软件开发
训练营

杨云 编著

ASP.NET开发一站式学习

——难点/案例/练习

ASP.NET

知识点范例翔实　本书对每个知识点提供精选范例和详细讲解，便于读者夯实基础知识

项目实例完整　本书精选完整项目实例，详细阐述了采用ASP.NET开发网页解决实际问题的基本思路和方法

难点解析与习题　难点解析与习题互补，使读者在深入理解知识点的基础上进一步巩固知识

作者微博答疑　地址为http://weibo.com/UFeedBack

清华大学出版社

软件开发
训练营

ASP.NET开发一站式学习

——难点/案例/练习

杨云 编著

清华大学出版社
北 京

内 容 简 介

本书讲解了ASP.NET的各个方面，共14章。全书采用知识讲述+代码示例+难点解析+习题的方式，使读者易于理解和学习。

除了配合ASP.NET特性讲解的小示例，本书最后还加入企业级ASP.NET大型站点示例。针对大型示例的讲解，本书采用数据库构架讲解、系统构架剖析和关键代码讲解来让读者对示例有结构上和功能上的认识，然后通过对示例添加一些功能，详细展示如何在现有功能的基础上开发自己的应用。

本书适合从事.NET Web开发的人员作为实际开发的辅导用书，也适合想从事Web开发的大中专院校的学生作为教材学习使用。

图书在版编目(CIP)数据

ASP.NET开发一站式学习——难点/案例/练习/杨云编著. —北京：清华大学出版社，2013.8
（软件开发训练营）

ISBN 978-7-302-31828-6

Ⅰ. ①A… Ⅱ. ①杨… Ⅲ. ①网页制作工具—程序设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2013)第062948号

责任编辑：袁金敏
封面设计：陈晓兵
责任校对：胡伟民
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**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：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：28　　**字　　数**：700千字
版　　次：2013年8月第1版　　**印　　次**：2013年8月第1次印刷
印　　数：1～3000
定　　价：59.00元

产品编号：049690-01

前　言

你是否想摆脱只会拖曳控件进行开发的菜鸟状态？那么，这本书正是你需要的！

.NET 技术领域存在着一个“教科书与项目应用严重脱节”的怪圈：教科书中讲 ASP.NET 开发基本是拖拉 Web Form 控件的傻瓜化开发，而企业中则很少使用拖控件方式进行开发。

ASP.NET 作为 Web 开发的主要技术，要求开发人员把更多精力投入到改善设计和完善用户体验上。为了达到这个目的，就需要开发人员具备良好的基本功。

有些书中讲解数据库访问都是使用 DataSource 数据绑定，而企业中则使用 Enterprise Library、Linq 等技术；讲解 AJAX 都是使用 UpdatePanel、AJAX Toolkit 等控件，而企业中则一般使用 JQuery 等轻量级封装进行 AJAX 开发。这造成了很多读者根本无法适应用人单位的技术要求。

本书讲授的内容都是最贴近企业实际开发的技术，选择的大型系统都是企业级框架技术，读者掌握之后能很快适应实际的开发工作。

本书对基础章节都配有难点解析和练习题，提醒读者哪些知识点需要注意，哪些需要重点掌握，让你对知识做到“心里有数”。

本书主要面向使用 C#语言开发 ASP.NET 应用的人员，共 14 章，从 ASP.NET 的运行原理、控件使用、C#基础知识到时尚的图形控件技术，力争使读者通过学习能掌握如何使用 VS2010 开发基于 ASP.NET 的应用。对于比较重要的理论知识点都安排有相应的短小实例代码进行讲解。读者可以按照书中的示范编写代码来巩固知识点。

本书对知识点采取“引入”+“讲解”+“巩固”的方法，使读者能够从浅到深逐步理解，增加学习的兴趣。在最后一章安排了 1 个流行的综合实例，也就是 1 个完整的 Web 系统。在实例的讲解中都采用了精讲的方法，力求用最短的篇幅覆盖最多的知识点。

本书的读者对象不要求必须有 ASP.NET 的知识，适合于 ASP.NET 的初学者和 ASP.NET 开发人员进行技术升级，也可作为大中专院校相关专业教材使用。

