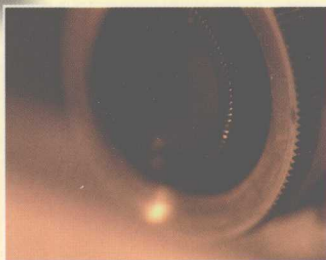
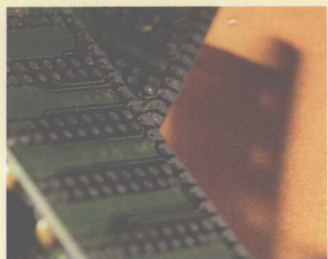




普通高等教育“十一五”国家级规划教材  
高职高专计算机系列

# ASP.NET 程序设计

李德奇 主 编  
梁洁婷 副主编



图件设计(ASP)目录

# 普通高等教育“十一五”国家级规划教材

## 高职高专计算机系列

内 容 提 要

# ASP.NET 程序设计

主 编 李德奇

副主编 梁洁婷

高职高专计算机系列

设计程序

◆ 主 编 李 奇

副 主 编 梁 洁 婷

责任编辑 燕 春 蕾

◆ 人 民 邮 电 出 版 社 出 版 发 行 所 址 北 京 市 朝 阳 门 外 大 街 13 号

邮 政 代 理 局 出 版 分 行 邮 政 代 理 局 出 版 分 行 邮 政 代 理 局 出 版 分 行

网 址 <http://www.ptpress.com.cn>

北 京 五 环 中 路 甲 5 号 网 公 司

编 审 处 行 发 京 北 总 店 行 发 经 销

◆ 开 本 787×1092 1/16

印 张 22.75

字 数 224 千

2007 年 9 月 第 1 版

印 数 1-3 000 册 2007 年 9 月 北 京 第 1 次 印 刷



人民邮电出版社  
POSTS & TELECOM PRESS

北 京

售 发 经 理 热 线：(010) 67179982 印 刷 质 量 热 线：(010) 67179323

## 图书在版编目 (CIP) 数据

ASP.NET 程序设计/李德奇主编. —北京: 人民邮电出版社, 2007.9

普通高等教育“十一五”国家级规划教材. 高职高专计算机系列

ISBN 978-7-115-16390-5

I. A... II. 李... III. 主页制作—程序设计—高等学校: 技术学校—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 089689 号

## 内 容 提 要

本书通过大量的实例, 详细介绍了基于 C# 的 ASP.NET 程序设计的基础知识和编程技巧。主要内容包括 ASP.NET 的概念、ADO.NET 数据库访问技术、Web 服务器控件的使用、内部对象的概念及其应用, 最后一章给出了一个综合应用实例, 详细讲述了使用 ASP.NET 开发电子商务网站的过程和方法。

本书内容翔实, 实例丰富, 讲解透彻, 注释详细, 实用性强, 便于读者对 ASP.NET 的理解和应用。

本书作为高职高专计算机及其相关专业的教材, 也适合自学 ASP.NET 的读者使用。

普通高等教育“十一五”国家级规划教材

高职高专计算机系列

ASP.NET 程序设计

- 
- ◆ 主 编 李德奇
  - 副 主 编 梁洁婷
  - 责任编辑 潘春燕
  - ◆ 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>  
北京华正印刷有限公司  
新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16  
印张: 22.75  
字数: 554 千字 2007 年 9 月第 1 版  
印数: 1—3 000 册 2007 年 9 月北京第 1 次印刷

---

ISBN 978-7-115-16390-5/TP

定价: 32.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

## 编者的话

自 2000 年 7 月 Microsoft 公司发布它的新一代软件开发工具——ASP.NET 以来，它就备受 IT 业界的关注，在应用程序开发领域中开创了一个新时代。作为 Microsoft .NET 平台的一部分，ASP.NET 是一个革命性的编程框架，它实现了企业级 Web 应用程序的快速开发，通过提供最简便、最具有扩张能力的方式，来开发、部署和运行将任何浏览器或客户端设备作为目标的 Web 应用程序。目前，学习和使用 ASP.NET 的计算机软件爱好者和从业人员越来越多，很多学校也将其作为必修课程。

Visual C# 作为 Visual Studio.NET 的原生语言，更是体现了 Visual Studio.NET 的特点。C# 并不是从 C 或 C++ 派生而来，而是 Microsoft 公司彻底从头开发的一种可视化的完全面向对象的编程语言。C# 汇集了 Microsoft 公司的技术精华，具有简单、先进、类型安全、面向对象和可视化等特点。本书采用 C# 来设计 ASP.NET 程序，对于熟悉 C++ 或 Java 的编程者来说，不会有太大的距离感，初学者也可选择 C# 支持。若是熟练的 VB 程序员，不妨采用 VB.NET 支持，那样更轻车熟路。

