

本书素材和网页源文件
3个多小时的Dreamweaver CS5多媒体教程
HTML、CSS和ASP教程的CHM文件



Dreamweaver CS5 & ASP动态网页设计

朱印宏 编著

- 本书由浅入深，循序渐进地介绍了使用Dreamweaver CS5和ASP开发动态网站的方法和技巧。
- 精心设计18个经典实例，且在每个实例的基础上展开了一系列相关的技术要点并加以变通，使读者能够做到活学活用。
- 将复杂的动态网站分解成一个个模块，简单而清晰的知识结构不只是学习快捷，更可灵活应用这些模块构建自己的动态网站。

超值赠送DVD光盘，多媒体教程让您快速上手，CHM教程让您速查速用。



中国电力出版社
CHINA ELECTRIC POWER PRESS



ISBN 978-7-5123-2305-6



9 787512 323056 >

定价:65.00元(含1DVD)

上架建议: 计算机/网页设计

本书是《Dreamweaver CS4 & ASP 动态网页》一书的升级版，是在上一本图书良好的市场基础上，结合作者多年网站开发积累的宝贵经验，以及读者反馈，对全书进行了全面的优化和完善。

在本次升级中，主要进行了三方面的改进。

基础知识更系统、更扎实

升级前用了 5 章篇幅介绍 ASP 相关基础知识，升级后在章节上虽然没有大的变化，但内容得到了完全的更新和充实，由以前的泛泛介绍过渡到现在的系统讲解。基础部分的更新主要体现在下面几个特点。

第一，紧扣 ASP 动态网站开发的核心技术（如 VBScript、ASP、SQL 和 ADO）展开介绍，删除了 JavaScript、HTML 等知识章节，避免在知识讲解中试图面面俱到、结果什么都达不到的弊端。

第二，讲究取舍，绝对实用。在 VBScript、ASP、SQL 和 ADO 中，任何一种动态网站开发技术都可以够写一本书，其深度非一般读者所能及的，知识点如何取舍就非常有讲究。我们认为知识点的取舍首先要体现实用，然后适当扩展，以方便部分读者的再深入学习。同时还必须保证读者能够快速吸收、轻松理解。如何使概念的讲解更清晰，如何使疑点和难点的剖析更透彻，这些看起来很容易，但却是大部分图书的通病，也是困扰读者的心病。毫不妥协地面对基础技术中的所有疑点和难点，是本书有别于其他同类图书的最大亮点。

第三，语言更亲切，知识点讲解追求细致，帮助读者在轻松的学习中快速掌握基础知识。在基础知识讲解中我们重新完善并提升贯穿基础部分中的大量的示例代码的品质，使其更具代表性和典型性，力争使它们更能够为知识讲解服务，更能够解决读者对所学知识存在的疑团。

实例模块功能更完善、更实用

本次升级中，我们对每个综合实例的代码和结构都进行了重新设计和完善，增加了大量的功能，使实例更具实用性。例如，在用户注册模块中就增加了多种类型的验证方法，增加了邮件确认功能，增加了密码加密方法；在留言板模块中增加很多后台管理功能；在聊天室模块中增加了很多扩展功能，使聊天室更实用，如此等等。基本上每个实例模块的功能都被充实和更新，代码结构进行重新设计，以体现实用的原则。

实例学习更容易、更方便

在本书中，我们统一选用 VBScript 脚本语言作为服务器端语言进行介绍，删除了前一个版本中混合使用 VBScript 和 JavaScript 脚本语言进行开发，而给读者带来混乱和迷茫的弊病。我们还规范和统一了全书实例模块的文件和文件夹的名称和站点结构，避免给初学者在学习上制造不必要的障碍。

在数据库连接上，本书所有实例都统一采用自定义连接字符串的方法实现数据库连接，这样读者就不再为 DSN 而犯难了。直接复制本书光盘实例就可以在本地运行（当然前提是构建 ASP 服务器环境和定义动态站点），或者上传到远程服务器上进行预览。

读者普遍反映动态网站技术类图书所提供的实例都不能很好运行，可能这与动态技术的复杂性有关系。因此，我们提供给读者的实例都经过反复测试，绝对保证它们不需要修改设置就可以正常运行，这也符合很多读者在学习之前先运行实例看看效果的习惯。如果光盘实例都不能正常运行，自然也就没有兴趣去学习书中的实例。

本书特色

本书专门介绍了如何利用 Dreamweaver CS5 和 ASP 技术开发动态网站。作为完美网页设计专业图书，它具有以下三个特点。

- 内容全面。本书紧扣基础和实例两个基点，内容安排由浅入深、循序渐进，系统地介绍了用 Dreamweaver CS5 开发动态网站的各方面知识和技术。
- 可读性强。全书内容重点突出，详略得当，将错综复杂的概念梳理为井井有条的系统知识结构，帮助读者迅速掌握 Dreamweaver CS5 的操作技巧，以提高运用能力，并打下坚实的理论基础，为读者系统学习提供了全面、翔实知识，以便使用户从软件的庞大功能介绍中解脱出来。
- 面向应用。根据当前 ASP 动态网站开发中不同应用形式，精心设计了 18 个经典实例，每个实例展开了一系列相关的技术要点并加以变通，使读者能够做到活学活用。实例训练针对性强，技术要点全面而又精练。精心设计的思路和操作环节，力求以多种方式帮助读者理解各项技术要点，并方便对技术的检索。

阅读本书能学到什么

挑选本书作为学习 Dreamweaver CS5 设计和编写动态网站的工具，能够为读者提供下面技术帮助和知识技能。

- 了解浏览器/服务器架构的构成和原理。
- 架设 Web 应用程序和数据库服务器。
- 运用 ASP、VBScript、ADO 和 SQL 语言。
- 运用 Dreamweaver CS5 服务器行为技术构建动态网页。
- 借助 Dreamweaver CS5 辅助工具编写动态网页。

如何阅读本书

相信读者在购书之后，最迫切了解的是如何正确阅读本书。由于本书核心是介绍 ASP 动态网站开发，因此需要构建一个虚拟的服务器环境。在学习实例之前，读者应该正确完成下

面操作步骤。

- 第1步 应在个人计算机上搭建并设置好本地服务器环境，详细操作可参见本书第 2.4 节介绍。
- 第2步 定义两个同名文件夹，如 `mysite`，作为本地根文件和服务器文件夹。
本地文件夹最好放在非系统盘的根目录下，例如，系统盘为 C 盘，则可以把命名为 `mysite` 的本地文件夹放在 D 盘，即把 `D:\mysite` 作为本地文件夹。
如果读者在上一步中定义了 WWW 服务虚拟目录，则应把服务器端文件夹放置在虚拟目录中；否则应把服务器端文件夹放置在系统服务器 `wwwroot` 文件夹内，例如，在 XP 系统中的 `C:\Inetpub\wwwroot\mysite`。
- 第3步 定义好本地根文件夹和服务器端文件夹之后，就可以启动 Dreamweaver CS5 定义站点了，详细操作可参见第 2.5 节介绍。
- 第4步 打开本书附赠光盘，找到实例模板文件夹，把要学习的实例模板全部复制到本地根文件夹和服务器端文件夹中。
- 第5步：启动 Dreamweaver CS5，此时系统会自动读取本地文件夹中的所有文件，并在【文件】面板的列表框中显示这些模板文件。
- 第6步 参照第 5.4 小节介绍建立数据库链接，实例模板已经帮助你完成这个工作。
- 第7步 此时，读者就可以按照书中介绍的方法或实例操作步骤，在【文件】面板中打开某个文件进行学习了。

另外介绍两条学习本书最快捷有效的方法。

● 逆向学习法

先把本书附赠光盘中的实例效果文件夹中的最后效果实例复制到本地根文件夹和服务器端文件夹中，完成上面操作后运行。如果正确运行，说明环境配置和站点定义完全正确，否则，上面操作步骤中某个环节可能存在问题。

运行并欣赏实例，找到成功的自信，然后尝试一步步删除文件中某项功能，再自己动手把该功能恢复原貌。如果成功，可进一步操作，否则说明在操作设置或代码输入中出现了错误。这样操作目标明确，而且便于集中突破。

● 比较学习法

不妨定义两个站点，例如，一个为 `mysite`，在里边放置模板文件；另一个为 `result_site`，在里边放置实例效果文件。

学习时，同时打开两个站点的相同文件，如果出现错误，可以一步步比较，这样能够快速发现自己出错的地方和原因。

本书结构和技术要点

全书可分为两个部分：基础知识和经典实例。

- 第一部分为第 1~5 章，即动态网站技术基础部分。

本部分重点介绍动态网站各种技术基础，为初学者构建一个完整的动态网站技术背景，使读者在学习完前面 5 章后能够独立架设个人服务器，定义完整网站，并能够清楚动态网页制作的方法和过程。同时，本部分也为后面实例学习奠定基础，提供技术和知识参考。

- 第二阶段为第 6~24 章，即动态网站建设实例部分。

本部分共提供了 18 个实例，实例介绍由浅入深，由具体到抽象，从基础细节介绍逐步上升到探索动态网站流程设计，使读者既能站得住，又能望得远，避免一叶障目。这些实例所运用的知识和技术比较全面，每个实例所要介绍的技术要点各有侧重，力求覆盖所有动态网站技术。其中附录为电子版，读者可以在光盘中进行阅读。

关于光盘的内容

本书赠送 DVD 光盘，其中包含长约 5 小时 Dreamweaver 操作演示视频教程、本书用到的素材和网页源文件，还有 HTML、CSS 和 ASP 教程的 CHM 文件。

本书作者

本书由朱印宏编写。同时，参与资料整理等工作的还有常才英、袁祚寿、袁衍明、张敏、袁江、田明学、唐荣华、毛荣辉、卢敬孝、刘玉凤、李坤伟、旷晓军、陈万林、陈锐、钱佩林、苏敬波、冉东林、杨龙贵、张炜、王慧明、涂怀清、卢国才、苏恢定、司成向、胡体清、陈宗亮、徐清银、周秀成、颜昌学、王幼平、冉原洲、李经键、胡厚成等，在此对大家的辛勤工作表示衷心的感谢！

由于时间有限，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。有关本书的任何问题请发电子邮件到 css148@163.com。



Contents

目 录

Dreamweaver

前言

第 1 章 Dreamweaver CS5 快速上手.....1

- 1.1 Dreamweaver 工具概述.....1
- 1.2 Dreamweaver CS5 操作环境.....2
 - 1.2.1 标题栏.....3
 - 1.2.2 菜单栏.....3
 - 1.2.3 工具面板和工具栏.....4
 - 1.2.4 状态栏.....5
 - 1.2.5 属性面板.....5
 - 1.2.6 浮动面板.....6
- 1.3 Dreamweaver CS5 基本操作.....6
 - 1.3.1 输入文本和版式设计.....6
 - 1.3.2 插入图像和多媒体.....7
 - 1.3.3 创建链接.....8
 - 1.3.4 创建表格.....10
 - 1.3.5 创建 CSS 样式.....12
 - 1.3.6 创建 AP 布局元素.....13
 - 1.3.7 创建框架.....14
 - 1.3.8 定义行为.....14
 - 1.3.9 创建表单.....15
- 1.4 实践总结.....15

第 2 章 搭建动态网站运行环境.....16

- 2.1 认识动态网站架构.....16
- 2.2 安装 IIS 组件.....17
- 2.3 测试 IIS 服务器.....19
 - 2.3.1 在本地测试 IIS.....19
 - 2.3.1 在远程测试 IIS.....20
- 2.4 配置 IIS 服务环境.....20
 - 2.4.1 配置 WWW 服务属性.....21
 - 2.4.2 建立 WWW 服务虚拟目录.....21
 - 2.4.3 配置 FTP 服务属性.....22
 - 2.4.4 建立 FTP 服务虚拟目录.....23

- 2.5 在 Dreamweaver 中定义站点.....23
 - 2.5.1 认识站点及其规划.....23
 - 2.5.2 定义本地站点.....24
 - 2.5.3 定义动态站点.....24
 - 2.5.4 定义 FTP 站点.....27
- 2.6 实践总结.....28

第 3 章 ASP 开发基础.....29

- 3.1 ASP 简介.....29
 - 3.1.1 认识服务器技术.....29
 - 3.1.2 认识 ASP.....30
 - 3.1.3 ASP 特点.....31
- 3.2 ASP 工作原理.....31
- 3.3 创建 ASP 文件.....32
 - 3.3.1 编写 ASP 文件.....32
 - 3.3.2 运行 ASP 文件.....33
- 3.4 ASP 对象概述.....34
- 3.5 Response 对象.....35
 - 3.5.1 Response 对象的属性.....35
 - 3.5.2 Response 对象的方法.....38
 - 3.5.3 Response 对象的数据集合.....42
- 3.6 Request 对象.....42
 - 3.6.1 Request 对象的属性.....43
 - 3.6.2 Request 对象的方法.....43
 - 3.6.3 Request 对象集合概述.....43
 - 3.6.4 Form 数据集合.....44
 - 3.6.5 QueryString 数据集合.....46
 - 3.6.6 GET 和 POST 数据传输方式.....49
 - 3.6.7 Cookies 数据集合.....50
 - 3.6.8 ServerVariables 数据集合.....51
 - 3.6.9 ClientCertificate 数据集合.....55
- 3.7 Session 对象.....55
 - 3.7.1 使用 Session 对象.....56

3.7.2	Session 对象的属性	57	4.6.1	静态数组	87
3.7.3	Session 对象的方法	58	4.6.2	动态数组	88
3.7.4	Session 对象数据集合	58	4.6.3	使用 Erase 语句清除数组	89
3.7.5	Session 对象的事件	59	4.7	VBScript 运算符	90
3.9	Cookie 对象	59	4.7.1	算术运算符	90
3.9.1	Cookie 简介	60	4.7.2	比较运算符	90
3.9.2	创建 Cookie 对象	60	4.7.3	连接运算符	91
3.9.3	读取 Cookie 的值	61	4.7.4	逻辑运算符	92
3.10	Application 对象	63	4.7.5	运算符优先级	92
3.10.1	使用 Application	63	4.8	使用条件语句	93
3.10.2	Application 对象的属性	63	4.8.1	If...Then...Else 语句	93
3.10.3	Application 对象的方法	63	4.8.2	Select Case 语句	95
3.10.4	Application 对象数据集合	64	4.9	使用循环语句	96
3.10.5	Application 对象的事件	64	4.9.1	Do 循环	96
3.11	Server 对象	66	4.9.2	退出 Do 循环	97
3.11.1	Server 对象的属性	66	4.9.3	While 循环	98
3.11.2	Server 对象的方法	66	4.9.4	For 循环	98
3.12	ObjectContext 对象	69	4.9.5	For Each 循环	99
3.12.1	ObjectContext 对象的方法	70	4.10	VBScript 过程	100
3.12.2	ObjectContext 对象的事件	70	4.10.1	Sub 过程基本用法	100
3.13	ASP 组件	70	4.10.2	Sub 过程使用技巧	101
3.14	ASP 文件系统组件	71	4.10.3	调用 Sub 过程	102
3.14.1	系统驱动器操作	71	4.11	VBScript 函数	103
3.14.2	系统文件夹操作	73	4.11.1	Function 过程基本用法	103
3.14.3	系统文件夹操作	74	4.11.2	Function 过程使用技巧	103
3.15	实践总结	78	4.11.3	Function 过程的返回值	104
第 4 章 VBScript 程序设计		79	4.12	VBScript 内置函数	105
4.1	脚本语言简介	79	4.12.1	数学函数	105
4.1.1	服务器端脚本	80	4.12.2	字符串函数	106
4.1.2	客户端脚本	81	4.12.3	转换函数	108
4.2	VBScript 脚本语言概述	82	4.12.4	日期/时间函数	109
4.3	VBScript 数据类型	82	4.12.5	判断函数	109
4.4	VBScript 变量	83	4.12.6	数组函数	110
4.4.1	使用 Dim 语句声明变量	84	4.13	VBScript 类	110
4.4.2	使用 Public 语句声明变量	84	4.14.1	认识类	110
4.4.3	使用 Private 语句声明变量	84	4.14.2	创建类	111
4.4.4	隐式声明变量	85	4.14.3	实例化类	111
4.4.5	变量命名规则	85	4.14.4	定义类的属性	112
4.4.6	变量的作用域	85	4.14.5	定义类的方法	114
4.4.7	变量的存活期	86	4.14.6	定义类的事件	114
4.4.8	变量赋值	86	4.14	VBScript 编码规范	115
4.5	VBScript 常量	86	4.14.1	代码注释方法	115
4.6	VBScript 数组	87	4.14.2	注释规范	115
			4.14.3	常量命名规范	116

4.14.4 变量命名规范	116	6.5 表单验证	161
4.14.5 代码格式化	116	6.5.1 注册信息验证	162
4.15 实践总结	117	6.5.2 完善信息验证	163
第 5 章 数据库访问、查询和操作	118	6.6 Spry 表单验证	165
5.1 认识数据库	118	6.7 自定义 JavaScript 脚本验证	167
5.2 SQL 语言	119	6.8 注册信息确认	169
5.2.1 SQL 语言规范	120	6.8.1 使用 cdonts 组件发送确认邮件	169
5.2.2 SQL 数据查询	120	6.8.2 使用 JMail 组件发送确认邮件	169
5.2.3 Select 聚合查询	124	6.9 注册信息加密	171
5.2.4 Select 分组查询	125	6.10 实践总结	171
5.2.5 Select 多表查询	127	第 7 章 用户登录模块	173
5.2.6 Insert 插入数据	128	7.1 程序设计分析	173
5.2.7 Update 更新数据	129	7.2 数据库设计	174
5.2.8 Delete 删除数据	129	7.3 页面设计	174
5.3 认识 ADO	129	7.4 登录基本功能实现	175
5.4 数据库连接	130	7.4.1 客户端验证	175
5.4.1 创建 ODBC 连接	131	7.4.2 服务器端用户登录	176
5.4.2 使用 DSN 创建数据库连接	133	7.4.3 传递会话变量	177
5.4.3 使用自定义连接字符串创建数据库连接	134	7.4.4 注销用户	178
5.4.4 使用 Connection 对象创建数据库连接	135	7.5 自动记忆输入密码	179
5.5 Connection 对象	136	7.6 增加图形验证码	182
5.5.1 Connection 对象的属性	136	7.7 防止 SQL 注入式攻击	184
5.5.2 Connection 对象的方法	137	7.8 防止用户绕过登录	186
5.5.3 Connection 对象的集合	142	7.9 实践总结	187
5.6 Command 对象	143	第 8 章 用户管理模块	188
5.6.1 Command 对象的属性	143	8.1 程序设计分析	188
5.6.2 Command 对象的方法	143	8.2 数据库设计	189
5.7 RecordSet 对象	146	8.2.1 admin 表	189
5.7.1 RecordSet 对象的属性	146	8.2.2 infor 表	190
5.7.2 RecordSet 对象的方法	147	8.3 页面设计	191
5.7.3 Field 对象和 Fields 集合	152	8.3.1 登录页面	192
5.8 实践总结	154	8.3.2 导航页面	192
第 6 章 用户注册模块	155	8.3.3 信息操作页面	193
6.1 模块设计分析	155	8.3.4 信息查询页面	193
6.2 设计和连接数据库	156	8.3.5 用户管理页面	194
6.2.1 设计数据库	156	8.4 用户管理基本功能实现	194
6.2.2 连接数据库	157	8.4.1 用户登录验证和引导	195
6.3 设计交互表单	159	8.4.2 用户搜索	196
6.4 实现基本的注册功能	160	8.4.3 显示搜索结果	198
6.4.1 添加【插入】服务器行为	160	8.4.4 修改用户信息	201
6.4.2 添加【检查新用户】服务器行为	161	8.4.5 用户账号管理	203
		8.5 高级管理功能实现	204

8.5.1	高级账户管理	204	10.2.2	userandip 表	240
8.5.2	纠正 Dreamweaver 服务器行为 存在的 Bug	206	10.2.3	history 表	241
8.5.3	账户高级删除	208	10.3	页面设计	241
8.5.4	录入用户信息	209	10.4	网站流量统计功能实现	241
8.5.5	用户身份分级限制	209	10.4.1	流量统计	241
8.5.6	注销用户	210	10.4.2	流量显示	243
8.5.7	管理员分级管理	210	10.4.3	指定要统计的站点	246
8.6	实践总结	212	10.5	数字图形化显示技术	246
10.6	实践总结	247			
第 9 章 留言板模块			第 11 章 友情链接管理模块		
9.1	程序设计分析	213	11.1	程序设计分析	248
9.2	数据库设计	214	11.2	数据库设计	250
9.2.1	admin 表	214	11.3	页面设计	251
9.2.2	feedback 表	215	11.3.1	友情链接显示页面	251
9.3	页面设计	216	11.3.2	链接显示嵌入页面	252
9.3.1	显示留言页面	216	11.3.3	增加链接页面	252
9.3.2	编辑留言页面	217	11.3.4	后台管理主页面	253
9.3.3	增加留言页面	217	11.4	前台功能实现	254
9.3.4	其他页面	218	11.4.1	显示友情链接	254
9.4	留言板基本功能实现	218	11.4.2	增加友情链接	256
9.4.1	定义记录集	219	11.4.3	随机数字验证码功能实现	258
9.4.2	绑定记录集	221	11.4.4	客户端信息验证	258
9.4.3	重复显示记录	225	11.4.5	服务器端验证和表单提交 处理	259
9.4.4	记录集分页	226	11.5	后台管理功能实现	261
9.4.5	控制导航条	227	11.5.1	管理链接主页面	261
9.4.6	增加记录	228	11.5.2	编辑主页面	264
9.5	留言板后台管理功能实现	229	11.5.3	删除、审核和退出管理	266
9.5.1	管理登录	229	11.6	实践总结	267
9.5.2	限制访问	230			
9.5.3	修改留言	230	第 12 章 网站搜索模块		
9.5.4	删除留言	231	12.1	程序设计分析	268
9.5.5	增加记录集分页提示信息	232	12.2	数据库设计	269
9.5.6	修改用户信息	234	12.3	页面设计	270
9.5.7	注销用户登录	234	12.3.1	高级查询主页面	270
9.6	扩展留言板功能	235	12.3.2	查询页面	271
9.6.1	留言回复	235	12.3.3	查询结果页面	271
9.6.2	留言审核	236	12.4	基本搜索功能实现	271
9.7	实践总结	238	12.4.1	查询数据库记录数	271
			12.4.2	获取客户端系统信息	273
			12.4.3	查询手机号码	275
			12.5	完善搜索功能	278
			12.5.1	完善手机号码查询	278
第 10 章 网站流量统计与分析					
10.1	程序设计分析	239			
10.2	数据库设计	240			
10.2.1	stat 表	240			

12.5.2 查询 IP 地址信息	279	14.4.4 发表评论	320
12.6 实践总结	281	14.4.5 显示评论	321
第 13 章 网站调查模块	282	14.5 图片分享功能实现	322
13.1 程序设计分析	282	14.5.1 图片列表	322
13.2 数据库设计	283	14.5.2 欣赏图片	323
13.3 页面设计	284	14.6 音乐分享功能实现	323
13.3.1 调查表页面	285	14.7 留言功能模块实现	324
13.3.2 增加信息页面	285	14.8 后台管理	325
13.3.3 管理页面	286	14.9 实践总结	326
13.4 前台功能实现	287	第 15 章 投票管理系统	327
13.4.1 多条件查询	287	15.1 程序设计分析	327
13.4.2 记录集横向重复显示	289	15.2 数据库设计	329
13.4.3 记录集横向换行显示	291	15.3 页面设计	329
13.4.4 记录集条件显示	292	15.3.1 导航菜单页面	330
13.4.5 动态样式表	294	15.3.2 标题页面	331
13.4.6 动态信息提示	296	15.3.3 后台管理主页面	332
13.5 后台管理	297	15.3.4 登录页面	332
13.5.1 后台控制台设计	298	15.3.5 增加选项页面	333
13.5.2 登录验证和信息审核	299	15.3.6 其他页面	334
13.5.3 增加信息	300	15.4 投票管理功能实现	335
13.6 实践总结	302	15.4.1 用户登录	335
第 14 章 博客	303	15.4.2 账户管理	336
14.1 程序设计分析	303	15.4.3 管理投票标题	337
14.2 数据库设计	305	15.4.4 增加投票标题	338
14.3 前台公共模块功能实现	307	15.4.5 编辑投票标题	339
14.3.1 导航模块	307	15.4.6 删除投票标题	340
14.3.2 脚部版权模块	309	15.4.7 增加选项	341
14.3.3 通用功能模块	309	15.4.8 编辑选项	343
14.3.4 博客照片动态控制	309	15.4.9 删除选项	344
14.3.5 博客公告	311	15.4.10 选择投票	344
14.3.6 最新日志列表	312	15.5 投票分析和图表化显示	346
14.3.7 日志分类	312	15.5.1 生成投票	346
14.3.8 设计日历	313	15.5.2 投票监测	348
14.3.9 最新评论列表	314	15.5.3 显示投票结果	348
14.3.10 最新留言列表	315	15.6 实践总结	350
14.3.11 友情链接列表	316	第 16 章 新闻发布与管理系统	351
14.3.12 博客统计	316	16.1 程序设计分析	351
14.3.13 博客动态背景音乐	317	16.2 数据库设计	353
14.4 前台日志功能模块实现	318	16.2.1 class 表	353
14.4.1 日志列表	318	16.2.2 message 表	353
14.4.2 日志详细页	319	16.3 页面设计	354
14.4.3 日志内容显示	319	16.4 新闻发布	356

16.4.1	系统首页	356	18.3.2	主题列表页面	394
16.4.2	新闻列表页面	357	18.3.3	帖子列表页面	394
16.5	新闻管理	359	18.3.4	发布页面	395
16.5.1	栏目管理	360	18.4	论坛基本功能实现	396
16.5.2	信息管理	362	18.4.1	定义自定义连接字符串	396
16.6	实践总结	363	18.4.2	栏目显示	397
第 17 章	网站即时聊天模块	364	18.4.3	主题列表	399
17.1	程序设计分析	364	18.4.4	帖子列表	401
17.2	数据库设计	366	18.4.5	发布主题	403
17.2.1	chat 表	366	18.4.6	发帖子	405
17.2.2	users 表	366	18.5	实践总结	407
17.3	页面设计	367	第 19 章	网上购物 (上)	408
17.3.1	聊天信息发布页面	367	19.1	程序设计分析	408
17.3.2	聊天信息显示页面	368	19.2	数据库设计	410
17.3.3	导航页面	369	19.3	页面设计	413
17.3.4	主体框架集	370	19.3.1	快速查询表单	414
17.3.5	聊天室框架集	371	19.3.2	高级查询表单	415
17.3.6	登录页面	371	19.3.3	评级和评论表单	416
17.4	聊天室基本功能实现	372	19.4	商品展示	417
17.4.1	用户登录和用户信息存储	372	19.4.1	栏目导航	417
17.4.2	聊天信息发布	374	19.4.2	购物首页	421
17.4.3	聊天信息即时显示	376	19.4.3	商品分类	426
17.4.4	在线用户即时显示	377	19.4.4	二级商品显示页面	429
17.4.5	聊天信息实时刷新	378	19.4.5	商品详细页面	434
17.4.6	退出聊天室	379	19.4.6	网上交互	436
17.5	扩展聊天室功能	380	19.5	实践总结	438
17.5.1	发布图片和点歌	380	第 20 章	网上购物 (下)	439
17.5.2	用户登录功能完善	382	20.1	程序设计分析	439
17.5.3	禁止恶意用户登录	383	20.2	数据库设计	441
17.5.4	踢人	385	20.3	页面设计	441
17.5.5	私聊	386	20.4	购物系统基本功能	442
17.5.6	离开提示	388	20.4.1	用户注册	442
17.5.7	清除过时聊天信息	388	20.4.2	用户登录	444
17.6	实践总结	389	20.4.3	注销用户	445
第 18 章	BBS 论坛模块	390	20.4.4	消息中心	445
18.1	程序设计分析	390	20.4.5	个人资料	446
18.2	数据库设计	392	20.4.6	收货资料	447
18.2.1	Catalogs 表	392	20.4.7	修改密码	449
18.2.2	TalkArea 表	392	20.4.8	购物列表	450
18.2.3	topic 表	393	20.4.9	下订单	451
18.3	页面设计	393	20.4.10	结账	453
18.3.1	论坛栏目页面	393	20.4.11	完成订单	455

20.5 实践总结	459	23.3.3 表单样式设计	497
第 21 章 文件下载演示模块	460	23.3.4 班级名片页面设计	498
21.1 程序设计分析	460	23.3.5 注册表单设计	499
21.2 数据库设计	462	23.3.6 动态脚本设计	500
21.2.1 class 表	462	23.4 同学录前台主要功能实现	501
21.2.2 nclass 表	462	23.4.1 同学录主页面	502
21.2.3 download 表	462	23.4.2 班级留言	506
21.3 页面设计	463	23.4.3 班级名片	508
21.4 软件下载功能实现	464	23.4.4 班级相册	509
21.4.1 下载排行页面	464	23.5 实践总结	510
21.4.2 分类列表页面	466	第 24 章 播客	511
21.4.3 具体下载页面	468	24.1 程序设计分析	511
21.5 根据字段进行排序	469	24.2 安装 FMS 服务器	
21.6 实践总结	471	(Flash Media Server)	513
第 22 章 电子相册	472	24.3 管理 FMS 服务器	516
22.1 程序设计分析	472	24.3.1 应用程序管理	517
22.2 数据库设计	473	24.3.2 用户管理	519
22.3 页面设计	474	24.3.3 服务器管理	519
22.3.1 上传表单页面	474	24.4 创建播客 FMS 应用程序	520
22.3.2 头部页面	475	24.5 用 Dreamweaver CS5 实现与 FMS	
22.3.3 其他表单页面	475	服务器通信	521
22.4 电子相册浏览与管理	476	24.5.1 Flash 方式	521
22.4.1 注册、登录和用户信息操作	476	24.5.2 累进式下载	522
22.4.2 浏览图片	479	24.5.3 流媒体	523
22.4.3 图片评论	484	24.6 播客系统数据库设计	525
22.4.4 删除服务器上的图片文件	486	24.6.1 flash 表	525
22.5 图片上传	488	24.6.2 author 表	526
22.5.1 无组件上传程序包	488	24.7 页面设计	526
22.5.2 上传图片	490	24.8 视频发布	528
22.6 实践总结	491	24.8.1 热门分类	528
第 23 章 同学录	492	24.8.2 视频分类显示	529
23.1 程序设计分析	492	24.9 视频播放	532
23.2 数据库设计	494	24.10 实践总结	533
23.3 页面设计	494	附录 A 上机操作可能出现的错误及	
23.3.1 标题导航栏页面	495	对策 (见光盘)	
23.3.2 主页面设计	497	附录 B ASP 错误代码解释汇编 (见光盘)	

Dreamweaver 是专业级的 Web 设计和开发工具，利用它可以轻松设计网页、开发网站和实现各种 Web 应用。借助 Dreamweaver 提供的可视化编辑工具，或者直接在【代码】视图中快速编写代码（当然也可以两者配合），就能够自由轻松地完成你的开发项目。