本书主要由杨云编写，参加编写的还有李渝平、王春中、冉剑、陈代勇、陈佳佳、陈家云、王磊、王宇晟、胡浩、蔡芳、刘扬、陈雪郊、谢晓锋、王毅、刘小鹿、胡建、成梅花。

感谢母亲对我无微不至的照顾和支持。

目　录

第1章　Microsoft.NET简介

随着微软最新的操作系统 Windows 7 的发布，微软.NET Framework 也随之发展到了一个新的阶段。Microsoft.NET Framework 4.0 即是.NET Framework 平台的一个新的里程碑。它可以在 Vista，Windows 7 和 XP 操作系统上运行。

Microsoft.NET Framework 4.0 是微软.NET 软件平台的最新发展成果。它基于.NET4.0 运行时环境，并整合.NET 4.0 新增的多种新的功能，例如：Windows Presentation Foundation（Windows 界面基础框架）、Windows Communication Foundation（Windows 通信基础构架）、Workflow foundation（工作流基础构架）和 Windows CardSpace，并对 C#和 VB 等语言进行增强。

本章将对.NET Framework 4.0 及其组件进行整体描述，目的是让大家对这一新版本有一个清晰地了解，同时分析了采用的技术，并给出较为详细的说明。

1.1　.NET Framework 4.0 在.NET 技术体系中的位置

与.NET 4.0 给开发者带来大量新鲜而强大类库（WCF/WPF/WF）所不同的是，.NET Framework 4.0 来带了语法上的大幅增强，提供了新的编译器，优化了.NET Framewor 底层运行时，并对 WPF/WCF/WF 这几个重要的组成部分进行了升级和优化。虽然如此，.NET Framework 4.0 仍然是基于.NET 的公共语言运行时框架（CLR），如图 1-1 所示。

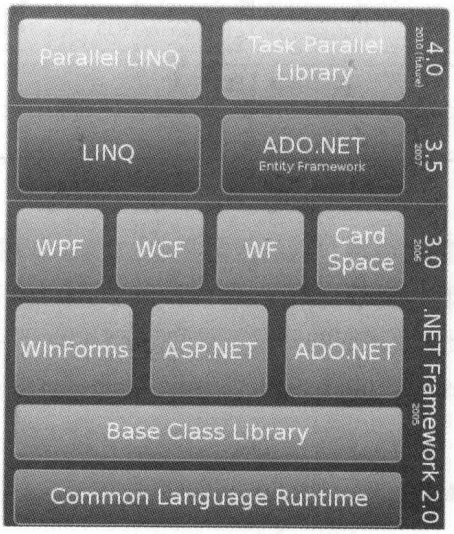

图 1-1　.NET Framework 4.0 的结构

如图 1-1 所示，.NET 4.0 并未对.NET 现存的技术进行任何改动，包括 ASP.NET，ADO.NET 和 WinForm 在内的主要技术都保持原样，这对熟练掌握.NET 的技术人员是个好消息，他们所掌握的技术仍然很有价值。从微软.NET 平台发展趋势看，.NET 平台将为.NET 4.0 和 4.5 提供基础类库，4.0 将着重引入语法、数据访问和 Web 客户端上的一些创新。可以把.NET 4.0/4.5 看做是.NET 的超集，是对.NET 的一次补充。

如果开发者从.NET 2.0 或者 3.5 迁移到 4.0，那么需要考虑代码的兼容性问题，尽管.NET 框架已经尽力保证向后兼容性，不过由于一些安全方面的改进，仍然有少数重要操作存在不兼容的现象。不过从.NET3.5 到.NET 4.0 则完全不存在这个问题，因为.NET Framework 3.5 的所有组件都可在支持 .NET Framework 4.0 的平台上运行。

从上面的描述，可以得出一个推论，.NET 4.0 是承上启下的一代，它继承了.NET 3.5 强大的基础平台，使.NET 平台在用户界面、网络通信、工作流和身份管理等功能上彻底进入了一个新的时代。.NET 4.0 在 3.5 的基础上增加了 LINQ（语言整合查询）和 ASP.NET 的 OR/M 框架（ASP.NET Entity Framework）等新功能，这些新功能极大地增强了 VB 和 C# 等语言，简化了开发工作，它们可以和 WPF/WCF 或 WF 共同工作，创建强大的应用程序。