本书主要介绍了 ASP.NET 的概念、ADO.NET 数据库访问、Web 服务器控件的使用和内部对象的应用，最后给出了一个综合应用实例，详细讲述了使用 ASP.NET 开发电子商务网站的过程和设计技巧。学习完本书的内容，读者可以掌握使用 C# 语言编写 ASP.NET 的 Web 应用程序的方法。全书共分 8 章，其结构安排如下：

第 1 章 ASP.NET Web 应用程序开发入门，介绍了 .NET 的基本概念和在 Visual Sudio.NET 中设计 ASP.NET Web 应用程序的方法，还介绍了在 ASP.NET 中使用的 HTML 语法和 JavaScript 脚本的基本知识。

第 2 章 ASP.NET 程序构成和处理过程，介绍了 Web Form 网页的构成和 Page 类的属性、方法和事件。

第 3 章 ADO.NET 数据库访问技术，介绍了 ADO.NET 的基本知识，ASP.NET 中访问数据库的方法。

第4章 基本的 Web 服务器控件应用, 详细介绍了基本的 Web 服务器控件的使用方法, 编写了大量的应用实例。

第5章 数据综合处理控件应用, 详细介绍了数据综合处理控件的使用方法, 通过大量实例讲解3种数据综合处理控件的应用。

第6章 多网页项目和内部对象, 详细介绍了网页间的相互关系和数据交换方法, 讲述了内部对象的使用方法。

第7章 高级 Web 服务器控件应用, 介绍了 XML 的基本知识, 详述了 AdRotator 广告控件和 TreeView 树型控件等两种高级 Web 服务器控件的使用方法和应用技巧。

第8章 综合应用实例——网上图书商城, 通过该实例介绍了电子商务网站的基本知识, 详细讲解了使用 ASP.NET 开发电子商务网站的过程和方法, 给出了完整的程序代码。

本书作为高职高专院校计算机及其相关专业的教材, 也可以作为相关领域技术人员的参考资料。

本书写作中融合了作者多年的软件开发和教学经验, 加入了近年来对 ASP.NET 的理解和实践。本书力求做到深入浅出、联系实际, 使读者通过实例来学习和掌握 ASP.NET 这个先进的 Web 应用程序开发工具。参加本书编写工作的有李德奇(主编), 梁洁婷(副主编), 喻丕珠, 李蓓蓓, 何帆。由于水平和经验的局限, 本书的不足和纰漏之处在所难免, 恳请广大读者批评指正。在本书的策划和编写过程中, 得到了彭勇及其他同行的帮助和支持, 在此深表感谢。

编者

2007年2月

E-mail: hnldq@163.com

# 目 录

29	2.1.1	Web Form 网页的构成	1
32	2.1.2	ASP.NET 程序的处理过程	1
33	2.1.3	事件和事件处理程序	2
36	2.2	Page 类	2
36	2.2.1	Page 类的常用属性	2
37	2.2.2	Page 类的常用方法	2
38	2.2.3	Page 类的常用事件	2
38		本章小结	2
39		习题	2
40	第 1 章	ASP.NET Web 应用程序开发入门	1
40	1.1	.NET 框架和公共语言运行库	1
40	1.1.1	.NET Framework 结构	1
41	1.1.2	公共语言运行库	2
42	1.1.3	中间语言 (MSIL) 和即时编译 (JIT)	2
44	1.1.4	无用内存单元收集器 (GC)	3
47	1.2	ASP.NET 程序的系统需求	4
47	1.2.1	IIS 对 Web 程序的支持	4
47	1.2.2	安装 ASP.NET	5
51	1.3	VS.NET 集成开发环境	6
52	1.3.1	起始页	6
60	1.3.2	创建 C# 的 ASP.NET 应用程序项目	7
60	1.3.3	设置网页设计状态的网格布局模式	8
	1.3.4	加入页面元素	9
	1.3.5	编译和运行程序	10
63	1.4	HTML 视图	10
63	1.4.1	网页结构	10
63	1.4.2	表单与对象标记	11
64	1.4.3	HTML 语法	13
66	1.5	使用 JavaScript 脚本	16
66	1.5.1	弹出信息框	17
70	1.5.2	弹出确认框	20
70	1.5.3	打开和关闭窗口	22
69	1.5.4	打开模态对话框	25
75		本章小结	27
75		习题	28
75	第 2 章	ASP.NET 程序构成和处理过程	29
18	2.1	Web Form 网页	29

