

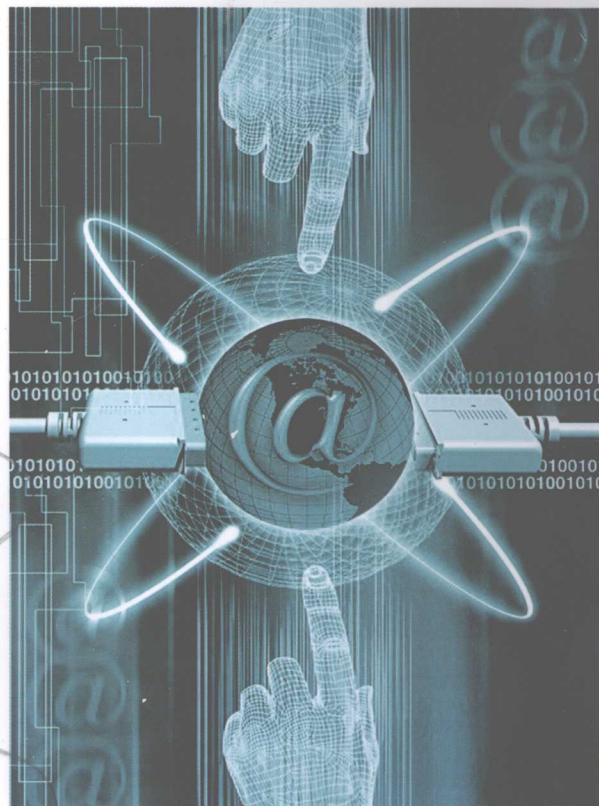
高等学校计算机应用规划教材

ASP.NET 2.0

动态网站开发基础教程

(C# 2005 篇)

- ◆ ASP.NET 2.0 基础
- ◆ HTML 和脚本语言
- ◆ C# 2005 语言基础
- ◆ ASP.NET 2.0 常用对象
- ◆ 服务器控件
- ◆ ADO.NET 数据库编程
- ◆ ADO.NET 数据库高级操作
- ◆ 导航与用户控件
- ◆ ASP.NET Web 服务
- ◆ 配置 ASP.NET 应用程序



唐植华 郭兴峰 编著



清华大学出版社

内 容 简 介

ASP.NET 是由微软公司推出的新一代 Web 开发架构, ASP.NET 技术的推出简化了网络开发人员的工作, 提高了工作效率。

本书详细介绍了使用 C# 语言进行 ASP.NET 2.0 动态网站开发的各种技术。全书共分 10 章, 主要介绍了 ASP.NET 2.0 基础、HTML 和脚本语言、C# 2005 语言基础、ASP.NET 2.0 常用对象、服务器控件、ADO.NET 数据库编程、ADO.NET 数据库高级操作、导航与用户控件、ASP.NET Web 服务、配置 ASP.NET 应用程序等相关技术。

本书理论与实践相结合, 语言通俗易懂, 结构清晰, 突出 ASP.NET 2.0 在开发动态网站方面的强大功能, 可作为高等院校计算机相关专业的教材, 也可供广大使用 ASP.NET 进行动态网站开发的技术人员参考。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET 2.0 动态网站开发基础教程(C# 2005 篇)/唐植华, 郭兴峰 编著. —北京: 清华大学出版社, 2008.6
(高等学校计算机应用规划教材)

ISBN 978-7-302-17511-7

I. A… II. ①唐…②郭… III. ①主页制作—程序设计—高等学校—教材②C 语言—程序设计—高等学校—教材 IV. TP393.092 TP312

中国版本图书馆 CIP 数据核字(2008)第 061046 号

责任编辑: 王 定

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

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: 北京市人民文学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 24 字 数: 554 千字

版 次: 2008 年 6 月第 1 版 印 次: 2008 年 6 月第 1 次印刷

印 数: 1~5000

定 价: 35.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 025010-01

前 言

随着网络技术的飞速发展，网络应用系统的需求越来越大，ASP.NET 2.0 技术的推出简化了网络开发人员的工作，提高了网络人员的工作效率。ASP.NET 2.0 技术使用 Visual Studio 2005 开发环境，该环境能够与微软操作系统平台和数据库系统平台高度融合，因此开发人员把越来越多的目光投向了该开发环境。

本书共分为 11 章，系统、全面地介绍了基于 C# 语言的 ASP.NET 2.0 网站开发的基础知识以及开发方法和技巧，结构清晰，案例丰富，内容完整。具体包括的内容如下：

