



高职高专  
网络专业系列规划教材

# ASP.NET程序设计

新世纪高职高专教材编审委员会组编  
主编 屈武江 主审 李德奇



大连理工大学出版社



新世纪

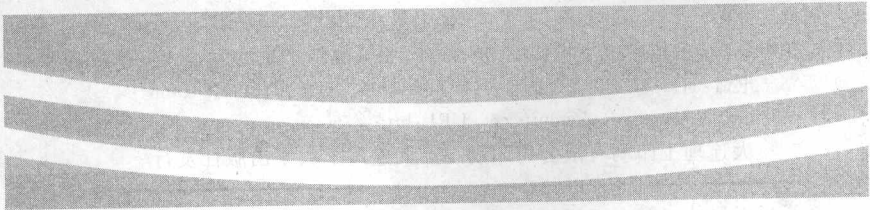
高职高专网络专业系列规划教材

# ASP.NET 程序设计

新世纪高职高专教材编审委员会组编

主 审 李德奇

主 编 屈武江 副主编 姜俊颖 文继权 何保荣 赵春霞 何赤平



ASP.NET CHENGXU SHEJI

大连理工大学出版社

DALIAN UNIVERSITY OF TECHNOLOGY PRESS

林 彦 俊 陈 庆 系 业 专 案 网 专 高 职 高

图书在版编目(CIP)数据



ASP.NET 程序设计 / 屈武江主编. —大连:大连理工大学出版社,2007.9

(高职高专网络专业系列规划教材)

ISBN 978-7-5611-3710-9

I. A… II. 屈… 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 138248 号

李 彦 俊 主 审

平 永 同 董 春 强 荣 景 同 刘 继 文 解 尚 姜 编 主 隔 王 先 显 编 主

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

电话:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:http://www.dutp.cn

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:22.5 字数:400 千字  
附赠光盘一张 印数:1~4000  
2007 年 9 月第 1 版 2007 年 9 月第 1 次印刷

责任编辑:潘弘喆

责任校对:程 亮

封面设计:苏儒光

ISBN 978-7-5611-3710-9

定 价:29.80 元

# 总序

我们已经进入了一个新的充满机遇与挑战的时代,我们已经跨入了21世纪的门槛。

20世纪与21世纪之交的中国,高等教育体制正经历着一场缓慢而深刻的革命,我们正在对传统的普通高等教育的培养目标与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20世纪最后的几年里,高等职业教育的迅速崛起,是影响高等教育体制变革的一件大事。在短短的几年时间里,普通中专教育、普通高专教育全面转轨,以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步,其来势之迅猛,发人深思。

无论是正在缓慢变革着的普通高等教育,还是迅速推进着的培养应用型人才的高职教育,都向我们提出了一个同样的严肃问题:中国的高等教育为谁服务,是为教育发展自身,还是为包括教育在内的大千社会?答案肯定而且惟一,那就是教育也置身其中的现实社会。

由此又引发出高等教育的目的问题。既然教育必须服务于社会,它就必须按照不同领域的社会需要来完成自己的教育过程。换言之,教育资源必须按照社会划分的各个专业(行业)领域(岗位群)的需要实施配置,这就是我们长期以来明乎其理而疏于力行的学以致用问题,这就是我们长期以来未能给予足够关注的教育目的问题。

如所周知,整个社会由其发展所需要的不同部门构成,包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门,等等。每一个部门又可作更为具体的划分,直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标,就不能很好地完成社会分工所赋予它的使命,而教育作为社会分工的一种独立存在就应受到质疑(在市场经济条件下尤其如此)。可以断言,按照社会的各种不同需要培养各种直接有用人才,是教育体制变



改革的终极目的。

随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走研究型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,它从专科层次起步,进而应用本科教育、应用硕士教育、应用博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高等职业教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)研究型人才培养的教育并驾齐驱,还需假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职高专教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高职高专教材编审委员会就是由全国 100 余所高职高专院校和出版单位组成的旨在以推动高职高专教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职高专教材的特色建设为己任,始终会从高职高专教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职高专教材市场需求的广阔视野,以其创新的理念与创新的运作模式,通过不断深化的教材建设过程,总结高职高专教学成果,探索高职高专教材建设规律。

