



康祥顺 编著 吴欣妍 审校

A SP.NET 3.5 开发技术 基础与实践教程



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

内容简介

ASP.NET 3.5 开发技术基础与实践教程

康祥顺 编著

吴欣妍 审校

图书在版编目(CIP)数据

ASP.NET 3.5 开发技术基础与实践教程 / 康祥顺编著. —北京: 电子工业出版社, 2009.7

ISBN 978-7-121-08200-9

I. A... II. 康... III. 网页制作—教材 IV. TP393.032

中国版本图书馆CIP数据核字(2009)第012325号

责任编辑: 曹海

文字编辑: 曹海

印刷: 北京天竺源丰印刷厂

发行: 电子工业出版社

地址: 北京市西城区百万庄大街24号

邮编: 100037

北京市东城区东直门内大街2号

邮编: 100000

电话: 010-88254888

网址: www.eipub.com.cn

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要介绍用微软公司最新的 ASP.NET 3.5 技术开发 Web 应用系统的基本原理和技术技巧。在编写过程中, 全书按照程序设计的基本规律进行编排, 以实例来验证开发技术和原理, 力求做到有理有据, 用事实说话。全书共分 11 章, 包括基本环境搭建、开发控件的学习、页面布局 and 外观控制、数据处理和输出、Web 服务应用程序的编写、安全控制原理和技巧, 以及大型的综合案例。大型综合案例不仅是对前面知识点的回顾、巩固和提高, 还引入了项目架构设计的知识和理念。

本书适合于学习 ASP.NET 技术的各层次读者阅读。如果你是初学者, 本书将带你顺利地进入 ASP.NET 开发领域; 如果你已具备 ASP.NET 技术基础, 本书将是你提高开发技能和更新开发理念的必备工具。

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 开发技术基础与实践教程 / 康祥顺编著. —北京: 电子工业出版社, 2009.3
ISBN 978-7-121-08200-9

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

中国版本图书馆 CIP 数据核字 (2009) 第 011535 号

责任编辑: 易 昆

文字编辑: 徐 磊

印 刷: 北京天竺颖华印刷厂

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

北京市海淀区翠微东里甲 2 号 邮编: 100036

开 本: 787×1092 1/16 印张: 28.75 字数: 730 千字

印 次: 2009 年 3 月第 1 次印刷

定 价: 50.00 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 zlt@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

基于.NET 框架的 ASP.NET 技术,使开发新一代的网络信息服务平台更加方便与快捷。在微软随后发布的产品中,几乎全是围绕.NET 展开的。.NET 计划正在全世界如火如荼地进行着,基于.NET 框架下的电子商务、电子政务等 Web 应用服务平台也越来越多地出现在人们面前,它的优越性已渐渐地显露出来,而 ASP.NET 编程技术也越来越被程序员们所喜爱。如果说 ASP.NET 1.x 是蹒跚学步的话,那么 ASP.NET 2.0 就已经实现了慢跑,而现在的 ASP.NET 3.5 已基本实现了展翅飞翔。可以预料,在不久的将来,微软还会在 ASP.NET 方面带给我们更大的惊喜。

我们还是再次温习一下.NET 的概念吧。.NET 框架是个开放的结构体系,所有的代码都被编译成中间语言,并且受框架托管。ASP.NET 严格基于.NET 框架,并为开发 Web 应用程序提供了一个革命性的编程模型。一方面它放弃了以往 ASP 的解释运行机制,在 ASP.NET 中,所有的业务逻辑代码和前台的 HTML 标记代码是分离的,并且所有的业务逻辑代码都是先编译后执行的;另一方面 ASP.NET 提供了丰富的类库,在类库中包含了丰富的控件和基本结构,Web 程序员可以像使用 Visual Basic 一样快速高效地搭建 Web 应用程序。所以 ASP.NET 从根本上解决了编程语言的问题,程序员基本上都可以使用自己所熟悉、喜爱的一种语言来编写 Web 应用程序。它在同一个项目中,不同语言所编写的类库可以彼此互用。它抛弃了 VBS 脚本语言,采用 Visual Basic.NET 作为默认的开发语言,并且为.NET 平台开发了一种全新的专用语言——C#。ASP.NET 通过改良的 ADO.NET 来实现对数据库的操作,ADO.NET 技术采用了离线式的数据操作,提高了分散式应用程序的效率与扩展性。另一方面 ADO.NET 与 XML 的无缝结合,使得 ASP.NET 操作数据的跨平台性大大增强。

本书按照学习 ASP.NET 技术的基本规律,从最基本的知识概念开始,到最后开发 Web 应用项目,每个章节,每个知识点,都有详细的解释说明和举例应用。读者可以跟随章节的推进和知识的深入,循序渐进,一步一个脚印地学习,最后一定可以系统地掌握 ASP.NET 编程技术,并将其应用到实际的解决方案中。

本书共有 11 章。第 1 章是入门基础,简要介绍了 ASP.NET 3.5 开发的基本环境配置,并通过一个简单的 ASP.NET 程序的开发过程,向读者展示了 ASP.NET 开发的简单和快捷性。只有搭建好了开发平台,才能够让程序员顺利地进行开发工作。第 2 章简要地分析了 ASP.NET 页面的基本结构,以及运行的基本原理,读者只有了解了基本原理,才能够在后面的学习中有的放矢。读者不但要注重基本知识和技能的掌握,还需要注重 ASP.NET 程序执行效率的提高,如果对 ASP.NET 程序的基本运行过程和结构没有清晰的概念,是无法做到更好的。第 3 章介绍 ASP.NET 3.5 中基本控件的应用,如文本控件、选择控件和列表控件等。更为重要的是介绍了 ASP.NET 3.5 中那些具有新特性的控件,虽然很多的控件在以前的版本中都已经提供给开发者了,但是对于大多数的新用户来说还相当陌生,因此,这一章通过具体的实例详细深入地进行了解。熟练地掌握了这一章的技巧和原理,可为后面的学习,甚至是整个 ASP.NET 开发打下良好的基础。第 4 章的主要内容是介绍 ASP.NET 页面的布局和导航功能,因为随着 Web 开发技术的发展,人们发现开发应用程序已经变得十分简单,但是对于用户来说,好的、人性化的用户界面也是他们追求的一个方面。因为用户不会去关心程序是如何实现的,他们只关心程序是