第 1 章介绍了 .NET Framework 和 ASP.NET 的基本概念、Web Form 的原理、ASP.NET 2.0 的新特性、ASP.NET 网站的结构、ASP.NET 2.0 的开发环境，以及如何使用 Visual Studio 2005 集成开发环境创建基于 C# 语言的 ASP.NET Web 应用程序和发布 ASP.NET Web 应用程序。

第 2 章介绍了开发 Web 页面的基础语言 HTML 语言及开发动态网页的脚本语言和 JavaScript 语言。

第 3 章介绍了 C# 语言基础，包括数据类型、运算符、控制语句、字符串处理、类和对象、委托与事件以及 C# 2005 的新特性等。

第 4 章详细介绍了 ASP.NET 中几个常用的内置对象，包括 Response 对象、Request 对象、Server 对象、Application 对象、Session 对象和 Cookie 对象。

第 5 章主要介绍了内部控件、ASP.NET 2.0 新控件、增强控件和验证控件的使用方法。

第 6 章介绍了 ADO.NET 的常用对象、如何连接数据源、DataSet 对象如何操作数据库，以及 XML 的基本知识。

第 7 章介绍了数据绑定技术，以及如何使用 Repeater 控件、DataList 控件、DetailsView 控件、FormView 控件和 GridView 控件。

第 8 章介绍了如何利用导航控件、主题、母版页和用户控件来统一并强化页面的布局与外观。

第 9 章介绍了 Web 服务的基本概念、Web 服务的创建和使用，以及如何在 Web 服务中传递数据。

第 10 章介绍了如何配置 Web.config 和 Global.asax 文件，以及如何使用 ASP.NET 的缓存模式。

本书从 ASP.NET 基础知识讲起，语言通俗易懂，并配有很多实例和插图，使读者对每一章所讲述的内容都有深刻的理解并加以巩固，可作为高等院校计算机相关专业的教材，也可供广大 ASP.NET 动态网站开发人员参考。

本书主要由唐植华、郭兴峰编写，其中第1~8章由唐植华编写，第9章和第10章由郭兴峰编写。参与本书编写和修改的还有叶明、崔宁、卢宏、汪昔玉、卫平峰、程冬丁、王勤、张锐、汪小锋、李葵、叶浩、肖飞、宋海剑、林勇、朱衡等同志，在此致以诚挚的谢意！

由于时间仓促，加之作者水平有限，书中不足之处在所难免，敬请读者批评指正。

编者

2008年2月

目 录

第 1 章 ASP.NET 2.0 基础	1
1.1 .NET Framework 2.0 简介	1
1.1.1 概述	1
1.1.2 公共语言运行库	2
1.1.3 .NET Framework 类库	3
1.2 ASP.NET 的基本概念	4
1.3 Web Form 原理	6
1.4 ASP.NET 网站的结构	7
1.4.1 ASP.NET 网站的布局	8
1.4.2 网站文件的类型	9
1.4.3 ASP.NET 网站路径	12
1.4.4 ASP.NET 网站中共享代码 文件夹	14
1.5 ASP.NET 2.0 开发环境	16
1.5.1 安装和配置 IIS	16
1.5.2 安装 Visual Studio 2005	19
1.6 创建第一个 ASP.NET Web 应用程序	20
1.7 本章小结	27
1.8 思考与练习	28
第 2 章 HTML 和脚本语言	29
2.1 HTML 基础	29
2.1.1 基本标记	29
2.1.2 超链接	36
2.1.3 表格	38
2.1.4 表单	41
2.1.5 图像	45
2.1.6 框架	47
2.2 JavaScript 语言	51
2.2.1 数据类型	52
2.2.2 运算符	53

2.2.3 控制语句	58
2.2.4 函数	63
2.3 本章小结	64
2.4 思考与练习	65
第 3 章 C# 2005 语言基础	66
3.1 数据类型	66
3.1.1 值类型	67
3.1.2 引用类型	73
3.1.3 语句	75
3.1.4 标识符与关键字	76
3.1.5 变量和常量	77
3.1.6 装箱和拆箱	78
3.2 运算符	79
3.2.1 算术运算符	79
3.2.2 赋值运算符	80
3.2.3 关系运算符	81
3.2.4 逻辑运算符	82
3.2.5 条件运算符	83
3.2.6 对象创建运算符	83
3.2.7 运算符的优先级	84
3.3 控制语句	85
3.3.1 分支语句	85
3.3.2 循环语句	89
3.3.3 异常处理语句	93
3.4 字符串处理	95
3.4.1 使用 String 和 StringBuilder	95
3.4.2 格式化字符串	97
3.5 类和对象	100
3.5.1 定义类	100
3.5.2 定义成员	101
3.5.3 对象的创建与回收	104
3.5.4 封装性、继承性和多态性	106