1.2 .NET 4.0 各部分的功能

.NET 4.0 的基本结构在图 1-1 中已经比较清晰地展现出来了。图 1-2 更清晰地显示了.NET4.0 相关的开发组件之间的层次结构。

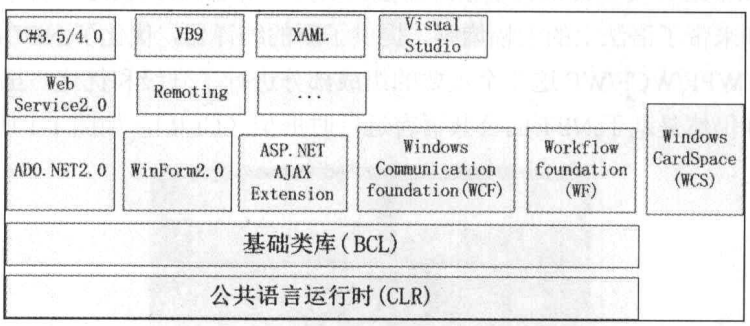

图 1-2 .NET 4.0 组件关系图

如图 1-2 所示，较深颜色方块包含的内容是.NET 4.0 新增的,其他的都是继承自.NET 2.0。开发.NET 3.5 的应用程序需要使用 Visual Studio 2008，而开发.NET 4.0 仍然可以使用 Visual Studio 2008，只需要安装相应的开发工具和.NET 4.0，即可使用熟悉的开发工具进行开发，也可以直接安装 Visual Studio 2010，Visual Studio 2010 直接支持 2.0/3.0/4.0 框架下的应用程序开发。

1. Windows Presentation Foundation（WPF）

WPF 是微软潜心开发的新一代用户界面框架。Windows 系统的用户界面的历史可以从

Windows 1.0 开始计算，期间经历了 GDI、GDI+直到 WinForm，用户界面技术一直都没有得到根本性的发展，只不过可以看做一次次的重构和封装，其本质依然是基于软件模拟渲染，不支持硬件图形加速。与此同时，Windows 游戏的图形界面、DirectX 则飞速发展，体积阴影、高精度纹理、硬件光源等技术的大规模应用，让 Windows 游戏世界一步一步向乱真的方向发展。桌面应用程序和游戏的图形效果差距过大、显卡的图形加速功能被普通应用程序所浪费，以及用户渐渐对一层不变的界面渐渐厌倦等原因，让微软决心推出新一代的用户界面框架，让普通应用程序的用户界面也能利用 DirectX 中的图形特效，并充分利用图形卡的加速功能，带给用户更好的体验。

2．XAML（eXtensible Application Markup Language：可扩展应用程序标记语言）

XAML 是为 WPF 量身定做用于用户界面描述的标记语言。XAML 采用了 XML 的格式，易于阅读，结构规范，也可充分利用 XML 的强大扩展性与其他图形制作工作进行交互。

3．Windows Communication Foundation（WCF）

在.NET 4.0 之前，Windows 操作系统下存在数种分布式消息交换的技术，例如：MSMQ、Web Service、Remoting、企业服务（Enterprise Services）、WSE，这些技术所涉及的编程模型也千差万别。微软创建 WCF 的其中一个目的就是统一 Windows 平台的分布式通信技术，让所有这些技术以统一的模型对外提供服务。这种统一的模型让.NET 框架成为了更完善的面向服务（SOA）的平台。

4．Workflow Fondation（WF）

工作流基础类库是新增加的大型基础类库。微软在此之前从未为外界提供过类似的开发包，因此这也是一次尝试。根据笔者的观察和试用，该开发平台相当强大，足够作为工作流引擎的基础平台。

