



高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

ASP.NET 应用 程序开发实用教程

宋楚平 主编

免费提供

教学相关资料



人民邮电出版社
POSTS & TELECOM PRESS

高职高专计算机技能型紧缺人才培养规划教材
计算机软件技术专业

ASP.NET 应用程序开发实用教程

宋楚平 主编 ←

人 民 邮 电 出 版 社
北 京

图书在版编目(CIP)数据

ASP.NET 应用程序开发实用教程 / 宋楚平主编. —北京:
人民邮电出版社, 2008.9

高职高专计算机技能型紧缺人才培养规划教材. 计算机软
件技术专业

ISBN 978-7-115-18071-1

I. A… II. 宋… 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 (2008) 第 063460 号

内 容 提 要

ASP.NET 是 Microsoft 公司推出的新一代 Web 开发架构技术。软件开发人员运用 ASP.NET 技术能快速完成企业级 Web 应用程序和 Web 服务的开发任务。

Microsoft 公司力推的 C#语言是 .NET 技术的首选语言。本书从实用的角度出发, 结合软件企业对 .NET Web 应用人才的技能要求, 详细讲解 ASP.NET 应用程序开发技术。全书共分为 9 章, 内容包括 ASP.NET 简介、Web Form 和服务控件、ASP.NET 常用内置对象、ADO.NET 应用技术、数据展现和报表技术、XML 数据管理、Web 服务、ASP.NET 的配置和安全以及项目实战。各章节的编排以软件开发周期为主线来展开, 在其中由浅入深地渗透专业技能和项目经验, 以便使读者快速掌握和运用 ASP.NET 的编程技术和实用技能。

本书可作为高职高专院校软件技术专业和计算机相关专业的教材, 对于在 .NET 框架下开发 Web 应用程序的软件设计人员和编程爱好者也具有较高的参考价值。

高职高专计算机技能型紧缺人才培养规划教材

计算机软件技术专业

ASP.NET 应用程序开发实用教程

-
- ◆ 主 编 宋楚平
责任编辑 刘雁斌
 - ◆ 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>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.25
字数: 365 千字
印数: 1—3 000 册
 - 2008 年 9 月第 1 版
2008 年 9 月北京第 1 次印刷

ISBN 978-7-115-18071-1/TP

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

高职高专计算机技能型紧缺人才培养

规划教材编委会

主 任 武马群

副 主 任 王泰峰 徐民鹰 王晓丹

编 委 (以姓氏笔画为序)

马 伟	安志远	向 伟	刘 兵	吴卫祖	吴宏雷
余明辉	张晓蕾	张基宏	贺 平	柳 青	赵英杰
施晓秋	姜 锐	耿 壮	郭 勇	曹 炜	蒋方纯
潘春燕					

丛书出版前言

目前,人才问题是制约我国软件产业发展的关键。为加大软件人才培养力度和提高软件人才培养质量,教育部继在2003年确定北京信息职业技术学院等35所高职院校试办示范性软件职业技术学院后,又同时根据《教育部等六部门关于实施职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》(教职成[2003]5号)的要求,组织制定了《两年制高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。示范性软件职业技术学院与计算机应用与软件技术专业领域技能型紧缺人才培养工作,均要求在较短的时间内培养出符合企业需要、具有核心技能的软件技术人才,因此,对目前高等职业教育的办学模式和人才培养方案等做较大的改进和全新的探索已经成为学校的当务之急。

据此,我们认为做一套符合上述一系列要求的切合学校实际的教学方案尤为重要。遵照教育部提出的以就业为导向,高等职业教育从专业本位向职业岗位和就业为本转变的指导思想,根据目前高等职业教育院校日益重视学生将来的就业岗位,注重培养毕业生的职业能力的现状,我们联合北京信息职业技术学院等几十所高职院校和普拉内特计算机技术(北京)有限公司、福建星网锐捷网络有限公司、北京索浪计算机有限公司等软件企业共同组建了计算机应用与软件技术专业领域技能型紧缺人才培养教学方案研究小组(以下简称研究小组)。研究小组对承担计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的79所院校的专业设置情况做了细致的调研,并调查了几十所高职院校计算机相关专业的学生就业情况以及目前软件企业的人才市场需求状况,确定首批开发目前在高职院校开设比较普遍的计算机软件技术、计算机网络技术、计算机多媒体技术和计算机应用技术4个专业方向的教学方案。

同时,为贯彻教育部提出的要与软件企业合作开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的精神,使高等职业教育培养出的软件技术人才符合企业的需求,研究小组与许多软件企业的专家们进行了反复研讨,了解到目前高职院校的毕业生的实际动手能力和综合应用知识方面较弱,他们和企业需求的软件人才有着较大的差距,到企业后不能很快独当一面,企业需要投入一定的成本和时间进行项目培训。针对这种情况,研究小组在教学方案中增加了“综合项目实训”模块,以求强化学生的实际动手能力和综合应用前期所学知识的能力,探索将企业的岗前培训内容前移到学校的教学中的实验之路,以此增强毕业生的就业竞争力。

