

计算机应用**能力培养丛书**

ASP.NET 2.0 开发技术 简明教程

田 辉 编著



清华大学出版社

内容简介

亲爱的读者：

ASP.NET 2.0 是微软公司最新推出的 Web 应用程序开发技术，它结合了 ASP 和 .NET 的优点，具有强大的功能和灵活的扩展性。本书是 ASP.NET 2.0 开发技术的入门教材，也是从事 Web 开发工作的必备参考书。本书从 Web 开发的基本概念入手，详细介绍了 ASP.NET 2.0 的语法、控件、数据访问、安全、部署等方面的知识。本书可作为高等院校计算机专业及相关专业的教材，也可作为从事 Web 开发工作的工程技术人员自学之用。

ASP.NET 2.0 开发技术

简明教程

读者资料卡

请您填写以下信息，以便我们为您提供更好的服务。您的信息将被严格保密，仅用于读者调查和售后服务。

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

您使用本书是作为：☐指定教材 ☐教辅用书 ☐个人自学 ☐其他

您对本书的满意程度：

语言文字：☐很满意 ☐比较满意 ☐一般 ☐较不满意

技术含量：☐很满意 ☐比较满意 ☐一般 ☐较不满意

封面设计：☐很满意 ☐比较满意 ☐一般 ☐较不满意

印刷质量：☐很满意 ☐比较满意 ☐一般 ☐较不满意

您希望本书在哪些方面进行修改或改进？(可附页)

☐很满意 ☐比较满意 ☐一般 ☐较不满意

您希望本书在哪些方面进行修改或改进？(可附页)

☐很满意 ☐比较满意 ☐一般 ☐较不满意

您希望本书在哪些方面进行修改或改进？(可附页)

☐很满意 ☐比较满意 ☐一般 ☐较不满意

影响您购买图书的因素：

☐书名 ☐作者名声 ☐出版机构 ☐封面封底 ☐其他

☐价格(□20 元内 □30 元内 □50 元内 □100 元内) ☐其他

☐知名专家学者的书评或推荐 ☐其他

您是如何获得图书信息的：

☐通过书店 ☐通过网站 ☐通过杂志 ☐通过其他

您对我们的建议：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：

姓名： 性别：☐男 ☐女 年龄： 职业： 单位： 电话： 电子邮箱：



清华大学出版社

010-62794504 或发电子邮件至 wkservice@tup.tsinghua.edu.cn

北京

内 容 简 介

本书共 14 章,从使用 ASP.NET 2.0 进行 Web 应用程序开发的角度,较为全面地介绍了 ASP.NET 2.0 的各种基础知识和开发技术。针对初学者的特点,本书从 Web 页和 Web 站点的基础知识、Visual C# 的基本语法和数据库的基础知识入手,逐步过渡到数据库的读取、写入和管理,组件化编程、状态管理、WebServices 等高级话题,易于学习和掌握。为了提高读者的实际开发技能,第 14 章提供了 3 个实训,它们具有较强的代表性。附录 B 还专门针对数据库连接字符串的关键字和连接池配置进行了详细的解释和说明。

本书内容丰富,结构清晰,核心概念和关键技术讲解清楚,同时提供了丰富的示例来展示具体应用,具有很强的操作性和实用性,本书可作为高等学校、高职学校,以及社会各类培训班“ASP.NET 程序设计”课程的教材,同时对从事 ASP.NET Web 编程或开发的人员也具有较高的参考价值。

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 开发技术简明教程/田辉 编著. —北京:清华大学出版社,2008.3

(计算机应用能力培养丛书)

ISBN 978-7-302-17085-3

I. A… II. 田… III. 主页制作—程序设计—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字(2008)第 021387 号

责任编辑:王 军 鲍 芳

装帧设计:康 博

责任校对:胡雁翎

责任印制:杨 艳

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 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:20.75 字 数:514 千字

版 次:2008 年 3 月第 1 版 印 次:2008 年 3 月第 1 次印刷

印 数:1~5000

定 价:28.00 元

清华大学出版社

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:025211-01

前言

2005 年 10 月

高职高专教育以就业为导向,以技术应用型人才为培养目标,担负着为国家经济高速发展输送一线高素质技术应用人才的重任。近年来,随着我国高等职业教育的发展,高职院校数量和在校生人数均有了大幅激增,已经成为我国高等教育的重要组成部分。

根据目前我国高级应用型人才的紧缺情况,教育部联合六部委推出“国家技能型紧缺人才培养培训项目”,并从 2004 年秋季起,在全国两百多所学校的计算机应用与软件技术、数控项目、汽车维修与护理等专业推行两年制和三年制改革。

为了配合高职高专院校的学制改革和教材建设,清华大学出版社在主管部门的指导下,组织了一批工作在高等职业教育第一线的资深教师和相关行业的优秀工程师,编写了适应新教学要求的计算机系列高职高专教材——《计算机应用能力培养丛书》。该丛书主要面向高等职业教育,遵循“以就业为导向”的原则,根据企业的实际需求来进行课程体系设置和教材内容选取。根据教材所对应的专业,以“实用”为基础,以“必需”为尺度,为教材选取理论知识;注重和提高案例教学的比重,突出培养人才的应用能力和实际问题解决能力,满足高等职业教育“学校评估”和“社会评估”的双重教学特征。

