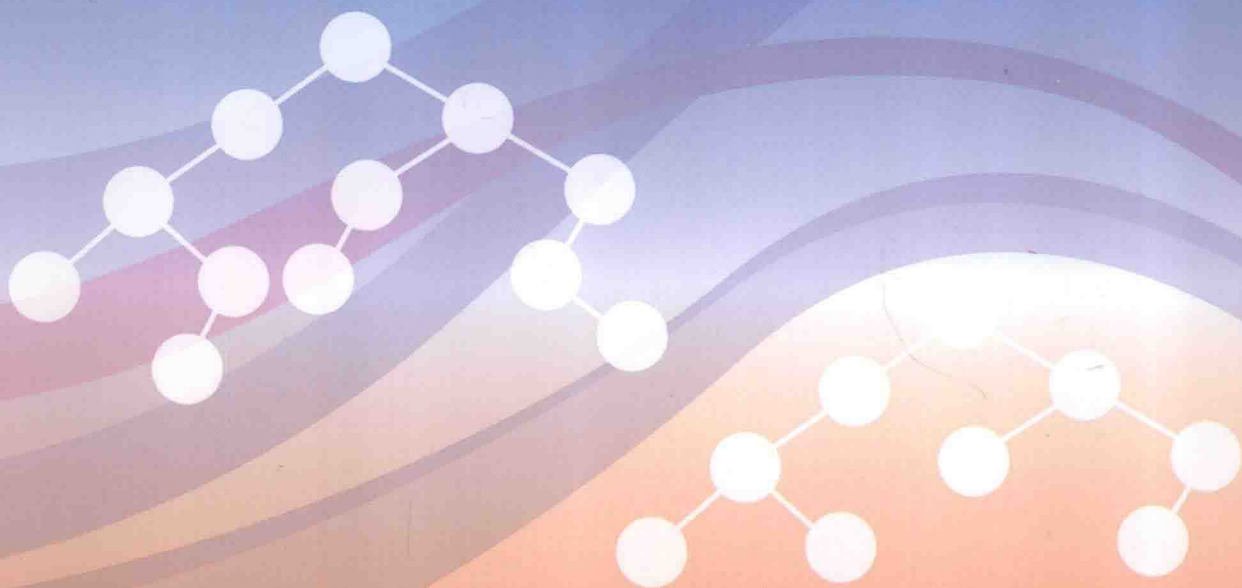


高等学校计算机专业规划教材

# ASP.NET Web 应用开发



马月坤 刘亚志 李志昕 编著

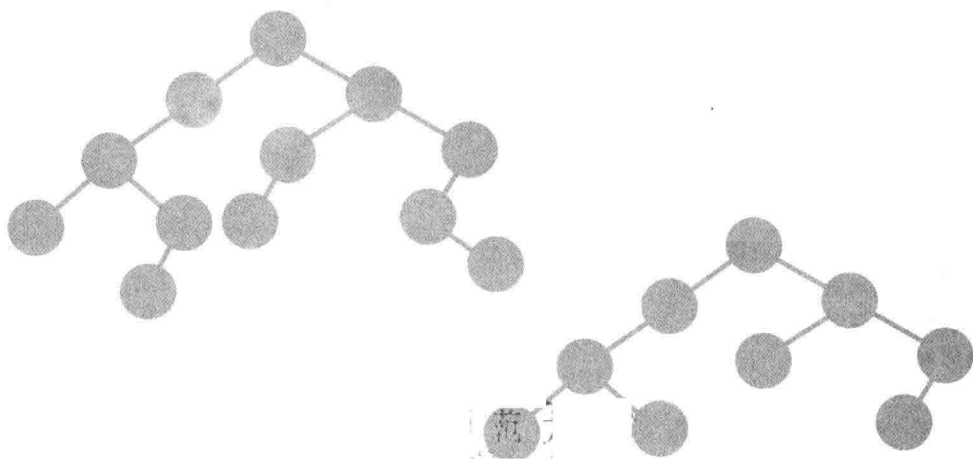
清华大学出版社



高等学校计算机专业规划教材

# ASP.NET Web 应用开发

马月坤 刘亚志 李志昕 编著



清华大学出版社  
北京

## 内 容 简 介

本书介绍利用 ASP.NET 进行 Web 程序开发需要掌握的知识。全书共分 15 章,内容包括 ASP.NET 简介、构建 ASP.NET 网站、ASP.NET 页面、ASP.NET 的服务器控件、数据验证控件、ASP.NET 内置对象、网站导航、站点设计、ADO.NET 与数据库操作、数据控件、网站安全、主题、Web 服务、ASP.NET Web 应用三层架构和一个包含核心设计的实际项目案例。书中所有知识都结合具体示例进行介绍,涉及的程序代码都给出了详细的注释,可以使读者轻松领会使用 ASP.NET 进行 Web 程序开发的精髓,从而快速提高开发技能。

