

国家信息专业技术人才知识更新工程（“653工程”）指定参考用书

项目实践精解：

ASP.NET应用开发

—— 基于ASP.NET、C#和 ADO.NET的三层架构案例分析

梁立新 编著
亚思晟科技 审校

光盘内包含
书中**实例代码**，
项目案例的**源代码**、
部署和运行



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

项目实践精解丛书

国家信息专业技术人才知识更新工程（“653工程”）指定参考用书

项目实践精解：

ASP.NET应用开发

—— 基于ASP.NET、C#和 ADO.NET的三层架构案例分析

梁立新 编著

亚思晟科技 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书通过丰富的项目实践案例和深入浅出的讲解,引领读者学习了 ASP.NET 应用完整的开发体系,以及 ASP.NET、C#和 ADO.NET 等各个组件在多层架构项目开发中的实际应用。

本书全面地介绍了 ASP.NET 应用开发的分析、设计、开发和测试技术。本书的特色是以项目实践作为主线,提供了一个完整的 AscentWeb 医药商务系统,通过对该项目各个功能模块从需求分析、系统分析设计、部署实现及测试等全过程的分析,使读者能够快速掌握 ASP.NET 开发中最核心的组件。本书包括了 ASP.NET 开发概述、C#和 ADO.NET 背景知识、ASP.NET Web 控件、ASP.NET 内置对象、ASP.NET 应用程序配置、ASP.NET 用户控件和自定义控件、ASP.NET Web 服务、ASP.NET 性能优化等内容。另外,本书还补充介绍了 XML、JavaScript 和测试等技术的概念及应用,同时,特别奉献了 Web 开发的前沿技术——Ajax。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ASP.NET 应用开发:基于 ASP.NET、C#和 ADO.NET 的三层架构案例分析 / 梁立新编著. —北京:电子工业出版社, 2010.9

(项目实践精解丛书)

ISBN 978-7-121-11557-8

I. ①A… II. ①梁… III. ①主页制作—程序设计 IV. ①TP393.09

中国版本图书馆 CIP 数据核字 (2010) 第 154729 号

责任编辑:贾 莉

印 刷:北京东光印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:24 字数:614 千字

印 次:2010 年 9 月第 1 次印刷

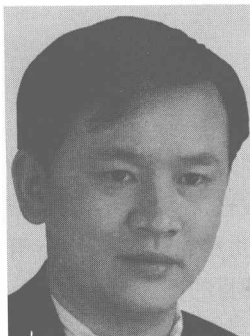
印 数:4000 册 定价:55.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

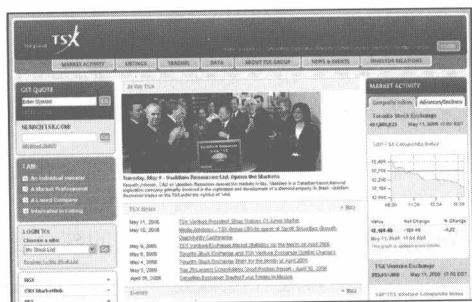
服务热线:(010) 88258888。

作者简介

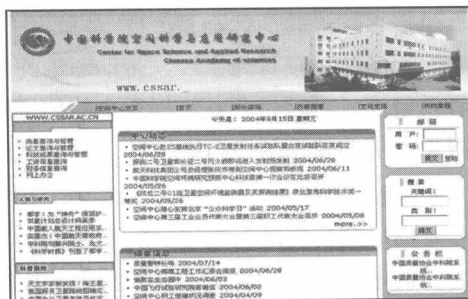


梁立新, 外籍软件专家, 具有十多年专业的软件开发、架构设计和项目管理的经验, 擅长面向对象的分析设计及基于 J2EE 和 .NET 架构的软件开发技术和基于 UNIX 平台的 C/C++ 编程技术。毕业于中国科学技术大学, 获硕士学位。之后留学于美国, 获伊利诺依理工大学硕士学位。曾先后工作于美国华尔街咨询服务公司和加拿大多伦多证券交易所, 担任高级软件设计师。参与设计建设了美国著名银行 JP Morgan 网上人力资源系统和加拿大最大的证券交易中心 Toronto Stock Exchange 股票交易系统和市场数据传输及分析系统。回国后, 创办北京亚思晟商务科技有限公司, 设计和开发了中科院空间中心电子政务系统, 网上企业财务中心管理系统及 eBiz 企业 ERP 管理系统等, 同时从事高端 IT 的培训, 软件教育产品研发和咨询工作。