总之，只要读者简单了解网页制作的简单知识，就可以制作出比较漂亮、复杂的页面。本章重点介绍 Dreamweaver CS5 的基本概念和基本操作方法，为后面的实战学习奠定基础。

技术要点

- 了解 Dreamweaver CS5 功能
- 熟悉 Dreamweaver CS5 界面
- 掌握 Dreamweaver CS5 基本操作

1.1 Dreamweaver 工具概述

作为 Adobe CS5 开发套件的组成部分，Dreamweaver 是目前最流行的 Web 设计和开发工具之一。它提供了可视化操作环境和代码编程环境，提供了超强的所见即所得的功能，与其他网页编辑软件相比，Dreamweaver 具有如下主要特点。

1. 操作方便、快捷

Dreamweaver 提供了可视化操作环境，用户基本上不需要编写多少代码，就可以快速制作网页或架设站点。利用其提供的各种面板，可以查看站点或 Web 应用的所有元素或资源，并将它们从面板直接拖到页面中。另外，Dreamweaver 能够与 Photoshop、Flash、Fireworks 等其他 Adobe 的 CS 套装软件实现无缝协作，使图像制作和编辑、动画设计变得更加轻松自如。

Dreamweaver 提供了 HTML 代码编辑器，并且可方便地在可视化编辑状态与源代码编辑状态之间自由切换。在 Dreamweaver 的【代码】视图中，光标所处代码的位置，切换到【设计】视图之后，依然显示对应网页的位置。

2. 生成高效网页源代码

Dreamweaver 可视化的网页编辑器可以把设计者的操作转换成高效的 HTML 源代码，不会产生大量冗余代码，而且还可以利用 Dreamweaver 提供的工具清除网页源代码中已有的冗余代码。

3. 提供强大的动态网站开发功能

Dreamweaver 支持多种服务器技术，例如，ASP、JSP 和 PHP 等。可以创建和管理强大

的服务器站点，开发 Web 应用程序。在本地站点中，Dreamweaver 可以自动更新相应的超链接，大大简化了工作，并通过 FTP 等协议上传、更新或管理远程站点。

利用 Dreamweaver 行为，用户不需要了解 JavaScript 脚本语言，也能为网页添加动态效果，通过【服务器行为】面板，还可以在页面中添加各种复杂的服务器端操作。

4. 提供强大的扩展功能

Dreamweaver 提供了功能全面的代码编辑工具，例如，HTML、CSS、JavaScript 编辑工具，并可以自由导入导出各种代码。同时，Dreamweaver 支持第三方插件，允许用户自定义插件，扩展 Dreamweaver 操作功能，使 Dreamweaver 更方便使用。

1.2 Dreamweaver CS5 操作环境

安装完 Dreamweaver CS5 并启动程序之后，会弹出一个对话框，要求用户选择一种默认编辑器，这样 Adobe Dreamweaver CS5 会自动设置所选文件类型为默认编辑器，当用户双击相应类型的文件时，会自动启动 Dreamweaver CS5 进行编辑。

Dreamweaver CS5 包含 8 种工作区布局，在顶部的【设计器】下拉菜单中可以选择切换布局，以实现在不同开发环境中操作，如图 1.1 所示，当然读者也可以自定义主界面的布局风格，以实现个性化开发需要。

其中常用布局包括编码器、设计器和双重屏幕，默认为设计器工作区布局，它们分别针对编程人员、设计师和网站测试人员而设计了不同的工作环境。这些环境能够最大化方便不同角色的工作，发挥 Dreamweaver 的潜力。

在编码器布局下，重要的面板和编辑窗口集成在一起，缩小了编辑窗口。右边是 Dreamweaver CS5 的代码布局，将面板组停靠在左侧，如图 1.2 所示。而设计器布局与编码器布局正好相反，如图 1.3 所示。本书是在设计器布局中进行介绍，建议用户选择进入设计器工作区布局。虽然两者在功能上没有本质差别，但使用习惯分别适合于开发人员和设计人员。



图 1.1 工作区布局下拉列表

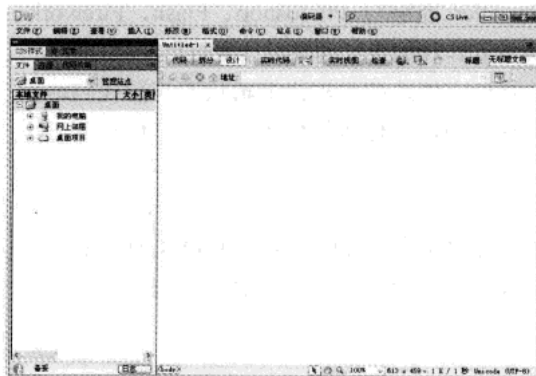


图 1.2 编码器工作区布局

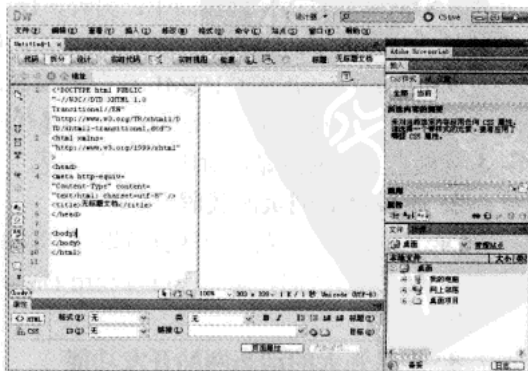


图 1.3 设计器工作区布局



如果要切换工作区布局,可以选择【窗口】|【工作区布局】命令,在打开的子菜单中选择相应的布局命令就可以了。在该菜单中,用户可以选择不同样式的布局。同时也可以保存个人使用布局。由于使用习惯的差异,经常使用时,建议保存符合个人习惯的工作区布局,当工作区布局被改变后,可以快速恢复到以前保存的布局。

Dreamweaver CS5 主窗口包括标题栏、菜单栏、插入工具栏、文档编辑窗口、属性面板、浮动面板 6 个部分,如图 1.4 所示。

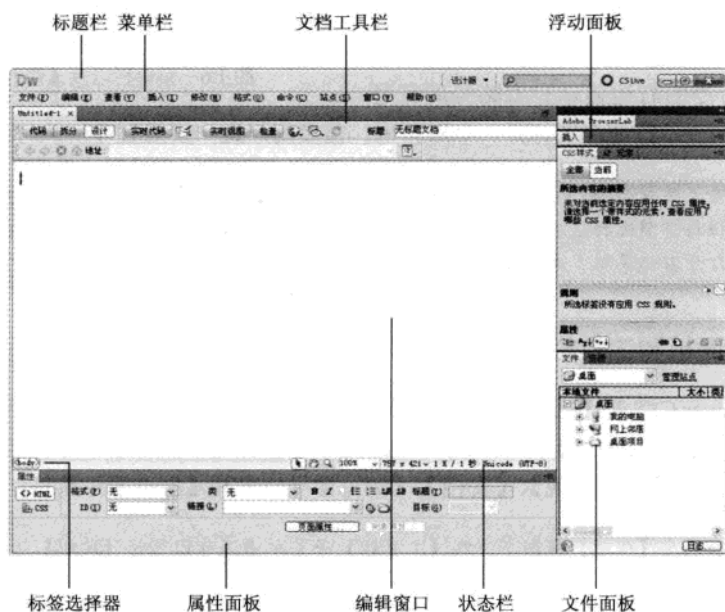


图 1.4 Dreamweaver CS5 主窗口布局示意图

1.2.1 标题栏

在 Dreamweaver CS5 主窗口的顶部是标题栏,如图 1.5 所示。与传统的标题栏截然不同,CS5 标题栏左侧显示 DW 图标,右侧显示 3 个常用的操作项:设计器、搜索框和 CSLive。单击对应图标右侧的向下按钮,可以从弹出的下拉菜单中选择相应的操作命令。

标题栏最右侧显示布局视图、搜索文本框和 3 个窗口操作按钮,分别对应主窗口的【最小化】、【最大化】和【关闭】命令。



图 1.5 标题栏

1.2.2 菜单栏


Dreamweaver CS5 菜单栏共有 10 种菜单,包括文件、编辑、查看、插入、修改、格式、命令、站点、窗口、帮助。单击其中任意一个菜单,就会打开一个下拉菜单,例如单击【修改】菜单,会打开该下拉菜单,如图 1.6 所示。

- 如果菜单命令选项显示为浅灰色，则表示该指令在当前的状态下不能执行。
- 如果命令的右边显示有键盘的代码，则表示该代码为该命令的快捷键，熟练使用快捷键可以提高工作效率。
- 如果命令的右边显示有一个小黑三角的符号 ▾，则表示该命令还包含有子菜单，鼠标停留在该菜单项上片刻即可显示子菜单，也可以单击打开子菜单。
- 如果命令的右边显示有省略号的符号 ...，则表示该命令能打开一个对话框，需要用户进一步设置才能执行命令。



图 1.6 菜单栏以及【修改】菜单



除了菜单栏外, Dreamweaver CS5 还提供各种快捷菜单, 利用这些快捷菜单可以很方便地选择与当前选择区域相关的命令。例如, 在 Dreamweaver CS5 主窗口中单击鼠标右键, 可以打开右键菜单; 单击面板右上角的菜单按钮, 可以打开面板菜单。

1.2.3 工具面板和工具栏

Dreamweaver CS5 把【插入】面板设计为面板，停靠在右侧面板组中。选择【窗口】|【插入】命令，可以打开【插入】面板。另外，选择【查看】|【工具栏】命令，在打开的子菜单中还可以选择显示【样式呈现】、【文档】、【浏览器导航】和【标准】工具栏。



如果在【代码】视图下，在【工具栏】子菜单中还可以显示【编码】工具栏。

在默认状态下, Dreamweaver CS5 只显示【文档】工具栏。另外, 选择【窗口】|【插入】菜单命令, 可以打开或关闭【插入】面板。在【插入】面板的子菜单中可以选择显示不同类型的工具项, 如图 1.7 所示。



图 1.7 显示所有工具栏和工具面板

【插入】面板中包含有 8 类快捷控制按钮，包括常用、布局、表单、数据、Spry、文本、InContext Editing 和收藏夹。系统默认显示为【常用】工具栏，用户可以快速进行切换。

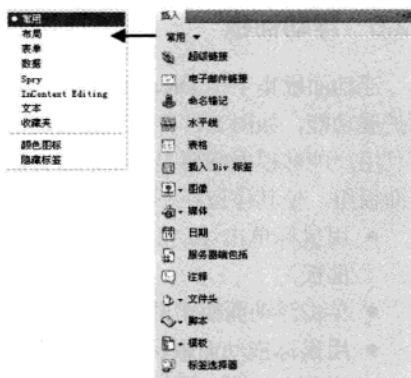


图 1.8 【插入】面板中的工具分类

1.2.4 状态栏

状态栏位于文档编辑窗口的底部，如图 1.9 所示。在状态栏最左侧是【标签选择器】，显示当前选定内容标签的层次结构。单击该层次结构中的任何标签可以选择该标签及其全部内容。例如，单击<body>标签可以选择整个文档。

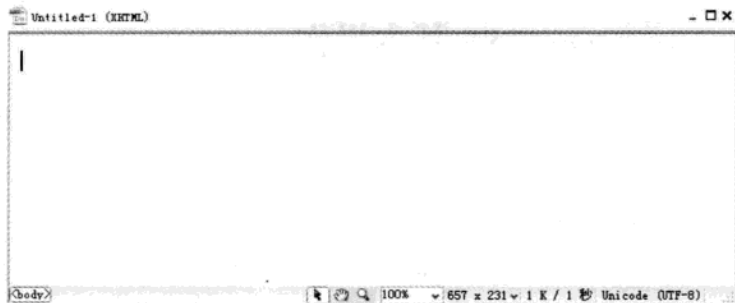





图 1.9 状态栏

状态栏中的【选取工具】按钮、【手形工具】按钮和【缩放工具】按钮分别用来选取对象、移动页面工作区域和缩放页面。按住 Alt 键，可以使用【缩放工具】缩小页面工作区域。只有当页面被放大到超出窗口后，才可以使用【手形工具】按钮移动页面。

1.2.5 属性面板

在【属性】面板中可以设置选中对象的各种属性。【属性】面板的设置项目会根据被设计对象的不同而不同。

选择【窗口】|【属性】命令，可以打开或关闭【属性】面板，如图 1.10 所示。【属性】面板上的大部分选项都可以在【修改】菜单项中找到。



Dreamweaver CS5 把所有对象的属性分为两大类：HTML 和 CSS。单击面板左侧的按钮，可以切换到对象的 HTML 属性设置状态，在这种状态下，所设置的属性都是以 HTML 标签属性的方式定义的。单击面板左侧的按钮，可以切换到对象的 CSS 样式设置状态，在这种状态下，所设置的对象属性都是以 CSS 样式的方式定义的。



图 1.10 【属性】面板

1.2.6 浮动面板

浮动面板集中了 Dreamweaver CS5 的大部分功能,包括很多重要的扩展功能,如图 1.11 所示。使用浮动面板不仅可以节省屏幕空间,而且用户能够根据需要显示不同的浮动面板。例如,拖动面板可以脱离面板组,使其停留在不同的位置。

- 用鼠标单击浮动面板标题栏中的灰色区域,可以折叠或展开面板。
- 单击浮动面板组的标题栏,可以展开或收缩所有面板。
- 用鼠标拖动面板标题栏,可以把面板从面板组中拖出来,作为独立的浮动面板放置在 Dreamweaver 工作区域的任意位置。同样,用相同的方法可以将自由浮动的面板拖回默认状态。

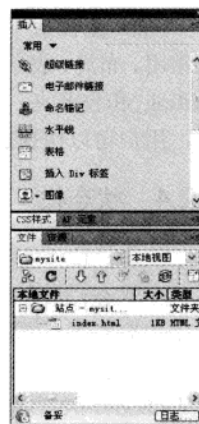


图 1.11 浮动面板组

1.3 Dreamweaver CS5 基本操作

Dreamweaver CS5 支持各种流行的 Web 技术,使用它提供的工具和面板能够轻松插入和设置各种网页元素。

1.3.1 输入文本和版式设计

文本是网页的基本元素,在 Dreamweaver CS5 文档编辑窗口中可以自由输入文本,也可以从其他窗口中复制文本到文档编辑窗口中。

此外,选择【文件】|【导入】菜单命令,还可以快速导入 XML、表格式数据、Excel 文档、Word 文档,并保留原文档中已设置好的格式。

下面重点介绍特殊文本的输入方法。

1. 输入特殊字符

输入特殊字符的操作方法有如下 5 种。


- 选择【插入】面板的【文本】选项,然后选择最右边的扩展按钮 ,在弹出的下拉菜单中选择一种特殊字符,也可以选择【其他字符】选项,能提供更多的字符,如图 1.12 所示。



图 1.12 利用【插入】面板插入特殊字符

- 选择【插入】|【HTML】|【特殊字符】菜单命令，可以在弹出的子菜单中选择并插入各种特殊字符；如果在该子菜单中没有需要的字符，可以选择【其他字符】命令，打开【插入其他字符】对话框，可以在其中选择要插入的对象，如图 1.12 所示。

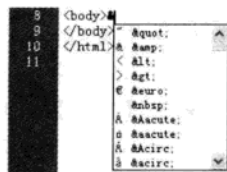


图 1.13 使用代码快速输入特殊字符

- 在【代码】视图下，输入“&”字符，Dreamweaver CS5 会自动以下拉列表的方式显示全部特殊字符，如图 1.13 所示，从中选择一个特殊字符即可。
- 使用输入法直接输入特殊字符。在【设计】视图下，使用区位码或者其他标准输入法，按下 V 键的同时直接输入数字（零除外），会弹出一个列表框，从中选择一个特殊字符即可。
- 复制其他文档中的特殊字符，然后粘贴到 Dreamweaver 的文档中。


2. 输入空格

浏览器在解析 HTML 源代码时，将忽略连续多个空格，只显示一个空格。因此，在 Dreamweaver 中不能直接输入多于一个的空格，只能用其他方法，如下所示。

- 切换中文输入法为全角模式，输入全角的空格。
- 选择【插入】|【HTML】|【特殊字符】|【不换行空格】命令，插入空格。
- 直接按 Ctrl+Shift+Space 组合键，快速插入多个空格。
- 在【插入】面板的【文本】选项中单击【特殊字符】按钮，选择【不换行空格】命令。
- 在【代码】视图下，输入多个“ ”， 表示空格的意思。

3. 输入日期

在 Dreamweaver 中输入日期的方法有以下两种。

- 在【设计】视图中直接以文本方式输入日期。
- 选择【插入】面板的【常用】选项中单击【日期】按钮，在打开的【插入日期】对话框中设置插入日期的格式，如图 1.14 所示。

输入文本之后，用户可以选中该文本，然后在【属性】面板中设置该文本的格式和版式，如图 1.15 所示。

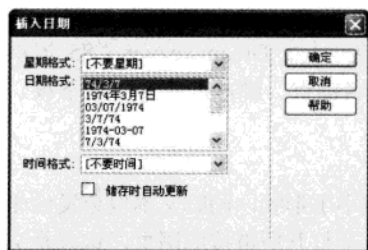


图 1.14 【插入日期】对话框

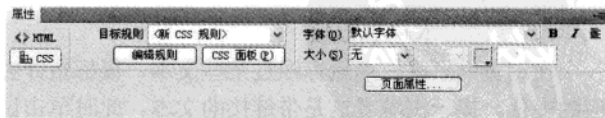




图 1.15 文本【属性】面板

1.3.2 插入图像和多媒体

图像和多媒体在网页中比较常见。在网页中加入合适的图像和动画，能给浏览者留下深刻的印象。插入图像和多媒体的方法基本相同，下面列出常用的两种方法。


- 选择【插入】|【图像对象】或者【插入】|【媒体】子菜单中选择相应的命令，即可插入所需要的对象。
- 在【插入】面板的【常用】选项中，单击【图像】按钮或者【媒体】按钮，从弹出的下拉菜单中选择相应的选项即可插入所需要的对象。



在插入的过程中，可能需要多个操作步骤，要求用户确定相关属性和指定对象源文件，因此在插入图像和多媒体之前应先制作好对象。

在 Dreamweaver 中插入图像，Dreamweaver 会自动生成该图像的路径引用，如果要使图像能够正确地网页中显示，必须保证此图像文件在当前的站点内。如果不在站点内，Dreamweaver 会提示是否将此图像复制到当前站点的文件夹中。

在 Dreamweaver CS5 中，插入的图像包括 4 种对象。

- 插入单个图像。
- 插入鼠标经过图像。鼠标经过图像由两幅图像组成：一个是主图像，就是首次载入页面时显示的图像；另一个是次图像，就是当鼠标指针移过主图像时显示的图像，当鼠标移开时又恢复成原来的图像。这两个图像应该大小相等，如果这两个图像的大小不同，Dreamweaver CS5 会自动调整第 2 幅图像，使之与第 1 幅图像相匹配。
- 插入图像占位符。图像占位符是指没有设置 src 属性的标签。在编辑窗口中默认显示为灰色空白，在浏览器中浏览时显示为一个红叉，如果为其指定了 src 属性，则该图像占位符就会立即显示该图像。图像占位符的使用极大地方便了网页设计，用户可先插入图像占位符，然后再制作图像，或者把工作交给专业设计师独立完成，这样极大提高了网页制作效率。
- 插入多媒体。在 Dreamweaver CS5 中，插入的媒体包括 6 种对象，如图 1.16 所示。插入图像和多媒体之后，选中该对象可以在【属性】面板中设置该对象的属性。

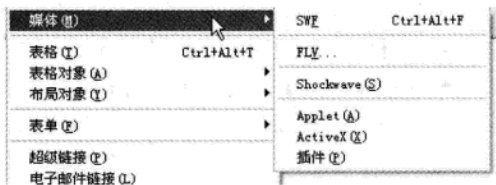


图 1.16 6 种媒体对象

1.3.3 创建链接

在网站内或网站之间，网页都是通过超链接技术联系在一起，Dreamweaver CS5 提供了各种链接方法和使用技巧，如设置 E-mail 链接、创建命名锚点、脚本链接、文件下载、跳转菜单以及空链接等。

1. 创建文本超级链接

在浏览网页时，当鼠标经过某些文本，鼠标指针会变成小手形状，同时文本也会发生相应的变化，提示浏览者这是带链接的文本。此时单击鼠标，会打开所链接的网页，这就是文本超级链接。创建文本超链接的具体操作步骤如下。

- 01 在【设计】视图中选择文本。
- 02 选择【窗口】|【属性】命令，打开【属性】面板，单击【HTML】按钮，切换到 HTML 属性设置状态下。
- 03 单击【链接】文本框右边的【选择文件】图标按钮，在打开的【选择文件】对话框中浏览并选择一个文件，如图 1.17 所示。在【URL】文本框中显示被链接文件的路径，在【相

对于】下拉列表中选择【文件】选项（设置相对路径）或【站点根目录】选项（设置根路径），然后单击【确定】按钮。

- 04 选择被链接文件的载入目标。在默认情况下，被链接文件在当前窗口或框架中打开。要使被链接的文件显示在其他地方，需要从【属性】面板的【目标】下拉列表选择一个选项，如图 1.18 所示。



图 1.17 【选择文件】对话框

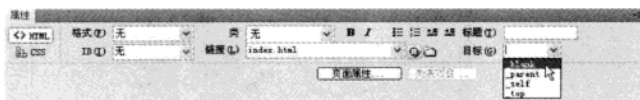



图 1.18 设置【目标】属性

- _blank: 将被链接文件载入到新的未命名浏览器窗口中。
- _parent: 将被链接文件载入到父框架集或包含该链接的框架窗口中。
- _self: 将被链接文件载入到与该链接相同的框架或窗口中。
- _top: 将被链接文件载入到整个浏览器窗口并删除所有框架。

2. 创建 E-mail 链接

E-mail 链接可以帮助用户快速的反馈意见。当浏览者单击电子邮件链接时，就打开浏览器默认的电子邮件处理程序（如 Outlook Express），收件人邮件地址被电子邮件链接中指定的地址自动更新，浏览者不用手工输入。

创建 E-mail 链接的步骤如下。

- 01 在【设计】视图中将光标置于希望显示电子邮件链接的地方。
- 02 选择【插入】|【电子邮件链接】命令，或者在【插入】面板中单击【电子邮件链接】按钮。
- 03 在打开的【电子邮件链接】对话框的【文本】文本框中输入作为电子邮件链接显示在网页中的文本，中英文都可以。
- 04 在 E-mail 文本框中输入邮件应该送达的 E-mail 地址，如图 1.19 所示。

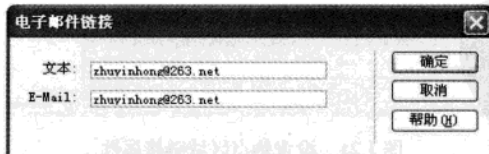


图 1.19 设置【电子邮件链接】对话框

- 05 单击【确定】按钮。

3. 创建图像热点链接

在同一幅图像上单击不同区域可以链接到不同的页面，这就是图像热点链接。创建图像热点链接的操作步骤如下。

- 01 在【设计】视图下选择【插入】|【图像】命令，插入图像，然后选中图像。
- 02 在【属性】面板的【地图】文本域中输入热点区域名称。如果一个网页的图像中有多个热点区域，则必须依次为每个图像热点区域起一个唯一的名称。
- 03 然后选择热区绘图工具，根据对象形状可以选择不同的热区绘图工具，这里选择【矩形热点工具】按钮，在选定的图像上拖动鼠标指针创建图像热区，如图 1.20 所示。



图 1.20 创建热点区域

- 04 选中该热区，在【属性】面板中设置链接属性，如图 1.21 所示（可参照创建文本超级链接的方法）。

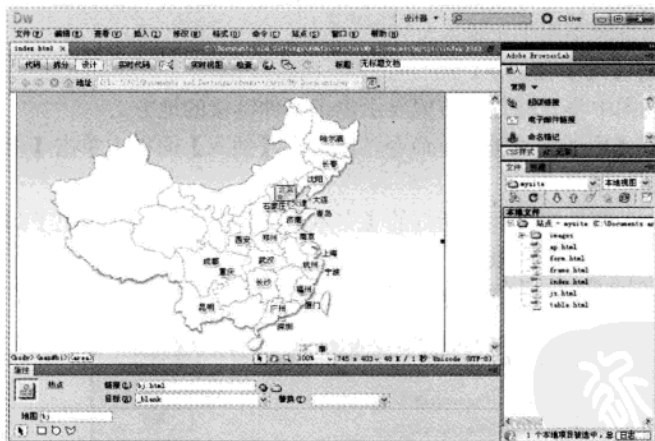



图 1.21 设置热点区域链接属性

1.3.4 创建表格

表格在网页设计中非常有用，主要用于排列数据。在传统网页布局还可以使用表格定位网页对象。通过设置表格宽度、高度及彼此之间的比例大小等参数，就可以把不同的网页元素分别插入在不同的单元格之中以达到页面的平衡。

一张表格横向叫行，纵向叫列。行列交叉部分就叫单元格。单元格中的内容和边框之间的距离叫边距。单元格和单元格之间的距离叫间距。整张表格的边缘叫边框。表格各部分名称如图 1.22 所示。

插入表格的操作步骤如下。

- 01 将光标置于页面中要插入表格的位置。
- 02 选择【插入】|【表格】命令（组合键为 Ctrl+Alt+T），或者单击【插入】面板中第 4 个【表格】按钮图标，打开【表格】对话框，如图 1.23 所示。

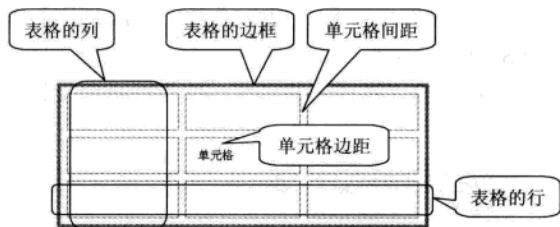


图 1.22 表格各部分名称

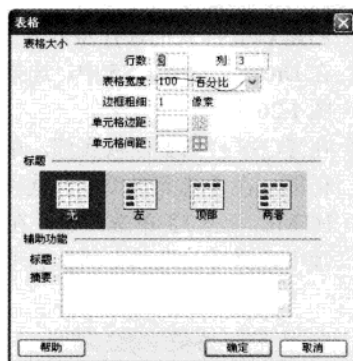


图 1.23 【表格】对话框

- 03 在【表格】对话框中设置表格的行数和列数，设置表格宽度、边框、间距等属性后，单击【确定】按钮即可插入表格。

插入表格后选中表格，则可以在【属性】面板中设置表格的显示属性，如图 1.24 所示。

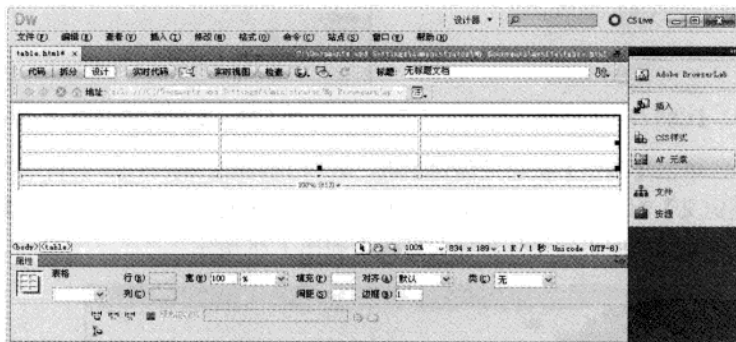


图 1.24 表格【属性】面板

如果选中单元格，或者把光标置于单元格内部，则可以在【属性】面板的底部设置单元格的显示属性，如图 1.25 所示。

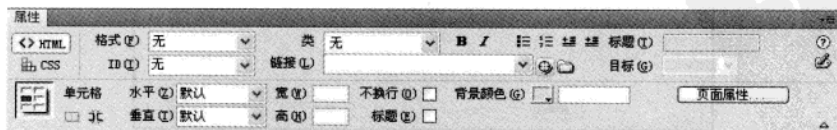



图 1.25 单元格【属性】面板

1.3.5 创建 CSS 样式

CSS 样式用来控制网页的外观。CSS 属于网页表现层的语言，而 HTML 属于网页结构层的语言，CSS 和 HTML 结构就可以设计出结构合理且显示漂亮的页面。

使用 CSS 样式表可以方便地控制 HTML 标签的显示属性。对页面布局、字体、颜色、背景和其他图文效果实现更加精确的控制。用户只修改一个 CSS 样式表文件就可以改变一批网页的外观和格式，保证在所有浏览器和平台之间的兼容性，拥有更少的编码、更少的页数和更快的下载速度。

创建 CSS 样式的操作步骤如下。

- 01 在 Dreamweaver CS5 中，选择【窗口】|【CSS 样式】命令，打开【CSS 样式】面板，如图 1.26 所示。
- 02 在【CSS 样式】面板底部，单击【新建 CSS 规则】按钮，打开【新建 CSS 规则】对话框。在【选择器类型】选项中选择 CSS 样式的类型，这里选择【类（可应用于任何 HTML 元素）】单选按钮；在【名称】文本框中输入样式表的名称；在【规则定义】选项中选择【仅限该文档】选项，如图 1.27 所示。

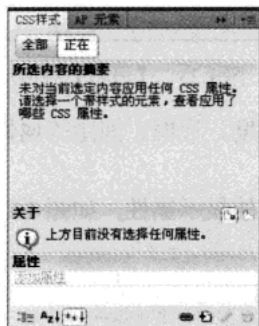


图 1.26 【CSS 样式】面板



图 1.27 【新建 CSS 规则】对话框

- 03 设置完毕，单击【确定】按钮关闭对话框，此时 Dreamweaver CS5 会自动打开【.red 的 CSS 规则定义】对话框，进入样式表编辑状态。在这里可以详细定义 red 类的规则。在【类型】分类选项中设置字体颜色为红色，如图 1.28 所示。

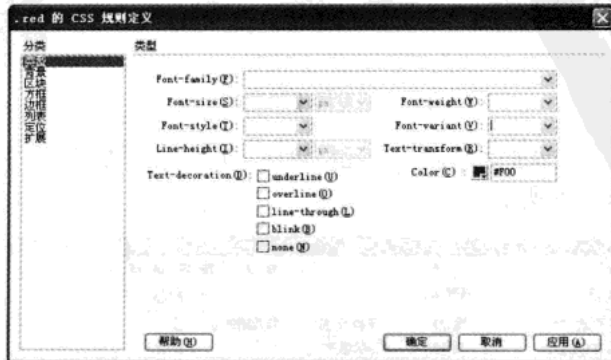


图 1.28 设置类样式的规则

- 04 设计完毕，单击【确定】按钮返回到编辑窗口，在页面中输入一行文本，然后选中该行文本，在【属性】面板中的【样式】下拉菜单中选择刚定义的样式类 red，如图 1.29 所示。
- 05 按 Ctrl+S 快捷键保存网页，然后按 F12 键在浏览器中预览，效果如图 1.30 所示。当然，用户也可以组合使用不同的样式，设计出不同的效果。



图 1.29 应用样式



图 1.30 应用样式效果

1.3.6 创建 AP 布局元素

AP 元素 (Absolutely Positioned Elements, 绝对定位元素) 和表格一样是网页中重要的布局元素，它可以包含文本、图像或其他 HTML 文档，可以使页面上的元素进行重叠和复杂的布局。Dreamweaver CS5 可以方便地在页面上创建 AP 元素，并精确地定位 AP 元素，可以对 AP 元素进行选择、移动、调整大小和对齐等操作。

将光标置于页面中，选择【插入】|【布局对象】|【AP Div】菜单命令，即可在文档中插入 AP 元素，如图 1.31 所示。

一个页面中可以绘制出很多 AP 元素，这些 AP 元素都会列在【AP 元素】面板中。AP 元素之间也可以相互重叠，选择【窗口】|【AP 元素】命令可以打开【AP 元素】面板，如图 1.32 所示。

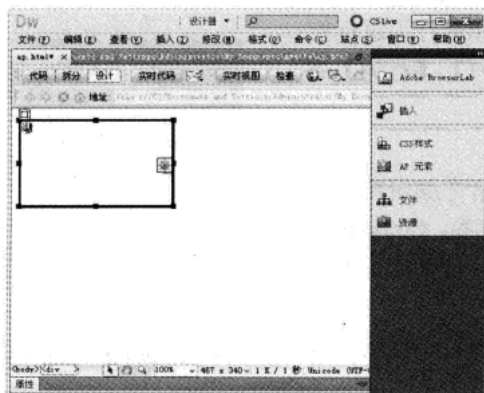


图 1.31 插入 AP 元素

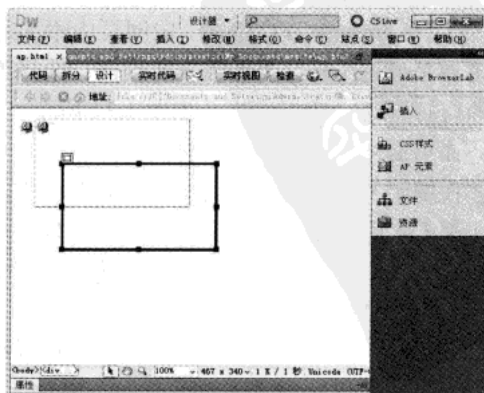



图 1.32 【AP 元素】面板

由于 AP 元素在网页布局上非常方便,很多用户喜欢通过 AP 元素来进行布局设计。Dreamweaver CS5 可以使用 AP 元素来创建布局。同时,AP 布局可以转换为表格,反之也可以。

1.3.7 创建框架

框架是除了表格、AP 元素之外另一个非常重要的网页布局工具。表格和 AP 元素是以对象为单位,而框架则以插入的网页为单位,它提供了一种较为固定的结构。

插入框架集的操作步骤如下。

- 01 把光标置于要插入框架集的编辑窗口中。
- 02 选择【插入】|【HTML】|【框架】命令,在【框架】子菜单中选择预定义的框架集。或者选择【窗口】|【插入】菜单项,打开【插入】面板,在工具栏中选择【布局】选项。单击【布局】工具栏右侧第 3 个【框架】按钮 ,打开框架下拉列表,列表中列出了 13 种预定义的框架集,从中选择一种即可。
- 03 选择一种框架集之后,会弹出【框架标签辅助功能属性】对话框,在该对话框中可以设置各个框架的标题,也可以保持默认设置。
- 04 单击【确定】按钮即可完成预定义框架集的创建,创建框架集后的编辑窗口如图 1.33 所示。

插入框架集后,可以选择【窗口】|【框架】菜单命令,打开【框架】面板,在该面板中可以控制框架集中各个框架。选中框架集或者框架后,就可以在【属性】面板中设置框架集或框架的属性。

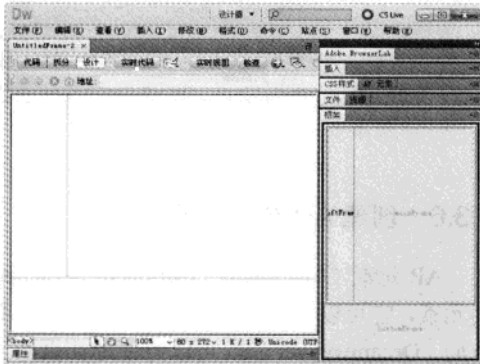


图 1.33 插入框架集效果

1.3.8 定义行为

行为是为响应某一具体事件而采取的一个或多个动作,当指定的事件被触发时,将运行相应的 JavaScript 程序执行相应的动作。所以在创建行为时必须先指定一个动作,然后再指定触发动作的事件。

在 Dreamweaver CS5 中,向网页中添加行为和对行为进行控制主要是通过【行为】面板来实现的。选择【窗口】|【行为】命令即可打开【行为】面板,如图 1.34 所示。

在页面中定义行为的具体步骤如下。


- 01 在编辑窗口中,选择要增加行为的对象,或者在编辑窗口底部的【标签选择器】中单击相应的页面元素标签,如<body>。
- 02 单击【行为】面板中的【加号】按钮 ,在打开的行为菜单中选择一种行为。
- 03 选择行为后,一般会打开一个参数设置对话框,根据需要设置完成即可。
- 04 单击【确定】按钮,此时在【行为】面板的列表中将显示添



图 1.34 【行为】面板

加的事件及对应的动作。

- 05** 如果要设置其他触发事件，可单击事件列表右边的下拉箭头，打开事件下拉菜单，从中选择一个需要的事件。

1.3.9 创建表单

表单是实现网页交互的基础。通过表单把用户的信息提交给服务器，实现信息的动态交互。表单由一个表单域和若干个表单对象组成。制作表单页面的第一步是要插入表单域。

在 Dreamweaver CS5 中插入表单域的操作步骤如下。


- 01** 将光标置于要插入表单的位置。
- 02** 选择【插入】|【表单】|【表单】菜单命令。也可以在【插入】面板中选择【表单】项，然后在【表单】选项中单击【表单】按钮。
- 03** 此时在编辑窗口中显示表单框，如图 1.35 所示。其中红色虚线界定的区域就是表单，它的大小随包含的内容多少自动调整，虚线不会在浏览器中显示。



图 1.35 插入表单域

- 04** 在该表单域中插入其他表单对象，实现动态交互。

当插入表单域或者表单对象之后，选中该表单域或者表单对象，然后可以在【属性】面板中设置其属性。

1.4 实践总结

本章主要介绍了 Dreamweaver CS5 的界面组成和基本操作。通过本章的学习，用户能够初步熟悉 Dreamweaver CS5 工具的用法，掌握 Dreamweaver CS5 的基本操作。在学习过程中建议用户反复练习，只有这样才能掌握 Dreamweaver CS5 的全部功能。

如果想更深入地学习 Dreamweaver CS5 详细功能，建议参考 Dreamweaver CS5 基础类图书，例如，中国电力出版社出版的《中文版 Dreamweaver CS5 标准教程》，该书对 Dreamweaver CS5 所有功能都进行了全面详细的介绍。此外，用户也可以查看 Dreamweaver CS5 的【帮助】菜单。

第 2 章

Dreamweaver

搭建动态网站运行环境

本章将讲解如何搭建 ASP 动态网站运行环境。帮助用户在本地计算机上模拟远程服务器组建小型动态网站运行环境，以实现上机学习 ASP 应用开发的目的。

构建动态网站环境比较复杂，涉及多方面知识和技术，例如，Windows IIS 组件、数据库操作组件 ADO、数据库管理系统（如 Access）、Dreamweaver CS5 网页设计、开发和网站管理等。为了方便用户参考，本书内容将基于表 2.1 所示的参数环境（或更高级环境）进行讲解和上机练习。

表 2.1 动态网站开发和运行环境参考列表

参 数	版 本
操作系统 (OS)	Windows 2000 及更高版本 (本书参考为 Windows XP)
Web 服务技术 (Web Service)	IIS 5.0 及更高版本 (本书参考为 IIS 5.1)
服务器端脚本环境 (Web Script Server)	ASP (本书参考为 ASP 3.0)
数据库管理系统 (ODBC)	Access 2000 及更高版本 (本书参考为 Access XP，并同时提供 Access 2007 和 SQL Server 数据库连接参考)
服务器端脚本语言 (Server Script)	VBScript (本书参考为 VBScript 5.6)
Web 设计和开发工具 (Design Tool)	Dreamweaver CS3 及更高版本 (本书参考为 Dreamweaver CS5)
网站测试、上传和管理工具 (Site Manager)	Dreamweaver CS3 及更高版本 (本书参考为 Dreamweaver CS5)

技术要点

- 了解动态网站运行框架
- 创建 IIS 服务器
- 配置 IIS 服务器
- 定义本地站点
- 定义 FTP 站点

2.1 认识动态网站架构

准备学习制作动态网页时，读者都会存在着这样的疑问：什么是动态网站，动态网站与静态网页有什么联系和区别，动态网站是如何工作的，如何开发动态网站等？

WWW (World Wide Web, 万维网) 技术是构建在浏览器/服务器 (B/S) 架构和 HTTP 协议基础上的，其运行模式可以概括为：请求→处理→响应，B/S 运行原理示意如图 2.1 所示。

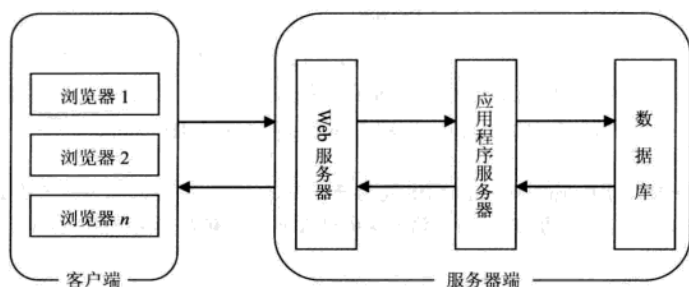


图 2.1 B/S 架构示意图

在图 2.1 所示的 B/S 架构示意图中，每个箭头都表示一个过程，这些过程具体说明如下。

过程①：表示客户端（本地计算机）浏览器（如 Internet Explorer）提交表单信息或者 URL 参数到服务器，向服务器提出服务请求。

过程②：表示 Web 服务器把用户请求交给执行具体任务的应用程序服务器进行具体分析和处理。至于对提交的信息如何处理，则由网站开发人员编写的应用程序来决定反馈到浏览器的内容。



提示

Web 服务器和应用程序服务器可以在同一台计算机上运行，也可以分开独立运行于不同服务器上，这主要根据服务的业务量来确定。

过程③：表示应用程序服务器如果需要访问数据库，实现后台数据操作，则需要提交 SQL 语句给 DBMS 管理系统进行处理。如果需要数据库进行访问，开发人员需要利用应用程序服务器所提供的接口对数据库进行访问。

过程④：表示从数据库中获取操作信息或查询记录。

过程⑤：表示把处理的结果返回到 Web 服务器。

过程⑥：表示将生成的静态网页反馈给客户端浏览器。

一般来说，在 Web 服务器上可以通过多种技术途径来实现动态网站，最常见的技术包括 CGI、JSP、ASP/ASP.NET 和 PHP 等。

基于 Internet 的浏览器/服务器 (B/S) 架构系统目前已经成为行业标准，它具有如下优点。

- 运行比较稳定。由于服务程序一般都运行在服务器上，客户端不需要运行服务程序，因此，只要服务器能正常运行，整个系统就会稳定运行。
- 方便远程维护。由于是基于 Internet 技术，通过上网便可对用户技术支持，而不需要上门服务。
- 客户端要求比较低。计算机的速度非常快，对于新的软件系统，往往因为难以及时升级硬件而不能使用。而基于 Internet 的系统，只要计算机可以上网，就可以使用系统。
- 方便使用。一般来说，只要用户能够上网，无论在世界什么地方、什么时间，都可以随时随地使用系统，而不像局域网那样受地域限制。
- 便于拓展服务。基于 Internet 技术优势，更容易开展符合网络传播的各种信息服务，如电子商务、信息查询、信息搜索和各种个性化服务项目。

2.2 安装 IIS 组件

IIS 是 Internet Information Services 的简称，意为 Internet 信息服务。Internet 信息服务可

以帮助用户在 Internet（互联网）或者 Intranet（局域网）上非常容易地发布信息。IIS 包含许多管理网站和 Web 服务器的功能。用户可以利用它创建并配置灵活的 Web 应用程序。

在默认情况下，Windows XP Professional 系统不会安装 Internet 信息服务，而是需要用户通过【控制面板】中的【添加或删除程序】应用程序安装 IIS 附加组件。



提示

Windows XP Professional 的 IIS 是为开发 Web 服务的个人、家庭或者办公用户而设计的组件。IIS 仅能同时为 10 个客户端连接提供服务，没有充分挖掘 Windows Server 服务器的全部功能。

如果用户已经安装了 IIS，可以在浏览器的地址栏中输入 <http://localhost/iisHelp/>，然后按 Enter 键可以查看相关 IIS 产品文档。

下面以 Windows XP 为例介绍 IIS 的安装，操作步骤如下。

- 01 在 Windows XP 面上，选择【开始】|【控制面板】命令，在【控制面板】窗口中单击【添加/删除程序】图标，如图 2.2 所示。
- 02 打开【添加或删除程序】对话框，在【添加或删除程序】对话框的左侧项目栏中单击【添加或删除 Windows 组件】图标，如图 2.3 所示。



图 2.2 【控制面板】窗口



图 2.3 【添加或删除程序】对话框

- 03 打开【Windows 组件向导】对话框，在该对话框中显示了可供安装的组件，有 Internet Explorer、Internet 信息服务（IIS）、MSN Explorer 等。选中【Internet 信息服务（IIS）】选项，如图 2.4 所示。
- 04 单击【详细信息】按钮，可以查看或选择所需安装的组件，如图 2.5 所示。然后按照向导提示操作即可，中间还会要求用户插入 Windows XP 安装盘。



图 2.4 【Windows 组件向导】对话框

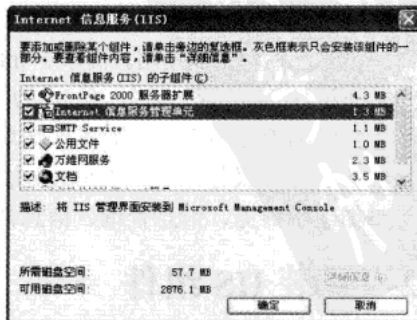


图 2.5 【Internet 信息服务（IIS）】对话框

- 05 安装完成后, 选择【开始】|【控制面板】命令, 在打开的【控制面板】窗口中双击【管理工具】图标, 打开【管理工具】面板, 如图 2.6 所示。
- 06 双击【Internet 服务管理器】图标, 启动【Internet 信息服务】窗口, 如图 2.7 所示。在【Internet 信息服务】窗口中可以建立和管理 IIS。

02



图 2.6 【管理工具】面板



图 2.7 【Internet 信息服务】窗口

也可以使用 Windows XP 安装盘进行安装。插入安装盘后, 会自动打开一个窗口, 如图 2.8 所示。然后按照提示逐步安装, 方法同上。

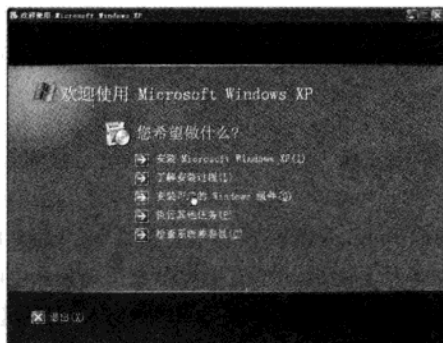


图 2.8 Windows XP 安装窗口页面

要卸载 IIS 组件, 可以按上面操作步骤, 在其中取消对【Internet 信息服务 (IIS)】选项的勾选, 然后继续执行后面操作, 此时系统会自动删除 IIS 组件。卸载 IIS 之后, 下列包含用户信息的目录仍将保留在计算机上: \inetpub、\systemroot\Help\IisHelp、\systemroot\System32\Inetsrv。

2.3 测试 IIS 服务器

安装 IIS 服务之后, 用户可以测试 IIS 服务器, 以验证 IIS 服务器是否安装成功。

2.3.1 在本地测试 IIS

打开网页浏览器, 例如 Microsoft Internet Explorer。在浏览器的【地址】文本框中输入服务器主目录的 URL。例如, “http://localhost/”, 如果显示 Microsoft 默认的 IIS 欢迎页面, 如图 2.9 所示, 则说明已经正确安装, 否则 IIS 没有正确安装。



图 2.9 IIS 欢迎页面

默认情况下, IIS 在主目录下以名为 default.htm、default.asp 或 iisstart.asp 的主页形式存放。如果浏览器返回错误信息, 请根据错误信息参阅疑难解答表寻求帮助。

2.3.1 在远程测试 IIS

如果已经连接到互联网，可以在本地测试远程服务器的状态。例如，在 IE 浏览器的地址栏中键入服务器主目录 URL。例如，“<http://www.css8.cn/>”。默认情况下，IIS 在主目录下以名为 default.htm、default.asp 或 iisstart.asp 的网页显示为主页，当访问远程站点时，首先会看到这些主页页面。

2.4 配置 IIS 服务环境

安装完毕，用户就可以打开 Internet 信息服务管理器了。选择【开始】|【控制面板】|【管理工具】命令，打开【管理工具】面板，在该面板中双击【Internet 服务管理器】图标，打开【Internet 信息服务】窗口，如图 2.7 所示。也可以右击 面上的【我的电脑】图标，在弹出的菜单中选择【管理】命令，打开【计算机管理】窗口。展开窗口左侧的【服务和应用程序】目录项，选择【Internet 信息服务】选项，在右侧的窗口中同样可以使用 IIS 服务，如图 2.10 所示。

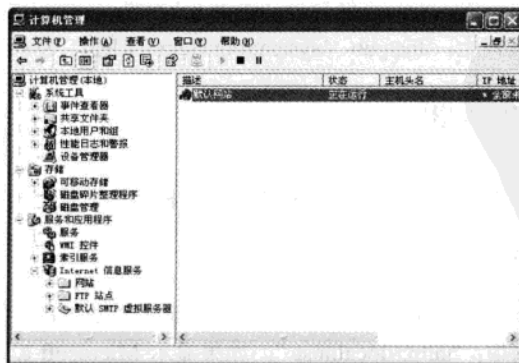


图 2.10 【计算机管理】面板

2.4.1 配置 WWW 服务属性

在【Internet 信息管理】窗口中，展开左侧网站目录树，右击【默认站点】项，在弹出的菜单中选择【属性】命令，打开【默认网站属性】对话框，如图 2.11 所示。

- 【网站】选项卡：配置网站的 IP 地址、TCP 端口、连接以及是否启用日志记录。
- 【ISAPI 筛选器】选项卡：安装 ISAPI 筛选器，并设置 ISAPI 筛选器的状态、名称和优先级，ISAPI 筛选器是系统 DLL 程序，在 HTTP 请求事件时被触发。
- 【主目录】选项卡：设置站点资源信息，包括来源、本地路径、读写权限和应用程序设置。
- 【文档】选项卡：设置启用默认文档。
- 【目录安全性】选项卡：设置匿名访问和认证控制、IP 地址以及域名验证和安全通信。
- 【HTTP 头】选项卡：设置启动内容过期选项。
- 【自定义错误】选项卡：自定义错误信息。
- 【服务器扩展】选项卡：调整服务器性能、安全设置和邮件选项。



图 2.11 【默认网站属性】对话框

2.4.2 建立 WWW 服务虚拟目录

虚拟目录并不是真实存在的网页目录，但虚拟目录与实际站点文件的目录之间存在一种映射关系。用户通过浏览器可以访问虚拟目录。默认状态下服务器路径是 C:\inetpub\wwwroot。利用虚拟目录，可以将数据分 保存在多个目录或计算机上，方便站点的管理和维护。此外，浏览者不知道网页在服务器中的实际存放位置，也就不能使用该信息修改文件，这在一定程度上保证了站点的安全。

在 XP 上建立虚拟目录的具体操作步骤如下。

- 01 在【Internet 信息服务】窗口中，展开左侧的【网站】目录树，右击【默认网站】选项，在弹出的菜单中选择【新建】|【虚拟目录】命令，如图 2.12 所示，
- 02 打开【虚拟目录新建向导】对话框，如图 2.13 所示。



图 2.12 选择【虚拟目录】命令



图 2.13 【虚拟目录新建向导】对话框

- 03 单击【下一步】按钮，进入【虚拟目录别名】对话框，输入用于获得 Web 虚拟目录访问

权限的别名，例如“mysite”，如图 2.14 所示。

- 04** 单击【下一步】按钮，进入【虚拟内容目录】对话框，在【目录】文本框中输入网站内容所在硬盘上的路径，也可以单击【浏览】按钮来选择合适的文件夹，例如“F:\mysite”，如图 2.15 所示。

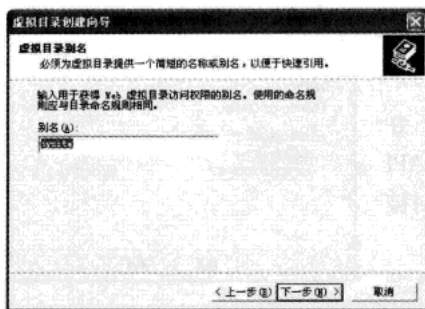


图 2.14 设置虚拟目录访问别名

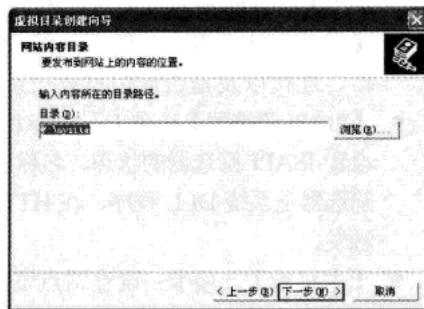


图 2.15 设置虚拟目录访问路径

- 05** 单击【下一步】按钮，在【访问权限】对话框中可设置权限，如图 2.16 所示。如果没有什么特殊要求就可以使用默认的设置。单击【下一步】按钮，在打开的提示对话框中单击【完成】按钮，完成虚拟目录的设置。

设置完毕后，在【Internet 信息服务】窗口的左侧就会出现一个名为 mysite 的虚拟目录图标，如图 2.17 所示。

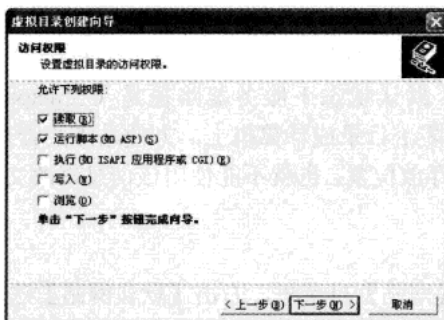


图 2.16 设置虚拟目录的访问权限



图 2.17 创建的虚拟目录文件夹

2.4.3 配置 FTP 服务属性

FTP（文件传输协议）是一个面向文件的传输协议，用于服务器和浏览器之间进行文件的传输，虽然 WWW 服务和 FTP 服务有许多相同功能，但是 FTP 协议能够保证数据传输的可靠性，仍然被广泛使用。

安装 FTP 之后，默认的路径为 C:\inetpub\ftproot。在【Internet 信息服务】窗口中，展开左侧的【FTP 站点】目录树，右击【默认 FTP 站点】选项，在弹出的菜单中选择【属性】命令，打开【默认 FTP 站点属性】对话框，如 2.18 所示。

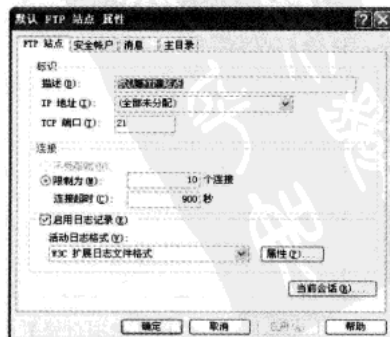


图 2.18 【默认 FTP 站点属性】对话框

- 【安全账户】选项卡：设置允许匿名连接、FTP 站点操作员等安全控制。
- 【消息】选项卡：设置 FTP 站点的标题、进入和退出、最大连接数的消息。
- 【主目录】选项卡：设置用户权限、FTP 主目录及目录列表样式。

2.4.4 建立 FTP 服务虚拟目录

和定义 WWW 服务虚拟目录一样，用户也可以定义 FTP 服务虚拟目录，为用户设置其他可用的目录，而不仅局限于 FTP 缺省目录下访问 FTP 服务器的文件。在 IIS 上提供了建立虚拟目录的方法，其方法与建立网站的虚拟目录相似。右击【默认 FTP 站点】选项，在弹出的菜单中选择【新建】|【虚拟目录】命令，然后根据向导操作即可。

2.5 在 Dreamweaver 中定义站点

安装 IIS 之后，在默认状态下，用户的计算机就具备了服务器运行环境（当然可以对 IIS 进行各种更复杂的配置，实现更安全、更高效的服务运行环境）。此时就可以在其中定义和测试站点了。由于 Dreamweaver 提供了超强的站点管理功能，因此本节重点介绍在 Dreamweaver 中如何定义各种类型站点的方法。

2.5.1 认识站点及其规划

站点也称网站，它是互联网中的一个节点，是一系列通过超链接相互联系的网页集合，站点内文件包括首页和若干页面。它们被分为不同级别，其中首页是 1 级页面，当访问者访问网站时会首先打开首页。由于网站内网页众多，根据内容可能还需要进行分类，形成 2 级页面，以此类推，整个网站就形成了金字塔的网状结构。

在 Dreamweaver CS5 中，站点既可以表示位于远程服务器端的远端站点，也可以表示位于本地计算机上的本地站点。一般来说，用户应首先在本地计算机上构建本地站点，设计合理的站点结构，使用合理的组织形式来管理站点中的文件，并对站点进行必要的测试。在一切都准备好之后，再将站点上传到远程服务器上，供浏览者访问。

在创建站点之前，一般应做好规划，解决好网站建设的基本问题。

- 如何设计网站的结构。
- 文件的组织与管理。
- 存储信息的方法，即数据库和文件系统设计。
- 站点维护设计，即后台管理设计。

在规划站点结构时，一般用文件夹来合理构建站点的结构。首先为站点创建一个根文件夹（根目录），然后在该文件夹中创建多个子文件夹，再将网页文件分门别类存储到相应的文件夹中。

在命名文件夹或文件时，使用合理的文件名非常重要，特别是在网站的规模很大时。文件名应该简洁易懂，让人看了就能够知道网页表述的内容。如果不考虑那些使用不支持长文件名的操作系统，那么可以使用长文件名来命名文件，以充分表述文件的含义和内容。

应避免使用中文文件名，因为很多服务器使用的是英文操作系统，不能很好地解析中文文件名；而且浏览网站的用户也可能使用英文操作系统，中文的文件名称同样可能导致浏览错误或访问失败。如果对英文不熟悉，可以使用汉语拼音作为文件名。同时，有些操作系统是区分文件大小写的，如 UNIX 操作系统。因此，建议在构建的站点中全部使用小写的文件名称。

2.5.2 定义本地站点

在个人计算机上安装了 Internet 信息服务 (IIS) 程序, 实际上就是将本地计算机构建成为一个真正的远程服务器, 但在真正使用之前还需要定义本地站点。本书后面实例操作都是在这个基础上进行的。

定义本地站点的具体操作步骤如下。

- 01 启动 Dreamweaver CS5, 选择【站点】|【新建站点】命令, 打开【未命名站点 1】的站点定义为对话框。
- 02 在【站点名称】文本框中输入站点名称, 如 mysite, 在【本地站点文件夹】文本框中设置站点在本地文件中的存放路径, 可以直接输入, 也可以用鼠标单击右侧的【选择文件】按钮选择相应的文件夹。
- 03 单击【高级设置】选项卡, 展开高级设置选项, 在左侧的选项列表中单击【本地信息】选项。然后在【本地信息】对话框中设置本地信息, 如图 2.19 所示。
 - 【默认图像文件夹】文本框: 设置默认的存放站点图片的文件夹。但是对于比较复杂的网站, 图片往往不仅仅只存放在一个文件夹中, 因此可以不输入。
 - 【链接相对于】选项: 定义当在 Dreamweaver CS5 为站点内所有网页插入超链接时是采用相对路径, 还是绝对路径, 如果希望是相对路径, 则可以勾选【文档】单选按钮; 如果希望以绝对路径的形式定义超链接, 则可以勾选【站点根目录】单选按钮。
 - 【Web URL】文本框: 输入网站的网址, 该网址能够供链接检查器验证使用绝对地址的链接。在输入网址的时候需要输入完全网址, 例如, <http://localhost/msite/>。该选项只有在定义动态站点后有效。
 - 【区分大小写的链接】复选框: 选中该复选框可以对链接的文件名称大小进行区分。
 - 【启用缓存】: 选中该复选框可以创建缓存, 以加快链接和站点管理任务的速度, 建议用户要选中。

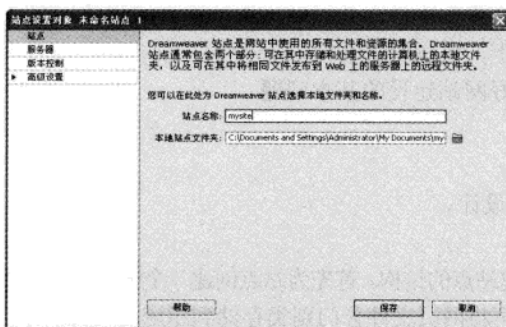


图 2.19 定义本地信息

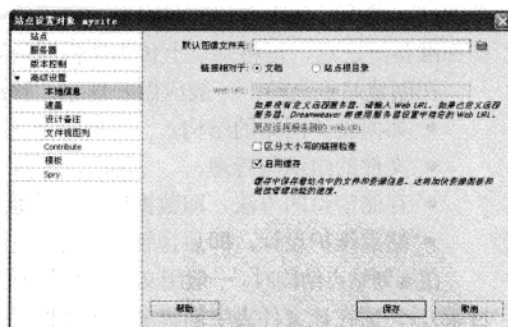



图 2.20 定义远程信息

2.5.3 定义动态站点

为了方便学习, 本节将介绍如何建立一个 ASP 技术、VBscript 脚本的动态网站, 本书后面章节实例都是在这样的动态网站上制作运行的。如果用户熟悉其他服务器技术或脚本语言, 也可以按这种方法建立其他类型的动态网站。操作步骤如下。

- 01 用户应该根据前面章节介绍的方法建立一个站点虚拟目录，用来作为服务器端应用程序的根目录，然后在本地计算机的其他硬盘中建立一个文件夹作为本地站点目录。建议建立的两个文件夹名称最好相同。用户也可以在默认站点 C:\inetpub\wwwroot\内建立一个文件夹作为一个站点的根目录，但这种方法有很多局限性，ASP 的很多功能无法实现，所以不建议使用这种简单方法建立服务器站点。
- 02 在 Dreamweaver CS5 中，选择【站点】|【新建站点】命令，打开【站点设置】对话框，单击【服务器】选项，切换到服务器设置面板。
- 03 在【服务器】选项面板中单击  按钮，如图 2.21 所示。显示增加服务器技术面板，在该面板中定义服务器技术，如图 2.22 所示。