在微观层面上,我们将充分依托众多高职高专院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高职高专教材编审委员会在推进高职高专教材建设事业的过程中,始终得到了各级教育主管部门以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意;也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高职高专教材编审委员会

2001年8月18日

# 前 言

《ASP.NET 程序设计》是新世纪高职高专教材编审委员会组编的网络专业系列规划教材之一。

ASP.NET 不是 ASP 的简单升级,而是 Microsoft 公司推出的新一代 Web 编程语言。它是微软发展的新的体系结构.NET 的一部分,为 Web 编程提供了稳定的性能、优秀的升级性、更快速的开发、更简便的管理、全新的语言以及网络服务。《ASP.NET 程序设计》也是计算机教学中开发 Web 应用程序必不可少的内容,是计算机类专业及相关专业必修的一门课程。

《ASP.NET 程序设计》是针对高职教学目标要求编写的一部有特色的教材。本教材共分 12 章:第 1 章,ASP.NET 概述;第 2 章,VB.NET 程序设计基础;第 3 章,HTML 控件和 Web 控件;第 4 章,ASP.NET 内部对象;第 5 章,数据库读取与存储;第 6 章,深入数据库编程;第 7 章,文件及文件夹管理;第 8 章,后台代码与业务逻辑;第 9 章,Web 服务;第 10 章,配置应用程序及调试程序;第 11 章,综合设计实训。

本教材的特点是:

1. 充分考虑了高职学生的特点和要求,采用案例教学法,由浅入深,从一般的例子开始讲解,逐步深入,对每一个具体实例给出详细的解释和说明,叙述过程简明透彻,通俗易懂。

2. 每一章前面精心设计了“本章重点”与“本章难点”。使教师有的放矢地讲授。

3. 综合运用比较流行的 Visual Studio.NET 程序开发环境,并结合其他网页制作工具,重在体会编程代码,实用性强。

4. 注重培养学生的实际编程能力。在教材最后给出“综合设计实训”,通过一个完整的电子商务系统的开发,介绍了 Web 应用程序所涉及的相关内容及处理方法,为学生奠定了良好的程序设计基础。



5. 本书附有各章例题源代码光盘,为教师教学和学生自学提供了必要、方便的参考资料。本书的配套课件、习题答案及模拟试卷,请到大连理工大学出版社网站下载,网址:<http://gz.dutp.cn>。

《ASP.NET 程序设计》由屈武江担任主编,姜俊颖、文继权、何保荣、赵春霞、何赤平任副主编。各章编写的具体分工如下:第1、2、3章由姜俊颖编写;第5、6章由文继权编写;第8、9章由何保荣编写;第4、7、10、11章由屈武江编写,霍艳飞参与第4、11章的编写;赵春霞、何赤平、孙如欣参与了部分章节的编写工作。

尽管我们在本教材的特色建设方面做了许多努力,但由于作者的水平有限,加之时间仓促,书中内容难免有疏漏之处,恳请各相关教学单位和读者在使用本教材的过程中给予关注,并将意见及时反馈给我们,以便在教材修订时加以改进。

所有意见和建议请发往:[gzjckfb@163.com](mailto:gzjckfb@163.com)