在上述工作的基础上,研究小组于2004年多次组织召开了包括企业专家、教育专家、学校任课教师在内的各种研讨会和方案论证会,对各个专业按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”一步步进行了认真的分析和研讨:

- 列出各专业的岗位群及核心技能。针对教育部提出的以就业为导向,根据目前高职高专院校日益关心学生将来的就业岗位的现状,在前期大量调研的基础上,首先提炼各个专业的岗位群。如对某专业的岗位群进行研究时,首先罗列此专业的各个岗位,以便能正确了解

每个岗位的职业能力，再根据职业能力进行有意义的合并，形成各个专业的岗位群，再对每个岗位群总结和归纳出其核心技能。

- 根据岗位群及核心技能做出教学方案。在岗位群及核心技能明确的前提下，列出此岗位应该掌握的知识点，再依据这些知识点推出应该学习的课程、学时数、课程之间的联系、开课顺序并进行必要的整合，最终形成一套科学完整的教学方案。

为配合学校对技能型紧缺人才的培养工作，在研究小组开发上述 4 个专业的教学方案的基础上，我们组织编写了这套包含计算机软件技术、计算机网络技术、计算机多媒体技术及计算机应用技术 4 个专业的教材。本套教材具有以下特点：

- 注重专业整体策划的内涵。对各专业系列教材按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”的思路组织开发教材。

- 按照“理论够用为度”的原则，对各个专业的基础课进行了按需重新整合。

- 各专业教材突出了实训的比例，注重案例教学。每本教材都配备了实验、实训的内容，部分专业的教材配备了综合项目实训，使学生通过模拟具体的软件开发项目了解软件企业的运行环境，体验软件的规范化、标准化、专业化和规模化的开发流程。

为了方便教学，我们免费为选用本套教材的老师提供部分专业的整体教学方案及教学相关资料。

- 所有教材的电子教案。
- 部分教材的习题答案。
- 部分教材中实例制作过程中用到的素材。
- 部分教材中实例的制作效果以及一些源程序代码。

本套教材以各个专业的岗位群为出发点，注重专业整体策划，试图通过对系列教材的整体构架，探索一条培养技能型紧缺人才的有效途径。

经过近两年的艰苦探索和工作，本套教材终于正式出版了，我们衷心希望，各位关心高等职业教育的读者能够对本套教材的不当之处给予批评指正，提出修改意见，也热切盼望从事高等职业教育的教师以及软件企业的技术专家和我们联系，共同探讨计算机应用与软件技术专业的教学方案和教材编写等相关问题。来信请发至 panchunyan@ptpress.com.cn。

前 言

ASP.NET 是微软.NET 的一门核心技术,也是一种软件编程框架,可以在部门间或企业级构建功能强大的 Web 应用程序。它为软件编程人员提供了一种快捷和全新的编程模式。利用 ASP.NET 技术来构建 Web 应用和安装部署项目变得更加容易和快捷,并能与其他微软技术共同协作,来满足开发人员各方面的要求,目前 ASP.NET 已经成为众多软件企业首选的软件开发技术。

本书共分 9 章,从 ASP.NET 基础开始,系统、全面地介绍 ASP.NET 开发应用程序的方法和技术,各章内容安排如下。

第 1 章主要介绍开发 Web 应用程序必备的一些 Web 基础知识、ASP.NET 的编程框架和工作原理以及如何搭建一个运行 ASP.NET 的平台。通过这些基础知识的介绍让读者对 ASP.NET 有一个整体的了解,为后续的学习打下基础。

第 2 章主要介绍 ASP.NET 支持的 3 种控件。通过学习本章,读者能学会合理使用服务器控件,以提高编程效率,丰富程序的功能。

第 3 章主要对 ASP.NET 常用的 6 个对象进行介绍,通过学习本章,读者能学会利用这些对象解决一些常见的应用问题,开发出功能强大的应用程序。

第 4 章主要介绍 ADO.NET 的基础知识、ADO.NET 数据结构、如何利用 ADO.NET 连接到不同的数据库、如何使用 ADO.NET 为 Web 应用程序添加数据访问功能,以及怎样在 Web 应用程序中通过 Microsoft ADO.NET 访问存储过程来完成数据访问任务等。

第 5 章主要介绍在 ASP.NET 中展示和操作数据的 3 个控件,以及如何利用自定义报表和水晶报表的方式来拓展数据展现功能。通过学习本章,读者能掌握 Web 应用程序中最常用的一门数据显示和报表技术。