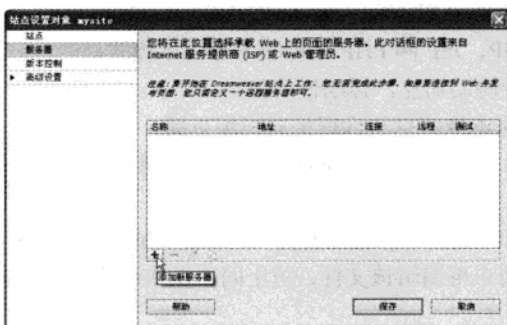


图 2.21 增加服务器技术

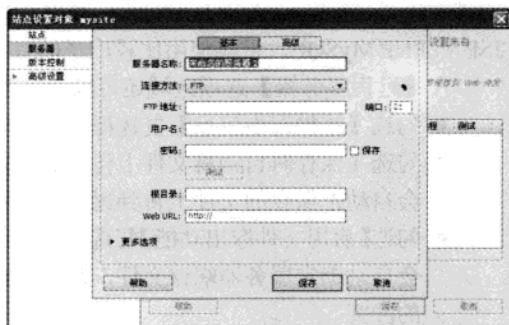


图 2.22 定义服务器技术

- 04 在【基本】选项卡中设置服务器基本信息，如图 2.23 所示。

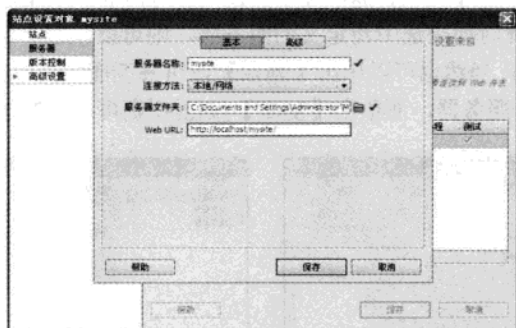



图 2.23 定义基本信息

- (1) 在【服务器名称】文本框中输入站点名称，如“mysite”。
- (2) 在【连接方式】下拉列表框中选择【本地/网络】选项。实现在本地虚拟服务器中建立远程连接，也就是设置远程服务器类型为在本地计算机上运行网页服务器。其他几个选项说明如下。

- FTP: 使用 FTP 连接到 Web 服务器。该类型在实际网站开发中比较常用，其中涉及很多方法和技巧。
- WebDAV: 该选项表示基于 Web 的分布式创作和版本控制，它使用 WebDAV 协议连接到网页服务器。对于这种访问方法，必须有支持该协议的服务器，如 Microsoft Internet Information Server (IIS) 6.0 和 Apache Web 服务器。

- **RDS:** 该选项表示远程开发服务, 使用 RDS 连接到网页服务器。对于这种访问方式, 远程文件夹必须位于运行 ColdFusion 服务器环境的计算机上。

(3) 在【服务器文件夹】文本框中设置站点在服务器端的存放路径, 可以直接输入, 也可以用鼠标单击右侧的【选择文件】按钮  选择相应的文件夹。为了方便管理, 可以把本地文件夹和远程文件夹设置相同的路径。





(4) 在【Web URL】文本框中输入 HTTP 前缀地址, 也可以先暂时不管, 等设置测试服务器之后, Dreamweaver CS5 会自动设置, 其他选项可以保持默认值。

05 在【站点设置】对话框中选择【高级】选项, 设置服务器的其他信息, 如图 2.24 所示。

在【服务器模型】下拉列表中选择 ASP VBScript 技术。服务器模型用来设置服务器支持的脚本模式, 包括无、ASP JavaScript、ASP VBScript、ASP.NET C#、ASP.NET VB、ColdFusion、JSP 和 PHP MySQL。目前使用比较广泛的有 ASP、JSP 和 PHP 三种服务器脚本模式。

在【远程服务器】选项区域还可以设置各种协助功能, 详细说明如下。

- 勾选【维护同步信息】复选框, 可以确保本地信息与远程信息同步更新。
- 勾选【保存时自动将文件上传到服务器】复选框, 可以确保在本地保存网站文件时, 会自动把保存的文件上传到远程服务器。
- 勾选【启用文件取出功能】复选框, 则在编辑远程服务器上的文件时, Dreamweaver CS5 会自动锁定服务器端该文件, 禁止其他用户再编辑该文件, 防止同步操作可能会引发的冲突。
- 在【取出名称】和【电子邮件地址】对话框中输入用户的名称和电子邮件地址, 确保网站团队内部即时进行通信, 相互沟通。

06 设置完毕, 单击【保存】按钮, 返回【站点设置】对话框, 这样就可建立一个动态网站, 如图 2.25 所示。此时如果选中新定义的服务器, 则可以单击下面的【编辑】按钮  重新设置服务器选项。当然也可以单击【删除】按钮  删除该服务器, 或者单击【增加】按钮  再定义一个服务器, 而单击【复制】按钮  则可以复制选中的服务器。

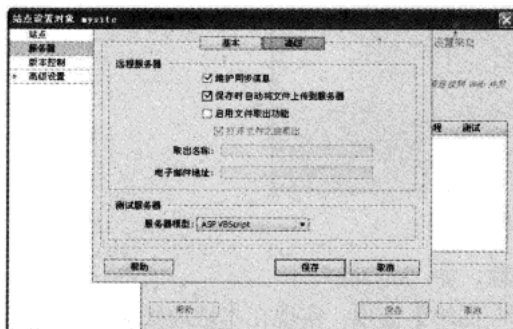


图 2.24 定义高级信息

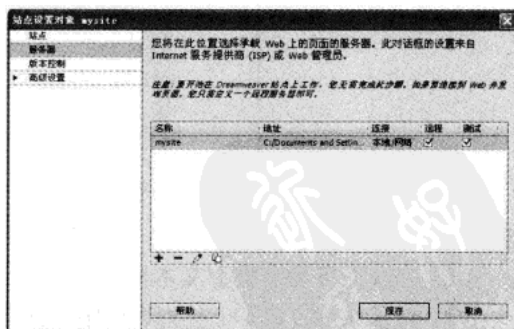


图 2.25 定义测试服务器信息

07 选择【站点】|【管理站点】命令, 打开【管理站点】对话框, 用户就可以看见刚刚建立的动态站点, 如图 2.26 所示。


08 选择【窗口】|【文件】命令, 或者按 F8 键, 打开【文件】面板。单击【文件】下拉列表右侧的向下三角按钮 , 在打开的下拉列表中选择刚建立的 mysite 动态网站, 这时就可以打开 mysite 站点, 如图 2.27 所示。



图 2.26 定义的站点



图 2.27 启动站点

这样，用户就可以在该站点下建立不同文件夹和各种类型的网页文件了。要注意 ASP 动态网页的扩展名为 .asp。本书后面的实例都是在这样的环境下建立并运行的，否则网页浏览器不能识别和显示。

2.5.4 定义 FTP 站点

IIS 功能非常强大，除了提供 WWW 服务之外，还提供 FTP 服务，利用它很容易架设一个功能卓越的 FTP 服务器。通过 FTP 服务器可以把本地站点的文件和文件夹内容上传到远程服务器上。

定义 FTP 站点的具体步骤与定义 WWW 服务站点的的基本方法相同。不同的是在上一节操作步骤的第 4 步中，选择【访问】的方式为“FTP”选项，如图 2.28 所示。



图 2.28 定义 FTP 站点

- **【FTP 地址】**文本框：设置 FTP 主机的名称，例如，ftp.mindspring.com。请输入完整的主机名，并且不要附带其他任何文本。特别是不要在主机名前面加上 FTP 协议名。如果不知道 FTP 主机，可以联系 Web 站点 管服务商索取。如果知道主机名的 IP，也可以直接输入 IP 地址，例如，221.195.63.111，这样能够更快速地连接到服务器。
- **【用户名】**和**【密码】**文本框：输入用于连接到 FTP 服务器的用户名和密码。该登录名和密码由 Web 站点 管服务商提供，也可以通过 Web 站点 管服务商的后台管理网站进行修改。

- **【根目录】**文本框：在该文本框输入用户在远程站点上存储公开显示的文档的主机目录（文件夹）。如果不能确定应输入哪些内容作为主机目录，请与服务器管理员联系或将文本框保留为空白。在有些服务器上，根目录就是用户首次使用 FTP 连接到的目录。若要确定这一点，请连接到服务器。如果出现在**【文件】**面板**【远程文件】**视图中的文件夹具有像 public_html、www 或用户的登录名这样的名称，它可能就是应该在**【主机目录】**文本框中输入的目录。
- **【Web URL】**文本框：输入网站的网址，该网址能够供链接检查器验证使用绝对地址的链接。在输入网址的时候需要输入完全网址，例如 http://localhost/msite/。
- 单击**【测试】**按钮可以测试登录名和密码是否正确，如果连接成功会显示如图 2.29 所示的提示对话框。默认情况下，Dreamweaver 会保存密码。如果用户希望每次连接到远程服务器时 Dreamweaver 都提示输入密码，可以取消选择**【保存】**复选框。
- 如果防火墙配置要求使用被动式 FTP，用户可以单击**【更多选项】**选项，展开更多选项，如图 2.30 所示。在这里如果选择**【使用被动式 FTP】**复选框，一般服务商都会提供说明。**【使用被动式 FTP】**能够使用被动方式建立本地软件与 FTP 服务器的连接，而不是请求远程服务器来建立它。如果不能确定是否使用被动式 FTP，可以询问系统管理员。

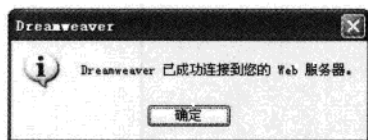


图 2.29 提示 FTP 服务连接成功



图 2.30 更多选项

- 如果使用支持 IPv6 的 FTP 服务器，请选择**【使用 IPv6 传输模式】**复选框。随着 Internet 协议第 6 版 (IPv6) 的发展，EPRT 和 EPSV 已分别替代 FTP 命令 PORT 和 PASV。因此，如果用户正试图连接到支持 IPv6 的 FTP 服务器，必须为数据连接使用被动扩展 (EPSV) 和主动扩展 (EPRT) 命令。

2.6 实践总结

本章重点介绍了动态网站的构建与配置。对于个人用户来说，通过简单的操作就可以在本地计算机上建立服务器环境，把自己的电脑设置为服务器，如果用户申请一个 IP 地址，也可以在网上申请一个免费的动态 IP 地址，就可以在网上让任何人浏览你的计算机了。同时，本章还介绍了在 Dreamweaver CS5 中定义动态站点的步骤和方法，要想深入学习服务器的管理方法，还需要参考更专业的网管员书籍。

动态网页与静态网页有着本质的区别。对于静态网页而言,访问者只能被动地接收网页上的信息,访问同一个网页的用户所看到的页面完全相同;而动态网页则能够根据用户需求,动态地为用户提供服务。从技术角度看,动态网页为浏览器与 Web 服务器的实时交互提供了一种途径。

对于初学者来说,学习动态网页开发也相对要困难些,因为动态网页不如静态网页设计那样简单易学。静态网页只需要读者安装一款网页编辑器和一种网页浏览器就可以实现设计、预览等工作,但动态网页需要在服务器环境中才能够正常运行,所需要的技术也比较复杂,它涉及服务器环境的配置、数据库连接操作、服务器技术的应用,以及脚本语言如何控制页面动态变化。动态网页环境类型有很多种,本书主要以 ASP 服务器技术为基础展开讲解。

技术要点

- 了解 ASP 服务器技术
- 熟悉 ASP 对象
- 灵活使用 Response 和 Request 对象
- 掌握 Session 和 Cookie 对象
- 能熟练操纵 Server 对象
- 能够使用 FileSystemObject 对象操作系统文件

3.1 ASP 简介

ASP 是 Microsoft 公司开发的服务器端脚本环境运行技术,使用 ASP 并结合 HTML 语言、VBScript 脚本语言和 ASP 组件可以创建交互式 Web 页和基于 Web 的功能强大的应用程序。

3.1.1 认识服务器技术

要开发动态网站,读者需要先指定一种服务器端技术,目前实现动态网页的服务器技术主要包括 CGI、ASP/ASP.NET、PHP 和 JSP 等。

1. CGI

CGI (Common Gateway Interface) 是一种通用的网关接口,是外部程序与网页服务器之间的标准编程接口。用户可以使用不同的语言编写 CGI 程序,例如, Visual Basic、Delphi 或 C/C++ 等。可以将已经写好的 CGI 代码放在网页服务器的计算机上运行,再将其运行结果通过网页服务器传输到客户端的网页浏览器上。事实上,由于 CGI 技术比较低级,普通用户在

编写代码时会比较困难，而且效率低，而且每一次修改程序都必须将 CGI 程序编译成可执行文件，因此现在很少再有用户使用。

2. ASP

ASP (Active Server Pages) 是在 CGI 技术基础上由微软公司开发的一种快速、简便的服务器技术，由于它的学习门槛比较低，初学者很容易学习，且功能强大，一推出就受到了众多专业人士的好评。凭借微软公司强有力的技术支持，ASP 可以说是时下网站建设中最为流行的技术之一。

3. ASP.NET

ASP.NET 是微软公司在 ASP 基础上推出的一种服务器技术，它全面采用效率较高的、面向对象的方法来创建动态 Web 应用程序。在原来的 ASP 技术中，服务器端代码和客户端 HTML 混合、交织在一起，常常导致页面的代码冗长而复杂，程序的逻辑难以理解，而 ASP.NET 就能很好地解决这个问题，而且能与浏览器独立，且可以支持 VB.NET、C#、VC++.NET、JS.NET 等 4 种编程语言。

4. PHP

PHP (Hypertext Preprocessor, 超文本预处理器) 是一种 HTML 内嵌式的语言，PHP 与微软的 ASP 很相似，都是一种在服务器端执行的嵌入 HTML 文档的脚本语言，语言的风格类似于 C 语言，现在被很多的网站编程人员广泛运用。由于 PHP 源代码是开放的，所有的 PHP 源代码事实上都可以得到。同时 PHP 技术又是免费的，因此深受一些用户欢迎。

5. JSP

JSP (Java Server Pages) 是 Sun 公司推出的网站开发技术，是将纯 Java 代码嵌入 HTML 中实现动态功能的一项技术。目前，JSP 已经成为 ASP 的有力竞争者。

与 ASP 技术非常相似，JSP 和 ASP 都是在 HTML 代码中嵌入某种脚本并由语言引擎解释执行程序代码，它们都是面向服务器的技术，客户端浏览器不需要任何附加软件的支持。两者最明显的区别在于 ASP 使用的编程语言是 VBScript 之类的脚本程序，而 JSP 使用的是 Java。此外，ASP 中的 VBScript 代码被 ASP 引擎解释执行，而 JSP 中的脚本在第一次执行时被编译成 Servlet 并由 Java 虚拟机执行，这是 ASP 与 JSP 本质的区别。

3.1.2 认识 ASP

ASP 是微软公司倾力开发的一套服务器端脚本环境，也就是说 ASP 本身并不是一种脚本语言，它只是提供了一种使镶嵌在 HTML 页面中的脚本程序得以运行的环境。如果要编写 ASP 程序或者建立 ASP 动态网站，读者还必须指定一种脚本语言，默认为 VBScript 脚本语言（有关 VBScript 脚本语言可以参阅第 4 章详细介绍）。

通过 ASP 并结合 HTML 网页、脚本语言和 ASP 组件，读者能够建立动态、交互式且高效的 Web 服务器应用程序。有了 ASP 就不必担心客户端的浏览器是否能运行用户所编写的代码，因为所有的程序都将在服务器端执行，包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后，服务器仅将执行的结果返回给客户端浏览器，这样就减轻了客户端浏览器的负担，大大提高了交互速度。

ASP 包含在 IIS 中，安装 Windows 的 IIS 组件之后，本地计算机即可支持 ASP 技术（可以参考第 2 章介绍了解 IIS 安装和配置方法）。采用 ASP 服务器技术，并结合数据库管理系统，读者能够建立起功能强大、信息丰富的动态网站。要更新网页内容时，只需查询数据库

即可, 如果对于站内搜索等查询应用, 这种方式的优越性就更为突出。

常说的“动态页面”还可以包括页面动态效果, 如 GIF 动画、客户端 JavaScript 小程序、JavaApplet 特效等, 而本章所指的动态网页是指页面内容的动态更新, 这种更新主要包括以下几个方面内容。

- 信息交互。即网页能够根据用户的不同操作而动态改变页面显示内容。
- 内容自动更新。即无须手动更新 HTML 文档, ASP 能够自动生成新的页面, 可以大大节省工作量。
- 因时因人而显示页面内容。即当不同的时间、不同的人访问同一张网页时会产生不同的页面显示效果。

3.1.3 ASP 特点

ASP 动态网页是用服务器端脚本、对象和组件扩展了的标准 HTML 网页, 使用 ASP 可以用动态内容创建网站。简单地说, ASP 具有如下几个重要特点。

- ASP 可以包含服务器端脚本。将服务器端脚本包含在 ASP 中就可以用动态内容创建网页。
- ASP 提供了几种内置对象。在 ASP 中使用内置对象可以使脚本功能更强。另外, 利用这些对象还可以从客户端浏览器中获得信息或者向客户端浏览器发送信息。
- 使用组件可以扩展 ASP 的功能。ASP 可以与标准的服务器端组件 (ActiveX) 捆绑在一起, 从而可以更方便地处理数据库。
- ASP 可以与数据库建立连接, 例如, SQL Server、Microsoft Access 等, 通过对数据库的操作建立功能强大的 Web 应用程序。

总之, ASP 是目前网页开发技术中最容易学习、灵活性最大的开发工具之一。最重要的是 ASP 拥有非常好的可扩充性。ASP 是用附加特性扩展了的标准 HTML 文件, 像标准的 HTML 文件一样, ASP 包含可被 Web 浏览器显示并解释的 HTML 标记。通常插入 HTML 文件中的 Java 小程序、JavaScript 代码、客户端 ActiveX 控件都可以插入到 ASP 中。

3.2 ASP 工作原理

ASP 采用 B/S 模型, 但其执行过程与静态 HTML 网页有所不同, 具体执行过程如下。



提示

B/S 模型就是 Browser/Server, 用户通过 WWW 浏览器访问服务器, 服务器运行业务逻辑, 从数据库提取数据并返回结果给用户。

- (1) 用户在浏览器的地址栏中输入 ASP 文件的 URL, 并回车触发这个 ASP 的申请。
- (2) 浏览器将这个 ASP 的请求发送给 Web 服务器。
- (3) Web 服务器接收这个申请并根据.asp 的扩展名判断这是 ASP 请求。
- (4) Web 服务器从服务器端计算机硬盘或内存中读取该文件的 ASP 源代码。
- (5) Web 服务器将这个文件的源代码发送到名为 Asp.dll 的动态链接库文件中。这是一个可执行文件, 它允许程序共享执行特殊任务所必需的代码和其他资源。
- (6) Asp.dll 可执行文件将 ASP 文件的源代码从头至尾执行, 并根据命令要求生成相应的 HTML 文件。

(7) 生成的 HTML 文件被送回客户端浏览器。

(8) 客户端浏览器解释这些 HTML 文件并将结果显示出来。

上述过程是一个简化的过程,但从中可以看出 ASP 与 HTML 有着本质的区别。对于 Web 服务器来说,HTML 不经过任何处理就被送到了客户端浏览器,而 ASP 中的每一个命令都首先要在服务器端执行并根据执行结果生成相应的 HTML 页面,再将 HTML 页面送给客户端浏览器。利用 ASP 的这种特性可以根据实际情况定制网页,在用户浏览器中显示不同的内容,即 ASP 可以根据需要动态地向客户端浏览器显示内容,例如,用户登录、网络搜索引擎等。

3.3 创建 ASP 文件

ASP 文件是以 .asp 为扩展名的文本文件,其中可以包含文本信息、HTML 标记、脚本代码。创建 ASP 文件非常容易,使用任何文本编辑器都可以创建,例如,Windows 记事本、写字板等。但建议读者使用 Dreamweaver 等专业网页编辑器以提高工作效率。



提示

读者也可以在 HTML 文件中添加脚本代码,并将该文件的扩展名 .htm 或 .html 替换为 .asp。

ASP 文件可以和 HTML 文件放置在相同的目录中。



提示

由于 ASP 文件需要服务器额外的处理,因此不要将所有的 HTML 页面都转换为 ASP 文件,这样会增加服务器的负担,应该只将包含脚本命令的文件转换为 .asp 文件。

3.3.1 编写 ASP 文件

下面来编写一个简单的 ASP 网页,看看 ASP 文件的代码形式和文件的执行方式。在编写代码之前,读者可以先配置 IIS 环境和用 Dreamweaver CS5 定义一个动态网站(有关详细操作可参考第 2 章内容介绍)。

01 在 Dreamweaver CS5 中新建一个文件,命名为 hello.asp (要注意文件的扩展名为 .asp,在 ASP 动态网站中新建文件的默认扩展名为 .asp)。

02 打开该文件,切换到【代码】视图,请输入下面代码:

```
<%@language="vbscript" codepage="936"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Hello World</title>
</head>
<body>
<%
for i = 6 to 1 step -1
%>
<h<%=i%>> Hello World! </h<%=i%>>
<%
next
%>
</body>
</html>
```

在 ASP 中,所有脚本命令都必须包含在定界符“<%”和“%>”之间,任何在这个定界

符中包含的内容都被认为是 ASP 脚本代码,读者可以在其中插入任何命令,只要这些命令遵循本页所定义的脚本语言的语法规则即可。

ASP 文件一般在第一行包含一些处理指令,处理指令的语法格式如下:

```
<%@ keyword %>
```

其中@是 ASP 处理指令的标识符,keyword 表示处理指令的关键字。ASP 处理指令为 ASP 提供处理 ASP 文件所需的信息。处理指令必须出现在 ASP 文件的第一行。读者可以在单个@指令中包含多个关键字,关键字/值对之间必须由空格分开。不要在等号(=)左右加入空格。例如,下面的指令设置了脚本语言和代码页:

```
<%@language="vbscript" codepage="936"%>
```



@命令在 ASP 文件中只能使用一次。

一般在 ASP 文件的首行要指定本页面所使用的脚本语言,例如,<%@language="vbscript" codepage="936"%>。

在上面的指令中,Language 关键字表示设置页面的脚本语言,这里将主脚本语言设置为 VBScript。ASP 的默认主脚本语言为 VBScript,读者也可以将任何一种具有脚本引擎的脚本语言设置为页面的主脚本语言。主脚本语言是用来处理在分界符“<%”和“%>”之间的命令的语言。

codepage 关键字可以设置页面的代码页,即脚本代码的字符编码,936 表示简体中文字符集。950 表示中文繁体字符集,932 表示日文字符集,65001 表示 UTF-8 字符集。

本页面使用 VBScript 作为脚本语言,因此 ASP 文件的脚本代码也将遵循 VBScript 脚本语言的语法规则。其中 For...Next 是一个循环语句,将循环显示所包含的内容(该内容可以是文本信息、HTML 源代码,或者是脚本命令)。

除了脚本语言外,ASP 也提供了很多指令。这是一些输出指令和处理指令。ASP 的输出指令<%= expression %>可以在客户端浏览器中输出表达式的值,expression 为任何有效的表达式。例如,在本例中使用的输出命令<%=i%>,它表示在浏览器中输出变量 i 的值,由于<%=i%>与<h>标记捆绑在一起,就很容易被浏览器解析为标题标记的一部分,因为变量 i 在服务器端执行,返回到客户端时已经是一个数字了。

这是一种信息输出的快捷方式,读者也可以调用 ASP 的 Response 对象的 Write 方法向客户端输出信息,例如:

```
<%Response.Write(i) %>
```

3.3.2 运行 ASP 文件

ASP 文件不能够直接被网页浏览器浏览访问,读者必须将创建的 ASP 文件保存到 Web 站点内(可参考第 2 章介绍方法配置 IIS,并建立动态网站,确保启用站点目录的脚本和执行权限,默认为开启),然后在 Dreamweaver CS5 中按 F12 键,在 IE 浏览器中的浏览,则效果如图 3.1 所示。

读者也可以直接把文件保存到 IIS 默认 Web 站点文件夹中,具体地址为: C:\Inetpub\wwwroot。



图 3.1 运行第 1 个动态网页

然后在网页浏览器的地址栏中输入默认站点的 URL:

```
http://localhost/hello.asp
```

按回车键即可在网页浏览器中预览 ASP 文件的运行效果。

如果建立了虚拟目录,则应该在 URL 中添加虚拟目录的名称,例如:

```
http://localhost/虚拟目录的名称/hello.asp
```

读者还可以在网页浏览器的地址栏中输入 IP 地址(一般本地计算机的 IP 地址为 127.0.0.1),例如:

```
http://127.0.0.1/hello.asp
```

或者

```
http://127.0.0.1/虚拟目录的名称/hello.asp
```



ASP 脚本命令都在服务端执行,而 HTML 源代码是在客户端浏览器中解析,所以当把脚本代码与 HTML 代码捆绑在一起时,不用担心会发生什么代码混乱现象。

3.4 ASP 对象概述

通过上面的介绍,读者对 ASP 已经有了一个基本的了解。如果想编写 ASP 应用程序,除了应该掌握一种脚本语言外(如 VBScript),读者还必须熟练掌握 ASP 的各种内置对象和常用组件。这些对象和组件都可以用来拓展 ASP 应用程序的功能。实际上,读者只有掌握了内置对象和组件,才能编写出功能强大的 ASP 应用程序。

ASP 内置对象能够帮助用户更容易地收集通过浏览器请求发送的信息、响应浏览器以及存储应用程序和用户信息。更简单地说,使用 ASP 内置对象可以方便客户端浏览器与服务器端计算机之间进行信息交流,以及应用程序中各类信息的存储和传递,从而使开发者摆脱了很多烦琐的工作。这些功能仅靠脚本语言是很难实现的。

在编程语言中,对象都具有方法、属性或者集合。其中对象的方法决定了用户可以用这个对象做什么事情,对象的属性可以读取描述对象状态的信息或者设置对象的状态,对象的集合包含了很多与对象有关系的键和值的配对,相当于一个二维数组。

ASP 的内置对象和组件十分类似。不过,当读者使用 ASP 时,两者之间还是存在着明显的差异。

- 首先,一个组件可能包含很多个对象。
- 其次,在使用组件之前,需要明确地创建一个实例,相当于类的实例化。而读者在使用内置对象时可以直接使用,不需要先创建对象的实例。在整个 Web 应用程序中内置对象的所有方法、属性以及集合都是可以自由访问的。



对象实际上就是一个复杂变量,使用时被调到内存中。

同时,ASP 还支持用户自定义组件类对象,要使用组件提供的对象,应创建对象的实例并为这个新的实例分配变量名。在 ASP 环境中可以使用 Server.CreateObject 方法创建对象的实例,例如:

```
Set db=Server.CreateObject("ADODB.Connection")
```

这里的变量 `db` 就是 ASP 程序创建的访问数据库的对象实例。

ASP 主要内置了 6 大对象，见表 3.1。

表 3.1 ASP 内置对象

内置对象	说 明
Request	负责服务器端从客户端浏览器接收信息
Response	负责把服务器端信息传递到客户端浏览器
Session	负责存储用户的私有信息，该信息只能对用户本人开放，可以在 Web 应用程序中反复使用
Application	负责存储应用程序的公共信息，该信息可以对所有人开放，可以在 Web 应用程序中反复使用
Server	负责控制 ASP 的运行环境
ObjectContext	可以用来控制 ASP 的执行。这种执行过程由 Microsoft Transaction Server (MTS) 来进行管理

3.5 Response 对象

Response 对象在 ASP 中负责将信息传递给客户端浏览器。Response 对象实现了动态响应客户端浏览器的请求，并将动态生成的响应结果返回到客户端浏览器中。使用 Response 对象可以动态创建 Web 页面、改变 HTTP 标题头，还可以自动将客户端浏览器页面重新定位到一个指定的页面中，另外，如果要向客户端浏览器写入 Cookies 信息时，Response 对象也是一种很好的工具。

Response 的语法格式如下：

```
Response.collection|property|method
```

其中，collection 表示 Response 对象的集合，property 表示 Response 对象的属性，method 表示 Response 对象的方法。三个参数只能选择其中的一个。

3.5.1 Response 对象的属性

Response 对象提供了很多属性，利用这些属性可以在服务器端操纵客户端浏览器的显示状态。Response 对象的具体属性见表 3.2。

表 3.2 Response 对象的属性

属 性	说 明
Buffer	设置 ASP 是否使用缓冲页输出
CacheControl	决定代理服务器是否能缓存 ASP 生成的输出
Charset	将字符集的名称添加到内容类型标题中
ContentType	指定响应的 HTTP 内容类型
Expires	在浏览器中缓存的页面超时前，指定缓存的时间
ExpiresAbsolute	指定浏览器上缓存页面超时的日期和时间
IsClientConnected	指示客户端是否与服务器断开
Pics	将 Pics 标记的值添加到响应的标题的 PICS 标记字段中
Status	服务器返回的状态行的值

1. Buffer

Buffer 属性用于设置页面输出时是否要用到缓冲区，ASP 3.0 版本中默认值为 True。当它为

True 时, ASP 会等到整个页面被执行完毕之后才会将结果输出到客户端浏览器上。语法格式如下:

```
Response.Buffer=Flag
```

其中 Flag 为布尔值。当 Flag 为 False 时, 表示不缓冲, 这时服务器在处理脚本的同时也把已处理的结果输出发送给客户端浏览器。

当 Flag 为 True 时, 表示使用缓冲页输出。也就是说只有当前页的所有服务器脚本处理完毕或者调用了 Flush 或 End 方法后, 服务器才将响应结果发送给客户端浏览器。

服务器将输出结果发送给客户端后就不能再设置 Buffer 属性。因此, 读者应该在 ASP 文件脚本代码前面设置 Response.Buffer 属性, 一般都放置在 @ 指令后面。

在没有缓冲输出的时候, 向客户端浏览器输出的内容会立即发送到浏览器, 处理的信息就无法再修改。设置了缓冲输出, 此时用户还可以根据实际情况利用 Clear 方法中途清除缓冲区中的数据, 并重新进行处理。例如:

```
<%@language="vbscript" codepage="936"%>
<%Response.Buffer=True%>
<html><Head>
<title>Buffer示例</title>
</head><body>
<%
  for i=1 to 50000
    response.write(i & "<br>")
  next
%>
</body></html>
```

在上面页面代码被服务器执行时, 整个页面的所有处理结果先存在服务器端计算机缓存区中直到脚本执行结束, 然后再同时发送到浏览器上, 此时会明显感觉浏览器有一个很长的等待过程。因此, 当 ASP 文件需要执行很长时间时, 为了避免用户长时间的等待, 建议设置没有缓冲输出。

2. CacheControl

CacheControl 属性用来设置 Web 服务器是否将 ASP 的处理结果暂时存放在代理服务器上, 如果客户端的浏览器没有设置, 则这个属性值不起作用。语法格式如下:

```
Response.CacheControl=Cache Control Header
```

CacheControl 的属性值有 Public 和 Private 两种, 为 Public 时表示 Web 服务器将 ASP 的处理结果暂时存放在代理服务器上, 后者则不会。默认情况下, 服务器不会将 ASP 执行结果放入缓存。

3. Charset

Charset 属性能够将字符集名称 (如 GB2312) 附加到 Response 对象中 content-type 标题的后面, 用来设置 Web 服务器发送给客户端的网页文件的字符编码。通俗地说, 使用 Charset 属性可以动态改变网页的字符编码。语法格式如下:

```
Response.Charset("字符集名称")
```

例如:

```
<% Response.Charset("ISO-LATIN-7") %>
```

能够将当前处理的网页文件的字符编码设置为拉丁文。

4. ContentType

ContentType 属性用来指定响应的 HTTP 内容类型。语法格式如下:

```
Response.ContentType="内容类型"
```

一般来说, ContentType 都是以“类型/子类型”的字符串来表示的,常用的有 Text/HTML、Image/Gif、Application/MSWord 等,默认为 Text/Html 类型。例如:

```
<% Response.ContentType = "text/HTML" %>
<% Response.ContentType = "image/GIF" %>
<% Response.ContentType = "image/JPEG" %>
```

5. Expires

Expires 属性用来设置 Web 页面保留在客户端浏览器缓冲区的时间长度,即设置网页的有效期。在有效期内,如果用户重新浏览该页面,就会显示缓冲区中的页面内容。语法格式如下:

```
Response.Expires=分钟数
```

如果分钟数设为 0,就不会在客户端浏览器保存数据,这样用户每次访问网页时,都必须重新向服务器发出请求并下载该页面。该属性必须放在<HTML>标记之前,否则会出错。

例如:

```
<%Response.Expires=0%>
```

在 ASP 文件中加入上面这一行代码,会要求用户每次请求都刷新页面,因为浏览器一收到页面就会过期。对于聊天室等提供即时信息的页面,设置该属性为 0 非常不错。

6. ExpiresAbsolute

ExpiresAbsolute 属性用于设置清除缓存于浏览器中数据的日期和时间。在未到期之前,若用户返回到该页,该缓存的页就显示。如果未指定时间,该主页在当天午夜到期。如果指定日期,则该主页在指定时间到期。语法格式如下:

```
Response.ExpiresAbsolute=日期时间
```

其中参数“日期”为指定页的到期日期,该值在符合 RFC-1123 日期格式的到期标题中发送。参数“时间”指定页的到期时间,该值在到期标题发送之前转化为 GMT 时间。

例如:

```
<% Response.ExpiresAbsolute=#May 31,2008 13:30:15# %>
```

7. IsClientConnected

IsClientConnected 属性是只读属性,用于判断客户端是否能与服务器相连。语法格式如下:

```
布尔值=Response.IsClientConnected()
```

当用户向服务器发出请求,如果用户离开了该站点,那么被请求的程序就没有必要再执行下去了,这时可以用下面代码终止程序执行:

```
<%
If not Response.IsClientConnected Then Response.End
%>
```

8. PICS

PICS 属性用来设置 PICS 标签,并把响应添加到标头(response header)。语法格式如下:

```
Response.PICS(PICS字符串)
```

例如：

```
<%
Response.PICS("PICS-1.1 <http://www.rsac.org/ratingv01.html> labels on " & chr(34) &
"1997.01.05T08:15-0500" & chr(34) & " until" & chr(34) & "1999.12.31T23:59-0000" & chr(34)
& " ratings (v 0 s 0 1 0 n 0))")
%>
```

9. Status

Status 属性用来设置 Web 服务器要响应的状态行的值。语法格式如下：

```
Response.Status="状态描述字符串"
```

HTTP 规格中定义了 Status 值。“状态描述字符串”由一个三位整数和一串说明文字组成，客户端可以根据这些代码和说明信息得到服务器端的执行情况。

例如：

```
<% Response.Status = "401 没有权限" %>
```

读者必须把该属性放在<HTML>标记之前，否则会出错。

3.5.2 Response 对象的方法

Response 对象可以使用的方法见表 3.3。

表 3.3 Response 对象的方法

方 法	说 明
AddHeader	设置 HTML 标题
AppendToLog	在该请求的 Web 服务器日志条目后添加字符串
BinaryWrite	将给出信息写入到当前 HTTP 输出中，并且不进行任何字符集转换
Clear	清除任何缓冲的 HTML 输出
End	停止处理.asp 文件并返回当前的结果
Flush	立即发送缓冲区中的输出信息
Redirect	将重指示的信息发送到客户端浏览器，尝试链接另一个 URL
Write	将变量作为字符串写入当前的 HTTP 输出中

1. Write

Write 方法用于将指定的数据写到当前的 HTTP 中输出，即把数据发送到客户端浏览器中。它是 Response 对象最常用的方法，也是在浏览器中动态显示信息的最有用的方法。语法格式如下：

```
Response.Write 任何数据类型
```

例如，下面示例将在浏览器中显示当前时间。

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Response.Write "现在是："
Response.Write Now
%>
```

Write 方法与<%= %>语法都是 ASP 服务器向客户端输出信息的方法，虽然两者的结果相同，但是其效果是不同的。一般来说，<%= %>语法常用于 HTML 与 ASP 代码相混合的情况，而 Response.Write 用于<%.....%>脚本段中。

但<%= %>语法的效率不及 Write 方法，原因如下：

- 从 VBScript 语法 `<%= expression %>` 将表达式的值写入 ASP 输出流来分析这个方法，可以发现如果响应缓冲没有打开，则这些语句的每一句都会导致通过网络，以许多小型包的形式向浏览器写入数据，这是非常慢的。
- 另外，解释少量脚本和 HTML，将导致在脚本引擎和 HTML 之间反复切换，也降低了性能。

因此，使用 `Response.Write` 方法替换内嵌的 `<%= %>` 语法，在应用程序的性能上可以得到很大的提高。例如，对于下面的记录集循环显示代码：

```
<% do while not rs.eof %>
<%=rs("flides") %>
<% rs.movenext %>
<% loop %>
```

可以替换为下面代码：

```
<%
out_str=""
for i = 1 to rs.reccount
out_str=out_str+rs("flides")
rs.movenext
next
response.write( out_str)
%>
```

这种方法执行的性能会大大提高。用 `For...Next` 循环语句替换 `Do...Loop` 循环语句也能够提高服务器运行效率，因为 `For...Next` 循环语句避免了在每个循环中反复检测 `rs.eof` 的数值，如果用记数循环，效率可以进一步提高。

很多时候读者可以利用 `Write` 方法重新定义页面的 HTML 结构和显示内容，当 `Response.Write` 方法与 HTML 源代码混合使用的过程中，为了避免错误，可以把 HTML 源代码中的引号（"）改为单引号（'）。例如：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Response.Write "<h5 align='center'>表3.3 Response对象的方法 </h5>"
Response.Write "<table border='1' cellpadding='0' align='center' cellspacing='0'
width='96%'>"
Response.Write " <tr>"
Response.Write " <th width='23%' valign='top'>方 法</th>"
Response.Write " <th width='76%' valign='top'>说 明</th>"
Response.Write " </tr>"
Response.Write " <tr>"
Response.Write " <td>AddHeader</td>"
Response.Write " <td>设置HTML标题 </td>"
Response.Write " </tr>"
Response.Write " <tr>"
Response.Write " <td>AppendToLog</td>"
Response.Write " <td>在该请求的Web服务器日志条目后添加字符串 </td>"
Response.Write " </tr>"
Response.Write " <tr>"
Response.Write " <td>BinaryWrite</td>"
Response.Write " <td>将给出信息写入当前HTTP输出中，并且不进行任何字符集转换 </td>"
Response.Write " </tr>"
Response.Write " <tr>"
Response.Write " <td>Clear</td>"
Response.Write " <td>清除任何缓冲的HTML输出 </td>"
Response.Write " </tr>"
Response.Write " <tr>"
```

```

Response.Write "      <td>End</td>"
Response.Write "      <td>停止处理.asp文件并返回当前的结果 </td>"
Response.Write "    </tr>"
Response.Write "  <tr>"
Response.Write "    <td>Flush</td>"
Response.Write "    <td>立即发送缓冲区中的输出信息 </td>"
Response.Write "  </tr>"
Response.Write "  <tr>"
Response.Write "    <td>Redirect</td>"
Response.Write "    <td>将重指示的信息发送到客户端浏览器, 尝试链接另一个URL</td>"
Response.Write "  </tr>"
Response.Write "  <tr>"
Response.Write "    <td>Write</td>"
Response.Write "    <td>将变量作为字符串写入当前的HTTP输出中 </td>"
Response.Write "  </tr>"
Response.Write "</table>"
%>

```

则上面代码的显示效果如图 3.2 所示。



图 3.2 用 Write 方法重新定义页面结构

2. Clear

Clear 方法用于清除缓冲区中的所有 HTML 输出, 但 Clear 方法只清除响应正文, 而不清除响应标题, 可以用该方法来处理错误情况; 但如果未将 Response.Buffer 设置为 True, 则该方法运行时将会出错。语法格式如下:

```
Response.Clear
```

3. End

End 方法使 Web 服务器停止处理脚本并返回当前结果, 文件中剩余的内容将不被处理。如果 Response.Buffer 已设置为 True, 则调用 Response.End 将缓冲输出。所以要取消所有向客户端的输出信息, 可以先清除缓冲区, 然后使用 End 方法结束程序运行。例如:

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Response.buffer=true
On error resume next
Err.clear
if Err.number<>0 then
  Response.Clear
  Response.End
end if
%>

```

4. Flush

Flush 方法用于立即把缓冲区中的信息输出到客户端浏览器中。如果未将 Response.Buffer

设置为 True, 则该方法运行时将会出错。语法格式如下:

```
Response.Flush
```

在一个文件中可以多次使用 Flush 方法实现数据的即时发送, 避免浏览器总处于等待状态。但是在很多时候, 读者会发现使用了 Flush 方法, 但是并没有将前面的信息发到客户端显示, 浏览器依然显示为白屏。这主要是因为 Flush 方法发送的内容至少要有 256 字节。也就是只有当服务器产生了至少 256 字节的数据, 才能在执行 Response.Flush 以后将信息发到客户端并显示。

5. Redirect

Redirect 方法可以使客户端浏览器立即重新定位到程序指定的 URL, 这样 ASP 程序就可以根据客户的请求来指向不同的页面。一旦使用了 Redirect 方法, 任何在页面中显式设置的响应正文内容都将被忽略。例如:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%response.buffer=true%>
<%
if Request.Form("flag") = "submit" then
    where=Request.form("where")
    Select Case where
        case "baidu"
            response.redirect "http://www.baidu.com/"
        case "google"
            response.redirect "http://www.google.cn/"
        case "css8"
            response.redirect "http://www.css8.cn/"
    End Select
end if
%>
<html>
<head>
<title>Redirect示例</title>
</head>
<body>
<h3>你要到哪里去? </h3>
<form aciton="" method="post">
    <select name="where">
        <option selected value="baidu">百度</option>
        <option value="google">谷歌</option>
        <option value="css8">样吧</option>
    </select>
    <input type="hidden" name="flag" value="submit">
    <input type="submit" name="submit" value="跳转">
</form>
</body>
</html>
```

在上面示例中当用户选择不同的下拉选项, 然后单击“跳转”按钮提交表单, 服务器接到申请后将进行逻辑判断, 并根据用户选择定位到相应的 URL。运行之后, 显示效果如图 3.3 所示。

6. BinaryWrite

BinaryWrite 方法可以不经任何字符转换就将指定的信息写到 HTTP 中输出。该方法主要用于把非字符串信息发送到客户端浏览器中, 例如, 客户端应用程序所需的二进制数据



图 3.3 Redirect 示例效果

等。语法格式如下：

```
Response.BinaryWrite "二进制数据"
```

例如：

```
<%
Set BinGen = Server.CreateObject(MY.BinaryGenerator)
Pict = BinGen.MakePicture
Response.BinaryWrite Pict
%>
```

7. AppendToLog

AppendToLog 方法用于将字符串添加到 Web 服务器日志条目的末尾，以跟踪用户行为。读者可以在脚本的同一部分中多次调用该方法，每次调用该方法时都会在日志的当前条目中添加指定的字符串。由于 IIS 日志中的字段由逗号分隔，所以该字符串中不能包含逗号，而且字符串最大长度为 80 个字符。语法格式如下：

```
Response.AppendToLog "要记录的字符串"
```

8. AddHeader

浏览器的请求和服务器的响应都会包含 Header（头部）。Header 提供了请求和响应的附加信息，也包含了浏览器生成请求和服务端提供响应的过程信息。

AddHeader 方法用于将指定的值添加到 HTML 标题中，该方法常常用来响应要添加新的 HTTP 标题。它并不会替代现有的同名标题。一旦标题被添加，将不能删除。此方法仅供高级用户使用。语法格式如下：

```
Response.AddHeader Name, Value
```

其中 Name 表示一个新的 HTTP 标题变量的名称，Value 表示这个变量的初始值。为避免命名不明确，Name 中不能包含任何下划线字符“_”。同时，由于 HTTP 协议要求所有的标题都必须在内容之前发送，所以必须在任何输出到客户端之前在脚本中调用 **AddHeader**。但当 Buffer 属性被设置为 True 时例外。

如果输出被缓冲，那么可以在脚本中的任何地方调用 **AddHeader** 方法，只要它在 Flush 之前执行即可。否则，对 **AddHeader** 方法的调用将产生一个运行错误。

3.5.3 Response 对象的数据集合

Response 对象有一个数据集合 Cookies，它负责取得记录在客户端的 Cookies 数据，允许用户检索在 HTTP 请求中发送的 Cookie 的值。

由于 HTTP 是一个无状态的通信协议，每一个链接都是全新的开始，因此，如果要判断某个用户是否曾经进入本网站，或者浏览过某个页面，可以在 ASP 中使用 Cookies。当然，Cookies 的用途并不仅限于此，还可以根据需要自行设置相关信息，取得客户端的一些机器信息，有关 Cookies 详细内容将在后面章节介绍。

3.6 Request 对象

Request 对象通常能够帮助服务器获取客户端浏览器的相关信息，例如，浏览器的种类、表头信息、表单参数及 Cookies 等。**Request** 对象的语法格式如下：

```
Request[.collection|property|method] (variable)
```

其中 collection 表示 Request 对象的集合, property 表示 Request 对象的属性, method 表示 Request 对象的方法, collection、property 和 method 三个参数只能选择一个, 也可以三个都不选。变量参数 (variable) 是一些字符串, 这些字符串是指定要从集合中检索的项目, 或作为方法与属性的输入。

3.6.1 Request 对象的属性

Request 对象只提供了一个 TotalBytes 属性, 这是一个只读的属性, 表示从客户端接收数据的字节长度。其语法格式如下:

字节长度=Request.TotalBytes

例如, 下面示例显示从客户端接收的数据字节大小:

```
<body>
<%
Response.Write "从客户端接收的数据大小为: "&Request.TotalBytes
%>
</body>
```

在浏览器中显示如图 3.4 所示。

由于直接从浏览器上运行该程序, 因此返回值为 0, 说明没有传递任何数据给 Web 服务器端。



图 3.4 显示从客户端接收的数据大小

3.6.2 Request 对象的方法

Request 对象仅提供 BinaryRead 方法。BinaryRead 方法是以二进制方式来读取客户端使用 POST 方式所传递的数据。其语法格式如下:

变量数组=Request.BinaryRead (Count)

BinaryRead 方法的返回值为通用变量数组 (Variant Array), 其参数是一个单数, 用以表示每次读取的数据字节大小, 范围介于 0 到 Request 对象 BinaryRead 方法所取得的字节大小之间。例如:

```
<%
Dim binread
Dim bytecount
bytecount = Request.TotalBytes
binread = Request.BinaryRead(bytecount)
%>
```

3.6.3 Request 对象集合概述

Request 对象提供了众多的集合, 利用这些集合可以获取不同类型的客户端发送的信息或者服务器端预定的环境变量的值。Request 对象的数据集合见表 3.4。

表 3.4 Request 对象的数据集合

数据集合	说 明
ClientCertificate	用于检索存储在发送到 HTTP 请求中客户端证书中的字段值
Cookies	用于检索在 HTTP 请求中发送的 Cookie 的值
Form	用于检索 HTTP 请求正文中表单元素的值
QueryString	用于检索 HTTP 查询字符串中变量的值
ServerVariables	用于检索预定的环境变量的值

使用下面的语法可以获取某个特定数据集合的一个内容值:

```
内容值=Request[.collection]("变量名称")
```

其中 **collection** (集合名称) 可以省略, 此时语法格式如下:

```
内容值=Request("变量名称")
```

不过使用这种简洁的写法, 一定要注意当不同数据集合中有相同的变量名称时, ASP 会按照 **QueryString**、**Form**、**Cookies**、**ClientCertificate** 及 **ServerVariables** 等数据集合的顺序返回所想要取得的变量内容。而且如果不指明数据集合名称, 服务器每次执行时都要检索所有集合内的数据, 会占用服务器的资源, 因此一般建议读者要指明所检索的集合名称。

3.6.4 Form 数据集合

Form 数据集合是 **Request** 对象中最常用的数据集合。**Form** 数据集合专门用来存储客户端网页表单中传递的各种元素的值。这些表单元素都是由 **<form>** 标记组合而成的, 包括单行文本框 (**Text**)、多行文本框 (**TextArea**)、复选框 (**CheckBox**)、单选按钮 (**Radio**)、列表/菜单框 (**Select**) 和按钮 (**Button**) 等。



提示

使用 **Form** 数据集合传递数据时, 必须定义表单 **Form** 元素的 **method** 的属性值为 **post**, **post** 传递表单信息的方式与 **get** 通过 **URL** 查询字符串变量进行传递的方式不同, 而是对信息进行特殊的封装再发送给 **ASP** 服务器。每个表单元素的数据都被单独打包的, 并存储在 **Form** 数据集合中, **ASP** 应用程序能够利用 **Request** 对象的 **Form** 集合获取这些数据, 它所能传递的信息量也没有限制。

在 **Form** 数据集合中每个成员均为只读, 只用于检索。其语法格式如下:

```
表单对象内容=Request.Form("表单对象名称")
```

或

```
表单对象内容=Request.Form(索引值)
```

其中表单对象名称是指定集合要检索的表单元素的名称, 索引值是访问某参数多个值中的一个, 可以是 1 到 **Request.Form(parameter).Count** 之间的任意整数。表单元素的索引顺序与其在网页中的排列位置是对应的。

例如, 下面示例用来获取并显示用户输入的姓名和爱好。具体操作步骤如下。

01 新建一个 **HTML** 表单输入程序, 存储为 **index.asp**, 详细代码如下:

```
<html>
<head>
</head>
<body>
<form action="result.asp" method="post">
  <p>姓名: <input type="text" size="20" name="Name"></p>
  <p>爱好: <input type="text" size="20" name="Love"></p>
  <p><input type="submit" value="提交"> </p>
</form>
</body>
</html>
```

02 再建立一个文件, 命名为 **result.asp**, 在其中输入下面代码:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Name = Request.Form("Name")
```

[illegible]

03 保存并运行 index.asp，在浏览器中输入姓名和爱好，如图 3.5（a）所示；单击“提交”按钮，效果如图 3.5（b）所示。

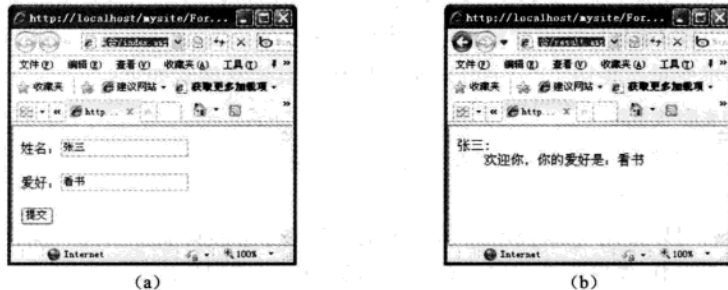


图 3.5 获取并显示用户输入姓名和兴趣
(a) 表单输入; (b) 显示表单信息

读者也可以利用索引值来访问表单中各个元素的值，例如，把上面示例中 `result.asp` 文件中的代码改写为下面形式：

[illegible]

其中 Request.Form(1)表示获取表单中第 1 个元素的值, Request.Form(2)表示获取表单中第 2 个元素的值。

当一个页面内出现多个表单域时，`Request.Form` 只获取当前提交的表单信息，因此索引值不会受其他表单元素的影响。例如，如果在上面示例基础上增加两个表单域：

```
<html>
<head>
</head>
<body>
<h3>表单1</h3>
<form action="result.asp" method="post" name="form1">
  <p>姓名: <input type="text" size="20" name="Name1"></p>
  <p>爱好: <input type="text" size="20" name="Love1"></p>
  <p><input type="submit" value="提交"> </p>
</form>
<h3>表单2</h3>
<form action="result.asp" method="post" name="form2">
  <p>姓名: <input type="text" size="20" name="Name2"></p>
  <p>爱好: <input type="text" size="20" name="Love2"></p>
  <p><input type="submit" value="提交"> </p>
</form>
</body>
</html>
```

这样不管用户使用哪个表单域提交信息，使用索引值都可以准确检索到提交的信息，而



图 3.8 自动获取查询字符串信息
(a) 单击链接; (b) 显示路径传递信息

3.6.6 GET 和 POST 数据传输方式

在 ASP 应用程序中, 前台与后台的数据交互, 都是通过 HTML 中的 Form 表单来完成的。Form 提供了两种数据传输的方式——GET 和 POST。虽然它们都是数据的提交方式, 但是在实际传输时确有很大的不同, 并且可能会对数据产生严重的影响。

Form 中的 GET 和 POST 方法, 在数据传输过程中分别对应了 HTTP 协议中的 GET 和 POST 方法。二者主要区别如下。

- GET 是用来从服务器上获得数据, 而 POST 是用来向服务器上传递数据。
- GET 将表单中数据的按照 `variable=value` 的形式, 添加到 `action` 属性所指向的 URL 后面, 并且使用 “?” 字符连接, 而各个变量之间使用 “&” 连接。POST 是将表单中的数据放在 Form 的数据体中, 按照变量和值相对应的方式, 传递到 `action` 所指向 URL。
- GET 是不安全的, 因为在传输过程, 数据被放在请求的 URL 中, 而如今现有的很多服务器、代理服务器或者用户代理都会将请求 URL 记录到日志文件中, 然后放在某个地方, 这样就可能会有一些隐私的信息被第三方看到。另外, 用户也可以在浏览器上直接看到提交的数据, 一些系统内部消息将会一同显示在用户面前。POST 的所有操作对用户来说都是不可见的。
- GET 传输的数据量小, 这主要是因为受 URL 长度限制; 而 POST 可以传输大量的数据, 所以在上传文件只能使用 POST 方式。
- GET 限制 Form 表单的数据集的值必须为 ASCII 字符, 而 POST 方式支持整个 ISO10646 字符集。
- GET 方式是 Form 表单的默认数据传递方式。

如果请求的信息是一些简单的参数值, 便可以使用 GET 方法, 但 GET 方法可能会产生很长的 URL, 或许会超过某些浏览器和服务对 URL 长度的限制。一般情况下, 读者可以参考下面条件来选择不同传递方式。

使用 POST 方法。

- 请求的结果有持续性的副作用, 例如, 向数据库内添加新的数据行。
- 若使用 GET 方法, 则表单上收集的数据可能让 URL 过长。
- 要传送的数据不是采用 7 位的 ASCII 编码。

使用 GET 方法。

- 请求是为了查找资源（作为参数而存在），HTML 表单数据仅用来帮助搜索。
- 请求结果无持续性的副作用。
- 收集的数据及 HTML 表单内的输入字段名称的总长不超过 1024 个字符。

由于 GET 和 POST 是分别传递的，并存储在不同的数据集合中，因此即使它们之间存在同名变量的现象，也不会相互影响，读者可以同时使用它们，这在 ASP 编程中比较常用，也是很重要的技巧。最常见的应用是通过 GET 方式把一些配置变量传递给服务器脚本，同时允许用户以 POST 方式输入其他参数。

3.6.7 Cookies 数据集合

Cookie 实际上是一个字符串或一个标志，当一个包含 Cookie 的页面被用户浏览器读取时，一个 Cookie 就被存入到本地硬盘中，当需要时该网站可以从用户的本地硬盘中读取这些 Cookie。所有的 Cookie 都被存放在用户本地计算机上，而非服务器上。在 XP 系统中的具体目录为：

C:\Documents and Settings\本地计算机名称\Cookies。

存放 Cookie 的文件名命名规则为：

用户名@网站名.txt

例如，zhuyinhong@bj.58[2].txt，这个文件是纯文本文件，可以使用任何文本编辑器来打开它。应用 Cookies 数据集合的语法格式如下：

Response.Cookies(Cookie) [(key) [attribute]]=value

- 参数 Cookie 用来创建或设置 Cookie 的名称。
- 参数 key 为可选参数，表示这个 Cookie 会以目录的类型加以存储。如果指定 key，则 Cookie 就是一个字典，而且该 key 将被设置为 value，以便在以后的接收中使用。
- 参数 attribute 指定 Cookie 自身的有关信息。
- value 用来指定分配给 key 或 attribute 的值。

一般来说，当用户第一次进入网站时，可以利用 Response 对象的 Cookies 数据集合将数据存储在用户的计算机中。当用户再次进入该网站时，就可以利用 Request 对象的 Cookies 数据集合取得相关信息。

例如，下面使用 Cookie 数据集合进行数据传输。具体操作步骤如下。

01 新建一个表单输入页面，保存为 index.asp，在其中输入下面代码：

```
<html>
<head>
</head>
<body>
<form method="post" action="result.asp">
请输入你的姓名: <input type="text" name="name" size="12" />
<input type="submit" value="提交">
</form>
</body>
</html>
```

02 建立一个数据接收页面，保存为 result.asp，输入下面代码：

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>数据接收</title>
</head>
<body>
```

```

<%
Response.Buffer=True
If Request.Form("name")<>" Then
    Response.Cookies("name")=Request.Form("name")
    Response.Redirect("index1.asp")
Else
%>
<h3>你没有输入姓名, 请返回输入! </h3>
<%End if%>
</body>
</html>

```

'判断用户是否输入信息
'把表单信息写入用户计算机的Cookies中
'跳转到信息显示页面

03 新建一个信息显示页面, 保存为 result1.asp, 在其中输入下面代码:

```

<%
Response.Buffer=True
If Request.Form("name")<>" Then
    Response.Cookies("name")=Request.Form("name")
    Response.Redirect("result1.asp")
Else
    Response.Redirect("index.asp")
End if
%>

```

'判断用户是否输入信息
'把表单信息写入用户计算机的Cookies中
'跳转到信息显示页面

04 保存并运行 index.asp, 在表单中输入姓名, 如图 3.9 (a) 所示, 效果如图 3.9 (b) 所示, 如果没有输入姓名就提交表单, 则没有任何反应。



(a)



(b)

图 3.9 用 Cookie 进行数据传输

(a) 输入姓名; (b) 显示姓名

Cookie 是一个比较复杂的技术, 在后面章节中还将更深入地介绍。

3.6.8 ServerVariables 数据集合

ServerVariables 数据集合负责取得 Web 服务器端的环境变量, 使用它可以提取 HTTP 请求的头部信息。其语法格式如下:

```
Request.ServerVariables("环境变量名称")
```

例如, 新建一个 ASP 文件, 在页面中输入下面代码, 保存并运行, 读者可以查询当前服务器所有的服务器环境变量及其值, 效果如图 3.10 所示。

```

<%
Response.Write("<h3><center>服务器变量</center></h3>")
For each Key in Request.ServerVariables
    Response.Write("Request.ServerVariables("&Chr(34)&key&Chr(34)&")")
    Response.Write("<br>"&Request.ServerVariables(key)&"</br><br>")
Next
%>

```



图 3.10 显示服务器环境变量

由于服务器环境变量非常多，不便于记忆，下面将常用环境变量列表说明于表 3.5 中。

表 3.5 常用服务器环境变量列表

环境变量名称	说 明
ALL_HTTP	客户端发送的所有 HTTP 标题文件
ALL_RAW	检索未处理表格中所有的标题。ALL_RAW 和 ALL_HTTP 不同，ALL_HTTP 在标题文件名前面放置 HTTP_prefix，并且标题名称总是大写的。使用 ALL_RAW 时，标题名称和值只在客户端发送时才出现
APPL_MD_PATH	检索 ISAPI DLL 的 (WAM) Application 的元数据库路径
APPL_PHYSICAL_PATH	检索与元数据库路径相应的物理路径。IIS 通过将 APPL_MD_PATH 转换为物理 (目录) 路径以返回值
AUTH_PASSWORD	该值输入到客户端的鉴定对话中。只有使用基本鉴定时，该变量才可用
AUTH_TYPE	这是用户访问受保护的脚本时，服务器用于检验用户的验证方法
AUTH_USER	未被鉴定的用户名
CERT_COOKIE	客户端验证的唯一 ID，以字符串方式返回。可作为整个客户端验证的签字
CERT_FLAGS	如有客户端验证，则 bit0 为 1。如果客户端验证的验证人无效 (不在服务器承认的 CA 列表中)，bit1 被设置为 1
CERT_ISSUER	用户验证中的颁布者字段 (O=MS, OU=IAS, CN=user name, C=USA)
CERT_KEYSIZ	安全套接字层连接关键字的位数，如 128
CERT_SECRETKEYSIZE	服务器验证私人关键字的位数，如 1024
CERT_SERIALNUMBER	用户验证的序列号字段
CERT_SERVER_ISSUER	服务器验证的颁发者字段
CERT_SERVER_SUBJECT	服务器验证的主字段
CERT_SUBJECT	客户端验证的主字段
CONTENT_LENGTH	客户端发出内容的长度
CONTENT_TYPE	内容的数据类型。同附加信息的查询一起使用，如 HTTP 查询 GET、POST 和 PUT
GATEWAY_INTERFACE	服务器使用的 CGI 规格的修订，格式为 CGI/revision