3.5.5 定义接口和抽象类	113	5.1.4 HyperLink 控件	155
3.6 委托与事件	115	5.1.5 ImageButton 控件和 LinkButton 控件	156
3.6.1 委托	116	5.1.6 TextBox 控件	157
3.6.2 事件	117	5.1.7 CheckBox 与 CheckBoxList 控件	159
3.7 C# 2005 的新特性	120	5.1.8 DropDownList 控件	163
3.7.1 泛型	120	5.1.9 ListBox 控件	165
3.7.2 匿名方法	125	5.1.10 RadioButton 与 Radio ButtonList 控件	167
3.7.3 迭代器	126	5.1.11 Panel 控件	171
3.8 本章小结	127	5.1.12 Table 控件	173
3.9 思考与练习	128	5.2 新增控件	175
第 4 章 ASP.NET 2.0 常用对象	129	5.2.1 BulletedList 控件	175
4.1 Response 对象	129	5.2.2 ImageMap 控件	178
4.1.1 Response 对象的属性	129	5.2.3 MultiView 和 View 控件	181
4.1.2 Response 对象的方法	130	5.2.4 Wizard 控件	181
4.2 Request 对象	132	5.2.5 FileUpload 控件	183
4.2.1 Request 对象的属性	132	5.2.6 HiddenField 控件	185
4.2.2 Request 对象的方法	133	5.2.7 Substitution 控件	186
4.3 Server 对象	134	5.3 增强控件	188
4.3.1 Server 对象的属性	134	5.3.1 Calendar 控件	188
4.3.2 Server 对象的方法	135	5.3.2 AdRotator 控件	190
4.4 Application 对象	136	5.4 验证控件	193
4.4.1 Application 对象的属性	137	5.4.1 ASP.NET 验证控件概述	193
4.4.2 Application 对象的方法	137	5.4.2 RequiredFieldValidator 控件	194
4.5 Session 对象	139	5.4.3 CompareValidator 控件	196
4.5.1 Session 对象的属性	139	5.4.4 RangeValidator 控件	198
4.5.2 Session 对象的方法	140	5.4.5 RegularExpressionValidator 控件	199
4.6 Cookie 对象	141	5.4.6 CustomValidator 控件	200
4.6.1 Cookie 对象的属性	142	5.4.7 ValidationSummary 控件	202
4.6.2 Cookie 对象的方法	142	5.5 本章小结	203
4.7 本章小结	147	5.6 思考与练习	203
4.8 思考与练习	147	第 6 章 ADO.NET 数据库编程	207
第 5 章 服务器控件	150	6.1 ADO.NET 基础	207
5.1 内部控件	150		
5.1.1 Label 控件	150		
5.1.2 Button 控件	151		
5.1.3 Image 控件	154		