每本教材的内容均由“授课”和“实训”两个互为联系和支持的部分组成,“授课”部分介绍在相应课程中,学生必须掌握或了解的基础知识,每章都设有“学习目标”、“实用问题解答”、“小结”、“习题”等特色段落;“实训”部分设置了一组源于实际应用的上机实例,用于强化学生的计算机操作使用能力和解决实际问题的能力。每本教材配套的习题答案、电子教案和一些教学课件均可在该丛书的信息支持网站(<http://www.tupwk.com.cn/GZGZ>)上下载或通过 Email(wkservice@tup.tsinghua.edu.cn)索取,读者在使用过程中遇到了疑惑或困难可以在支持网站的互动论坛上留言,本丛书的作者或技术编辑会提供相应的技术支持。

如今,在开发复杂的 Web 站点或 Web 应用程序时,使用 ASP.NET 2.0 是很不错的选择。ASP.NET 2.0 构建在 .NET Framework 2.0 之上,它扩展了 ASP.NET 1.0 和 1.1 的功能,其内核是一个基于控件的、事件驱动的架构,开发者只需添加很少的代码,便可以根据用户的输入提供平滑、直观的用户体验。ASP.NET 2.0 还提供了强大的定制功能,这在本书中都有体现。

本书依据教育部《高职高专教育计算机公共基础课程教学基本要求》编写而成,全面介绍了使用 ASP.NET 2.0 进行 Web 页制作和开发的工具和技术。全书共 14 章,主要内容包括 Web 应用程序的基本概念和 ASP.NET 2.0 开发环境,Visual C# 的基本语法,ASP.NET 2.0 的标准服务器控件及其使用方法,如何设计 Web 站点,如何配置 Web 站点的安全性,数据库的基础知识,如何使用数据源控件和数据绑定控件对后台数据库读取和写入数据,ASP.NET 2.0 的组件化编程,创建定制页面和站点,页面和应用程序的状态管理,WebServices,调试和部署 Web 站点等。

本书的重点是 Web 应用程序的开发,而不是 Web 页的设计,因而对 Web 页界面设计的内容虽然有所涉及但并不深入介绍。为提高读者的实际开发技能,本书提供了大量的示

由于计算机科学技术发展迅速,再者受自身水平和编写时间所限,书中如有错误或不足之处,欢迎广大读者对我们提出意见或建议。

作者

目

录

80	登录页的创建	1.2.2
103	网站的部署	2.2.2
103	网站的部署	2.2.2
106	网站的部署	2.2.2
第 1 章 导论 1		
1.1	Web 应用程序概述	1
1.1.1	Web 应用程序的工作原理	1
1.1.2	常用的 Web 应用程序开发技术	3
1.2	ASP.NET 概述	4
1.2.1	.NET Framework 与 ASP.NET	4
1.2.2	ASP.NET 应用程序的执行过程	5
1.2.3	ASP.NET 的功能和特点	5
1.3	建立运行环境	6
1.3.1	安装 IIS	6
1.3.2	设置 IIS	7
1.3.3	配置虚拟目录	9
1.3.4	安装 .NET Framework	10
1.4	ASP.NET 的开发工具	10
1.5	创建一个简单的 ASP.NET 应用程序	10
1.5.1	创建和编辑 Web 站点	10
1.5.2	运行 ASP.NET 应用程序	13
1.5.3	配置 ASP.NET Web 站点	14
1.5.4	开发 ASP.NET 应用程序的注意事项	17
1.6	自定义 ASP.NET 开发环境	17
1.6.1	设计区	18
1.6.2	工具箱	18
1.6.3	属性窗口	19
1.6.4	错误列表窗口	20
	本章小结	21
	习题	21
第 2 章 Visual C# 程序设计基础 23		
2.1	Visual C# 简介	23

3.2.4	控件的声明	24
3.3	使用标准 Web 服务器控件	25
3.3.1	Label 控件	25
3.3.2	TextBox 控件	25
3.3.4	DropDownList 控件	26
2.2	变量和基本数据类型	24
2.2.1	变量的声明和使用	24
2.2.2	C# 的基本数据类型	25
2.2.3	常量	26
2.3	对象	27
2.3.1	基本概念	27
2.3.2	对象的生命周期	28
2.3.3	创建并使用类	28
2.3.4	命名空间和 using 语句	30
2.4	高级数据类型	31
2.4.1	数组和集合	31
2.4.2	结构	33
2.4.3	枚举	34
2.5	理解装箱和拆箱	34
2.6	泛型	35
2.7	语句	36
2.7.1	运算符和表达式	36
2.7.2	条件语句	39
2.7.3	循环语句	41
2.8	理解作用域	44
2.8.1	理解局部作用域	44
2.8.2	理解类作用域	44
	本章小结	45
	习题	45
第 3 章 使用 ASP.NET 2.0 服务器控件 47		
3.1	服务器控件概述	47
3.1.1	什么是服务器控件	47
3.1.2	服务器控件的类型	47
3.2	基本操作	48
3.2.1	添加控件	48
3.2.2	编辑控件属性	49
3.2.3	控件的事件	50

3.2.4 控件的对齐与删除	51
3.3 使用标准 Web 服务器控件	52
3.3.1 Label 控件	52
3.3.2 TextBox 控件	54
3.3.3 RadioButton 控件	55
3.3.4 DropDownList 控件	56
3.3.5 Button 控件	58
3.4 服务器控件的工作原理	59
3.4.1 HTTP 协议	61
3.4.2 服务器端事件	61
本章小结	62
习题	63
第 4 章 使用 ASP.NET 统筹 Web 站点	65
4.1 Web 站点的设计目标和基本原则	65
4.1.1 设计目标	65
4.1.2 基本原则	66
4.2 创建 Master 和 Content 页面	66
4.2.1 创建 Master 页面	67
4.2.2 创建 Content 页面	68
4.2.3 多层次的 Master 页面	71
4.2.4 Master 页面的内容占位符	73
4.3 创建站点地图和导航	73
4.3.1 创建站点地图	74
4.3.2 创建站点导航	75
4.4 设置站点样式	81
4.4.1 使用 style 属性设置页面元素样式	81
4.4.2 使用 CSS 级联样式表	82
4.4.3 使用主题(Theme)	85
4.5 样式和布局的最佳实践	94
4.5.1 站点的可用性	94
4.5.2 站点的可访问性	94
本章小结	95
习题	95
第 5 章 保护 Web 应用程序	97
5.1 基本的安全术语	97
5.2 构建安全的 Web 站点	98