联系电话:0411-84707492

编者

2007年9月

# 目 录

第 1 章 ASP.NET 概述 .....	1	2.1 VB.NET 简介 .....	18
1.1 网络程序设计概述 .....	1	2.2 VB.NET 的数据类型 .....	19
1.1.1 静态网页与动态网页 .....	1	2.3 VB.NET 常量 .....	21
1.1.2 网页的工作原理 .....	1	2.4 VB.NET 变量 .....	22
1.1.3 常见的网络程序设计语言 .....	2	2.5 VB.NET 数组 .....	24
1.2 .NET Framework 简介 .....	5	2.6 VB.NET 运算符 .....	25
1.3 ASP.NET 的运行环境 .....	6	2.7 使用条件语句 .....	26
1.3.1 安装 IIS 5.0 .....	7	2.7.1 If 语句 .....	26
1.3.2 .NET Framework SDK 的安装 .....	7	2.7.2 Select Case 语句 .....	28
1.4 ASP.NET 的开发工具 .....	8	2.8 使用循环语句 .....	29
1.5 新建一个简单的 ASP.NET 应用程序 .....	9	2.8.1 For...Next 循环 .....	29
1.5.1 新建 ASP.NET 项目 .....	9	2.8.2 Do...Loop 循环 .....	30
1.5.2 新建 ASP.NET 文件 .....	9	2.8.3 While...End While 循环 .....	32
1.5.3 编辑 ASP.NET 文件(Web 窗体文件)	10	2.8.4 For Each...In 循环 .....	32
1.5.4 保存 ASP.NET 文件 .....	11	2.8.5 循环嵌套 .....	33
1.5.5 浏览 ASP.NET 文件 .....	11	2.9 VB.NET 过程 .....	34
1.6 新建一个应用程序 .....	12	2.9.1 Sub 子程序 .....	34
1.6.1 什么是应用程序 .....	12	2.9.2 Function 函数 .....	35
1.6.2 新建一个文件夹 .....	12	2.10 VB.NET 的系统函数 .....	37
1.6.3 添加虚拟目录 .....	12	2.10.1 转换函数 .....	37
1.6.4 设置默认文档 .....	13	2.10.2 字符串函数 .....	38
1.6.5 建立 ASP.NET 文件 .....	13	2.10.3 日期和时间函数 .....	40
1.6.6 新建应用程序小结 .....	14	2.10.4 数学函数 .....	41
1.7 ASP.NET 的语法简介 .....	14	2.10.5 检验函数 .....	42
1.8 开发 ASP.NET 时的注意事项 .....	15	2.11 类 .....	43
本章小结 .....	16	2.11.1 基本概念 .....	43
习题 .....	16	2.11.2 创建一个简单的类 .....	43
第 2 章 VB.NET 程序设计基础 .....	18	2.11.3 继承 .....	46
		2.11.4 共享属性和共享方法 .....	48



2.11.5 构造函数 .....	50	3.5.1 使用验证控件的注意事项 .....	100
2.12 使用系统类 .....	51	3.5.2 必须字段验证控件 .....	101
2.12.1 转换数据类型 .....	51	3.5.3 比较验证控件 .....	102
2.12.2 字符串操作 .....	52	3.5.4 范围验证控件 .....	104
2.12.3 日期和时间操作 .....	53	3.5.5 正则表达式验证控件 .....	106
2.12.4 数学操作 .....	54	3.5.6 自定义验证控件 .....	108
2.12.5 数组操作 .....	55	3.5.7 验证总结控件 .....	109
2.12.6 数组变量 .....	56	3.6 自定义控件 .....	110
2.12.7 格式化数字 .....	56	3.6.1 代码分离技术 .....	111
2.12.8 格式化日期 .....	56	3.6.2 实现 Web 控件的代码分离 .....	111
2.13 注释语句 .....	57	3.6.3 编写自定义控件 .....	112
2.14 异常处理 .....	58	3.6.4 用户控件 .....	115
本章小结 .....	58	本章小结 .....	116
习题 .....	59	习题 .....	116
<b>第 3 章 HTML 控件和 Web 控件</b> .....	<b>61</b>	<b>第 4 章 ASP.NET 内部对象</b> .....	<b>118</b>
3.1 控件简介 .....	61	4.1 ASP.NET 内部对象概述 .....	118
3.1.1 什么是控件 .....	61	4.2 Page 对象 .....	119
3.1.2 Web 表单 .....	61	4.3 Request 对象 .....	120
3.2 HTML 服务器控件 .....	64	4.3.1 Request 对象简介 .....	120
3.2.1 HtmlForm(Web 表单) .....	64	4.3.2 使用 QueryString 数据集合 .....	121
3.2.2 Anchor 控件 .....	65	4.3.3 ServerVariables 数据集合 .....	122
3.2.3 Button 控件 .....	65	4.3.4 ClientCertificate 数据集合 .....	123
3.2.4 InputButton 控件 .....	67	4.3.5 Browser 数据集合 .....	123
3.2.5 InputCheckBox 控件 .....	68	4.4 Response 对象 .....	125
3.2.6 InputRadioButton 控件 .....	69	4.4.1 Response 对象简介 .....	125
3.2.7 InputImage 控件 .....	70	4.4.2 利用 Write 方法输出信息 .....	125
3.2.8 InputFile 控件 .....	71	4.4.3 使用 Redirect 方法引导客户端 至另一个 URL 位置 .....	127
3.3 Web 控件 .....	72	4.4.4 使用 End 方法终止 ASP.NET 程序 .....	128
3.3.1 用于文本输入和显示的内部控件 .....	72	4.4.5 关于 BinaryWrite 方法 .....	128
3.3.2 用于控制传送的内部控件 .....	75	4.5 HttpCookie 对象 .....	128
3.3.3 用于选择的内部控件 .....	81	4.5.1 什么是 Cookie .....	128
3.4 增强控件 .....	93	4.5.2 使用 Response 对象设置 Cookie .....	129
3.4.1 日历控件 .....	93	4.5.3 使用 Request 对象读取 Cookie .....	129
3.4.2 广告控件 .....	97	4.5.4 Cookies 综合示例 .....	130
3.5 验证控件 .....	100		