第 6 章主要介绍 XML 基础知识、在 ASP.NET 中处理 XML 数据的常用方法以及 DataSet 和 XML 的交互技术等。

第 7 章主要介绍 Web 服务的基本知识、如何在 VS.NET 平台下创建 Web 服务及如何在 ASP.NET 应用程序中调用 Web 服务,并列举出常用的一些 Web 服务实例。通过学习本章,读者能学会利用 Web 服务技术来解决一些实际问题。

第 8 章主要介绍配置文件的基本结构、配置文件中常用的一些配置设置以及如何利用配置文件来建立应用程序的安全运行体系,并从实用的角度讨论 Web 应用程序的安全保证体系。

第 9 章主要以一个项目为目标,从软件开发过程和实用技术的角度,让大家对开发 ASP.NET 应用程序有一个实际的了解,并将主要的目标集中在软件开发过程和开发方法上,逐步培养开发实际项目的能力和技能。

本书具有如下特点。

1. 内容新颖

本书根据软件人才的岗位能力要求和深入软件企业调研的结果确定编写内容,在教材中

融入了存储过程、XML 数据管理、Web 服务等新的内容，以符合.NET 应用人才的知识 and 能力要求，适应软件技术的发展趋势。

2. 结构清晰

本书是作者多年教学经验和科研工作的总结，在教材编排和写法上，既注重基本知识的讲解清晰和严谨，又采用由浅入深、前后衔接、逐步推进的方式来组织内容，犹如规律地完成一件软件作品。

3. 突出实用

为在掌握必须够用的 ASP.NET 知识的同时，提高实际应用技能和解决问题的综合能力。书中融入了笔者在多年教学和大量工程实践中总结出来的经验，对数据展示和报表技术、ADO.NET 的典型应用、应用程序的安全配置和打包进行了详细的描述，而这些内容都是实践开发中必须完成的步骤，在一般的书中很少见到。书中所有的实例都源于实际的应用项目，具有很强的实用性。很多实例可直接（或稍加修改）应用到读者的实际开发项目中去。

本书由宋楚平担任主编，周建辉、张海飞担任副主编。第 1、6 章由张海飞编写；第 2、3、7 章由周建辉编写；第 4、5、8、9 章由宋楚平编写。全书由宋楚平统稿。南通大学的沈学华对本书进行了全面的审阅，并提出了许多宝贵的意见。

由于时间仓促，加之编者水平有限，书中不足之处在所难免，恳请读者批评指正。

编 者
2008 年 5 月

目 录

第 1 章 ASP.NET 简介 1

1.1 Web 基础知识 1
1.1.1 HTTP 2
1.1.2 Web 服务器 2
1.1.3 Web 浏览器 3
1.1.4 URL 3
1.1.5 HTML 3
1.2 ASP.NET 的工作原理 4
1.2.1 .NET 4
1.2.2 .NET Framework 4
1.2.3 公共中间语言 5
1.2.4 ASP.NET 的优势 6
1.3 ASP.NET 的运行环境 6
1.3.1 安装 IIS 7
1.3.2 安装 .NET Framework SDK 9
1.3.3 目录管理 11
本章小结 14
思考练习 15

第 2 章 Web Form 和服务器

控件 16
2.1 Web Form 的编程模型 16
2.1.1 ASP.NET 页面的处理 过程 16
2.1.2 页面的属性、方法和 事件 17
2.1.3 页面指示符 20
2.2 Web 服务器端控件 24
2.2.1 Web 服务器控件的 公共属性 24
2.2.2 文本控件 25
2.2.3 按钮控件 27

2.2.4 选择控件 30
2.2.5 容器控件 34
2.2.6 其他控件 36
2.3 HTML 服务器端控件 40
2.3.1 HTML 服务器端控件的 属性 40
2.3.2 HtmlInputFile 控件 41
2.4 验证控件 43
2.4.1 RequiredFieldValidator 验证控件 43
2.4.2 CompareValidator 验证控件 46
2.4.3 RangeValidator 验证控件 48
2.4.4 RegularExpressionValidator 验证控件 51
2.4.5 CustomValidator 验证控件 54
2.4.6 ValidationSummary 控件 56
本章小结 58
思考练习 58

第 3 章 ASP.NET 常用内置对象 60

3.1 Response 对象 60
3.1.1 输出数据 61
3.1.2 输出文件 61
3.1.3 网页转向 63
3.1.4 停止输出数据 64
3.2 Request 对象 65
3.2.1 获取客户端机器和 浏览器的信息 66
3.2.2 获取客户端表单数据 66
3.3 Server 对象 69
3.3.1 获取物理路径 70
3.3.2 执行 URL 70
3.3.3 输出 HTML 代码 71

