



21 世纪高职高专规划教材

ASP.NET 技术与应用



主 编 杨 帆 赵义霞

副主编 赵立臻 胡 珏 索 剑



高等教育出版社

图书在版编目 (CIP) 数据

ASP.NET 技术与应用/杨帆, 赵义霞主编. —北京:
高等教育出版社, 2004.1
ISBN 7-04-013369-5

I .A... II .①杨...②赵... 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 (2003) 第 110838 号

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 010-82028899

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 中国农业出版社印刷厂

开 本 787×1092 1/16
印 张 18.5
字 数 450 000

版 次 2004 年 1 月第 1 版
印 次 2004 年 8 月第 2 次印刷
定 价 24.00 元

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

内 容 提 要

ASP.NET 技术是 Microsoft 推出的新一代动态 Web 技术开发工具,是今后电子商务、电子政务、远程医疗、远程网络教育等 Web 应用开发的主流技术之一。本书从应用开发的角度,介绍了 Microsoft.NET Framework 的基本结构、ASP.NET 的页面语法结构、Web 控件、HTML 控件、页面验证技术、数据库访问、Web 服务程序等基本内容,增加了“控件使用策略”、“数据控件比较”、“如何用正则表达式定义模式”等读者非常感兴趣的内容。同时还介绍了 ASP.NET 应用程序的开发工具(VB.NET)、开发过程和部署方法。通过 VB.NET 开发工具,突出了 ASP.NET 技术的“代码分离”思想,降低了 ASP.NET 应用开发和页面代码的阅读难度。为了方便查阅,在本书的最后以附录形式给出了相关内容的快捷参考。

本书内容丰富,结构安排合理,描述通俗易懂,清晰反映了 Web 应用开发与部署的完整思想,突出了简单、快捷、高效的开发与应用理念。每章配有“本章导读”、“本章小结”和“习题”,便于学习者了解各章内容的知识点以及知识点之间的联系。为了便于上机操作,本书将一般安排在后面的“调试与跟踪”内容特意调整到了第 2 章,便于理论教学和实践教学的计划安排。

本书可作为高职高专类计算机及相关专业教材,也可作为 Web 开发和从事电子商务等工作人员的技术参考书。

本书所配教学电子教案及书中程序源代码均可从高等教育出版社网站下载,网址为:

<http://cs.hep.com.cn> 或 <http://www.hep.com.cn>。

前 言

随着 Internet 技术的发展,Web 应用(如电子商务、电子政务、远程教育、远程医疗等)已经成为网络应用需求的主流,在此背景下 Microsoft 推出了 Microsoft.NET 平台,开创了软件开发的新时代。作为 Microsoft.NET 平台的主要部分,ASP.NET 首次在 Web 页面开发方面引入了控件技术,全面支持面向对象的设计思想,提供了一个功能强大的 Web 应用开发模式,使 Web 应用的桌面设计变得更加丰富、直观、简单、高效。不仅如此,ASP.NET 还具有更高的运行效率。因此,ASP.NET 将成为未来 Web 开发的主流技术。

在 VS.NET 环境下开发 ASP.NET 应用程序,最大的特点就是它在很大程度上提高了开发人员的工作效率。页面设计者只需编写最少量的代码便能够给 Web 应用程序提供更为丰富的处理功能。最为突出的特点是设计者可以选择任何一种可用的 .NET 编程语言来开发 Web 应用。为了简单、快捷和大众化的目的,本书选择 VB.NET 作为编程语言。

在数据处理方面,ASP.NET 提供了 ADO.NET 模块,专门支持两种不同数据源(SQLClient 和 OleDb)的高效连接与数据操作能力。本书选择 SQL Server 2000 数据库系统,所有举例均使用其中的 pubs 等数据库。关于具体的表结构等内容,读者可自行查阅对应的数据表。安装和配置数据库服务环境的内容不在本教材之列,请读者参考有关资料。

本书旨在以最通俗的语言、最简单的操作、最快捷的设计举例,使读者能够更深刻地了解 ASP.NET 技术,全面掌握 ASP.NET 技术的使用,以便能够以最高的效率设计出功能强大的 Web 应用。全书的基本内容如下:

第 1 章介绍 ASP.NET 的基本概念、新特性、运行模式、开发与运行环境等内容,还重点介绍 VS.NET 集成开发环境的建立和使用,使读者具备边学边练的条件。

第 2 章围绕 ASP.NET 应用开发,介绍 ASP.NET 页面的不同形式、ASP.NET 语法、ASP.NET 应用开发过程以及跟踪与调试方法等内容。使读者能以不同的方式认识 ASP.NET 页面,同时掌握了 ASP.NET 应用开发的全过程。

第 3 章在详细分类的基础上介绍各类控件的功能、语法格式、使用方法等内容,重要控件还给出了详细的示例。同时针对相同或相似功能的 Web 服务器控件和 HTML 服务器控件,特意增加了“控件使用策略”一节,以便读者选择合适的控件,提高 Web 页面的运行效率。

第 4 章重点介绍 ASP.NET 中的验证控件及验证技术,增加的“用正则表达式定义模式”一节使读者能真正掌握模式验证技术。

第 5 章是 ASP.NET 应用开发的核心内容,主要介绍 ADO.NET 访问数据库的基本思想和过程,详细介绍了 DataGrid、DataList、Repeater 数据控件的使用以及数据库的基本操作,给出了访问 XML 数据的基本方法。整个内容突出了 DataSet 的“断开式”数据访问模型。

第 6 章介绍 ASP.NET 内建对象的基本知识和使用方法。

第 7 章重点介绍 ASP.NET 对 Web Services 的支持,包括 Web 服务程序的特性、Web 服务程序的运行模式、如何使用 Web 服务程序以及创建 Web 服务程序的基本方法。

第8章主要介绍部署 Web 应用程序所要考虑的相关问题、配置 web.config 文件、配置 global.asax 文件、安全访问控制以及部署 Web 应用程序的基本方法和步骤。

本书第1章由杨帆编写,第2章、第4章、第5章由赵义霞编写,第3章和附录内容由赵立臻编写,第6章由索剑编写,第7章、第8章由胡珏编写。赵义霞和赵立臻还参与了全书编写大纲的讨论起草工作,全书由杨帆统稿。

在本书编写的过程中,得到了惠州学院和潍坊学院有关领导和部门的大力支持和帮助,李忠、王祥修、李勇军、赵秋云、卢丹、王嫦娥等同志在整理资料和调试范例程序方面做了大量工作,在此表示衷心感谢。

由于作者水平所限,编写时间仓促,书中可能存在某些错误,请读者不吝指正。

编 者

2003.10

郑 重 声 明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》,其行为人将承担相应的民事责任和行政责任,构成犯罪的,将被依法追究刑事责任。为了维护市场秩序,保护读者的合法权益,避免读者误用盗版书造成不良后果,我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为,希望及时举报,本社将奖励举报有功人员。

反盗版举报电话: (010) 58581897/58581896/58581879

传 真: (010) 82086060

E - mail: dd@hep.com.cn

通信地址: 北京市西城区德外大街 4 号

高等教育出版社打击盗版办公室

邮 编: 100011

购书请拨打电话: (010)64014089 64054601 64054588

目 录

第 1 章 ASP.NET 介绍	(1)	2.3 ASP.NET 应用程序的跟踪	(44)
1.1 .NET Framework 介绍	(1)	2.3.1 ASP.NET 中的跟踪服务	(44)
1.1.1 .NET Framework 的设计目标	(1)	2.3.2 跟踪日志	(44)
1.1.2 .NET Framework 的基本组成	(2)	2.3.3 使用页面级跟踪	(46)
1.2 ASP.NET 简介	(3)	2.3.4 使用应用程序级跟踪	(46)
1.2.1 Web 页面	(3)	2.4 ASP.NET 应用程序的调试	(48)
1.2.2 Web 服务	(4)	2.4.1 应用程序调试	(48)
1.2.3 ASP.NET 的新特性	(4)	2.4.2 调试模式	(49)
1.2.4 ASP.NET 的运行模式	(6)	2.4.3 程序调试方法	(49)
1.2.5 ASP.NET 的文件类型	(6)	2.4.4 独立调试工具——CLR 调试器	(50)
1.3 ASP.NET 运行环境	(7)	本章小结	(51)
1.3.1 运行环境要求	(7)	习题二	(52)
1.3.2 安装 .NET Framework	(9)	第 3 章 ASP.NET 控件	(53)
1.3.3 安装与配置 IIS	(9)	3.1 ASP.NET 控件介绍	(53)
1.3.4 安装与配置数据库服务	(11)	3.1.1 ASP.NET 控件	(53)
1.4 ASP.NET 开发环境	(12)	3.1.2 ASP.NET 控件分类	(54)
1.4.1 ASP.NET 开发环境概述	(12)	3.1.3 服务器控件的基本形式	(56)
1.4.2 ASP.NET 开发工具介绍	(13)	3.1.4 服务器控件的基本属性	(57)
1.4.3 Visual Studio.NET 的安装	(15)	3.2 Web 服务器控件	(58)
1.5 VS.NET 集成开发环境	(16)	3.2.1 基本 Web 控件	(58)
1.5.1 配置文件	(16)	3.2.2 增强控件	(64)
1.5.2 起始页	(17)	3.3 HTML 服务器控件	(69)
1.5.3 创建新项目	(17)	3.3.1 HTML 控件概述	(69)
1.5.4 集成开发环境	(18)	3.3.2 使用 HTML 控件	(70)
本章小结	(22)	3.4 扩展控件	(87)
习题一	(22)	3.4.1 TreeView 控件	(88)
第 2 章 ASP.NET 应用程序的开发	(23)	3.4.2 ToolBar 控件	(90)
2.1 ASP.NET 页面	(23)	3.4.3 TabStrip 和 MultiPage 控件	(94)
2.1.1 页面的不同形式	(23)	3.5 用户自定义控件	(97)
2.1.2 PAGE 类	(27)	3.6 控件使用策略	(99)
2.1.3 页面处理过程	(29)	3.6.1 控件比较	(99)
2.1.4 ASP.NET 语法	(30)	3.6.2 选择控件	(100)
2.2 用 VB.NET 开发 ASP.NET 应用	(37)	本章小结	(100)
2.2.1 需求说明	(37)	习题三	(101)
2.2.2 ASP.NET 应用的开发过程	(37)	第 4 章 Web 页面的验证技术	(102)
2.2.3 快捷方法	(43)	4.1 Web 页面验证	(102)

4.1.1 验证控件与验证类型	(102)	5.2.2 使用 Command 对象	(137)
4.1.2 验证时机	(103)	5.2.3 使用 DataReader 对象	(138)
4.1.3 验证多个条件	(103)	5.2.4 DataSet 和 DataAdapter 对象	(140)
4.1.4 显示错误信息	(104)	5.3 数据列表控件	(144)
4.1.5 服务器端验证和 客户端验证	(104)	5.3.1 DataGrid 控件	(144)
4.1.6 验证对象模型	(104)	5.3.2 Repeater 控件	(155)
4.1.7 自定义验证	(104)	5.3.3 DataList 控件	(161)
4.1.8 通过编程进行验证	(105)	5.3.4 数据控件的比较	(166)
4.2 显示验证信息	(105)	5.4 数据库的基本操作	(167)
4.2.1 显示方式	(105)	5.4.1 基于 SQL 的数据库操作	(167)
4.2.2 ValidationSummary 控件	(106)	5.4.2 ASP.NET 中的数据库操作	(167)
4.2.3 设置显示格式	(107)	5.4.3 数据库操作的基本过程	(168)
4.3 常规验证	(108)	5.5 XML 数据访问	(172)
4.3.1 必须验证	(108)	5.5.1 ASP.NET 对 XML 的支持	(172)
4.3.2 比较验证	(109)	5.5.2 访问 XML 数据	(173)
4.3.3 范围验证	(111)	本章小结	(177)
4.4 自定义验证	(112)	习题五	(177)
4.4.1 CustomValidator 控件	(112)	第 6 章 ASP.NET 内建对象	(178)
4.4.2 服务器端验证	(113)	6.1 概述	(178)
4.4.3 客户端验证	(114)	6.2 Request 对象	(179)
4.5 正则表达式与模式验证	(114)	6.2.1 Request 语法	(179)
4.5.1 正则表达式	(114)	6.2.2 Request 常用属性	(180)
4.5.2 用正则表达式定义模式	(116)	6.2.3 Request 常用方法	(180)
4.5.3 RegularExpressionValidator 控件	(117)	6.2.4 Request 应用举例	(181)
4.5.4 实现模式验证	(117)	6.3 Response 对象	(184)
4.6 编程验证	(120)	6.3.1 Response 语法	(184)
4.6.1 编程验证介绍	(120)	6.3.2 Response 常用属性	(184)
4.6.2 编程验证的方法	(121)	6.3.3 Response 常用方法	(185)
4.7 验证禁用	(123)	6.3.4 Response 应用举例	(186)
4.8 综合验证举例	(123)	6.4 Cookie	(188)
4.8.1 实例说明	(123)	6.4.1 Cookie 的描述	(188)
4.8.2 界面设计	(124)	6.4.2 Cookie 的传送流程	(188)
4.8.3 程序清单	(124)	6.4.3 编写 Cookie	(189)
本章小结	(128)	6.4.4 Cookie 的常用属性	(191)
习题四	(128)	6.4.5 Cookie 的限制	(192)
第 5 章 Web 页面的数据库技术	(129)	6.5 Server 对象	(192)
5.1 ADO.NET 简介	(129)	6.5.1 Server 常用属性	(192)
5.1.1 ADO.NET 概述	(129)	6.5.2 Server 常用方法	(192)
5.1.2 两种不同的数据访问机制	(131)	6.5.3 Server 应用举例	(193)
5.2 通过 ADO.NET 访问数据库	(132)	6.6 Application 对象	(194)
5.2.1 使用 Connection 对象	(133)	6.6.1 Application 语法	(194)
		6.6.2 Application 常用属性	(195)

6.6.3 Application 常用方法	(196)	8.1.3 Web 应用程序的部署过程	(219)
6.6.4 Application 事件	(196)	8.2 建立 Web 站点	(220)
6.6.5 Application 应用举例	(197)	8.2.1 添加 Web 站点	(220)
6.7 Session 对象	(198)	8.2.2 Web 站点属性设置	(223)
6.7.1 Session 常用属性	(198)	8.2.3 复制应用项目	(227)
6.7.2 Session 常用方法	(199)	8.2.4 举例:部署一个 Web 应用程序	(229)
6.7.3 Session 事件	(200)	8.3 web.config 文件	(229)
6.7.4 应用举例——观察 Session 对象的 生命周期	(200)	8.3.1 ASP.NET 配置简介	(229)
6.8 综合应用举例	(201)	8.3.2 配置文件的工作机制	(230)
本章小结	(203)	8.3.3 配置文件的语法规则	(230)
习题六	(203)	8.3.4 ASP.NET 定义的标准配置段	(232)
第 7 章 Web 服务程序	(204)	8.3.5 配置文件的访问	(233)
7.1 Web 服务程序简介	(204)	8.3.6 web.config 配置实例	(233)
7.1.1 什么是 Web 服务程序	(204)	8.4 global.asax 文件	(234)
7.1.2 Web 服务程序的特性	(205)	8.4.1 global.asax 简介	(234)
7.1.3 Web 服务程序的运行模式	(206)	8.4.2 应用程序事件	(235)
7.1.4 UDDI 简介	(207)	8.4.3 会话事件	(236)
7.2 使用 Web 服务程序	(208)	8.5 安全访问控制	(236)
7.2.1 寻找已有的 Web 服务程序	(208)	8.5.1 验证和授权	(236)
7.2.2 从浏览器调用 Web 服务程序	(209)	8.5.2 基于 Windows 的验证	(238)
7.2.3 使用代理调用 Web 服务程序	(211)	8.5.3 基于页面(form)的验证	(240)
7.3 创建 Web 服务程序	(213)	8.5.4 授权用户和角色	(244)
7.3.1 Web 服务程序的代码结构	(213)	本章小结	(245)
7.3.2 用 VB 创建一个简单的 Web 服务 程序	(213)	习题八	(245)
7.3.3 使用 VS.NET 创建并调用 Web 服务 程序	(214)	附录	(247)
本章小结	(217)	附录 I VB.NET 语言快捷参考	(247)
习题七	(217)	I-1 数据类型	(247)
第 8 章 部署 Web 应用程序	(218)	I-2 运算符	(248)
8.1 概述	(218)	I-3 函数与语句	(250)
8.1.1 Web 应用程序的部署	(218)	附录 II Web Matrix 的安装与使用	(265)
8.1.2 应用程序的安全	(219)	附录 III 在 VS.NET 集成环境中使用 DataSet 与 DataAdapter	(272)
		参考文献	(284)

第1章

ASP.NET 介绍

本章导读

ASP.NET 是 .NET Framework 平台上开发动态 Web 应用的一项新技术,全面支持面向对象的设计思想,因而使 Web 应用的桌面设计变得更加简单、直观、高效,使所建立的 Web 应用功能更为强大,特别是对标准化 Web 服务的支持能力。不仅如此,ASP.NET 还具有很高的运行效率。为了便于初步认识 ASP.NET,本章将从 .NET Framework 平台入手,介绍 ASP.NET 的基本思想、内容、最新特性、运行模式等;同时,从实用的角度介绍 ASP.NET 的运行环境、开发环境以及建立这些环境的基本过程。

1.1 .NET Framework 介绍

Microsoft .NET Framework 是 Microsoft 新一代视窗服务 (Next Generation Windows Service, NGWS) 体系的核心技术。它将引导未来软件开发策略,特别是基于 Web 的应用开发朝着自然、简单、高效、灵活的方向发展。所以,在介绍 ASP.NET 之前,有必要首先了解一下它。

1.1.1 .NET Framework 的设计目标

.NET Framework 是一种新的计算平台,它简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 旨在实现下列目标:

- (1) 提供一个一致的、面向对象的编程环境,而无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行。
- (2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- (3) 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行的代码执行环境。
- (4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- (5) 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。

(6) 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与任何其他代码集成。

1.1.2 .NET Framework 的基本组成

.NET Framework 具有两个主要组件:公共语言运行库和基本类库。具体结构如图 1-1 所示。

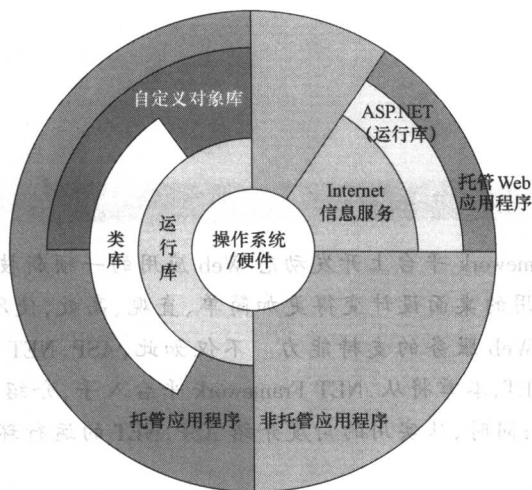


图 1-1 .NET Framework 结构示意图

1. 公共语言运行库

公共语言运行库(Common Language Run-time, CLR)是 .NET Framework 的基础,它包含 Windows 和 ASP.NET 两部分应用的支持。用户可以将运行库看做一个在执行时管理代码的代理,它提供核心服务(如内存管理、线程管理和远程处理),而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。

公共语言运行库实际上在组件的运行过程和开发时的操作中都起作用。在开发时,它的作用稍有变化。由于做了大量的自动处理工作(如内存管理),公共语言运行库使开发人员的操作非常简单,同时还大大减少了开发人员为将业务逻辑转变为可重用组件而必须编写的代码量。

尽管公共语言运行库是为未来的软件设计的,但是它也支持现在和以前的软件。托管和非托管代码之间的互操作性使开发人员能够继续使用所需的 COM 组件和 DLL。

公共语言运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务,但是它从不解释托管代码。一种称为实时(JIT)编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时,内存管理器排除了出现零碎内存的可能性,并增大了内存引用区域,以进一步提高性能。

2. 基本类库

.NET Framework 的另一个主要组件是基本类库,它是一个综合性的、面向对象的、面向重用的类型集合,它向开发人员提供了统一的、面向对象的、分层的和可扩展的类库集(Application

Programming Interface, API)。用户可以使用它开发包含从传统的命令行或图形用户界面 (GUI) 应用程序到基于 ASP.NET 所提供的最新创新的应用程序 (如 Web 窗体和 XML Web Services) 在内的应用程序。

在此之前, C++ 开发人员使用 Microsoft 基础类, 而 Java 开发人员使用 Windows 基础类。 .NET Framework 统一了这些完全不同的模型, 并且为 Visual Basic 和 JScript 程序员同样提供了对类库的访问。通过创建跨所有编程语言的公共 API 集, 公共语言运行库使得跨语言继承、错误处理和调试成为可能。从 JScript 到 C++ 的所有编程语言具有对 .NET Framework 的相似访问, 开发人员可以自由选择他们要使用的语言。

和一般面向对象的类库一样, .NET Framework 类也能够完成一系列常见编程任务 (包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务)。除这些常见任务之外, 类库还包括支持多种专用开发方案的类。例如, 可使用 .NET Framework 开发下列类型的应用程序和服务:

- (1) 控制台应用程序;
- (2) 脚本应用程序;
- (3) Windows GUI 应用程序 (Windows 窗体);
- (4) ASP.NET 应用程序;
- (5) XML Web Services;
- (6) Windows 服务。

例如, Windows 窗体类是一组综合性的、可重用的类型, 它们大大简化了 Windows GUI 的开发。如果要编写 ASP.NET Web 窗体应用程序, 可使用 Web 窗体类。

从上面的介绍可以看出, 公共语言运行库和基本类库是 .NET Framework 的核心内容。

1.2 ASP.NET 简介

ASP.NET 是 Microsoft 公司推出的新一代体系结构——Microsoft .NET Framework 的一部分, 它建立在基本类库之上, 用来在服务器端构建功能强大的 Web 应用, 包括 Web 页面 (Web Form) 和 Web 服务 (Web Services) 两部分。图 1-2 用另一种方式给出了 Microsoft .NET Framework 的结构, 突出 ASP.NET 在其中的地位和作用。

1.2.1 Web 页面

Web 页面 (Web Form) 是 ASP.NET 中的基本内容, 主要提供浏览界面设计及其交互功能的代码编程能力。与其他 Web 页面相比 (如单纯的超文本链接标示语言 (HTML) 页面、动态服务器 (ASP) 页面等), ASP.NET 的 Web 页面全面支持面向对象的设计思想, 即页面内容布局可通过可视化对象来设计, 其交互功能可通过捕捉这些对象的事件和调用其方法来完成。正是由于这一变化, 大大提高了 ASP.NET Web 页面的处理能力, 降低了基于 Web 数据库的动态 Web 页面的设计

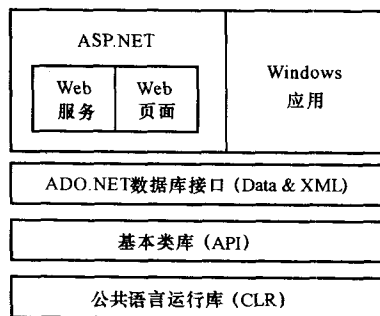


图 1-2 .NET Framework 模块示意图

与代码编程的技术要求。

由于有 CLR 的强大支持,所以,运行于 .NET 平台上的任何程序设计语言,均可用做 ASP.NET Web 页面的编程工具。出于技术兼容性和应用产品的连续性考虑,ASP.NET 和 ASP 页面可以在同一个 Web 站点使用。现存的 ASP 页面可以和新开发的 ASP.NET 页面(在 ASP.NET 运行环境下)正常运行。所不同的是,ASP 页面文件使用 .asp 扩展名,而 ASP.NET 页面文件则使用 .aspx 扩展名。

1.2.2 Web 服务

Web 服务(Web Services)是 ASP.NET 中提供的扩展 Web 应用能力的一个主要内容,目的是有效调用分布于 Internet 上的资源和计算能力,建立自然的、通用的、能适应不同网络设备的人-机交互界面,使人们享受到的 Internet 服务(与目前的 Internet 服务相比)发生质的变化。因此,Web Services 技术也将成为今后 Web 开发的主流技术。

从理论上说,Web Services 是一系列标准和正在发展中的标准的集合,这些标准由 World Wide Web Consortium(W3C)设计和制定,用来促进跨平台的程序与程序间的通信能力。目前,W3C 已经制定了一个模板和一个过程调用协议作为开发 Web 服务的基本标准,前者用来描述 Web 服务,因而也称为 Web 服务描述语言(Web Services Description Language, WSDL);后者用来协调 Web 服务的调用,因而被称为简单对象访问协议(Simple Object Access Protocol, SOAP)。从外部的使用者的角度看,Web 服务是一种部署在 Web 上的对象或组件,提供特定的服务功能,并具备完好的封装性、规范性、松散耦合、高度可集成等特性。为叙述方便,本书第 7 章将 Web Services 翻译成 Web 服务程序。

ASP.NET 中的 Web Services 支持能力是建立在 .NET 平台基础上的,所以,它能够使用 .NET 平台的各种特性,并采用上面所说的标准协议来描述和传输 Web Services,如 XML、WSDL、SOAP、HTTP-GET、HTTP-POST 等,从而保证了 ASP.NET 中构建的 Web 服务与其他平台上的 Web 服务相互沟通。

1.2.3 ASP.NET 的新特性

需要特别强调的是,虽然 ASP.NET 对先前的 ASP 应用提供了兼容支持,但这并不说明 ASP.NET 就是在 ASP 原技术架构下的版本升级产品。事实上,由于有了一个全新的 .NET 支持平台,ASP.NET 也在技术上做了重新设计,因而 ASP.NET 与 ASP 在技术和能力方面有着本质的区别,表现出了许多前所未有的新特性。

1. 多语言支持

多语言支持是 ASP.NET 的重要新特性之一,主要表现在所支持的编程语言种类多和单个语言功能强两个方面。首先,ASP.NET 为 Web 应用提供一种类似于 Java 编译技术的“二次编译技术”——中间语言 MSIL(Microsoft Intermediate Language)执行架构,先将 ASP.NET 应用编译成 MSIL,再将 MSIL 编译成机器语言执行。这样,只要能被编译成 MSIL 的编程语言都可以用来编写 ASP.NET 应用。目前,可以使用二十多种编程语言来创建 .NET 应用,Microsoft 公司拥有 Visual Basic、Microsoft Visual C#、Microsoft Visual C++ 以及 Microsoft Jscript 等语言的编译程序,第三方提

供者正在编写 Cobol、Pascal、Perl、Smalltalk 以及其他语言的 .NET 编译器。其次,ASP.NET 所支持的编程语言是指这种语言的功能全集(而不是子集),所以,ASP.NET 中每种编程语言的功能要比 ASP 中使用的 VB Script 和 Java Script 更为强大。

2. 增强的性能

在 ASP.NET 中,页面代码是被编译执行的,它利用提前绑定、即时编译、本地优化和缓存服务来提高性能。当第一次请求一个页面时,CLR 对页面程序代码和页面自身进行编译,并在高速缓存 Cache 中保存编译结果的副本。当第二次请求该页面时,就直接使用 Cache 中的结果(无需再次编译)。这就大大提高了页面的处理性能。

3. 类和名空间

ASP.NET 包含一整套有用的类和名空间(Namespace)。名空间被用做一种有组织的机制——一种表示可用于其他程序和应用的程序组件的方法。名空间包含类。和类库一样,名空间可以使 Web 应用程序的编写变得更加容易。HtmlAnchor、HtmlContrl 以及 HtmlForm 是 ASP.NET 中的几个类,它们被包含在 System.Web.UI.HtmlControls 名空间中。

4. 服务器控件

ASP.NET 提供了许多功能强大的服务器控件,这大大简化了 Web 页面的创建任务。这些服务器控件提供从显示、日历、表格到用户输入验证等通用功能,它们自动维护其选择状态,并允许服务器端代码访问和调用其属性、方法和事件。因此,服务器控件提供了一个清晰的编程模型,使得 Web 应用的开发变得简单、容易。

5. 支持 Web 服务

ASP.NET 提供了强大的、标准化的 Web 服务支持能力,通过使用 Internet 标准,可以将一个 Web 服务和其他 Web 服务集成在一起。Web 服务提供了构建分布式 Web 应用的基本模块。ASP.NET 允许使用 and 创建 Web 服务。

6. 更高的安全性

与 ASP 相比,在支持常规 Windows 身份验证方法的基础上,ASP.NET 还提供了 Passport 和 Cookie 两种不同类型的登录和身份验证方法。同时,ASP.NET 还采用了基于角色的安全模式,为不同角色的用户指定不同的安全授权。另一方面,ASP.NET 还使得创建基于页面的身份验证工作变得更为简单。

7. 良好的可伸缩性

在 ASP.NET 中,允许使用跨服务器会话(Cross - Server Sessions),其会话状态可以被另一台机器或另一个数据库上的其他的进程所维护。随着信息处理和传输流量的增加,可以为系统添加更多的 Web 服务器。

8. 无 Cookie 会话

即使在浏览器不允许使用 Cookie 的情况下,ASP.NET 仍然能够使用户使用会话状态。与带 Cookie 的会话不同,无 Cookie 会话是通过 URL 将会话标识(SessionID)传递到 ASP.NET 页面的。

9. 代码分离

在 ASP 中,一个 Web 页面是由 HTML 与脚本代码混合使用而形成的。这种混合增加了程序

代码的阅读、调试和维护难度。而在 ASP.NET 中,用户界面布局和用户界面的程序逻辑可以分别设计和存储,这种代码与内容的分离,大大提高了 ASP.NET 页面的设计效率以及程序代码的可阅读性、可调试性与可维护性。

10. 易于配置和管理

ASP.NET 使用一个基于文本的、分层次的配置系统,所有的详细配置信息被存放在一个名为 web.config 的文本文件中,配置内容用标准的可扩展标示语言(XML)语法来书写,每一个 Web 应用都会继承web.config文件中的默认配置,从而简化了服务器端环境和 Web 应用程序的设置。除此之外,部署 ASP.NET 应用程序到服务器,也被简化为只需复制必要的文件到服务器即可,甚至连替换正在运行的页面代码,也不需要重新启动服务器。这使得 Web 站点的管理更加容易。

1.2.4 ASP.NET 的运行模式

从使用的角度看,ASP.NET 的运行过程包含页面请求、分析、编译、组装、页面缓冲五大环节。当客户端通过浏览器请求(Request)页面时,服务器端首先由页面分析器(Parser)对被请求的页面进行分析;再将通过分析的内容传递给编译器(Compiler);经过编译器编译的内容被传递给组装缓存(Assembly Cache),同时,一些需要较高资源代价的元素可以创建一次后存入内存(Memory);将组装缓存和内存中的内容有机结合后形成一个完整页面(包括数据、编译代码、HTML 代码等),完整页面最后被送往输出缓存(Output Cache)。输出缓存中的内容将作为客户端的页面请求结果被送回浏览器。当同一页面被第二次请求时,服务器端将跳过所有中间环节,直接从输出缓存中送出页面请求结果。所以,第一次请求 ASP.NET 页面时,服务器端的响应速度较慢,以后的响应速度则很快,这使得 ASP.NET 的整体运行效率远高于 ASP。图 1-3 给出了 ASP.NET 的具体运行模式。

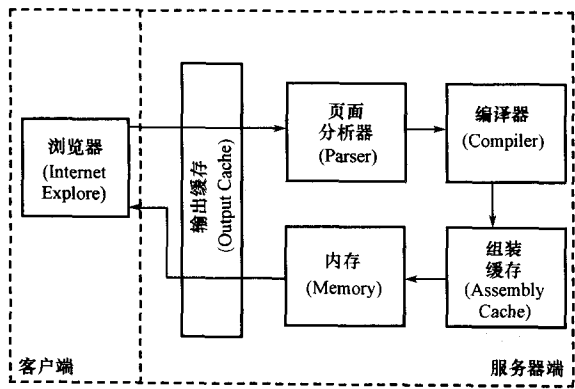


图 1-3 ASP.NET 的运行模式

1.2.5 ASP.NET 的文件类型

一个完整的 ASP.NET 应用,是指某个虚拟目录及其子目录中包括 Web 窗体、Web 服务、模板、执行代码、配置参数等所有文件的总和。每种文件用不同的类型后缀来区分,就像 ASP.NET 页面文件用.aspx 来表示一样。为了便于认识 ASP.NET 中的每一种文件,表 1-1 列出了常用文

件的类型与扩展名。

表 1-1 ASP.NET 应用中常见的文件类型

文件类型	扩展名
ASP.NET 页面文件	.aspx
ASP.NET Web 服务代码文件	.asmx
ASP.NET 页面代码文件	.aspx.vb (用 vb.net 语言)
	.aspx.cs (用 C# 语言)
ASP.NET Web 服务代码文件	.asmx.vb (用 vb.net 语言)
	.asmx.cs (用 C# 语言)
组件、自定义控件	.vb / .cs (源文件) .dll (编译后)
.NET XML 资源模板文件	.resx
用户控件	.ascx
XML Schema	.xsd
XML 文件	.xml
HTML Page	.html .htm
Active Server Page	.asp
Global Application Class	.asax
Jscript 文件	.js
VBScript 文件	.vbs
Web 配置文件	.config
样式表	.css
VB 源程序文件	.vb
C# 源程序文件	.cs

1.3 ASP.NET 运行环境

ASP.NET 的运行环境,是指仅运行已经开发好的 ASP.NET 应用所需要的基本资源条件,它包括硬件资源和软件资源两个方面。为了获得更好的系统运行性能,在介绍了 ASP.NET 的基本运行环境后,再给出一个推荐的运行环境。

掌握 ASP.NET 的运行环境,对于建立支持 ASP.NET 服务能力的 Web 服务环境非常重要,它将有效避免由于直接使用开发环境提供服务而带来的资源浪费和不安全隐患。有关详细内容请参见第 8 章。

1.3.1 运行环境要求

1. 基本运行环境

(1) 硬件

- ① CPU: Intel Pentium II - class 300 MHz;

- ② 内存: 128 MB;
- ③ 适当的可用磁盘空间;
- ④ 显示: 800 × 600, 256 色;
- ⑤ CD - ROM: 可选。

(2) 软件

- ① 操作系统: Window XP / Windows 2000 系列 / Windows NT4(带 SPk 6);
- ② Web 服务: IIS 5.0 以上 + Frontpage 2000 服务器扩展;
- ③ IE 版本: 5.01 以上;
- ④ 数据库访问组件: MDAC 2.6;
- ⑤ 数据库服务(可选): MS SQL Server 2000 / OLE DB;
- ⑥ 支持平台: Microsoft .NET Framework 1.0 + SPK2 / 1.1。

2. 推荐的运行环境

(1) 硬件

- ① CPU: Intel Pentium IV 2.0 GHz 以上;
- ② 内存: 256 MB 以上;
- ③ 足够的可用磁盘空间;
- ④ 显示: 1 024 × 768, 增强 16 位彩色;
- ⑤ CD - ROM: 可选。

(2) 软件

- ① 操作系统: Windows 2000 Advanced Server(带 SPK3);
- ② Web 服务: IIS 5.0 以上 + Frontpage 2000 服务器扩展;
- ③ IE 版本: 5.5 以上;
- ④ 数据库访问组件: MDAC 2.7;
- ⑤ 数据库服务: MS SQL Server 2000;
- ⑥ 支持平台: Microsoft .NET Framework 正式版 1.1。

在以上的基本环境中,支持平台如果使用 Microsoft .NET Framework 1.0,则最好应该下载并安装 .NET Framework Service Pack 2,因为它提供了 .NET Framework 的最新更新。推荐的运行环境既考虑了硬件处理能力,更考虑了操作系统的服务能力和软件系统之间的有效配合。比如,Windows 2000 Advanced Server 操作系统和 MS SQL Server 2000 数据库服务是最佳搭配,它们能比其他组合提供更有有效的数据库服务能力。

需要说明的是,2003 年 5 月 22 日,Microsoft 公司在中国大陆正式发布了其最新产品 Microsoft Windows Server 2003 简体中文版,同步上市的还有 Microsoft Visual Studio .NET Version 2003 及 Microsoft SQL Server 2000 企业版(64 位)。这标志着 Microsoft 在企业级市场战略上沿着 Microsoft .NET 的远景又迈出了一大步。这 3 个产品的组合能够更好地满足客户不断变化的业务需求和信息技术的挑战,使其 IT 建设具备更好的可管理性、连接性、安全性和可扩展性。所以,经过一段时间的使用和修补,Microsoft Windows Server 2003 简体中文版和 Microsoft SQL Server 2000 企业版(64 位)的配合使用,将成为 ASP.NET 的最佳软件运行环境。