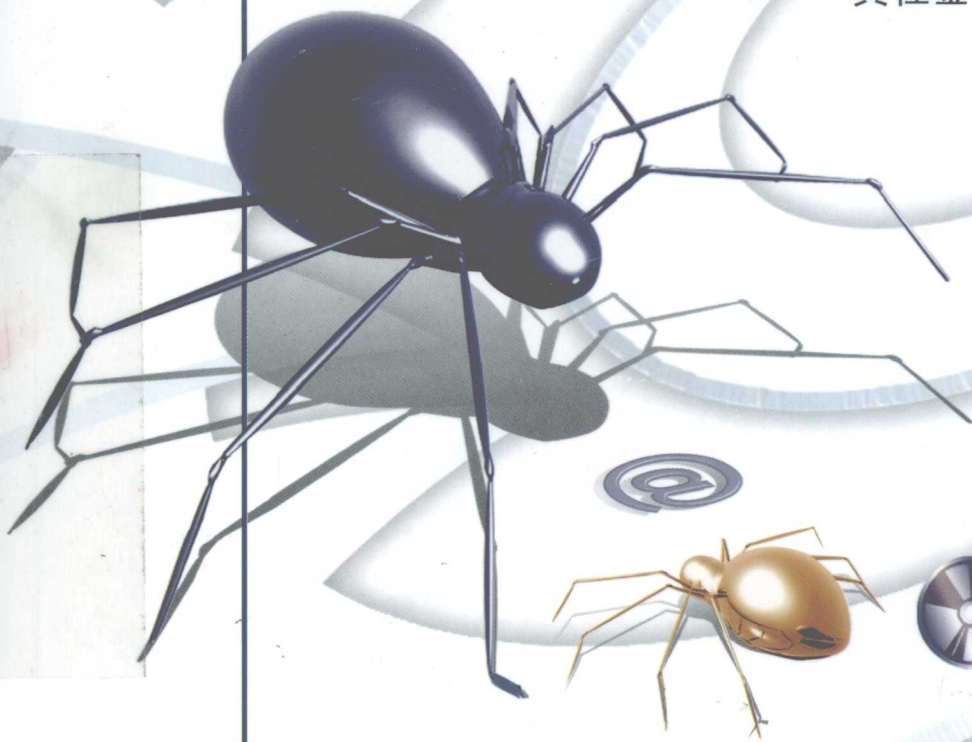


# ASP.NET Ajax

## 网站开发

## 从入门到精通

黄桂金 宋强 王占中 等编著



附  
光  
盘

- 16章完整实例源代码
- 20段全程配音教学视频
- 5个综合实例配置文件和代码



清华大学出版社

TP393.092/1040D

2008

# ASP.NET Ajax

## 网站开发

## 从入门到精通

黄桂金 宋强 王占中 等编著



清华大学出版社  
北京

## 内 容 简 介

ASP.NET Ajax 是 Microsoft 为在 .NET 环境中应用 Ajax 而开发的框架。本书共 16 章, 内容包括 ASP.NET 编程知识、使用 ASP.NET 访问外部数据、Ajax 技术概述、Ajax 核心内容、常用的 ASP.NET Ajax 框架、ASP.NET Ajax 框架中的基础控件和常用的 Ajax 服务器控件。本书最后讲解 5 个典型案例: 聊天室、留言本、通信录、在线学习系统、Blog 系统。本书配套光盘提供了本书源代码(包括 5 个完整的应用程序)和 ASP.NET Ajax 软件配置和使用教学视频。

本书适合于中、高级 ASP.NET Ajax 网站开发人员, 特别适合于有编程基础, 希望全面学习 ASP.NET Ajax 技术, 提高实际应用能力的读者群体。

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 Ajax 网站开发从入门到精通 / 黄桂金, 宋强, 王占中等编著. —北京: 清华大学出版社, 2008. 7

ISBN 978-7-302-17134-8

I. A… II. ①黄…②宋…③王… III. ①主页制作—程序设计②计算机网络—程序设计  
IV. TP393.092 TP393.09

中国版本图书馆 CIP 数据核字(2008)第 032848 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 李红英

出版发行: 清华大学出版社

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

社 总 机: 010-62770175

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

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市李旗庄少明装订厂

经 销: 全国新华书店

开 本: 203×260 印 张: 35.5 字 数: 924 千字

附光盘 1 张

版 次: 2008 年 7 月第 1 版 印 次: 2008 年 7 月第 1 次印刷

印 数: 1~4000

定 价: 68.00 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: 022310-01

# FOREWORD

## 前言

Ajax (Asynchronous JavaScript and XML) 是 Web 2.0 的核心技术, 由多种技术组合而成。使用 Ajax 技术不必刷新整个页面, 可以直接对页面的局部进行更新, 而且可以节省网络带宽, 提高网页加载速度, 从而缩短用户等待时间, 改善用户操作体验。

ASP.NET Ajax 是 Microsoft 为在 .NET 环境中应用 Ajax 而开发的框架。ASP.NET Ajax 客户端有极大的灵活性和可扩展性, 并且可以应用于非 .NET 环境下, 为 Ajax 程序员提供了广阔的开发空间。使用 ASP.NET Ajax 框架, 开发人员几乎不需要额外学习, 就可以开发出自己的 Ajax 程序。

本书以 ASP.NET 为基础详细介绍 Ajax 核心技术以及 ASP.NET Ajax 框架的应用, 并在最后讲解 5 个典型案例。全书包括如下内容。

第 1 章: 对 ASP.NET 2.0 进行简单的概述, 并重点介绍 ASP.NET 2.0 环境的安装和配置, 在本章最后给出一个 ASP.NET 2.0 的案例演示执行过程。