否能够给他们的工作带来帮助、是否便于操作，所以关于界面的设计现在已经被列为单独的一个方面来开发了。对于 Web 程序来说，ASP.NET 提供了丰富的、便捷的页面布局和导航工具，如果读者能够灵活地将它们应用到页面设计中，便可大大地提高程序的操作性和友好性。第 5 章主要介绍了页面外观控制技巧。在 ASP.NET 中，提供了对页面外观控制的多种技术支持，如可以通过外观文件来控制页面的布局、颜色等，这样就可以由用户自主地根据喜好调整页面的外观和布局，大大增加了页面的人性化，增强了程序的友好性。第 6 章和第 7 章主要介绍了数据库开发的关键知识和技巧。任何应用程序都离不开数据处理，而在 ASP.NET 开发中，数据库被广泛支持，特别是 ADO.NET 模型的推出更是提高了 Web 程序的性能。因此，只有掌握了数据处理的原理和技巧，才能够让你的 ASP.NET 开发变得丰富多彩。第 8 章主要介绍了 ASP.NET 中的数据绑定控件，为了简化 ASP.NET 开发和数据处理工作，ASP.NET 3.5 提供了丰富的、可以直接绑定数据的控件。这一章与第 6 章和第 7 章可以说是相互依存的，只有熟练地使用这些数据绑定控件，才能够更加熟练地处理数据，从而提高 ASP.NET 程序的数据处理能力。第 9 章主要介绍了一个新的 Web 开发的理念，即 Web 服务开发。尽管内容不是太多，但它的确让你对 Web 服务开发有了一定基础的了解。如果需要熟练地掌握，则还需要读者去学习其他专业性的书籍。第 10 章是 Web 程序开发，这也甚至是所有的程序开发都必不可少的一项工作——安全性管理。安全性管理是涉及到任何工作的重中之重，它可决定工作的成败。因此，在这一章中详细地介绍了 ASP.NET 提供的用于安全管理工作的技术和技巧，学习和掌握了本章知识，可为你的 ASP.NET 开发画上一个圆满的句号。第 11 章，通过对一个综合案例的编写和解读，让读者能够把前面所有章节的知识串联起来，为正式步入 ASP.NET 开发领域做一个简单的热身。同时，这一章也通过实例向读者传递了另一个极其重要的技巧，就是在 ASP.NET 开发中的系统架构方式，尽管都是分布式开发的，但本章的例子主要演示的是如何将系统的架构层分布到不同的机器位置。请读者在学习前面的章节后，系统地学习本章的综合案例，理清案例的实现逻辑，为即将开始的 ASP.NET 开发工作理清思路，做好准备。

本书内容翔实，案例丰富且极具应用性。案例配有大量的界面截图、源码及详细的代码解释。无论是 ASP.NET 技术的初学者，还是有一定基础的爱好者，都可以在本书中得到帮助。

本书由康祥顺编著，在编写过程中还有不少的同志付出了辛勤的劳动，他们是黄显堂、刘东鸿、张小丽、李欣、保春艳、杨继锋、张强、刘亮、马煜、崔竞、张庆、梅光耀、蔡宏，在此表示衷心的感谢！另外在本书的编写过程中，电子工业出版社和北京美迪亚电子信息有限公司的全体员工也付出了辛勤的劳动，在此一并表示最诚挚的谢意！本书中所有实例，编者都在 Visual Studio.NET 2008 的开发环境中进行了反复测试，并将所有的源代码按章节组织，形成了配套资料，便于读者学习使用。由于水平有限，尽管做了严格的审核和测试，书中难免会有一些错误，敬请广大读者不吝赐教，编者在此表示感谢！