5.2.1 创建站点的登录系统	98
5.2.2 站点的个性化设置	103
5.2.3 为站点添加用户角色	103
5.2.4 身份验证技术	106
5.3 角色和配置文件	109
5.3.1 为 LoginView 控件编辑角色	110
5.3.2 使用角色锁定页面内容	111
5.3.3 基于角色的安全性	113
5.3.4 构建配置文件	114
本章小结	115
习题	116
第 6 章 数据库基础知识	117
6.1 数据库概述	117
6.1.1 数据的存储手段	117
6.1.2 了解 SQL Server 2005	118
6.1.3 SQL Server 的基本术语	119
6.2 创建和维护 SQL Server 数据库	120
6.2.1 规划数据库	120
6.2.2 创建数据库	121
6.2.3 规划数据表	122
6.2.4 创建数据表	123
6.2.5 修改数据表	124
6.3 T-SQL 语言	125
6.3.1 T-SQL 变量和常量	125
6.3.2 Select 语句	126
6.3.3 Insert 语句	128
6.3.4 Update 语句	129
6.3.5 Delete 语句	130
6.4 XML 数据	130
6.4.1 XML 标记	131
6.4.2 XML 属性	131
6.4.3 XML 声明	132
6.4.4 XML 文档的结构	132
6.4.5 XML 命名空间	133
本章小结	133
习题	134

第 7 章 读取数据	135
7.1 使用数据源控件绑定数据库	135

7.1.1	数据源控件的常见类型	135
7.1.2	创建数据源	136
7.1.3	隐藏连接字符串	140
7.2	使用数据绑定控件	141
7.2.1	数据绑定控件的类型	141
7.2.2	使用 GridView 表格控件	143
7.2.3	使用 DataList 和 Repeater 控件	147
7.2.4	使用 DetailsView 和 FormView 控件	150
7.3	使多个数据控件协同工作	152
7.4	创建并读取 XML 文档	156
	本章小结	158
	习题	158
第 8 章	写入数据	159
8.1	写入数据概述	159
8.1.1	数据控件的信息组织方式	159
8.1.2	数据写入的相关术语	160
8.1.3	写入数据的控件选择	160
8.1.4	DataKeyNames 属性	160
8.2	更新数据库中的记录	161
8.3	向数据库插入记录	164
8.4	删除数据库中的记录	165
8.5	输入数据的有效性验证技术	166
8.5.1	理解服务器端验证和客户端验证	166
8.5.2	使用验证控件	167
8.6	上传文件或图片	171
	本章小结	176
	习题	176
第 9 章	组件化——深入理解 Web 应用程序的开发结构	179
9.1	代码与内容、设计的分离	179
9.2	后台代码	180
9.2.1	Page 指令	180
9.2.2	不完全类	180
9.2.3	事件处理程序/函数	181
9.2.4	创建并编译后台代码文件	182

9.3	数据层	184
9.3.1	两层应用程序	184
9.3.2	三层应用程序	184
9.3.3	创建业务组件	186
9.4	用户控件	189
9.4.1	用户控件与服务器控件的区别	189
9.4.2	用户控件的结构	189
9.4.3	创建用户控件	190
9.5	定制控件	195
9.5.1	定制控件与程序集的关系	195
9.5.2	定制控件与用户控件的区别	195
9.5.3	定制控件的创建方法和结构	195
	本章小结	198
	习题	199
第 10 章	Web Parts——创建可定制的面板或站点	201
10.1	Web Parts 控件概述	201
10.1.1	Web Part 与用户控件、定制控件(自定义控件)	202
10.1.2	Web Part 的定制功能	203
10.1.3	何时使用 Web Part	204
10.1.4	个性化及其框架	205
10.2	创建定制页面	206
10.3	在 WebPartZone 中添加控件	207
10.3.1	在 Web Part 中封装控件	208
10.3.2	访问 Web Part 中的控件	208
10.4	配置 WebPartZone 定制区域	209
10.4.1	配置定制操作	210
10.4.2	配置定制菜单	211
10.4.3	设置定制区域样式	211
10.4.4	设置定制模式	212
10.4.5	使用定制工具框架控件	214
10.4.6	配置定制工具框架控件	218
	本章小结	219
	习题	219
第 11 章	状态管理	221
11.1	使用 ViewState	221

11.1.1	直接访问 ViewState	222
11.1.2	管理控件的状态	225
11.1.3	与高速缓存(Cache)集成	227
11.1.4	对象的高效串行化	229
11.2	客户端状态管理的其他方法	232
11.2.1	隐藏表单字段	232
11.2.2	cookie	233
11.2.3	查询字符串	235
11.2.4	在窗体之间传递服务器控件的值	236
11.3	服务器端的状态管理	237
11.3.1	应用程序状态	237
11.3.2	会话状态	238
	本章小结	239
	习题	239
第 12 章	WebServices	241
12.1	WebServices 概述	241
12.1.1	基本术语	242
12.1.2	查看和使用第三方 WebServices	242
12.1.3	WebServices 的生存期	243
12.1.4	WebServices 的结构	246
12.2	创建和调用 WebServices	248
12.2.1	创建 WebServices	248
12.2.2	测试 WebServices	252
12.2.3	发现 WebServices	253
12.2.4	调用 WebServices	255
12.3	WebServices 的安全性	256
12.3.1	加密和基于消息的安全性	256
12.3.2	服务的身份验证和访问控制	257
	本章小结	257
	习题	258
第 13 章	调试和部署	259
13.1	编写防御性代码	259
13.1.1	参数检查	259
13.1.2	避免假设	260