第 2 章: 主要介绍创建 ASP.NET 网站时必须掌握的编程知识, 包括基础语法、各种服务器控件以及 ASP.NET 的内置对象等。

第 3 章: 重点介绍使用 ASP.NET 访问外部数据, 包括 System.IO 命名空间、读写普通文件和二进制文件、文件上传、ADO.NET 对象以及各种数据显示控件, 比如 Repeater 和 GridView 等。

第 4 章: 主要介绍使用 ASP.NET 进行网站开发时的 6 个高级课题, 分别是 GDI+ 绘图、用户控件、WebParts 控件、导航控件、应用程序配置和处理 XML。

第 5 章: 主要对 Ajax 技术进行简单的概述, 包括传统 Web 应用解决方案与使用 Ajax 的区别、Ajax 的开发模式以及运行机制等。

第 6 章: 主要介绍 Ajax 技术的 6 个核心部分, 分别是 XHTML、XML、JavaScript、CSS、DOM 和 XMLHttpRequest。

第 7 章: 重点介绍使用 Ajax 与 ASP.NET 结合开发的基础, 包括客户端发送普通请求和 XML 请求, 以及服务器处理普通响应和 XML 响应等。

第 8 章: 主要介绍常用的 ASP.NET Ajax 框架, 包括 Ajax 框架概述、Magic Ajax .NET 框架、Ajax.NET 框架和 ASP.NET Ajax 框架。

第 9 章: 重点介绍 ASP.NET Ajax 框架中的基础控件, 包括 ScriptManager、ScriptManagerProxy、UpdatePanel、UpdateProgress 及 Timer。

第 10 章: 主要介绍 ASP.NET Ajax 框架中提供的 16 个实用的 Ajax 服务器控件。

第 11 章: 主要介绍 ASP.NET Ajax 框架中提供的另外 14 个实用的 Ajax 服务器控件。

第 12 章: 介绍使用 Ajax 的核心技术, 结合 ASP.NET 制作一个即时响

应、无刷新的聊天室，包括聊天室系统分析、客户端和服务端实现等。

第 13 章：介绍使用 Ajax 核心技术与 ASP.NET 结合非框架的实例——留言本，包括系统的需求分析、数据库设计及配置、发表留言、回复和删除留言等。

第 14 章：介绍使用 ASP.NET Ajax 框架制作一个多用户的通信录，包括实例需求分析、结构设计、数据库和通用类设计、用户管理以及好友管理等。

第 15 章：详细介绍使用 ASP.NET Ajax 框架制作的在线学习系统，包括显示课堂列表、选择教程、浏览视频、用户管理、使用纸条功能以及系统后台管理等。

第 16 章：介绍使用 ASP.NET Ajax 框架实现 Blog 系统的过程，包括系统页面 DIV+CSS 布局、文章列表和显示、管理员登录以及管理文章和分类等。

本书全面介绍 ASP.NET Ajax 面向实践的开发和应用知识，本书具有如下特点。

- **内容全面** 本书是一本大全性质的 ASP.NET Ajax 编程图书，突出介绍 ASP.NET Ajax 面向实际的开发知识。读者学习本书之后，可以全面掌握 ASP.NET Ajax 动态网站的开发实践知识。
- **实例丰富** 全书每个知识点的讲解都配有大量可实际运行的实例，读者可以边学习边实践，快速、全面地掌握 ASP.NET Ajax 的开发方法和技巧。书中最后提供 5 个典型开发案例，覆盖了 ASP.NET Ajax 开发和部署网站的全部过程。
- **本书配套光盘** 提供了本书源代码（包括 5 个完整的应用程序）、ASP.NET Ajax 软件配置和使用教学视频。读者只要按照书中的范例上机练习、举一反三，就可以根据自己的需要开发出功能强大的 Web 动态网站。

本书适合于中、高级 ASP.NET Ajax 网站开发人员，特别适合于有编程基础，希望全面学习 ASP.NET Ajax 技术，提高实际应用能力的读者群体。

除了封面署名人员之外，参与本书编写的人员还有李乃文、张仕禹、夏小军、赵振江、李振山、李文采、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林等。在编写过程中难免会有疏漏，欢迎读者批评指正并与我们联系，帮助我们改正提高。

作者

2007 年 12 月

# CONTENTS

## 目 录

第 1 章 ASP.NET 概述	1
1.1 Internet 与 ASP.NET	1
1.2 ASP.NET 简介	3
1.2.1 .NET Framework 简介	3
1.2.2 ASP.NET 2.0 特点	5
1.3 配置开发环境	7
1.3.1 配置 Web 服务器 IIS	7
1.3.2 配置 Visual Studio 2005	9
1.3.3 熟悉 Visual Studio 2005	11
1.4 ASP.NET 开发实例	13
1.4.1 创建 ASP.NET 应用程序	13
1.4.2 编写实现代码	15
第 2 章 ASP.NET 网页编程基础	18
2.1 C# 基础语法	18
2.1.1 C# 语法结构	18
2.1.2 数据类型	20
2.1.3 表达式和运算符	23
2.1.4 条件语句	25
2.1.5 循环语句	27
2.2 面向对象实现	29
2.2.1 对象和类	29
2.2.2 封装性	33
2.2.3 继承性	35
2.2.4 多态性	37
2.3 ASP.NET 控件	38
2.3.1 ASP.NET 控件基本属性	38
2.3.2 简单控件	39
2.3.3 增强控件	45
2.3.4 验证控件	49
2.4 内置对象	55
2.4.1 Request 对象	55
2.4.2 Response 对象	57
2.4.3 Application 对象	58
2.4.4 Server 对象	59
2.4.5 Session 对象	61
2.4.6 Cookie 对象	62
2.5 Session 购物车	63