2.1.1	Web Form 网页的构成	29
2.1.2	ASP.NET 程序的处理过程	32
2.1.3	事件和事件处理程序	33
2.2	Page 类	36
2.2.1	Page 类的常用属性	36
2.2.2	Page 类的常用方法	37
2.2.3	Page 类的常用事件	38
	本章小结	38
	习题	39
<b>第 3 章</b>	<b>ADO.NET 数据库访问技术</b>	<b>40</b>
3.1	ADO.NET 概述	40
3.1.1	ADO.NET 的概念和对象模型	40
3.1.2	管理提供者类	41
3.1.3	一般性数据库类	42
3.1.4	管理提供者类的属性和方法	44
3.2	ASP.NET 程序对数据库的访问	47
3.2.1	建立数据库	47
3.2.2	添加 ASP.NET 用户并设置访问权限	47
3.2.3	在应用程序中连接数据库	51
3.2.4	在 Web Form 网页中显示数据	59
	本章小结	61
	习题	61
<b>第 4 章</b>	<b>基本的 Web 服务器控件应用</b>	<b>63</b>
4.1	控件概述	63
4.1.1	HTML 控件	63
4.1.2	Web 服务器控件	64
4.1.3	HTML 控件和 Web 窗体控件的选用	65
4.1.4	控件使用概要	66
4.2	数据输入与提交	67
4.2.1	Label 控件	67
4.2.2	TextBox 控件	69
4.2.3	Button 控件	72
4.2.4	登录页面设计	72
4.2.5	注册页面设计	75
4.2.6	取回密码页面设计	79
4.2.7	修改密码页面设计	81

4.3	信息展示	84
4.3.1	DropDownList 控件	84
4.3.2	ListBox 控件	86
4.3.3	Image 控件	87
4.3.4	信息展示页面设计	87
4.3.5	ListBox 控件应用示例	91
4.4	信息收集	94
4.4.1	CheckBox 控件	95
4.4.2	CheckBoxList 控件	95
4.4.3	RadioButton 控件	97
4.4.4	RadioButtonList 控件	97
4.4.5	信息收集页面设计	98
4.5	数据验证	101
4.5.1	非空字段验证	102
4.5.2	数据类型和数据比较验证	104
4.5.3	数据范围验证	108
4.5.4	正则表达式验证	108
4.5.5	控制错误信息的显示形式	109
4.5.6	停用验证	111
4.5.7	综合验证页面设计	111
	本章小结	118
	习题	119
<b>第 5 章</b>	<b>数据综合处理控件应用</b>	<b>121</b>
5.1	Repeater 控件	121
5.1.1	模板	122
5.1.2	Repeater 控件的项模板	122
5.1.3	Repeater 控件的交替项模板	125
5.1.4	Repeater 控件的分隔模板	126
5.1.5	Repeater 控件的页眉和页脚模板	128
5.2	DataList 控件	129
5.2.1	常用属性与事件	129
5.2.2	自动套用格式	131
5.2.3	使用模板编辑器	131
5.2.4	使用属性生成器	135
5.2.5	在 DataList 控件中展示文本和图片	137
5.2.6	在 DataList 控件中选择数据	141
5.2.7	在 DataList 控件中编辑数据	145



48	5.2.8 在 DataList 控件中删除数据记录	152
48	5.2.9 在 DataList 控件中分页显示	155
48	5.2.10 DataList 控件应用示例	159
78	5.3. DataGrid 控件	165
78	5.3.1 使用 DataGrid 控件的一般步骤	166
10	5.3.2 数据字段列设计	168
10	5.3.3 超级链接列设计	170
20	5.3.4 按钮列设计	172
20	5.3.5 编辑列和删除列设计	178
70	5.3.6 模板列设计	185
70	5.3.7 分页功能	191
80	本章小结	197
101	习题	198
501		
72	第 6 章 多网页项目和内部对象	200
801	6.1 多网页项目	200
801	6.1.1 向项目中添加网页	200
1001	6.1.2 页面间的跳转	201
1111	6.1.3 使用查询字符串在页面间交换数据	203
1111	6.1.4 使用 Session 对象在页面间交换数据	204
8111	6.1.5 添加网页属性在页面间交换数据	205
9111	6.2 内部对象	209
1511	6.2.1 内部对象概述	209
1511	6.2.2 状态管理和 ViewState	212
1511	6.2.3 Application 对象	213
1551	6.2.4 Session 对象	214
1551	6.2.5 Server 对象	215
2511	6.3 内部对象应用示例	216
2511	6.3.1 使用 Session 对象保护网页	216
2511	6.3.2 使用 Application 和 Session 统计在线人数	220
2501	本章小结	221
2511	习题	222
1311		
72	第 7 章 高级 Web 服务器控件应用	223
1321	7.1 AdRotator 广告控件	223
1371	7.1.1 XML 文档简介	223
1411	7.1.2 AdRotator 控件	226
1421	7.1.3 AdRotator 控件应用举例	227