13.1.3	防止 SQL 注入攻击	261
13.1.4	使用验证	262
13.2	异常处理	262
13.2.1	异常类	262
13.2.2	Exception 对象	263
13.2.3	捕获异常	263
13.2.4	记录异常	265
13.2.5	邮寄异常	266
13.2.6	全局异常处理	268
13.2.7	异常处理的基本原则	269
13.3	Visual Studio 2005 调试环境	269
13.3.1	设置调试工具	269
13.3.2	理解语法和错误消息	270
13.3.3	理解调试工具窗口	271
13.3.4	设置并使用断点	272
13.4	使用跟踪和调试器	272
13.4.1	使用 ASP.NET 跟踪	272
13.4.2	使用调试器	276
13.5	站点的部署	278
13.5.1	部署前的工作	278
13.5.2	Xcopy 部署	280
13.5.3	复制站点	280
13.5.4	发布站点	282
13.5.5	创建安装程序	284
	本章小结	286
	习题	286
第 14 章	实训	287
实训 1	设计网页验证码功能	287
实训 2	设计网页搜索功能	290
实训 3	设计数据库的动态安装与配置	297
附录 A	.NET Framework 下常用命名空间	311
附录 B	Web 服务器控件及其说明	315
附录 C	ConnectionString 数据库连接字符串说明	317

第 1 章

导 论

本章介绍 Web 应用程序的工作原理及常用开发技术, ASP.NET 的发展历程、与 .NET Framework 的关系、功能、特点、运行环境和开发工具, 创建和开发 ASP.NET 应用程序的工作流程和注意事项等内容。通过本章的学习, 应完成以下学习目标:

- ✓ 理解并掌握网页的工作原理
- ✓ 了解 Web 应用程序的常用开发技术
- ✓ 了解 ASP.NET 与 .NET Framework 的关系
- ✓ 了解 ASP.NET 的功能和特点
- ✓ 学会搭建 ASP.NET 应用程序的运行环境
- ✓ 熟悉 Visual Studio 2005 的 ASP.NET 开发环境
- ✓ 学会创建一个简单的 ASP.NET 应用程序
- ✓ 了解开发 ASP.NET 应用程序开发过程中的注意事项

1.1 Web 应用程序概述

应用程序主要有两种模式: C/S(客户机/服务器)模式和 B/S(浏览器/服务器)模式。C/S 模式主要用于 Windows 应用程序的开发, 这类应用程序常见的如 Microsoft 的 Office 办公套件; B/S 模式则主要用于 Web 应用程序的开发, 这类应用程序基于网络, 具有与网页浏览者进行交互的功能, 如聊天室、留言板、电子商务等。

1.1.1 Web 应用程序的工作原理

1. Web 服务器与客户端

一般而言, 凡是提供服务的一方即称为服务器端, 而接受服务的一方则称为客户端。例如浏览新浪首页的时候, 新浪首页所在的服务器就称为服务器端, 而访问者的计算机称为客户端。但服务器端和客户端也并不绝对, 例如提供服务的服务器如果接受了别的服务器端的服务, 它也就成了为其提供服务的服务器端的客户端, 对于客户端则同样如此。我们在调试自己的 Web 应用程序时, 通常是将自己的计算机既当服务器端, 又当客户端来使用。

Internet 上的用户通过在浏览器地址栏输入 Web 服务器的地址来访问 Web 服务器上的资源。我们做的网站, 如果希望网上的用户能够访问, 也必须将网站文件放在 Web 服务器

上。放在自己机器上, Internet 上的其他人是访问不到的(除非您的机器本身就是一台 Web 服务器)。

网站发布到 Web 服务器上以后, Internet 上的用户通过网站的域名来访问网站。我们知道, IP 可以用于标记一台主机的位置, 但用户不可能通过 IP 地址来访问网页, 因为这太难以记忆了。而域名就相对容易记忆得多, 当 Internet 上的用户在浏览器地址栏输入域名后, 会通过服务器对该域名进行解析, 自动找到该 IP 地址所在的主机, 从而访问该 Web 服务器。

2. 交互原理

要查看某个网页, Internet 上的用户必须首先打开浏览器, 在地址栏中输入该网页的 URL 地址, 然后由浏览器向指定地址的 Web 服务器发出请求, 这称为客户端请求。客户端请求是包含一连串客户端信息的字符串, 包括诸如客户端请求类型、浏览器类型、操作系统类型、客户端的 IP 地址等信息。

当 Web 服务器接收到这些信息后, 就会根据客户请求的文件类型作相应的处理: 如果用户请求的页面是一个静态页, 那么服务器是不对用户请求的数据作任何处理的, 它将直接把用户所请求页面的 HTML 发给客户机; 如果用户请求的是一个动态页, 那么服务器会执行动态页中的动态程序代码片段(如 ASP、JSP、PHP 等), 然后将动态程序生成好的 HTML 代码发送给客户端。

当客户端收到这些 HTML 信息后, 会把它们存储在客户端本地计算机的一个临时文件夹中(这种存储过程称之为缓冲或缓存), 最后将临时文件夹中的 HTML 装入到客户的浏览器中, 由浏览器将 HTML 代码解释成浏览者所能看到的网页信息。这就是整个客户机与 Web 服务器之间请求与应答的交互过程。