<b>第 3 章 ASP.NET 访问外部数据</b> .....	68	4.6.1 System.Xml 命名空间	130
3.1 System.IO 命名空间 .....	68	4.6.2 读取 XML	132
3.1.1 System.IO 命名空间概述 .....	68	4.6.3 创建 XML	136
3.1.2 Directory 类方法 .....	69	<b>第 5 章 Ajax 概述</b> .....	140
3.1.3 File 类方法 .....	71	5.1 传统 Web 应用解决方案 .....	140
3.2 文件操作 .....	74	5.2 新的 Web 开发模式 Ajax .....	142
3.2.1 读写普通文件 .....	74	5.2.1 初识 Ajax .....	142
3.2.2 读写二进制文件 .....	77	5.2.2 Ajax 开发模式 .....	144
3.2.3 文件上传 .....	79	5.3 Ajax 的运行机制 .....	145
3.3 ADO.NET 对象 .....	82	5.4 一个简单的 Ajax 示例 .....	147
3.3.1 ADO.NET 简介 .....	82	<b>第 6 章 Ajax 相关技术简介</b> .....	154
3.3.2 Connection 对象 .....	84	6.1 XHTML .....	154
3.3.3 Command 对象 .....	85	6.1.1 XHTML 概述 .....	154
3.3.4 DataSet 对象 .....	87	6.1.2 XHTML 与 HTML 的区别 .....	157
3.3.5 DataAdapter 对象 .....	88	6.1.3 XHTML 的语法 .....	159
3.3.6 DataReader 对象 .....	89	6.1.4 XHTML 的未来——XHTML 2.0 .....	169
3.4 数据显示控件 .....	90	6.2 CSS .....	170
3.4.1 Repeater 控件 .....	90	6.2.1 CSS 语法 .....	170
3.4.2 DataList 控件 .....	93	6.2.2 引入样式表 .....	174
3.4.3 GridView 控件 .....	96	6.2.3 CSS 属性 .....	176
<b>第 4 章 ASP.NET 高级技术</b> .....	101	6.3 XML .....	182
4.1 GDI+绘图 .....	101	6.3.1 XML 文档 .....	182
4.1.1 System.Drawing 命名空间 .....	102	6.3.2 XML 命名空间 .....	185
4.1.2 绘制图形 .....	106	6.3.3 文档类型定义 (DTD) .....	186
4.2 用户控件 .....	109	6.4 JavaScript .....	191
4.2.1 创建用户控件 MemberLogin .....	109	6.4.1 JavaScript 简介 .....	191
4.2.2 使用 MemberLogin 用户控件 .....	112	6.4.2 JavaScript 的基本结构 .....	192
4.3 WebParts 控件 .....	114	6.4.3 JavaScript 表达式与运算符 .....	193
4.3.1 WebParts 控件概述 .....	114	6.4.4 JavaScript 的流程控制 .....	194
4.3.2 使用 WebParts 控件 .....	115	6.4.5 JavaScript 的函数 .....	197
4.4 导航控件 .....	118	6.4.6 事件驱动及事件处理 .....	197
4.4.1 Menu 控件 .....	118	6.4.7 创建对象 .....	199
4.4.2 SiteMapPath 控件 .....	121	6.4.8 JavaScript 中的对象 .....	201
4.4.3 TreeView 控件 .....	122	6.5 DOM .....	204
4.5 配置 ASP.NET 应用程序 .....	125	6.5.1 DOM 对象 .....	204
4.5.1 Web.config 文件 .....	125	6.5.2 使用 DOM 操作 HTML 文档 .....	207
4.5.2 ASP.NET 缓存 .....	128	6.5.3 DOM 解析 XML 文档 .....	210
4.6 处理 XML .....	130		