7.2 TreeView 树型控件	230
7.2.1 加入 IEWeb 控件到 VS.NET 中	231
7.2.2 初步使用 TreeView 控件	232
7.2.3 动态添加和删除 TreeNode 节点	234
7.2.4 获取用户选择的节点	237
7.2.5 TreeView 控件应用示例	238
7.2.6 从 XML 文件读取节点数据	242
本章小结	244
习题	244
<b>第 8 章 综合应用实例——网上图书商城</b>	<b>246</b>
8.1 电子商务网站开发过程	246
8.1.1 系统分析	246
8.1.2 系统实现	248
8.1.3 系统测试与运行	249
8.2 网上图书商城总体设计	249
8.2.1 系统功能设计	249
8.2.2 数据库设计	252
8.2.3 全局数据设计	255
8.3 前台页面设计	257
8.3.1 登录和公告页面设计	257
8.3.2 修改密码页面设计	267
8.3.3 取回密码页面设计	271
8.3.4 修改会员信息页面设计	276
8.3.5 注册页面设计	284
8.3.6 购物页面设计	294
8.3.7 购物车页面设计	312
8.3.8 图书详情页面设计	324
8.3.9 查看定单页面设计	329
8.3.10 查看留言页面设计	337
8.3.11 撰写留言页面设计	343
8.4 后台管理设计	348
8.4.1 框架 Frameset1.htm 设计	349
8.4.2 TreeView.aspx 页面设计	350
本章小结	353
习题	353

# 第 1 章 ASP.NET Web 应用程序开发入门

Microsoft.NET 作为新一代的程序运行平台，在应用程序开发领域中开创了一个新时代。ASP.NET 是 Microsoft.NET 的一个组成部分，担当了 Internet 应用的重要任务，特别是在 Web 应用程序和 Web 服务程序的开发和运行上，体现了方便、快捷和高效的特点，受到业界的普遍追捧。

通常所说的 .NET，指的是 .NET Framework，它管理和支持 .NET 程序的执行；而 Visual Studio .NET 则是目前最优秀的应用程序集成开发环境之一，为开发 ASP.NET 应用程序提供了极大的便利。学习 ASP.NET，首先需要了解 .NET Framework 和 Visual Studio .NET 的基本知识，掌握 Web 应用程序开发的基本技术。

在这一章将学习到：

- .NET Framework 框架的基本知识
- ASP.NET 程序的系统需求
- VS.NET 中设计 ASP.NET 程序的初步知识
- HTML 和 JavaScript 基础

## 1.1 .NET 框架和公共语言运行库

### 1.1.1 .NET Framework 结构

.NET Framework 称为 .NET 框架，是一种托管的、类型安全的代码执行环境。.NET Framework 管理程序执行的各个方面：启动与运行代码，给予其相应权限，为数据和指令分配内存，对不再需要的内存进行回收管理。图 1.1 显示了 .NET 的层次结构。

图 1.1 中底部的两层被 .NET 封装，意味着在 .NET 中编程时，托管代码中不能调用 Win32 API 函数，也不能使用指针。

图中的第 3 层和第 4 层称为公共语言运行库（Common Language Runtime）和 .NET 框架类库（.NET Framework Class Library），它们是 .NET 的核心基础。

在公共语言运行库和 .NET 框架类库之上有 ADO.NET、ASP.NET 和 Web Services 等基类库，这个庞大的基类库为编程任务提供了对象模型和服务。.NET 基类库提供的大多数类型是安全的和可扩展的，并可以继承这些基类而派生出新类来。

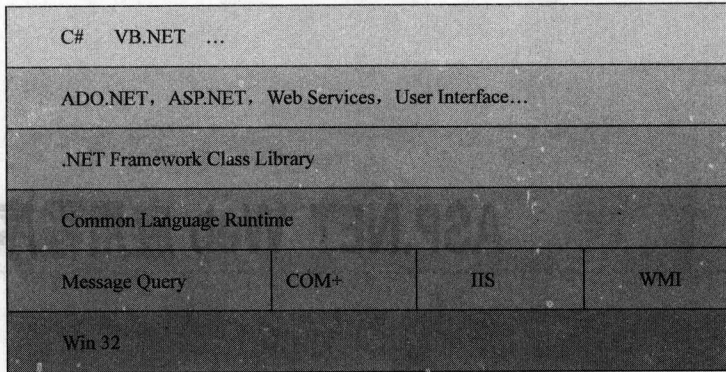


图 1.1 .NET 的层次结构

整个结构的顶层是 Visual Studio .NET 的应用程序开发工具,支持像 C#、VB.NET 和 Visual C++ .NET 等 Microsoft 编程语言,也可以是其他公司的 Visual Studio .NET 编译器。