本书可作为高等学校相关专业课程教材,也可供网站应用开发人员自学参考。

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 Web 应用开发/马月坤,刘亚志,李志昕编著. —北京:清华大学出版社,2016  
高等学校计算机专业规划教材  
ISBN 978-7-302-44564-7

I. ①A… II. ①马… ②刘… ③李… III. ①网页制作工具—程序设计—高等学校—教材  
IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2016)第 174864 号

责任编辑:龙启铭 战晓雷  
封面设计:何凤霞  
责任校对:时翠兰  
责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:21

字 数:487千字

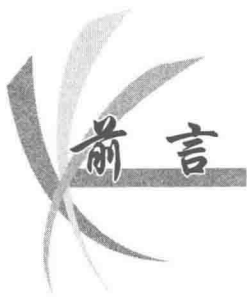
版 次:2016年8月第1版

印 次:2016年8月第1次印刷

印 数:1~2000

定 价:39.00元

产品编号:054807-01



ASP.NET 是微软公司推出的新一代建立动态 Web 应用程序的开发平台,它可以把程序开发人员的工作效率提高到用其他技术都无法比拟的程度。与 Java、PHP、Perl 等相比,ASP.NET 具有方便、灵活、性能优越、生产效率高、安全性高、完整性强及面向对象等特点,是目前主流的网络编程工具之一。

本书集 ASP.NET 教学和应用开发的经验,系统介绍了 ASP.NET 4.0 及其应用开发。知识系统,实践性强,各章都有应用实例,用以帮助读者掌握当前和此前的主要内容与知识,最后是一个大型的综合应用实例。

本书由华北理工大学马月坤、刘亚志、李志昕编著,参加本书编写的还有杨世超、李朋苗、游会迪、黄彪、段鹏飞、张博伟等。

由于作者水平有限,书中不足在所难免,欢迎广大读者批评指正!

作 者

2016 年 4 月



## 第 1 章 ASP.NET 简介 /1

1.1 .NET Framework 架构 .....	1
1.1.1 .NET Framework 简介 .....	1
1.1.2 .NET Framework 概况 .....	1
1.1.3 .NET Framework 核心技术特点 .....	2
1.1.4 ASP.NET 简介 .....	3
1.2 ASP.NET 的运行环境 .....	8
1.2.1 安装 IIS7 .....	8
1.2.2 配置 ASP.NET 运行环境 .....	10
1.3 ASP.NET 开发环境及开发工具 .....	14
1.3.1 Visual Studio 简介 .....	14
1.3.2 解决方案资源管理器 .....	14
1.3.3 控件工具箱 .....	14
1.3.4 服务器资源管理器 .....	15
1.3.5 错误列表与任务列表 .....	16
1.3.6 页面设计窗口与代码编辑窗口 .....	16
1.4 第三方框架 .....	17
本章小结 .....	20
习题 .....	20

## 第 2 章 构建 ASP.NET 网站 /21

2.1 创建 ASP.NET 网站 .....	21
2.1.1 创建网站 .....	21
2.1.2 设计页面 .....	23
2.1.3 网站项目文件夹设置 .....	25
2.1.4 配置文件 Web.config .....	26
2.1.5 网站运行调试和发布 .....	26
2.2 服务器控件 .....	28
2.2.1 服务器控件类型 .....	28
2.2.2 ASP.NET 服务器控件的类层次结构 .....	28



2.3 事件与事件处理程序.....	30
2.3.1 事件简介 .....	30
2.3.2 向页面添加事件以及页面调试 .....	32
本章小结 .....	35
习题 .....	35

