计	368
15.2.1	模块设计	368
15.2.2	数据表设计	369
15.3	代码组织	372
15.3.1	ORM 层代码组织	373
15.3.2	业务逻辑层(BLL)代码组织	375
15.3.3	BLL 中与用户权限相关的代码	382
15.3.4	界面层中的共用代码	383
15.4	界面规划与实现	383
15.4.1	整体规划	383
15.4.2	页面文件规划	384
15.4.3	Web.sitemap 文件	385
15.4.4	母版设计	387
15.4.5	用户功能菜单	391
15.4.6	页面父类	392
15.5	业务逻辑层功能与实现	393
15.5.1	用户相关功能	394
15.5.2	项目相关功能	402
15.6	界面层实现	407
15.6.1	项目列表模块	407
15.6.2	项目信息与操作模块	414
15.6.3	项目交流帖显示模块	420
15.6.4	用户评价统计显示	423
15.6.5	上传作品缩略图显示	426
15.6.6	验证码图片的生成与使用	430
15.6.7	用户注册界面中使用 Ajax	433
15.6.8	ASPX 页面的实现	435
15.7	小结	436

第 1 部

基础整合篇

第 1 章 ASP.NET 基础

微软公司于 2000 年 6 月向全球宣布了自己的 .NET 战略,作为 .NET 战略最重要的基石——.NET Framework 便不断地更新与发展。2007 年 Visual Studio 2008(本书以后简称为 VS2008)发布,同时集成到 VS2008 中的技术除了 WCF, WPF, WF, WCS 之外,还有 C#3.0, Ajax, Linq 等等;但其核心的 CLR 还是 2.0, C#3.0 经过编译的 IL 代码也完全是基于 CLR2.0,因此在安装 .NET Framework 3.0 和 3.5 时,必须先安装 .NET Framework 2.0。

ASP.NET 的版本号习惯性地跟随 .NET Framework 的版本号变化,因此虽然 CLR 的版本号是 2.0,但仍习惯地将 .NET Framework 3.5 中的 ASP.NET 称之为 ASP.NET 3.5,其准确一点的称呼应该是 ASP.NET Running on Framework 3.5。

1.1 Visual Studio 2008

最好的 ASP.NET 集成开发环境 (IDE) 自然就是微软自家的 Visual Studio。Visual Studio 系列开发工具是微软公司的战略产品,目前最新的版本是 VS2008 SP1。

对于 Web 开发来说,VS2008 相对于 VS2005 在文件组织方式上并没有革命性的变化,主要是引入了对 Ajax, WPF, WCF, Linq 等技术的支持。最近推出的 SP1 包中,主要增加了 Entity Framework 和 Dynamic Data 等功能。

VS2008 的安装过程这里不再赘述,但请读者注意,最好是在安装 VS2008 之前先装好 IIS,否则就需要使用 aspnet_regiis.exe 工具将当前版本的 aspnet_isapi.dll 在 IIS 中注册,并设置好相应资源文件的处理方式。

图 1-1 是 VS2008 的主界面,这个界面几乎是时下流行的 IDE 标准模式。一直以来,业界都知道微软公司的产品不一定是技术最先进的,但一定会是最好用的。对于程序员来说,最好用就代表着最高开发效率、节约时间、增加编程乐趣;.NET 的开发机制配合 Visual Studio 这个开发环境,正是微软公司产品特点的杰出范例。.NET 之所以能够普及得如此迅猛,Visual Studio 系列的 IDE 工具也起到了关键性的作用。

VS2008 的基本使用方法本书不做过多的叙述,请读者参阅其他基础书籍。菜单、工具栏、拖曳式面板都是典型的 Windows 软件风格,相信读者一看就会。在 VS2008 中,通过“解决方案”来统一管理文件,根据应用的不同,在解决方案下又分为“网站”和“项目”两种