6.1.1	什么是 ADO.NET	207	7.1.6	使用 XmlDataSource 数据源 控件	270
6.1.2	ADO.NET 常用对象	208	7.2	复杂数据绑定控件	273
6.2	使用 ODBC .NET Data Provider	216	7.2.1	使用 Repeater 控件	274
6.2.1	ODBC .NET Data Provider 简介	216	7.2.2	使用 DataList 控件	277
6.2.2	连接 ODBC 数据源	217	7.2.3	使用 DetailsView 控件	281
6.3	使用 OLE DB .NET Data Provider	219	7.2.4	使用 FormView 控件	285
6.3.1	OLE DB .NET Data Provider 简介	219	7.3	GridView 控件	291
6.3.2	连接 OLE DB 数据源	220	7.3.1	绑定 GridView 数据	295
6.4	DataSet 对象	221	7.3.2	利用“自动套用格式”设置 页面效果	296
6.4.1	DataSet 对象简介	221	7.3.3	利用 GridView 进行分页	297
6.4.2	插入记录	228	7.3.4	利用 GridView 进行排序	298
6.4.3	删除记录	230	7.3.5	利用 GridView 进行选择	299
6.4.4	更新记录	232	7.3.6	使用 GridView 控件模板列	301
6.5	XML	234	7.3.7	使用 GridView 控件 进行编辑	302
6.5.1	XML 基础	234	7.4	本章小结	304
6.5.2	创建 XML 文件	241	7.5	思考与练习	304
6.5.3	从 DataSet 读取 XML 数据	246	第 8 章	导航与用户控件	307
6.5.4	使用 XML 填充 DataSet	247	8.1	导航控件	307
6.6	本章小结	248	8.1.1	Menu 控件	307
6.7	思考与练习	249	8.1.2	TreeView 控件	313
第 7 章	ADO.NET 数据库高级操作	251	8.2	母版页	317
7.1	数据绑定	251	8.2.1	母版页界面设计	318
7.1.1	简单数据绑定	251	8.2.2	使用母版页	319
7.1.2	使用 ObjectDataSource 数据源控件	253	8.3	统一站点主题	321
7.1.3	使用 SqlDataSource 数据源 控件	259	8.3.1	主题设计	321
7.1.4	使用 AccessDataSource 数据 源控件	265	8.3.2	主题的应用	326
7.1.5	使用 SiteMapDataSource 数据源控件	268	8.4	用户控件	327
			8.4.1	概述	328
			8.4.2	创建用户控件	328
			8.4.3	用户控件的使用	330
			8.5	本章小结	332
			8.6	思考与练习	332

第 9 章 ASP.NET Web 服务 334

9.1 Web 服务概述 334

9.1.1 Web 服务的概念 334

9.1.2 Web 服务的基础结构 335

9.1.3 Web 服务的组成 336

9.1.4 Web 服务的调用过程 339

9.2 Web 服务的创建和使用 340

9.2.1 创建 ASP.NET Web 服务 340

9.2.2 使用 ASP.NET Web 服务 343

9.3 在 Web 服务中传送数据 346

9.3.1 用 Web 服务传送简单数据 346

9.3.2 Web 服务和 DataSet 348

9.4 本章小结 350

9.5 思考与练习 350

第 10 章 配置 ASP.NET 应用程序 353

10.1 配置 Global.asax 文件 353

10.1.1 Global.asax 文件的结构 353

10.1.2 Global.asax 文件的应用 356

10.2 配置 Web.config 文件 358

10.2.1 Web.config 文件的结构 359

10.2.2 使用 Web.config 文件
存放常量 361

10.2.3 网站的安全性配置 362

10.2.4 Web.config 文件的
其他配置 364

10.3 ASP.NET 缓存技术 367

10.3.1 整页输出缓存 368

10.3.2 部分页缓存 370

10.3.3 应用程序数据缓存 372

10.4 本章小结 374

10.5 思考与练习 374

第1章 ASP.NET 2.0基础

ASP.NET 在继续发展,从 Active Server Pages 3.0 到 ASP.NET 1.0 的发布是革命性的,而从 ASP.NET 1.0/1.1 到 ASP.NET 2.0 也是激动人心的。

ASP.NET 1.0/1.1 的引入改变了 Web 编程模式,而 ASP.NET 2.0 大大提高了编程人员的开发效率。ASP.NET 2.0 的主要目标是使用尽可能少的代码,建立强大、安全、动态的应用程序。

本章重点内容:

- .NET Framework 和 ASP.NET 的基本概念
- Web Form 的原理
- ASP.NET 网站的结构
- ASP.NET 2.0 的开发环境
- 如何创建和发布 ASP.NET Web 应用程序

1.1 .NET Framework 2.0 简介

.NET Framework 是 .NET 的核心部分,它提供了 .NET 应用程序所需的核心服务,是开发配置与运行分布式的 Web 服务和 Web 应用的基础平台。

1.1.1 概述

.NET Framework 是支持生成、运行下一代应用程序和 XML Web Services 的内部 Windows 组件。.NET Framework 旨在实现下列目标:

- 提供一个一致的面向对象的编程环境,而无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行的。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码(包括由未知的或不完全受信任的第三方创建的代码)执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与任何其他代码集成。

.NET Framework 具有两个主要组件：公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。可以将运行库看作一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且还强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，可以使用它开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面 (GUI) 应用程序，也包括基于 ASP.NET 所提供的最新创新的应用程序(如 Web 窗体和 XML Web Services)。

1.1.2 公共语言运行库

.NET Framework 提供了一个称为公共语言运行库的运行环境，它运行代码并提供使开发过程更轻松的服务。公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。