4.6 Session 对象	131
4.6.1 Session 对象简介	131
4.6.2 利用 Session 存储信息	132
4.6.3 利用 Session 存储数组信息	134
4.6.4 Timeout 属性	135
4.6.5 Abandon 方法	135
4.7 Application 对象	136
4.7.1 Application 简介	136
4.7.2 利用 Application 存储信息	136
4.7.3 利用 Application 存储数组信息	138
4.8 Server 对象	139
4.8.1 Server 对象简介	139
4.8.2 ScriptTimeout 属性	140
4.8.3 CreateObject 方法	140
4.8.4 HTMLEncode 方法	140
4.8.5 URLEncode 方法	141
4.8.6 MapPath 方法	141
4.8.7 Execute 方法	142
4.8.8 Transfer 方法	143
4.9 Mail 对象	143
4.9.1 Mail 对象简介	144
4.9.2 发送 E-mail 示例一	145
4.9.3 发送 E-mail 示例二	146
本章小结	148
习题	148
<b>第 5 章 数据库读取与存储</b>	<b>151</b>
5.1 ADO.NET 概述	151
5.2 数据库的简单存取	152
5.2.1 准备数据库	152
5.2.2 数据库的读取	153
5.2.3 数据库的插入、修改与删除	156
5.3 使用 DataGrid 控件存取数据库	159
5.3.1 DataGrid 控件简介	160
5.3.2 使用 DataGrid 控件查询记录	160
5.3.3 插入记录	163
5.3.4 更新记录	166
5.3.5 删除记录	169
5.4 综合示例	171
本章小结	177
习题	177
<b>第 6 章 深入数据库编程</b>	<b>179</b>
6.1 DataGrid 控件再探讨	179
6.1.1 分页显示数据	179
6.1.2 对数据进行排序	181
6.1.3 定制列	183
6.1.4 链接到详细页	185
6.1.5 自定义列	187
6.2 使用 DataList 控件存取数据库	192
6.2.1 DataList 控件简介	192
6.2.2 查询记录	192
6.2.3 选择记录	194
6.2.4 更新和删除记录	197
6.3 使用 Repeater 控件存取数据库	201
6.3.1 Repeater 控件简介	201
6.3.2 查询记录	202
6.3.3 选择记录	204
6.4 使用含有参数的 SQL 语句	207
6.5 对多个表进行操作	208
6.5.1 组合查询	209
6.5.2 对多个表进行更新和删除操作	209
6.6 DataSet 对象再探讨	212
6.6.1 建立 DataSet 对象	212
6.6.2 对 DataSet 对象的操作	214
6.6.3 把 DataSet 对象中的更新写回到数据库	216
6.7 综合示例	221
本章小结	226
习题	226
<b>第 7 章 文件及文件夹操作</b>	<b>228</b>
7.1 文件及文件夹管理	228
7.1.1 显示文件及文件夹的信息	228
7.1.2 文件的新建、复制、移动和删除	230