3. 选择静态网页还是动态网页

所谓静态网页, 是指由网页设计者最初设计网页时, 网页中的所有元素(如文字内容、表格和图像等等)都由设计者设计好了的网页。并且这个网页发布到服务器上后, 不能由用户的相关操作(如表单提交、单击链接), 来动态地改变网页的显示形式(包括版式和内容)。如果要更改网页中的内容, 则必须由设计者重新对该页进行设计。

所谓动态网页。是指在用户浏览网页的时候, 可以通过相关的用户事件, 来动态地改变网页中的内容甚至版式的一种网页。例如网页中显示用户的浏览次数, 即是一个典型的动态网页, 它随浏览者访问该页的次数而显示不同的数字。

静态网页最大的缺点是没有交互性, 不能够实现浏览者与网站之间的信息交互。另外, 由于它是直接把 HTML 代码发送给客户机的, 因此, 没有安全性可言。在现今的信息网络中, 没有交互性的网页已经形同虚设, 而更多的网站都采用了各种各样的动态技术。但是, 不容质疑, 静态网页比任何一种动态网页的速度都要快, 因为它在服务器端不需要进行任何处理, 由服务器直接将 HTML 发给客户机。

而动态网页需要在服务器中对数据进行处理, 将处理好后生成的 HTML 代码发给客户机, 所以它一般要比静态网页的速度慢, 再者, 由于动态网页的代码是在服务器端执行的, 而客户机看到的始终是动态程序生成后的 HTML 源代码, 所以它有了数据的安全性。

由此, 大家可能觉得动态页都要优于静态网页, 那么直接学习动态网页就好了, 而这样的想法却是大错! 要知道, 任何一种动态网页都是建立在 HTML 的基础上的, 动态网页

的实质就是将动态的代码嵌入到 HTML 中,来实现各种交互性的。任何一种动态网页执行后,都将生成 HTML 代码,并最终将 HTML 发给客户机。所以,可以说一个不会 HTML 的人,是不可能学好动态网页的,它是所有动态语言的基础。

1.1.2 常用的 Web 应用程序开发技术

早期的动态网页主要采用 CGI(Common Gateway Interface)技术,即公用网关接口技术,虽然 CGI 技术已经发展成熟并且功能强大,但由于编程困难、效率低下、维护困难,已经逐渐被新兴的技术所取代,这些新兴技术主要有 ASP、PHP、JSP、ASP.NET,采用这些动态网页技术开发的网页文档后缀名通常为 .asp、.php、.jsp、.aspx。

1. PHP(Hypertext Preprocessor)

PHP 是 Rasmus Lerdorf 于 1994 年提出的,最初仅提供留言簿、计数器等简单功能,后来其解析器被不断修改和完善(当前最新版本为 PHP5),并逐步成为当今 Internet 上最流行的 Web 开发技术之一。PHP 大量借鉴了 C、Java 和 Perl 语言的语法,将脚本语言嵌入到 HTML 文档中,代码在服务器端执行,转化为标准的 HTML 文件后发送到客户端并呈现。

PHP 程序可以运行在 Unix、Linux 和 Windows 操作系统下,PHP、MySQL 数据库、Apache Web 服务器是当前开发 Web 应用程序的一个完美组合。但 PHP 缺乏大公司的支持,其运行环境配置起来比较复杂,并且由于商业原因,采用 PHP 开发的 Web 应用程序存在较多的安全隐患。

2. JSP(Java Server Pages)

JSP 是由 Sun Microsystem 公司于 1999 年 6 月推出的,基于 Java Servlet 和整个 Java 体系的 Web 开发技术。JSP 其实就是将 Java 程序片段和 JSP 标记嵌入到普通的 HTML 文档中,当用户访问一个 JSP 网页时,就执行其中的程序片段,然后将标准的 HTML 文档返回给客户端并呈现。

JSP 的执行效率很高,当用户第一次请求 JSP 文件时,该文件被编译成 Servlet 并由 Java 虚拟机执行,用户以后再请求该 JSP 文件时便可以不用编译而直接运行了。此外, JSP 可以运行在几乎所有的服务器系统上,包括 Windows NT、Windows 2000、Unix、Linux 等,但 JSP 的运行环境配置起来比较复杂,并且学习起来比较困难,不是十分适合初学者使用。

3. ASP(Active Server Pages)

ASP 是由 Microsoft 开发的一种类似 HTML(超文本标识语言)、Script(脚本)与 CGI(公用网关接口)的结合体,它没有提供自己专门的编程语言,而是在普通的 HTML 文档中嵌入 VBScript 或 JavaScript 脚本,当客户请求一个 ASP 文件时,服务器端就运行 ASP 文件中的脚本代码,并转化为标准的 HTML 文件,最后发送到客户端并呈现。ASP 当前的最新版本是 3.0。

ASP 的最大优点是简单、易学,且有 Microsoft 的强大支持。但 ASP 存在很多缺陷,具体表现在以下几个方面。

- 编码量大:即使是做一些非常简单的事情,例如显示一个数据库中的数据页,也必须手工编写大量的代码,程序员工作量太大。

- 程序可读性和维护性差：ASP 将应用程序代码嵌入到 HTML 页面中，使得应用程序代码和 HTML 代码混合在一起，造成程序的可读性和维护性较差；
- 性能不理想：即使是相同的 HTTP 请求，ASP 也要重新运行相同的代码，造成程序的性能低下。

此外，ASP 不能跨平台运行，一般只能运行在 Windows 系列的操作系统上。为此，Microsoft 在 ASP 3.0 的基础上推出了 ASP.NET 技术。