为了方便读者阅读，本书配套资料请登录“华信教育资源网”(<http://www.hxedu.com.cn>)，在“资源下载”频道的“图书资源”栏目下载。

反侵权盗版声明

· 电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任 and 行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章 构建 ASP.NET 开发平台	1
1.1 你所不知道的 ASP.NET	1
1.2 有什么必要使用 ASP.NET	2
1.3 建立 ASP.NET 开发运行平台	3
1.4 使用 Visual Studio.NET 2008 建立第一个 Web 应用程序	4
1.4.1 设计网页页面	9
1.4.2 为项目添加代码	10
1.5 搭建真正的 Web 站点 环境——IIS 虚拟目录	11
1.6 本章小节	14
第 2 章 ASP.NET 页面结构	15
2.1 ASP.NET 页面的基本结构	15
2.2 神奇的 Runat 属性	18
2.3 认识 ASP.NET 页面的特性	20
2.4 了解 Web 窗体的代码模型	21
2.4.1 利用行内代码模型编写 Web 应用程序	22
2.4.2 利用代码分离模型编写 Web 应用程序	23
2.5 Web 窗体的生命周期与 状态管理	27
2.6 认识 Web 窗体的事件模型	30
2.6.1 如何定义 Web 窗体的 组件事件过程	30
2.6.2 了解访问事件的必备参数	31
2.6.3 使用组件的 AutoPostBack 属性	32
2.7 本章小结	33
第 3 章 ASP.NET 核心服务器控件	34
3.1 熟练使用 HTML 控件	34
3.1.1 认识 HTML 控件	34
3.1.2 几个常用的 HTML 控件	35
3.2 常用 Web 服务器控件	45
3.2.1 文本类控件的应用	45
3.2.2 选择类控件的应用	46
3.2.3 列表类控件的应用	49
3.3 验证控件的应用	54
3.3.1 RangeValidator 控件	55
3.3.2 RegularExpressionValidator 控件	56
3.3.3 RequiredFieldValidator 控件	57
3.3.4 CompareValidator 控件	58
3.3.5 CustomValidator 控件	59
3.3.6 验证控件应用综合实例	62
3.4 Web 杂项控件应用	65
3.4.1 AdRotator 服务器控件应用	65
3.4.2 Calendar Web 服务器控件 概述	77
3.5 设计 Web 用户控件	84
3.5.1 什么是 Web 用户控件	84
3.5.2 在 Visual Studio .NET 2008 中 编写用户控件	87
3.5.3 含有属性和方法的用户 控件的编写	92
3.5.4 在用户控件中使用保护成 员和自定义方法	100
3.6 本章小节	104
第 4 章 页面布局和导航	105
4.1 SiteMapPath 控件的应用	105
4.1.1 创建站点地图数据	105
4.1.2 SiteMapPath 服务器控件	107
4.1.3 PathSeparator 属性	111
4.1.4 PathDirection 属性	113
4.2 Wizard 服务器控件应用	115
4.2.1 一个简单的 Wizard 控件 应用案例	115
4.2.2 Wizard 控件基本元素	118
4.3 ASP.NET Menu 菜单控件	127
4.3.1 如何控制菜单的显示行为	127

4.3.2	如何通过菜单层样式 控制菜单项的外观	128
4.3.3	如何将图像用于 Menu 控件	135
4.3.4	如何将 Menu 控件与网站 地图文件 Web.sitemap 绑定	137
4.4	TreeView Web 服务器控件	140
4.4.1	如何在 TreeView 控件中 显示数据	141
4.4.2	自定义 TreeView Web 服务器控件的外观	154
4.4.3	如何使 TreeView 控件与 Menu 控件协同工作	156
4.5	本章小结	159
第 5 章	页面外观控制	160
5.1	母版页 (Master Page) 的 应用	160
5.2	主题应用: 外观文件 (Skin File)	165
5.2.1	创建和应用简单主题	166
5.2.2	默认外观与命名外观	168
5.2.3	将主题应用于整个应用 程序	170
5.2.4	主题与 StyleSheetTheme 样式主题	170
5.2.5	图像和主题	172
5.2.6	级联样式表和主题	172
5.2.7	动态加载主题	174
5.2.8	通过控件对象的 CssClass 属性控制外观	175
5.3	本章小结	182
第 6 章	ADO.NET 技术与远程 数据源的连接	183
6.1	为什么要使用 ADO.NET 处理数据	183
6.2	连接到远程物理数据源	184
6.2.1	使用“服务器资源管理器” 创建数据连接	185
6.2.2	在代码中动态创建到数据 源的连接	187

6.3	认识 AccessDataSource 数据 组件	191
6.3.1	通过 AccessDataSource 组件 连接到数据源	191
6.3.2	数据库文件位置的设定 技巧	194
6.3.3	处理 Access 连接失败	195
6.4	SqlDataSource 数据组件	197
6.4.1	通过 SqlDataSource 组件取得 数据源数据	197
6.4.2	将连接保存在 Web.config 配置文件中	203
6.4.3	如何取得数据源中的数据 库的相关信息	206
6.4.4	处理 SqlDataSource 组件的 连接错误	209
6.5	连接到任何数据源	210
6.6	本章小结	213
第 7 章	ADO.NET 数据处理技术	214
7.1	利用数据适配器 DataAdapter 对象与数据源通信	214
7.2	利用 DataSet 对象将数据源中 的数据存取到本地缓存中	216
7.3	利用 DataTable 对象对数据集 中的表进行操作	217
7.3.1	动态创建数据表	218
7.3.2	操作数据表中的数据	224
7.3.3	一对多关系表的处理	230
7.4	利用 DataView (数据视图) 对象显示数据	237
7.4.1	通过数据视图操作数据集 中的数据	238
7.4.2	通过数据视图处理相关表	245
7.5	通过 DataSet 数据集更新数 据源	247
7.6	直接对数据源进行操作	253
7.7	在数据集中处理 XML 数据	261
7.8	本章小结	265
第 8 章	在 ASP.NET 页面中绑定数据	266
8.1	数据绑定方法	266