运行库的安全性取决于包括托管组件的来源(如 Internet、企业网络或本地计算机)在内的一些因素，托管组件被赋予不同程度的信任。这意味着即使用在同一活动应用程序中，托管组件既可能能够执行文件访问操作、注册表访问操作或其他须小心使用的功能，也可能不能够执行这些功能。运行库强制实施代码访问安全。例如，用户可以相信嵌入在网页中的可执行文件能够在屏幕上播放动画或唱歌，但不能访问他们的个人数据、文件系统或网络。这样，运行库的安全性功能就使通过 Internet 部署的合法软件能够具有特别丰富的功能。运行库还通过实现称为通用类型系统(CTS)的严格类型验证和代码验证基础结构来加强代码可靠性。CTS 确保所有托管代码都是可以自我描述的。各种 Microsoft 和第三方语言编译器生成符合 CTS 的托管代码。这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。

运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务，但是它从不解释托管代码。一种称为实时(JIT)编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时，内存管理器排除了出现零碎内存的可能性，并增大了内存引用区域以进一步提高性能。

运行库可由高性能的服务器端应用程序[如 Microsoft SQL Server 和 Internet 信息服务 (IIS)]承载。此基础结构使在享受支持运行库宿主的行业最佳企业服务器的优越性能的同时，能够使用托管代码编写业务逻辑。

若要使公共语言运行库能够向托管代码提供服务，语言编译器必须生成一些元数据来描述代码中的类型、成员和引用。元数据与代码一起存储；每个可加载的公共语言运行库可移植执行(PE)文件都包含元数据。公共语言运行库使用元数据来完成以下任务：查找和加载类，在内存中安排实例，解析方法调用，生成本机代码，强制安全性，以及设置运行时上下文边界。

有了公共语言运行库，就可以很容易地设计出对象能够跨语言交互的组件和应用程序。也就是说，用不同语言编写的对象可以互相通信，并且它们的行为可以紧密集成。例如，可以定义一个类，然后使用不同的语言从原始类派生出另一个类或调用原始类的方法。还可以将一个类的实例传递到用不同的语言编写的另一个类的方法。这种跨语言集成之所以成为可能，是因为基于公共语言运行库的语言编译器和工具使用由公共语言运行库定义的通用类型系统，而且它们遵循公共语言运行库关于定义新类型以及创建、使用、保持和绑定到类型的规则。

1.1.3 .NET Framework 类库

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类型集合。该类库是面向对象的，并提供用户的托管代码可从中导出功能的类型。这不但使 .NET Framework 类型易于使用，而且还减少了学习 .NET Framework 的新功能所需要的时间。此外，第三方组件可与 .NET Framework 中的类无缝集成。

例如，.NET Framework 集合类实现一组可用于开发用户的集合类的接口。用户的集合类将与 .NET Framework 中的类无缝地混合。

正如对面向对象的类库所希望的那样，.NET Framework 类型使得能够完成一系列常见编程任务(包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务)。除这些常见任务之外，类库还包括支持多种专用开发方案的类型。例如，可使用 .NET Framework 开发下列类型的应用程序和服务：

● 控制台应用程序

- Windows GUI 应用程序(Windows 窗体)
- ASP.NET 应用程序

● XML Web Services

● Windows 服务

例如，Windows 窗体类是一组综合性的可重用的类型，它们大大简化了 Windows GUI 的开发。如果要编写 ASP.NET Web 窗体应用程序，可使用 Web 窗体类。

.NET Framework 包括类、接口和值类型，它们可加速和优化开发过程并提供对系统功能的访问。为便于语言之间进行交互操作，.NET Framework 类型是符合 CLS 的，并因此可在任何编程语言中使用，只要这种语言的编译器符合公共语言规范 (CLS)。

.NET Framework 类型是生成 .NET 应用程序、组件和控件的基础。.NET Framework 包括的类型执行下列功能：

- 表示基础数据类型和异常
- 封装数据结构
- 执行 I/O
- 访问关于加载类型的信息
- 调用 .NET Framework 安全检查

- 提供数据访问、多客户端 GUI 和服务器控制的客户端 GUI

.NET Framework 包含一组丰富的接口以及抽象类和具体类,提供了上百个类来处理各种操作,从简单的数学运算和字符串操作到在桌面应用中创建对话框及在浏览器中创建 Web 页面等的高层操作。可以按原样使用这些具体的类,或者在多数情况下从这些类派生自己的类。若要使用接口的功能,既可以创建实现接口的类,也可以从某个实现接口的 .NET Framework 类中派生类。下面介绍一下 .NET Framework 类库中的命名约定和基本类型的命名空间。