参与设计建设的项目展示

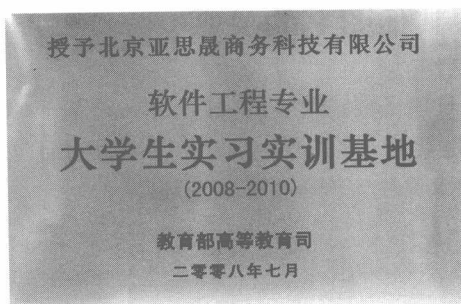


项目开发: Toronto Stock Exchange 股票交易系统



项目开发: 中国科学院空间中心电子政务系统

回国后创建的亚思晟商务科技有限公司风貌



教育部授牌



公司实训环境

前言

作者从事软件设计开发工作十多年，最近才萌发了写书的想法。因为作者希望推广一种最有效的学习捷径，这就是 Project-Driven Training，也就是用项目实践来带动理论学习的方法。基于此点，作者围绕一个实践项目（AscentWeb 医药商务系统）来贯穿 ASP.NET Web 开发各个模块的理论讲解。通过项目实践，读者可以对技术应用有明确的目的性（为什么学），也可以对技术原理更好地融会贯通（学什么），还可以更好地检验学习效果（学得怎样）。

本书的内容是以作者在亚思晟高端实训过程中的讲义和项目实践为基本素材整理出来的，同时结合了多年项目设计开发经验，具有良好的实践性和可操作性。

本书特点：

1. 重项目实践

笔者多年经验的体会是“IT 是做出来的，不是想出来的”，理论虽然重要，但一定要为实践服务！以项目为主线，带动理论的学习是最好、最快、最有效的方法！通过此书，笔者希望读者对项目开发流程有个整体了解，减少对项目实践的盲目感和神秘感，并且能够根据本书的体系循序渐进地动手做出自己的真实项目来！

2. 重理论要点

本书以项目实践为主线，着重讲述 ASP.NET Web 开发理论中最重要最精华的部分及它们之间的融会贯通，这是本书的特色！作者认为最有效的学习途径应该是读者首先通过项目把握整体概貌，再深入局部细节，系统学习理论，最后不断优化和扩展细节，完善整体框架、改进项目。

本书既有整体框架，又有重点技术。一书在手，理论清晰，项目无忧！

本书与《项目实践精解》系列中其他图书的关系

本书介绍的是 ASP.NET 应用。它是系列丛书《项目实践精解》中的一本，是基于《项目实践精解：C#核心技术应用开发》之上的，也是学习《项目实践精解：IT 项目的面向对象分析设计、开发及管理——基于.NET 平台的 ERP 系统案例分析》的基础。

本书的组织结构和学习指南

常常有开发人员和学员问我，什么是学习软件技术的最佳途径？我的回答是“多做，以项目驱动”。学员又继续问我“那学习理论就不重要了吗？”，我的回答是“当然也重要，但同样需要以项目驱动”。为什么呢？试想一下，当你面对上百页的参考书或者文档时，是否会茫然不知所措，不知如何开始，不知彼此谁更重要？所以只有以项目作为驱动，才

能更好地明确重点、明确主线，才能更好地分配时间和精力，这样才是最有效的。

因此，本书围绕一个完整的 AscentWeb 医药商务项目来组织和设计学习 ASP.NET Web 开发。

第一部分 整体介绍

第 1 章，ASP.NET 开发概述，主要包括学习 ASP.NET 应用开发需要掌握的技术要点及它们之间的关系。通过这一章的学习，读者会对 ASP.NET 开发架构有一个宏观的了解，建立一个整体印象。

第 2 章，AscentWeb 医药商务项目背景知识。作者希望推广一种最有效的学习与培训的捷径，这就是 Project-Driven Training，也就是用项目实践来带动理论的学习。这一章介绍了项目开发的背景知识，包括项目整体流程 RUP（Rational Unified Process）和 UML（Unified Modeling Language）语言。接下来重点介绍了完整的 AscentWeb 医药商务项目，提供了完整的需求分析、系统分析设计 UML 图、数据库设计 E-R 图，等等。我们使用基于 ASP.NET、C# 和 ADO.NET 的三层架构技术来设计开发这个项目。

第 3 章，C# 核心技术，主要内容包括 C# 概述和程序结构、C# 基本语法、C# 面向对象核心技术和高级语法，以及 C# 在 AscentWeb 医药商务项目中的应用实例。

第 4 章，ADO.NET 核心技术，主要内容包括数据库基础和 SQLServer 的基本使用、ADO.NET 概述和基本原理、ADO.NET 应用程序开发基本步骤和结构、ADO.NET 常见对象介绍，以及 ADO.NET 在 AscentWeb 医药商务项目中的应用实例。

第二部分 ASP.NET 开发核心技术

根据项目开发流程，下一步就进入了项目具体实现阶段，也就是使用 ASP.NET 核心技术进行开发和编码的阶段。

第 5 章，ASP.NET Web 控件技术，主要内容包括 ASP.NET Web 页面框架和原理，Web 窗体常用控件的属性、方法和使用介绍，以及 ASP.NET Web 控件在 AscentWeb 医药商务项目中的应用实例。