3.3.4 对 URL 进行编码	72
3.4 Application 对象	73
3.4.1 聊天室应用	73
3.4.2 网页计数器应用	75
3.5 Session 对象	76
3.5.1 利用 Session 验证用户 身份	77
3.5.2 Session 的有效时间	79
3.6 Cookie 对象	80
3.6.1 将 Cookie 写到浏览器	81
3.6.2 从浏览器读取 Cookie	82
本章小结	82
思考练习	83

第 4 章 ADO.NET 应用

4.1 ADO.NET 数据访问概述	84
4.1.1 ADO.NET 的优势	85
4.1.2 ADO.NET 的结构	85
4.1.3 ADO.NET 的基本组件	86
4.2 数据的增删改查	89
4.2.1 建立数据库连接	89
4.2.2 检索数据	89
4.2.3 增加、删除和更改数据	91
4.3 数据集	96
4.3.1 创建数据集	96
4.3.2 数据表、数据列和 数据行	97
4.3.3 DataView	99
4.4 使用存储过程	102
4.4.1 返回记录的存储过程	103
4.4.2 返回值的存储过程	104
4.4.3 行为存储过程	106
4.5 用 ADO.NET 进行数据绑定	108
4.5.1 使用数据表达式实现数据 绑定	109
4.5.2 使用 DataBinder 类实现数据 绑定	112
4.6 使用事务处理	115
4.6.1 在数据库级使用事务	

处理	115
4.6.2 在 ADO.NET 级使用事务 处理	116
本章小结	117
思考练习	117

第 5 章 数据展现和报表技术

5.1 使用 Repeater 控件展现数据	119
5.2 使用 DataList 控件展现和 操作数据	122
5.2.1 使用 DataList 控件展现 数据	122
5.2.2 使用 DataList 操作数据	125
5.3 使用 DataGrid 展现和操作 数据	131
5.3.1 使用 DataGrid 展现数据	132
5.3.2 DataGrid 的数据排序和 分页	137
5.3.3 使用 DataGrid 操作数据	139
5.4 自定义报表	144
5.5 水晶报表技术	147
5.5.1 使用拉模型实现数据 展现	148
5.5.2 使用推模型实现数据 展现	153
本章小结	156
思考练习	156

第 6 章 XML 数据管理

6.1 XML 基础	157
6.2 XML.NET 简介	158
6.3 读取 XML 文件	161
6.3.1 使用 XML 控件读取 XML 文件	161
6.3.2 使用 DOM 技术读取 XML 文件	162
6.3.3 按文本方式读取 XML 文件	163
6.3.4 利用 DataSet 对象读取 XML	

文件.....	164	8.1.1 配置文件的特点.....	187
6.4 创建 XML 文件.....	165	8.1.2 配置文件的结构.....	188
6.4.1 利用 XmlTextWriter 创建 XML 文件.....	165	8.2 常用配置设置.....	190
6.4.2 利用 DataSet 创建 XML 文件.....	166	8.2.1 httpRuntime 通用配置.....	190
6.4.3 利用 DOM 技术创建 XML 文件.....	167	8.2.2 pages 配置.....	190
6.5 XML 与 ADO.NET 交互.....	168	8.2.3 appSettings 配置.....	191
本章小结.....	169	8.2.4 customErrors 配置.....	192
思考练习.....	169	8.2.5 globalization 配置.....	193
第 7 章 Web 服务.....	170	8.3 安全性配置.....	194
7.1 Web 服务概要.....	170	8.3.1 基于 Windows 账号验证.....	194
7.1.1 XML、SOAP 与 Web Service.....	170	8.3.2 基于表单验证.....	196
7.1.2 Web 服务的体系结构.....	171	8.3.3 基于 Passport 验证.....	201
7.2 创建 Web 服务.....	172	本章小结.....	201
7.3 使用 Web 服务.....	176	思考练习.....	201
7.3.1 通过 Web 引用方式使用 Web 服务.....	177	第 9 章 项目实战——论文评审系统.....	204
7.3.2 通过代理类使用 Web 服务.....	178	9.1 功能需求分析.....	204
7.4 使用 Web 服务的数据服务.....	181	9.2 数据库设计.....	206
7.4.1 使用 Web Service 完成数据查询.....	182	9.3 接口程序的编写.....	207
7.4.2 使用 Web Service 向数据库添加数据.....	183	9.4 主要技术的实现.....	213
7.4.3 使用 Web Service 传送二进制文件.....	184	9.4.1 菜单窗口的实现.....	213
本章小结.....	185	9.4.2 动态表格的生成.....	215
思考练习.....	186	9.5 提高应用程序性能的几种方法.....	218
第 8 章 ASP.NET 的配置和安全.....	187	9.6 应用程序的打包和部署.....	220
8.1 配置文件的基本结构.....	187	9.6.1 应用程序的打包.....	220
		9.6.2 应用程序的部署.....	228
		本章小结.....	230
		思考练习.....	231
		参考文献.....	232