8.1.1	通过“<% %>”标识符在 页面中嵌入逻辑代码	266	9.2.1	创建 Web 服务	384
8.1.2	“<%= %>”和“<%# %>” 绑定表达式	270	9.2.2	在 ASP.NET Web 应用程序 中调用 Web 服务	388
8.1.3	数据绑定控件的 DataBind () 方法	274	9.3	调用 Web 服务的限制	391
8.1.4	DataBinder 类	277	9.4	调用网络上的 Web 服务	391
8.2	Repeater 控件的应用	281	9.5	本章小结	397
8.2.1	将 Repeater 控件绑定到 数据源	281	第 10 章	ASP.NET 程序的安全性管理	398
8.2.2	在 Repeater 控件中编辑 数据	286	10.1	安全控制的作用及原理	398
8.2.3	Repeater 控件中数据的分页 显示	292	10.2	Web.config 配置文件	400
8.2.4	Repeater 控件中的数据排序	298	10.2.1	认识 Web.config 文件	400
8.2.5	Repeater 控件的事件及其 应用案例	303	10.2.2	Web.config 配置文件	400
8.3	DataList 控件	314	10.2.3	自定义 Web.config 文件 配置节点	401
8.3.1	DataList 控件的分列输出 功能	314	10.3	ASP.NET 窗体身份验证	402
8.3.2	DataList 控件的选择输入 功能	316	10.4	ASP.NET 安全管理类	407
8.4	GridView 控件	327	10.4.1	FormsAuthentication 类	407
8.4.1	将数据绑定到 GridView 控件	327	10.4.2	Membership 类	407
8.4.2	筛选 GridView 数据	331	10.4.3	Roles 类	413
8.4.3	GridView 控件的分页功能	337	10.5	ASP.NET 中的安全验证 控件	416
8.4.4	GridView 控件的排序功能	346	10.5.1	Login 控件	416
8.4.5	GridView 控件的编辑功能	354	10.5.2	LoginName 控件	417
8.4.6	优化 GridView 控件的 编辑功能	364	10.5.3	LoginStatus 控件	417
8.5	本章小节	382	10.5.4	LoginView 控件	418
第 9 章	ASP.NET Web 服务	383	10.5.5	PasswordRecovery 控件	418
9.1	什么是 Web 服务	383	10.5.6	ChangePassword 控件	419
9.2	创建和使用 Web 服务	384	10.5.7	CreateUserWizard 控件	419
			10.6	本章小结	426
			第 11 章	综合应用案例	427
			11.1	基于 Web 服务技术的 B/S 架构程序	427
			11.1.1	创建 Web 服务	427
			11.1.2	在 ASP.NET 程序中访问 远程 Web 服务	436

第1章 构建 ASP.NET 开发平台

ASP.NET 作为一种 Web 开发平台, 提供了构建企业级应用所需的服务、编程模型和软件基础结构。全新的编程模型, 旨在更加快速地开发 Web 应用, 它提供了一种基于组件的、可扩展的、易于使用的方法, 用以构建、部署和运行供所有浏览器或移动设备使用的 Web 应用系统。为了引导读者使用 ASP.NET 开发 Web 应用系统, 本章将介绍 ASP.NET 的开发理念, 为什么需要用 ASP.NET 开发 Web 应用, 如何建立 ASP.NET 开发平台, 以及如何在 ASP.NET 开发工具之一的 Visual Studio.NET 2008 中快速开发 Web 应用程序。

1.1 你所不知道的 ASP.NET

ASP (Active Server Page) 是微软公司研发的一种交互式网页编程技术。从 1996 年发布的 ASP 1.0 开始, ASP 开始从实验室走向实际应用, 但是它并没有为人们所追捧。1998 年微软发布了 ASP 2.0。2000 年, 微软公司发布了它的革命性的服务器系统 Windows 2000, 该系统上集成了 IIS 5.0, 并捆绑了 ASP 3.0。由于 ASP 提供了一系列的可以用来执行高级功能的(如用 ADO 对象来实现对数据库的操作) Web 应用程序组件, 再加上 Windows 系统的稳定支持, 因此 Windows 2000+ASP 3.0 便成了当时最流行的 WWW 服务器模式, 也因此使 ASP 在全球风靡起来。ASP 的编程语言为 VBScript 和 JavaScript, 运行机制是解释型的。ASP 页面文件的后缀名为 .asp。当客户机提交访问时, Web 服务器就找到该页面, 并交给解释引擎对 ASP 页面执行一次解释, 然后把结果发送给客户机。在当时, 这种技术是具有先进性的。但是随着 WWW 服务的广泛应用, 越来越多的 Web 应用程序应用到 WWW 服务上, 解释型的 ASP 技术在处理大型 Web 程序和进行频繁访问的时候, 会给服务器带来瞬间几何级系统开销, 因此, ASP 的改进就显得很必要了。

2001 年, 微软公司推出了 ASP.NET (当时的命名为 ASP+)。从命名上看, 可以说 ASP.NET 是 ASP 3.0 的升级。实际上 ASP.NET 是一种全新的交互式网页编程技术, 是网站和 XML Web 服务的产物, 也是微软公司新的应用开发平台 .NET 框架中的核心要素。如果说微软公司的 .NET 计划是编程技术的一项革命, 那么, ASP.NET 则无疑是 ASP 的一项革命, ASP.NET 技术把面向对象的编程技术引入到 Web 编程中, 这使得在编制 Web 应用程序的时候, 可以像编制 Windows 应用程序一样简便快捷。

首先说说在运行机制上 ASP.NET 的改变。ASP.NET 被设计成首编译方式。在页面代码成功调试完毕后, 可以进行人工编译, 也可以在系统得到第一次访问请求时自动编译源代码, 编译后生成了一个基类, 以后每次访问都是直接调用该基类。单纯从页面运行效率上讲, 编译式的运行机制肯定要高于解释式的。

其次在编码方式上, ASP.NET 采用了代码分离的技术。在以往 ASP 编程的时候 Web 程序的 UI 代码 (HTML 布局代码) 是和程序逻辑代码 (VBScript 或 JavaScript) 混编在一起的, 这样所带来的弊端就是美工人员不得不熟悉代码, 程序人员不得不注意美工的需要, 除此之外就是整个页面代码的层次、结构混乱不堪, 特别是在调试阶段, 如果对整个程序没有很深的了

解是很难快速推断并找出错误的。而 ASP.NET 的代码分离技术则杜绝了这种现象,在 ASP.NET 中所有的逻辑代码是被封装在<Script></Script>代码块中的,由程序员负责编写逻辑代码。而 UI 部分则由控件和 HTML 标记组成,由美工人员负责美工处理。这样不仅使程序员和美工人员的工作量大大减少,可提高程序的编制效率,而且缩短了程序开发周期(这大概是每个项目经理都希望的事情吧)。