6.6 XMLHttpRequest .....	212	9.2.1 ScriptManagerProxy 控件概述 .....	280
6.6.1 XMLHttpRequest 简介 .....	212	9.2.2 ScriptManagerProxy 控件使用实例 .....	281
6.6.2 XMLHttpRequest 的属性和方法 .....	214	9.3 UpdatePanel 控件 .....	284
6.6.3 XMLHttpRequest 与服务器通信 .....	216	9.3.1 UpdatePanel 控件概述 .....	285
6.6.4 XMLHttpRequest 实例 .....	218	9.3.2 UpdatePanel 控件属性与方法 .....	286
<b>第 7 章 Ajax 开发基础</b> .....	<b>221</b>	9.3.3 UpdatePanel 控件使用实例 .....	287
7.1 客户端 HTTP 请求 .....	221	9.4 UpdateProgress 控件 .....	292
7.1.1 HTTP 请求概述 .....	221	9.4.1 UpdateProgress 控件概述 .....	292
7.1.2 普通请求 .....	224	9.4.2 UpdateProgress 控件属性与方法 .....	292
7.1.3 XML 请求 .....	228	9.4.3 UpdateProgress 控件使用实例 .....	292
7.2 服务器端响应 .....	231	9.5 Timer 控件 .....	296
7.2.1 HTTP 响应概述 .....	232	9.5.1 Timer 控件概述 .....	296
7.2.2 处理普通响应 .....	233	9.5.2 Timer 控件属性与方法 .....	297
7.2.3 处理 XML 响应 .....	236	9.5.3 Timer 控件使用实例 .....	298
7.3 一个简单的 Ajax 实例 .....	241	<b>第 10 章 Ajax 实用技能之一</b> .....	<b>305</b>
<b>第 8 章 常用 Ajax 框架</b> .....	<b>248</b>	10.1 禁止自动执行行为 (NoBot) .....	305
8.1 Ajax 框架概述 .....	248	10.2 数值调整器 (NumericUpDown) .....	309
8.2 Magic Ajax.NET 框架 .....	250	10.3 分页项目符号列表 (PagingBulletedList) .....	312
8.2.1 Magic Ajax.NET 简介 .....	250	10.4 密码强度 (PasswordStrength) .....	313
8.2.2 部署 Magic Ajax.NET 框架 .....	251	10.5 弹出控件 (PopupControl) .....	315
8.2.3 Magic Ajax.NET 简单应用 .....	252	10.6 评级 (Rating) .....	317
8.2.4 Magic Ajax.NET 使用技巧 .....	255	10.7 顺序重排列表 (ReorderList) .....	320
8.3 Ajax.NET 框架 .....	259	10.8 可调整大小控件 (ResizableControl) .....	323
8.3.1 部署 Ajax.NET .....	259	10.9 圆角 (RoundedCorners) .....	324
8.3.2 第一个示例 .....	261	10.10 滑块 (Slider) .....	326
8.3.3 Ajax.NET 简单应用 .....	262	10.11 相册播放 (SlideShow) .....	327
8.4 ASP.NET Ajax 框架 .....	268	10.12 选项卡标签 (Tab) .....	329
8.4.1 ASP.NET Ajax 组成部分 .....	268	10.13 水印文本框 (TextBoxWatermark) .....	331
8.4.2 安装 ASP.NET Ajax .....	269	10.14 双模按钮 (ToggleButton) .....	332
8.4.3 配置 ASP.NET Ajax .....	271	10.15 更新面板动画效果 (UpdatePanelAnimation) .....	334
8.4.4 ASP.NET Ajax 简单应用 .....	272	10.16 模式对话框验证器 (ValidatorCallout) .....	335
<b>第 9 章 ASP.NET Ajax 基础控件</b> .....	<b>275</b>	<b>第 11 章 Ajax 实用技能之二</b> .....	<b>338</b>
9.1 ScriptManager 控件 .....	275	11.1 自动输入完成扩展器 (AutoCompleteExtender) .....	338
9.1.1 ScriptManager 控件概述 .....	275		
9.1.2 ScriptManager 控件属性与方法 .....	276		
9.1.3 ScriptManager 控件使用实例 .....	277		
9.2 ScriptManagerProxy 控件 .....	280		



11.2	日历扩展器 (CalendarExtender)	342
11.3	级联下拉菜单 (CascadingDropDown)	344
11.4	可收缩面板 (CollapsiblePanel)	349
11.5	确认按钮 (ConfirmButton)	351
11.6	拖曳面板 (DragPanel)	352
11.7	下拉菜单 (DropDown)	354
11.8	阴影面板 (DropShadow)	356
11.9	动态呈现 (DynamicPopulate)	358
11.10	文本框过滤器 (FilteredTextBox)	361
11.11	停靠菜单 (HoverMenu)	363
11.12	强制弹出 (ModalPopup)	366
11.13	互斥复选框 (MutuallyExclusiveCheckBox)	369
11.14	可折叠面板 (Accordion)	371
<b>第 12 章 聊天室</b> 375		
12.1	系统分析	375
12.2	实现客户端	376
12.2.1	登录页面	377
12.2.2	聊天页面	378
12.2.3	布局样式	380
12.2.4	公共模块	384
12.3	实现服务器端	390
12.3.1	通用类	390
12.3.2	进入聊天室	394
12.3.3	发送消息	397
12.3.4	消息管理	399
12.4	案例演示	401
<b>第 13 章 留言本</b> 406		
13.1	需求分析和系统概述	406
13.1.1	需求分析	406
13.1.2	系统概述	407
13.1.3	数据库设计及配置	408
13.2	客户端实现	410
13.2.1	客户端原理分析	410
13.2.2	通用部分实现	411

13.2.3	首页实现	414
13.2.4	显示留言实现	416
13.2.5	发表留言	420
13.2.6	回复留言	423
13.2.7	删除留言	426
13.3	服务器端实现	427
13.3.1	显示留言实现	427
13.3.2	发表留言实现	430
13.3.3	回复留言实现	432
13.3.4	删除留言实现	433
13.4	案例演示	434
<b>第 14 章 通信录</b> 437		
14.1	系统概述	437
14.1.1	需求分析	437
14.1.2	结构设计	438
14.2	数据库设计	439
14.2.1	设计表	439
14.2.2	设计通用类	441
14.3	系统首页	445
14.3.1	页面布局	445
14.3.2	实现代码	449
14.4	管理组	452
14.4.1	设计布局	452
14.4.2	实现代码	455
14.5	管理用户	457
14.5.1	用户注册	457
14.5.2	用户登录	458
14.5.3	用户退出	461
14.6	管理好友	461
14.6.1	添加好友	461
14.6.2	查看好友	463
14.6.3	修改好友	467
14.6.4	查询好友	471
<b>第 15 章 在线学习系统</b> 472		
15.1	系统概述	472
15.1.1	需求分析	472
15.1.2	结构设计	473