从用户界面、Web 迅速成长为全球范围内信息速率。1994 年, W3 联盟在 Tim Berners-Lee 的领导下成立, 该组织通过制定技术标准来规范 Web 的技术, 并促进 Web 产品之间的互操作性。

第 1 章

ASP.NET 简介

本章要点

目前, Internet 已经成为人们生活、学习和工作中不可缺少的一部分。Web 浏览器向人们提供了很多单位和个人都开始建立自己的网站。人们都希望自己制作的网站信息量丰富, 功能尽可能强大。但是如果只使用 HTML, 就只能保证网页的美观, 却不能引入更多更强大的功能。为了实现复杂的业务流程、提高数据的集成和提供强大的交互功能, 一些 Web 应用开发技术, 如 ASP、JSP 等就应运而生了。

随着时间的推移, 人们又发现像 ASP 这样的技术虽然为网站的设计带来了方便, 但也使得网站的各种代码难以维护, 程序的模块化和可重用性都很低。另外, 由于 VBScript 脚本语言的局限性, 很多功能都不能轻松实现, 即使实现了, 也需要编写大量的代码, 为了解决这些问题, Microsoft 公司推出了 ASP.NET。

本章主要介绍开发 Web 应用程序必备的一些 Web 基础知识、ASP.NET 的工作原理和如何搭建一个运行 ASP.NET 的平台。

通过本章的学习, 读者应该能够:

- 了解 Web 的基础知识
- 了解 ASP.NET 的工作原理
- 掌握如何建立 ASP.NET 运行环境

1.1 Web 基础知识

Web (World Wide Web, 万维网) 技术是电子商务的核心技术。Web 的思想可追溯到 Tim Berners-Lee 于 1989 年 3 月在 CERN (Centre European pour la Recherche Nucleaire, 或称 European Laboratory for Particle Physics, 欧洲粒子物理实验室) 编写的一个关于信息管理的项目建议书 (Information Management: A Proposal)。该建议书提出了分布式超文本系统的设想, 旨在将 CERN 已有的几个信息服务器一体化, 并提供一个简单的用户界面来存取各种形式的信息。1990 年 Web 浏览器和 Web 服务器使用面向对象技术相继在 CERN 实现。Berners-Lee 和他的合作伙伴成功引入了构成 Web 体系结构的基本元素: Web 服务器、Web 浏览器、浏览器与服务器之间的 HTTP (Hypertext Transfer Protocol, 超文本传输协议)、写 Web 文档的 HTML (Hypertext Markup Language, 超文本标记语言) 以及用来标识 Web 上资源的 URL (Uniform Resource Locator, 统一资源定位器)。1993 年, 美国伊利诺斯大学国家超级计算应用中心 (National Center for Supercomputing Applications, NCSA) 的 Marc Andreessen 及其合作者发布了称为 Mosaic 的浏览器, 这是第一个较健壮的易用的浏览器, 它具有友好的

图形用户界面。从此, Web 迅速成长为全球范围内的信息宝库。1994 年, W3 联盟在 Tim Berners-Lee 的领导下成立, 该组织通过制定技术规范与提供参考软件来发展 Web 的技术标准并促进 Web 产品之间的互操作性。

1.1.1 HTTP

HTTP (Hypertext Transfer Protocol, 超文本传输协议) 是 Web 上最常用也是最重要的协议, 也是 Web 服务器和 Web 客户 (如浏览器) 之间传输 Web 页面的基础, 浏览器默认使用这个协议。当用户在浏览器的地址栏中输入 `www.sina.com` 时, 浏览器会自动使用 HTTP 来搜索 `http://www.sina.com` 网站的首页。

从浏览器向 Web 服务器发出的搜索某个 Web 网页的请求是 HTTP 请求。当 Web 服务器收到这个请求之后, 就会按照请求的要求, 向客户端提供响应。如果可以找到这个网页, 那么就把网页的 HTML 代码通过网络传回浏览器; 如果没有找到这个网页, 就发送一个错误信息给发出 HTTP 请求的浏览器。

HTTP 是无状态协议。也就是说, 当使用这种协议的时候, 所有的请求都是为搜索某一个特定的 Web 网页而发出的。它不知道现在的请求是第一次发出还是已经多次发出, 也不知道这个请求的发送来源。当用户请求一个 Web 网页的时候, 浏览器会与相关的 Web 服务器相连接, 检索到这个页面之后, 就会把这个连接断开。