7.1.3 文件夹的新建、复制、移动和删除	231	9.5 返回 DataSet 对象示例	267
7.2 文本文件操作	232	9.5.1 建立 Web 服务	268
7.2.1 新建和写入文本文件	233	9.5.2 生成 SOAP 代理类	268
7.2.2 读取文本文件	234	9.5.3 编译并部署代理类	268
7.2.3 计数器示例	234	9.5.4 通过代理类调用 Web 服务	269
7.3 XML 文件操作	236	本章小结	270
7.3.1 读取 XML 文件	236	习题	270
7.3.2 写入 XML 文件	237	<b>第 10 章 配置应用程序及调试程序</b>	272
7.4 Excel 文件操作	237	10.1 关于配置应用程序	272
7.5 综合示例	239	10.2 配置 Global.asax 文件	273
本章小结	241	10.2.1 Global.asax 文件的基本格式	273
习题	242	10.2.2 Global.asax 文件示例	274
<b>第 8 章 后台代码与业务逻辑</b>	243	10.3 配置 web.Config 文件	276
8.1 使用后台代码(Code-behind)	243	10.3.1 web.Config 文件的基本格式	276
8.1.1 在普通页面中使用 Code-behind	243	10.3.2 读取自定义属性示例	277
8.1.2 在用户控件中使用 Code-behind	246	10.4 安全考虑	278
8.2 使用业务逻辑	249	10.4.1 基于 Windows 的认证	279
8.2.1 业务逻辑简介	249	10.4.2 基于表单的认证	280
8.2.2 不包含界面的业务逻辑	249	10.4.3 关于授权	281
8.2.3 包含界面的业务逻辑	253	10.4.4 网络安全综合示例	281
8.3 使用 COM 组件	258	10.5 调试程序	286
8.3.1 使用 Server 对象的 CreateObject	258	10.5.1 使用 Response.Write 方法	286
8.3.2 将 COM 组件转换成 .NET 组件	259	10.5.2 使用 Debug	286
本章小结	260	10.5.3 使用 trace(跟踪)	286
习题	261	10.6 错误处理	287
<b>第 9 章 Web 服务</b>	262	10.6.1 语法错误	287
9.1 Web 服务简介	262	10.6.2 执行时错误	287
9.1.1 什么是 Web 服务	262	10.6.3 逻辑错误	287
9.1.2 Web 服务的工作原理	263	本章小结	287
9.1.3 Web 服务能返回什么类型的数据	263	习题	288
9.2 建立一个简单的 Web 服务	263	<b>第 11 章 综合设计实训</b>	289
9.3 简单调用 Web 服务	264	11.1 工程项目背景及意义	289
9.4 通过代理类调用 Web 服务	265	11.2 数据模型和数据库设计	293
		11.2.1 数据模型与数据库设计	293
		11.2.2 存储过程设计	295
		11.3 电子商务系统的数据访问层	298

11.3.1	商品目录和商品信息数据服务	298	11.6	电子商务的购物流程	328
11.3.2	客户管理数据服务	302	11.6.1	商品浏览设计	328
11.3.3	购物车管理数据服务	306	11.6.2	商品购物流程	331
11.3.4	订单处理数据服务	314	11.6.3	订单的处理	335
11.4	电子商务网站设计	318	11.7	电子商务的 Web 服务	339
11.4.1	创建自定义 Web 控件	318	11.7.1	订单的 Web 订阅服务	339
11.4.2	创建网站页面的分析	322	11.7.2	订单的 Web 查询服务	343
11.4.3	电子商务网站首页	323		本章小结	344
11.5	电子商务的客户管理	324	<b>附 录</b>	<b>常见问题答疑</b>	345
11.5.1	客户注册页面	324	<b>参 考 文 献</b>		347
11.5.2	客户登录页面	326			

## 第 1 章

### ● 本章重点

- ASP.NET 运行环境的组建
- 新建一个 ASP.NET 应用程序

### ● 本章难点

- 建立 ASP.NET 应用程序

## 1.1 网络程序设计概述

### 1.1.1 静态网页与动态网页

目前的 WWW 网页根据其生成方式,大致可以分为静态网页和动态网页两种。所谓静态网页,就是网页里面没有供服务器端执行的程序代码,不会被服务器执行。这种网页通常在服务器端以扩展名 .htm 或是 .html 存储,表示里面的内容是以 HTML 语言所撰写。HTML 语言是由许多叫做标注(Tag)的元素所组成。这种语言指示了文字、图形等元素在浏览器上面的配置、样式以及这些元素实际上是存放于因特网上的哪个地方(地址),或是点选了某段文字或图形后,应该要链接到哪个网址。在浏览这种扩展名为 .htm 的网页的时候,网站服务器不会执行任何程序,会把网页直接传给客户端的浏览器进行解读工作。所以除非网站设计师更新过网页文件的内容,否则网页不会因为执行程序而出现不同的内容。