15.2 数据库设计 .....	475	15.5.4 维护作者和纸条 .....	517
15.2.1 设计表和存储过程 .....	475		
15.2.2 设计通用类 .....	479	<b>第 16 章 Blog 系统 .....</b>	<b>520</b>
15.2.3 文件处理类 .....	482	16.1 系统概述 .....	520
15.3 系统首页 .....	487	16.2 数据库设计 .....	522
15.3.1 布局页面 .....	487	16.3 通用模块设计 .....	524
15.3.2 实现代码 .....	490	16.3.1 顶部导航菜单 .....	524
15.3.3 教程分类 .....	493	16.3.2 底部版权及站长信息 .....	526
15.3.4 教程资源列表 .....	496	16.3.3 CSS 样式表 .....	528
15.3.5 打开资源 .....	499	16.4 系统实现 .....	531
15.4 会员功能 .....	502	16.4.1 显示文章列表 .....	532
15.4.1 注册 .....	502	16.4.2 显示文章简介及详细信息 .....	535
15.4.2 登录与退出 .....	505	16.4.3 分类显示文章列表 .....	539
15.4.3 查看信息 .....	507	16.4.4 管理员登录 .....	540
15.4.4 使用纸条 .....	510	16.4.5 文章搜索 .....	543
15.5 系统管理 .....	512	16.4.6 文章管理 .....	545
15.5.1 维护课堂信息 .....	513	16.4.7 分类管理 .....	551
15.5.2 维护教程分类 .....	514	16.4.8 会员管理 .....	554
15.5.3 添加教程 .....	516		

## 第 1 章

### 学习目标 | Objective

在 Internet 迅速发展的今天, ASP.NET 以其简单、快捷、高效的特性, 得到了用户的青睐。特别是 ASP.NET 2.0 的推出, 它是基于 .NET 平台的一个革命性突破, 解决了过去 Web 开发技术中存在的各种不足和局限。与其他动态网页技术相比较, 如 JSP、PHP 等, ASP.NET 具有运行速度快、安全等特点。

本章将从基础开始, 对 ASP.NET 2.0 进行简单的介绍, 并重点介绍 ASP.NET 2.0 开发环境的安装和配置, 在本章最后给出一个 ASP.NET 2.0 的案例演示执行过程。

### 内容摘要 | Abstract

- 了解 .NET Framework 与 ASP.NET 的关系
- 掌握 Web 服务器的安装
- 掌握 Visual Studio 2005 的安装
- 熟悉 Visual Studio 2005 开发环境
- 掌握如何创建 ASP.NET 应用程序
- 了解 ASP.NET 应用程序的结构

## 1.1 Internet 与 ASP.NET

Internet 是一个无限大的网络, 通过它访问的信息和数据可能非常遥远。这对应用程序将有一定的影响。例如, 当用户在一个小的桌面应用程序中访问数据库时, 也许能够完全地锁定数据。但是, 对于通过 Internet 来访问的应用程序来说, 这个策略可能无法行得通。在 Internet 上, 资源的使用比本地应用程序敏感得多。

显示一个请求从离开 Web 浏览器被目标站点接收到之前, 可能会经历大量的服务器。类似地, 服务器的应答在传回时, 也会经历曲折的路由。作为 Internet 基础的网络协议和数据表示方式决定了网络的可靠性并不稳定, 而且服务器上运行的应用程序可能会由多种不同操作系统上运行的一个或多个不同的 Web 浏览器来访问。

用户为了访问 Internet 上的应用程序, 需要 Web 浏览器, 并通过超文本传输协议 HTTP 来与应用程序通信。应用程序通常在某种形式的 Web 服务器上运行, 这种服务器能识别 HTTP 请求, 并判断具体应该由哪一个应用程序来处理请求。“应用程序”在这里是一个很广泛的术语——Web

服务器可能调用一个可执行程序来采取一项行动,也可能使用它自己的内部逻辑或者其他方式来处理请求。不管采取什么方式处理请求,Web 服务器都会向客户端发送一个应答,这时同样会用到 HTTP。HTTP 应答的内容通常表示成 HTML 页,即 Web 页。

HTTP 是一种无连接的协议,这意味着一个请求(或是一个应答)是一个独立的数据包。客户端和 Web 服务器上运行的应用程序之间的一次典型的交换可能涉及到几个请求。例如,用户可能显示一页,输入一些数据,单击按钮,并希望显示结果能相应地发生改变,允许用户输入更多的数据,依次类推。客户端发送到服务器的每个请求都独立于该客户端发送的其他请求以及正在使用同一个服务器的其他任何客户端。但问题在于,客户端请求通常要求某种形式的上下文请求或者状态。

例如,考虑一个常见的情况,一个 Web 应用程序允许用户浏览正在出售的商品。用户可能希望同时买几种商品,把它们都放到一个虚拟的购物车内。对于这种应用程序来说,一个有用的功能是显示购物车中的当前内容。

那么,购物车的内容(也就是客户端的状态)应该放在哪里呢?如果将这些信息放到 Web 服务器上,那么 Web 服务器就必须有能力将不同的 HTTP 请求放到一起,判断哪个请求来自哪个客户端。虽然这是一种可行的解决方案,但它要求服务器执行附加的处理,使客户端请求与状态信息保持一致。另外,还必须维护一个数据库,在不同的客户请求之间,持久性地保存那些状态信息。这个方法的缺点在于,Web 服务器无法保证在保存了状态信息之后,客户端会继续提供另一个请求来使用或者删除这些信息。假如,Web 服务器实时地保存每个客户端的每一个请求信息,那么最终产生的数据库将是非常大的。