从程序设计的角度来看, 无状态的特点对于 HTTP 来说是一个缺点, 因为这使得某些功能很难实现, 但是由于网络本身的特点, 这也是没有办法改变的。可以假设一下, 如果 HTTP 是有状态的协议, 那么就应该让一个连接长时间地存在下去, 这样就可以判断一个用户到底使用了多长时间, 在这段时间内都做了些什么事情。这样在 Internet 环境中, 一个 Web 服务器要保存太多的连接 (因为在 Internet 环境中, 用户的数量是很难估计的), 大量的连接可能会导致服务器瘫痪。正因为如此, 对于所有的 HTTP 请求, Web 服务器都会以同样的方式进行处理。

1.1.2 Web 服务器

Web 服务器 (或称为 HTTP 服务器) 提供 HTTP 服务, 它不仅是一台物理的机器, 更是一种软件, 可以管理各种 Web 文件, 并为提出 HTTP 请求的浏览器提供 HTTP 响应。本来 Web 服务器只提供“静态”内容, 即返回在 URL 里指定的文件的内容, 一般具备将 URL 名映射到文件名的功能, 并能实施某种安全策略。现在, 可采用 Java Servlet 技术或 ASP.NET 技术从一个运行的程序里得出“动态”内容, 而且现在的 Web 服务器通常还具备连接数据库的功能, 这些就形成了 Web 应用。通常, 一个 Web 服务器还提供其他服务, 如 FTP 服务。有的还可作为代理服务器, 代理服务器是一个可以从别的服务器上为它的客户取文件的服务器。代理服务器可以通过缓存应答 (页面) 使得响应时间变短, 也可以减少网络流量, 对外能隐藏内部网信息。

在大多数情况下, Web 服务器和浏览器处于不同的机器, 但是它们也可以并存在同一台机器上。比较常见的 Web 服务器有 Apache 和 IIS。由于 ASP.NET 只能在 IIS 上运行, 下面只对 IIS 进行介绍。IIS 是 Microsoft 公司的操作系统 Windows 2000/XP 所提供的, 后续章节将详细介绍 IIS。

1.1.3 Web 浏览器

Web 客户通常指的是 Web 浏览器,如 Netscape Navigator 和 Microsoft Internet Explorer。这种浏览器能理解多种协议,如 HTTP、HTTPS、FTP;也能理解多种文档格式,如 TXT、HTML、JPEG (一种图像文件格式)、XML (Extensible Markup Language, 可扩展标记语言);也具备根据对象类型调用外部应用的功能。需要指出的是,HTML 文档中的链接在 Web 浏览器中通常以带下划线的方式显示,用户点击某个链接就能浏览到所链接的 Web 资源,这也是 Web 的魅力所在。

1.1.4 URL

URL (Uniform Resource Locator, 统一资源定位器)用来唯一标识 Web 上的资源,包括 Web 页面、图像文件(如 GIF 格式文件和 JPEG 格式文件)、音频文件(如 AU 格式)、视频文件(如 MPEG 格式文件)。URL 的格式为:协议://主机名[:端口号]/标识符(例如 http://www.163.com:80/index.aspx)。协议可以是 HTTP、HTTPS (安全的超文本传输协议)、FTP;主机名用来标识被请求的服务器;端口通常为不同协议保留,例如,FTP 和 HTTP 守护进程分别侦听不同的端口,FTP 默认的端口号为 21,HTTP 默认的端口号为 80;标识符说明被请求的是什么,可以是文件名(含路径)或一个应用关键字。例如,用户输入 URL 格式的地址: http://www.163.com:80/index.aspx;浏览器请求主机 www.163.com 在 80 端口提供的 HTTP 服务,并要求取得该服务器上的 index.aspx 文件;服务器接受请求,取得该文件后把文件返回浏览器,并告诉浏览器这是一个 HTML 文件;浏览器在显示器上显示这个页面。在浏览器和 Web 服务器之间使用的协议是 HTTP。

1.1.5 HTML

HTML (Hypertext Markup Language, 超文本标记语言)是 Web 产生与发展的要素之一,它旨在使得 Web 页面能显示在任何识别 HTML 的浏览器中,而与连网的机器平台无关。HTML 并不是一个程序设计语言,而是一个标记语言,它所提供的标记是由 SGML (Standard Generalized Markup Language, 标准的通用标记语言)定义的。SGML 是 ISO (国际标准化组织)在 1986 年推出的一个用来创建标记语言的语言标准,它源自 IBM 早在 1969 年开发的 GML (Generalized Markup Language),该语言的名称也正好包含了 3 位创始人姓的第一个字母,他们分别是 Charles F. Goldfarb、Edward Mosher 和 Raymond Lorie。SGML 是一种元语言,即用来定义标记语言的语言,它提供了一种将数据内容与显示分离开来的数据表示方法,使得数据独立于机器平台和处理程序。这些特性促使 Tim Berners-Lee 采用 SGML 来创建称之为 HTML 的标记语言。1993 年形成 HTML 1.0,以后不断完善,HTML 4.0 发表于 1997 年。特别需要指出的是,HTML 提供的链接机制是 Web 的本质特性之一。但是,HTML 更多的是关注 Web 浏览器如何在页面上安排文本、图像和按钮等,过多地考虑外观使其缺乏对结构化数据的表示能力。另外,HTML 中有限的标记不能满足很多 Web 应用的需要,如基于 Web 的大型出版系统和新一代的电子商务,而为各种应用需要不断地往 HTML 中增加标记显然不是最终的解决方法,究其原因 HTML 缺乏可扩展性。解决方案应该是简化 SGML 使之能应用到 Web 上。为此,从 1996 年开始,W3C (World Wide Web Consortium)的一个