4. ASP.NET 与 ASP 相比，ASP.NET 不是简单的升级，而是进行了全面的革新，是一种全新的开发架构，1.2 节将对其做全面介绍。

1.2 ASP.NET 概述

ASP.NET 是 Microsoft 所推出的一个全新的 Web 开发架构，可构建能在任意浏览器 (Internet Explorer、Netscape Navigator、Firefox 等) 上显示的 Web 应用程序。

1.2.1 .NET Framework 与 ASP.NET

Microsoft 所极力打造和推广的 .NET 战略主要是为了解决互联网中的“信息孤岛”问题，打破不同的上网设备、操作系统、网站，以及各大机构和工业界的网络障碍，解决网站之间的协同合作，使网站之间形成自动交流，从而最大限度地共享网络中的各种资源。

作为第一步，Microsoft 推出了 Visual Studio .NET，它是一个使用 Visual Basic.NET (简称 VB.NET)、C#、C++.NET 等语言开发应用程序的总框架，包括各种开发工具、支持组件、运行环境、示例文档等。.NET Framework 主要由公共语言运行库 (CLR, Common Language Runtime)，基础类库 (Base Class Library)，ASP.NET、VB.NET、C# 等语言运行库组成，如图 1-1 所示 (显示的是最新的 .NET Framework 3.0)。使用 ASP.NET 开发 Web 应用程序，实际上就是使用 .NET Framework 的基础类库和 Web 开发构件来编写应用程序。

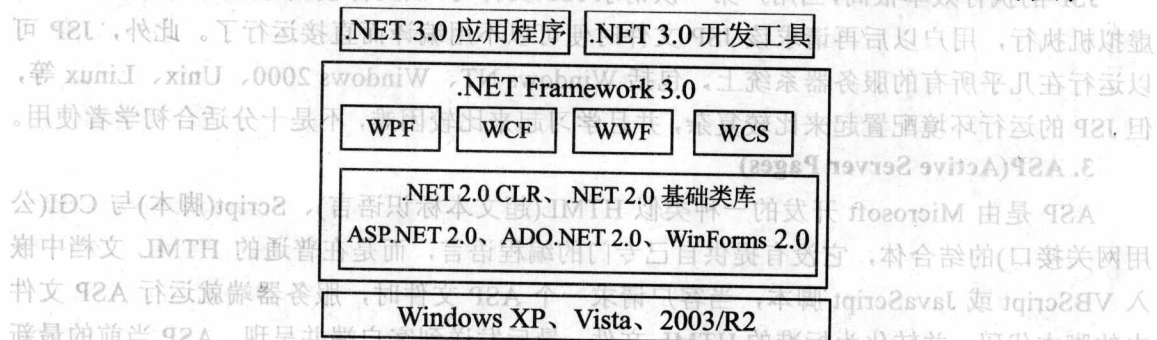


图 1-1 .NET Framework 与 ASP.NET 的关系

总结：Visual Studio 2005 是 Visual Studio .NET 的升级版本，集成的是 .NET Framework 2.0。 .NET Framework 3.0 并没有对 .NET Framework 2.0 做太大改进，而是引入了 4 种基本的新技术：Windows 描述基础 (WPF)、Windows 通信基础 (WCF)、Windows 工作流基础 (WWF) 和 Windows CardSpace (WCS)。

1.2.2 ASP.NET 应用程序的执行过程

ASP.NET 使用 IIS(Internet Information Server)来传送内容,以响应 HTTP 请求。图 1-2 说明了一个典型的 Web 请求过程,客户机向服务器请求浏览一个 Web 页,例如 default.aspx(所有的 ASP.NET Web 页扩展名均为 aspx),ASP.NET 运行库开始工作,对该 Web 页第一次请求时会首先启动 ASP.NET 分析器,编译器将 default.aspx 文件和与此相关的 C#文件一起编译,并创建一个程序集。该程序集包含一个 Page 类,通过该类将 HTML 代码返回给客户端,之后该类会被删除,但该程序集会被保留,以用于日后的请求。最后,JIT 编辑器将程序集编译成机器码并执行。

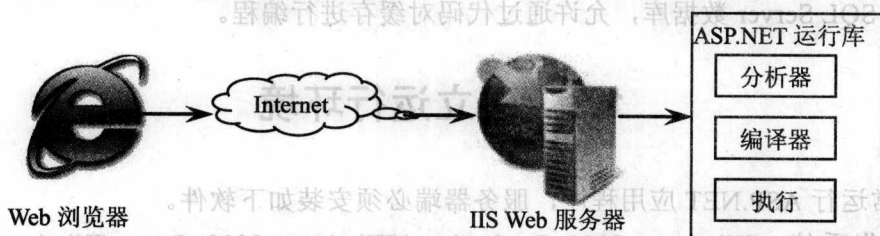


图 1-2 ASP.NET 应用程序的执行过程

1.2.3 ASP.NET 的功能和特点

作为新一代的 Web 开发架构,开发人员可以在 ASP.NET 下采用 VB.NET、C#等其他.NET 语言来开发 Web 应用程序,相对于 ASP,ASP.NET 具有以下显著优势。