### 第 3 章 ASP.NET 页面 /36

3.1 ASP.NET 页面构成 .....	36
3.1.1 Web 窗体页 .....	36
3.1.2 代码隐藏文件 .....	37
3.2 页面设计 .....	38
3.3 页面与程序代码 .....	40
3.3.1 单文件页模型代码 .....	40
3.3.2 代码隐藏页模型代码 .....	41
3.4 .aspx 文件与.html 文件 .....	42
本章小结 .....	47
习题 .....	47

### 第 4 章 服务器控件 /48

4.1 概述 .....	48
4.2 HTML 服务器控件 .....	48
4.2.1 HTML 服务器控件简介 .....	48
4.2.2 编程创建 HTML 控件 .....	50
4.2.3 处理服务器端事件 .....	51
4.3 Web 服务器控件 .....	54
4.3.1 基本 Web 服务器控件介绍 .....	54
4.3.2 应用 .....	55
4.4 控件属性设置 .....	56
4.4.1 Web 服务器控件属性 .....	56
4.4.2 Web 服务器控件属性设置 .....	57
4.5 标准控件 .....	58
4.5.1 文本类型控件 .....	58
4.5.2 按钮类型控件 .....	59
4.5.3 选择类型控件 .....	61
4.5.4 图形显示类型控件 .....	63
4.5.5 文本上传控件 .....	64
4.5.6 应用实例 .....	65
本章小结 .....	69



习题 .....	69
<b>第 5 章 数据验证控件 /70</b>	
5.1 数据验证方式 .....	70
5.1.1 服务器端数据验证 .....	70
5.1.2 客户端数据验证 .....	71
5.2 数据验证控件类型 .....	72
5.2.1 验证控件的分类 .....	72
5.2.2 CompareValidator 控件 .....	72
5.2.3 RequiredFieldValidator 控件 .....	75
5.2.4 RegularExpressionValidator 控件 .....	77
5.2.5 RangeValidator 控件 .....	80
5.3 ValidationSummary 控件 .....	83
5.3.1 ValidationSummary 控件简介 .....	83
5.3.2 ValidationSummary 控件应用实例 .....	83
5.4 CustomValidator 控件 .....	85
5.4.1 CustomValidator 控件简介 .....	85
5.4.2 CustomValidator 控件应用实例 .....	85
本章小结 .....	87
习题 .....	87
<b>第 6 章 ASP.NET 内置对象 /89</b>	
6.1 概述 .....	89
6.2 Application 对象 .....	90
6.2.1 Application 对象的作用 .....	90
6.2.2 Application 对象的属性 .....	90
6.2.3 Application 对象的方法 .....	91
6.2.4 Application 对象的常用事件 .....	92
6.3 Request 对象 .....	93
6.3.1 Request 对象的作用 .....	93
6.3.2 Request 对象的属性 .....	93
6.3.3 Request 对象的方法 .....	95
6.3.4 Request 对象的事件 .....	96
6.4 Response 对象 .....	97
6.4.1 Response 对象的作用 .....	97
6.4.2 Response 对象的属性 .....	97
6.4.3 Response 对象的常用方法 .....	98
6.4.4 Response 对象的事件 .....	99



6.5	Session 对象	100
6.5.1	Session 对象的作用	100
6.5.2	Session 对象的属性	101
6.5.3	Session 对象的方法	102
6.5.4	Session 对象的事件	102
6.6	Cookie 对象	104
6.6.1	Cookie 对象的作用	104
6.6.2	Cookie 对象的属性	104
6.6.3	Cookie 对象的方法	105
6.7	Server 对象	107
6.7.1	Server 对象的作用	108
6.7.2	Server 对象的属性	108
6.7.3	Server 对象的方法	109
	本章小结	110
	习题	110
<b>第 7 章 网站导航 /112</b>		
7.1	站点地图	112
7.1.1	站点地图简介	112
7.1.2	站点地图的应用	114
7.2	Menu 控件	115
7.2.1	Menu 控件简介	115
7.2.2	Menu 控件的应用	116
7.3	TreeView 控件	118
7.3.1	TreeView 控件简介	118
7.3.2	TreeView 控件的应用	120
7.4	SiteMapPath 控件	121
7.4.1	SiteMapPath 控件简介	121
7.4.2	SiteMapPath 控件的应用	122
	本章小结	124
	习题	124
<b>第 8 章 站点设计 /126</b>		
8.1	站点设计目标及原则	126
8.2	母版页面	127
8.2.1	创建母版页	127
8.2.2	母版页举例	129
8.2.3	在母版页中使用层叠样式表	129