工作组在 Jon Bosak 的领导下致力于设计一个超越 HTML 能力范围的新语言, 这个语言后来被命名为 XML (Extensible Markup Language, 可扩展标记语言)。1998 年 2 月, W3C 发布了 XML 1.0 作为其推荐标准。现在, W3C 已经用 XML 设计出一个与 HTML 4.01 功能等价的语言, 称为 XHTML 1.0 (Extensible HyperText Markup Language), 目前该语言得到越来越广泛的应用。

1.2 ASP.NET 的工作原理

ASP.NET 是 Microsoft 公司为了快速构建互联网应用程序而提出的一个统一的 Web 开发模型。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。要详细了解 ASP.NET 的工作原理, 首先要了解什么是 .NET, 什么是 .NET Framework, 什么是公共中间语言。

1.2.1 .NET

.NET 是建立在开放体系结构基础上的一套可以用来构建和运行新一代 Microsoft Windows 和 Web 应用程序的平台, 其目的是简化 Web 开发。.NET 平台包括以下一些核心技术: .NET Framework、.NET 企业服务器、构建模块服务和 .NET 开发工具 (Visual Studio .NET)。

.NET Framework 为 .NET 应用程序提供全新的运行环境和基础类库。所有的 .NET 应用程序必须基于 .NET Framework 才能正常运行。

.NET 企业服务器是指 SQL Server 2000 之类由 .NET Framework 应用程序使用的服务器端产品。它们虽然不是由 .NET Framework 编写成的, 但是它们可以集成、运行、操作、管理 XML、Web Service 和应用程序等。

构建模块服务是以用户为中心的 XML Web Service 集合, 可将用户数据的控制由应用程序转移到用户。例如, Microsoft Passport 是 .NET 首创的一个核心组件, 它使集成各种应用程序变得更容易。

为了能够在 .NET Framework 上进行程序开发, Microsoft 公司把 Visual Studio 进行了升级, 并把升级后的产品命名为 Visual Studio .NET, 用来帮助开发 XML Web Service 以及 Windows 应用程序和 Web 应用程序等, 极大地丰富了用户体验, 这就是 .NET 开发工具。

1.2.2 .NET Framework

.NET Framework 是 .NET 战略的核心。.NET Framework 分为以下几个部分: MS 中间语言、CLR、.NET Framework 类库、.NET 语言、ASP.NET 和 XML Web Service。

MS 中间语言是 .NET 的通用语言。无论使用哪一种 .NET 语言编写的程序代码, 在执行之前, 都会把它编译成为 MS 中间语言。

CLR (Common Language Runtime, 公共语言运行时) 用于执行 MS 中间语言, 为与 .NET Framework 配合使用的任何语言提供了诸如内存管理、安全性和错误处理等功能。

.NET Framework 类库中包含了大量可以实现重要功能的代码库。用户在编写程序的时候可以很方便地把这些库调用到应用程序中, 实现更加复杂的功能。由于类库的存在, 编写功能强大的程序更加容易了。

.NET 语言是指可以将使用其编写的代码编译成为 MS 中间语言的编程语言。常见的语

言有 Visual Basic、Visual C++、C# 和 JScript 等。

XML Web Service 是被运行在 Internet 或 Intranet 上的应用程序所共享的可编程 Web 组件。.NET Framework 提供创建、测试和部署 XML Web Service 的工具和类。.NET Framework 的结构如图 1.1 所示。



图 1.1 .NET Framework 体系结构

1.2.3 公共中间语言

在 .NET Framework 中使用高级语言（例如，Visual Basic、C#）编写的程序，需要在运行前将其编译成为中间语言。需要注意的是，中间语言并不是一种可以直接执行的机器代码。与高级语言编写的代码相比，它的可读性很差，但是进行了一系列的优化。

为了执行中间语言，需要一个执行环境 CLR。CLR 在 .NET Framework 中的地位十分重要，可以说是 .NET Framework 的基础。CLR 用 JIT (Just-In-Time) 编译器把中间语言代码编译成可以执行的代码，并对程序进行最后的、与机器相匹配的优化，使得程序可以在所在计算机上尽可能高效地运行。