### 1.1.2 公共语言运行库

公共语言运行库 (Common Language Runtime, 简称为 CLR), 它负责管理和执行 .NET 框架代码, 符合 Visual Studio .NET 编译器规则的代码在执行时需要 CLR 的支持, 这些代码称为托管代码 (Managed Code)。CLR 的管理工作包括以下几项:

- 确定加载代码的方式和时机, 并管理对象在内存中的分配。代码执行时, CLR 将决定代码所需要的类和方法, 并且按照需要将原先编译过的中间语言 (IL) 进行即时编译 (JIT), 形成可执行代码。
- 对托管所占用的内存进行管理, 通过垃圾收集器 (GC) 实现无用内存的回收。
- 确保代码 (由 CLR 执行的代码) 的类型安全。CLR 负责确保以安全的方式使用类和其他类型。
- 通过使用基于异常处理机制的通用错误处理方式, 对托管代码中的错误进行处理和传递。
- 维护 CLR 和应用程序的安全性。在 .NET 框架中, 安全性有两种形式: 代码访问的安全性和角色的安全性。前者用于确保在安全的上下文中执行代码, 后者用于控制对系统资源的访问。

### 1.1.3 中间语言 (MSIL) 和即时编译 (JIT)

当编译 .NET 程序时, 应用程序不是被直接编译成二进制代码, 而是被编译成中间语言 (Microsoft Intermediate Language, 简称 MSIL), 这是一种能被 CLR 理解的相当低级的指令集。事实上, MSIL 是由一个或多个可执行文件和 DLL 文件组成的程序集, 这个程序集中至少包含有一个作为应用程序入口点的可执行文件。

在 Visual Studio.NET 环境下, 各种语言 (例如 C# 和 VB.NET) 开发的应用程序, 都会被编译成运行时环境的通用语言——中间语言, 当应用程序加载后, CLR 根据需要将其再次编

译成可执行代码。这种编译称为即时编译 (Just-in-Time, 简称 JIT)。MSIL 有点像 Java 语言的字节代码, 不同之处在于, Java 的字节代码在执行时逐字节解释, 而 .NET 的中间语言则会被以模块为单位编译成本地二进制代码。MSIL 代码的每一模块仅被编译一次, 当程序执行到 MSIL 代码时, JIT 编译器将在执行前编译它并且将它作为本地二进制代码存储在内存中。这就是说, 不需要运行的代码不会被编译, 执行过的代码不需要再次编译, 因而整个应用程序的性能得到了最大的优化。

由于中间语言在 .NET 下的通用性, 就使得在 Visual Studio.NET 下开发程序时, 一种语言可以使用另一种语言开发的类。例如, 用 C# 开发的应用程序可以引用 VB.NET 的类, 原因是 VB.NET 的类被编译成 MSIL, C# 应用程序也被编译成 MSIL, 它们在 CLR 中自然是相通的。

### 1.1.4 无用内存单元收集器 (GC)

无用存储单元收集器简称为 GC (Garbage Collection)。有了 GC, 意味着当一个对象不再被使用时, .NET 框架能够自动地回收这个对象所使用的内存, 这个过程被称为垃圾回收。

垃圾回收的原理来自于 .NET 的数据类型与内存分配机制。 .NET 将数据类型分为值类型和引用类型两大类。值类型的例子包括基元类型, 例如: 整型 (int)、布尔型 (bool)、字符型 (char) 等, 还有一些用户定义的结构型 (struct) 和枚举型 (enum) 等; 引用类型包括类、接口、委托和数组。值类型与引用类型的主要区别在于变量数据的访问方式, 若要理解这个区别, 需要一些内存分配的背景知识。

应用程序在内存中的数据主要存放在两个区域: 栈 (Stack) 和堆 (Heap)。堆栈其实是一块内存, 其一端为堆, 另一端为栈, 是应用程序为保存程序数据而保留的内存空间。

栈供函数调用使用, 当某函数 fn1 被调用时, 其所有的非静态变量分配在栈中, 称为压栈; 当又有函数 fn2 被调用时, fn2 也被压栈, 这时 fn1 被压在 fn2 的底下。当对 fn2 的调用结束时, fn2 的所有变量被从栈中清除, 称为弹出。等到 fn1 调用结束时, fn1 也将被弹出。所以栈是“先进后出”的机制。而堆则是为创建可重用的对象而保留的。