8.2.4	多级母版页	132
8.2.5	母版页支持多个内容占位符	134
8.3	内容页	136
8.3.1	创建内容页	136
8.3.2	内容页举例	137
	本章小结	140
	习题	140
<b>第 9 章</b>	<b>ADO.NET 与数据库操作</b>	<b>/141</b>
9.1	SQL Server 数据库管理	141
9.1.1	使用 SQL Server 2008 创建数据库	141
9.1.2	Visual Studio 2010 管理数据库	144
9.2	ADO.NET	145
9.2.1	ADO.NET 简介	145
9.2.2	ADO.NET 的发展	145
9.2.3	ADO.NET 的作用	145
9.3	连接数据库	147
9.4	数据库检索	148
9.4.1	数据库检索对象	148
9.4.2	数据库访问和操作控件	153
9.5	数据更新	161
9.5.1	数据添加	161
9.5.2	数据修改	162
9.5.3	数据删除	163
	本章小结	164
	习题	164
<b>第 10 章</b>	<b>数据控件</b>	<b>/165</b>
10.1	概述	165
10.2	GridView 控件	165
10.2.1	GridView 控件属性	166
10.2.2	GridView 控件的方法和事件	168
10.2.3	GridView 控件绑定数据	169
10.2.4	GridView 控件定义列	172
10.2.5	使用模板列	174
10.2.6	数据分页和排序	175
10.3	DetailsView 控件	177
10.3.1	DetailsView 控件简介	177



10.3.2	DetailsView 控件字段 .....	179
10.3.3	DetailsView 控件常用属性 .....	179
10.3.4	DetailsView 常用事件 .....	184
10.4	FormView 控件 .....	185
10.5	Repeater 控件 .....	186
10.5.1	在 Repeater 控件中使用模板 .....	186
10.5.2	Repeater 控件的事件处理 .....	188
10.6	DataList 控件 .....	196
	本章小结 .....	199
	习题 .....	200

## 第 11 章 网站安全 /201

11.1	网站安全概述 .....	201
11.1.1	安全系统之间的关系 .....	201
11.1.2	模型威胁 .....	202
11.2	ASP.NET 安全配置 .....	203
11.2.1	ASP.NET 安全配置 .....	203
11.2.2	成员资格配置 .....	206
11.3	扩展 ASP.NET 安全框架 .....	207
11.3.1	成员资格管理 .....	207
11.3.2	登录控件 .....	209
11.3.3	使用角色管理授权 .....	211
11.3.4	授权 .....	213
11.3.5	身份验证 .....	214
	本章小结 .....	220
	习题 .....	220

## 第 12 章 用主题进行样式化 /222

12.1	主题 .....	222
12.1.1	不同类型的主题 .....	222
12.1.2	应用主题 .....	222
12.2	样式化站点 .....	223
12.2.1	Style 属性 .....	223
12.2.2	CSS 级联样式表 .....	224
12.3	对站点应用样式 .....	227
	本章小结 .....	228
	习题 .....	228

**第 13 章 Web 服务 /229**

13.1	Web 服务简介 .....	229
13.2	Web 服务的生命周期 .....	229
13.2.1	调用 Web 服务 .....	229
13.2.2	传送 Web 服务 .....	230
13.2.3	返回响应 .....	232
13.2.4	使用响应 .....	234
13.3	Web 服务的架构 .....	234
13.3.1	处理指令 .....	236
13.3.2	Web 服务范例 .....	236
13.4	Web 核心服务 .....	241
13.4.1	WSDL .....	241
13.4.2	SOAP .....	242
13.4.3	UDDI .....	242
	本章小结 .....	242
	习题 .....	243

**第 14 章 ASP.NET Web 应用三层架构 /244**

14.1	软件分层设计思想 .....	244
14.2	三层体系结构 .....	245
14.2.1	三层体系结构定义 .....	245
14.2.2	三层体系结构说明 .....	245
14.3	三层架构详解 .....	246
14.3.1	表示层 .....	246
14.3.2	业务逻辑层 .....	247
14.3.3	数据访问层 .....	247
14.4	三层架构范例 .....	248
14.4.1	实例应用 .....	248
14.4.2	实例详解 .....	254
	本章小结 .....	255
	习题 .....	255