(1) 命名约定

.NET Framework 类型使用点语法命名方案,该方案隐含了层次结构的意思。此技术将相关类型分为不同的命名空间组,以便可以更容易地搜索和引用它们。全名的第一部分(最右边的点之前的内容)是命名空间名,全名的最后一部分是类型名。例如, `System.Collections.ArrayList` 表示 `ArrayList` 类型,该类型属于 `System.Collections` 命名空间。`System.Collections` 中的类型可用于操作对象集合。

此命名方案使扩展 .NET Framework 的库开发人员可以轻松创建分层类型组,并用一致的、带有提示性的方式对其进行命名。库开发人员在创建命名空间的名称时应使用以下原则:

“公司名称.技术名称”

例如, `Microsoft.Word` 命名空间就符合此原则。

利用命名模式将相关类型分组为命名空间是生成和记录类库的一种非常有用的方式。但是,此命名方案对可见性、成员访问、继承、安全性或绑定无效。一个命名空间可以被划分在多个程序集中,而单个程序集可以包含来自多个命名空间的类型。程序集为公共语言运行库中的版本控制、部署、安全性、加载和可见性提供外形结构。

(2) System 命名空间

`System` 命名空间是 .NET Framework 中基本类型的根命名空间。此命名空间包括表示由所有应用程序使用的基础数据类型的类: `Object`(继承层次结构的根)、`Byte`、`Char`、`Array`、`Int32` 和 `String` 等。在这些类型中,有许多与编程语言所使用的基元数据类型相对应。当使用 .NET Framework 类型编写代码时,可以在应使用 .NET Framework 基础数据类型时使用编程语言的相应关键字。

`System` 命名空间包含基本类和基类,这些类定义常用的值和引用数据类型、事件和事件处理程序、接口、属性和异常处理。

其他类提供的服务支持数据类型转换、方法参数操作、数学运算、远程和本地程序调用、应用程序环境管理和对托管与非托管应用程序的监控。

1.2 ASP.NET 的基本概念

ASP.NET 是一种统一的 Web 平台,它提供了生成企业级应用程序所必需的全部服务。

下面介绍 ASP.NET 中的一些基本概念, 包括 ASP.NET 页框架、Web Form、ASP.NET 应用程序、XML Web 服务、配置和部署等。

1. ASP.NET 页框架

ASP.NET 页框架是一种编程框架, 它在 Web 服务器上运行以动态地生成和管理 Web 窗体页。ASP.NET 页框架创建了传统客户端/服务器 Web 交互的抽象模型, 使开发人员能够使用支持快速应用程序开发和面向对象编程(OOP)的传统方法和工具来进行应用程序编程。

ASP.NET 页框架为响应在服务器上运行的代码中的客户端事件提供统一的模型, 从而使开发人员不必考虑基于 Web 的应用程序中固有的客户端和服务器隔离的实现细节。该框架还会在页处理生命周期中自动维护页及该页上控件的状态。

2. Web Form

Web Form 又称为 Web 窗体, Web Form 提供了窗体设计器、编辑器、控件和调试功能, 这些功能结合在一起, 能够为浏览器和 Web 客户端设备快速地生成基于服务器的可编程序用户界面。Web 开发人员可以访问许多事件和属性, 就像在标准的 Windows 环境中编程一样, 因而极大地提高了程序的开发效率。ASP.NET 会把 WebForm 转换为独立于浏览器和平台核心的 HTML 页面, 使得 Web 窗体页在任何浏览器或客户端设备上都可以运行并显示。

3. ASP.NET 应用程序

ASP.NET 应用程序是指可以从特定位置(该特定位置指 Web 应用程序服务器上的虚拟目录及其子目录)调用的所有文件、页、处理程序、模块和可执行代码。ASP.NET 应用程序和 ASP.NET 服务相对应, 主要包括以下主要元素:

- 项目管理功能。可以对创建的 ASP.NET 应用程序项目进行管理, 管理哪些文件需要编译以及哪些文件需要部署。
- 用户界面。开发人员可以在 Web Form 中添加显示信息, 并最终输出到客户端浏览器上显示。
- 组件。组件的使用可以提到代码的重用性, 在 Web 应用程序中, 可以将这些组件创建为 XML Web 服务, 然后再在 Web 应用程序中进行调用。
- 数据。大多数应用程序都要求某种形式的数据访问。在 ASP.NET Web 应用程序中, 可使用 ADO.NET 数据服务访问数据库。