另一个解决方案是将状态信息保存在客户端上。Cookie 正是为此而设计的,它允许服务器将必要的信息保存到客户端上的 Cookie 中。这个方案的缺点在于,应用程序必须随同每个 HTTP 请求传输到 Cookie 中的信息,使 Web 服务器能够利用这些信息。除此之外,应用程序还必须保证 Cookie 非常小。Cookie 技术最明显的缺点或许就是用户能随时禁用它,使 Web 服务器无法保存它,同时这将导致无法保存状态信息。

根据上述的这些讨论,可以总结出在设计一个框架来构建并运行 Web 应用程序时,必须解决多个问题。它必须能够完成如下几个功能:

- 支持标准 HTTP。
- 有效地管理客户端状态。

提供工具简化 Web 应用程序的开发。

生成的应用程序可由支持 HTML 的任何浏览器访问。

响应迅速并且易于扩展和部署。

随着 .NET 平台的出现,Microsoft 推出了新一代的 ASP.NET,它的主要特点包括如下 4 点:

(1) 一个使用 Web 窗体的关系式程序模型。Web 窗体中包含与事件处理逻辑区分开的表示逻辑和代码文件。用户可以使用任何支持 .NET 的语言来编写代码,其中包括 C#。ASP.NET Web 窗体在 Web 服务器上编译和缓存,能有效地提升性能。

(2) 提供 Server 控件,这些控件支持服务器端事件,但呈现为 HTML,允许它们在任何 HTML 相容的浏览器中工作。Microsoft 还扩展了许多标准 HTML 控件,允许用户在自己的代码中操作它们。

(3) 提供强大的 Data 控件来显示、编辑和处理来自同一个数据库的数据。

(4) 为了缓存客户端状态, 允许选择在客户端上使用 Cookie、使用 Web 服务器上的一个特殊服务或者使用一个 SQL Server 数据库, 可以使用代码来非常容易地对缓存进行编程。

在 Visual Studio 2005 附带的 .NET Framework 2.0 中, Microsoft 进一步增强了 ASP.NET。设计者实现了大量的增强功能, 以优化吞吐速度和网站的可维护性。其中这一版本中新增的主要特性如下所述:

(1) 使用 Master Pages、Themes 和 WebParts 支持增强的页面设计和布局。可以使用 Master Pages 为应用程序中的所有网页快速提供公共布局。Themes 有助于在网站中保持一致性的外观和感觉, 可以确保所有控件在必要时都以相同的方式显示。WebParts 允许用户创建模块化的网页, 用户可以根据自己的需求来定制。

(2) 利用新的数据源控件将数据绑定到网页。这些新控件允许用户构建出快速、方便显示并易于编辑数据的应用程序。数据源控件支持大量的数据源, 例如 SQL Server、Access、XML 文件、Web 服务以及能够返回数据集的事务处理对象。使用数据源控件, 用户可以采取一致性的方式来处理数据, 用户的所有处理都将独立于数据的来源。

(3) 新增和增强的控件。针对数据的显示和编辑, Microsoft 现在为用户提供了 GridView、DetailsView 和 FormView 控件。可以利用 TreeView 控件来显示层次化的数据, 而且可以使用 SiteMapPath 和 Menu 控件来帮助用户在自己的 Web 应用程序中导航。

(4) 增强的安全特性, 内置了对身份验证和用户授权的支持。可以方便地向用户授权, 允许他们访问自己的 Web 应用程序; 在用户登录时验证其身份及查询用户信息, 以便了解正在访问的网站。可以使用 Login 控件来提示用户输入他们的身份凭据, 并对其进行验证。使用 PasswordRecovery 控件, 则可帮助用户记住或重置他们的密码。

(5) 使用 ASP.NET Web Site Administration 工具来实现增强的网站配置和管理。这个工具提供了许多向导来配置和保护 ASP.NET Web 应用程序。

在本章的其他部分, 将进一步介绍 ASP.NET 应用程序的结构。

## 1.2 ASP.NET 简介

ASP.NET 是统一的 Web 应用程序平台, 它提供了为建立和部署企业级 Web 应用程序所必需的服务。ASP.NET 为面向任何浏览器或设备的更安全的、更强的可升级性、更稳定的应用程序提供新的编程模型和基础结构。

ASP.NET 是 Microsoft .NET Framework 的组成部分, 是一种可以在高度分布的 Internet 环境中简化应用程序开发的计算环境。.NET Framework 包含公共语言运行库 (CLR), 它提供了各种核心服务, 如内存管理、线程管理和代码安全。它也包含 .NET Framework 类库, 这是一个开发人员用于创建应用程序的全面的、面向对象的类型集合。

### 1.2.1 .NET Framework 简介

Microsoft .NET Framework 是一种全新的计算机平台, 它简化了在 Internet 分布式环境下应用程序的开发。.NET Framework 旨在提供保证代码安全执行的面向对象的编程环境, 并消除脚本环

境的性能问题。

.NET Framework 包含两个主要组件：公共语言运行时和 .NET Framework 类库。Microsoft ASP.NET 为运行库提供核心服务，例如，内存管理、线程管理，以及当加强确保安全性和稳定性的限制类型安全和其他代码窗体的准确性时进行的远程处理。代码管理是运行库的基本原则；针对运行库的代码称为“托管代码”，而不针对运行库的代码称为“非托管代码”。.NET Framework 类库是一个综合的、面向对象的、可重复使用类的集合，开发人员可用其创建 ASP.NET 应用程序。

### 1. 公共语言运行库的功能

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。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。它可以用图 1-1 表示。

公共语言运行时负责实施由开发人员设计，用来使 .NET 应用程序变得健壮的各种规则。这些规则包括对数据类型、内存使用和应用程序安全的约束。由于上述所有管理工作都发生在公共语言运行时内，所以即使是编写很差的 .NET 代码，也不可能包含许多常见的错误类型。例如，内存泄漏在托管代码中是不可能的，而且这种保护对开发人员来说是毫无代价的。他们不必编写任何代码就能确信自己的应用程序都将不会包含内存泄漏。