最后从支持语言上来看,ASP.NET 放弃了对 VBScript 的支持,新的支持语言包括 Visual Basic.NET (有的书中称为 Visual Basic 7.0)、C# (读 C Sharp) 和 J# 等。其中 C# 是微软推出的一种专门用于 ASP.NET 的新编程语言,它具有 Visual Basic 的编程高效性和 C++ 语言的功能强大性,编程语法规则和 C/C++ 很相似。因此 ASP.NET 从一诞生就有了深厚的“群众基础”。

1.2 有什么必要使用 ASP.NET

在学习本书之前,读者肯定心里要有疑问,即为什么要学习 ASP.NET。

要弄清这个问题,我们必须要了解微软公司的 .NET 计划,正所谓“知己知彼,百战不殆”。对于 .NET,IT 界是众说纷纭,各抒己见,甚至就连微软公司也没有给人们提供一个清晰的解释。根据微软给出的解释我们可以总结出 .NET 技术的要点:.NET 的核心技术包括分布式计算、XML、组件技术和即时编译技术等。分布式计算技术是现代网络的本质;XML 奠定了新一代电子数据交换的标准,真实数据交换使网络计算成为可能;组件技术是软件技术发展的技术成果,组件的利用可以成倍地提高程序开发效率,可以解决绝大部分的代码重用问题;即时编译技术能够使应用程序根据特定的主机环境进行运行时的代码优化,简化代码发放过程。.NET 的核心内容之一就是要搭建第三代互联网平台,这个网络平台将解决网站之间的协同合作问题,从而最大限度地获取信息。在 .NET 平台上,不同网站之间通过相关的协定联系在一起,自动交流,协同工作,并提供最全面的服务。

.NET 的核心开发产品主要包括 Visual Studio.NET、.NET Framework,以及基于 .NET Framework 的 ASP.NET。.NET Framework (即 .NET 框架)是在 .NET 平台上进行开发的基础,除了提供传统的编程开发平台外,.NET 框架还为智能设备提供了开发平台,是实现微软的目标(为用户提供精彩的体验——任何时间、任何地点、任何设备)的关键部分。.NET 精简框架把托管代码的世界从 Web 服务带到了智能设备上,允许在个人数字助理(PDA)、移动电话、机顶盒设备上开发安全的、可下载的应用程序。.NET 框架之所以可以这样做,主要归功于它的核心技术,即通用语言运行库(Common Language Runtime: CLR)。那么,在此基础上的 ASP.NET 又是怎样运行的呢?我们来看如图 1-1 所示的在 .NET 框架下的 ASP.NET。

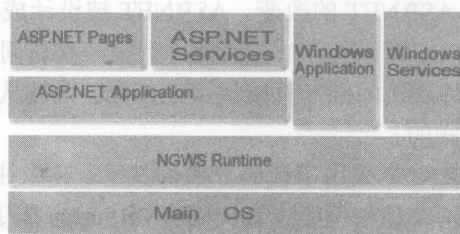


图 1-1 .NET 框架下的 ASP.NET

在 .NET 框架下,所有代码都是受系统托管的,即代码托管技术。受托管代码在 NGWS(Next Generation Windows Services 下一代视窗服务)运行时运行。而 NGWS 框架的最大优越之处在于它有一个中间语言编译执行环境,无论源程序的编写语言是什么种类(Visual Basic、C#、JScript 或 Perl 等),均被编译成标准的中间语言,然后再生成二进制代码被执行,直到源程序改变的时候才再次执行新的编译。

ASP.NET 是建立在公共语言运行库上的编程框架,可用于在服务器上生成功能强大的 Web 应用程序。与以前的 Web 开发模型相比,ASP.NET 提供了数个重要的优点(实际上在 1.1 节中已经列举了 ASP.NET 的几个优点):

(1) 增强的性能。ASP.NET 是在服务器上运行的编译好的公共语言运行库代码。与被解释的前辈不同,ASP.NET 可利用早期绑定、实时编译、本机优化和盒外缓存服务,这相当于在编写代码行之前便显著提高了性能。

(2) 世界级的工具支持。ASP.NET 框架补充了 Visual Studio 集成开发环境中的大量工具箱和设计器。“所见即所得”式的编辑、拖放服务器控件和自动部署只是这个强大的工具所提供的功能中的少数几种。

(3) 威力和灵活性。由于 ASP.NET 基于公共语言运行库,因此 Web 应用程序开发人员可以利用整个平台的威力和灵活性。.NET 框架类库、消息处理和数据访问解决方案都可从 Web 无缝访问。ASP.NET 与语言无关,所以可以选择最适合应用程序的语言,或者跨多种语言分割应用程序。另外,公共语言运行库的交互性可保证在迁移到 ASP.NET 时保留基于 COM 的开发中的现有投资。

(4) 简易性。从简单的窗体提交和客户端身份验证到部署和站点配置,ASP.NET 使执行常见任务变得容易。例如,ASP.NET 页框架可以生成将应用程序逻辑与表示代码清楚分开的用户界面,这和在 Visual Basic 的简单窗体处理模型中处理事件类似。另外,公共语言运行库利用托管代码服务(如自动引用计数和垃圾回收)简化了开发。