续表

环境变量名称	说 明
HTTP_<HeaderName>	HeaderName 存储在标题文件中的值。未列入该表的标题文件必须以 HTTP_ 作为前缀, 以使 ServerVariables 集合检索其值。 服务器将 HeaderName 中的下划线 (_) 解释为实际标题中的破折号。例如, 如果您指定 HTTP_MY_HEADER, 服务器将搜索以 MY-HEADER 为名发送的标题
HTTPS	如果请求穿过安全通道(SSL), 则返回 ON。如果请求来自非安全通道, 则返回 OFF
HTTPS_KEYSIZE	安全套接字层连接关键字的位数, 如 128
HTTPS_SECRETKEYSIZE	服务器验证私人关键字的位数, 如 1024
HTTPS_SERVER_ISSUER	服务器验证的颁发者字段
HTTPS_SERVER_SUBJECT	服务器验证的主字段
INSTANCE_ID	文本格式 IIS 实例的 ID。如果实例 ID 为 1, 则以字符形式出现。使用该变量可以检索请求所属的 (元数据库中) Web 服务器实例的 ID
INSTANCE_META_PATH	响应请求的 IIS 实例的元数据库路径
LOCAL_ADDR	返回接受请求的服务器地址。如果在绑定多个 IP 地址的多宿主机器上查找请求所使用的地址时, 这个变量非常重要
LOGON_USER	用户登录 Windows NT 的账号
PATH_INFO	客户端提供的额外路径信息。可以使用这些虚拟路径和 PATH_INFO 服务器变量访问脚本。如果该信息来自 URL, 在到达 CGI 脚本前就已经由服务器解码了
PATH_TRANSLATED	PATH_INFO 转换后的版本, 该变量获取路径并进行必要的由虚拟至物理的映射
QUERY_STRING	查询 HTTP 请求中间号 (?) 后的信息
REMOTE_ADDR	发出请求的远程主机的 IP 地址
REMOTE_HOST	发出请求的主机名称。如果服务器无此信息, 它将设置为空的 REMOTE_ADDR 变量
REMOTE_USER	用户发送的未映射的用户名字符串。该名称是用户实际发送的名称, 与服务器上验证过滤器修改过后的名称相对应
REQUEST_METHOD	该方法用于提出请求, 相当于用于 HTTP 的 GET、HEAD、POST 等等
SCRIPT_NAME	执行脚本的虚拟路径, 用于自引用的 URL
SERVER_NAME	出现在自引用 URL 中的服务器主机名、DNS 化名或 IP 地址
SERVER_PORT	发送请求的端口号
SERVER_PORT_SECURE	包含 0 或 1 的字符串。如果安全端口处理了请求, 则为 1, 否则为 0
SERVER_PROTOCOL	请求信息协议的名称和修订, 格式为 protocol/revision
SERVER_SOFTWARE	应答请求并运行网关的服务器软件的名称和版本, 格式为 name/version
URL	提供 URL 的基本部分

ServerVariables 数据集合是一个非常有用的数据集合, 利用该集合可以编写功能强大的 Web 应用程序, 下面举例介绍 ServerVariables 数据集合的常规应用。

1. 检索和处理用户 URL 信息

在 ServerVariables 数据集合中包含 Web 服务器的详细信息和当前页面的路径信息。在任何地方创建一个页面都可使用这些信息。例如, 创建一个“自引用”页面, 此页面能够再次调用自身完成另一项任务, 我们可以使用下面代码:

```
<form action="<% = request.servervariables("path_info") %>" method="post">
```

同样也可以用 HTTP 的“SCRIPT_NAME”值来获得:

```
<form action="<% = request.servervariables("script_name") %>" method="post">
```

下面是利用 PATH_INFO 服务器环境变量来打开同一个站点内另一个文件的代码:

```

<%
strfullpath = request.servervariables("path_info")
strpathonly = left(strfullpath, instrrev(strfullpath, "/"))
strnextpage = strpathonly & "blog/default.asp"
%>
<a href="<%=strnextpage%>">打开另一个文件</a>

```

利用这种方式,即使原始页面的名称或位置发生变化,这些代码都能够正常工作,因为使用了当前页面的路径信息。当然,第2个例子在分离的目标页的名称发生变化时运行会失败。读者也可以为搜索引擎的子会话自动建立 URL,实现收集 ServerVariable 的一些值。

在下面的代码中可以创建一个完整的 URL 字符串,包括端口号:

```

<%
strFullURL = "http://" & Request.ServerVariables("LOCAL_ADDR") _
& ":" & Request.ServerVariables("SERVER_PORT") _
& Request.ServerVariables("PATH_INFO")
Response.Write strFullURL
%>

```

运行后会返回类似下面完整的 URL 信息:

http://127.0.0.1:80/site/ServerVariables数据集合3.asp

2. 检索和处理用户浏览器信息

在 ServerVariables 数据集中,另外一个有用的值是用户浏览器的用户代理字符串。使用 ServerVariables 集中的“HTTP_USER_AGENT”值来获得用户代理字符串,然后通过一些脚本来解析该信息并寻找生产厂家名称和浏览器版本。例如:

```

<%
strUA = Request.ServerVariables("HTTP_USER_AGENT")
Response.Write "你的用户代理 (网页浏览器) 是<br>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<B>" & strUA &
"</B><br>"
If InStr(strUA, "MSIE") Then
    Response.Write "<p>你的浏览器可能是Internet Explorer。</p>"
    intVersion = Cint(Mid(strUA, InStr(strUA, "MSIE") + 5, 1))
    If intVersion >= 4 Then
        Response.Write "<p>你能够使用Dynamic HTML技术。</p>"
    End If
Else
    If InStr(strUA, "Mozilla") Then
        If InStr(strUA, "compatible;") = 0 Then
            Response.Write "<p>你的浏览器可能是非Internet Explorer浏览器。</p>"
            intVersion = Cint(Mid(strUA, InStr(strUA, "/") + 1, 1))
            If intVersion >= 4 Then
                Response.Write "你能够使用Netscape Dynamic HTML技术"
            End If
        Else
            strVersion = Mid(strUA, InStr(strUA, "compatible;") + 12)
            strProduct = Left(strVersion, InStr(strVersion, " "))
            Response.Write "你的浏览器版本是Navigator-compatible."
        End If
    End If
End If
%>

```



提示

Netscape 在用户代理字符串中不提供厂家的名称,因而无法绝对保证一个浏览器一定是 Navigator。

3. 检索和处理用户浏览器语言

ServerVariables 集合中另外一个有用的值是“HTTP_ACCEPT_LANGUAGE”，它包含了一个当浏览器安装时指定的用户地区版本的语言代码。例如，zh-cn（中国）、en-us（英国、美国）、de-at（德国、澳大利亚）和 es-pe（西班牙、秘鲁）等。读者可以利用这个信息来检测语言代码并自动装载一个合适的特定地区或指定语言版本的页面。例如：

```
<%
StrLocale = Lcase(Left(Request.ServerVariables("HTTP_ACCEPT_LANGUAGE"),2))
Select Case strLocale
    Case "zh": Response.Redirect "http://www.site.com.cn/"
    Case "en": Response.Redirect "http://www.site.com.uk/"
    Case "de": Response.Redirect "http://www.site.com.de/"
    Case "fr": Response.Redirect "http://www.site.com.fr/"
    Case Else: Response.Redirect "http://www.sitel.com/"
End Select
%>
```

3.6.9 ClientCertificate 数据集合

ClientCertificate 数据集合用于从 Web 浏览器发布请求的 HTTP 中获取验证字段。如果 Web 浏览器使用 SSL3.0/PTI 协议，即使用以 https://开头的 URL，而不是 http://连接服务器及服务器请求验证，则浏览器将发送验证字段。如果没有发送验证，ClientCertificate 集合将返回 Empty。要注意的是，必须先将 Web 服务器设置为要求客户端验证，然后才能使用 ClientCertificate 集合，其语法格式如下：

```
Request.ClientCertificate (关键字[子字段])
```

3.7 Session 对象

Session 中文翻译为“会话”的意思，在 ASP 中 Session 对象代表服务器端与客户端之间的“会话”。它只针对单个用户，也就是说，当某个用户在网站的页面之间切换时，只能访问自己的 Session 变量，而无法访问其他用户的 Session 变量。

Session 的作用时间可以从浏览者到达某个特定 Web 页开始，直到该用户离开 Web 站点，当然也可以在 ASP 程序中利用代码终止某个 Session。



提示

HTTP 协议是一种无状态的协议，利用 HTTP 协议无法跟踪用户。从网站的角度看，每一个新的请求都是单独存在的。当服务器完成用户的请求后，服务器将不能再继续保持与该用户浏览器的连接。当用户在 Web 站点的多个页面间切换时，根本无法知道用户的相关信息。Session 对象弥补了这个缺陷。当用户在 Web 站点的多个页面间切换时，利用 Session 可以保存该用户的一些有用信息，网站可以利用这些信息获得该用户在网站的活动情况。

利用 Session 可以存储浏览者的一些特定信息，例如，浏览者的姓名、性别、所用浏览器的类型以及访问停留时间等。Session 还可以作为存储访问者的信息容器，例如，电子商务中的虚拟购物篮，无论什么时候浏览者在网站上选择了一种产品，这种产品就会被存储在购物篮；当浏览者准备离开时，可以立即订购以前所选择的产品。Session 还可以用来跟踪浏览者的访问路径，从中挖掘有用的信息，如浏览者的兴趣爱好等，这些信息对站点定位非常有益。



图 3.11 用 Session 变量实现用户会话
(a) 输入姓名; (b) 显示姓名

3.7.2 Session 对象的属性

Session 对象提供的属性包括 TimeOut、SessionID、CodePage、LCID 属性等。

1. SessionID

SessionID 记录着每个 Session 的代号, 这个代号是由服务器产生的, 是一个不重复的长整型数字。语法格式如下:

```
长整数=Session.SessionID
```

在创建会话时, 服务器自动为每一个会话分配一个唯一的 SessionID。可以通过获取 SessionID 属性值来确定 Session。

有两种方法可以屏蔽掉 SessionID, 一种是将全站进行屏蔽, 另一种是在一个单独的 ASP 程序中进行相应的屏蔽。

如果想将整个站点的 Session 操作进行屏蔽, 可以在 IIS 中的【应用程序配置】对话框中的【应用程序选项】选项卡中取消【启用会话状况】选项。在 ASP 文件中可以将下面代码放在首行以屏蔽掉 Session 操作。

```
<%EnableSessionState = False%>
```

2. TimeOut

TimeOut 属性用来设置 Session 的最长间隔时间, 这里的间隔时间是指 Session 对象最后一次被访问后能够存在的时间, 以分钟为单位。系统默认的时间是 900 秒, 这个时间可以在 IIS 中重新设置, 也可以根据实际情况在程序代码中指定, 其语法格式如下:

```
Session.Timeout=分钟数
```

如果用户刷新了网页, 将被认为是新的访问者, 服务器将为其重新分配 SessionID。

3. CodePage

由于用户可能来自世界各地, 无法事先编写各种不同的版本, 所以 ASP 的 Session 对象提供了 CodePage 属性。CodePage 属性表示字符串编码及转换的依据。代码页是一个可以包括数字、标点符号, 以及其他字母的字符集。对于不同的语言和地区可以使用不同的代码页。例如, ANSI 代码页 1252 用于美国英语和大多数欧洲语言, OEM 代码页 932 用于日文汉字, 中文的代码页为 950。其语法格式如下:

```
Session.CodePage = Codepage值
```

其中参数 CodePage 是一个无符号整数, 这个整数代表正在运行 ASP 脚本引擎的系统的有效代码页。

4. LCID

LCID 属性与 CodePage 属性有相同的作用，都是提供给不同国家的用户浏览用的。不同的是 LCID 属性是设置网页以客户端当地的相关设置显示，如设定局部日期、货币和事件格式的标识符。其语法格式如下：

```
Session.LCID = LCID值
```

3.7.3 Session 对象的方法

Session 对象提供了一个 Abandon 方法。该方法将删除所有存储在 Session 对象中的数据并释放它们所占有的资源。如果没有明确调用 Abandon 方法，服务器在 Session 对象的 Timeout 属性规定的时间后删除对象数据。当服务器删除 Session 对象时，将无法再取得其变量值，而且 Session_OnEnd 事件将一起被激活。语法格式如下：

```
Session.Abandon
```



提示

Abandon 方法被调用时，将顺序删除当前的 Session 对象，不过在当前页中所有脚本命令都处理完后，对象才会被真正删除。这就是说，在调用 Abandon 方法时，可以在当前页上访问存储在 Session 对象中的变量，但在随后的 Web 页上不行。

例如：

```
<%  
Session.Abandon  
Session("MyName") = "Mary"  
Response.Write(Session("MyName"))  
%>
```

上面示例将清除所有会话信息，但第 3 行代码依然能够输出 MyName 的值。这是因为在服务器处理完脚本前没有删除 Session 对象。

如果在随后 Web 页上访问 MyName 变量，读者就会发现该变量已经为空。由于当包含上一个例子的页面结束处理时，MyName 同前面的 Session 对象一起被删除了。

放弃会话并打开后面的 Web 页时，服务器会创建新的 Session 对象。读者可以在新的 Session 对象中存储变量和对象。

3.7.4 Session 对象数据集合

Session 对象的数据集合包括 Contents 和 StaticObjects 两个数据集合。

1. Contents

Contents 集合包含当前会话中使用的会话变量。语法格式如下：

```
Session.Contents(Session变量名称)
```

几乎所有在 Session 中存储的内容都存在于 Contents 集合中，但是不包括使用 <object> 标记为该会话建立的项目。此集合可用于确定指定会话项的值或遍历集合并检索出会话中所有项的列表。例如，在下面代码中使用一个循环结构通过 Contents 集合的关键字来遍历集合：

```
<%  
Dim sessitem  
For Each sessitem in Session.Contents  
    Response.write(sessitem & " : " & Session.Contents(sessitem) & "<BR>")  
Next  
%>
```

2. StaticObjects

StaticObjects 集合包含 Session 对象范围中用<object>标记创建的所有对象,该集合可用于确定对象特定属性的值,或用于遍历集合并获取所有对象的全部属性。语法格式如下:

```
Session.StaticObjects("要检索的属性")
```

使用循环语句可以遍历 StaticObjects 集合中的关键字。例如:

```
<%
Dim objprop
For Each objprop in Session.StaticObjects
    Response.write(objproperty & " : " & Session.StaticObjects(objprop) & "<BR>")
Next
%>
```

3.7.5 Session 对象的事件

Session 对象有两个事件: Session_OnStart 事件和 Session_OnEnd 事件。其中, Session_OnStart 事件对应 Session 对象的起始事件,每当产生一个 Session 对象时,该事件所定义的程序码都将被激活; Session_OnEnd 事件对应 Session 对象的结束事件,当 Session 对象被终止或失效时,该事件所对应的代码被激活。

这两个事件的处理过程都必须写在 Global.asa 文件中。其语法格式如下:

```
<script language="vbscript" runat="server">
sub session_onstart
'会话开始时处理的程序代码
end sub
sub session_onend
'会话结束时处理的程序代码
end sub
</script>
```

在<script>标记中, language 属性指定用于编写事件脚本的脚本语言类型,可以是任何一种支持脚本的语言,常用的有 VBScript 或 JScript。



提示

Global.asa 文件必须用<script>标记引用这两个事件,而不能用<%和%>标记引用。Global.asa 文件对于 ASP 程序非常重要,该文件中包括了一些通用程序段和一些网站的通用说明。Global.asa 文件必须放在网站应用程序的根目录下,文件中不能含有任何输出语句,例如,HTML 的标记和 Response.Write 方法的代码都不能出现在该文件中。

在 Web 服务器启动后,当一个应用程序目录中的任何一个 ASP 文件第一次提出 HTTP 请求时,服务器就会读取该应用程序的 Global.asa 文件。此外,当不具有 Session 的客户端向服务器请求一个 ASP 文件时也会读取该文件。

3.9 Cookie 对象

在 Response 对象、Request 对象以及 Session 对象中都曾提及 Cookies 数据集合。Cookies 数据集合实际上是 Cookie 对象的一个集合。本节将详细介绍 Cookie 对象使用。



提示

Session 对象实际上是利用 Cookie 进行信息处理的,当用户向服务器提出了某个 Session (会话)请求后,服务器就在用户的浏览器上创建了一个 Cookie;当这个 Session 结束时,也就意味着这个可以利用 ServerVariables 集合的 HTTP_COOKIE 来接收这个 Cookie 信息,代码是: <%=Request.ServerVariables("HTTP_COOKIE")%>。

3.9.1 Cookie 简介

Cookie 俗称小甜饼, Netscape 首先在它的浏览器中引入了 Cookie 概念, 后来 WWW 协会支持了 Cookie 标准。目前大部分浏览器都支持 Cookie 技术。

Cookie 实际上是一个字符串或一个标志, 当一个包含 Cookie 的页面被用户浏览器读取时, 一个 Cookie 就被存入到本地硬盘中, 而非服务器上。当需要时该网站可以从用户的本地硬盘中读取这些 Cookie。

目前有些 Cookie 是临时的, 还有一些是持续的。例如, 当 Cookie 被 ASP 用来跟踪用户进程直到用户离开网站时, Cookie 就是临时的。如果 Cookie 被保持在 Cookie 文件中直到用户返回时又进行调用, 这时的 Cookie 就是持续的。

由于 Cookie 能够读、写用户本地硬盘中的数据, 对于 Cookie 的使用一直有很大的争议, 很多用户担心个人隐私被泄露或对本地计算机的安全构成威胁。但从目前的使用看, Cookie 只能向用户本地硬盘的固定目录中写入文本文件 (可以参考 3.6.7 节介绍), 而不是可执行文件, 它们对计算机不会构成危害。

3.9.2 创建 Cookie 对象

要创建一个 Cookie 对象, 应使用 Response 对象的 Cookies 数据集合。Response 对象的 Cookies 数据集合用来设置 Cookie 的值, 如果指定的 Cookie 不存在就创建它; 如果存在的话就赋予它一个新值。其语法格式如下:

```
Response.Cookies (Cookie) [(key) [attribute] = Value
```

其中参数 Cookie 用来创建或设置 Cookie 的名称, 参数 key 为可选参数, 表示这个 Cookie 会以目录的类型加以存储。如果指定 key, 则 Cookie 就是一个字典, 而且该 key 将被设置为 value, 以便在以后的接收中使用。参数 attribute 指定 Cookie 自身的有关信息, 见表 3.6。

value 用来指定分配给 key 或 attribute 的值。

表 3.6 attribute 参数列表

名 称	说 明
Domain	只允许写。用来指定 Cookie 将发送到哪个域的请求中去。如果不指定, 就默认为创建 Cookie 所在的域
Expires	只允许写。用来指定 Cookie 的过期日期。为了在会话结束后将 Cookie 存储在客户端磁盘上, 必须设置该日期, 因为该属性的默认值是一旦会话结束, 就删除所创建的 Cookie。另外, 若此项属性的设置未超过当前日期, 则在任务结束后 Cookie 将到期
HasKeys	只允许读。用来指定 Cookie 是否包含关键字, 它的值是布尔值, 如果 Cookie 有关键字 (即 Cookie 含有 key 参数, 是一个 Cookie 字典), 则 HasKeys 的值是 True, 否则是 False
Path	只允许写。若被指定, 则 Cookie 将只发送到对该路径的请求中。如果未设置该属性, 则使用应用程序的路径
Secure	只允许写。用来指定 Cookie 是否安全, 它的取值是布尔值。如果指定 Cookie 是安全的, 就给它赋值 True, 否则就给它赋值 False

在实际应用中, 可以创建两种类型的 Cookie 集合, 不带关键字 key 的 Cookie (单值 Cookie) 和带有多个关键字 key 的 Cookie 字典。下面分别讨论两种 Cookie 的创建。

1. 创建单值的 Cookie

要创建不带关键字 key 的 Cookie 只要指定参数 Cookie 的值以及 value 的值就可以了。另外, 也可以为该 Cookie 指定属性。语法格式如下:

```
<%
Response.Cookies("item-name") = "item-value"
Response.Cookies("item-name").domain = "domain-url"
Response.Cookies("item-name").path = "virtual-path"
Response.Cookies("item-name").expires = #date#
%>
```

例如:

```
<%
Response.Cookies("name") = "Adobe Dreamweaver CS5"
Response.Cookies("name").Expires = #5 1,2008#
Response.Cookies("name").Domain = "asp.com"
Response.Cookies("name").Path = "/site/asp"
Response.Cookies("name").Secure = "True"
Response.Redirect("Cookie(遍历Cookies).asp")
%>
```

在上面代码中创建了一个名为 `name` 的不带关键字的 Cookie。它的值是“Adobe Dreamweaver CS5”，同时还指定了相应的属性值：`Expires` 属性指定了该 Cookie 要到 2008 年 5 月 1 日零时才能过期，`Domain` 属性指定了该 Cookie 只能发送到 `asp.com` 这个域中，`Path` 属性指定了该 Cookie 只能发送到路径为 `/site/asp` 的请求中，`Secure` 属性指定在数据传输过程中采用加密算法。

Cookie 是利用 HTTP 的 Header 进行数据传送的，因此应该在 ASP 的任何输出语句之前进行上述操作，也可以使用 Buffer 输出。

2. 创建带有关键字的 Cookie

创建带有关键字的 Cookie 字典时，只要带上 `key` 参数就可以了。例如：

```
<%
Response.Cookies("myCookie")("name") = "Adobe Dreamweaver CS5"
Response.Cookies("myCookie")("pass") = 123456
%>
```

上面示例创建了一个名为 `myCookie` 的 Cookie 字典，其中含有 `name` 和 `pass` 两个关键字。需要注意的是，如果想给一个带有关键字的 Cookie 字典指定属性值，一定不要带上关键字，否则会产生语法错误。正确的语法如下：

```
<%
Response.Cookies("myCookie")("name") = "Adobe Dreamweaver CS5"
Response.Cookies("myCookie")("pass") = "123456"
Response.Cookies("myCookie").Expires = #May 1,2008#
Response.Cookies("myCookie").Domain = "asp.com"
Response.Cookies("myCookie").Path = "/site/asp"
Response.Cookies("myCookie").Secure = "True"
%>
```

这里给名为 `myCookie` 的 Cookie 字典指定了过期的日期，使整个 Cookie 字典将于 2008 年 5 月 1 日零点过期，当然这也意味着在这个 Cookie 字典中的所有关键字值也随之过期了。

3.9.3 读取 Cookie 的值

要读取一个 Cookie 的值，或 Cookie 字典中的关键字值，需要使用 `Request` 对象的 `Cookies` 数据集合。该集合允许用户读取在 HTTP 请求中发送过来的 Cookie 值。基本语法格式如下：

```
Request.Cookies (Cookie) [(key) [attribute]]
```

其中 `Cookie` 用来指定被检索或读取其值的 Cookie 的名称；`key` 为一个可选项，用来指定

Cookie 自身的有关信息,且只有一个取值 HasKeys。HasKeys 具有只读的性质,用来表示 Cookie 是否带有关键字,如果 Cookie 带有关键字,就返回 True,否则就返回 False。

1. 读取单值的 Cookie

对于一般不带有关键字的 Cookie,可以采用指定 Cookie 名称的方式来检索 Cookie 的值。

例如:

```
<%=Request.Cookie("Type")%>
```

此外,也可以采用指定序号的方式来检索 Cookie 的值。例如,假设建立了两个 Cookie,分别是 name 和 pass,则可以使用下面代码获取它们的值。

```
<%
Response.write(Request.Cookies(1))
Response.write(Request.Cookies(2))
%>
```

2. 读取 Cookie 字典

对于 Cookie 字典的检索和读取,可以通过使用关键字来进行,也可以使用序号来进行。

例如,要检索前面建立的 myCookie 字典的值,可以用下面代码。

```
<%
Response.write(Request.Cookies("myCookie")("name"))
Response.write(Request.Cookies("myCookie")("pass"))
%>
```

或者

```
<%
Response.write(Request.Cookies("myCookie")(1))
Response.write(Request.Cookies("myCookie")(2))
%>
```

在上面语句中,前两条语句和后两条语句是等价的。它们都可以显示 myCookie 这个 Cookie 字典中的关键字。

因为所有的 Cookie 都保存在 Cookies 数据集中,所以也可以通过循环的方式遍历 Cookies 数据集,以检索所有 Cookie 或 Cookie 对象关键字的值。下面是一个遍历 Cookies 并将其中 Cookie 的值显示出来的示例代码。

```
<%
For Each objItem In Request.Cookies
    If Request.Cookies(objItem).HasKeys Then
        For Each objItemKey in Request.Cookies(objItem) '用循环读取Cookie字典的所有关键字
            Response.Write objItem & " (" & objItemKey & ") = " & Request.Cookies(objItem)
            (objItemKey) & "<BR>"
        Next
    Else
        Response.Write objItem & " = " & Request.Cookies(objItem) & "<BR>" '显示所有Cookie对象
    End If
Next
%>
```

上面代码非常类似于前面介绍的遍历 Request.Form 数据集方法。但是这里可以使用 Haskeys 属性来判别每个 Cookie 对象是否带有关键字。而在 Form 数据集里,必须查询 Request.Form(item_name).Count 属性,这是因为 Form 数据集(和所有的除 Cookies 以外的其他数据集)成员不可能是真正的集合。

3.10 Application 对象

Application 对象是一个应用程序级的对象，Application 包含的数据可以在整个站点或应用程序中被所有用户引用，并且可以在网站运行期间持久保存数据。Application 对象可以包括任何类型数据和变量，完成网站公共信息的共享和存储，例如，网站在线人数统计、创建多用户游戏以及多用户聊天室等功能。

Application 对象和 Session 对象都是用于在不同的 ASP 页面之间共享信息。但是它们也存在如下区别。

- 应用范围不同。Application 对象是针对所有用户，可以被多个用户共享。一个用户接收到的 Application 变量可以传递给另外的用户。而 Session 对象是针对单一用户，某个用户是无法访问其他用户的 Session 变量信息的。
- 存活时间不同。由于 Application 变量是多个用户共享的，因此不会因为某一个用户甚至全部用户离开而消失，一旦建立了 Application 变量，就会一直存在，直到网站关闭或这个 Application 变量被卸载。而 Session 变量会随着用户离开网站而被自动删除。

3.10.1 使用 Application

Application 变量的创建和使用方法与 Session 变量类似。其语法格式如下：

```
Application.property|method
```

其中 property 表示 Application 对象的属性，method 表示 Application 对象的方法。例如，在下面代码中创建一个 Application 变量，并测试该变量在赋值前后的效果。

```
<%  
data1 = Application("data1")  
Response.Write "进入网页时, data1=" & data1 & "<BR>"  
data1 = data1 + 1000  
Response.Write "网页结束时, data1=" & data1 & "<BR>"  
Application("data1") = data1  
%>
```

一旦当一个 Application 变量被赋值，就可以被本站点的其他 ASP 文件调用，即也可以被其他用户调用。由于 Application 变量创建后不会自己消亡，在使用时就要特别小心，因为它要始终占用内存，创建过多 Application 变量会降低服务器的响应速度。Application 变量可以在下面任一情况下被终止：

- 服务被终止
- Global.asa 文件被改变
- Application 变量被卸载

3.10.2 Application 对象的属性

Application 对象只有一个 ApplicationID 属性，它能够返回当前应用的标识，每个虚拟主机都拥有自己的标识，以便与浏览器维护会话状态。其语法格式如下：

```
Application.ApplicationID
```

3.10.3 Application 对象的方法

因为多个用户可以共享 Application 对象，对于同一个 Application 变量，如果多人同时调

用就可能会出现错误。因此 Application 对象提供了 Look 方法和 Unlook 方法。它们用于解决多个用户同时使用一个 Application 变量而发生的冲突。

Look 方法禁止其他用户修改存储在 Application 对象中的变量, 以确保在同一时刻仅有一个用户可修改和存取 Application 变量。与 Look 方法相反, Unlook 方法允许其他用户修改 Application 对象的属性。

例如, 下面代码可以实现现在同一个站点内访问人数统计:

```
<%
Application.Lock
Application("visitor_num")=Application("visitor_num")+1
Application.Unlock
%>
```

要真正实现该功能, 读者还必须把上面代码放入 Session 对象的 Session_OnStart 事件中, 具体说明可以参见 3.7.5 节内容。

3.10.4 Application 对象数据集合

Application 对象的数据集合包括 Contents 数据集合与 StaticObjects 数据集合。它们的使用方法与 Session 对象相同。

Contents 数据集合允许程序取得用户能够使用的所有 Application 对象, 但不包括以 <Object> 标记建立的对象及 Server 对象的 CreateObject 方法建立的对象。可以使用 Contents 集合获取给定的应用程序作用域的项目列表或指定一个特殊项目为操作对象。其语法格式如下:

Application.Contents (Application 对象名称)

例如:

```
<%
For Each Key in Application.Contents
    Response.Write (Key)
Next
%>
```

或者

```
For i = 1 to Application.Contents.Count
    Response.Write (i)
Next
%>
```

StaticObjects 数据集合可以取得以 <Object> 标记建立的 Application 对象, 可以使用该集合确定某对象指定属性的值或检索所有静态对象的所有属性。其语法格式如下:

Application.StaticObjects (Application 对象变量名称)

3.10.5 Application 对象的事件

Application 对象有 Application_OnStart 和 Application_OnEnd 两个事件。当每个 Application 对象开始时, 会激活 Application_OnStart 事件, 当 Application 对象结束时会激活 Application_OnEnd 事件。

与 Session 对象类似, Application 对象两个事件的代码也必须放在 Global.asa 文件中。

与 Session 对象的事件不同的是, Application 事件发生在整个 Web 应用程序的开始和结

束，而 Session 事件发生在一次会话的开始和结束。

Application_OnStart 事件一般是用来初始化这个 Application 的变量。Application_OnEnd 事件在应用程序退出时于 Session OnEnd 事件之后发生。

其语法格式如下：

```
<script language="vbscript" runat="server">
sub application_onstart
    '应用程序开始时执行该代码
end sub
sub application_onend
    '应用程序结束时执行该代码
end sub
</script>
```

例如，下面以一个在线人数统计的功能实现演示 Session 和 Application 对象的事件综合使用。具体操作步骤如下。

01 建立一个虚拟目录（该步骤是必需的）。新建 Global.asp 文件，放在被调试的虚拟目录的根目录下，在其中输入下面代码：

```
<script language=vbscript runat=server>
sub application_onstart '程序启动时，初始计数变量值为0
    application("online") = 0
end sub
sub session_onstart '当一个新用户访问站点时进行计数加1
    application.lock
    application("online") = application("online") + 1
    application.unlock
end sub
sub session_onend '当用户会话结束后，计数减1（如果没有该事件程序，则执行的就是页面访问程序了。）
    application.lock
    application("online") = application("online") - 1
    application.unlock
end sub
</script>
```

02 建立 online.asp 文件，在其中输入下面代码：

```
<%
if request.querystring("logout")="true" then
    session.Abandon()
    response.end
end if
%>
<body onunload=javascript:window.open("exit.asp")>
当前共有<%=Application("OnLine")%>在线
<a href="online.asp?logout=true">退出</a>
```

03 再新建一个文件，保存为 exit.asp，该文件可以防止用户通过关闭窗口右上角按钮来退出页面，从而躲避计数检测。具体代码如下：

```
<%session.Abandon()%>
<script>
self.close()
</script>
```

04 保存并运行 online.asp，显示效果如图 3.12（a）所示，如果单击“退出”文本链接，则不会显示计数，显示效果如图 3.12（b）所示。



(a)

(b)

图 3.12 在线人数统计

(a) 显示计数; (b) 不显示计数

3.11 Server 对象

Server 对象提供对服务器上的方法和属性的访问，其中大多数方法和属性是为应用程序提供服务的。Server 对象在 ASP 中是一个很重要的对象，许多高级功能都是由它完成的，如使用 Server 对象的 CreateObject 方法创建 ActiveX 组件。其语法格式如下：

```
Server.property|method
```