第 6 章，ASP.NET 内置对象，主要内容包括 Request 对象、Response 对象、Session 对象、Server 对象、Cache 对象，Global.asax 配置文件的使用，以及 ASP.NET 内置对象在 AscentWeb 医药商务项目中的应用实例。

第 7 章，ASP.NET 应用程序配置，主要内容包括 ASP.NET 应用程序概述、ASP.NET 应用程序配置文件格式、ASP.NET 应用程序配置文件配置节点介绍，以及应用程序配置和部署在 AscentWeb 医药商务项目中的应用实例。

第 8 章，用户控件和自定义控件，主要内容包括用户控件原理和开发、自定义控件原理和开发，以及用户控件和自定义控件的比较。

第 9 章，ASP.NET Web 服务，主要内容包括 Web 服务概述和基本原理、ASP.NET 中 Web 服务的创建和使用，以及 Web 服务在 AscentWeb 医药商务项目中的应用实例。

第 10 章，ASP.NET 数据缓存技术，主要内容包括页面输出缓存和页面数据缓存等技术。

第三部分 其他相关技术

第 11 章,XML 技术,主要内容包括 XML 原理概述(XML、namespace 和 DTD/ Schema)、在 ASP.NET 中使用 XML、使用 XSL 转换 XML,以及 XML 在 AscentWeb 医药商务项目中的应用实例。

第 12 章, Javascript 技术,主要内容包括 JavaScript 语言概况、JavaScript 基本数据结构、JavaScript 程序构成、基于对象的 JavaScript 语言、使用内部核心对象系统,以及 JavaScript 在 AscentWeb 医药商务项目中的应用实例。

第 13 章, Web 开发的前沿技术——Ajax,主要内容包括 Web 2.0 简介和 Web 2.0 编程思想的 16 条法则、Ajax 简介、Ajax 核心技术概述及 Ajax 框架和开发工具,以及 Ajax 在 AscentWeb 医药商务项目中的应用实例。

第 14 章, 测试技术,主要内容包括测试概述和原理、测试分类、.NET 单元测试工具 NUnit 和 TestDriven.NET 及测试跟踪工具 Bugzilla 的使用等,以及单元测试在 AscentWeb 医药商务项目中的应用实例。

全书以完整的 AscentWeb 医药商务项目贯穿,以项目驱动理论学习。

这本书是否适合您?

阅读此书,要求读者具备 C#核心技术和 ADO.NET 的基础知识。

本书结构清晰、注重实用、深入浅出。它非常适合作为 ASP.NET Web 开发的培训教材或自学教材,同时,也适合作为 ASP.NET Web 开发人员的参考用书。

感谢

本书由梁立新主持编写,其他参与编写的人员有武永琪、于亚杰、孙夏、张瑞、梁恒、张洪亮、沈彬、郭锐、刘执勇。

本书编写过程中得到了许多支持和帮助。北京亚思晟科技有限公司负责本书的审校和监制工作;公司研发人员利用宝贵时间为本书提供项目相关代码与文档,并测试了项目的功能和性能;朋友和家人对本书提供了有益的建议和帮助;在此表示衷心的感谢。最后,感谢电子工业出版社对本书出版的协助。

联系方式:

北京亚思晟商务科技有限公司

地址:北京海淀上地东路 1 号院鹏寰国际大厦 501

网址: www.ascenttech.com.cn

电话: 58859825/26/27/28/29

目 录

第一部分 背景知识

第1章 ASP.NET 开发概述	2	3.2 C#面向对象技术	49
1.1 Microsoft .NET 介绍	2	3.2.1 面向对象的概念	49
1.1.1 Microsoft .NET 概述	2	3.2.2 封装	53
1.1.2 Microsoft .NET 平台的意义	2	3.2.3 继承	55
1.1.3 Microsoft .NET 的基本模块	3	3.2.4 多态	57
1.2 ASP.NET 概述	4	3.3 C#高级技术	61
1.2.1 ASP.NET 历史和特性	4	3.3.1 静态 (static) 变量和方法	61
1.2.2 ASP.NET 的优势	5	3.3.2 密封 (sealed) 类和方法	62
1.3 ASP.NET 相关技术	6	3.3.3 访问控制 (Access Control)	63
1.4 Visual Studio 集成开发环境 (IDE)		3.3.4 抽象类与抽象方法	63
介绍	8	3.3.5 接口 (interface)	65
第2章 AscentWeb 医药商务项目		3.3.6 集合 (Collection)	68
背景知识	16	3.4 AscentWeb 医药商务系统应用实例	74
2.1 项目开发背景知识	16	第4章 ADO.NET 数据库编程	81
2.1.1 项目开发流程	16	4.1 数据库基础	81
2.1.2 UML 概述	20	4.2 SQL (Structured Query Language)	
2.2 AscentWeb 医药商务系统概述	24	的使用	82
2.2.1 项目需求分析	25	4.3 ADO.NET 概述	82
2.2.2 项目系统分析和设计	30	4.3.1 ADO.NET 对象模型	82
2.2.3 项目运行指南	43	4.3.2 ADO.NET 开发步骤	84
第3章 C#基础	47	4.4 AscentWeb 医药商务系统应用	
3.1 C#程序的基本结构	47	实例	87