1.3　.NET 4.0 的组件

1.3.1　Windows Presentaion Foundation（WPF）

WPF 是一个全新 UI 体系结构，它不仅能比以往的 UI 构架做的更多，还能做的更好，更容易。在 WPF 中，可以发现多种用户界面技术的痕迹。例如 GDI 和 GDI+，这一点其实毫无疑问，毕竟 WPF 是 GDI、GDI+的接班人。又如 HTML，WPF 引入了 XAML 语言作为界面描述语言，显然受到了 HTML 广泛应用的影响。在引入 XAML 进行 UI 描述以后，Windows 程序就可以采用 ASP.NET 那样的代码后置，将界面和程序逻辑分离。WPF 处理动画的方式显然吸取了 Flash 时间线（Timeline）的优点并且发展出独特的 StoryBoard 系统。最后，必须提到的是 WPF 为用户界面的 3D 化提供了强大的支持。WPF 之前，如果要在用户界面中提供 3D 元素，必须采用 2D 模拟 3D 的办法，或者采用 Directx/OpenGL 渲染的

方式。这两种方式存在的问题是：2D 模拟 3D 的性能非常低，不可能为 3D 元素提供多少特效；采用 DirectX 或者 Opengl 模式实现的用户界面不容易与其他 Windows 界面元素进行交互。在 WPF 中，这些都不再是问题，因为 WPF 采用了 DirectX9.0C 渲染 3D 元素，不仅原生地支持了 3D 元素，而且由于采用硬件加速，大大提升了用户界面的显示效率，使得开发人员有机会为用户界面提供更多更酷的效果。

WPF 吸取了多种技术的有点创造出一个新型的平台，在应用程序类型的支持种类上也比它的前辈们有不少提高。除了传统的 Windows 窗口程序以外，WPF 还支持浏览器应用程序（XBAP），XBAP 可以在客户端的浏览器中运行，它受浏览器的安全性限制；WPF 还吸收了 Web 程序的特点——功能按页面进行分离，新增了 Navigation Application（浏览型应用程序），可以让用户在同一个窗体中，如同访问 Web 页面那样执行程序，这个特性将会在后面的章节中进行较为详细的讲解。

知识点：

WPF 还有一个子集，叫做 SilverLight（开发代号为 WPF/Everywhere），它将作为一个浏览器的客户端技术运行在多个平台的浏览器中。在本书开始编写时，Silverlight 的 1.0 版本的 beta1 才公布不久，因此本书就不对它进行更多的说明。感兴趣的读者可以自行查阅微软站点的资料。

WPF 虽然是一个全新的 UI 框架，不过为了在最大程度上使 WinForms 的知识得以保留，开发人员仍然可以用相似的代码结构创建窗体和控件。下面的两段代码分别使用了 Winform 和 WPF 技术创建一个窗体和一个按钮。Winform 代码如下：

```
using System.Windows.Forms;
using System;

class program
{
[STAThreadAttribute()]
public static void Main()
{
  Form form = new Form();
   form.Text = "Winform";
   Button btn = new Button();
   btn.Text = "click me";
   btn.Location = new System.Drawing.Point(100, 70);
   form.Controls.Add(btn);
   Application.Run(form);
}
}
```

WPF 代码如下：

```
class program : Application
{
        [STAThreadAttribute()]
        public static void Main()
        {
            program app = new program();
            Window window = new Window();
            window.Title = "WPF Window";
            Button btn = new Button();
            btn.Width = 100; btn.Height = 40;
            btn.Content = "Click me";
            window.Content = btn;
            app.Run(window);
        }
}
```

两段代码的结构大同小异，分别定义了一个 Form 和 Window 对象作为主窗口，然后在主窗口中添加了一个按钮对象。稍微有所不同的地方是，在 WPF 中，如果不定义按钮的长和宽，那么默认情况下，按钮会充满整个包含它的容器。读者可以执行尝试去掉 btn.Width = 100; btn.Height = 40;这两个语句，看看窗口的效果。WinForm 和 WPF 在表面上看起来非常相似，但是它们却有本质上的不同。WinForm 技术实际上只是 GDI 的托管封装，内部仍然使用相同的控件、消息模型。

除了使用 C#和 VB.NET 等语言进行编程以外，还可使用上文已经提到的 XAML 对窗口进行描述，下面的代码是采用 XAML 标记语言编写的，它用来描述一个窗口，它的功能与上面的代码相同：