**第 15 章 综合案例 /257**

15.1	高校教师教学发展中心网站 .....	257
15.1.1	系统规划 .....	257
15.1.2	数据库设计 .....	260
15.1.3	程序设计 .....	261



15.2 高校教师培训管理系统.....	270
15.2.1 系统规划.....	270
15.2.2 系统设计.....	272
15.2.3 程序设计.....	285
本章小结.....	323

**参考文献     /324**

### 本章要点

- 初步认识 ASP.NET。
- 从总体上对本书内容及 ASP.NET 4.0 所使用的工具做介绍。

### 学习目标

- 从总体上认识 ASP.NET。
- 学会配置开发环境以及使用 Visual Studio 2010。

## 1.1 .NET Framework 架构

ASP.NET 是 .NET Framework 的一部分,要学习 ASP.NET,首先从 .NET Framework 谈起。

### 1.1.1 .NET Framework 简介

.NET Framework 即 .NET 框架,是 .NET 应用程序开发和运行的必备条件,是微软公司为开发应用程序而创建的一个平台,其功能主要体现在统一的程序设计模式、多平台应用程序、多语言集成、对 Web 应用和服务的支持以及自动资源管理。微软公司首席执行官鲍尔默说,“.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。”在这个平台上,可以开发 Windows 桌面应用程序、Web 应用程序、Web 服务以及其他类型的应用程序。

.NET Framework 的设计方式确保它可以用于各种语言编程,包括 C#、C++、Visual Basic、JavaScript 等。为此,微软公司还推出了这些语言的 .NET 版本,所有这些语言都可以访问 .NET Framework,它们彼此之间还可以通信。C# 开发人员可以使用 Visual Basic 程序员编写的代码,反之亦然。

.NET Framework 主要包含一个非常大的代码库,可以在客户语言(如 C#)中通过面向对象编程技术来使用这些代码。这个库分为不同的模块,这样就可以根据希望得到的结果来选择使用其中的各个部分。使用 .NET Framework 编写应用程序,就是通过支持 .NET Framework 的任何一种语言使用 .NET 代码库来编写代码。

### 1.1.2 .NET Framework 概况

从 .NET Framework 出现到本书中介绍的 .NET Framework 4.0,其框架结构没什

么大的变化,都是公共语言运行库在操作系统上面,基类库在公共语言运行库上面,向上是 ADO.NET、XML 等,再向上是 ASP.NET、WinForms 等应用程序开发技术,最上层就是经常用到的各种 .NET 开发工具,每个较高的层都使用一个或多个较低的层。.NET Framework 框架结构如图 1-1 所示。

### 1. 应用程序开发技术

应用程序开发技术位于框架的最上方,程序员面向应用的软件开发主要基于该层开展工作,这样应用程序开发人员不需要具有开发操作系统级别的能力即可开发工作需要的应用程序。它包括 ASP.NET 技术和 WinForms 技术等高级编程技术。

### 2. .NET Framework 类库

.NET Framework 类库是一个综合性的类型集合,用于应用程序开发的一些支持性的通用功能。开发人员可以使用它开发多种模式的应用程序,可以是命令行形式的应用,也可以是图形界面形式的应用。.NET Framework 中主要包括以下类库:数据库访问类库(ADO.NET 等)、XML 支持、目录服务(LDAP 等)、正则表达式和消息支持。

### 3. 基类库

基类库提供了支持底层操作的一系列通用功能。.NET Framework 框架主要覆盖了集合操作、线程支持、代码生成、输入输出(I/O)、映射和安全等领域的内容。

### 4. 公共语言运行库(CLR)

公共语言运行库是 .NET Framework 的基础内容,也是 .NET 程序的运行环境,用于执行和管理任何一种针对 .NET 平台的所有代码。CLR 可以为应用程序提供很多核心服务,如内存管理、线程管理和远程处理等,并且还强制实施代码的安全性和可靠性管理。

## 1.1.3 .NET Framework 核心技术特点

.NET Framework 框架具有一些核心技术特点,通过对核心技术特点的了解,读者能够更深入地了解 C# 程序后台运行的原理。下面介绍 .NET Framework 的核心技术特点。