早期的编译方式是，把程序代码直接编译成机器代码。这时编译好的程序虽然也进行了与机器相匹配的优化，但是这些优化都是针对编译源代码的机器进行的。如果把编译好的程序放到其他类型的机器上，那么所进行的优化就有可能没有任何意义，并且如果机器的硬件发生变化，那么还有编译后的程序无法执行的可能，因为新的机器可能没有原来机器所拥有的某种资源。而如果采用了公共中间语言的方式，就可以很好地解决这个问题。因为中间语言与机器无关，所以它可以在任何一个可以运行 CLR 的机器上运行。并且所有关于机器的优化都是由 CLR 进行的，所以不存在早期编译所产生的由于机器不同而不兼容的问题。

综上所述，ASP.NET 的工作原理是：首先，有一个 HTTP 请求发送到 Web 服务器要求访问一个 Web 网页。Web 服务器通过分析客户的 HTTP 请求来定位所请求的网页的位置。如果所请求的网页的文件名的后缀是 .aspx，那么就在这个文件传送到 aspnet_isapi.dll 进行处理，由 aspnet_isapi.dll 把 ASP.NET 代码提交给 CLR。如果以前没有执行过这个程序，那么就由 CLR 编译并执行，得到纯 HTML 结果；如果已经执行过这个程序，那么就直接执行编译好的程序并得到纯 HTML 结果。最后把这些纯 HTML 结果传回浏览器作为 HTTP 响应。浏览器接收到这个响应之后，就可以显示 Web 网页了。

1.2.4 ASP.NET 的优势

由此可见, ASP.NET 是一种建立在通用语言上的程序构架, 能被用于一台 Web 服务器来建立强大的 Web 应用程序。与传统的其他 Web 开发模式相比, ASP.NET 提供了以下强大的优势。

(1) 执行效率的大幅提高。在 ASP.NET 中, 基于通用语言的程序在服务器上运行。不像以前的 ASP 即时解释程序, 而是当程序在服务器端首次运行时对其进行编译, 这样的执行效果当然比一条一条地解释强很多。

(2) 世界级的工具支持。ASP.NET 构架是可以用 Microsoft 公司最新的产品 Visual Studio.NET 开发工具进行开发, 实现所见即所得的编辑。这些仅是 ASP.NET 强大的软件支持的一小部分。

(3) 强大性和适应性。因为 ASP.NET 是基于通用语言编译运行的程序, 所以它的强大性和适应性可以使它运行在几乎所有 Web 应用软件开发平台上。通用语言的基本库、消息机制、数据接口的处理都能无缝地整合到 ASP.NET 的 Web 应用中。另外可以选择一种最适合的语言来编写程序, 或者用很多种语言来编写程序, 现在已经支持的有 C#、Visual Basic、JScript 等。将来, 这样的多种程序语言协同工作的能力能够保护现在基于 COM 开发的程序, 能够完整地向 ASP.NET 移植。

(4) 简单性和易学性。ASP.NET 使运行一些很平常的任务如表单的提交、客户端的身份验证、分布系统和网站配置变得非常简单。例如, ASP.NET 页面构架允许建立自己的用户分界面, 使其不同于常见的其他界面。另外, 采用 C# 这样的通用语言使得把各种代码结合成软件变得更加简单。

(5) 高效可管理性。ASP.NET 使用一种基于字符的、分级的配置系统, 使服务器环境和应用程序的设置变得更加简单。因为配置信息都保存在简单文本中, 新的设置有可能不需要启动本地的管理工具就可以实现。在一台服务器上安装一个 ASP.NET 应用程序只需要简单地复制一些必需的文件, 不需要重新启动系统, 比较简单。

(6) 自定义性和可扩展性。ASP.NET 在设计时考虑了让网站开发人员可以在代码中自定义 plug-in 模块。这与原来的包含关系不同, ASP.NET 可以加入自己定义的组件。

(7) 安全性。ASP.NET 与 IIS、.NET 框架和操作系统所提供的基础安全服务配合使用, 共同提供一系列身份验证和授权机制, 并且 ASP.NET 支持不同类型的登录和用户验证, 如 Windows、Forms、Passport 认证技术和每个应用程序配置, 可以确保应用程序的安全。

1.3 ASP.NET 的运行环境

ASP.NET 运行时对软硬件有以下要求。

- (1) 硬件要求。
- CPU: Pentium III 级, 600MHz 以上。
 - RAM: 256MB 以上。
 - 硬盘: 有 800MB 以上自由空间。

(2) 软件要求。

- 操作系统: Windows 2000 Professional、Windows 2000 Server、Windows 2000 Advanced