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="WPFStart" Height="300" Width="300">
  <Button Content="Click Me" Height="40" Width="100"/>
</Window>
```

代码如上所示，总体感觉和 HTML 比较类似。Window 标签用于声明一个 Window 对象，xmlns 的作用是声明一个 XML 的命名空间，xmlns:x 也是相同的作用，关于它们的详细解释可以查看第 4 章。<Button/>标签用于声明一个按钮对象，并且通过按钮对象的 Content 属性指定按钮上显示的内容，可以指定除文字以外的内容，例如图片、视频、其他控件容器，甚至可以指定一个按钮，基本上能用在 UI 上的元素都可以使用。

关于 WPF 更多的内容，将从第 3 章开始系统地介绍给读者。

1.3.2　Windows Communication Foundation（WCF）

WCF 除了建立一个统一的分布式通信编程模型以外，还有以下三个重要目标：

（1）基于面向服务的构架，并成为面向服务的基础平台。

（2）以更健壮得方式实现了 Web Service 和.NET Remoting。

（3）全面实现了 Web Service 的标准协议簇，WS-*。

微软创建 WCF 来为.NET 平台提供一种统一的分布式计算平台，它应当具有广泛的互操作性以及对面向服务构架的直接支持。

微软以及许多领先 IT 企业已经开始为面向服务（SOA）的平台做了准备，WCF 即微软公司的 SOA 战略中重要的一环。

面向服务的应用程序和先前的分布式应用程序有显著的不同。面向服务的应用的服务端和客户端是松散耦合的，客户端和服务端不必事先知道对方的存在，只要通过一个 URL，按照约定的通信方式就可以进行互操作。

或许会有一些疑问，在.NET 环境中，使用 Visual Studio、.NET Framework 以及 WSE 可以非常容易地编写 Web Service，创建服务端或者客户端，并且可以和运行在其他平台上的客户端或服务端进行交互。为什么还需要 WCF？

如开发者所知，Web Service 只是.NET 平台上一种可以用来构建分布式应用程序的技术，其他技术还包括 Remoting、Enterprise、Remoting 和 MSMQ 等。运用这些技术中创建分布式服务的方式和互操作性有很大的区别，而且当试图转换分布式计算的基础到另一种技术时，代价将相当大。WCF 就是要解决这个问题，它统一.NET 环境下分布式计算的编程模式，让开发人员可以不用太关心基础的分布式计算的机制。WCF 不仅可以编写 Web 环境可以访问的服务，在其他协议的网络中部署服务也不过是小菜一碟。

1.3.3 Workflow Foundation（WF）

Workflow 即工作流，对大多数人来说是比较新鲜的概念。但是在实际上，非常多的软件中都涉及工作流的处理。在 Workflow Foundation 中，工作流就是用一系列抽象出来的活动（Activities）描述现实世界中的一个流程。在现实中，工作流往往代表一件工作从一个人传递到另一个人，并伴随着状态的改变。一个典型的工作流往往定义了几个必要的步骤，步骤可以是有顺序的，也可以是无顺序的。举例来说，一个员工报销医疗保险费用，他的流程必须是：填写理赔单→提交表单→公司审核→医保机构审核→发放理赔金额。这个流程的每个步骤都是必须而且有序的，到最终发放理赔金额之前，任何一个步骤的失败都将导致流程中断。这个工作流比较简单，路径非常单一，复杂的工作流中涉及更多的分支大多数程序中，或多或少涉及的过程处理都可以看做是广义上的工作流。

工作流是由活动元素组成，可以将工作流拆解成为一个一个的活动（Activity）对象，然后按照逻辑组织起来描述工作流。

目前，大多数软件对工作流的处理方式都是私有的，缺乏通用性。不同的软件定义的工作流无法重用和扩展，这对软件的快速开发显然是极为不利的。WF 是将创建一个工作流所需要的各个环节抽象出来，供开发者使用。WF 并不关心开发者使用 WF 搭建的工作流用来处理什么具体事务，可以说只要处理的事务带有流程的性质，就可以应用 WF。

WF 中，工作流可以分为两种：一种是顺序流；另一种是基于状态机的工作流。在什么样的情况下使用哪一种工作流呢？通常情况下，如果一个流程重点关注的是顺序，较少