3.11.1 Server 对象的属性

Server 只有一个 ScriptTimeout 属性，该属性用于设置一个 ASP 脚本所允许的最长执行时间。如果在指定的时间内脚本没有执行完毕，系统将停止其执行，并且显示超时错误。默认时间为 90 秒。

例如，下面代码可以设置脚本最长执行时间为 100 秒：

```
<% Server.ScriptTimeout = 100 %>
```

以下代码能够获取 ScriptTimeout 属性当前值，并将其存储在变量 TimeOut 中：

```
<% TimeOut = Server.ScriptTimeout %>
```

3.11.2 Server 对象的方法

Server 对象提供了 7 个非常重要的方法：CreateObject、MapPath、HTMLEncode 和 URLEncode、Execute、Transfer 和 GetLastError。这些在动态网站开始都是不可缺少的重要方法。

1. CreateObject

CreateObject 方法用于创建已注册到服务器上的 ActiveX 组件的实例，通过使用 ActiveX 组件能够扩展 ActiveX 的能力。它是 Server 对象中最重要方法，在后面学习中读者会发现，许多功能都必须使用到它。



提示

这里的 ActiveX 组件包括所有 ASP 内置的组件，也可以是一些第三方的组件，但不能创建 ASP 的内置对象。默认情况下，CreateObject 方法创建的对象具有页作用域。

CreateObject 方法的语法格式如下：

```
Server.CreateObject (progID)
```

其中参数 **progID** 指定要创建的组件对象的类型。**progID** 的语法格式为:

```
[Vendor.]component[.Version]
```

其中 **Vendor** 表示组件的父对象, **component** 表示组件的名称, **Version** 表示组件的版本号。

例如:

```
<%  
Set MyAd = Server.CreateObject("MSWC.AdRotator")  
%>
```

上面的代码创建一个名为 **MyAd** 的 **MSWC.AdRotator** 服务器组件实例, **MSWC.AdRotator** 组件可用于在页面中自动轮换广告。

在默认情况下, 由 **Server.CreateObject** 方法创建的对象具有页作用域, 也就是说, 当前 ASP 页处理完成之后, 服务器将自动破坏这些对象。要创建有会话或应用程序作用域的对象, 可以使用 **<object>** 标记并设置 **Session** 或 **Application** 的 **Scope** 属性, 也可以在会话或应用程序变量中存储该对象。

例如, 在下面代码中, 当 **Session** 会话超时或 **Abandon** 方法被调用时, 存储在会话变量中的对象也将被破坏。

```
<%  
Set Session("ad") = Server.CreateObject("MSWC.AdRotator")  
%>
```

也可以通过将变量设置为 **Nothing** 或新的值来破坏对象, 例如:

```
<%  
Session ("ad") = Nothing  
%>  
<%  
Session ("ad") = " Other Value "  
%>
```

不能创建与内建对象同名的对象实例。例如, 下列脚本将返回错误:

```
<%  
Set Response = Server.CreateObject("Response")  
%>
```

2. MapPath

MapPath 方法可以将指定的虚拟路径转换为服务器上相应的物理路径, 这里的虚拟路径是指在 Web 服务器中建立的虚拟目录。其语法格式如下:

```
Server.MapPath(String)
```

其中 **String** 是用于指定虚拟路径的字符串。**MapPath** 方法不支持相对路径语法 **(.)** 或 **(..)**, 但将 **"\"** 字符与 **"/"** 字符视为相同。如果在 **String** 中以字符 **"\"** 或 **"/"** 开始, 说明 **String** 中的路径是一个完整的路径, 如果 **String** 中不以字符 **"\"** 或 **"/"** 开始, 说明 **String** 中指定的路径是当前 ASP 文件所在的路径。



提示

用户可以使用 **ServerVariables** 数据集合的服务器变量 **PATH_INFO** 映射当前文件的物理路径。

MapPath 方法不能检查路径在这个服务器下是否存在。因为 **MapPath** 转化路径时是不管这个路径是否在这个服务器下存在的。

如果读者能够发挥 **MapPath** 方法的作用，用它来将一个相对路径转化为一个物理路径，就可以在远程操作服务器上的文件。

- 使用 **ServerVariables** 数据集合的服务器变量 **PATH_INFO** 获取当前文件的物理路径。

```
<%= server.mappath( Request.ServerVariables("PATH_INFO")) %>
```

显示为：

```
C:\Documents and Settings\zhu\My Documents\site_server\untitled.asp
```

- 在下面示例中 URL 参数没有以 “/” 开头：

```
<%=server.mappath("MapPath.asp") %>
<%= server.mappath("Server/MapPath.asp") %>
```

显示为：

```
C:\Documents and Settings\zhu\My Documents\site_server\MapPath.asp
C:\Documents and Settings\zhu\My Documents\site_server\Server\MapPath.asp
```

- 下面示例中是以 “\” 开头：

```
<%=server.mappath("\MapPath.asp") %>
<%= server.mappath("\Server/MapPath.asp") %>
```

则显示为：

```
c:\inetpub\wwwroot\MapPath.asp
c:\inetpub\wwwroot\Server\MapPath.asp
```

- 也直接用 “/” 或 “\” 来获取服务器的主目录：

```
<%=server.mappath("/") %>
<%=server.mappath("\") %>
```

显示为：

```
c:\inetpub\wwwroot
c:\inetpub\wwwroot
```

3. HTMLEncode

HTMLEncode 方法是对指定的字符串进行 HTML 编码，其语法格式如下：

```
Server.HTMLEncode(String)
```

其中 **String** 指定要编码的字符串。当从服务器端向浏览器输出 HTML 标记时，浏览器就将其解释为 HTML 标记，并以指定的格式显示在浏览器上。如果想使浏览器原样输出 HTML 标记字符，不对这些标记进行解释，就可以使用本方法。例如：

```
<%
str = "<h2>HTML标题</h2>"
Response.Write str
Response.Write Server.HTMLEncode(str)
%>
```

在浏览器中预览的效果如图 3.13 所示。

4. URLEncode

URLEncode 方法将指定的字符串进行 URL 编码，其语法格式如下：

```
Server.URLEncode(String)
```

其中 **String** 指定要编码的字符串。URL 中的 ASCII 字符具有特殊的含义，使用编码的方法可以



图 3.13 HTMLEncode 方法使用比较

使这些字符加入 URL 时忽略它们自身的含义。例如：

```
<%  
Response.Write(Server.URLEncode("http://www.microsoft.com"))  
%>
```

则会显示：

```
http%3A%2F%2Fwww%2Emicrosoft%2Ecom
```

5. Execute

Execute 方法能够导入一个 ASP 文件并且执行它，导入的 ASP 文件将存在当前 ASP 文件中。这与 `<!--#include File="ASP 文件" -->` 包含命令相似，但 `#include File` 命令在文件执行前被执行，如果希望包含的文件随代码顺序执行，则使用 Execute 方法非常方便。其语法格式如下：

Server.Execute (Path)

其中参数 Path 表示指定将执行的 ASP 文件的路径。如果它是一个相对路径，那么它必须是一个和这个 ASP 应用程序相同目录的文件。

Execute 方法提供了一个将复杂 ASP 应用程序分解为小块单位来执行的方法。通过这种方法，读者可以实现模块化程序开发。

当 IIS 根据指定的 ASP 文件路径执行完这个 ASP 文件之后，就会自动返回以前的 ASP 文件。这个刚刚执行完的 ASP 文件有可能会改变了 HTTP head，但是和其他的 ASP 文件一样，当程序试图改变 HTTP head 时就会报错。

如果 ASP 文件与被导入的 ASP 文件中都含有相同的子函数，那么这些子函数只在本 ASP 文件中起作用。

6. Transfer

Transfer 方法能够把一个正在执行的 ASP 文件的所有信息传给另外一个 ASP 文件。其语法格式如下：

Server.Transfer (path)

其中参数 Path 表示将要接收信息的 ASP 文件的位置。当读者调用 Server.Transfer 方法时，所有内建对象的状态信息都会包含在这次传送之中。这就是说，所有在保存在 Session 或 Application 中的信息都会被传送，而且所有当前请求的信息都会被接收信息的 ASP 文件所接收。

7. GetLastError

GetLastError 方法返回一个 ASP Error Object 来描述一个错误信息。这个方法只适用于在 ASP 文件中发送任何内容给客户端之前。其语法格式如下：

Server.GetLastError ()

3.12ObjectContext 对象

ObjectContext 是一个以组件为主的事务处理对象，它可以保证事务的成功完成。使用 ObjectContext 对象，就允许程序在网页中直接配合 Microsoft Transaction Server (MTS) 使用，从而可以管理或开发高效率的 Web 服务器应用程序。

在 ASP 中使用 “@TRANSACTION” 关键字来标识正在运行的页面要以 MTS 事物服务器来处理。其语法格式如下：

```
<%@TRANSACTION=value%>
```

3.12.1ObjectContext 对象的方法

1. SetAbort

SetAbort 方法将当前事务声明为中止，在脚本结束时取消参与此事务的所有操作。

2. SetComplete

SetComplete 方法将当前事务标记为提交。如果事务中的所有组件都调用 SetComplete，事务将完成。

3.12.2ObjectContext 对象的事件

1. OnTransactionAbort

在事务失败时被激发。

2. OnTransactionCommit

在事务完成时被激活。

3.13 ASP 组件

利用 ASP 组件可以扩充 Web 应用程序的功能。ASP 组件均遵循 Microsoft 的 ActiveX 标准。ActiveX 组件是一个文件，一般扩展名为.dll。该文件包含执行某项或一组任务的代码，组件可以执行某些特定的功能，这样就不必自己去创建执行这些任务的代码。使用这些组件可以完成诸如 Web 广告条轮换、浏览器兼容、数据库存取、服务器端文件存取等多种功能。由于 ASP 提供的组件是在服务器端运行的，所以不需要客户端支持 ActiveX。

ASP 的内置组件有 13 种，而且读者还可以安装第三方提供的组件。ASP 组件与前面提到的 ASP 对象十分相似，利用它们都可以使 ASP 完成某些特定的功能，它们之间的区别在于：一个组件可能包含不止一个对象，而且在使用组件之前，要明显地创建一个实例。

在安装好 ASP 平台后，所有的 ASP 内置组件都被安装注册到服务器上，可以在 ASP 中方便地使用这些组件。调用内置组件的操作步骤如下。

01 必须创建组件的一个实例，并将该实例赋予一个变量。创建组件实例的方法是使用 ASP 的 Server 对象的 CreateObject 方法，该方法只有一个参数，即组件的注册名，用 Set 命令赋予变量。语法格式如下：

```
<%  
Set 变量名 = Server.CreateObject(组件名)  
%>
```



提示

另外，使用 HTML 的<Object>标签来创建对象实例，但必须为 RUNAT 属性提供服务器值，同时也要为将在脚本语言中使用的变量名提供 ID 属性组。使用注册名 (PROGID) 或者注册号码 (CLSID) 可以识别该对象。语法格式如下：

```
<object runat=server id="注册号" progid="注册名"></object>
```

可以在 Global.asa 中通过上述方法创建一个组件对象实例，这样就可以在 Session 或 Application 中使用该组件。

02 调用组件的方法和属性，完成相应的功能。

3.14 ASP 文件系统组件

ASP 不仅可以实现与用户交互的功能,还可以对服务器端的文件资源进行一定的操作,利用微软提供的 File Access 组件就可以完成对服务器本地文件系统的操作。由于 IIS 3.0 以前的版本在文件操作上有许多限制,在某种程度上影响了文件操作组件的使用和发展;而在 IIS 3.0 以后的版本中,ASP 提供了很多组方法、属性和集合来对文件进行操作处理,ASP 对文件系统的操作变得强大起来。利用 IIS 的 File Access 组件可以实现读写系统日志文件、存储用户数据、设计站点计数器等功能。

File Access 组件主要由 FileSystemObject 对象、TextStream 对象、File 对象、Folder 对象和 Drives 对象组成。各对象的功能见表 3.7。

表 3.7 File Access 组件对象列表

对象名称	功 能
FileSystemObject	包含了处理文件系统的所有基本方法
TextStream	用来读写文本文件
File	用来处理单个文件
Folder	用来处理文件夹
Drives	用来取得系统中所有可用驱动器的信息

在 File Access 组件的对象中最常用的是 FileSystemObject 和 TextStream 对象。下面以 FileSystemObject 对象为重点展开介绍。

FileSystemObject 对象简称为 FSO,或者称为 FSO 组件,该组件可以用来处理驱动器、文件夹以及文件。它可以检测并显示出系统驱动器的信息分配情况,创建、改变、移动和删除系统文件夹,并能检测一些给定的文件夹是否存在,提取文件夹信息(如名称、被创建或最后一次修改的日期等)。

创建 FileSystemObject 对象的实例方法如下:

```
Set fso=Server.CreateObject("Scripting.FileSystemObject")
```

由于 FileSystemObject 等对象都存在于 Microsoft Scripting 运行时库中,它相当于 Java 或 .NET 语言中的类库。因此在实例化时,必须指明 Scripting 运行时库。

3.14.1 系统驱动器操作

系统驱动器的操作主要包括对服务器端计算机的驱动系统的信息提取,例如,使用 GetDriveName 方法可以提取驱动器的名称,使用 GetDrive 的方法提取标准驱动器名。

1. GetDriveName

GetDriveName 方法可以提取指定路径中驱动器的名称,其语法格式如下:

```
object.GetDriveName(path)
```

其中 object 表示 FileSystemObject 对象的实例名称, path 指定路径。如果无法确定驱动器,则 GetDriveName 方法将返回零长度字符串("")。

例如:

```
<%
Set fso = Server.CreateObject("Scripting.FileSystemObject")
drv = fso.getDriveName("d:")
```

```
Response.Write drv
%>
```

上面代码将返回“d:”，使用`<%=fso.GetDrive("d:")%>`则可以提取标准的“D:”，所以一般这样写`fso.GetDrive(fso.GetDriveName(drvPath))`来提取某个具体的驱动盘符。

2. GetInfo

`GetInfo` 方法可以提取指定驱动器的具体信息。其语法格式如下：

`object.GetInfo(path)`

例如：

```
<%
Function ShowDriveInfo(drvPath)
    Dim fso, drv, s
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set drv = fso.GetDrive(fso.GetDriveName(drvPath))
    s = "驱动盘" & drv & "的卷标是: "
    s = s & drv.VolumeName & "<br>"
    s = s & "总计空间: " & drv.TotalSize & "<br>"
    s = s & "剩余空间: " & drv.FreeSpace & "<br>"
    s = s & "文件类型: " & drv.DriveType & "<br>"
    s = s & "文件系统: " & drv.FileSystem
    Response.Write s
End Function
%>
<%
on error resume next
whatpath=request.form("path")
if whatpath<>"" then
    ShowDriveInfo(whatpath)
end if
%>
<form action="" method="post">
    <input name="path">
    <input type="submit">
</form>
```

在上面代码中，先定义一个通用函数，该函数能够根据用户输入的路径检测该路径对应的驱动器，并提取相关信息。然后建立一个交互表单，当用户输入驱动器的路径之后，调用通用函数，测试驱动器并显示详细信息，如图 3.14 所示。



(a)



(b)

图 3.14 提取服务器端计算机盘符信息

(a) 输入盘符; (b) 显示该盘符详细信息

其中 `drv.TotalSize` 和 `drv.FreeSpace` 返回的是字节数，读者可以用 `FormatNumber()` 函数对该字节进行处理，例如，`FormatNumber(Drive.TotalSize/1024,0)` 就可以知道磁盘有多少容量。

drv.DriveType 表示文件类型, 返回 2 就表示该磁盘为硬盘驱动器, 1 表示软盘驱动器, 4 表示光盘驱动器等。

3.14.2 系统文件夹操作

系统文件夹的操作主要包括: 提取文件夹、创建文件夹、删除文件夹、复制文件夹、移动文件夹等。下面重点介绍文件夹常用操作。

1. GetFolder

GetFolder 方法可以获取指定的路径中某文件夹相应的 Folder 对象。其语法格式如下:

object.GetFolder(folderspec)

其中 folderspec 表示指定文件夹的路径(绝对路径或相对路径)。如果指定文件夹不存在, 则会出现错误。例如:

```
<%
Set fso = CreateObject("Scripting.FileSystemObject")
Set fldr = fso.GetFolder("c:\Program Files")
Response.Write "文件夹名称是: " & fldr & "<br>"
If fldr.IsRootFolder = True Then
    Response.Write "该文件夹是一根文件夹" & "<br>"
Else
    Response.Write "该文件夹不是根文件夹" & "<br>"
End If
Response.Write "驱动器名字是: " & fldr.Drive & "<br>"
%>
```

在上面代码中首先初始化 FSO 组件, 然后用语句 Set fldr = fso.GetFolder("c:\Program Files") 设定 fldr 对象被赋值, 以方便后面代码引用。fldr.IsRootFolder 即为判断该文件夹是否为一根文件夹, 返回值为布尔值, fldr.Drive 则显示该文件夹所在的驱动器盘符。

2. CreateFolder

CreateFolder 方法可以创建文件夹。其语法格式如下:

object.CreateFolder(foldername)

其中 foldername 是一个字符串表达式, 指明要创建的文件夹。如果指定的文件夹已经存在, 则会出现错误。因此, 创建文件夹之前应使用 FolderExists 方法判断该文件夹是否存在。例如:

```
<%
Sub AddNewFolder(path, folderName)
    Dim fso, f, fc, nf
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f = fso.GetFolder(path)
    Set fc = f.SubFolders
    If folderName <> "" Then
        if fso.FolderExists(path & "/" & folderName) Then
            Response.Write("该文件夹已经存在!")
        Else
            Set nf = fc.Add(folderName)
        End if
    Else
        Set nf = fc.Add("新建文件夹")
    End If
End Sub
%>
<%
AddNewFolder Server.MapPath("/"), "myfolder"
%>
```

上面代码将在 C:\inetpub\wwwroot 文件夹下建立一个文件夹 myfolder, 如果再次运行该

程序代码, 则会提醒该文件夹已经存在, 如果没有指定文件夹的名称, 则新建文件夹的名称为“新建文件夹”。

3. DeleteFolder

DeleteFolder 方法可以删除文件夹和其中的内容。其语法格式如下:

```
object.DeleteFolder folderspec[,force]
```

其中 folderspec 参数表示要删除的文件夹名称; folderspec 参数表示在路径的最后一个组成部分中可以包含通配符; force 参数是一个布尔值, 如果要删除只读文件夹, 则该值为 True, 否则为 False (默认值)。

DeleteFolder 方法不区分文件夹中是否包含内容。无论文件夹是否包含内容, 都将删除该文件夹。

如果未找到匹配文件夹, 则会出现错误。DeleteFolder 方法在遇到出现的第一个错误时停止。该方法不会撤销错误发生前所作的任何更改。

例如, 在上面实例基础上删除已建立的文件夹 myfolder, 代码如下:

```
<%
myfolder =Server.MapPath("/") & "myfolder"
Set fso = CreateObject("Scripting.FileSystemObject")
If not fso.FolderExists(myfolder) Then
    Response.Write("该文件夹不存在!")
Else
    fso.DeleteFolder(myfolder)
    Response.Write "myfolder文件夹已删除"
End If
%>
```

需要注意的是, 删除操作不会弹出提示信息。上机试验时要谨慎处理, 特别是对于系统文件夹。

3.14.3 系统文件夹操作

FileSystemObject 对象除了可以对驱动器、文件夹操作以外, 功能最强大的就是对文件的操作。它可以用来记数、内容管理、搜索还可生成动态 HTML 页面等。

1. OpenTextFile

OpenTextFile 方法可以打开某个文件并返回一个 TextStream 对象, 常用来打开文本文件。该方法还可以读取、写入该对象或将其追加到文件。语法格式如下:

```
object.OpenTextFile(filename[, iomode[, create[, format]]])
```

其中 object 表示 FileSystemObject 对象的实例名称; filename 表示要打开的文件名称; iomode 参数设置输入/输出模式, 取值包括 ForReading、ForWriting 和 ForAppending, 详细说明见表 3.8; create 设置当指定的 filename 不存在时是否能够创建新文件, 允许创建新文件时为 True, 否则为 False, 默认值为 False; format 参数指出以何种格式打开文件, 详细说明见表 3.9, 若忽略该参数, 则文件以 ASCII 格式打开。

表 3.8 iomode 参数设置列表

常 数	取 值	说 明
ForReading	1	以只读模式打开文件。不能对此文件进行写操作
ForWriting	2	以只写方式打开文件。不能对此文件进行读操作
ForAppending	8	打开文件并在文件末尾进行写操作

表 3.9 format 参数设置列表

常 数	取 值	说 明
TristateUseDefault	-2	以系统默认格式打开文件
TristateTrue	-1	以 Unicode 格式打开文件
TristateFalse	0	以 ASCII 格式打开文件

例如, 下面示例用来打开一个文本文件。具体操作步骤如下。

01 新建一个 info.txt, 在该文本文件中输入下面信息:

姓名: 朱印宏
爱好: 编程

02 在文本文件同一目录下再建立一个文件, 命名为 opentxt.asp, 在其中输入下面代码:

```
<%
whichfile=server.mappath("info.txt")
Set fso = CreateObject("Scripting.FileSystemObject")
Set txt = fso.OpenTextFile(whichfile,1)
rline = txt.ReadLine
rline = rline & "<br>" & txt.ReadLine
Response.Write rline
txt.Close
%>
```

03 保存并运行 opentxt.asp, 在浏览器中读取文本文件内容的效果如图 3.15 所示。

打开文本文件后, 读者需要通过 ReadLine 方法读取文本中的一整行, 如果还需要继续读取下一行, 则使用 ReadLine 方法。当然还有其他的读取方法, 例如, Read(number)方法可以读取指定数量的字符, ReadAll 方法可以返回文本中的全部内容。



图 3.15 打开文本文件

2. CreateTextFile

CreateTextFile 方法能够创建指定文件并返回 TextStream 对象, 该对象可用于创建文件。其语法格式如下:

```
object.CreateTextFile(filename[, overwrite[, unicode]])
```

其中 overwrite 参数是一个布尔值, 指明是否可以覆盖现有文件。如果可覆盖文件, 该值为 True; 如果不能覆盖文件, 则该值为 False。如果省略该参数, 则不能覆盖现有文件。

Unicode 参数也是一个布尔值, 指明是否以 Unicode 或 ASCII 文件格式创建文件。如果以 Unicode 文件格式创建文件, 则该值为 True; 如果以 ASCII 文件格式创建文件, 则该值为 False。如果省略该参数, 则默认为创建 ASCII 文件。

例如:

```
<%
whichfile=server.mappath("info.txt")
Set fso = CreateObject("Scripting.FileSystemObject")
Set MyFile = fso.CreateTextFile(whichfile,True)
MyFile.WriteLine("姓名: 朱印宏")
MyFile.WriteLine("爱好: 编程")
MyFile.Close
%>
<a href="<%=whichfile %>">查看内容</a>
```

下面再来练习一个综合实例——故事接龙。本实例涉及文本的打开和写入操作。要实现接龙就需要先要显示原有故事，然后把用户添加故事写入文件。这其中的写入文件应设置为追加写入。具体操作步骤如下。

01 新建一个 story.txt，在该文本文件中可以不输入任何信息。

02 在同一目录下再建立一个文件，命名为 story.asp，在其中输入下面代码：

```
<%
If not request.Form("NextLine")="" then
    Set fso=Server.CreateObject("Scripting.FileSystemObject")
    textfile1=Server.MapPath("story.txt")
    set cnrs=fso.OpenTextFile(textfile1,8)
    cnrs.WriteLine(Request.Form("NextLine"))
    cnrs.Close
end if
%>
<h2 align="center">故事接龙</h2>
故事内容:
<%
Set fso=Server.CreateObject("Scripting.FileSystemObject")
textfile1=Server.MapPath("story.txt")
set cnrs=fso.OpenTextFile(textfile1,1)
while not cnrs.AtEndOfStream
    Response.Write "&nbsp;" & cnrs.ReadLine
wend
cnrs.close
%>
<hr>
<form method="post" action="story.asp">
请续写故事的结尾:<input name="NextLine" type="text" size="70">
<input type="submit" value="提交">
</form>
```

03 保存并运行 story.asp，在浏览器中显示效果如图 3.16 所示。



图 3.16 故事接龙游戏效果

3. GetFile

GetFile 方法可以获取指定路径中某个文件相对应的 File 对象。其语法格式如下：

```
object.GetFile(filespec)
```

其中 object 参数表示 FileSystemObject 对象的实例名称，filespec 参数表示指定文件的路径（绝对路径或相对路径）。如果指定文件不存在，则会出现错误。

例如:

```
<%
whichfile=Server.MapPath("myfile.txt")
Set fso = CreateObject("Scripting.FileSystemObject")
Set fl = fso.CreateTextFile(whichfile,true)
fl.Close
Set f2 = fso.GetFile(whichfile)
s = "文件名称: " & f2.name & "<br>"
s = s & "文件短路径名: " & f2.shortPath & "<br>"
s = s & "文件物理地址: " & f2.Path & "<br>"
s = s & "文件属性: " & f2.Attributes & "<br>"
s = s & "文件大小: " & f2.size & "<br>"
s = s & "文件类型: " & f2.type & "<br>"
s = s & "文件创建时间: " & f2.DateCreated & "<br>"
s = s & "最近访问时间: " & f2.DateLastAccessed & "<br>"
s = s & "最近修改时间: " & f2.DateLastModified
response.write(s)
%>
```

运行代码则显示效果如图 3.17 所示。

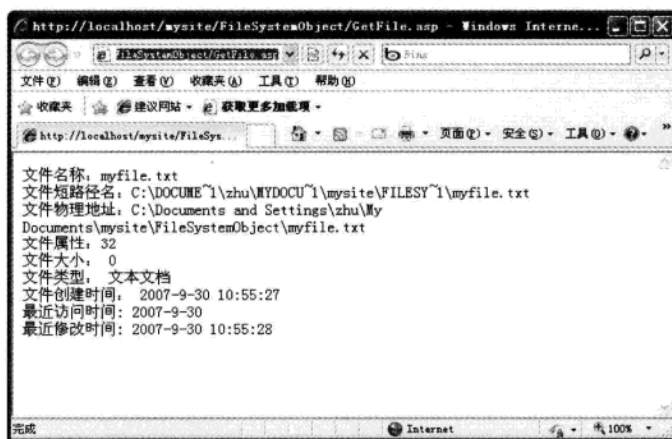


图 3.17 获取文件详细信息效果

其中 attributes 返回的数值 32 表示上次备份后已更改的文件, 该参数取值的详细说明见表 3.10。

表 3.10 attributes 参数设置列表

常 数	取 值	说 明
Normal	0	普通文件
ReadOnly	1	只读文件
Hidden	2	隐藏文件
System	4	系统文件
Directory	16	文件夹或目录
Archive	32	上次备份后已更改的文件
Alias	1024	链接或快捷方式
Compressed	2048	压缩文件

3.15 实践总结

本章重点介绍了 ASP 的基本概念、语法以及 ASP 对象和组件，用户要熟练掌握本章知识，为后面实例制作奠定基础。不断掌握使用 ASP 编程技巧后，用户可以创建更复杂的脚本，还可以便捷地使用 ActiveX 组件来执行复杂的任务，如连接数据库来存储和检索信息。如果已经掌握了一种脚本语言，如 VBScript，用户将会发现 ASP 是快速创建 Web 应用程序的一个非常灵活的方法。没有编写 Web 应用程序的经验也没有关系，只要有兴趣，就能够轻松掌握编写 ASP 动态网页的方法。



动态网页都需要脚本语言来进行逻辑控制，离开了脚本语言，整个网页只能显示为静态信息或效果。目前比较流行的脚本语言有两种：VBScript 和 JavaScript。这些脚本语言都可以在 ASP 服务器中使用，但 ASP 服务器脚本默认语言为 VBScript。VBScript 也是 ASP 服务器中使用最广泛的脚本语言。本章将重点介绍 VBScript 脚本语言的语法规则和使用技巧，为用户学习和开发动态网站奠定基础。JavaScript 在客户端浏览器中使用比较广泛，限于篇幅就不再详细介绍 JavaScript 脚本语言的语法规则。

技术要点

- 了解网页脚本语言
- 认识 VBScript 基本语法
- 熟悉 VBScript 数据类型
- 认识 VBScript 的变量和常量
- 能够灵活使用数组
- 能够使用运算符
- 掌握条件语句和循环语句的综合运用
- 掌握子过程和函数的使用

4.1 脚本语言简介

网页脚本语言是嵌入在网页中的一小段程序，是介于 HTML 和 Java、Visual Basic 等编程语言之间的一种逻辑控制脚本。脚本语言的最大优点是编写简单，可以在任何文本编辑器中编写，不需要编译就可以在网页浏览器或者 Web 服务器中被解释和执行。

目前在 Web 设计和开发中，比较流行的脚本语言有两种：VBScript 和 JavaScript。VBScript 是 Microsoft Visual Basic 编程语言的简化版。

在网页中定义脚本语言的类型主要有 3 种途径。

- 在 IIS 服务器环境中配置应用程序的脚本语言类型，如图 4.1 所示。
- 在网页中定义脚本语言的类型，例如，可以用如下格式设置网页包含脚本语言的类型为 VBScript：

```
<%@LANGUAGE="VBSCRIPT"%>
```



图 4.1 在 IIS 中配置脚本语言类型



提示

该行语句要在任何一行脚本语言命令之前定义。

- 使用<script>标记定义所包含代码的脚本语言，例如：

```
<html>
<head><title>VBScript代码示例</title></head>
<body>
<script type="text/vbscript">
    MsgBox("Hello,World!")
</script>
</body>
</html>
```

在网页中插入 VBScript 脚本时常使用<script>标记，其中 type="text/vbscript"属性定义<script>和</script>标记之间的代码是 VBScript 脚本语言。用户也可以使用 language="vbscript"属性指明脚本语言类型，但是 language 属性没有被 W3C 定义为 HTML 标准属性，已不再推荐使用，所以用户应少用。

使用 VBScript 和 JavaScript 都可以编写服务器端脚本，也可以编写客户端脚本。

4.1.1 服务器端脚本

服务器端脚本在 Web 服务器上被执行，由服务器根据脚本的执行结果生成相应的 HTML 页面并发送到客户端浏览器中显示。只有服务器端脚本才能真正实现动态信息显示效果，不过借助现在比较流行的 Ajax 技术，用户使用 JavaScript 脚本语言也可以实现客户端页面信息的动态更新。服务器端脚本的执行不受浏览器的限制，脚本在网页通过网络传送给浏览器之前被执行，Web 浏览器收到的只是标准的 HTML 文件。

在 ASP 中编写服务器端脚本的方法有如下两种。

- 使用分隔符<%和%>将脚本包含起来。例如：

```
<%@LANGUAGE="VBSCRIPT"%>
<head>
</head>
<body>
<% Response.Write("欢迎您光临!") %>
</body>
</html>
```