- **效率增强:** ASP.NET 应用程序可以在 Visual Studio 2005 中开发,具有所见即所得、控件拖放和自动部署等特性,从而极大提高开发效率。ASP 采用解释执行的方式,用户每访问一次页面,就需要从头到尾解析一遍,效率十分低下,而 ASP.NET 借鉴了 PHP,采用先编译后运行的方式,用户第一次访问页面时进行编译,以后再访问时就直接运行,从而极大提高了执行效率。
- **程序可读性好、易于维护:** 在 ASP 中,ASP 代码和 HTML 标记是混合在一起的,程序的可读性较差,维护起来也很麻烦。而 ASP.NET 将程序代码和 HTML 标记分开,使得程序结构十分清晰。此外,在 ASP 中,如果要使用第三方组件,就必须在服务器端注册该组件,而在 ASP.NET 中,可以直接向目标服务器复制组件。
- **一致性和个性化:** 在 Web 设计过程中,一方面希望整个站点具有统一的外观,但又希望在色彩、字体大小和其他特性上为用户提供定制站点的服务。ASP.NET 2.0 提供了 MasterPage 控件,使得一个站点在标题、菜单和链接的布局上保持一致,在这种一致范围内,设计人员可以添加控件,设计富于个性化的用户体验。
- **提供 Web 窗体的关系式程序模型。** Web 窗体中包含与事务处理逻辑区分开的表示逻辑和代码文件,开发人员可以在 Web 窗体设计器中通过拖拉各种 UI 控件来设计用户界面,系统将自动生成所有代码,同时也可以在任何一种.NET 语言来编写代码。ASP.NET Web 窗体会在 Web 服务器上编译和缓存所有代码,有效提升性能。
- **提供支持服务器端事件的服务器(Server)控件,**但呈现为 HTML,允许在任何支持

HTML 的浏览器上运行，此外，Microsoft 还扩展了许多 HTML 控件，使得开发人员可以在代码中自由地操作它们。

- 登录、安全和角色：许多 Web 站点都需要登录系统，虽然它实现起来并不复杂，但很少有程序员能够开发一个符合安全要求的登录系统。使用 ASP.NET 2.0 提供的登录控件，设计人员可以轻松创建站点的登录系统，且具有很高的安全性，可为不同用户指定角色、权限等。
- 提供强大的数据控件来显示、编辑和处理来自一个数据库中的数据。
- 为了缓存客户端状态信息，允许在客户端上使用 Cookie、ASP.NET State 服务或一个 SQL Server 数据库，允许通过代码对缓存进行编程。

1.3 建立运行环境

要正常运行 ASP.NET 应用程序，服务器端必须安装如下软件。

- 操作系统：Windows 2000 Professional/Windows 2000 Server/Windows Advance Server/Windows Server 2003/Windows XP Professional 或更高版本，对于 Windows 2000 系列则需要安装 Service Pack 2.0，建议读者使用 Windows Server 2003。
- Web 服务器：IIS 5.5 或更高版本。
- .NET Framework。
- MDAC(Microsoft 数据访问组件)2.7。

对于客户端而言，只要是普通的浏览器即可，但 Internet Explorer 版本必须在 5.0 以上，推荐使用 7.0。

对于运行 ASP.NET 应用程序时的硬件而言，当前主流的 PC 机均可满足要求。

由于读者在开发 ASP.NET 应用程序时，通常是先在自己的计算机上编写并调试好后，再部署到 Web 服务器上的，这样自己的计算机就既是服务器，又是客户端，因而必须同时安装服务器端和客户端必需的软件。

1.3.1 安装 IIS

IIS 是由 Microsoft 开发、以 Windows 操作系统为平台的 Web 服务器软件，对于 Windows 2000 或更高版本的 Windows NT 版本，IIS 已经是内建的服务，因此不需要另外配置。对于 Windows XP 或 Windows 2000 专业版，则必须手动安装并配置 IIS 服务，否则将无法测试和运行 ASP.NET 应用程序。要检查系统是否已配置 IIS 服务，方法是查找 C:\Inetpub 文件夹，如果不存在，则需要及时安装。下面以 Windows XP 版本为例，介绍一下 IIS 的安装与设置方法。

例 1-1 安装并测试 IIS 服务。

① 在计算机光驱中放入 Windows XP 安装盘，在任务栏执行【开始】|【设置】|【控制面板】命令，打开【控制窗口】面板。

② 双击【添加或删除程序】图标，在打开的对话框中单击【添加/删除 Windows 组件】按钮，在打开的【Windows 组件向导】对话框中，选中【Internet 信息服务】复选框，单击

【下一步】按钮，开始安装 IIS，如图 1-3 所示。

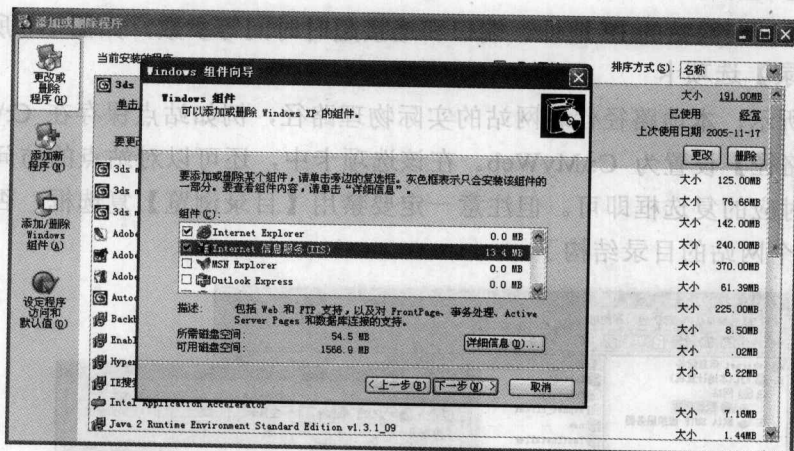


图 1-3 安装 IIS 信息服务软件

③ IIS 安装完成后，需要对其进行测试，判断是否正确安装。打开浏览器，在地址栏输入 <http://localhost>，如果安装成功无误，将会同时打开两个窗口，如图 1-4 所示。