### 1. 多语言支持

在 .NET 平台上,所有的语言都是等价的,它们都是基于公共语言运行库(CLR)的运行环境进行编译运行。所有 .NET 支持的语言,不管是 Visual Basic、Visual C++、C# 还是 JavaScript,都是平等的。用这些语言编写的代码都被编译成一种中间代码,在公共语言运行库中运行。在技术上,这些语言与其他语言相比没有很大的区别,用户可以根据自己熟悉的编程语言进行操作。在本书中使用 C# 进行编程,因为 C# 是一种优秀的程序开发语言,它简洁、高效且便于使用。多语言支持主要用于 .NET 框架中面向组件的



图 1-1 .NET Framework 框架结构

领域。

## 2. 多平台支持

.NET 框架的另一个重要特点就是多平台支持。不过相对于 Java 技术能够跨越 UNIX、Linux 和 Windows 等众多平台,目前 .NET 的跨平台性仅限于各种 Windows 操作系统,如 Windows 95/98、Windows XP、Windows NT、Windows 2000 和 Windows 7 等。

## 3. 性能

.NET 的基本设计目标之一就是具有强大的性能和可伸缩性。

对于 .NET 来说,要具备很好的性能,需要靠公共语言运行库来执行中间代码。为了确保最佳性能,在某种意义上公共语言运行库将所有引用程序代码都编译成本机代码。这种转换既可以在应用程序运行时完成,也可以在应用程序首次安装时完成。

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给用不同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的是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。图 1-2 给出了各种程序与公共语言运行库(CLR)之间的关系。公共语言运行库改进了性能,能够轻松使用其他语言开发的组件。

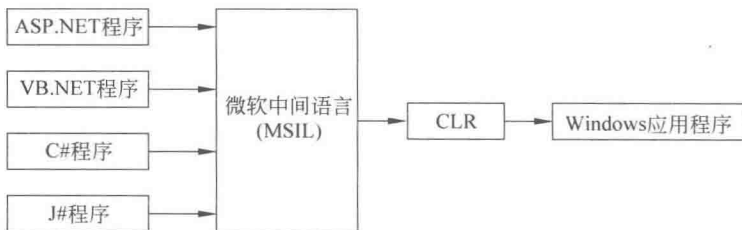


图 1-2 各种程序与 CLR 的关系图

### 1.1.4 ASP.NET 简介

ASP.NET 是 .NET Framework 的一部分,是一种嵌入网页中的、可由因特网服务器执行的服务器端脚本技术。所谓服务器端脚本技术是指当通过 HTTP 请求文档时才在 Web 服务器上动态创建网页,然后将动态生成的网页返回给客户端。

ASP(Active Server Pages)是动态服务器页面。静态页面是随着 HTML 代码生成的页面,其内容和显示效果基本不会改变,而动态页面的页面代码虽然不变,但其显示内容却随着时间、环境或者数据库操作的结果而发生改变。ASP 运行于 IIS(Internet Information Services, Internet 信息服务)之上。IIS 是 Windows 开发的 Web 服务器,1.2 节将介绍其安装过程。

ASP 技术作为 ASP.NET 的前身,是在 IIS 2.0(Windows NT 3.51)上与 ADO 1.0

一起推出的,在 IIS 3.0(Windows NT 4.0)上发扬光大,成为服务器端应用程序的热门开发工具,微软公司还特意为 ASP.NET 量身打造了 Visual Inter Development 开发工具。

1994—2000 年,ASP 技术已经成为微软公司推广 Windows NT 4.0 平台的关键技术之一,数以万计的 ASP 网站也是这个时候开始如雨后春笋般地出现在互联网上的。其简单以及高度可定制化的能力,是它能迅速崛起的原因之一。不过 ASP 的缺点也逐渐浮现出来,面向过程的程序开发方法让它维护的难度很大,尤其是对于大型 ASP 应用程序。解释型的 VBScript 或 JScript 语言让其性能无法完全发挥。其扩展性由于其基础架构的不足而受限,虽然有 COM 元件可用,但开发一些特殊功能(如文件上传)时,没有来自内置控件的支持,需要借助于第三方的控件。

