

程序员项目开发实践系列



ASP.NET数据库系统 项目开发实践

曹 斌 韩中孝 主编



科学出版社
www.sciencep.com

程序员项目开发实践系列

ASP.NET 数据库系统

项目开发实践

曹 斌 韩中孝 主编

科学出版社

北 京

内 容 简 介

本书按照软件开发的基本过程,以系统的需求分析、系统设计、数据库设计和系统的详细设计为基本开发步骤,讲述以ASP.NET (C#)为工具开发Web应用程序的全过程。本书内容包括ASP.NET概述、eHR人力资源管理系统、企业网络办公自动化综合平台、企业资源维护综合管理系统、企业信息文档综合管理系统、企业电子商务销售管理系统、网络多彩留言版和ASP.NET应用安装程序等系统的开发。本书光盘中的实例可以直接运行。

本书内容紧凑、实例丰富、结构严谨、深入浅出,不论是对初学ASP.NET Web应用程序开发的用户,还是对ASP.NET Web应用程序开发有一定经验的用户,都会给予有益的帮助。

图书在版编目(CIP)数据

ASP.NET 数据库系统项目开发实践/曹斌,韩中孝主编.—北京:科学出版社,2005

(程序员项目开发实践系列)

ISBN 7-03-015601-3

I.A… II.①曹…②韩… III.主页制作-程序设计 IV.TP393.092

中国版本图书馆CIP数据核字(2005)第051666号

责任编辑:吕建忠 韩洁/责任校对:耿耘

责任印制:吕春珉/封面设计:飞天创意

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005年6月第一版 开本:787×1092 1/16

2005年6月第一次印刷 印张:31 1/2

印数:1—4 000 字数:730 000

定价:45.00元(含光盘)

(如有印装质量问题,我社负责调换<环伟>)

销售部电话 010-62136131 编辑部电话 010-62135397-8001 (B101)

前 言

ASE .NET 是一种强大的 Web 服务器端技术,与 ASP 相比,ASP .NET 拥有更高性能的编译特性与缓存机制。其特点包括:与其底层框架 .NET 紧密结合;将事件模型引入到 Web 应用程序的开发领域;支持多种语言开发,包括 C#、Visual Basic 和 Jscript 等。

本书属于案例开发系列,以开发会员制信息发布系统为基线,贯穿介绍 ASP .NET 的相关高级实现技术和理论,通过实例将知识点与开发实战紧密结合,从而达到学以致用的目的。通读本书,不仅可以全面掌握 ASP .NET 高级开发知识,而且可以了解更多 ASP .NET 应用技巧。研究本书案例还有助于读者体会到使用 ASP .NET 开发大型应用时需要注意哪些问题,帮助读者建立大局观。

本书第 1 章讲述了 ASP .NET 的基础知识、程序运行原理等。第 2 章讲解如何建立人力资源管理系统,本系统以是一种全新的动态加载的版面,可随用户的权限严格调整动态方式实现;系统设计采用三层架构模式,表面呈现层、业务逻辑层和数据访问层完全独立开来,系统的扩展性强。第 3 章实现了企业网络办公自动化综合平台,本系统是基于用户为对象的开发原则,每个用户可以设计自己的个性平台,实用性强。第 4 章实现了企业资源维护综合管理系统,本系统是以资源维护的及时性为原则,以及及时反馈资源维护申请的各种信息为宗旨,同时还包含了员工的业绩管理平台的企业资源维护综合管理系统。第 5 章实现了企业信息文档综合管理系统,本系统以企业信息文档的系统为主要数据,从信息文档的知识归档和共享为突破口,以简单、全新、全面、动态管理的方式管理企业信息文档。第 6 章实现了企业电子商务销售管理系统,本系统是一个典型现代电子商务网站,以消费的可靠性,系统的可扩展性,信息的及时性为原则而开发的企业电子商务销售管理系统。第 7 章实现了网络多彩留言板,在本系统中,能够满足网络用户文本输入的绝大多数需求,如上传图片、动画、文件,媒体等,同时还实现了字体的大小、颜色和格式的选择等功能。全书最后一章讲述 ASP .NET 应用程序的安装程序的制作和 ASP .NET 应用程序中 IIS 的基本配置,为其 Web 应用程序的使用和产品的推广带来了方便性。

本书不仅适合于从事 Web 开发的软件工程师和架构师,而且适合于本科、大专院校的计算机及其相关专业学生;本书还可以作为软件培训班的教材或者辅助材料。

本书由曹斌、韩中孝任主编,韩中领、王红卫、王凌志、王龙、曹树林、陈军、杜榕、龚燕平、苏瑞、黄裕荣、姜海英、金城、彭文明、宋秀坤、陶伟杰、王浩、向肃一、殷丽云、张佳祥、张磊、郑枫等同志参与了本稿的修改、整理工作。

由于编写时间较为仓促,书中难免会有疏漏和不足之处,恳请广大读者提出宝贵意见。

编者

目 录

第 1 章 ASP .NET概述	1
1.1 ASP .NET 简介	1
1.1.1 ASP .NET 简介	1
1.1.2 ASP.NET 程序运行原理	2
1.2 .NET Framework 简介	3
1.2.1 .NET Framework 概述	3
1.2.2 公共语言运行库	3
1.2.3 .NET Framework 基础类库	4
1.2.4 服务器应用程序开发	5
1.3 IIS 简介	5
1.3.1 IIS 简介	5
1.3.2 应用 IIS 配置 ASP.NET 程序	6
1.4 ASP .NET 中的状态管理	9
1.4.1 状态管理概述	9
1.4.2 Application 对象	10
1.4.3 Session 对象	10
1.4.4 状态管理注意事项	11
1.5 ASP .NET 安全分析	11
1.5.1 ASP .NET 安全	11
1.5.2 ASP .NET 的安全工作过程	13
1.5.3 ASP .NET 开发安全技巧	14
本章小结	16
第 2 章 eHR 人力资源管理系统	17
2.1 系统总体设计	17
2.1.1 应用背景	17
2.1.2 项目目标	17
2.1.3 解决方案设计	19
2.1.4 系统模块功能分析	19
2.2 数据库设计	22
2.2.1 数据库需求分析	22
2.2.2 数据库视图设计	22
2.2.3 数据库表设计	22
2.3 技术细节	24
2.3.1 存储过程的使用	24

2.3.2 动态生成页面类的封装	26
2.4 系统通用类和全局变量说明	29
2.4.1 通用类说明	29
2.4.2 全局变量	30
2.5 页面动态加载	31
2.5.1 程序主入口	31
2.5.2 动态页面加载	31
2.6 版面动态调整功能实现	35
2.6.1 界面设计	35
2.6.2 界面初始化	38
2.6.3 版面模块数据绑定	40
2.6.4 版面模块动态调整	41
2.6.5 保存修改到数据库	48
2.6.6 数据访问层类的封装	50
2.7 用户登录和权限动态分配	54
2.7.1 用户登录实现	54
2.7.2 用户权限分配	57
2.8 人事多彩留言板	63
2.8.1 多彩留言板	63
2.8.2 数据绑定	66
2.8.3 分页机制	70
2.9 数据管理模块	71
2.9.1 界面设计	71
2.9.2 界面初始化	73
2.9.3 界面数据绑定	73
2.9.4 模块和数据的动态映射	76
2.9.5 数据层类的封装	82
2.10 人事论坛管理模块	89
2.10.1 区域管理模块	89
2.10.2 主题管理模块	95
2.10.3 回复管理模块	99
2.10.4 版主管理模块	103
2.11 扩充与提高	106
本章小结	109
第3章 企业网络办公自动化综合平台	110
3.1 系统总体设计	110
3.1.1 应用背景	110
3.1.2 项目目标	110

3.1.3 解决方案设计	112
3.1.4 系统模块功能分析	112
3.2 数据库设计	115
3.2.1 数据库需求分析	115
3.2.2 数据库视图设计	116
3.2.3 数据库表设计	116
3.3 技术细节	118
3.3.1 Web.config 文件	119
3.3.2 本系统 Web.config 文件	120
3.4 系统通用类和安全加密说明	121
3.4.1 系统通用类说明	121
3.4.2 全局通用静态函数说明	123
3.4.3 数据库连接字段加密	124
3.5 用户注册登录功能	125
3.5.1 用户注册	125
3.5.2 用户登录	129
3.6 通讯簿功能	132
3.6.1 界面设计	132
3.6.2 动态加载数据控件	135
3.6.3 加载个性平台数据	138
3.6.4 添加功能	142
3.6.5 修改功能	146
3.6.6 删除功能	151
3.6.7 多功能检索	151
3.7 邮件功能	157
3.8 亲友记录功能	161
3.8.1 界面设计	161
3.8.2 添加功能	164
3.8.3 修改功能	170
3.8.4 删除功能	171
3.9 日程安排功能	172
3.9.1 界面设计	172
3.9.2 添加日程	174
3.9.3 修改日程	176
3.9.4 删除日程	177
3.9.5 日程提醒	178
3.10 快捷网络平台功能	179
3.10.1 界面设计	179

3.10.2 添加平台	180
3.10.3 修改平台	182
3.10.4 删除平台	182
3.11 部分后台功能	183
3.11.1 新用户参数构建	184
3.11.2 动态创建菜单	184
3.12 扩充与提高	187
本章小结	188
第4章 企业资源维护综合管理系统	189
4.1 系统总体设计	189
4.1.1 项目目标	189
4.1.2 解决方案设计	190
4.1.3 系统模块功能分析	191
4.2 数据库设计	194
4.2.1 数据库总体设计	194
4.2.2 数据库视图设计	195
4.2.3 数据库具体的表	195
4.3 技术细节	199
4.4 用户管理功能	204
4.4.1 用户登录	204
4.4.2 用户注册	209
4.4.3 用户信息修改	212
4.5 角色管理功能	215
4.5.1 添加角色	216
4.5.2 修改角色	221
4.5.3 删除角色	224
4.6 维护类别维护	225
4.6.1 方向管理设计	226
4.6.2 频道管理设计	230
4.6.3 栏目管理设计	236
4.6.4 主题管理设计	242
4.7 维护功能设计	249
4.7.1 申请维护	250
4.7.2 维护反馈	254
4.7.3 查看维护	265
4.7.4 维护监控	271
4.8 扩充与提高	273
本章小结	273

第 5 章 企业信息文档综合管理系统	274
5.1 系统总体设计	274
5.1.1 应用背景	274
5.1.2 项目目标	274
5.1.3 解决方案设计	276
5.1.4 系统模块功能分析	277
5.2 数据库设计	278
5.2.1 数据库总体设计	278
5.2.2 数据库视图设计	278
5.2.3 数据库具体的表设计	279
5.3 技术细节	282
5.3.1 添加组件引用	282
5.3.2 树目录界面设计	282
5.3.3 树目录代码实现	283
5.4 页表管理设计	286
5.4.1 添加页表	288
5.4.2 修改页表	290
5.4.3 删除页表	292
5.4.4 页表排序	293
5.5 模块管理设计	295
5.5.1 添加模块	298
5.5.2 修改模块	299
5.5.3 删除模块	301
5.5.4 模块排序	302
5.6 栏目管理设计	305
5.6.1 添加栏目	308
5.6.2 修改栏目	310
5.6.3 删除栏目	312
5.6.4 栏目排序	314
5.7 内容管理设计	316
5.7.1 添加内容	323
5.7.2 修改内容	327
5.7.3 删除内容	328
5.8 用户权限管理设计	331
5.8.1 角色添加	333
5.8.2 角色修改	335
5.8.3 角色删除	338
5.8.4 角色权限分配	339

5.9 文件管理设计	346
5.9.1 文件上传	346
5.9.2 系统文件维护	351
5.10 前台显示设计	354
5.10.1 界面设计	354
5.10.2 显示实现	355
5.11 扩充与提高	358
本章小结	358
第6章 企业电子商务销售管理系统	359
6.1 系统总体设计	359
6.1.1 应用背景	359
6.1.2 项目目标	360
6.1.3 解决方案设计	362
6.1.4 系统设计分析	364
6.2 数据库设计	365
6.2.1 数据库总体设计	365
6.2.2 数据库具体的表	365
6.3 技术细节	369
6.3.1 调用 Windows API	370
6.3.2 .NET 中使用非安全代码	370
6.3.3 加密、解密函数构造	371
6.4 数据访问层接口设计	374
6.5 SQL 数据访问层设计	376
6.5.1 SQL 数据访问层关系设计	376
6.5.2 SQLHelper 类实现	376
6.5.3 客户数据访问层	380
6.5.4 产品库存数据访问层	387
6.5.5 产品子项数据访问层	389
6.5.6 订单数据访问层	390
6.5.7 产品数据访问层	395
6.5.8 产品界面数据访问层	397
6.6 Oracle 数据访问层设计	398
6.6.1 Oracle 数据访问层关系设计	398
6.6.2 OracleHelper 类实现	398
6.7 数据访问层 Factory 设计	402
6.7.1 数据访问层 Factory 设计	402
6.7.2 数据访问层 Factory 具体实现	403

6.8 业务逻辑层模板设计	404
6.8.1 业务逻辑层模板设计	404
6.8.2 业务逻辑层模板具体实现	405
6.9 业务逻辑层设计	407
6.9.1 客户处理设计	407
6.9.2 购物车处理设计	409
6.9.3 订单处理设计	411
6.9.4 产品处理设计	412
6.9.5 库存处理设计	413
6.9.6 产品广告处理设计	413
6.10 表面呈现层设计	414
6.10.1 网站主界面	414
6.10.2 用户注册和登录	415
6.10.3 搜索实现	421
6.10.4 购物车实现	423
6.10.5 订单实现	428
6.10.6 结账实现	433
6.10.7 Web Services	434
6.11 扩充与提高	434
本章小结	440
第7章 网络多彩留言板	441
7.1 系统总体设计	441
7.1.1 应用背景	441
7.1.2 项目目标	441
7.1.3 解决方案设计	442
7.1.4 系统模块功能分析	442
7.2 数据库设计	443
7.2.1 数据库总体设计	443
7.2.2 数据库视图设计	443
7.2.3 数据库具体表设计	443
7.3 技术细节	445
7.3.1 数据库的移植技术	445
7.3.2 数据访问层实现	446
7.4 编辑主窗口设计	451
7.4.1 编辑界面设计	451
7.4.2 系统参数初始化	451
7.4.3 窗口创建初始化	455
7.4.4 按钮创建	455



7.4.5 按钮事件初始化	457
7.4.6 编辑和预览状态转换	460
7.4.7 辅助事件实现	462
7.5 对话框设计	469
7.5.1 文件上载模块	469
7.5.2 表情对话框	471
7.5.3 图片上载对话框	471
7.5.4 文件上载对话框	474
7.5.5 添加表格对话框	476
7.5.6 字体颜色对话框	481
7.6 扩充与提高	484
本章小结	485
附录	486
附录 1 系统安装	486
附录 1.1 安装程序的制作	486
附录 1.2 安装应用程序	489
附录 1.3 删除或修复系统	490
附录 2 应用程序的发布	490
附录 2.1 配置应用程序 IIS	490
附录 2.2 安装数据库	491

第 1 章



ASP.NET 概述

1.1 ASP.NET 简介

1.1.1 ASP.NET 简介

ASP 是 Active Server Pages 的简写,它是 Microsoft 公司推出的一种服务器端动态网页编写脚本,在 Microsoft IIS (Internet information server) 或 PWS (personal web server) 等 Web 服务器上执行。

随着 ASP 的广泛使用,软件的开发人员发现“经典”的 ASP 模型中有许多问题,其中主要有:

1) 性能: ASP 语法分析器是一个不太可靠的软件,用于处理分配给它的几乎所有的事物,依赖活动脚本编程引擎执行脚本,所以性能比较低。

2) 可维护性: ASP 页面混合了 HTML 代码和服务端脚本,如 JavaScript、VBScript,所有的代码在整个文件中混合,十分杂乱。

3) 状态管理:在 ASP 中使用 Application 和 Session 对象来管理状态时,这些对象需要用户的 Web 浏览器打开 Cookie。

4) COM 组件的使用:使用 COM 组件存在配置、部署、性能、兼容等多方面的问题,特别是集成调试方面非常困难。

正因为原有的 ASP 模型存在许多的问题,而且与 Microsoft 的 .NET 战略架构有许多抵触之处,因此,Microsoft 推出了基于 .NET 平台的新一代 Web Forms 窗体和 XML Web Services 的开发平台 ASP.NET。

ASP.NET 不仅仅是新版本的 ASP:也是统一的 Web 开发平台,用来提供开发人员生成企业级 Web 应用程序所需的服务。ASP.NET 是一个已编译的、基于 .NET 环境,可以用任何与 .NET 兼容的语言(包括 Visual Basic .NET、C# 和 Jscript .NET)创作应用程序,而不仅仅是原来 ASP 采用的 JavaScript 和 VBScript。ASP.NET 可以无缝地与 WYSIWYG (what you see is what you get) HTML 编辑器和其他编程工具(包括 Microsoft Visual Studio .NET)一起工作。另外,任何 ASP.NET 应用程序都可以使用整个 .NET Framework。

ASP.NET 提供一种简单的模型,该模型使 Web 开发人员能够编写在应用程序级运行逻辑。开发人员可以在 Global.asax 文本文件中或在作为程序集部署的已编译类中编写这种代码。ASP.NET 提供易用的应用程序和会话状态功能,它们容易与所有其他 .NET

Framework API 兼容。

ASP.NET 利用 .NET Framework 和公共语言运行库 (CLR) 中的性能增强功能。所有 ASP.NET 代码都是可编译的, 而不是解释的, 这就允许对本机代码采用早期绑定强类型处理, 以及实时 (JIT) 编译。ASP.NET 可方便地分解, 即开发人员可以移除那些与自己开发的应用程序不相关的模块 (如会话模块)。ASP.NET 提供丰富的缓存服务 (包括内置服务和缓存 API 两种)。ASP.NET 还提供性能计数器和默认授权和验证方案。ASP.NET 配置设置存储在基于 XML 的文件中, 这些文件都是可读和可写的。每一个应用程序都可以有不同的配置文件, 可以扩展配置方案, 以适应用户的要求。当应用程序安装在同一台计算机上, 但使用不同的 .NET Framework 版本时, 应用程序是并行运行的。

1.1.2 ASP.NET 程序运行原理

由于 ASP.NET 程序只能在服务器端执行, 如图 1.1 所示, 但客户端浏览器向服务器请求打开 ASP.NET 程序或页面时, 服务器会从硬盘中读取该程序代码, 然后加以执行, 并将执行的结果转化为浏览器能识别的 HTML 文件, 并将文件发送给浏览器, 浏览器根据收到的 HTML 文件转换为网页画面呈现给访问者。

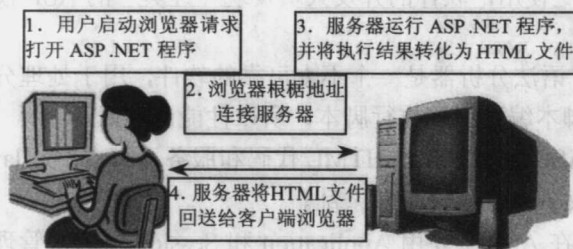


图 1.1 ASP.NET 程序客户端与浏览器交互过程

图 1.1 只是给出了 ASP.NET 程序客户端与浏览器的交互过程, 而究竟 ASP.NET 程序是怎样在服务器端被执行的呢? 其实在这其中有一个通过多层代码发送 HTTP 请求的调用链, 具体情况如图 1.2 所示。

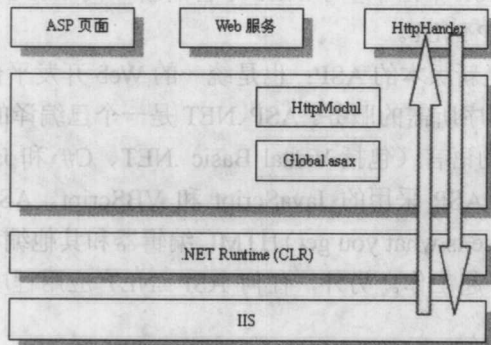


图 1.2 ASP.NET 执行过程中的调用链

但第一次执行 .aspx 文件时, 必须对之进行编译, 也可被托管代码在 ASP.NET 运行

环境内运行。在.aspx 文件完成编译后, HTTP 请求将通过“管道”发送到相应的 HTTP 处理器(Handler)。“管道”包含了应 HTTP 请求执行不同功能的多个 HTTP 模块(HttpModule)以及包括一些登录、认证、授权及会话状态信息(Global.asax), 然后经 CLR 做一些调度、处理, 例如: 分离 SOAP 访问。最后经 IIS 执行, 完成后再经相反的方向返回到 Handler。

1.2 .NET Framework 简介

1.2.1 .NET Framework 概述

ASP.NET 仅仅是 .NET Framework 中的一部分。 .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 具有两个主要组件: 公共语言运行库(CLR)和 .NET Framework 基础类库。

公共语言运行库是 .NET Framework 的核心, 它为 .NET Framework 提供核心服务(如内存管理、线程管理和远程处理), 而且还强制使用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。其中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。 .NET Framework 的另一个主要组件是 .NET Framework 基础类库, 它是一个综合性的面向对象的 reusable 类型集合, 可以使用它开发多种应用程序, 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 Framework 可由非托管组件承载, 这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行, 从而创建一个可以同时利用托管和非托管功能的软件环境。 ASP.NET 承载运行库为托管代码提供可伸缩的服务器端环境。 ASP.NET 直接使用运行库以启用 ASP.NET 应用程序和 XML Web Services。使用 Internet Explorer 承载运行库使用户能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码成为可能, 但是它具有只有托管代码才能提供的重大改进(如不完全受信任的执行和安全的独立文件存储)。 .NET Framework 中的 CLR 和类库以及应用程序和系统之间的关系, 如图 1.3 所示。

1.2.2 公共语言运行库

公共语言运行库(CLR)为 .NET 应用程序提供运行时的执行环境。它为托管代码提供管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。

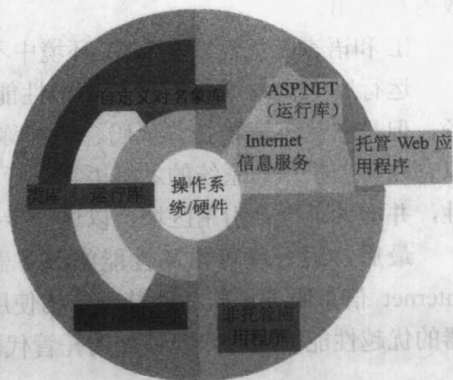


图 1.3 .NET Framework、应用程序和系统之间的关系图

无论在 CLR 中的托管代码是用何种语言编写的,实际上是一样的。每种语言都在.NET 环境下被编译成与平台无关的 Microsoft 中间语言。在.NET 运行时内部执行的就是这种中间语言。

运行库还通过实现通用类型系统(CTS)的严格类型验证和代码验证基础结构来加强代码可靠性。CTS 确保所有托管代码都是可以自我描述的。各种 Microsoft 和第三方语言编译器生成符合 CTS 的托管代码,这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。此外,运行库的托管环境还消除了许多常见的软件问题。例如,运行库自动处理对象布局并管理对象的引用,在不再使用它们时将它们释放。这种自动内存管理解决了两个最常见的应用程序错误:内存泄漏和无效内存引用。

IL 和语言无关性,在.NET 环境中不同编程语言可以无缝地共享代码和数据类型。

运行库大大地增强应用程序的性能。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。

最后,运行库可由高性能的服务器端应用程序(如 Microsoft® SQL Server™ 和 Internet 信息服务承载。此基础结构使用户在享受支持运行库宿主的行业最佳企业服务器的优越性能的同时,能够使用托管代码编写业务逻辑。

1.2.3 .NET Framework 基础类库

基类库(base class library)是可以由其他类继承和扩展或直接使用的最基本的类集。.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类库集合。该类库是面向对象的,并提供你自己的托管代码,从中导出功能的类型。这不但使.NET Framework 类型易于使用,而且还减少了学习.NET Framework 的新功能所需要的时间。此外,第三方组件可与.NET Framework 中的类无缝集成。

正如用户对面向对象的类库所希望的那样,.NET Framework 类型使用户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型。例如可使用.NET Framework 开发类型的应用程序和服务如下:

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

1.2.4 服务器应用程序开发

在托管领域中,服务器端应用程序是通过运行库宿主实现的。非托管应用程序承载公共语言运行库,后者使用户的自定义托管代码可以控制服务器的行为。此模型在获得主服务器的性能和可伸缩性的同时提供给用户公共语言运行库和类库的所有功能。

在不同服务器环境中运行托管代码的基本网络架构如图 1.4 所示。

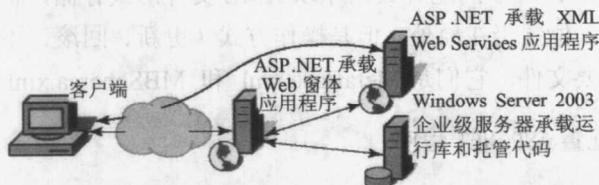


图 1.4 服务器端托管代码

ASP.NET 是使开发人员能够使用 .NET Framework 开发基于 Web 的应用程序的宿主环境。ASP.NET 可以不只是一个运行库宿主,也是使用托管代码开发 Web 站点和通过 Internet 分布的对象的完整结构。Web 窗体和 XML Web Services 都将 IIS 和 ASP.NET 用作应用程序的发布机制,并且两者在 .NET Framework 中都具有支持类集合。

XML Web Services 作为基于 Web 的技术的重要发展,是类似于常见 Web 站点的分布式服务器端应用程序组件。但是,与基于 Web 的应用程序不同,XML Web Services 组件不具有 UI 并且不以浏览器为目标。XML Web Services 由旨在供其他应用程序使用的可重用的软件组件组成,所谓的其他应用程序包括传统的客户端应用程序和基于 Web 的应用程序,甚至是其他 XML Web Services。因此,XML Web Services 技术正迅速地将应用程序开发和部署推向高度分布式 Internet 环境。

.NET Framework 还提供类和工具的集合来帮助开发和使用 XML Web Services 应用程序。XML Web Services 是基于 SOAP (一种远程过程调用协议)、XML (一种可扩展的数据格式) 和 WSDL (Web 服务描述语言) 这些标准生成的。基于这些标准生成 .NET Framework 的目的是为了提高与非 Microsoft 解决方案的互操作性。

1.3 IIS 简介

1.3.1 IIS 简介

IIS 是 Internet Information Server 的缩写,它是微软公司主推的 Web 服务器,Windows XP 里面包含的 IIS 5.0、IIS 与 Window NT Server 完全集成在一起,因而用户能够利用 Windows NT Server 和 NTFS (NT File System, NT 的文件系统) 内置的安全特性,建立强大、灵活而安全的 Internet 和 Intranet 站点。

现在最新的 IIS 是 6.0 版,它是以 Windows Server 2003 家族为平台的,其中有了一些新的功能改进,主要有:

- 1) 核心功能和服务: 对 IIS 6.0 进行了重新设计,以便利用基本 Windows 内核