所谓动态网页,就是网页内含有程序代码,并会被服务器执行。这种网页通常在服务器端以扩展名 .asp、.aspx、.jsp、.php 等存储,表示里面的内容是动态网页,有需要执行的程序。使用者要浏览这种网页时必须由服务器先执行程序后,再将执行完的结果下载给客户端的浏览器。这种动态网页会在服务器执行一些程序,由于执行程序时的条件不同,执行的结果也可能会有所不同,所以称为动态网页。

一般来说,静态网页的制作比较简单,利用 FrontPage、Dreamweaver 和 Windows 自带的记事本就可以方便地制作生成扩展名为 .htm 或 .html 的静态网页文件。而动态网页的制作就比较复杂了,需要用到 ASP、PHP、JSP 或 ASP.NET 等专门的动态网页设计语言。

### 1.1.2 网页的工作原理

要了解动态网页的工作原理,首先要了解什么是服务器端和客户端,并且要注意比较静态网页和动态网页工作原理的区别。

### (1) 什么是服务器端、客户端

一般来说,凡是提供服务的一方称为服务器端,而接受服务的一方称为客户端。比如,当用户在浏览某一网站的主页时,网站主页所在的服务器就称为服务器端,而用户所使用的计算机就称为客户端。如图 1-1 所示。

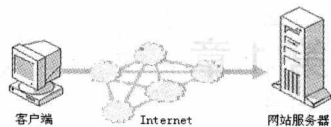


图 1-1 服务器端和客户端示例图

但是服务器端和客户端只是一种相对的关系,而不是绝对的关系。例如原来提供服务的服务器端要接受别的服务器端提供的服务,它就变为了客户端;或者原来接受服务的客户端要为别的客户端提供服务,它就变为了服务器端。

具体到使用的计算机:如果要浏览某一网站的主页,它就是客户端;如果给自己的计算机安装了 WWW 服务器软件,就可以把自己的计算机变为服务器,别人也可以通过网络来访问您的计算机。此时,您使用的计算机就是服务器端;在实际的学习中,初学者往往把自己的计算机既当作服务器端,又当作客户端,先在自己的计算机上调试好程序并保存,然后通过 IE 浏览器访问调试好的页面,浏览程序显示的结果。

### (2) 静态网页的工作原理

当在浏览器的地址栏里输入了一个静态网页的网址并回车后,就向服务器端提出一个浏览网页的请求。服务器端接到请求后,就会找到要浏览的静态网页文件,然后发送给客户端,如图 1-2 所示。

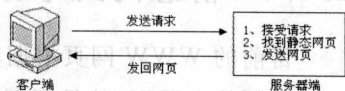


图 1-2 静态网页工作原理

### (3) 动态网页的工作原理

动态网页的工作原理与静态网页有很大的不同,当在浏览器里输入一个动态网页网址后,就向网站服务器提出了一个浏览网页的请求。网站服务器接到请求后,首先会找到要浏览的动态网页文件,网站服务器依据动态网页的执行条件,动态产生网页(静态网页),然后将产生的网页(静态网页)下载至客户端显示,如图 1-3 所示。