所有的值类型变量在栈上分配内存, 当值类型的变量脱离了作用域时, 它就会被撤销, 收回其内存。创建引用类型对象时, 对象的数据保存在堆中, 但在栈中有一个指向这个对象的引用。由于对象是可重用的, 一个对象可能有不止 1 个的引用存在。当一个引用的生命期结束时, 这个引用将被从栈中弹出, 这时并不能去撤销堆中与之联系的那个对象, 因为可能还有别的引用与之联系着。GC 的作用就是不时地搜索堆中对象的引用, 当发现某个对象无任何引用与之联系时, 断定这个对象是不必要的, 就可将其从堆中撤销, 收回其占用的堆内存。

在正常情况下, 垃圾回收是一个低优先级的线程。当处理器时间没有被更重要的任务占用时, 它才会运行。但是当内存不够时, 垃圾回收线程的优先级被提升, 使无用内存被迅速回收, 以缓解内存的紧张。此后, 垃圾回收器线程的优先级又被降低了。

得益于垃圾回收机制的应用, 程序员不必像原先那样使用 new 操作后, 时时记得使用 delete 操作。这样一来, 也使类的析构函数的任务大大减轻, 内存溢出的可能性几乎不存在了。

采用这种非确定性的内存回收方法, 是为了尽量提高应用程序的性能, 并为应用程序提

供出错较少的运行环境。但这样做也是要付出代价的，鉴于这种垃圾回收机制不能确定对象何时被回收，所以无法控制何时执行一个类的析构函数，也因此析构函数不能包括赖以在指定时间上运行的代码。实际上，需要较高资源开销的类通常需要实行 Dispose()方法，在该类不再被使用时，显式地释放资源。

## 1.2 ASP.NET 程序的系统需求

首先，ASP.NET 程序必须运行于 .NET Framework 框架之下，以取得程序运行平台的支持；其次，来自客户端对 ASP.NET 程序的请求必须通过 IIS 的允许，以取得访问的权限。因此，运行 ASP.NET 程序的计算机必须安装有 .NET Framework 和 IIS。此外，若要在此计算机上开发 ASP.NET 应用程序，通常需要安装集成开发环境 Visual Studio.NET，简称 VS.NET。当然，若直接安装 VS.NET，则 .NET Framework 框架会一并被安装。

### 1.2.1 IIS 对 Web 程序的支持

因特网信息服务器 (Internet Information Server，简称 IIS)，用来管理来自客户端对 Web 应用程序或 Web 服务程序的请求，任何形式的 Web 程序必须在它的管理之下才允许运行，当然 ASP.NET 程序也不能例外。IIS 是 Windows 2000 及以上版本操作系统的内建组件，若你的操作系统不是服务器版本，请按以下步骤来安装它。