总之,ASP.NET 为我们提供了一个丰富、高效、稳定的开发平台,也为 Web 程序员们描绘了一个美好的前景。

1.3 建立 ASP.NET 开发运行平台

为了便于开发设计人员使用 ASP.NET 开发平台,微软提供了两种方式,一种是轻量级的简化安装,另一种就是重量级的完全安装。所谓轻量级的简化安装就是微软提供的专门用于开发 ASP.NET 的基本环境 Visual Web Developer Express,简称 VWD Express。此开发平台包括很多的开发工具。首先是一个编辑器,开发人员将在此编辑器中创建页面,此编辑器带有智能感知功能 IntelliSense,这是一个可以帮助开发人员完成命令输入,并为开发人员提供恰当选择建议的功能。另外还包括一个包含图标工具条,可以通过拖放这些工具条,完成在代码块中添加代码的功能,另一种添加代码的途径就是通过自动化向导来完成。此外,还包括一个 File Manager 文件管理器(用于组织 ASP.NET 的相关文件和文件夹)和一个 Data Explorer 数据浏览器(可以浏览和管理 ASP.NET 页面的数据源)。最后,包括一个附带的用于测试 Web 站点的 Web 服务器 Cassini,其功能类似于 IIS 信息服务器,但是相对来说它是一个轻量级的 Web 服务器。

另一个重量级的开发环境就是 Visual Studio .NET 2008,这是一个综合性的开发环境,集

成了很多的可视化的设计工具，包括设计不同版本 .NET Framework SDK 语言包的项目和利用不同语言工具开发项目的集成开发环境 IDE，以及开发项目所需的工作，诸如引用、数据连接、文件和文件夹等都通过相应的管理器来完成。相对来说，Visual Studio .NET 2008 开发环境综合了很多 VWD 开发环境所没有的功能和工具，它能够显著地提高项目设计的效率和质量。所以，本书建议读者在此环境下进行学习和项目开发，同时本书也是在此环境下进行项目演示和讲解的。

出于学习 ASP.NET Web 编程技术的目的，我们推荐在 Windows XP 系统平台下安装 Visual Studio.NET 2008 安装包包括 Visual Studio.NET 2008 的安装程序文件和产品文档，里面带有 .NET 框架等系统必备组件，一般安装程序会把这些项目首先安装到计算机中。根据安装程序的要求，选择和填写必要信息，就可以很方便地把 Visual Studio.NET 2008 安装到电脑中。由于安装过程比较简单，基本上根据安装向导的提示就可以成功完成，在此就不再介绍安装的详细过程了。

1.4 使用 Visual Studio.NET 2008 建立第一个 Web 应用程序

使用 Visual Studio.NET 2008 新建一个 Web 应用程序（项目），可以按照下面的步骤进行。

（1）运行 Visual Studio.NET 2008，第一次运行该程序时，系统会提示你设置常规开发环境的配置项，如图 1-2 所示，进行此配置可以优化开发环境的配置质量，取消一些无关功能的访问，加快相关功能的访问速度。在这里将其设置为“常规开发设置”。

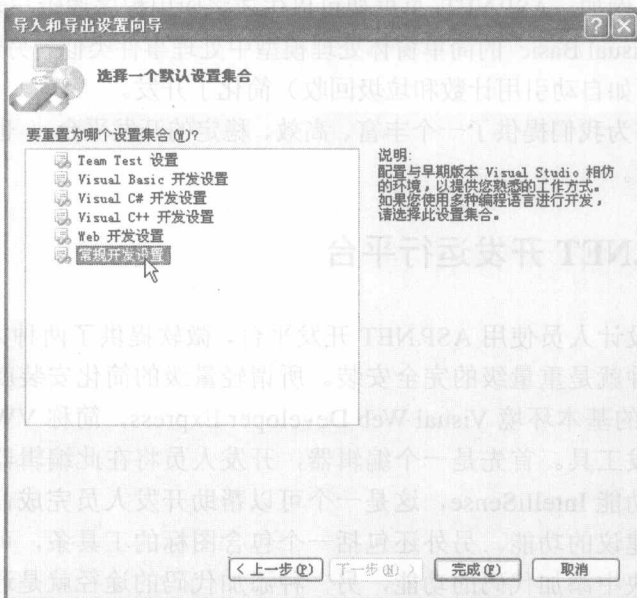


图 1-2 系统设置

当然，如果需要重新设置开发环境，可以通过选择系统菜单“工具”→“导入导出设置”，在“导入和导出设置向导”窗口中重新选择配置项，如图 1-3 所示。

（2）选择了“常规开发设置”配置项后，将出现主窗体，如图 1-4 所示。

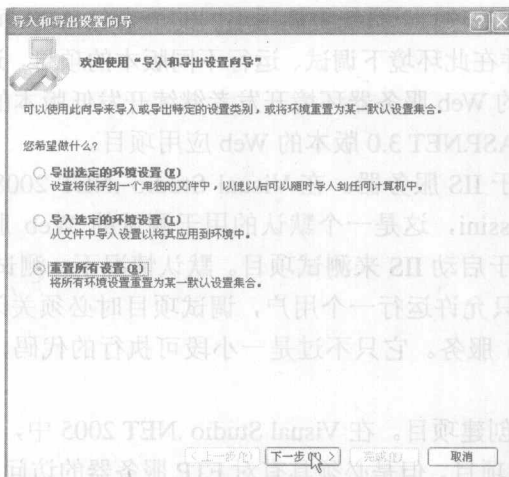


图 1-3 导入导出设置向导



图 1-4 主窗体

(3) 在主窗体中,“最近的项目”选项中列出了最近开发的项目,单击“网站”或“项目”链接,即可创建或打开一个项目。单击图 1-4 中“创建”后的“网站”链接,新建一个 Web 应用程序。系统会给出一个选择窗口,根据需要选择建立项目的类型及版本等信息,如图 1-5 所示。这里请选择创建“ASP.NET 网站”项目,并在.NET 版本框中选择“.NET Framework 3.5”。然后在“位置”项中,将系统默认的文件位置定义为“MyFirst”,即新建的 Web 应用程序名称为 MyFirst,保存位置为个人文档目录下的 MyFirst。当然也可以通过单击“浏览”按钮来选择类型为 FTP 或者 HTTP,同时选择语言工具为“Visual Basic”或“Visual C#”。