1997 年,微软公司开始针对 ASP 的缺点(尤其是面向过程的开发思想)开展了一个新的项目。当时 ASP.NET 的主要领导人 Scott Guthrie 刚从杜克大学毕业,他和 IIS 团队的 Mark Anders 经理一起合作两个月,于 1997 年圣诞节开发出了下一代 ASP 技术的原型,并给予其一个名称——XSP,这个原型产品使用的是 Java 语言。不过它马上就被纳入当时还在开发中的 CLR 平台,Scott Guthrie 事后也认为,将这个技术移植到当时的 CLR 平台,确实有很大的风险,但当时的 XSP 团队却是以 CLR 开发应用的第一个团队。

为了将 XSP 移植到 CLR 中,XSP 团队将 XSP 的内核程序全部以 C# 语言进行了重构(在内部的项目代号是 Project Cool,但当时对公开场合是保密的),并且改名为 ASP+。而且为 ASP 开发人员提供了相应的迁移策略。ASP+ 的首个 Beta 版本以及应用在 PDC 2000 中亮相,由 Bill Gates 主讲 Keynote(关键技术的概览),由富士通公司展示使用 COBOL 语言编写的 ASP+ 应用程序,并且宣布它可以使用 Visual Basic .NET、C#、Perl、Nemerle 与 Python 语言来开发,后两者由 ActiveState 公司开发的互通工具支持。

2000 年第二季度,微软公司正式推动 .NET 策略,ASP+ 应用程序也顺理成章地改名为 ASP.NET,经过 4 年的开发,第一个版本的 .NET Framework 1.0 和 ASP.NET 在 2002 年 1 月 5 日亮相。

### 1. ASP.NET 的几个重要优点

(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。与被解释的早期版本不同,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行之前便显著提高了性能。

(2) 世界级的工具支持。ASP.NET Framework 补充了 Visual Studio 集成开发环境中的大量工具箱和设计器。WYSIWYG(所见即所得)编辑、拖放服务器控件和自动部署只是这个强大的工具所提供的功能中的少数几种。

(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并且具有灵活性。.NET Framework 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或跨多种语言分割应用程序。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 的开发中的现有投资。

(4) 简易性。ASP.NET 使执行常见任务——从简单的窗体提交和客户端身份验证到部署和站点配置变得容易。例如,ASP.NET 页框架使用户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,并在类似 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) 可管理性。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 应用程序的工作。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。此“零本地管理”哲学也扩展到了 ASP.NET Framework 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 Framework 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) 可缩放性和可用性。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) 自定义性和扩展性。ASP.NET 提供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,可以保证应用程序是安全的。

## 2. ASP.NET 的版本升级过程

ASP.NET 已经是非常成熟的一项 Web 开发技术,它到目前为止已经经历了多个版本,一直在不断更新,每一次版本的升级都是 .NET 开发技术的升级,了解 ASP.NET 的发展历程对于每个 ASP.NET 开发人员来说是非常有意义的。

### 1) ASP.NET 1.0 和 ASP.NET 1.1

2002 年,随着微软公司 .NET 口号的提出和 Windows XP、Office XP 的发布,微软公司发布了代号为 Rainier 的 Visual Studio .NET。它最大的改进就是使用 .NET 框架(版本 1.0),引入了受控代码开发环境,使用 .NET 开发的程序并不会像 C++ 那样被编译为机器语言,而是被编译成一种微软中间语言(MSIL)或者通用中间语言(IL)的格式。当一个 MSIL 应用程序被执行时,它会被及时编译成适用于所运行平台的机器语言代码,这样使得代码可跨平台运行。此时 ASP.NET 1.0 相较于 ASP 有了质的变化:

- 改变了传统 ASP 的开发模式,使用设计与代码分离的代码隐藏模型。
- 消除了对脚本引擎的依赖性,支持多语言开发,如 C#、Visual Basic 等。
- 提供了丰富的 Web 服务器控件和代码调试等工具,能够节约开发者的设计成本。
- 功能强大的身份确认模型。2003 年微软公司对 Visual Studio 2002 进行了部分修订,发布了代号为 Everett 的 Visual Studio 2003。它将 .NET 框架由 1.0 升级到 1.1 版,同时为使用 ASP.NET 或者 .NET Compact Framework 来开发移动设备程序提供了内置支持。

### 2) ASP.NET 2.0

到 2005 年,微软公司发布了 Visual Studio 2005,同时也将 ASP.NET 由 1.1 版本升级为 2.0 版本。ASP.NET 2.0 做了如下改进: