



普通高等教育“十一五”国家级规划教材
(高职高专教育)

ASP.NET技术基础

许锁坤 编



高等教育出版社
Higher Education Press

普通高等教育“十一五”国家级规划教材
(高职高专教育)

ASP.NET 技术基础

许锁坤 编

高等教育出版社

内容提要

本书是普通高等教育“十一五”国家级规划教材。

全书从开发人员的角度出发,以 ASP.NET Web 应用程序开发所需的关键技术为主线组织讲授内容,体现了基于应用的系统性。本书内容主要包括 ASP.NET 应用程序开发和运行环境的配置,用户界面设计技术,基于 C# 的 ASP.NET 应用程序编码技术,ADO.NET 数据库访问技术,ASP.NET 应用程序的管理与 Web 应用程序架构技术等。本书目的是使读者通过阅读本书,对基于 C# 语言的 ASP.NET 应用编程有一个清晰的思路,并能快速掌握 ASP.NET Web 应用程序开发的实用技术,达到独立完成中小型商业化 Web 应用程序的设计与开发目标。

本书所附光盘主要包括三部分内容:一是书中所有实例源码,可直接在讲授过程中演示运行;二是所有实训项目源码,学生上机时可参照运行结果实现实训要求;三是本书的一套辅助教学软件《计算机辅助教学系统》,其功能包括教学资源的学习、在线答疑、上传实训作业、批改作业、作业浏览和作业分数查询等。

本书可作为应用性、技能型人才培养的各类教育“ASP.NET 技术基础”课程的教学用书,也可供各类培训、计算机从业人员和程序设计爱好者参考使用。

图书在版编目(CIP)数据

ASP.NET 技术基础/许锁坤编. —北京:高等教育出版社,2007.6

ISBN 978-7-04-021228-0

I. A... II. 许... III. 主页制作-程序设计-高等学校教材 IV. TP393.092

中国版本图书馆 CIP 数据核字(2007)第 067054 号

策划编辑	刘怀恩	责任编辑	刘怀恩	封面设计	张志奇	责任绘图	宗小梅
版式设计	张 岚	责任校对	杨雪莲	责任印制	毛斯璐		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京未来科学技术研究所
有限责任公司印刷厂

开 本 787×1092 1/16
印 张 18.75
字 数 450 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2007 年 6 月第 1 版
印 次 2007 年 6 月第 1 次印刷
定 价 29.60 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 21228-00

前言

每逢去书店选书，我总是喜欢先读前言，而后选择感兴趣的章节，看是否有新意。现在不同种类的书多了，相同种类的书也多了，而并非人云亦云的书却相对少了。一本书传达了一种设计思想，一种分析方法，一个解决问题的完整过程，读前言有助于透析作者的用意。

本书写作初衷

有不少学软件开发的学生，走向工作岗位时总是抱怨虽然学的东西很多，但大部分都用不上，能用上的还不知道怎么用。作者常年从事企业应用软件的开发和教学工作，如何使学生能够较快地成为一个专业程序员或至少成为一个准程序员，是自己一直在思考的问题。以往，老师总是希望尽可能多地将知识传授给学生，结果适得其反。一个专业程序员具备的素质，在课堂上是造就不出来的。因此，与其在课堂上浪费大量的时间去传授暂时不用、不常应用或很难用到的内容，倒不如让学生在最短的时间内先学会如何去做，在做中体验技术应用的扩展与深化，这或许是从一个学生成长为一名专业程序员的有效学习途径。就像一个刚学走路的孩子，大人只要告诉孩子向前迈步，甚至亲自用手将腿向前拉出就行了。尽管大人没有告诉孩子迈腿时要身体重心前移，如何保持平衡等，孩子还是很快就会自己走路了，甚至跑了起来。早上手实践是最关键的，正是基于此种思考，才决定将自己的教学体会整理成书。本书内容的组织是从开发人员的角度出发，按照完成 Web 应用程序项目技能的需求，将其分解为不同的模块，而后，再组织实现每一模块必备的知识点而形成章节，整个教学内容构成一个基于应用的逻辑整体。如开发 Web 应用程序，首先是开发和运行环境配置，接下来是设计应用程序界面、业务逻辑编码、用户数据验证和处理、访问数据库、系统架构等。希望本书能引导初学者顺利走上程序员之路，对于有一定基础的读者能进一步地提升自己的技术，此即作者初衷。

内容导读

2000年6月，微软宣布了.NET战略，并使其逐渐变成了现实。微软.NET开发平台带来了全新、快速和敏捷的企业计算能力。作为平台重要组成部分的ASP.NET，在Web应用程序开发方面迅速得到了广泛应用。本书将分为十章内容来讲述ASP.NET开发技术及其应用。

第一章内容为ASP.NET开发技术引论，介绍ASP.NET Web应用程序开发和运行环境的配置，创建第一个ASP.NET应用程序项目。学习本章后，读者会体验到ASP.NET这一具备如此强大功能的开发技术，操作竟然是如此容易和便捷。

第二章内容为ASP.NET应用程序界面的构建，介绍使用Web控件设计应用程序界面技术。ASP.NET引入了Web窗体的概念，窗体界面元素称为Web控件，像Windows窗体编程一样，可将Web控件拖到窗体中进行可视化设计。

第三章内容为基于C#的ASP.NET应用程序编码技术，C#是编写.NET Framework应用程

序最合适的语言。并不是说只有掌握了 C# 才能学好 ASP.NET，应该是在 ASP.NET 中如何使用好 C#。本章讲述 C# 编码技术也正是基于这种思路。读者会在不知不觉中学会使用 C#。

第四章内容为 用户输入数据处理及页面参数传递。Web 页面是无状态的，页面间是不相干的，怎么能保证用户的输入是正确的？接受了用户输入的数据后，可能需要在另外的页面上进行处理或显示，页面间又是如何传值的？用户数据的处理、页面转向控制和页面间的参数传递，对于编写 Web 应用程序的程序员来说，均是必须熟练掌握的基本技术。阅读完本章，这些问题均可以得到解决。

第五章内容为 ADO.NET 数据库访问技术。在商业应用程序中，最重要的部分就是数据处理，无论是在线商务应用程序，还是公司企业软件，都需要一种简便快速、有效可靠的数据存储、检索和处理方式。本章介绍 Microsoft 提供的全新数据访问技术 ADO.NET，借助于此，前面几章所学的知识与技术在本书会得到更深入的应用。一些 Web 服务器控件将实现与数据库中数据的关联；C# 编码将深入到与数据访问相关的业务逻辑。

第六章内容为 数据列表控件的应用技术。在 ASP.NET 中，专门有一类用于数据输出或操作的控件，这类控件特别适用于在表格中显示数据或用于复杂用户界面的数据输出。例如购物车、产品展示及论坛等。它们就是本章将学习的 DataGrid、DataList 和 Repeater 数据列表控件。

第七章内容为 Web 服务技术。未来人们开发的基于互联网的应用将成为一种服务，服务供应商在自己的服务器端提供一些方法，这些方法可以在互联网上被其他客户在任何平台上直接调用，而且不需要考虑这些方法是用哪种语言编写的。这样，便可实现在不同系统平台上开发出的应用程序之间的相互通信。本章介绍的 Web 服务技术就是使应用系统跨平台，跨编程语言的技术。

第八章内容为 ASP.NET 应用程序的管理与部署，介绍 Web 应用程序的稳定性及安全防护方面的相关技术。发布在互联网上的应用程序，可能会遇到一些新的问题。例如，可能会面临各种类型的访问者和巨大访问量，应用程序是否可以稳定地运行，是否可以抵御非法攻击，这是发布 Web 应用程序必须考虑的问题。

第九章内容为 软件开发架构，主要介绍 Web 应用程序的分层架构设计。该章内容对于提高团队应用程序开发效率，构建健壮的商业化应用程序是不可缺少的。希望读者仔细研读本章提供的实例，理解设计方法。

第十章内容为 ASP.NET 开发技术综合应用——新闻发布系统，介绍面对一个实际的软件项目，需要做些什么，如何着手去做？使读者尽快达到实际应用的水平。

如何学习

本书的章节组织和内容的导入是在系统思考的基础上设计的，为了让读者更好地学习本书，特提供以下学习建议：

1. 本书前六章内容是前后呼应的，初学者要按顺序阅读，学完前六章后可以按任意顺序学习后面的章节。有一定基础的读者可任意选择感兴趣的章节学习。

2. 书中提供了大量源于工程实践的例子，包括实训部分，读者要亲自上机完成。这些例子大多不需修改或稍加修改即可用于实际软件项目中。

3. 在实现某一个功能时，多考虑可能的解决途径，比较哪个是最优的，从而在学习过程

中逐步提高自己的实际应用能力。

4. 在学习的过程中, 学生可以利用本书所附光盘中的实例源码、实训项目源码进行练习, 另外, 利用光盘中的《计算机辅助教学系统》辅助课堂教学。

真诚致谢

首先要感谢大红鹰职业技术学院院长俞瑞钊教授, 作者在本书中贯穿的应用型高等教育理念, 主要得益于俞瑞钊教授的指导。

感谢我的学生们, 他们经常提出一些无拘无束的问题, 他们表现出来的轻松和喜悦是我写作本书的最大动力。此外, 感谢孙振业老师、李材老师在百忙之中对全书进行了认真的审阅, 并提出了宝贵的意见。

由于作者水平所限, 书中难免有纰漏之处, 恳请批评指正。

编 者

2007 年 4 月

目 录

第一章 ASP.NET 开发技术引论1	
1.1 Web 应用程序.....1	
1.1.1 什么是 Web 应用程序.....2	
1.1.2 Web 应用程序 开发技术.....2	
1.2 Microsoft .NET 开发平台.....4	
1.2.1 Microsoft.NET 开发 平台简介.....4	
1.2.2 什么是 ASP.NET.....6	
1.2.3 ASP.NET 主要技术.....6	
1.3 ASP.NET 开发与运行 环境配置.....7	
1.3.1 安装软件.....7	
1.3.2 安装与配置 IIS.....8	
1.4 初识 ASP.NET 应用程序.....10	
1.4.1 第一个 ASP.NET 应用 程序.....10	
1.4.2 代码编程方式.....15	
1.4.3 ASP.NET 应用程序 文件.....16	
本章小结.....17	
思考题.....17	
实训 ASP.NET 应用程序运行 环境配置与测试.....17	
第二章 ASP.NET 应用程序界面 的构建19	
2.1 ASP.NET 编程模型.....19	
2.1.1 窗体用户界面.....20	
2.1.2 事件驱动编程.....22	
2.1.3 窗体生命周期.....24	
2.2 窗体界面元素——Web 服务器控件.....25	
2.2.1 基本输入输出控件.....26	
2.2.2 按钮类控件.....27	
2.2.3 页面导航控件.....29	
2.2.4 选择类控件.....30	
2.2.5 容器和占位控件.....34	
2.2.6 广告和日历控件.....35	
2.3 第三方控件.....39	
2.3.1 菜单设计控件.....40	
2.3.2 文本编辑控件.....42	
2.4 用户控件.....44	
2.4.1 设计用户控件.....44	
2.4.2 在程序界面中添加 用户控件.....46	
2.5 程序界面设计原则.....47	
本章小结.....49	
思考题.....49	
实训 2.1 Web 服务器控件的 取值和属性设置.....49	
实训 2.2 使用 TreeView 控件 设计树型菜单.....50	
实训 2.3 使用 FreeTextBox 控 件设计文本编辑界面.....51	
第三章 基于 C# 的 ASP.NET 应用程序 编码技术53	
3.1 C# 语法基础.....53	
3.1.1 C# 语句标识.....55	
3.1.2 数据类型.....55	
3.1.3 命名变量.....57	
3.1.4 操作符.....57	
3.1.5 基本数据类型的转换.....58	
3.2 字符串处理.....60	
3.2.1 字符串操作.....60	



3.2.2 检索字符串中的 特定字符	61
3.3 程序流程控制	62
3.3.1 程序分支控制及其应用	62
3.3.2 程序循环控制 及其应用	65
3.4 数组和数组列表	67
3.4.1 数组 (Array)	68
3.4.2 数组列表 (ArrayList)	69
3.5 编写和使用类	70
3.5.1 项目内添加类库文件	71
3.5.2 创建新的类库项目	72
3.5.3 使用属性	74
3.6 异常处理	77
本章小结	78
思考题	78
实训 3.1 字符串处理	79
实训 3.2 C#程序流程控制	80
实训 3.3 编写和使用类	81
第四章 用户数据处理及页面 参数传递	82
4.1 用户数据验证——编码 方法	82
4.1.1 编写验证方法	83
4.1.2 用户注册信息验证	85
4.2 用户数据验证——控件 方法	88
4.2.1 数据验证控件	89
4.2.2 验证控件的综合应用	92
4.3 常用的 ASP.NET 内部 对象	94
4.3.1 Response 对象	94
4.3.2 Request 对象	96
4.3.3 Session 对象	98
4.3.4 Application 对象	99
4.3.5 Server 对象	100
4.4 页面导航与参数传递	101

4.4.1 页面导航方法	101
4.4.2 页面间参数传递	102
4.5 在 ASP.NET 中使用 客户端脚本	106
4.5.1 直接使用客户端脚本	106
4.5.2 调用 JavaScript 函数	107
本章小结	109
思考题	109
实训 4.1 用户数据验 证——C#	109
实训 4.2 用户数据验 证——控件	110
实训 4.3 页面间参数传递	111
第五章 ADO.NET 数据库访问技术	113
5.1 ADO.NET 概述	113
5.1.1 ADO.NET 组成	113
5.1.2 数据访问机制	115
5.2 连接数据库	116
5.2.1 Connection 对象	116
5.2.2 不同数据库的连接 方式	117
5.2.3 数据库的连接状态	121
5.3 连接式数据库操作	122
5.3.1 Command 对象	122
5.3.2 使用 Command 对象 操作数据库	123
5.3.3 DataReader 对象	128
5.4 断开式数据库操作	134
5.4.1 DataSet 和 DataAdapter 对象	134
5.4.2 DataGrid 数据绑定	135
5.4.3 选择控件数据绑定	138
5.4.4 使用 DataSet 对象 操作数据库	140
5.5 XML 数据访问	146
5.5.1 XML 文档结构	146
5.5.2 DataSet 访问 XML 文件	148

5.6 事务处理.....	150	第七章 Web 服务技术.....	188
本章小结.....	152	7.1 Web 服务技术基础.....	188
思考题.....	153	7.1.1 什么是 Web 服务.....	189
实训 5.1 连接数据库.....	153	7.1.2 Web 服务技术基础.....	189
实训 5.2 Command 对象操作 数据库.....	154	7.2 创建和使用 Web 服务.....	192
实训 5.3 DataReader 对象的 使用.....	155	7.2.1 创建和测试 Web 服务.....	192
实训 5.4 DataSet 对象操作 数据库.....	156	7.2.2 调用 Web 服务.....	195
实训 5.5 事务处理.....	157	7.3 Web 服务中的数据类型.....	198
第六章 数据列表控件的应用技术	159	7.3.1 数据类型处理.....	198
6.1 DataGrid 控件.....	159	7.3.2 使用 DataSet 传递 数据.....	199
6.1.1 定制 DataGrid 控件.....	159	7.4 SOAP 的使用.....	200
6.1.2 DataGrid 分页技术.....	165	7.4.1 SOAP 消息结构.....	200
6.1.3 在 DataGrid 中操作 数据库.....	169	7.4.2 使用 SOAP 传输 信息.....	201
6.1.4 在 DataGrid 中使用 模板列.....	172	7.5 ASP.NET Web 服务应用 实例.....	203
6.2 DataList 控件.....	174	本章小结.....	207
6.2.1 DataList 控件使用 模板.....	174	思考题.....	207
6.2.2 在 DataList 控件中 显示数据.....	174	实训 7.1 创建第一个 Web 服务.....	207
6.3 Repeater 控件.....	180	实训 7.2 数据库 Web 服务.....	208
6.3.1 Repeater 控件模板.....	180	第八章 ASP.NET 应用程序的 管理与部署	210
6.3.2 在 Repeater 控件中 显示数据.....	180	8.1 状态管理.....	210
本章小结.....	183	8.1.1 服务器端和客户端的 状态管理.....	211
思考题.....	183	8.1.2 进程内和进程外 状态管理.....	215
实训 6.1 DataGrid 中使用 模板.....	183	8.1.3 Global.asax 文件.....	216
实训 6.2 DataGrid 操作 数据库.....	184	8.2 缓存优化.....	218
实训 6.3 DataList 显示 数据.....	185	8.2.1 页面级输出缓存.....	219
实训 6.4 Repeater 显示 数据.....	186	8.2.2 片段缓存.....	219
		8.2.3 使用 Cache 对象.....	220
		8.3 配置 ASP.NET 应用 程序.....	222
		8.3.1 Machine.config 文件.....	222



8.3.2 Web.config 文件	223	9.3.2 创建业务逻辑组件	248
8.4 部署 ASP.NET 应用		9.3.3 创建表现层	250
程序	228	9.4 三层应用开发实例——	
8.4.1 使用复制部署	228	用户管理	252
8.4.2 使用 Web 安装项目	228	9.4.1 需求分析与设计	252
8.5 Web 应用程序安全		9.4.2 编码实现	254
策略	230	本章小结	266
8.5.1 攻击类型	231	思考题	266
8.5.2 常见漏洞分析	231	实训 用户管理模块的设计	
8.5.3 安全威胁模型		与实现（三层）	266
STRIDE	232	第十章 ASP.NET 开发技术综合	
8.5.4 安全应用实例	233	应用——新闻发布系统	268
本章小结	237	10.1 系统需求分析	268
思考题	237	10.1.1 系统功能描述	268
实训 8.1 数据加密处理	238	10.1.2 系统功能模块	
实训 8.2 使用 Web 安装		的划分	269
项目	238	10.2 新闻系统的设计	270
第九章 软件开发架构	240	10.2.1 数据库设计	270
9.1 企业应用开发模式	240	10.2.2 系统结构设计	272
9.1.1 客户机/服务器模式	241	10.2.3 界面流程设计	272
9.1.2 浏览器/服务器模式	242	10.3 系统功能的实现	273
9.2 Web 应用程序的分层		10.3.1 新闻浏览	273
架构	244	10.3.2 新闻类别管理	275
9.2.1 Web 应用程序的		10.3.3 新闻管理	277
分层	244	10.3.4 用户管理	280
9.2.2 三层体系架构	245	10.3.5 权限管理	281
9.3 使用 ASP.NET 开发三层		本章小结	286
应用程序	245	思考题	286
9.3.1 创建数据库访问		实训 ASP.NET 开发技术	
组件	245	综合应用实践	286

1

第一章 ASP.NET 开发技术引论

在一个开放标准的世界中，为基于互联网的应用提供最好的开发平台和工具是微软的战略目标之一。为此，微软在打造 .NET 开发平台上投入了大量的人力物力，包括产品开发、技术开发以及市场营销等方面。近年来，微软战略引起了业界的极大关注，作为 .NET 开发平台的重要组成部分，ASP.NET Web 应用开发技术迅速得到了广泛应用。在接受这些新的技术、新的设计思想以及新的编程方法时，读者或许会想，我是否适合学习这门技术？我能否学会这门技术？通过本章的学习读者就会明白，ASP.NET 这一具备如此强大功能的开发技术，操作竟然是如此的容易和便捷，这或许就是微软声称的简单而令人神往的用户体验。



【学习目标】

了解 Microsoft .NET 战略
Microsoft .NET 开发平台的构成

理解 Web 应用程序的概念
Web 应用程序目前采用的主流开发技术
什么是 ASP.NET
ASP.NET 应用程序的基本结构

学会 ASP.NET 开发与运行环境的配置
ASP.NET 开发工具的使用
创建 ASP.NET 应用程序项目



【关键技术】

配置 ASP.NET 运行环境
创建和运行 ASP.NET 应用程序

1.1 Web 应用程序

Tim Berners-Lee 在 1991 年发明的万维网 (World Wide Web) 给全球信息的交流和传播带来了革命性的变化，打开了人们利用 Web 来获取信息的方便之门。随着网络经济的到来，有效运用网络技术来构建高效能、稳定的 Web 应用系统，以提升信息利用及拓展商机，已成为现代企业追求的目标。同时，互联网已经从各个方面逐渐改变着人们的工作和生活方式，比如，人们可以足不出户，在家里进行办公、网上购物、收发电子邮件，享受远程医疗和远程教育服务等



等。当今,网络技术正以人们始料不及的惊人速度向前发展,基于 Web 的应用开发已成为一个极具诱惑力的学习和研究领域,所有人在此领域都梦想着将来会发生的任何事情。

1.1.1 什么是 Web 应用程序

如果我们需要从网上获取信息,至少涉及两个方面,一方请求和显示信息,另一方提供信息。借助浏览器请求和显示信息的一方称为客户端;通过 Web 服务器提供信息的一方称为服务端。相应的硬件为客户机和服务器,实际上服务器就是一种高性能计算机。

Web 服务器是指驻留在服务器上的计算机程序。当客户机连接到服务器上,通过 Web 浏览器请求信息服务时,Web 服务器会处理该请求,并将信息发送给浏览器。常见的 Web 服务器包括 Microsoft Internet Information Server(IIS)、Microsoft Personal Web Server、Apache HTTP Server、Netscape Enterprise Server 和 iPlanet Web Server。

在 Web 应用的初期,人们全部使用 HTML 语言设计网页。这些简单的 HTML 文档在被手工编辑完成之后保存在互联网中的计算机内,任何连入互联网的用户都可以访问其中的信息,这种形式的页面被称为静态页面。所有的静态页面都是事先编写好的,是不变的。因此,访问同一页面的用户看到的都将是相同的内容,静态页面无法根据不同的用户、不同的需求提供不同的页面内容。随着网上信息量的增加,以及企业或个人希望通过网络发布产品和信息的需求的增强,人们越来越需要一种能够通过简单的 Web 页面访问服务端后台数据库的方式以满足不同的需求,这是静态页面所远远不能实现的。

为了克服静态页面的种种不足,人们将传统单机环境下的编程技术引入到了互联网,并与 Web 技术相结合,从而形成了 Web 编程技术。Web 编程技术通过在传统的静态页面中加入各种程序和逻辑控制,在网络中的客户端和服务端之间实现了动态和个性化的交流与互动。人们将这种使用 Web 编程技术创建的页面称为动态页面。

什么是 Web 应用程序呢?Web 应用程序是一种通过互联网能够让 Web 浏览器和服务器通信的计算机程序。Web 服务器不仅能够存储信息,还能在用户通过 Web 浏览器提供的信息的基础上运行脚本和程序。例如,假设用户要求免费提供公司资讯,只要建立一张表单,填写需求并提交,该表单将送至服务器上的某一程序,它负责处理该请求并返回结果。也可以让该程序把客户提供的信息保存于数据库中。再如,一个在线电子商城,为了购买货物,人们会进行反复的观察和选择,浏览一系列程序页面,收集所需要的信息,满意后支付相应的费用,最后发出定单。这一切,均可轻松地通过 Web 应用程序来实现。

1.1.2 Web 应用程序开发技术

目前,Web 应用系统已广泛结合了商业、数据库以及企业的运用,因此,人们对 Web 应用系统的要求也越来越严格。Web 应用系统必须具备高度的扩展性,合理的执行效率,以及安全稳定的执行环境。也就是说,现在的 Web 应用系统必须能够安全及时地服务于大量的客户端用户。要做到这些,就要求 Web 应用系统具有一种良好的体系结构,以及实现此结构所采用的先进的开发技术。图 1-1 是 Web 应用体系的基本架构。客户端借助浏览器请求和显示信息,服务端通过 Web 服务器提供信息。Web 应用程序开发技术包括客户端技术和服务端技术。

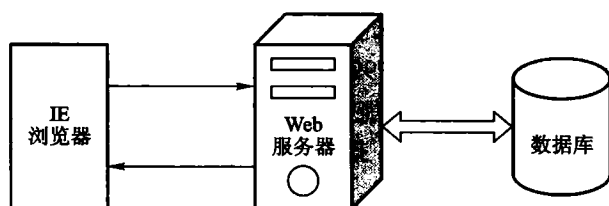


图 1-1 Web 应用基本架构

(1) 客户端技术

Web 客户端的主要任务是展现信息内容，HTML 语言是显示信息的基础语言。HTML 语言本身只能在浏览器中展现静态的文本或图像信息，显然满足不了人们对信息的丰富性和多样性的强烈需求，这就迫使客户端技术由静态向动态方向演进。1997 年，Microsoft 发布了 IE 4.0，并将动态 HTML 标记、CSS 和动态对象模型（DHTML Object Model）发展成了一套完整、实用、高效的客户端开发技术体系，Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果，DHTML 技术无需启动 Java 虚拟机或其他脚本环境，可以在浏览器的支持下，获得更好的展现效果和更高的执行效率。

(2) 服务端技术

与客户端技术从静态向动态的演进过程类似，Web 服务端的开发技术也是由静态向动态逐渐发展、完善起来的。最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器，这种方式无法支持后台数据库。因此，可以满足更复杂应用的服务端开发技术相继出现。

- CGI (Common Gateway Interface) 技术。第一种真正使服务器能根据运行时的具体情况，动态生成 HTML 页面的技术是 CGI 技术。CGI 技术允许服务端的应用程序根据客户端的请求，动态生成 HTML 页面，这使客户端和服务端的动态信息交换成为了可能。随着 CGI 技术的普及，聊天室、论坛、电子商务、信息查询、全文检索等各式各样的 Web 应用蓬勃兴起，人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。其编程语言可以是 C、C++、Pascal、Perl 等任何通用的程序设计语言。

- PHP (Personal Home Page Tools) 技术。1994 年，Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP 语言。与以往的 CGI 程序不同，PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面，Web 应用的开发者可以用一种更加简便、快捷的方式实现动态 Web 功能。

- ASP 技术。1996 年，Microsoft 在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 使用的脚本语言是我们熟悉的 VBScript 和 JavaScript。借助 Microsoft Visual Studio 等开发工具在市场上的成功，当时，ASP 迅速成为 Windows 系统下 Web 服务端的主流开发技术。

- JSP 技术。1997 年，Servlet 技术（Servlet 是用 Java 编写的服务端程序，用于处理客户请求）问世，随即诞生了 JSP 技术。Servlet 和 JSP 的组合，让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能。此外，Java 的编译技术也大大提高了 Servlet 和 JSP 的执行效率。Servlet 和 JSP 已成为 J2EE 平台的核心技术。

- 两种重要的企业开发平台。Web 服务端开发技术的不断完善，使得开发复杂的 Web 应用成为可能。为了适应企业级应用开发的各种复杂需求，给最终用户提供更可靠、更完善的信息服务，两个最重要的企业级开发平台 J2EE 和 .NET 在 2000 年前后分别诞生，J2EE 和 .NET 开

发平台的发布,使 Web 开发技术以前所未有的速度向前发展。

J2EE 是纯粹基于 Java 的解决方案。1998 年, Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中的数据封装、事务处理、交易控制等功能提供了良好的技术基础。J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。目前, J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。

Microsoft 的 .NET 平台是一个强调多语言间交互的通用开发及运行环境。2001 年, 欧洲计算机制造联合会 (ECMA) 通过了 Microsoft 提交的 C# 语言和通用语言架构 (CLI) 标准, 这两个技术标准构成了 .NET 平台的基石, 它们也于 2003 年成为 ISO 的国际标准。2002 年, Microsoft 正式发布 .NET Framework 和 Visual Studio .NET 开发环境。 .NET 平台及相关的开发环境, 不但为 Web 服务端应用提供了一个支持多种语言的、通用的运行平台, 而且还引入了 ASP.NET 这样一种全新的 Web 开发技术。ASP.NET 超越了以往服务端技术的局限, 可以使用 VB.NET、C# 等编译型语言, 支持许多高级特性。为广大致力于 Web 应用开发的程序员带来了全新的 Web 开发技术和轻松愉快的开发体验。

1.2 Microsoft .NET 开发平台

尽管随着 Web 服务端开发技术的不断完善, 使得开发复杂的 Web 应用成为可能, 新的企业需求不断被提出并逐步被实现, 但由于 Web 应用系统受限于其低效能及高成本, 企业应用仍多局限在小型或局部的系统, 并未得到真正全面广泛地运用。鉴于此, 微软公司不惜投入大量人力物力, 研发新一代的网络应用架构。2002 年, Microsoft 正式发布 .NET Framework 和 Visual Studio.NET 集成开发环境。

1.2.1 Microsoft.NET 开发平台简介

Microsoft .NET 开发平台包括一个用于加载和运行应用程序的新的软件基础结构 (.NET Framework 和 ASP.NET), 一个新的开发环境 (Visual Studio .NET), 以及支持该结构的编程语言。如图 1-2 所示。

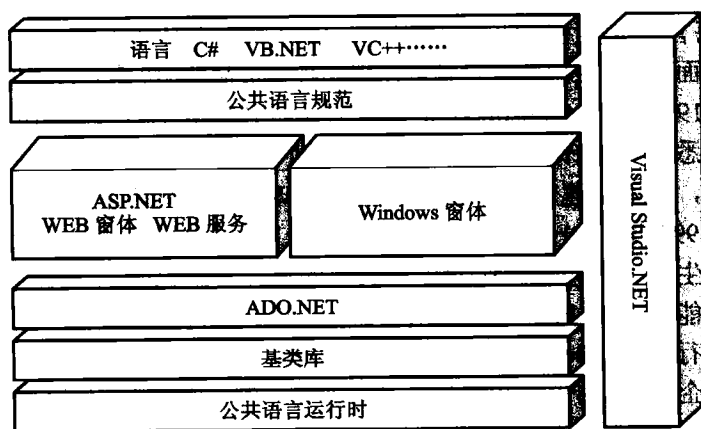


图 1-2 Microsoft .NET 开发平台

从上图我们可以简要地了解.NET 开发框架的几个主要组成部分。首先是整个开发框架的基础,即通用语言运行时以及它所提供的一组基础类库;在开发技术方面,.NET 提供了全新的数据库访问技术 ADO .NET、网络应用开发技术 ASP .NET 和 Windows 编程技术 Win Forms;在开发语言方面,.NET 提供了 VB、VC++、C#、Jscript 等多种语言支持;Visual Studio.NET 则是全面支持.NET 的开发工具。

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

Microsoft .NET 给开发人员带来了一种全新的开发框架,公共语言运行时处于这个框架的最低层,是这个框架的基础,为多种语言提供了一种统一的运行环境。CLR 是一种受控的执行环境,其功能需要通过编译器与其他工具共同展现。基于公共语言运行时开发的代码称为受控代码,其运行步骤大致为:使用框架中的一种语言编写的源代码,首先被编译成独立于机器的微软中间语言 MSIL,而后由 CLR 使用即时编译器 JIT 将代码编译成真正的机器码。凡是符合公共语言规范的程序语言(如 C#和 Visual Basic.NET 等)所开发的程序都可以在任何有 CLR 的操作系统上执行。

(2) 基础类库 (Base Class Library)

基础类库提供了一个统一的面向对象的、层次化的、可扩展的编程接口,包括了从输入、输出到数据访问等各方面。在使用时,开发者只需在自己的应用中添加所需的基础类库的引用,即可使用这个类库中的所有方法、属性。

(3) ADO.NET

在 Microsoft.NET 中,使用 ADO.NET 访问数据库,其提供了一组连接数据库、执行命令、返回数据集等类库,是数据库访问的通用接口。它不仅提供了对 XML (Extensible Markup Language) 的强大支持,而且引入了 DataSet 的概念。DataSet 是一个驻留于内存的数据缓冲区,它提供了数据的关系型视图。数据库中的数据填充到 DataSet 后,可以对 DataSet 中的数据进行操作,从而实现了数据库的断开操作方式。

(4) ASP.NET

ASP.NET 是建立在公共语言运行库 (CLR) 上的应用程序框架,是一项功能强大的、非常灵活的服务器端技术,ASP.NET 采用了全新的编程思想用于在服务器端构建高效、安全的 Web 应用程序。

(5) Windows 应用

在 Microsoft .NET 中,开发传统的基于 Windows 的应用程序时,除了可以利用现有的技术,如 ActiveX 控件以及丰富的 Windows 接口外,还可以基于通用语言运行时开发,如 ADO.NET、网络服务等。

(6) 公共语言规范

公共语言规范是.NET 框架的所有语言的基础,定义了公共类型系统的标准和所有语言必须实现的功能。这是第三方语言与框架集成的基础。

(7) 开发语言

Visual Studio.NET 自身包含四种语言: Visual Basic、Visual C++、Visual C# 和 JScript。在.NET 平台下,除了语法上的区别外,各种编程语言没有本质的不同。它们共享公共类库,具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。.NET

将彻底改变软件的开发方式、使用方式和发行方式,这一切改变会使得我们构建一个复杂的商业化的 Web 应用系统时变得更加容易。

总之,Microsoft.NET 开发平台,在通用语言运行时的基础上,给开发者提供了完善的基础类库,提供了新一代的数据库访问技术 ADO.NET 和网络开发技术 ASP.NET,开发者可以使用多种语言及 Visual Studio.NET 来快速构建 Web 应用程序。

1.2.2 什么是 ASP.NET

ASP.NET 是建立在通用语言运行时 (CLR) 上的应用程序框架,是一种 Web 应用开发技术,使用 ASP.NET 可以在服务器端构建功能强大的 Web 应用程序。自从 Microsoft 推出 .NET 开发平台以来,ASP.NET 在 Web 应用程序开发方面迅速得到了广泛地应用。ASP.NET 与微软以往的服务端开发技术相比较,其优越性主要表现在以下几个方面。

- 更高的开发效率。2002 年 2 月,微软发布了 Visual Studio.NET (VS.NET) 集成开发环境,这是微软极力推荐的创建 .NET 应用程序的软件开发工具集。VS.NET 让开发者体会到了微软在开发语言上进行的革命性的改变,同时也解决了许多过去困扰开发者使用微软工具创建应用程序的问题。随着它在调试方面的改进,以及它将开发者所需的工具紧密集成到了一个单一的环境上,这些都使得 VS.NET 大大提高了 Web 应用程序的开发效率。