④ IIS 本身提供 FTP、发送电子邮件、网页浏览等服务。当利用 IIS 创建网站时，所有网页都必须存放在“网站”这个特定目录中，IIS 会根据指定的网址对应路径，将其中子文件或子目录对应到计算机上存储网页文件的真正位置。在控制面板中双击【管理工具】图标，打开【管理工具】窗口，双击【Internet 信息服务】图标，打开【Internet 信息服务】窗口，如图 1-5 所示。

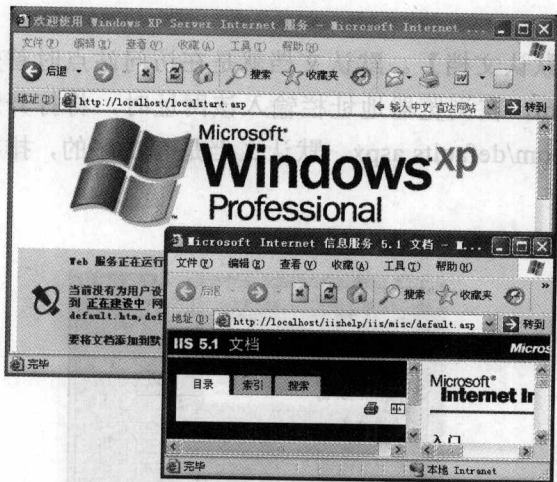


图 1-4 测试 IIS

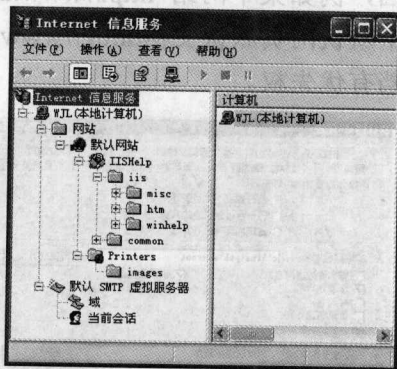


图 1-5 浏览 IIS 目录窗口

1.3.2 设置 IIS

打开【Internet 信息服务】窗口，展开【网站】节点，右击【默认网站】节点，在弹出的快捷菜单中单击【属性】命令，打开该【默认网站属性】对话框。

1. 【网站】选项卡

在这里可以设置网站的 IP 地址、端口和连接超时时间等参数，如图 1-6 所示。

2. 【主目录】选项卡

如图 1-7 所示，本地路径代表网站的实际物理路径，例如站点保存在 C:\MyWeb，就需要将【本地路径】设置为 C:\MyWeb。在该选项卡中，还可以对站点的访问进行权限设置，只需选中对应的复选框即可。但注意一定要禁用【目录浏览】复选框，否则就只能通过 Web 浏览整个网站的目录结构了。

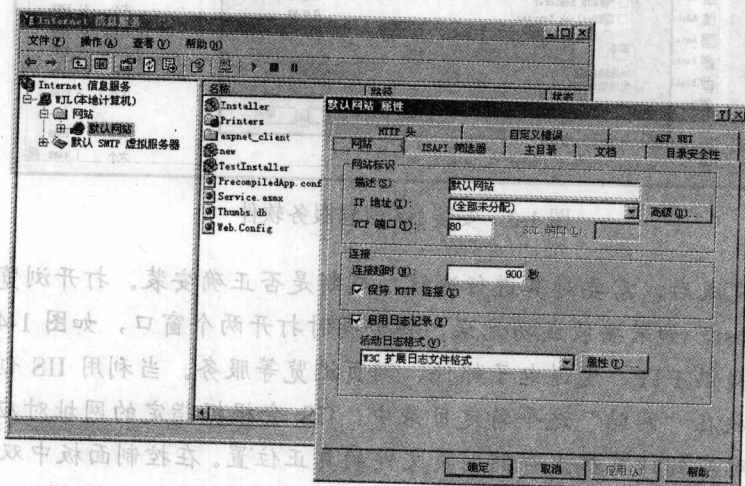


图 1-6 【网站】选项卡

3. 【文档】选项卡

如图 1-8 所示，系统默认启用【启用默认文档】。默认文档就是在访问站点时首先打开的文档，例如某个网站 <http://www.a.com>，在浏览器地址栏输入该网址时，将打开该网站的某个默认网页默认文档 <http://www.a.com/defaults.aspx>。默认文档是有次序的，排在前面的文档有优先权。

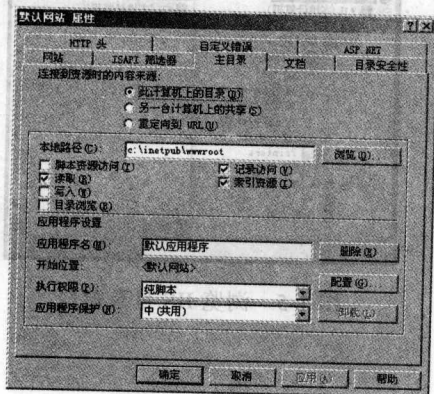


图 1-7 【主目录】选项卡

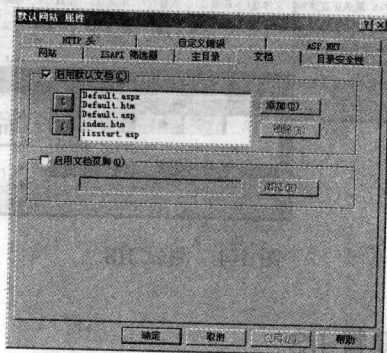


图 1-8 【文档】选项卡

4. 【目录安全性】选项卡

该选项卡用于设置身份控制，单击【匿名访问和身份验证控制】选项区域的【编辑】