通过“新建网站”窗口的选项可以发现, Visual Studio .NET 2008 具有如下的改进。

①可以新建基于不同.NET Framework SDK 语言包版本的项目,包括 2.0、3.0 和 3.5 三个

版本,一旦建立了不同版本的项目,系统将调用不同的语言包。这样可以使用户根据自己的需要创建不同版本的项目,并在此环境下调试、运行不同版本的项目,这也有利于一些还没有升级的旧项目或还没有升级的 Web 服务器环境开发者继续开发低版本的 ASP.NET 应用程序,如开发基于 ASP.NET 2.0 或 ASP.NET 3.0 版本的 Web 应用项目。

②创建项目不再依赖于 IIS 服务器。在 Visual Studio .NET 2008 版本中,系统自带了一个小型的 Web 服务器 Cassini,这是一个默认的用于测试的 Web 服务器,可以方便地测试 Web 项目,而不需要依赖于启动 IIS 来测试项目。默认情况下,测试项目将自动启动 Cassini 服务器。但是该服务器将只允许运行一个用户,调试项目时必须关闭正在运行的页面,而且它也不能用于正式的 Web 服务。它只不过是一小段可执行的代码,只适合于单独的页,不能代替 IIS。

③在 FTP 中同样可以创建项目。在 Visual Studio .NET 2005 中,只能通过 FTP 打开项目,而在 2008 版本中可以新建项目。但是必须具有对 FTP 服务器的访问权,并且对特定的目录具有读写权限。

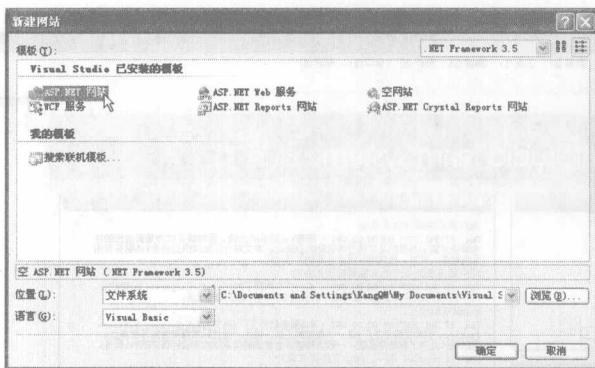


图 1-5 新建项目信息

(3) 单击“确定”按钮后, Visual Studio.NET 就完成了 MyFirst Web 应用程序的建立,并自动打开该项目,如图 1-6 所示。

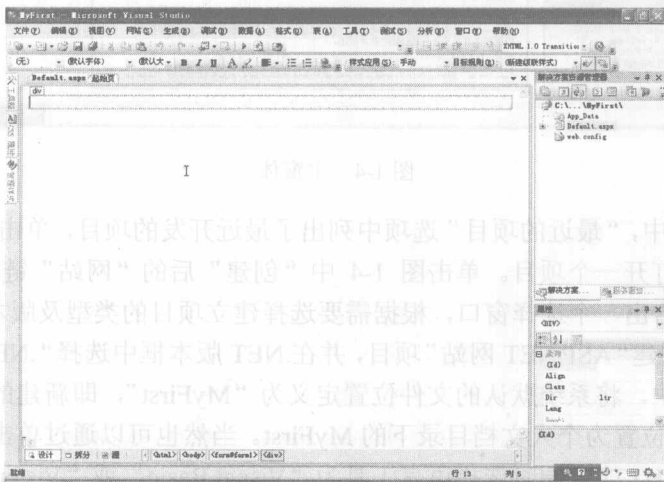


图 1-6 打开 MyFirst Web 应用程序

经过上面简单的步骤，我们就基本搭建好了第一个 Web 应用程序框架，接下来需要完成页面的设计及逻辑代码的定义。

首先来了解一下 Visual Studio.NET 2008 工具的基本结构,以便于开发应用程序。在图 1-6 中,我们可以看到 Visual Studio .NET 2008 开发界面的基本布局。在窗体的左侧为自动打开的工具面板,包括“工具箱”、“CSS 属性”和“管理样式”工具面板,右侧为“解决方案资源管理器”和“属性”组合窗口,该项目已经有了一个默认的 Web 窗体,并且处于激活状态。Visual Studio.NET 为每个 Web 窗体提供了 3 个不同的视图设计时窗口和一个逻辑代码窗口。

(1) 设计窗口：采用“所见即所得”的方式，支持控件的拖拉放置。注意，在设计面板上提供了一个全新的窗口，类似于 Dreamwaver 的“拆分”窗口，可以很方便地查看 HTML 代码和设计效果。

(2) HTML 窗口，即“源”窗口：显示 Web 窗体页面中所有 HTML 代码和控件声明代码。

(3) 页面元素窗口: 在该窗口显示了页面中的 HTML 元素, 单击元素可以快速定位到指定的元素代码块。

(4) 逻辑代码窗口：在设计窗口中，双击鼠标就可以进入逻辑代码窗口了。由于 Visual Studio.NET 采用隐藏代码技术，所以，每个 Web 窗体都有一个对应的逻辑代码文件。