① 依次单击【开始】→【设置】→【控制面板】→【添加或删除程序】，启动“Windows 应用程序向导”，如图 1.2 所示。

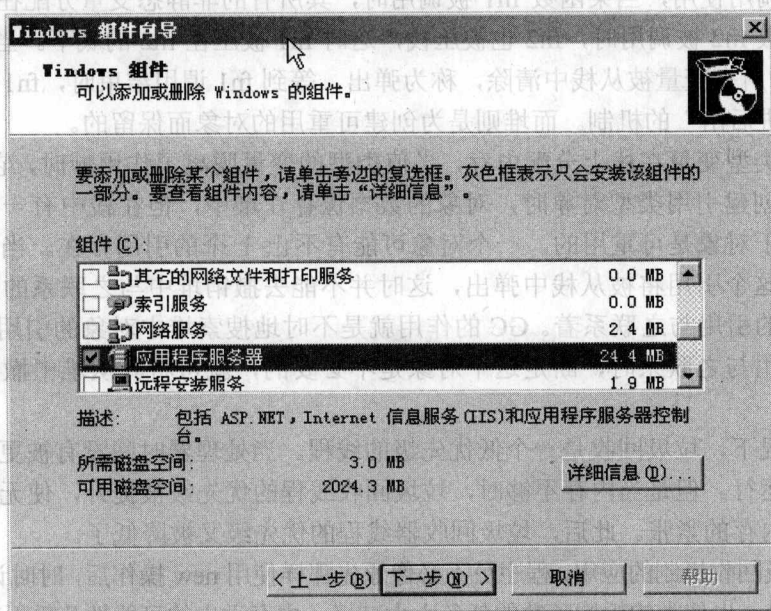


图 1.2 Windows 应用程序向导

② 勾选“应用程序服务器”后，单击【详细信息】，启动“应用程序服务器”，如图 1.3 所示。

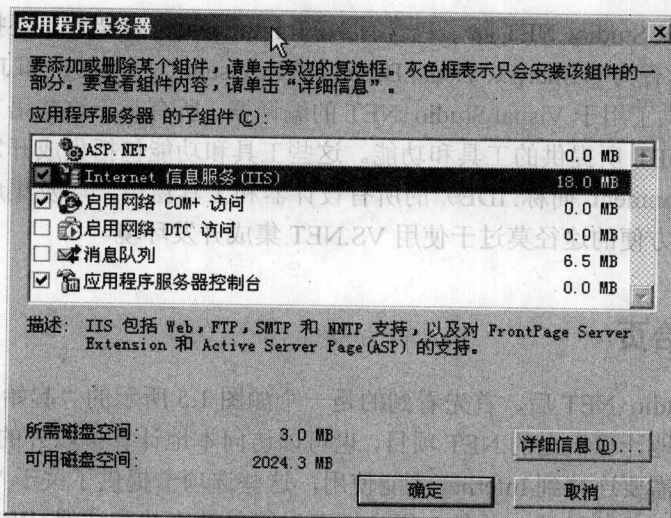


图 1.3 在应用程序服务器中选择 IIS

③ 勾选“Internet 信息服务 (IIS)”后，单击【确定】后即可安装 IIS。

## 1.2.2 安装 ASP.NET

为了能够创建 ASP.NET 程序，必须安装 ASP.NET。其安装过程与安装 IIS 的做法是类似的。

① 重复安装 IIS 的第一步和第二步。

② 在应用程序服务器中勾选“ASP.NET”后，单击【确定】即可安装 ASP.NET，如图 1.4 所示。

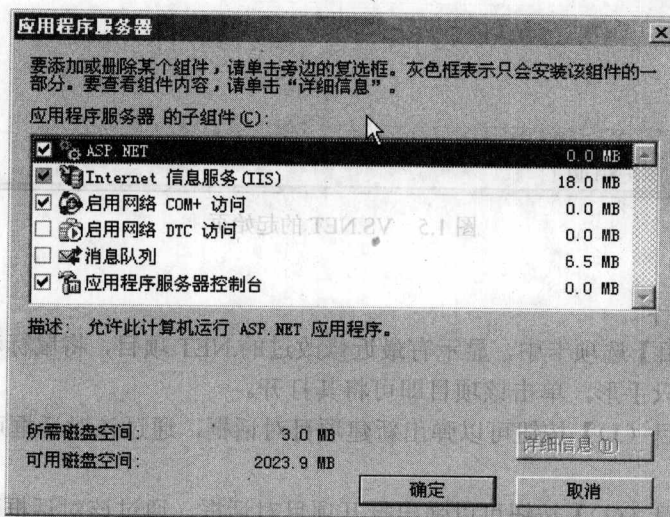


图 1.4 在应用程序服务器中选择 ASP.NET

## 1.3 VS.NET 集成开发环境

Microsoft Visual Studio .NET 除了包含有 .NET Framework 框架外，还提供对多种编程语言的支持，如 C#(读音为 C Sharp)、Visual Basic .NET、Visual C++ .NET 和 J# 等。除 Microsoft 外，其他公司也提供了用于 Visual Studio .NET 的编译器。所有基于这些语言的编程都可以使用 Visual Studio .NET 所提供的工具和功能。这些工具和功能包括集成开发环境 (Integrated Development Environment 简称 IDE) 的所有设计器和工具窗口，以及集成帮助系统。开发 ASP.NET 程序，最方便的途径莫过于使用 VS.NET 集成开发环境。

### 1.3.1 起始页

启动 Visual Studio .NET 后，首先看到的是一个如图 1.5 所示的“起始页”界面，通过该页面既可以使用本地计算机上的 .NET 项目，也可以访问本地计算机以外的信息和服务。起始页上的有些选项卡需要连接到 Internet 才能使用，这些选项卡提供了关于 Visual Studio .NET 的最新信息和最新下载的连接。此外，用户还可以简单地查找和注册 Web Service。若启动 Visual Studio .NET 后看不到起始页，可以单击帮助菜单中的【显示起始页】；若想隐去起始页，可单击起始页上的【×】关闭起始页。