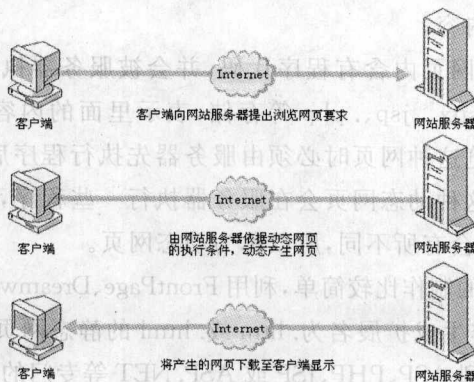


图 1-3 动态网页工作原理

## 1.1.3 常见的网络程序设计语言

目前,常见的动态网络程序设计语言有 ASP、PHP、JSP 和 ASP.NET 等。下面依次

进行介绍。

### (1) ASP

ASP 是 Active Server Page 的缩写,意为“动态网页”,ASP 的网页文件的格式是 .asp,常用于各种动态网站中。它是一套微软开发的服务器端脚本环境,ASP 内含于 IIS 3.0 和 4.0 之中,通过 ASP 可以结合 HTML 网页、ASP 指令和 ActiveX 组件建立动态、交互且高效的 Web 服务器应用程序。有了 ASP 你就不必担心客户的浏览器是否能运行你所编写的代码,因为所有的程序都将在服务器端执行,包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后,服务器仅将执行的结果返回给客户端浏览器,这样也就减轻了客户端浏览器的负担,大大提高了交互的速度。

ASP 的最大优点就是简单易学,也正因为这个优点,又有微软的强大支持,所以目前 ASP 使用非常广泛,很多大型的站点都是用 ASP 开发的。

ASP 的主要缺点是不能跨平台,一般只能在 Windows 操作系统上运行。

### (2) JSP

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导,许多公司参与一起建立的一种动态网页技术标准。该技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 技术的设计,使得构造基于 Web 的应用程序更加容易和快捷,而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。JSP 规范是 Web 服务器、应用服务器、交易系统以及开发工具供应商间广泛合作的结果。在传统的网页 HTML 文件(.htm,.html)中加入 Java 程序片段(Servlet)和 JSP 标记(tag),就构成了 JSP 网页(.jsp)。Web 服务器在遇到访问 JSP 网页的请求时,首先执行其中的程序片段,然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送 Email 等,这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行,网络上传送给客户端的仅是得到的结果,对客户端浏览器的要求最低,可以实现无 Plugin、无 ActiveX、无 Java Applet,甚至无 Frame。

### (3) PHP

PHP 是一种用于创建动态 Web 页面的服务器端脚本语言。如同 ASP 和 JSP,用户可以混合使用 PHP 和 HTML 编写 Web 页面,当访问者浏览到该页面时,服务器端会首先对页面中的 PHP 命令进行处理,然后把处理后的结果连同 HTML 内容一起传送到访问端的浏览器。但是与 ASP 或 JSP 不同,PHP 是一种源代码开放程序,拥有很好的跨平台兼容性。用户可以在 Windows NT 系统以及许多版本的 UNIX 系统上运行 PHP,而且可以将 PHP 作为 Apache 服务器的内置模块或 CGI 程序运行。

除了能够精确地控制 Web 页面的显示内容之外,用户还可以通过使用 PHP 发送 HTTP 报头。用户可以通过 PHP 设置 Cookies,管理用户身份识别,并对用户浏览页面进行重定向。PHP 具有非常强大的数据库支持功能,能够访问几乎目前所有较为流行的数据库系统。此外,PHP 可以与多个外库集成,为用户提供更多的实用功能,如生成 PDF 文件等。

用户可以直接在 Web 页面中输入 PHP 命令代码,因而不需要任何特殊的开发环

境。在 Web 页面中,所有 PHP 代码都被放置在“< ? php”和“ ? >”中。此外,用户还可以选择使用诸如<script language="php"></script>等形式。PHP 引擎会自动识别并处理页面中所有位于 PHP 定界符之间的代码。

PHP 脚本语言的语法结构与 C 语言和 Perl 语言的语法风格非常相似。用户在使用变量前不需要对变量进行声明。使用 PHP 创建数组的过程也非常简单。PHP 还具有基本的面向对象组件功能,可以极大地方便用户有效组织和封装自己编写的代码。

#### (4) ASP.NET

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 具有如下几个重要的优点:

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。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运行之前便显著提高了性能。

②强大的工具支持。ASP.NET 可以在 Visual Studio.NET 集成开发环境中开发,支持所见即所得、拖放控件和自动部署等功能,可以使开发效率大大提高。

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 也与语言无关,所以可以选择最适合应用程序的语言,也可以跨多种语言开发应用程序。另外,公共语言运行库的交互性保证在迁移到 ASP.NET 时保留基于 COM 开发中的现有投资。

④简易性。ASP.NET 使执行常见任务变得容易,从简单的窗体提交和客户端身份验证到部署和配置站点。例如,ASP.NET 页面框架使您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,可以在类似 Visual Basic 的简单窗体处理模型中处理事件。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。