类型。下一小节，作者就示范一个网站的建立方法，项目的建立和使用将在后面章节中讲解。

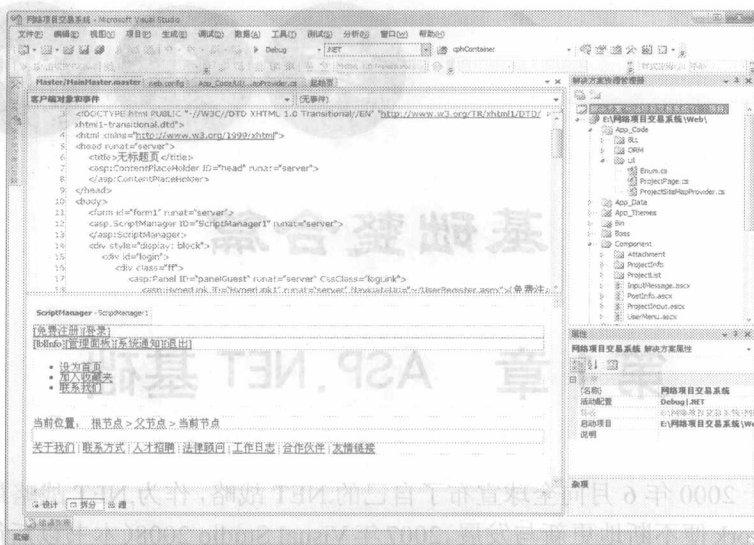


图 1-1 Visual Studio 2008 主界面

1.1.1 新建一个 ASP.NET 网站

例 1-1: 新建一个 ASP.NET 网站，其网站的根目录为“E:\学习 ASP.NET\Calculator”。

步骤 (1) 选择“文件→新建→网站”选项，弹出的对话框如图 1-2 所示。

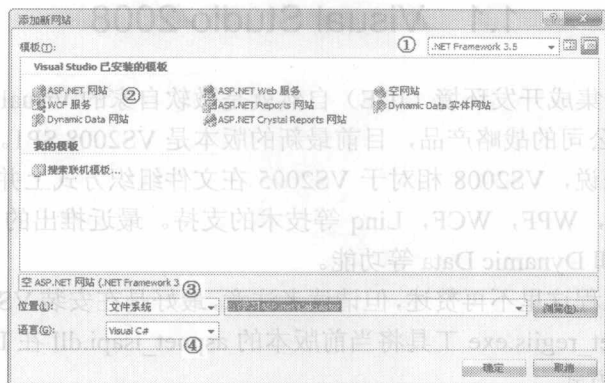


图 1-2 “添加新网站”对话框

如图 1-2 所示，这个对话框有以下 4 点值得注意。

(1) VS2008 中可以选择不同的 .NET framework 版本对项目进行编译，不同的版本所支持的语法和类库是不一样的，但高版本对低版本能够完全兼容。一般情况下选择 3.5 版本即可，提供其他低版本选项的主要目的是为了更方便程序员对旧版本的“解决方案”升级。

(2) 网站类型选择“ASP.NET 网站”，这里所列出来的项目都是 Web 站点类型，只有“WCF”这个类型比较特殊，因为它没有写“ASP.NET”前缀。WCF 是一种远程访问技术，

可以基于 Web 方式提供服务（类似于 WebService），WCF 也可以通过 Console、WinForm、Windows 服务等多种方式提供服务，因此在建立“项目”时也会看到 WCF。

(3) 网站位置选择“文件系统”选项。网站位置是指 ASP.NET 网站的根目录所在的文件路径，有文件系统、HTTP、FTP 三种方式。“文件系统”这种方式就是指在本机找一个目录作为网站的根目录，读者这里要重点注意的是路径中最后那个目录名称，图中所示为“Calculator”，这个名称将会自动作为解决方案的名称，解决方案的含义在稍后将会讲解。

(4) 选择编程语言。对于 ASP.NET 网站，VS2008 默认只提供 C# 和 VB.NET 两种选择。

步骤 (2) 单击“确定”按钮后，VS2008 会自动生成一系列的文件和目录，在 VS 右边的“解决方案资源管理器”中出现如图 1-3 所示的右上方的列表，所对应的本机目录中也生成了相应的文件，如图 1-4 所示。

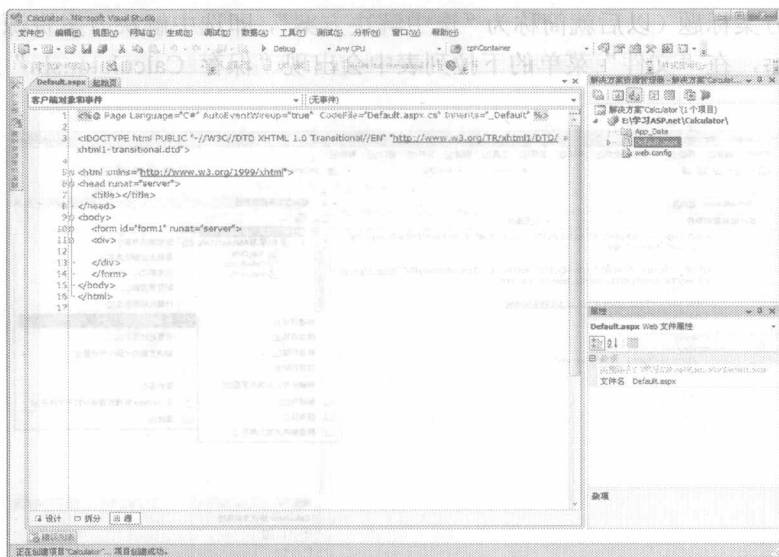
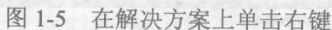


图 1-3 解决方案资源管理器



图 1-4 解决方案所对应的磁盘目录

解决方案的后缀名是“.sln”，上例中的解决方案全名应为“Calculator.sln”。选中图 1-5 中所示的解决方案标题（以后就简称为“选中解决方案”，即选中解决方案资源管理器中的解决方案名称）后，在“文件”菜单的下拉列表中会出现“保存 Calculator.sln”选项，以保存解决方案文件。



根据作者的经验, G1.1.1.1 文件最好只保留到“三、第三阶段”为止, 后面的“四、第四阶段”和“五、第五阶段”最好另起文件。

先看这个“E:\我的文件\Visual Studio 2008\Projects\Calculator\Calculator.sln”默认的保存