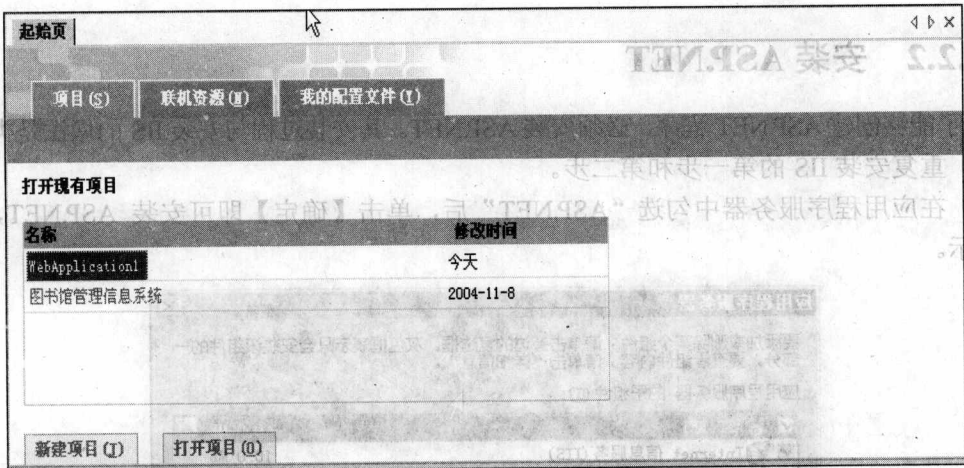


图 1.5 VS.NET 的起始页

#### 1. 项目选项卡

起始页的【项目】选项卡中，显示有最近修改过的 .NET 项目，将鼠标移动到某项目标题上，光标会变成一只手形，单击该项目即可将其打开。

单击【新建项目 (J)】按钮可以弹出新建项目对话框，通过该对话框可以建立一个 .NET 新项目。

单击【打开项目 (O)】按钮可以弹出打开项目对话框，通过该对话框可以打开一个已经存在的 .NET 项目。



图 1.2.4 联机资源选项卡

当本地计算机连接到 Internet 后，可以使用起始页中的【联机资源】选项卡，通过该选项卡可以访问到外部资源。例如，可以获得 Visual Studio .NET 新增功能的信息，联机搜索到 MSDN 数据库，下载到与 Visual Studio .NET 开发相关的内容和代码示例等。特别地，可以从【XML Web services】选项中搜索当前项目所需要的 XML Web Service。使用【Web 宿主】，还可以轻松将自己的程序发布出去。

### 3. 我的配置文件选项卡

通过【我的配置文件】选项卡，开发人员可以按照自己的喜好来配置 Visual Studio .NET 用户界面。通过选择不同的配置文件，开发人员可以根据自己对 C#、C++、VB 等开发环境的熟悉程度，调整工具栏和其他窗口，也可以将键盘映射和窗口布局设定到 VB 6.0 或 Visual C++ 6.0 的风格。

## 1.3.2 创建 C# 的 ASP.NET 应用程序项目

学习 ASP.NET 程序设计总需要从创建第一个项目开始。单击菜单【文件】→【新建】→【项目】，打开新建项目对话框，如图 1.6 所示。

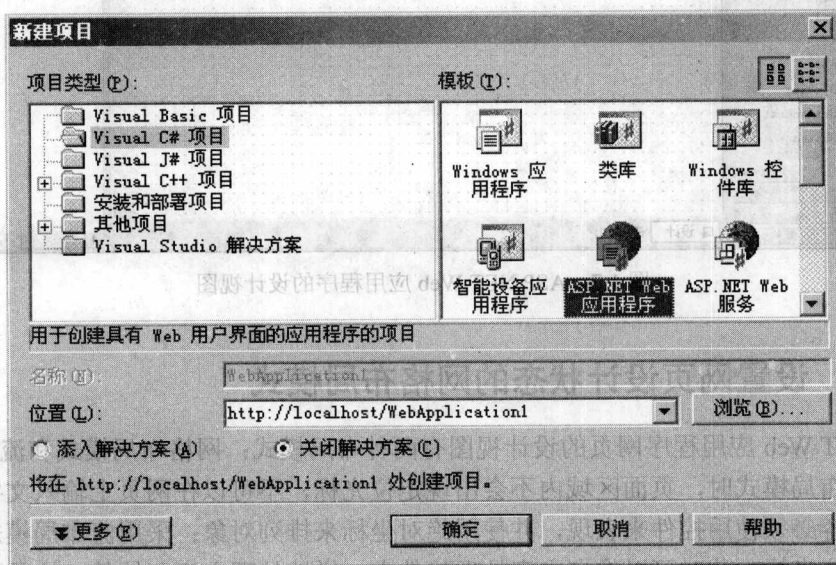


图 1.6 新建项目对话框

在左边的【项目类型】列表框中选择【Visual C#项目】，在右边的【模板】列表框中选择“ASP.NET Web 应用程序”。

在位置框中显示了项目的位置及名称为“http://localhost/WebApplication1”，起步学习 ASP.NET 编程时一般不要修改这个默认的位置及名称，待熟练后再去修改。单击【确定】按钮，就建立了一个 C# 语言支持的 ASP.NET 应用程序项目。该项目默认的名称为 WebApplication1，默认的虚拟路径为 http://localhost，对应的物理路径为 C:\Inetput\wwwroot。