⑤管理性。ASP.NET 采用基于文本的分层配置系统,简化了将设置应用于服务器环境和 Web 应用程序。由于配置信息是以纯文本形式存储的,因此可以在没有本地管理工具帮助的情况下应用新设置。这种“零本地管理”哲学也扩展到了 ASP.NET 框架应用程序的部署。只需将必要的文件复制到服务器,即可将 ASP.NET 框架应用程序部署到服务器。即使是在部署或替换运行的编译代码时,也不需要重新启动服务器。

⑥缩放性和可用性。ASP.NET 在设计时考虑了可缩放性,增加了专门用于在聚集环境和多处理器环境中提高性能的功能。另外,进程受到 ASP.NET 运行库的密切监视和管理,以便当进程行为不正常(泄漏、死锁)时,可就地创建新进程,以帮助保持应用程序始终可用于处理请求。

⑦定义性和扩展性。ASP.NET 随附了一个设计周到的结构,它使开发人员可以在适当的级别“插入”代码。实际上,可以用自己编写的自定义组件扩展或替换 ASP.NET 运行库的任何子组件。

⑧安全性。借助内置的 Windows 身份验证和基于每个应用程序的配置,可以保证



应用程序的安全性。

综上所述,ASP.NET 是现今功能最为强大的开发 Web 应用程序的模型。

## 1.2 .NET Framework 简介

Web 应用程序(以因特网为基础的应用程序)的优点在于可以让企业间的商业数据及交易等行为,通过因特网来彼此交换信息。这样不但可以节省数据交换的时间,而且可以简化操作流程。但是在新一代的 .NET 开发平台还没有出现之前,要让因特网应用程序达到上述的功能是一项浩大的工程。牵涉到的技术及程序开发语言很多,包括了 HTML、ASP、VBScript、JavaScript、C++、ADO、SQL、COM、MTS 等。这样的环境对于开发人员来说,想要快速开发一个功能强大且稳定可靠的 Web 应用程序,不是一项轻松的工作。之前的 Microsoft Visual Studio TM(微软的解决方案开发平台)及 Windows 上的一些架构及服务,已经帮助程序设计师由单机平台的程序开发转为主从式(Client/Server)的架构来开发应用程序。但是对于现在的环境而言,因为企业的客户或供货商所使用的系统或软件可能不尽相同,开发人员所要面临的挑战是要如何整合所有的资源,所以必须把焦点转到如何发展分布式的因特网应用程序架构,整合这些存在于不同平台或不同软件的资源。目前全世界有超过六百万的专业程序设计师,而 70%以上的人使用微软的 Visual Studio 开发平台;现今最受欢迎的架构则为主从式的架构,而发展最快速的架构则是以 Web 为基础的架构。在这种架构下程序逻辑及数据的处理都在服务器端,使用者通过网络用浏览器来存取服务器端的数据。这种架构的开发工具,微软早在前两版的 Visual Studio 中通过 Active Server Pages(ASP)以及 IIS 3.0 就已经提供了。为了适应 Web 架构的快速发展及广大程序开发人员的需求,微软也举办了不下百场的研讨会和开发人员沟通、交换意见,就是为了提供给开发人员更好的解决方案开发平台。而这个新一代开发平台目前已经推出了,即为 Visual Studio .NET。为了更容易开发以因特网为基础的应用程序,这套开发平台做了许多和以往不同的改革,这个改革就是 .NET Framework。

.NET Framework(.NET 也叫做 Dot Net)的目的就是要让建立 Web Services 以及因特网应用程序的工作变得简单,.NET Framework 包括了三大部分:

第一部分是公共语言运行库(Common Language Runtime,CLR);

第二部分是基础类库;

第三部分是 VB.NET、C# 等语言运行库。

(1) 公共语言运行库(Common Language Runtime,CLR)

公共语言运行库是 .NET 框架的运行环境,负责运行和维护用户所编写的任何程序代码,包括内存管理、线程执行、代码执行、代码安全验证、编译及其他系统服务等。简单地说,不管是 ASP.NET 还是 Windows 应用程序,都由它负责运行。具体运行过程如下。

首先程序员创建源代码,当网页被第一次访问时,就会编译生成一种叫做微软中间