- 更强的应用功能。ASP.NET 不同于已往的 ASP,ASP 只能使用脚本语言 JavaScript 和 VBScript (尽管可以使用由 Windows 系统支持的任何一种脚本语言)。脚本语言就像是编程语言的简明版本或低等级版本,不能实现编程语言的全部功能。ASP.NET 可以使用有很大选择余地的、功能完善的编程语言,可以充分利用有巨大潜力的 .NET Framework。ASP.NET 可以帮助用户利用由 .NET Framework 支持的任何一种编程语言创建更快、功能更强大的 Web 应用程序。

- 更快的执行速度。ASP.NET 比传统 ASP 的运行速度快,ASP 是解释执行的,ASP.NET 是编译执行的。ASP.NET 可以自动检测变化,根据需要动态编译文件,并且保存编译结果,以便后续请求的再次使用。动态编译保证了应用程序总是最新的,编译后执行的操作会使其运行速度更快。

- 更容易的配置管理。ASP.NET 提供了一个分层配置结构,该结构使其能够在整个应用程序、站点或计算机中定义和使用可扩展的配置数据。利用文本格式的配置文件,可以随意更改服务器配置,而不需要进入 IIS 的管理页。

- 更高的安全性。ASP.NET 应用程序借助内置的 Windows 身份验证和基于每个应用程序的配置,保证了 ASP.NET 应用程序的安全性。

1.2.3 ASP.NET 主要技术

ASP.NET 和以前的 Web 应用程序开发技术相比,从本质上是完全不同的,它几乎全是基于组件和模块化,每一个页、对象和 HTML 元素,都是一个运行的组件对象。下面是 ASP.NET 包含的主要技术。

(1) Web 窗体技术

ASP.NET 为用户提供了功能强大、外观丰富的基于表单 (Form) 的可编程 Web 页面。ASP.NET 带有大量的控件,这些控件大幅度减少了开发者不得不编写的代码量,开发者只需把

控件定位在一个页面中，初始化控件的属性和方法，然后编写代码来响应控件可能产生的事件即可，这种事件是用户某些行为（如单击按钮）的结果。这些控件还支持数据绑定，这使开发者很容易从数据库取回特定信息，并与指定控制值关联起来。Web 窗体技术使得程序员在 Web 应用中采用向往已久的 Windows 开发模式成为可能。

（2）Web 服务技术

Web 服务是一种新型的 Web 应用程序，未来人们开发的基于互联网的应用将成为一种服务。服务供应商在自己的服务器端提供一些方法，这些方法可以在互联网上被其他用户在任何平台上直接调用，而且不需要考虑这些方法是用哪种语言编写的。这样，就可以实现在不同系统平台上开发出来的应用程序之间的相互通信。Web 服务技术就是使应用系统跨平台、跨编程语言的技术。

（3）代码分离技术

在 ASP.NET 中，不再像以往的动态页面那样，将脚本代码与 HTML 代码混在一个文件中，它可将界面设计与程序编码进行分离，分别放在两个文件中，使 ASP.NET 页面也呈现出一种事件驱动的特性。这样处理可大大提高代码的重用性。ASP.NET 的代码分离技术不但提高了开发效率，也提高了运行效率，ASP.NET 不再是解释执行，而是在用户第一次访问一个新的页面时，对新页面的代码进行编译，以后的用户访问执行的是编译后的代码，这就使运行效率有了很大的提高。此外，代码分离技术提高了程序部署的安全性。

（4）数据绑定技术

数据绑定是运用编程方式指定控件的数据源，以便使控件动态显示数据的方法。这种方法不仅允许开发人员可以绑定到数据表的视图，而且可以绑定到简单属性、集合、表达式甚至是从方法调用返回的结果。数据绑定技术在使用数据控件的应用程序开发中提高了开发效率。数据绑定不仅仅是快速应用程序开发的一个重要方法，它还有助于将数据结构的细节对应用程序的表示层隐藏起来。

1.3 ASP.NET 开发与运行环境配置

目前，从计算机硬件来说，无论是计算速度还是磁盘空间，运行 ASP.NET 都是没有问题的，需要重点考虑的是软件环境的配置。现在流行使用的 Windows 操作系统如 Windows 2000、Windows XP、Windows 2003 Server 等均支持 ASP.NET 运行，只是在配置上有一定区别。

1.3.1 安装软件

除了操作系统之外，软件环境主要是指 ASP.NET 基础的运行环境 .NET Framework 和开发工具 VS.NET。

（1）安装 .NET Framework

.NET Framework 包含了用于开发的基本类库，它是 Microsoft 已编写好的代码。程序员可以将它们包括到任一程序中，这样可以节省大量编码时间。此外，.NET Framework 也是 ASP.NET 的基础运行环境。

如果操作系统选择 Windows 2000 Server，则推荐安装 Service Pack 3 或更新版本、Internet