公共语言运行时提供了下列重要的服务：

(1) 公共类型系统 (Common Type System, CTS) 定义了所有 .NET 语言的标准数据类型及其格式。例如，CTS 定义了整型是 32 位大小，还指定了整型值的内部格式。

(2) 公共语言规范 (Common Language Specification, CLS) 定义了语言间互操作性的规定。由于 CLS 定义了规则，一个 .NET 语言创建的类就可以由其他 .NET 语言使用。

(3) 当 .NET 应用程序第一次编译时，编译为一种可以由所有 .NET 语言共享的中间语言。在应用程序执行时，Just-In-Time (JIT) 编译器把中间语言转换为可以在目标计算机上执行的可执行文件。

(4) CLR 管理应用程序的执行，也就是说，CLR 负责在创建和销毁对象时，为其分配和解除分配内存。

(5) 垃圾搜索器 (Garbage Collector, GC) 负责解除分配内存。

### 2. .NET Framework 类库

.NET Framework 类库是一个可重复使用类型的集合，它高度集成了 CLR。该类库是面向对象的，提供了应用程序能够派生附加功能的类型。由于是面向对象的，所以不但可以方便地使用 .NET Framework 类型，而且可以节省学习 .NET Framework 新增功能的时间。另外，第三方的

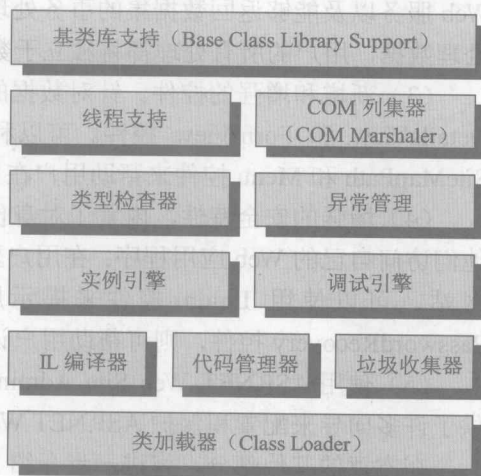


图 1-1 公共语言运行时

组件可以与 .NET Framework 的类紧密地集成。

.NET Framework 类库使用户能够完成一系列常见的编程任务,包括字符串管理、数据收集、数据库连接以及文件访问等。除了这些常见的任务外,类库还包括多种专用开发方案的类型。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 窗体。
- ASP.NET 应用程序。
- XML Web 服务。
- Web 窗体。

.NET Framework 类库包含了 .NET Framework 中定义的所有类型。它们之间的关系如图 1-2 所示。

表 1-1 列出了主要的 .NET Framework 数据类型、每种数据类型的内存需求,以及可能的值。表中列出的内存需求可能会有所变化,这主要取决于 CPU 的体系结构、采用的是 32 位还是 64 位的系统等因素。

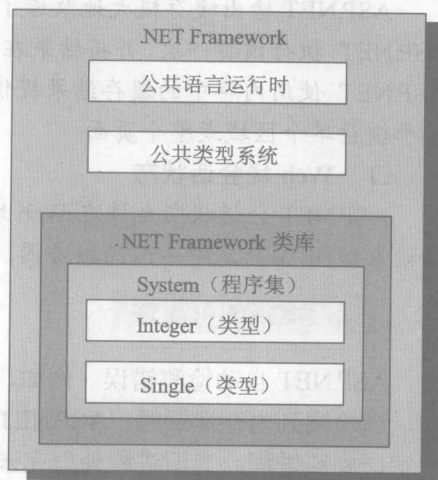


图 1-2 .NET Framework 类库与 .NET Framework 类型的关系

表 1-1 .NET Framework 数据类型

数据类型	内存需求	可能值
Boolean(System.Boolean)	2B	true 或 false
Byte(System.Byte)	1B	0~255
Short(System.Short)	2B	-32768~32767
Integer(System.Integer)	4B	-2147483648~2147483647
Long(System.Long)	8B	-9223372036854775808~9223372036854775807
Single(System.Single)	4B	带有 6 位精度的浮点数
Double(System.Double)	8B	带有 16 位精度的浮点数
Char(System.Char)	2B	0~65535
String(System.String)	根据字符串的长度而定	0~约 20 亿字符
Date(System.Date)	8B	0001 年 1 月 1 日~9999 年 12 月 31 日

## 1.2.2 ASP.NET 2.0 特点

随着 .NET Framework 2.0 的发布,ASP.NET 的版本也被更新到 2.0。它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,它为创建利用 Internet 的网络应用程序提供了全新的编程模型。

### 1. 改进的性能和可伸缩性

#### 编译后执行

ASP.NET 比旧版 ASP.NET 1.1 的运行速度更快,同时保留了 ASP“只需单击保存”的更新模型。无须显式的编译步骤。ASP.NET 自动检测变化,根据需要动态编译文件,并且保存编译结果,以便后续的请求再次使用。动态编译保证了用户的应用程序总是最新的,并且编译后执行使其运

行速度更快。对于多数从传统的 ASP 迁移到 ASP.NET 的应用程序,其处理的页数增长了 3~5 倍。

#### □ 大容量输出缓存

ASP.NET 输出缓存极大地改进了应用程序的性能和可伸缩性。当输出缓存在页面中启用时,ASP.NET 执行该页一次,并将结果在发送至用户之前保存在内存中。当其他用户请求同一页面时,ASP.NET 使用内存中的缓存结果提供服务,而不是重新执行该页面。输出缓存是可配置的,可以用于缓存单个区域或整个页面。

#### □ Web 场会话状态

ASP.NET 会话状态允许在 Web 场中的所有计算机之间共享会话数据。现在,用户可以通过多个请求访问 Web 场中不同的服务器,并且仍然具有完全访问会话数据的权限。

### 2. 增强的可靠性

ASP.NET 自动检测错误(例如,死锁和内存泄漏)并进行恢复以确保应用程序始终可用。例如,当检测到内存泄漏时,ASP.NET 将自动启动新的 ASP.NET 工作进程副本,并将所有的新请求定向到该进程。当旧进程处理完未决的请求后,就会除去该进程并将释放泄漏的内存。

### 3. 部署简单

#### □ “非接触式”应用程序部署

使用 ASP.NET,用户可以通过将其复制到服务器来进行整个应用程序的部署。配置设置将存储在应用程序的 XML 文件中。

#### □ 运行程序的动态更新

ASP.NET 允许用户不必重新启动 Web 服务器而更新已编译的组件。与传统的 COM 组件(这些组件需要部署更新后手动重新启动 Web 服务器)不同,ASP.NET 自动检测更改并使用新的代码启动。

#### □ 迁移路径简单

ASP.NET 可以和传统的 ASP 应用程序一起在 Windows 2000、Windows XP 和 Windows Server 2003 的 IIS 上运行。用户可以一次迁移一个应用程序,甚至是单独的页面。ASP.NET 还允许继续使用现有的传统 COM 业务组件。

### 4. 新的应用程序模型

#### □ XML Web 服务

XML Web 服务允许应用程序通过 Internet 进行通信和数据共享,而不管操作系统和编程语言如何。ASP.NET 使得公开和调用 XML 网络服务变得简单。

#### □ 移动 Web 设备支持

ASP.NET 移动控件允许处理超过 80 台使用 ASP.NET 的移动 Web 设备。用户只需一次性写入应用程序,移动控件就能自动生成请求设备的页面。

### 5. 开发人员的效率

#### □ 简单的编程模型

具有服务器控件(这些控件允许通过比传统 ASP 更少的代码来构建极佳的页面)的 ASP.NET,



使得动态构建真实的 Web 应用程序变得更加容易。

#### □ 灵活的语言选项

ASP.NET 不仅支持 VBScript 和 JScript, 而且支持 .NET 语言, 包括对 Visual Basic.NET、Microsoft C# 和 JScript .NET 的内置支持。

#### □ 丰富的类框架

.NET Framework 类库提供了 4500 多种类, 这些类封装了大量的功能, 诸如 XML、数据访问、文件上传、正则表达式、图像生成、性能监视和日志记录、事务、消息队列和 SMTP 邮件。

7

## 1.3 配置开发环境

到目前为止, 读者已经为进行 ASP.NET 开发有了一定的概念, 下面要进行的就是创建 ASP.NET 应用程序的实践操作了。因为在这之前, 还需要配置 ASP.NET 运行环境, 之后再熟悉 Visual Studio .NET 2005 开发环境。

### 1.3.1 配置 Web 服务器 IIS

ASP.NET 和其他 Web 应用程序一样, 需要在系统中配置运行环境, 安装 ASP.NET 需要安装的组件有:

- IIS 5.0 (Windows 2000) 或 IIS 4.0 (Windows NT Server 4.0) 以上版本 (必需)。
- Internet Explorer 5.5 或者以上版本 (可选)。
- .NET Framework SDK (必需)。

以在 Windows XP Professional 操作系统下配置为例, 首先安装 IIS。通常在默认安装系统完成后, 不包括 IIS 组件。检查和安装 IIS 组件的步骤如下:

(1) 单击【开始】菜单, 选择【设置】|【控制面板】命令, 打开【控制面板】窗口。

(2) 双击【添加和删除程序】图标, 打开【添加或删除程序】窗口, 在窗口左侧单击【添加/删除 Windows 组件】按钮, 弹出如图 1-3 所示的【Windows 组件向导】对话框。

(3) 如果系统中安装过 IIS, 则【Internet 信息服务 (IIS)】复选框是被启用的, 如果没有安装则该复选框没有被选中。单击选中该复选框, 将操作系统的安装光盘插入光驱中, 单击【下一步】按钮, 按照提示操作即可安装 IIS, 如图 1-4 所示。

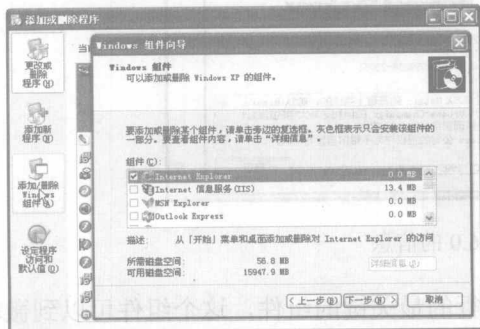


图 1-3 【Windows 组件向导】对话框

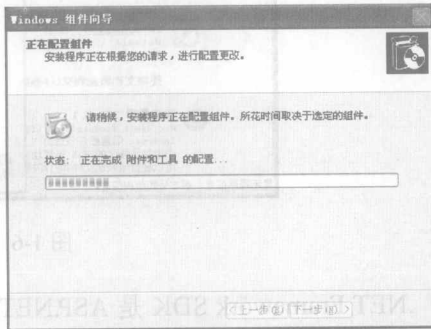


图 1-4 IIS 的安装过程