下面我们简单地了解一下窗口的各个区域的基本功能。

1. 工具箱

工具箱位于窗体的左边,默认情况下工具箱面板自动隐藏,当鼠标停留在工具箱图标上时,系统自动展开工具箱面板,通过单击钉子形状的按钮可以改变工具箱的状态,如图 1-7 所示。在工具箱面板上提供了很多功能各异的用于设计 Web 窗体的控件。

标准控件：包含大多数的 ASP.NET 3.5 页面设计的服务器端核心控件。

数据控件：包含数据源连接和数据显示绑定控件。

验证控件：包含用于验证客户端输入数据的有效性的控件，如必选字段的验证、数据格式的验证、数据范围的验证等。

导航控件：包含菜单、网站地图、目录树等导航控件。

登录控件：包含简单的用户身份验证的控件。

WebParts 控件：包含站点中可以由用户重新安排或隐藏的比较大的控件。

AJAX 扩展控件：包含用于创建 AJAX 类型的 Web 页面的控件。AJAX 是 ASP.NET 3.5 支持的一项比较好的技术，该项技术可以使页面只更新需要更新的部分，从而可以提高网页刷新的效率，减轻网络的负担。

报表控件：包含用于创建 Web 水晶报表的控件。

HTML 控件：包含常用的 HTML 标记控件，通过拖放这些控件将在 HTML 源文件中添加相应的 HTML 标记，并且设计出所见即所得的页面效果。

常规控件：该面板一般是空的，在创建了页面的一部分之后，可以选中该页面并将其拖放到常规面板，从而创建一个可以重复利用的工具。如果希望复制一组已经调整好格式的控件到其他页面，则这个工具正好可以派上用场。

另外，我们还可以安装很多的第三方控件。

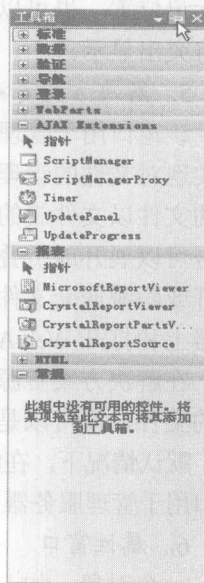


图 1-7 工具箱

在工具箱标题栏的右边有一个图钉形状的图标，单击该图标，它将变为水平方向，这时可以将工具箱在没有使用的时候隐藏起来，仅显示标题栏。注意，工具箱的控件在使用过程中是动态变化的，它将根据当前的项目类型提供可以使用的控件。

2. CSS 属性窗口

该窗口用于创建页面的样式，与工具箱一样，可以动态展开和隐藏该窗口。该窗口可以可视化地定义页面中使用的 CSS 样式表和级联式样式表，从而应用于当前页面的外观定义。

3. 管理样式

可以管理在页面中创建的样式、预览样式的效果，以及创建新的样式表。

4. 设计区

位于窗体中间部分的空白区域，开发人员将在这里完成 ASP.NET 页面设计的大多数工作。可以在设计视图和源代码视图之间切换，还可以同时以拆分的方式显示两种视图。设计视图以一种直观的方式显示页面的设计效果，而源代码视图可显示页面的 HTML 代码和内嵌的脚本代码。通常情况下，我们可以在设计视图中迅速地完成大多数的设计任务，而一些需要精确完成的设计任务可以在源代码视图中完成。

在 ASP.NET 3.5 的 IDE 集成开发环境中还提供了一种快速设置控件属性的方式，即在拖放控件时，将弹出一个快捷任务面板，在这个面板中包含了控件大多数的最常用的属性设置。并不是所有的控件都带有常用任务面板，如果有的话，可以通过控件右上角的“>”箭头图标打开任务面板，如图 1-8 所示。

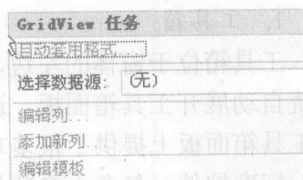


图 1-8 控件的智能任务面

在设计区的底部还有一个导航工具，它在大型的复杂的文档中很有用处。通过阅读导航标记可以迅速地定位到指定的标记位置，也可以迅速地知道插入光标的位置，当前的位置将突出显示。

5. 解决方案资源管理器窗口

该窗口用于管理和组织 ASP.NET 站点的文件，它通过类似于 Windows 资源管理器的功能来实现对文件的管理，用户通过该窗口可以查看、复制、创建及删除这些文件。该资源管理器中的文件以文件簇的形式进行组织，其最上层的目录就是项目文件的根目录，通过单击加号或减号可以展开或折叠文件夹。工具条上的几个按钮，从左至右依次为打开“属性”窗口，刷新、展开或隐藏嵌套文件的子文件，查看后台隐藏代码，打开设计视图，将整个站点部署到目标服务器上 and 打开 Web Administrator 站点管理工具。

在解决方案资源管理器窗口中的文件夹下，可以单击鼠标右键，在弹出的菜单中选择添加新的文件，既可以是新建的项目文件，也可以是已经存在的任何位置的文件。

默认情况下，在解决方案资源管理器窗口后面还叠加了一个“服务器资源管理器”窗口，该窗口用于管理服务器上的资源和数据，开发人员可以在窗口中链接远程数据源，然后浏览数据。

6. 属性窗口

一个对象，如一个 Web 页面、一个服务器端核心控件或一个数据源的链接，都具有很多属性。属性是一组设置，决定了对象的外观和操作，在 ASP.NET 中，已经预定义了很多控件的属性。属性的设置可以在代码中完成，也可以在“属性”窗口中完成，而且在属性窗口中完成简单直观，操作比较准确方便。为此，在窗口的右下角专门提供了一个“属性”窗口。属性显示在多个面板中，可以展开和折叠这些面板。在属性窗口的顶部有一个下拉列表框，其中包