第二部分 核心知识

第5章 ASP.NET Web 控件	94	5.2.4 HTML 控件和 Web 控件	
5.1 Web 页面框架	94	的比较	136
5.2 Web 窗体控件	98	5.3 AscentWeb 医药商务系统应用实例	137
5.2.1 Web 窗体控件概述	98	第6章 ASP.NET 内置对象	145
5.2.2 HTML 控件	105	6.1 Response 对象	145
5.2.3 Web 控件	106	6.2 Request 对象	147

6.3 Application 对象	149	8.1.1 用户控件概述	178
6.4 Session 对象	151	8.1.2 将 Web 窗体页转换为用户 控件	179
6.5 Server 对象	154	8.1.3 用户控件的开发	180
6.6 Cookie 对象	157	8.2 自定义控件	182
6.7 Cache 对象	158	8.2.1 简单自定义控件	183
6.8 “Global.asax” 文件	159	8.2.2 复合自定义控件	185
6.9 AscentWeb 医药商务系统应用实例	160	8.3 用户控件与自定义控件的比较	190
第 7 章 ASP.NET 应用程序配置、 编译和部署	163	第 9 章 ASP.NET 与 Web Services	192
7.1 应用程序概述	163	9.1 面向服务的软件架构概述	192
7.2 Web.config 配置	164	9.1.1 面向服务的软件架构 (SOA)	192
7.2.1 ASP.NET 应用程序配置 简介	164	9.1.2 SOA 与 Web2.0	193
7.2.2 ASP.NET 配置文件的继承 层次结构	164	9.2 Web Services 的概念	194
7.2.3 配置文件的格式	165	9.2.1 Web Services 的核心技术	195
7.2.4 配置元素	165	9.2.2 Web Services 原理	197
7.3 ASP.NET 网站的预编译和编译	171	9.3 ASP.NET 与 Web 服务	203
7.3.1 ASP.NET 网站的预编译	171	9.4 AscentWeb 医药商务系统中的应用 实例	206
7.3.2 ASP.NET 网站的编译	173	第 10 章 缓存技术	210
7.4 AscentWeb 医药商务系统应用实例	173	10.1 页面输出缓存	211
第 8 章 用户控件和自定义控件	178	10.2 页面部分缓存	212
8.1 用户控件	178	10.3 应用程序数据缓存	213

第三部分 其他相关技术

第 11 章 ASP.NET 与 XML	218	11.3 XML 的操作	236
11.1 XML 概述	218	11.3.1 XML 控件	236
11.1.1 XML 定义	218	11.3.2 XmlTextReader 和 XmlTextWriter	239
11.1.2 XML 语法	221	11.3.3 XmlDocument (W3C DOM)	244
11.1.3 DTD 与 XML Schema	224	11.3.4 使用 DataSet 对象	246
11.2 XML 转换 (Transformation)	231	11.4 AscentWeb 医药商务系统的应用 实例	247
11.2.1 XML 转换概述	231		
11.2.2 XSL 使用	232		

第 12 章 JavaScript	250
12.1 JavaScript 简介	250
12.1.1 什么是 JavaScript	250
12.1.2 JavaScript 开发	251
12.2 JavaScript 基本数据结构	253
12.2.1 基本数据类型	253
12.2.2 表达式和运算符	254
12.3 JavaScript 程序构成	256
12.3.1 流程控制	256
12.3.2 函数	257
12.3.3 事件驱动及事件处理	258
12.4 基于对象的 JavaScript 语言	261
12.4.1 对象的基础知识	261
12.4.2 创建新对象	263
12.4.3 JavaScript 内部核心对象	266
12.4.4 使用浏览器对象系统	268
12.5 JavaScript 在 AscentWeb 医药商务 系统中的应用实例	281
第 13 章 Ajax	288
13.1 Web 2.0 概述	288
13.2 Ajax 技术	289
13.2.1 Ajax 简介	289
13.2.2 Ajax 核心技术概述	291
13.2.3 XMLHttpRequest	294
13.3 AscentWeb 医药商务系统中的 应用实例	300
第 14 章 测试概述	305
14.1 软件测试概述	305
14.2 常用测试技术	306
14.3 测试工具	308
14.3.1 NUnit	308
14.3.2 TestDriven.NET	314
14.3.3 测试跟踪工具 Bugzilla	318
14.4 AscentWeb 医药商务系统应用 实例	321
附录	337
结束语	372

第一部分

背景知识

第 1 章 ASP.NET 开发概述

目前,国内外信息化建设已经进入以基于 Web 应用为核心的阶段,ASP.NET 作为应用于网络的技术,前景无限好。ASP.NET 作为 .NET 平台的一个核心组件,起着至关重要的作用。在这里我们首先概要地介绍一下 .NET 平台和 ASP.NET 核心技术,之后再简单介绍一些和 ASP.NET 相关的技术。

1.1 Microsoft .NET 介绍

1.1.1 Microsoft .NET 概述

相信你早已听过 .NET 这个概念。微软希望 .NET 能做到让任何人从任何地方、在任何时间、使用任何装置来获得互联网上的服务。Microsoft .NET 包括用于创建和操作新一代服务的 .NET 基础结构和工具,用于建立新一代高度分布式的数以万计的 .NET 组件服务,以及用于启用新一代智能互联网设备的 .NET 设备软件。

1.1.2 Microsoft .NET 平台的意义

我们来看一下 Microsoft .NET 对开发人员、IT 专业人员及企业应用的巨大意义。

1. 对于开发人员

Microsoft .NET 的策略是将互联网本身作为构建新一代操作系统的基础,对互联网和操作系统的设计思想进行合理扩展。.NET 对开发人员来说十分重要,它不但会改变开发人员开发应用程序的方式,而且会使得开发人员能创建出全新的各种应用程序。新型开发范例的核心是“Web 服务”这个概念的引入。Web 服务是一种应用程序,它可以通过编程并使用标准的 Internet 协议(像超文本传输协议 HTTP 和 XML 等),将功能展示在互联网和企业内部网上。我们可将 Web 服务视作 Web 上的组件编程,从理论上讲,开发人员可通过调用 Web 应用编程接口(API),将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似,不同的是 Web API 调用可通过互联网发送给位于远程系统中的某一服务。

.NET 正是根据这种 Web 服务原则创建的,微软目前正着手提供这个基本结构,以便通过 .NET 平台的每一部分来实现这种新型的 Web 服务。而 Visual Studio .NET、.NET 框架、Windows .NET 和 .NET 企业服务器,正是为进行基于 Web 服务模型的应用程序开发而量身定做的新一代开发工具和基本结构。

2. 对于 IT 专业人员

目前,IT 专业人员能够利用与构建 .NET 平台所使用的技术相同的技术。.NET Enterprise

Servers 和 Windows 操作系统, 为创建具有高度可管理性的、能迅速投入市场的应用程序提供了坚实基础。它们利用的是可扩展标记语言 (XML), 因此随着 Web 体系结构的革新, 在此平台上创建的程序依然很有价值。

开发应用程序的 .NET Web 服务模型将为企业应用程序的创建开辟一条新路。通过企业内外多种服务的联合, 很容易把企业内部数据和客户及合作伙伴的相关数据结合在一起, 大大简化了应用程序的创建过程, 这就为最终用户发掘了巨大的功能涵盖性。

3. 对于企业

Microsoft .NET 平台将从根本上改善计算机和用户之间进行交互的方式, 最大限度地发挥电子商务中计算技术的重要作用。.NET 能实现用户从任何地点、任何设备都可访问其个人数据和应用程序。除此之外, .NET 技术还可实现多个应用程序在逻辑上的松散或紧密的耦合连接。根据设计, .NET 使得用户无需在如何与计算机进行交互上费力, 从而全身心地投入到使计算机自动执行任务、实现最终目标的工作中。通过使用 XML 行业标准, 可将用户数据进行跨站点和应用程序的连接, 从而轻松实现当前很难实现的操作。

.NET 把雇员、客户和商务应用程序整合成一个协调的、能进行智能交互的整体, 而各公司无疑将是这场效率 and 生产力革命的最大受益者。简言之, .NET 将致力于为人类创造一个消除任何鸿沟的商务世界。

1.1.3 Microsoft .NET 的基本模块

Microsoft .NET 包括 5 个主要组成部分, 即 Windows .NET、.NET 框架 (.NET Framework)、.NET 企业服务器、模块构建服务 (Building Block Services) 和 Visual Studio .NET。Windows .NET 是融入 .NET 技术的 Windows, 它紧密地整合了 .NET 的一系列核心构造模块, 为数字媒体及应用间协同工作提供支持, 是 Microsoft 公司的下一代 Windows 桌面平台。.NET 框架便于开发商更容易地建立网络应用程序和 Web 服务。它的关键特色是提供了一个多语言组件开发和执行的环境。.NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础, 它为企业未来开展电子商务提供了高可靠性、高性能、可伸缩性和可管理性。模块构建服务是 .NET 平台中的核心网络服务集合。Visual Studio .NET 是基于 XML 的编程工具和环境, 它便于快速开发符合 .NET 体系的软件服务, 使其在独立设备、企业数据中心和互联网之间的传送更加容易。

其中, 最核心的部分是 .NET 框架, .NET 框架的设计目标包括:

- 提供一个一致的面向对象的编程环境, 无论对象代码是在本地存储和执行的、在本地执行但在互联网上发布的还是在远程执行的。
- 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。
- 提供一个可提高代码 (包括由未知的或不完全受信任的第三方创建的代码) 执行安全性的代码执行环境。
- 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。
- 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。
- 按照工业标准生成所有通信, 以确保基于 .NET 框架的代码可与任何其他代码集成。

.NET 框架包括两个主要组件: 公共语言运行时 (Common Language Runtime, CLR) 和 .NET 框架类库 (Framework Class Library, FCL)。

公共语言运行时是 .NET 框架的基础, 可以将它看做一个在执行时管理代码的代理, 它

提供内存管理、线程管理和远程处理等核心服务,并且强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码准确性。代码管理的概念是公共语言运行时的基本原则。以公共语言运行时为目标的代码称为托管代码。.NET 框架用统一的命令集来支持任何的编程语言,支持混合语言编程,确保程序的可移植性。托管代码只是意味着在内部可执行代码与运行自身间存在已定义好的合作契约。对于像生成对象、调用方法等这样的任务,被委托给了运行语言,这使得在运行语言能为可执行代码提供额外的服务。

.NET 框架类库是一个综合性的面向对象的 reusable 类的集合,可以使用它来开发多种应用程序,这些应用程序包括传统的命令或图形用户界面应用程序,也包括基于 ASP.NET 所提供的最新创新的应用程序(如 Web 窗体和 XML Web Services)。.NET 框架类库确保用户程序能够访问公共语言运行时环境。

语言互操作是一种代码与使用其他编程语言编写的另一种代码进行交互的能力。语言互操作有助于最大限度地提高代码的重用率,从而提高开发过程的效率。公共语言运行时提供内置的语言互操作支持。.NET 框架通过如下三个基础来保障语言互操作:

- 中间语言 (Microsoft Intermediate Language, 简称为 MSIL 或 IL), 所有的 .NET 语言都被编译为中间语言。这是一组可以有效地转换为本机代码且独立于 CPU 的指令。IL 包括用于加载、存储、初始化对象及调用对象的方法的指令,还包括用于算术和逻辑运算、控制流、直接内存访问、异常处理和其他操作的指令。要使代码可运行,必须先将 IL 编译为特定 CPU 的机器码,这通常是通过即时编译器 (JIT) 来完成的。在编译过程中,必须通过验证过程来查看是否可以将代码确定为类型安全。代码只需即时编译一次。当再次运行编译过的代码时,将运行已经通过 JIT 编译得到的机器码。这种进行 JIT 编译然后执行代码的过程一直重复到执行完成时为止。
- 通用类型系统 (Common Type System, CTS), 提供了定义、管理和使用类型的规范。它提供了所有支持语言互操作的语言都必须遵守的规则集,以确保不同语言所创建的类型能够进行交互操作。
- 公共语言规范 (Common Language Specification, CLS), 定义了编译器和库管理器必须遵守的规则,以确保它们所生成的语言和代码能够与其他 .NET 语言进行互操作。公共语言规范是通用类型系统的一个子集。

1.2 ASP.NET 概述

ASP.NET 是微软主推的新技术,是微软 .NET 战略中的一部分。ASP.NET 是运行于 Windows 平台 .NET 框架下的一种新型的功能强大的 Web 编程语言,一经推出就备受关注。发展至今,ASP.NET 经过几年的改进和优化,已逐渐成为一种成熟、稳定的 Web 编程语言。

1.2.1 ASP.NET 历史和特性

ASP 的第一个版本是 0.9 测试版。它给 Web 开发带来了巨大的影响,它能够将代码直接嵌入 HTML,使得设计 Web 页面变得更简单、更强大,并且通过内置的组件能够实现强大的功能。

之后出现的是 Active Server Pages 1.0,它作为 IIS 的附属产品免费发送,并且不久就在 Windows 平台上广泛使用。ASP 与 ADO 的结合使用使开发者很容易地在一个数据库中建立和打开一个记录集。这无疑是它如此快就被大众接受的原因,因为用户现在能使用这些

脚本建立和打开一个记录集，处理和输出任何数据，以任何顺序就能完成。

1998年，微软公司又发布了ASP 2.0。ASP 1.0和ASP 2.0的主要区别是外部的组件需要实例化。有了ASP 2.0和IIS 4.0，我们就有可能建立ASP应用了，而且每个组件就有了自己单独的内存空间。内置的Microsoft Transaction Server (MTS)也使制作组件变得简单。微软公司接着开发了Windows 2000操作系统。这个Windows版本给我们提供了IIS 5.0和ASP 3.0。此次并不是简单地对ASP进行补充，而是把很多的事情交给了COM来做。在Windows 2000中，微软结合了MTS与COM核心环境做出了COM+，这就让主机有了一种新的方法来使用组件，同时也给主机带来了更大的稳定性，成为了一个可以升级的效率高的工作平台。IIS 5.0在表面上似乎没有什么改动，但是在接口上却做了很大变化。在内部，它使用COM+组件服务来对组件提供一个更好的执行的环境。

有了这些，微软公司推出了ASP.NET，ASP.NET不是ASP的简单升级，而是Microsoft推出的新一代Active Server Pages。ASP.NET是微软.NET体系结构的一部分，其中全新的技术架构会让每个人的编程生活变得更加简单。

ASP.NET利用Common Language Runtime和服务框架网络应用程序提供了一个可靠的、自动化的可扩展的主机环境。ASP.NET也受益于Common Language Runtime集成模板，简化了应用程序的配制。另外，它提供简化应用程序开发的服务（如状态管理服务）及高水平的编程模板（如ASP.NET Web Forms和ASP.NET Web Services）。

ASP.NET使用基于构件的Microsoft .NET框架配制模板，因此它获得了如XCOPY配制、构件并行配制、基于XML配制等优点。ASP.NET的另一个主要的优点是，它支持应用程序的实时更新。管理员不必关掉网络服务器，甚至不用停止应用程序的运行，就可以更新应用文件。应用程序文件永远不会被加锁，因此在程序运行时，文件也可以被覆盖。当文件更新后，系统会流畅地转换到新的版本。系统检测文件变化，并用新的应用程序代码建立一个新的应用程序实例，然后将引入的请求转发到应用程序。当所有被现存的应用程序实例处理的未完成的请求处理完后，该实例就被销毁了。

ASP.NET已经不再支持“代码块”了，一些动态的HTML都由Web服务器控件控制，通过Visible=“False|True”属性来决定控件的显示或隐藏，从而实现页面代码与应用程序分开。简单地说，一个ASP.NET应用程序其实是由两个文件.aspx和.cs构成的。其中，页面代码通常放置HTML、HTML控件和Web控件，后缀名为“.aspx”，而主程序代码则放在另一个后缀名为“.cs”的C#类文件中，也就是我们常说的“Code Behind”。另外，我们可以把所有一些相关的.cs类代码编译成后缀名为“.DLL”的文件，称为“复用控件”，也就是把一些重用性高的用户自己开发的控件集成在一起一次性编译成DLL文件，这个DLL文件不能反编译回C#代码，从而可以很有效地保护代码，并且能用来开发商业控件。因此，读者除了要学会C#的基本语法外，还必须熟悉Web控件的各项基本属性、行为和事件。这些都不是很难，Web控件大约有20个，花几天时间练习这些控件的用法，一般都没问题了。当然不是让读者全部记忆这些控件的属性等，读者应该在本地的机器上安装Microsoft .NET框架和Visual Studio .NET开发平台，通过简单地拖放控件和属性面板来学习这些控件，这样，你的学习效率会比手写代码高好几倍。

1.2.2 ASP.NET 的优势

ASP.NET是一种建立在通用语言上的程序构架，能被一台Web服务器用来建立强大的Web应用程序。ASP.NET提供许多比现在的Web开发模式强大的功能。

- 执行效率的大幅度提高

ASP.NET 是让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，而是将程序在服务器端首次运行时进行编译，这样的执行效果，当然比一条一条地解释强很多。

- 世界级的工具支持

ASP.NET 构架是 Microsoft 公司最新的产品之一。Visual Studio .NET 开发环境支持 WYSIWYG (What You See Is What You Get, 所见即所得) 的方式。当然这些仅是 ASP.NET 强大软件支持的一小部分。

- 强大性和适应性

因为 ASP.NET 是基于通用语言的编译运行的程序，所以它的强大功能和高适应性可以使它运行在 Web 应用软件开发者的几乎全部的平台。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。ASP.NET 同时也是语言独立化的，所以，用户可以选择一种最适合他的语言来编写程序，或者把程序用很多种语言来写，现在已经支持的有 C# (C++和 Java 的结合体)、VB 和 JScript 等。这个多种程序语言协同工作的能力保证您现在的基于 COM+开发的程序能够完整地向 ASP.NET 移植。

- 简单性和易学性

ASP.NET 使得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等）变得非常简单。例如，ASP.NET 页面构架允许你建立你自己的用户分界面，使其不同于常见的 VB-Like 界面。另外，通用语言框架能够大大简化开发。

- 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，新的设置有可能都不需要启动本地的管理员工具就可以实现。这种被称为“Zero Local Administration”的哲学观念使 ASP.NET 的基于应用的开发更加具体和快捷。一个 ASP.NET 的应用程序在一台服务器系统上的安装只需要简单的复制一些必需的文件，不需要系统的重新启动，一切就是这么简单。

- 多处理器环境的可靠性

ASP.NET 已经被刻意设计为一种可以用于多处理器的开发工具，它在多处理器的环境下用特殊的无缝连接技术，将极大地提高运行速度。即使你现在的 ASP.NET 应用软件是为一个处理器开发的，将来多处理器运行时，不需要任何改变，就能提高它们的效能，但现在的 ASP 却做不到这一点。

- 自定义性和可扩展性

ASP.NET 设计时考虑了让网站开发人员可以在自己的代码中定义“plug-in”的模块。这与原来的包含关系不同，ASP.NET 可以加入自己定义的任何组件。网站程序的开发从来没有这么简单过。

- 安全性

基于 Windows 认证技术和应用程序配置，你可以确定你的程序是安全的。

1.3 ASP.NET 相关技术

1. C#

前面已经提到，在 ASP.NET 业务逻辑开发过程中，我们需要使用 C#。C#作为.NET 平台的一个核心组件，起着至关重要的作用。微软公司对 C#的定义是：“C#是一种类型安全

的、现代的、简单的，由 C 和 C++ 衍生出来的面向对象的编程语言，它是牢牢根植于 C 和 C++ 语言之上的，并可立即被 C 和 C++ 的使用者所熟悉。C# 的目的就是综合 Visual Basic 的高生产率和 C++ 的行动力。”这个定义是恰如其分的，因为在 Visual Studio.NET 中，我们已经可以利用 C# 极其简单地开发出强大的企业级分布式应用系统了。

C# 是面向对象的编程语言，面向对象，作为一种思想及编程语言，为软件开发的整个过程，包括从分析设计到维护，提供了一个完整的解决方案。要想学好 C#，首先就要建立起面向对象的思想。我们首先要了解 C# 语言的基本语法，之后要重点掌握面向对象的核心技术，如类(Class)、对象(Object)、方法(Method)、封装(Encapsulation)、继承(Inheritance)、多态(Polymorphism)等。

2. ADO.NET

在 ASP.NET 应用开发中，数据库管理系统(RDBMS)的使用是不可缺少的。由于数据访问如此普遍，开发人员不断地为简单的数据库任务重新生成复杂的代码就显得毫无意义了，开发人员需要从格式各异的不同数据源中快速访问数据。ADO.NET 解决了这一问题。ADO.NET 是一种用于执行 SQL 语句的 .NET 技术。它为工具/数据库开发人员提供了一个标准的 API，使他们能够编写数据库应用程序。

ADO.NET 对象模型中有五个主要的组件，分别是 Connection 对象、Command 对象、DataAdapter 对象、DataSet 对象和 DataReader 对象。这些组件中负责建立联机和数据操作的部分我们称为数据操作组件(Managed Providers)，分别由 Connection 对象、Command 对象、DataAdapter 对象和 DataReader 对象所组成。数据操作组件最主要是作为 DataSet 对象和数据源之间的桥梁，负责将数据源中的数据取出后植入 DataSet 对象中，以及将数据存回数据源。

3. XML 语言

作为一种工具，可扩展标记语言(XML)简化了数据交换、进程间消息交换这一类的事情，因而对开发者而言，逐渐变得很有吸引力，并开始流行起来。另外，在 ASP.NET 架构中访问或集成 XML 解决方案的想法也很诱人，因为这将是强大系统架构同高度灵活的数据管理方案的结合。

XML 的应用似乎是无穷无尽的，但它们大致上可以分为三大类：

- 简单数据的表示和交换(针对 XML 的简单 API (SAX) 和文档对象模型 (DOM) 语法解析，不同的文档类型定义 (DTDs) 和概要 (Schemas))。
- 用户界面相关的上下文(可扩展样式表语言 (XSL)，可扩展样式表语言转换 (XSLT))。
- 面向消息的计算 (XML-RPC (远程过程调用)，基于 SOAP 协议的 Web 服务 (Web Services)，电子化业务 XML (ebXML))。

4. 网页脚本语言

为了提高 Web 项目的整体性能，提高人机交互的友好界面，网页脚本语言是很有用处的，有的时候可以解决很大的难题或提高程序的性能及应用性。网页脚本语言的执行都是在客户端进行的，速度很快，并且大多数的操作与服务器没有交互运算，所以在一些应用中非常理想。在设计 Web 项目的应用中，网页脚本语言起着不容忽视的作用，所以在开发 Web 项目的应用前，对 JavaScript 应有一定的了解。

JavaScript 是一种基于对象 (Object Based) 和事件驱动 (Event Driven)，并具有安全性能 (Secure) 的脚本语言。使用它的目的包括实现在一个 Web 页面中链接多个对象，与 Web 客户交互作用，等等，从而开发客户端的应用程序等。它是通过嵌入在标准的 HTML