4. XML Web 服务

ASP.NET 支持 XML Web 服务。Web 服务是微软公司 .NET 计划的基础, 一个 Web 服务就是一个应用 Web 协议的可编程的应用程序逻辑。利用 Web 服务, 应用程序可以使用 HTTP 和 XML 消息等标准在基于 Web 的应用程序之间交换信息, 从而跨越防火墙移动数据。用任何语言编写的、使用任何组件模型和在任何操作系统上运行的程序都能够访问

Web 服务。Web 服务不仅可以为 ASP.NET 应用程序提供服务，而且还为具有 Web 访问权限的任何应用程序提供服务。

5. 配置和部署

ASP.NET 比较简单，配置设置存储在基于 XML 的文件中，对这些文件进行编辑即可完成配置。Visual Studio.NET 为向本地或远程服务器部署应用程序提供了一种简化的机制。只需将文件复制到指定的应用程序目录即可部署这些文件。

1.3 Web Form 原理

Web Form 又称为 Web 窗体或者 Web 表单，主要用来根据特定的信息生成动态的页面。Web Form 把 Visual Basic 表单的高生产性的优点带到了网络应用程序的开发中来，将 ASP.NET 的代码和页面的一般脚本分隔开来，这些分隔成块的代码可以像组件一样被重复使用，从而提高了开发的效率。

Web 窗体页由两部分组成：视觉元素(HTML、服务器控件和静态文本)和该页的编程逻辑。Visual Studio.NET 将这两个组成部分分别存储在一个单独的文件中。可视元素在一个 .aspx 文件中创建，而代码位于一个单独的类文件中，该文件称作代码隐藏类文件.aspx.vb(VB.NET)或.aspx.cs(C#)。Web Form 的后缀为 aspx，当一个浏览器请求一个 aspx 文件时，Web Form 将被 CLR 编辑器编译。当用户再次访问该页面时，由于 aspx 页面已经被编译过了，所以 CLR 会直接执行编译过的代码。

一般来讲，Web 窗体页的生命周期类似于在服务器上运行的任何 Web 进程的生命周期。ASP.NET 页框架通过不同的阶段对 Web 窗体页进行处理。在 Web 窗体处理的每一阶段，都可能会引发事件，并且将运行与该事件相对应的任何事件处理程序。开发人员可以在不同的事件中添加处理代码，以实现不同功能。下面分别介绍 Web Form 页面生命周期中各个阶段和事件定义。

1. ASP.NET 页框架初始化阶段

在 ASP.NET 页框架初始化阶段，将引发该页的 Page_Init 事件，并还原该页和控件视图状态。在此事件期间，ASP.NET 页框架还原控件属性和回发数据。

2. 用户代码初始化阶段

用户代码初始化阶段将引发页的 Page_Load 事件。在该阶段可以读取和还原以前存储的值。使用 Page.IsPostBack 属性，检查是否是第一次处理该页，如果是第一次处理该页，则执行初始数据绑定，否则还原控件值。

3. 验证阶段

验证阶段将调用任何验证程序 Web 服务器控件的 Validate 方法来执行该控件的指定验

证, 可以在事件处理程序中测试验证的结果。

4. 事件处理阶段

事件处理阶段将会调用该页中的相应事件处理程序, 处理不同的事件。例如, 如果该页包含验证类型 ASP.NET 服务器控件验证类型, 则在该阶段检查该页和各个验证控件的 IsValid 属性。

5. 清除阶段

清除阶段将调用 Page_Unload 事件, 因为该页已完成显示并准备好被放弃。在该阶段中将执行最后的清除工作, 如关闭文件、关闭数据库连接、放弃对象等。对于一些耗资源的内容, 在这个阶段进行清除是非常有必要的, 否则该内容将在执行垃圾回收前一直保持对资源的占有, 增加了服务器的负荷。

1.4 ASP.NET 网站的结构

若要开发 ASP.NET Web 应用程序(也称为网站), 必须具有以下工具:

- .NET Framework 2.0。若要使用 ASP.NET, 在承载 ASP.NET 网站的计算机上必须安装 .NET Framework 2.0。
- 代码创作环境。可以使用任何文本编辑器(如记事本)创建 ASP.NET 页和类。不过, Microsoft Visual Studio 和其他集成开发环境(IDE)提供了页模板、IntelliSense、代码自动完成和设计时编译等许多编程功能。这些功能可以加速开发过程并提供项目组织功能。
- 一个 Web 服务器[如 Internet 信息服务(IIS)]和将文件写入该 Web 服务器的权限。若要使用 ASP.NET Web 应用程序, 必须使用浏览器向承载应用程序的 Web 服务器发出请求。通常, 使用 IIS 作为 Web 服务器来承载 ASP.NET Web 应用程序。可以通过在计算机上本地运行 IIS 或将页和组件部署到共享服务器来测试应用程序。

如果使用的是 Visual Studio 2005, 则可以使用 ASP.NET Development Server 来测试 ASP.NET 网页。ASP.NET Development Server 在计算机上本地运行页, 而无需安装 IIS。若要将 IIS 作为 Web 服务器使用, 作为运行身份的用户账户必须具有写入应用程序所在目录的权限, 无论该目录是 Web 服务器根的子目录还是虚拟目录。如果安装了 IIS, 则在安装 .NET Framework 时会在 IIS 中注册 ASP.NET ISAPI 扩展名 (Aspnet_isapi.dll)。如果在已经安装了 .NET Framework 的情况下安装 IIS, 则必须使用 Aspnet_regiis.exe 实用工具在 IIS 中注册 ASP.NET。

在使用 Visual Studio 2005 创建 ASP.NET 网站时, 可以包含 ASP.NET 能够识别以进行处理的特定类型的文件。此外, 还可以创建用于特殊用途(如用于存储源代码)的文件夹。

1.4.1 ASP.NET 网站的布局

可以将网站的文件保存在方便应用程序访问的任何文件夹结构中。为了更易于使用应用程序, ASP.NET 保留了某些可用于特定类型的内容的文件和文件夹名称。

可以为应用程序建立默认页, 这将使用户更容易定位到站点。默认页是在用户定位到站点时没有指定特定页的情况下为用户提供的页。例如, 可以创建一个名为 `Default.aspx` 的页, 并将它保存在站点的根文件夹中。如果用户在定位到站点时没有指定特定页, 可以配置应用程序, 以便自动请求 `Default.aspx` 页。可以使用默认页作为站点的主页, 或者在页中写入代码以将用户重定向到其他页。在 Internet 信息服务 (IIS) 中, 默认页是作为网站的属性创建的。

ASP.NET 能识别可用于特定类型的内容的某些文件夹名称。表 1-1 列出了保留的文件夹名称以及文件夹中通常包含的文件类型。应用程序文件夹(App_Themes 文件夹除外)的内容并不在响应 Web 请求时提供, 但可以从应用程序代码进行访问。

表 1-1 ASP.NET 应用程序文件夹表

名 称	说 明
App_Browsers	包含 ASP.NET 用于标识个别浏览器并确定其功能的浏览器定义(.browser)文件 包含作为应用程序一部分进行编译的实用工具类和业务对象(如.cs、.vb 和 .jsl 文件)的源代码。在动态编译的应用程序中, 当对应用程序发出首次请求时, ASP.NET 编译 App_Code 文件夹中的代码。然后在检测到任何更改时重新编译该文件夹中的项。可以在 App_Code 文件夹中放置任意文件类型以创建强类型对象。在应用程序中将自动引用 App_Code 文件夹中的代码。此外, App_Code 文件夹可以包含需要在运行时编译的文件的子目录
App_Code	
App_Data	包含应用程序数据文件, 包括 MDF 文件、XML 文件和其他数据存储文件。ASP.NET 2.0 使用 App_Data 文件夹来存储应用程序的本地数据库, 该数据库可用于维护成员资格和角色信息
App_GlobalResources	包含编译到具有全局范围的程序集中的资源(.resx 和 .resources 文件)。App_GlobalResources 文件夹中的资源是强类型的, 可以通过编程方式进行访问
App_LocalResources	包含与应用程序中的特定页、用户控件或母版页关联的资源(.resx 和 .resources 文件)
App_Themes	包含用于定义 ASP.NET 网页和控件外观的文件集合(.skin 和 .css 文件以及图像文件和一般资源)
App_WebReferences	包含用于定义在应用程序中使用的 Web 引用的引用协定文件(.wsdl 文件)、架构(.xsd 文件)和发现文档文件(.disco 和 .discomap 文件)
Bin	包含要在应用程序中引用的控件、组件或其他代码的已编译程序集(.dll 文件)。在应用程序中将自动引用 Bin 文件夹中的代码所表示的任何类