

1+1 数据库混合开发技术丛书

ASP.NET + SQL Server

数据库开发与实例

吴晨 张亮 张静 编著



本书数据库开发案例

- ☐ 企业信息管理系统
- ☐ 员工绩效考核系统
- ☐ 校友录系统
- ☐ 学生管理系统
- ☐ 网络商城系统
- ☐ 网络论坛系统
- ☐ 旅行社管理系统
- ☐ 商务代表团管理系统



清华大学出版社

1+1 数据库混合开发技术丛书

ASP.NET + SQL Server

数据库开发与实例

吴晨 张亮 张静 编著

清华大学出版社

北 京

内 容 简 介

.NET 平台是 Microsoft 公司推出的一套开发框架和工具, ASP.NET 依托 .NET 平台的强大功能, 可以使用户轻松地实现 Web 应用程序的开发。SQL Server 作为数据库应用开发的利器, 被广大的数据库开发者所钟爱, 是 Web 应用程序开发最常用的后台数据库。

本书前 3 章主要对 ASP.NET 应用程序的基础知识以及 SQL Server 数据库应用高级技术分别进行说明和解析; 第 4 章~第 11 章通过 8 个经典的应用程序开发项目, 系统全面地介绍了使用 ASP.NET 和 SQL Server 开发数据库应用程序的精髓; 第 12 章总结性地介绍了 ASP.NET 数据库系统性能优化和安全知识。本书选材新颖, 实例涵盖范围广, 具有一定的实用价值, 通过本书的学习, 可以使读者更好地掌握 ASP.NET 和 SQL Server 开发数据库应用程序技术, 并将这些技术应用到实际的软件开发工程中。

本书适用于具有一定 Web 应用程序开发基础、准备学习 ASP.NET 编程或希望提高 ASP.NET 编程技巧的读者。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

ASP.NET+SQL Server 数据库开发与实例/吴晨, 张亮, 张静 编著. —北京: 清华大学出版社, 2006.8

(1+1 数据库混合开发技术丛书)

ISBN 7-302-13407-3

I. A… II. ①吴… ②张… ③张… III. ①主页制作—程序设计 ②关系数据库—数据库管理系统, SQL Server IV. ①TP393.092 ②TP311.138

中国版本图书馆 CIP 数据核字(2006)第 078056 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 王 定 文稿编辑: 鲍 芳

封面设计: 久久度文化 版式设计: 康 博

印 刷 者: 北京市清华园胶印厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 195×260 印张: 26.25 字数: 638 千字

版 次: 2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

书 号: ISBN 7-302-13407-3/TP·8420

印 数: 1~5000

定 价: 48.00 元(含光盘)

前言

ASP.NET 是由 Microsoft 推出的、基于 .NET 框架的新一代 Web 应用程序开发架构。其整体架构分为表示层、应用层和数据层，有效地提高了 Web 应用程序开发的效率。

SQL Server 2000 是 Microsoft 公司推出的关系数据库管理系统，它提供了功能全面、操作简单的图形化界面。因为在 ASP.NET 应用中提供了专用于 SQL Server 数据库应用系统的类库，可以有效地提高数据访问的效率，降低应用程序开发难度，因此，在 ASP.NET 开发中，SQL Server 成为了首选的数据库后台管理系统。

本书共分 12 章，第 1 章主要对 ASP.NET 基础知识做了具体介绍，包括 ASP.NET 运行机制、开发环境的配置、命名空间的应用、应用程序的配置以及应用程序的调试和发布，使读者对 ASP.NET 系统开发与配置有一个系统的了解。

第 2 章主要对 SQL Server 数据库的管理应用做了具体介绍，包括 SQL Server 数据库的管理、数据表的管理、生成和维护数据库对象、SQL 语言应用，使读者对 SQL Server 数据库应用有一个系统的了解。

第 3 章主要对 ADO.NET 的开发知识做了具体介绍，包括 ADO.NET 访问数据库的方法、ADO.NET 对象模型、ADO.NET 对数据库的操作、数据访问控件和数据绑定技术，使读者对使用 C#.NET 操作数据库的方法有一个详细的了解。

第 4~11 章，按照软件工程的应用步骤阐述了数据库系统的开发过程和技巧。各个数据库应用案例如下。

- 企业信息管理系统
- 员工绩效考核系统
- 校友录系统
- 学生管理系统
- 网络商城系统
- 网络论坛系统
- 旅行社管理系统
- 商务代表团管理系统

在应用系统的讲述过程中，首先阐述了系统需求分析、系统功能的描述、功能模块划分和系统流程分析，然后讲述了数据库分析设计、表的创建、表关系的创建和存储过程的创建，接下来对系统公共模块创建进行了介绍，最后讲解了系统中界面的设计和代码的实现方法。其中代码的实现部分重点介绍的是 C#.NET 的后台应用程序的使用，并添加了详细的说明，读者可以通过这些说明，了解 C#.NET 的应用方法。

第 12 章主要对 ASP.NET 系统的优化和安全性问题进行了介绍，包括 ASP.NET 应用程序在实际开发中注意的问题，系统性能优化，数据库的安全和优化，ASP.NET 应用程序的安全问题和 Web 应用程序的部署。通过对这些知识的学习，读者可以在应用系统开发时提高系统的安全性，并提高整个系统的运行效果。

本书中的所有界面和代码都通过作者调试和验证，读者可以参考具体系统进行学习。

本书由吴晨、张亮、张静编著，此外，参与本书编辑和修改的还有尹尚书、唐元新、何玉珠、戴永乐、郑立亮、祁强、杨帆、甄玉杰、孙守仁、张勇、费立红等同志，在此，编者对他们表示衷心的感谢。

由于本书编写的时间比较仓促，其不足之处在所难免，希望广大读者批评指正。

编 者

目 录

第 1 章 ASP.NET 的开发	1	2.3.2 主键的设置	28
1.1 ASP.NET 开发环境的配置	1	2.3.3 设置列值自动增加	29
1.1.1 IIS 的配置	1	2.3.4 查看和编辑数据	30
1.1.2 系统的配置	3	2.3.5 关系图的创建	30
1.2 ASP.NET 的命名空间	3	2.4 生成和维护数据库对象	32
1.2.1 System.Web 命名空间	3	2.4.1 视图	32
1.2.2 System.Web.UI 命名空间	3	2.4.2 触发器	34
1.2.3 System.Data 命名空间	4	2.4.3 存储过程	36
1.2.4 System.Net 命名空间	4	2.5 SQL 语言	36
1.2.5 System.Drawing 命名空间	4	2.5.1 数据检索	37
1.3 ASP.NET 应用程序配置	4	2.5.2 数据操作	37
1.3.1 Global.asax 的配置	5	2.5.3 存储过程	38
1.3.2 Web.config 的配置	6	2.5.4 E-R 图设计	38
1.4 调试和发布 Web Application	10	第 3 章 ADO.NET 开发	40
1.4.1 调试 ASP.NET 应用程序	10	3.1 ASP.NET 与 ADO.NET 的关系	40
1.4.2 Web 系统的发布	10	3.2 ADO.NET 访问数据库	40
第 2 章 SQL Server 2000 数据库管理	11	3.2.1 访问 SQL Server 数据库	41
2.1 SQL Server 2000 主要功能	11	3.2.2 访问 ODBC 数据源	41
2.2 SQL Server 的管理	11	3.3 ADO.NET 对象模型	43
2.2.1 生成和管理数据库	12	3.3.1 SqlConnection 对象	43
2.2.2 安全管理	18	3.3.2 SqlCommand 对象	44
2.2.3 数据库的备份与恢复	20	3.3.3 SqlDataReader 对象	45
2.2.4 数据转换服务	23	3.3.4 SqlDataAdapter 对象	46
2.3 数据表管理	27	3.3.5 DataSet 对象	47
2.3.1 表对象的生成和维护	27	3.3.6 DataTable 对象	48

3.4 使用 ADO.NET 操作数据库.....48	
3.4.1 使用 ExecuteReader() 操作数据库.....48	
3.4.2 使用 ExecuteNonQuery() 操作数据库.....49	
3.4.3 使用 ExecuteScalar() 操作数据库.....50	
3.4.4 使用 ExecuteXmlReader() 操作数据库.....50	
3.5 数据访问服务器控件.....51	
3.5.1 Repeater 控件.....51	
3.5.2 DataList 控件.....52	
3.5.3 DataGrid 控件.....53	
3.6 ADO.NET 数据绑定.....56	
3.6.1 DataSource 属性.....56	
3.6.2 数据绑定和 Item 集合的创建.....57	
3.6.3 Style 属性.....57	
3.6.4 Template 模板.....57	
第 4 章 企业信息管理系统.....58	
4.1 系统设计.....58	
4.1.1 需求分析.....58	
4.1.2 系统功能描述.....59	
4.1.3 功能模块划分.....60	
4.1.4 系统流程分析.....61	
4.2 数据库设计.....63	
4.2.1 数据库分析和设计.....63	
4.2.2 数据表的创建.....65	
4.2.3 数据表关系图.....67	
4.2.4 创建存储过程.....67	
4.3 系统公用模块创建.....74	
4.3.1 配置 web.config.....74	
4.3.2 创建 CSS.....75	
4.4 系统界面和代码的实现.....77	
4.4.1 企业信息管理系统首页.....77	
4.4.2 用户信息添加界面.....79	
4.4.3 用户信息列表界面.....82	
4.4.4 客户信息添加界面.....86	
4.4.5 客户信息列表界面.....88	
4.4.6 产品信息编辑界面.....91	
4.4.7 添加合同信息界面.....93	
4.4.8 销售信息编辑界面.....95	
4.4.9 销售信息添加界面.....98	
4.4.10 产品销售统计界面.....100	
4.5 实例演示.....101	
第 5 章 员工绩效考核系统.....105	
5.1 系统设计.....105	
5.1.1 需求分析.....105	
5.1.2 系统功能描述.....106	
5.1.3 功能模块划分.....107	
5.1.4 系统流程分析.....107	
5.2 数据库设计.....110	
5.2.1 数据库分析和设计.....110	
5.2.2 数据表的创建.....111	
5.2.3 数据表关系图.....113	
5.2.4 创建存储过程.....113	
5.3 系统公用模块创建.....117	
5.4 系统界面和代码的实现.....118	
5.4.1 员工信息添加界面.....118	
5.4.2 项目信息添加界面.....121	
5.4.3 项目信息列表界面.....125	
5.4.4 日志信息的添加.....129	
5.4.5 用户查询个人日志信息.....132	
5.4.6 领导查询日志信息.....135	
5.4.7 密码修改界面.....137	
5.5 实例演示.....139	
第 6 章 校友录系统.....142	
6.1 系统设计.....142	
6.1.1 需求分析.....142	
6.1.2 系统功能描述.....143	
6.1.3 功能模块划分.....143	
6.1.4 系统流程分析.....144	

6.2	数据库设计	147	7.4.6	学生信息维护界面	229
6.2.1	数据库分析和设计	148	7.5	实例演示	236
6.2.2	数据表的创建	150	第 8 章	网络商城系统	239
6.2.3	数据表关系图	153	8.1	系统设计	239
6.3	系统公用模块创建	153	8.1.1	需求分析	239
6.3.1	配置 web.config	153	8.1.2	系统功能描述	240
6.3.2	创建 CSS	154	8.1.3	功能模块划分	240
6.4	系统界面和代码的实现	154	8.1.4	系统流程分析	241
6.4.1	注册班级首页	154	8.2	数据库设计	242
6.4.2	查询学校区域界面	159	8.2.1	数据库分析和设计	242
6.4.3	注册学校信息界面	162	8.2.2	数据表的创建	244
6.4.4	班级信息添加界面	166	8.2.3	数据表关系图	246
6.4.5	班级通讯录界面	171	8.2.4	创建存储过程	247
6.4.6	留言信息界面	174	8.3	系统公用模块创建	256
6.4.7	留言信息固顶	177	8.3.1	配置 web.config	257
6.4.8	取消固顶界面	180	8.3.2	创建 Type.ascx	257
6.4.9	编辑留言信息界面	181	8.3.3	创建 likeware.ascx	258
6.4.10	删除信息界面	183	8.3.4	创建 IStoreDBO.cs	258
6.5	实例演示	184	8.4	系统界面和代码的实现	275
第 7 章	学生管理系统	189	8.4.1	分类商品信息界面	275
7.1	系统设计	189	8.4.2	商品详细信息界面	276
7.1.1	需求分析	189	8.4.3	搜索界面	278
7.1.2	系统功能描述	190	8.4.4	购物车界面	278
7.1.3	功能模块划分	191	8.4.5	提交订单界面	281
7.1.4	系统流程分析	192	8.4.6	订单信息界面	283
7.2	数据库设计	194	8.4.7	订单详细信息界面	284
7.2.1	数据库分析和设计	194	8.5	实例演示	285
7.2.2	数据表的创建	195	第 9 章	网络论坛系统	290
7.2.3	数据表关系图	196	9.1	系统设计	290
7.3	系统公用模块创建	197	9.1.1	需求分析	290
7.3.1	配置 web.config	197	9.1.2	系统功能描述	290
7.3.2	stu.cs 类文件	197	9.1.3	功能模块划分	291
7.4	系统界面和代码的实现	205	9.1.4	系统流程分析	292
7.4.1	学生管理系统首页	205	9.2	数据库设计	294
7.4.2	学生详细信息界面	216	9.2.1	数据库分析和设计	294
7.4.3	班级信息管理界面	219	9.2.2	数据表的创建	296
7.4.4	班级信息维护界面	220	9.2.3	数据表关系图	298
7.4.5	学生信息管理界面	225			

9.2.4 创建存储过程.....	298	第 11 章 商务代表团管理系统	367
9.3 系统公用模块创建	302	11.1 系统设计	367
9.3.1 配置 web.config	302	11.1.1 需求分析	367
9.3.2 创建 SqlClass.cs	303	11.1.2 系统功能描述	368
9.4 系统界面和代码的实现	308	11.1.3 功能模块划分	368
9.4.1 系统首页界面	309	11.1.4 系统流程分析	369
9.4.2 浏览帖子界面	311	11.2 数据库设计	371
9.4.3 发布新帖界面	316	11.2.1 数据库分析和设计	371
9.4.4 版主管理帖子	318	11.2.2 数据表的创建	372
9.4.5 添加讨论区界面	321	11.2.3 数据表关系图	374
9.4.6 添加版块界面	323	11.2.4 创建存储过程	375
9.4.7 编辑版块信息界面	324	11.3 系统公用模块创建	380
9.4.8 用户资料维护界面	328	11.4 系统界面和代码的实现	381
9.4.9 管理员维护用户资料界面	331	11.4.1 系统首页	381
9.5 实例演示	332	11.4.2 参展商信息界面	382
第 10 章 旅行社管理系统	336	11.4.3 住宿信息界面	384
10.1 系统设计	336	11.4.4 会展活动信息界面	386
10.1.1 需求分析	336	11.4.5 参展商管理界面	387
10.1.2 系统功能描述	337	11.4.6 参展商信息添加界面	389
10.1.3 功能模块划分	338	11.4.7 参展商信息维护界面	391
10.1.4 系统流程分析	338	11.5 实例演示	393
10.2 数据库设计	340	第 12 章 系统的优化和安全	396
10.2.1 数据库分析和设计	341	12.1 ASP.NET 程序开发	
10.2.2 数据表的创建	342	中应注意的问题	396
10.2.3 数据表关系图	343	12.2 系统性能的优化	397
10.2.4 创建存储过程	344	12.2.1 系统性能概述	397
10.3 系统公用模块创建	349	12.2.2 性能优化	398
10.4 系统界面和代码的实现	350	12.3 数据库的安全与优化	398
10.4.1 景点列表界面	350	12.3.1 安全	398
10.4.2 景点详细信息界面	351	12.3.2 性能优化	399
10.4.3 行程信息列表	353	12.4 ASP.NET 应用程序的安全性	400
10.4.4 新闻列表信息	354	12.4.1 ASP.NET 身份验证体系	400
10.4.5 新闻详细信息	356	12.4.2 IP 地址的安全控制	402
10.4.6 旅行订单	357	12.4.3 表单验证	403
10.4.7 添加景点信息	359	12.4.4 Windows 验证	405
10.4.8 景点信息维护	361	12.4.5 安全性验证实例	406
10.5 实例演示	365	12.5 Web 应用程序部署	411

ASP.NET 的开发

ASP.NET 是 Microsoft 公司推出的基于通用语言的编程框架，它提供了一种崭新的网络编程模型，使用它可以在服务器端架构强大的网络程序。本章将主要介绍 ASP.NET 开发的基础应用知识。

- ASP.NET 的运行机制
- ASP.NET 开发环境配置
- ASP.NET 的命名空间
- ASP.NET 应用程序配置
- 调试和发布 Web Application



1.1 ASP.NET 开发环境的配置

本节将对 ASP.NET 运行平台中的 Internet 信息服务和系统要求配置的内容进行讲解。下面分别介绍各部分的内容。

1.1.1 IIS 的配置

IIS 服务可以实现 ASP.NET 的运行环境，使用户能够运行和调试程序。IIS 的安装这里就不再介绍，下面主要介绍 IIS 的配置。通过配置 IIS 可以使应用程序更加安全、更加快速。在 Windows 2000 Server 中配置 IIS 的具体步骤如下。

(1) 在“控制面板”中双击“管理工具”打开“管理工具”窗口，双击“Internet 服务管理器”打开“Internet 服务管理器”控制台，在“默认 Web 站点”节点上右击，在弹出的快捷菜单中选择“属性”命令，打开“默认 Web 站点属性”对话框，如图 1-1 所示。

(2) 选择“Web 站点”选项卡，在“IP 地址”下拉列表框中，默认为“(全部未分配)”选项，也可以选择其他 IP 地址。“TCP 端口”文本框中默认的端口设置为 80。选择“限制到”单选按钮，在后面的文本框中设置此站点允许连接人数，可以根据计算机的承载量设置最大的

连接人数。在“连接超时”文本框中，可设置站点的连接超时时间，默认时间为 900 秒。

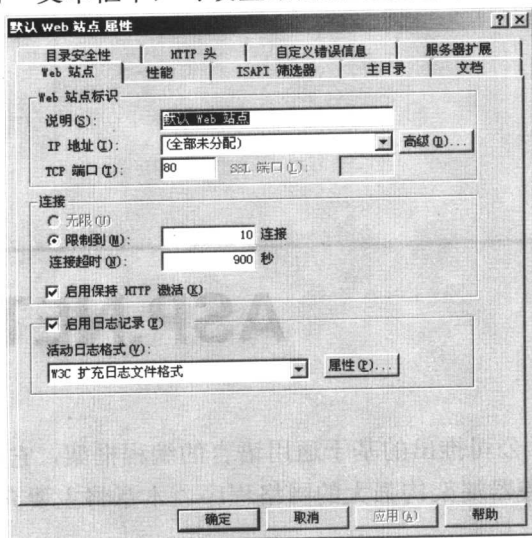


图 1-1 Web 站点选项卡

(3) 选择“主目录”选项卡，如图 1-2 所示。选中“此计算机上的目录”单选按钮，默认的 Web 站点存放文件的路径为 c:\inetpub\wwwroot。对此目录的操作权限包括脚本资源访问、读取、写入、目录浏览、日志访问、索引此资源。默认的权限只有“读取”、“日志访问”、“索引此资源”3 项，可以根据需要选择其他选项。

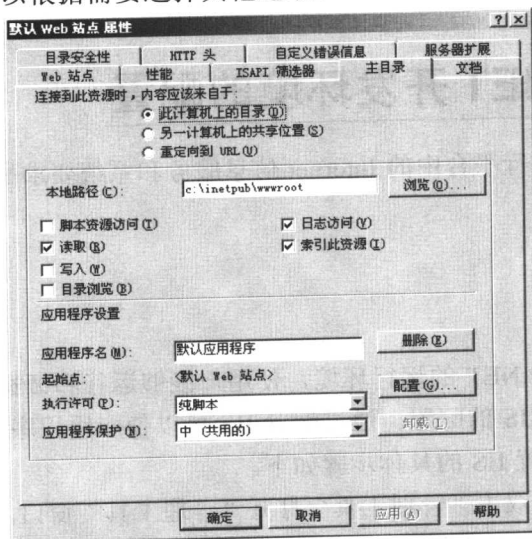


图 1-2 “主目录”选项卡

(4) 选择“文档”选项卡，如图 1-3 所示。选中“启用默认文档”复选框后，再次运行 Web 程序时，IIS 服务会自动读取默认文档。单击“添加”按钮，可增加新的默认文档，或者单击“删除”按钮，删除一个默认文档。

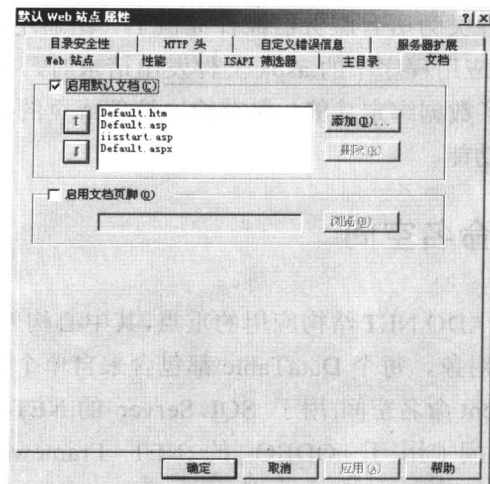


图 1-3 “文档”选项卡

1.1.2 系统的配置

ASP.NET 开发运行环境的基本要求为：具有 Service Pack 2 的 Windows 2000 Server 或 Advanced Server、Windows XP Professional 或 64 位版本，Windows Server 2003 系列产品之一，以及 MDAC 2.7 for Data 和 Internet 信息服务。

1.2 ASP.NET 的命名空间

ASP.NET 的命名空间中包含了大量的系统方法和类，.NET 框架就是使用命名空间来组织和交流程序代码的。可以说命名空间是 ASP.NET 编程中必不可少的部分，本节将介绍 ASP.NET 中几个重要的命名空间。

1.2.1 System.Web 命名空间

System.Web 命名空间是 ASP.NET 应用的基础命名空间，提供了浏览器与服务器通信的类和接口。此命名空间提供了 HTTP 请求的 `HttpRequest` 类、管理对客户端的 HTTP 输出的 `HttpResponse` 类，以及提供对服务器端实用工具与进程的访问的 `HttpServerUtility` 类。此命名空间还包括用于 Cookie 操作、文件传输、异常信息和输出缓存控制的类。

1.2.2 System.Web.UI 命名空间

System.Web.UI 命名空间提供了 Web 应用程序中的 ASP.NET 服务器控件以及页的类和接

口。该命名空间所包含的控件类为所有服务器控件提供了一组通用功能。`System.Web.UI` 命名空间还包含页类, 当向 Web 应用程序中的 .aspx 文件发出请求时, 都会自动生成页类。控件类和页类都可以继承, 并提供了数据绑定功能、保存给定控件或页的视图状态的能力, 以及对可编程控件和文本控件的分析功能。

1.2.3 System.Data 命名空间

`System.Data` 命名空间是 ADO.NET 结构应用的重点, 其中心构件是 `DataSet` 类。每个 `DataSet` 都可以包含多个 `DataTable` 对象, 每个 `DataTable` 都包含来自单个数据源的数据。

使用 `System.Data.SqlClient` 命名空间(用于 SQL Server 的 .NET Framework 数据提供程序)、`System.Data.Odbc` 命名空间(用于 ODBC 的 .NET Framework 数据提供程序)或 `System.Data.OleDb` 命名空间(用于 OLE DB 的 .NET Framework 数据提供程序)可访问要与 `DataSet` 结合使用的数据源。每个 .NET 数据提供程序都有对应的 `DataAdapter`, 可以将它用作数据源和 `DataSet` 之间的桥接器。

1.2.4 System.Net 命名空间

`System.Net` 命名空间为多种网络协议使用提供了简单的编程接口。此命名空间的 `WebRequest` 和 `WebResponse` 类实现了可插接式协议的基础。可插接式协议是网络服务的一种实现, 它能够开发出使用 Internet 资源的应用程序, 而不必考虑各种不同协议的具体细节。

1.2.5 System.Drawing 命名空间

`System.Drawing` 命名空间主要提供了对 GDI+ 基本图形功能的访问, 是应用图形类的基础。在 `System.Drawing.Drawing2D`、`System.Drawing.Imaging` 以及 `System.Drawing.Text` 命名空间中提供了更高级的功能。

在 `System.Drawing` 命名空间中的 `Graphics` 类提供了绘制到显示设备的方法。而 `Rectangle` 和 `Point` 等类可封装 GDI+ 基元。应用 `Pen` 类可用于绘制直线和曲线, 应用 `Brush` 派生出的类则用于填充形状的内部。

1.3 ASP.NET 应用程序配置

在 ASP.NET 应用程序中包括 `Global.asax` 和 `Web.Config` 两个文件, 它们是应用程序的配置文件, 下面分别介绍这两个配置文件的应用。

1.3.1 Global.asax 的配置

Global.asax 文件可以处理程序级别的事件。在 Global.asax 文件中使用 object 标签定义静态对象、.NET Framework 类以及 COM 组件。它们的作用范围可以为 appinstance、session 或者 application。appinstance 表示此对象只能在一个 HttpApplication 范围内使用。定义的格式如下：

```
<object id="id" runat="server" class=".NET Framework class Name" scope="appinstance"/>
<object id="id" runat="server" progid="COM ProgID" scope="session"/>
<object id="id" runat="server" classid="COM ClassID" scope="application"/>
```

Application_BeginRequest 方法负责处理 Portal 应用程序的每个 web 请求。下面的方法中使用请求的 querystring 对象获取 tabIndex 和 TabID 两个变量，并同时获取必需的配置文件来响应请求。Portal 的配置存储在应用程序的"Context"对象中，该对象在整个单一请求过程中对所有页面、控件和组件都有用。

```
void Application_BeginRequest(Object sender, EventArgs e) {
    int tabIndex = 0;
    int tabId = 0;
    //从 querystring 获取 TabIndex
    if (Request.Params["tabindex"] != null) {
        tabIndex = Int32.Parse(Request.Params["tabindex"]);
    }
    //从 querystring 获取 TabID
    if (Request.Params["tabid"] != null) {
        tabId = Int32.Parse(Request.Params["tabid"]);
    }
    Context.Items.Add("PortalSettings", new PortalSettings(tabIndex, tabId));
}
```

下面的程序中定义了 Application_AuthenticateRequest 事件，此事件用于验证用户客户端角色，并赋予相对应的管理权限。

```
void Application_AuthenticateRequest(Object sender, EventArgs e) {
    if (Request.IsAuthenticated == true) {
        String[] roles;
        //如果 session 不存在，则生成角色的 cookie
        if ((Request.Cookies["portalroles"] == null) || (Request.Cookies["portalroles"].Value == "")) {
            //从 UserRoles 表获取角色，添加到 cookie
            UsersDB user = new UsersDB();
            roles = user.GetRoles(User.Identity.Name);
            //生成字符串来保存角色
        }
    }
}
```

```

String roleStr = "";
foreach (String role in roles) {
    roleStr += role;
    roleStr += ";";
}
FormsAuthenticationTicket ticket = new FormsAuthenticationTicket(
    1, // version
    Context.User.Identity.Name, // user name
    DateTime.Now, // issue time
    DateTime.Now.AddHours(1), // expires every hour
    false, // don't persist cookie
    roleStr // roles
);
String cookieStr = FormsAuthentication.Encrypt(ticket);

//将 cookie 发送到客户端
Response.Cookies["portalroles"].Value = cookieStr;
Response.Cookies["portalroles"].Path = "/";
Response.Cookies["portalroles"].Expires = DateTime.Now.AddMinutes(1);
}
else {
    //从角色 cookie 获取角色
    FormsAuthenticationTicket ticket =
        FormsAuthentication.Decrypt(Context.Request.Cookies["portalroles"].Value);
    //将角色的字符串属性转换为字符串数组
    ArrayList userRoles = new ArrayList();
    foreach (String role in ticket.UserData.Split( new char[] { ';' } )) {
        userRoles.Add(role);
    }
    roles = (String[]) userRoles.ToArray(typeof(String));
}
Context.User = new GenericPrincipal(Context.User.Identity, roles);
}
}

```

1.3.2 Web.config 的配置

ASP.NET 应用程序的配置信息都定义和保存在名为 **Web.config** 的配置文件中。此文件在创建一个 ASP.NET 应用程序时会由 Visual Studio.NET 开发工具自动创建，并存放在工程文件的根目录下。Web.config 是基于 XML 格式的文本文件，Web.config 文件的根为<configuration>标记，也就是说，所有的配置信息都包含在<configuration>和</configuration>标记之间。

Web.config 文件的配置分为两个部分：配置节处理程序声明部分和配置节设置部分。配置节处理程序声明一般位于配置文件顶部的<configSections>和</configSections> 标记之间。每个声明都包含在一个<section>标记中，它们被用来指定提供特定配置数据集的节的名称和处理该节中配置数据的.NET 框架类的名称。

而配置节设置区域一般位于<configSections>标记后，位于<system.Web>节中。定义了应用程序的所有标记，并规范和管理 ASP.NET 运行时的行为。

在 Web.config 文件中定义了很多配置节处理程序声明和配置节处理程序，下面将主要介绍这些配置节。

1. <configuration>节

所有 Web.config 的元素都包含在<configuration>标记内，在它内部封装了其他的所有配置节。

```
<configuration>
.....
</configuration>
```

2. <appSettings>节

在<appSettings>节中可以定义应用程序设置项，反映了此节的可扩展性。<appSettings>节语法如下：

```
<appSettings>
  <add key="[key]" Value="[value]"/>
</appSettings>
```

<appSettings>节主要有两个属性定义：key 和 Value。

- key：指定该设置项的名称，便于在程序中引用。
- Value：指定该设置项的值。

在下面的<appSettings>节中定义了数据库连接的设置，server 定义了连接数据库的 IP 地址，localhost 代表本机，database 定义了数据库的名称，uid 为数据库用户的名称，sa 为系统默认的管理员，pwd 定义了数据库的登录密码，如果没有则设置为空。

```
<appSettings>
  <add key="ConnectionString" value="server=localhost;database=clientserver;uid=sa;pwd="/>
</appSettings>
```

3. <compilation>节

<compilation>节主要实现 ASP.NET 使用的编译设置，其语法结构如下：

```
<compilation
  defaultLanguage="c#"
```

```

    debug="true"
  />

```

<compilation>节中的两个参数的含义如下。

- **defaultLanguage**: 定义程序应用的后台代码语言, 可以选择 C# 或 Visual Basic.net 两种之一。
- **debug**: 其值为 true 时将启用 ASPX 调试; 其值为 false 时可以提高应用程序运行时的性能。一般来说, 在程序开发调试时设置为 debug="true"。而在其他时候则设为 false。

4. <customErrors>节

<customErrors>节可以完成在 ASP.NET 应用程序中自定义错误消息的功能, 其语法如下所示。

```

<customErrors
mode="RemoteOnly"
/>

```

mode 参数具有 On、Off 和 RemoteOnly 3 种状态, 含义如下所示。

- On 表示始终显示自定义的信息。
- Off 表示始终显示详细的 ASP.NET 错误信息。
- RemoteOnly 表示只对不在本地 Web 服务器上运行的用户显示自定义的信息。

5. <authentication>节

<authentication>节主要实现安全配置工作, 其语法结构如下。

```

<authentication mode="Windows" />
<!-- 授权
    此节设置应用程序的授权策略。可以允许或拒绝不同的用户或角色访问
    应用程序资源。通配符: "*" 表示任何人, "?" 表示匿名(未经身份验证的)用户。
-->
<authorization>
<allow users="*" /> <!-- 允许所有用户 -->
<!-- <allow users="[逗号分隔的用户列表]"
    roles="[逗号分隔的角色列表]" />
    <deny users="[逗号分隔的用户列表]"
    roles="[逗号分隔的角色列表]" />
-->
</authorization>

```

在<authentication>节用 mode 属性来控制应用程序的验证模式。该属性设定为 Windows 时表示使用 IIS 验证方式; 设定为 Forms 时表示使用基于表单的验证方式; 设定为 Passport 时表示采用 PassPort cookie 验证模式; 设定为 None 时表示不采用任何验证方式。