上面示例代码使用了 `Response.Write` 语句命令向客户端浏览器上输出动态文本信息。

- 使用 `<script>` 和 `</script>` 标记, 并在 `<script>` 标记中定义 `runat=server` 属性, 表示脚本在服务器端被执行。例如:

```
<%@LANGUAGE="VBSCRIPT"%>
<head>
<script language="vbscript" runat="server">
Sub welcome
    Response.Write("欢迎您光临!")
end sub
</script>
</head>
<body>
<%Call welcome%>
</body>
</html>
```

上例中定义了一个过程, 然后在程序中调用该过程, 并将结果发送到客户端。

总之, 服务器端脚本的编写可以嵌入在 HTML 中, 也可以单独编写为过程由程序调用执行。



提示

在服务器端使用 VBScript 时, 用户页面元素 (如 `MsgBox` 和 `InputBox`) 将不能在服务器端脚本中使用, 它们只能在客户端执行。另外, 在服务器端的脚本中, 也不要使用 VBScript 函数 `CreateObject` 和 `GetObject`, 而要使用 ASP 的对象 `Server` 的 `CreateObject` 方法来实现。

4.1.2 客户端脚本

客户端脚本只能够在浏览器中被解释执行。由于客户端脚本是随着 HTML 页面下载到客户端浏览器中, 在用户本地计算机中执行, 因此其执行速度明显快于服务器端脚本。客户端脚本常用于做简单的客户端验证, 例如, 用户表单信息验证或者设计网页特效等。

在 ASP 中, 必须把客户端脚本代码写在 `<script>` 和 `</script>` 标记之间, 并将其嵌入到 HTML 页面中。

一般来说, 脚本代码可以放在 HTML 文档的任何地方, 但是用户习惯于在 HTML 文档的 `<head>` 标记区域内编写代码, 以使所有脚本代码集中放置。这样可以确保在 `<body>` 页面区域调用代码之前所有脚本代码都被读取并解码。

但是也有一些例外情况, 例如, 在表单中嵌入内部 VBScript 脚本代码, 用来响应表单中对象的事件, 以实现响应提交按钮的单击事件。

```
<html>
<head><title>测试按钮事件</title></head>
<body>
<form name="form1">
    <input type="button" name="button1" value="单击">
    <script for="button1" event="onclick" language="vbscript">
        MsgBox "按钮被单击!"
    </script>
</form>
</body>
</html>
```

在上面代码中, 通过定义 `for="button1"` 属性, 把表单中的 VBScript 脚本绑定到按钮对象上, 通过定义 `event="onclick"` 属性, 指明触发该脚本的事件是鼠标单击。也就是说如果浏览者单击“单击”按钮即可以调用脚本并执行。

大多数 VBScript 脚本代码存储在 Sub 或 Function 过程中, 仅在其他代码要调用它们时才可以执行。不过, 用户也可以将 VBScript 代码放在过程之外, 直接包含在<script>标记之中。这类代码仅在 HTML 页面加载时执行一次。这样就可以实现加载页面时初始化数据或动态地改变页面的外观。

4.2 VBScript 脚本语言概述

VBScript (Visual Basic Script) 是程序开发语言 Visual Basic 家族的一员。作为微软公司开发的一种脚本语言, 它是 Visual Basic 的一个子集, 编程方法与 Visual Basic 基本相同, 但 Visual Basic 中的一些高级功能在 VBScript 中都被删除了, 这主要是从脚本的可移植性、运行速度和安全性等方面考虑的。

VBScript 将灵活的 Script (脚本) 应用于更广泛的领域, 它可以应用于 Internet Explorer 浏览器中的客户端网页 Script 和 IIS 中的 Web 服务器 Script。

如果用户已经了解了 Visual Basic 或者 Visual Basic for Applications, 就会很快掌握 VBScript 语言。即使你没有学过 Visual Basic, 只要学会 VBScript, 也能够使用所有的 Visual Basic 语言进行程序设计。

VBScript 脚本语言与 Visual Basic 编程语言存在如下区别。

- VBScript 不完全支持 Visual Basic 中的类及相关特性, 用<script>和</script>标记只能建立过程和函数, 新版本 VBScript 5 及更高版本支持简单的类定义。
- VBScript 只有一种数据类型, 即 Variant 类型, 而 Visual Basic 包含大多数程序语言所具有的数据类型。
- VBScript 不支持控件数组。
- VBScript 不支持 Visual Basic 中的文件操作功能, 这可以防止对客户机造成的危害。
- VBScript 不能调用 API 函数, 因为 API 函数调用会危及客户机安全。
- Visual Basic 支持 Screen、Printer、App、Debug、Err 和 Clipboard 等系统对象, 但 VBScript 只支持 Err 对象, 该对象用于为应用程序提供运行时的错误处理。
- VBScript 不提供调试功能, 没有调试窗口。

4.3 VBScript 数据类型

变量是用来表示内存中的一段存储空间, 这段空间被用来存储数据。变量都有名称和数据类型。变量的名称用来对内存中指定的存储空间的引用, 而变量的数据类型决定了如何将数据存储到计算机的内存中。在声明变量时也可指定它的数据类型。所有变量都具有数据类型, 以决定能够存储哪种数据。

VBScript 只有一种数据类型——Variant。Variant 是一种特殊的数据类型, 根据使用的方式, 它可以包含不同类别的信息。因为 Variant 是 VBScript 中唯一的数据类型, 所以它也是 VBScript 中所有函数的返回值的数据类型。

最简单的 Variant 可以包含数字或字符串信息。如果 Variant 用于数字上下文中时, 将被作为数字类型来处理; 如果 Variant 用于字符串上下文中时, 将被作为字符串类型来处理。这就是说, 如果使用看起来是数字的数据, 则 VBScript 会假定其为数字并以适用于数字的方式

处理。与此类似,如果使用的数据可能是字符串,则 VBScript 将按字符串来处理。当然,用户也可以将数字包含在引号“”中使其成为字符串。

除了简单的数字类型或者字符串类型以外,Variant 可以进一步区分数值信息的特定含意。例如,使用数值信息表示日期或时间。如果此类数据在与其他日期或时间数据一起使用时,结果也总是表示为日期或时间。当然,从 Boolean 值到浮点数,数值信息是多种多样的。Variant 包含的数值信息类型被称为子类型。大多数情况下,可将所需的数据放进 Variant 中,而 Variant 也会按照最适用于其包含的数据的方式进行操作。

Variant 包含的数据子类型见表 4.1。

表 4.1 Variant 类型变量的子数据类型列表

类型名称	解 释	说 明
Byte	字节	以一个无符号字节的二进制码存储,包含 0~255 之间的整数
String	字符串	由 ASCII 字符组成的字符串,包含可变长度字符串,最大长度可为 20 亿个字符。如果长度为 0 (即不包含任何字符),称为空字符串。在 VBScript 中,字符串由双引号作为定界符
Integer	整型	用 2 个字节(16 位)的二进制码表示,包含-32,768~2,767 之间的整数
Long	长整型	用 4 个字节(32 位)的二进制码表示,包含-2,147,483,648~2,147,483,647 之间的整数
Single	单精度浮点数	用 4 个字节(32 位)的二进制码表示,可精确到 7 位十进制数。负数范围为-4.402823E38~-1.401298E-45,正数范围为 1.401298E-45~4.402823E38
Double	双精度浮点数	用 8 个字节(64 位)的二进制码表示,可精确到 15 或 16 位十进制数。负数范围为-1.79769313486232E308~-4.94065645841247E-324,正数范围为 4.94065645841247E-324~1.79769313486232E308
Currency	货币	支持小数点右面 4 位和左面 15 位,是一个精确的定点类型,适用于货币计算。取值范围为-922337203685477.5808~922337203685477.5807
Boolean	布尔型	用 2 个字节存储,包含逻辑值,取值只能为 True 或 False
Date/Time	日期/时间	表示日期和时间值。日期值的有效范围从公元 100 年 1 月 1 日到公元 9999 年 12 月 31 日。时间值从 00:00:00~23:59:59。在代码中用日期和时间值,必须用一对# 将其括起来
Empty	空类型	声明一个变量后,如果还没有初始化,该变量的值是 Empty。这个值是自动分配的,对于数值变量,值为 0;对于字符串变量,值为零长度字符串(“”)。可以用 IsEmpty()函数测试变量是否已被初始化。当变量含 Empty 值时,可以在表达式中使用,至于是将其作为 0 还是作为零长度字符串来处理,要根据具体的表达式来定。只要将任何值(包括 0、零长度字符串或 Null)赋予给变量,Empty 值就会消失,而将关键字 Empty 赋予变量,就可以将变量恢复为 Empty 值
Null	空值	如果将关键字 Null 赋予变量,则使变量不包含任何有效的数据。如果表达式中包含 Null,那么计算结果总是 Null。将含 Null 的变量或计算结果作为参数传给大多数函数,将使函数返回 Null。可以使用 IsNull()函数测试表达式是否不含任何有效的数据
Object	对象	用 4 个字节存储,用来表示引用程序所能识别的任何对象
Error	错误	保存程序产生的错误代码

用户可以使用转换函数来转换数据的子类型。另外,可使用 VarType 函数返回数据的 Variant 子类型,在下面章节中还将详细介绍。

4.4 VBScript 变量

变量是一种使用方便的占位符,用于引用计算机内存地址,该地址可以存储脚本运行时

可更改的程序信息。使用变量并不需要了解变量在计算机内存中的地址，只需要通过变量名引用变量就可以查看或更改变量的值。在 VBScript 中只有一个基本数据类型 (Variant)，因此所有变量的数据类型都是 Variant。

在 VBScript 中，可以先声明变量再使用变量（显式声明），也可以不声明变量而直接使用（隐式声明）。在脚本中显式声明变量可以使用 Dim 语句、Public 语句和 Private 语句。

4.4.1 使用 Dim 语句声明变量

Dim 语句可以声明变量并为变量分配存储空间。语法格式如下：

```
Dim varname([[subscripts]])[,varname([[subscripts]])]...
```

其中 varname 表示变量的名称，subscripts 表示数组变量的维数，最多可以声明 60 维的数组。声明多个变量时，使用逗号分隔变量。

Dim 语句可以声明脚本级变量，该变量可以用于脚本中的所有过程；Dim 语句也可以声明过程级变量，该过程只能用于过程中。

用带空圆括号的 Dim 语句可以声明动态数组。声明动态数组后，在过程内使用 ReDim 语句来定义该数组的维数和元素。但不能重新定义在 Dim 语句中已经显式声明维数的数组，否则会发生错误。

在过程中使用 Dim 语句时，通常将 Dim 语句放在过程的开始处。

例如：

```
Dim MyVars(5)      '声明一个具有5个元素的数组
Dim MyVars()       '声明动态数组
Dim MyVar, MyNum    '声明两个变量
```

4.4.2 使用 Public 语句声明变量

Public 语句可以定义公共变量，并为公共变量分配存储空间。语法格式如下：

```
Public varname([[subscripts]])[, varname([[subscripts]])]...
```

Public 语句声明的变量可用于全部脚本中的所有过程。

在使用引用对象的变量之前，必须用 Set 语句将某个已有对象赋予该变量。在赋值之前，所声明的对象变量被初始化为 Empty。

也可以使用带空圆括号的 Public 语句来声明动态数组。声明动态数组后，可在过程内使用 ReDim 语句定义该数组的维数和元素。声明数组变量的维数之后，如该数组变量的大小已在 Private、Public 或 Dim 语句中显式地指定，就不能重新声明，否则就会发生错误。

例如：

```
Public MyVars(5)      '声明一个具有5个元素的公共数组
Public MyVars()       '声明公共动态数组
Public MyVar, MyNum    '声明两个公共变量
```

4.4.3 使用 Private 语句声明变量

Private 语句可以定义私有变量，并为该变量分配存储空间。语法格式如下：

```
Private varname([[subscripts]])[, varname([[subscripts]])]...
```

Private 语句变量只能在声明该变量的脚本中使用。在过程中使用 Private 语句时，通常将

Private 语句放在过程的开始处。

例如：

```
Private MyVars(5)      '声明一个具有5个元素的私有数组
Private MyVars()       '声明私有动态数组
Private MyVar, MyNum   '声明两个私有变量
```

4.4.4 隐式声明变量

隐式声明变量就是直接在代码行中使用变量，而不需要事先声明。隐式声明变量使用方便，但缺点也很明显，那就是在后面程序中使用该变量时，如果把变量名拼错了，会很难查找错误。

为了避免隐式声明时写错变量名而引起问题，VBScript 提供了 Option Explicit 语句来强制显式声明。如果在程序中使用该语句，所有变量必须先声明，然后才能使用，否则会出错。

Option Explicit 语句必须位于 ASP 脚本语言指定命令之后、任何 HTML 文本或脚本命令之前。例如：

```
<%@LANGUAGE="VBSCRIPT"%>
<% Option Explicit %>
<%
Dim strUserName
Dim sngNumber
%>
```

Option Explicit 语句只影响用 VBScript 编写的脚本代码，对其他脚本代码不起作用。

4.4.5 变量命名规则

变量命名必须遵循 VBScript 的标准命名规则。VBScript 中的变量不区分大小写，变量命名必须遵循如下条件。

- 变量名称的第一个字符必须是字母。
- 名称中不能包含嵌入的句点。
- 变量名长度一般不能超过 255 个字符。
- 在被声明的作用域内变量名必须是唯一的。
- 变量名不能与 VBScript 的关键字同名。

为了提高程序的可读性，在给变量命名时应尽量保证变量名语义要清楚，便于识别，最好能在变量名中体现出变量的数据类型及变量中所存放数据的信息。

4.4.6 变量的作用域

变量的作用域是指变量的有效范围，因为变量被声明后不是在任何地方都可以使用，每个变量都有它的使用范围，在作用域内可以使用，在作用域外则变量无效。

变量的作用域是由声明它的位置决定。如果在过程中声明变量，则只有在该过程中的代码可以访问或更改变量值，此时变量具有局部作用域，并被称为过程级变量。如果在过程之外声明变量，则该变量可以被脚本中所有过程识别，称为脚本级变量，具有脚本级作用域。

例如：

```
<%@LANGUAGE="VBSCRIPT"%>
<% Option Explicit
```

```

Dim strUserName
strUserName = "Dreamweaver CS5"
Call username
Response.Write(strUsername)
Sub username
    Dim strUserName
    strUserName = "Fireworks CS5"
End Sub
%>

```

上面程序运行后返回结果为全局变量 `strUserName` 的值，即“Dreamweaver CS5”，因为，`strUserName = "Fireworks CS5"` 只能在过程 `username` 中有效。

全局变量只能在当前文件内访问。如果要从其他网页文件中访问当前文件内的变量，用户必须提供变量的 `Session` 或 `Application` 作用域，有关 `Session` 和 `Application` 作用域详细介绍可以参见第 3 章内容。

当用 `#include` 命令在当前文件中包含其他文件时，虽然包含文件的脚本存储在单独的文件中，但会被当做是当前文件的一部分，因此可以访问其中的变量。同时，用户也应该避免包含文件与当前文件中存在名称相同的变量，防止变量名发生冲突。

4.4.7 变量的存活期

变量存在的时间称为存活期。脚本级变量的存活期从被声明的一刻起，直到脚本运行结束。对于过程级变量，其存活期仅是该过程运行的时间，该过程结束后变量随之消失。

在调用过程时，局部变量是理想的临时存储空间，可以在不同过程中使用同名的局部变量，这是因为每个局部变量只被声明它的过程所识别。

4.4.8 变量赋值

给变量赋值可以使用赋值运算符，语法格式如下：

```
variable = value
```

变量在表达式的左边，要赋的值在表达式右边。例如：

```
MyName = "Dreamweaver CS5"
```

4.5 VBScript 常量

常量是具有一定含意的名称，用于代替数值或字符串。在程序执行期间，常量的值不会发生改变，可以在代码的任何位置使用常量代替实际值。

使用 `Const` 语句在 VBScript 中创建用户自定义常量，并给它们赋原义值。例如：

```

Const siteURL = "http://www.css8.cn/"
Const siteName = "样吧"

```

注意，字符串常量值包含在两个引号（" "）之间。这是区分字符串型常量和数值型常量的最明显的方法。日期常量值和时间常量值包含在两个 # 号之间。例如：

```
Const CutoffDate = #8-8-2008#
```

用户在使用 VBScript 常量时，建议采用一个命名方案以区分常量和变量，这样可以避免在运行脚本时对常量重新赋值。例如，可以使用“vb”或“con”作常量名的前缀，或将常量名的所有字母大写，将常量和变量区分开可以在开发复杂的脚本时避免混乱。

同时, VBScript 也提供了大量内部常量。由于这些常量在 VBScript 中事先被声明, 使用时就不需要再定义, 用户可以在代码中任意位置使用它们以表示特定的常量值。用户不能禁用内部常量, 因此不能以相同的名称创建自定义常量。

常量的便利就在于直接使用特定的数值但不需要记住具体的值。如果某些常量的值可能改变, 则使用常量将有助于提高程序的可维护性。由于这些常量已经在 VBScript 中得到定义, 在程序代码中不需要显式地声明它们, 在需要的地方直接使用它们就可以了。

4.6 VBScript 数组

一般情况下, 用户只需为声明的变量赋一个值。只包含一个值的变量称为标量变量。有时候, 将多个相关值赋给一个变量会更为方便, 因此可以创建包含一系列值的变量, 这种变量称为数组变量, 简称为数组。数组变量和标量变量都是以相同的方式声明的, 唯一的区别是声明数组变量时变量名后面带有小括号(), 小括号内可以指明数组的大小。

当用户批处理数据时, 最有效的方法就是使用数组。通俗说, 数组就是具有相同变量名的一组变量, 数组中包含的具体变量称为数组的元素, 不同数组元素之间通过数组的下标值加以区分。

VBScript 数组具有如下特点。

- 使用数组之前必须要进行定义, 然后才能使用。通常用 Dim 语句来定义数组。
- 数组下标的下界一律从 0 开始。
- 一个数组中可以包含各种类型数据元素。

在 VBScript 中, 数组有两种类型: 静态数组和动态数组。

4.6.1 静态数组

静态数组就是定义数组的大小之后, 不可以再定义该数组。例如, 下面声明了一个包含 6 个元素的一维数组:

```
Dim MyArray(5)
```

在基于 0 的数组中, 数组元素的数目总是括号中显示的数目加 1。这种数组称为固定大小的数组。

定义数组之后, 可以使用索引为数组的每个元素赋值, 例如:

```
MyArray(0) = 1  
MyArray(1) = 2  
MyArray(2) = 3  
MyArray(3) = 4  
MyArray(4) = 5  
MyArray(5) = 6
```

同理, 使用索引可以检索到所需的数组元素的数据。例如:

```
MyVariable = MyArray(4)
```

则变量 MyVariable 的值就等于 5。

除了一维数组外, 还可以定义多维数组, 数组的维数最大可以为 60。一般情况下, 用户最多使用 3 维或 4 维的数组。声明多维数组时用逗号分隔括号中每个表示数组大小的数字。例如, 下面声明一个 2 行 3 列的二维数组:

```
Dim MyTable(1,2)
```

在二维数组中，括号中第 1 个数字表示行的数目，第 2 个数字表示列的数目。同一维数组赋值方法相同，可以使用索引为数组的每个元素赋值，例如：

```
MyTable(0,0) = 1
MyTable(0,1) = 2
MyTable(0,2) = 3
MyTable(1,0) = 4
MyTable(1,1) = 5
MyTable(1,2) = 6
```

给数组赋值的方法是给数组中的各个元素分别赋值，如同每一个数组元素是一个独立的变量。在 VBScript 中，数组中的各个元素可以赋给不同数据类型。

4.6.2 动态数组

动态数组就是在脚本运行时大小可重定义的数组。当程序没有运行时，动态数组不会占据内存空间，当程序运行时才在内存中开辟数组空间。定义动态数组一般分为两步。

(1) 用 Dim 语句或 ReDim 语句声明一个不包含下标的数组（即空数组）。对于动态数组来说，括号中不包含任何数字。例如：

```
Dim MyArray()
ReDim AnotherArray()
```

要使用动态数组时，必须随后使用 ReDim 语句确定数组的维数和每一维的大小。例如，在下代码中，ReDim 先将动态数组的初始大小设置为 5，而后面的 ReDim 语句将数组的大小重新调整为 10。

```
ReDim MyArray(5)
...
ReDim Preserve MyArray(10)
```

在上面代码中，使用 Preserve 关键字表示在重新调整数组大小时，保留原来数组的内容。每次执行 ReDim 语句时，存储在数组中的当前值都会全部丢失。如果希望改变数组大小，而又不丢失数组中的数据，可以使用关键字 Preserve。

重新调整动态数组大小的次数是没有任何限制的，将数组的大小调小时将会丢失被删除元素的数据。但用户只能改变数组的个数，而不能改变数组的维数。

下面看一个动态数组的综合示例，请在网页中输入下面脚本代码：

```
<script type="text/vbscript" language="vbscript">
<!--
    dim score()                ''声明动态数组
    redim score(3)             ''重新定义数组大小为4个
    score(0) = 1               ''给数组赋值
    score(1) = 2
    score(2) = 3
    score(3) = 4
    for n=0 to ubound(score)   ''第一次显示数组score的4个变量值
        document.write(score(n) & "nbsp;   ")
    next
    document.write("<hr>")      ''插入水平分隔线
    redim score(4)             ''重新定义数组，增加数组的大小为5，原来的数组元素数值都被丢失
    score(4) = 5               ''定义score(4)
    for n=0 to ubound(score)   ''第二次显示数组score的5个变量值
        document.write(score(n) & "nbsp;   ")
    next
```


4.7 VBScript 运算符

对于开发人员来说,编程的核心是如何设计算法,即如何对数据进行加工处理。基本运算关系可以用一些简洁的符号描述,这些符号称为运算符(或称为操作符)。被运算的对象称为操作数,操作数可以是变量、常量,也可以是函数或对象等。

VBScript 拥有一套完整的运算符,包括算术运算符、比较运算符、连接运算符和逻辑运算符,具体说明如下。

4.7.1 算术运算符

算术运算符是最基本的操作运算符,一般用来实现数字的加、减、乘、除混合四则运算。主要包括 8 种运算符,见表 4.2。

表 4.2 算术运算符

运算符	功 能	运算符	功 能
-	负号	/	除法
+	加法	\	整数除法
-	减法	Mod	余数
*	乘法	^	指数

例如:

```
n = 15 + 5      'n等于20
n = 15 - 5      'n等于10
n = 15 * 5      'n等于75
n = 15 ^ 3      'n等于3375
n = 15 / 3      'n等于5
n = 15 \ 6      'n等于2
n = 15 MOD 6    'n等于3
```

4.7.2 比较运算符

比较运算符能够比较两个表达式,如果比较的结果是正确的,就会返回“True(真)”;如果比较结果是错误的,就会返回“False(假)”。一般在程序中可以设计根据比较运算符所返回的值,做不同的处理。常用比较运算符包括 6 种,见表 4.3。

表 4.3 比较运算符

运算符	功 能	运算符	功 能
=	等于	>	大于
<>	不等于	<=	小于等于
<	小于	>=	大于等于

例如:

```
15 + 5 = 14 + 6      '返回True
15 + 5 <> 14 + 6     '返回False
15 + 5 < 6 + 6       '返回False
15 + 5 > 6 - 4       '返回True
15 + 5 <= 30 - 10    '返回True
15 + 5 >= 10 + 10    '返回True
```

VBScript 还提供了 Is 运算符，它表示一种特殊的比较功能，语法格式如下：

```
result = object1 Is object2
```

其中 object1 和 object2 表示任意对象名。如果 object1 和 object2 都引用同一个对象，则 result 为 True；否则 result 为 False。可以使用许多方法使两个变量引用同一个对象。例如：

```
Set A = B
```

在上面代码中，使 A 引用的对象与 B 的对象相同。再如：

```
Set A = C
Set B = C
```

在上面代码中，使 A 和 B 引用的对象与 C 的对象相同。

当比较两个表达式时，如果不能准确确定比较的表达式的数据类型，则用户可以根据表 4.4 所示对表达式进行比较，以及返回的比较结果。

表 4.4 子数据类型与比较运算结果列表

条 件	比较规则
两个表达式都是数值	根据数值大小进行比较
两个表达式都是字符串	根据字符串编码顺序进行比较
一个表达式是数值，而另一个表达式是字符串	数值表达式小于字符串表达式
一个表达式为 Empty，而另一个为数值	根据数值大小进行比较，0 作为 Empty 表达式的值
一个表达式为 Empty，而另一个表达式为字符串	根据字符串编码顺序进行比较，零长度的字符串("")作为 Empty 表达式的值
两个表达式都为 Empty	则两个表达式相等

4.7.3 连接运算符

连接运算符共有 2 个：&和+。

&运算符能够强制两个表达式进行字符串连接，语法格式如下：

```
result = expression1 & expression2
```

其中 result 表示任意变量，而 expression1 和 expression2 表示任意表达式。如果任意表达式不是字符串时，它将被转换为 String 子类型；如果两个表达式都为 Null，则 result 也为 Null；如果仅有一个 expression 为 Null，则它和其他表达式连接时，该表达式被作为零长度字符串("")处理；如果任意表达式为 Empty 时，也可以按零长度字符串来进行处理。

+运算符既可以执行相加操作，又可以执行连接操作，VBScript 将根据表 4.5 所示条件决定具体操作。

表 4.5 +运算符操作条件列表

条 件	操作类型
两个表达式都是数值	相加
两个表达式都是字符串	连接
一个表达式是数值，另一个表达式是字符串	相加

因此在使用时要特别小心，一般建议用户使用&运算符进行字符串的连接以避免混淆，提供易理解的代码。

在下面表达式中：

```
result = expression1 + expression2
```

如果一个表达式或两个表达式都为Null表达式, 则 result 为 Null; 如果两个表达式都为 Empty, 则 result 为 Integer 子类型; 如果一个表达式为 Empty, 则返回另一个表达式作为 result。

4.7.4 逻辑运算符

逻辑运算符就是结合两个逻辑表达式, 并根据逻辑关系返回一个 True 或 False 值。常用逻辑运算符包括 4 种, 见表 4.6。

表 4.6 逻辑运算符

运算符	功 能
Not	对逻辑表达式执行逻辑非运算
And	对两个逻辑表达式执行逻辑“与”运算
Or	对两个逻辑表达式执行逻辑“或”运算
Xor	对两个逻辑表达式执行逻辑“异或”运算

例如:

```
not 50 > 10           '返回False
not 50 < 10           '返回True
50 > 10 And 10 < 20    '返回True
50 > 10 And 10 > 20    '返回False
50 < 10 And 10 < 20    '返回False
50 < 10 And 10 > 20    '返回False
50 > 10 Or 10 < 20     '返回True
50 > 10 Or 10 > 20     '返回True
50 < 10 Or 10 < 20     '返回True
50 < 10 Or 10 > 20     '返回False
50 > 10 Xor 10 < 20    '返回False
50 > 10 Xor 10 > 20    '返回True
50 < 10 Xor 10 < 20    '返回True
50 < 10 Xor 10 > 20    '返回False
```

Xor 运算符还可以对两个数值表达式中位置相同的位执行逐位比较, 例如, 在下面表达式中:

```
result = expression1 Xor expression2
```

VBScript 将根据表 4.7 所示逻辑关系设置 result 中相应的位。

表 4.7 逻辑运算符

如果 expression1 中的位是	如果 expression2 中的位是	则 result 为
0	0	0
0	1	1
1	0	1
1	1	0

4.7.5 运算符优先级

当一个表达式中包含多个运算符时, 将根据预定顺序计算每一部分, 这个顺序称为运算符优先级。可以使用小括号越过这种优先级顺序, 强制首先计算表达式的某些部分。运算时, 总是先执行小括号中的运算符, 然后再执行括号外的运算符。但是, 在括号中仍遵循标准运算符优先级。

当表达式包含多种运算符时，首先计算算术运算符，然后计算比较运算符，最后计算逻辑运算符。所有比较运算符的优先级相同，即按照从左到右的顺序计算比较运算符。算术运算符和逻辑运算符的优先级见表 4.8。

表 4.8 算术运算符和逻辑运算符的优先级

算术运算符		逻辑运算符	
描 述	运算符	描 述	运算符
求幂	^	逻辑非	Not
负号	-	逻辑与	And
乘	*	逻辑或	Or
除	/		
整除	\		
求余	Mod		
加	+		
减	-		
字符串连接	&		

在上面列表中，靠上的运算符拥有较大的优先级。当乘号与除号同时出现在一个表达式中时，将按从左到右的顺序计算乘、除运算符；当加与减同时出现在一个表达式中时，按从左到右的顺序计算加、减运算符。

字符串连接（&）运算符不是算术运算符，但是在优先级顺序中，它排在所有算术运算符之后和所有比较运算符之前。Is 运算符是对象引用比较运算符，它并不比较对象或对象的值，而只是进行检查，判断两个对象引用是否引用了同一个对象。

4.8 使用条件语句

脚本代码的运行顺序一般都是从上到下执行，这种逻辑顺序属于顺序结构。但在实际编程中经常会打乱这种结构顺序，根据需要而使用其他逻辑结构，例如，条件结构和循环结构。

使用条件语句和循环语句可以控制脚本的流程。使用条件语句可以编写出能够执行条件判断和重复操作的逻辑结构。在 VBScript 中可使用以下条件语句：

- If...Then...Else 语句
- Select Case 语句

4.8.1 If...Then...Else 语句

If...Then...Else 语句用于计算条件是否为 True 或 False，并且根据计算结果运行指定的语句。通常，条件是使用比较运算符对值或变量进行比较的表达式。If...Then...Else 语句可以按照需要进行嵌套。

1. 单行结构

单行结构的语法格式如下：

```
If 条件表达式 Then 语句体1 [Else语句体2]
```

其中，条件表达式的返回值如果为 True，则执行 Then 后面的语句体 1，否则执行 Else 后面的语句体 2。如果省略 Else 部分，则执行 If 后面的语句。

例如:

```
<script type="text/vbscript">
function greeting()
    i=hour(time)
    if i = 8 then document.write("上课了!")
    if i = 10 then document.write("上操了!")
    if i = 12 then document.write("午休了!")
end function
greeting()
</script>
```

2. 多行结构

多行结构的语法格式如下:

```
If 条件表达式1 Then
    [语句体1]
[Elseif 条件表达式2 Then
    [语句体2]]
...
[Else
    [语句体n]]
End If
```

VBScript 先测试条件表达式 1, 如果为 True, 则执行 Then 后面的语句体 1, 然后退出块结构, 执行下面的语句。如果条件表达式 1 的结果为 False, 再测试条件表达式 2, 依次类推。当找到一个为 True 的条件时, 就执行相应的语句体, 然后执行 End If 后面的语句。如果条件都不为 True, 如果有 Else, 则执行 Else 后的语句体 n, 如果没有 Else, 则一个语句也不执行, 直接执行 End If 后面的语句。

例如, 下面代码是一个多层嵌套的条件语句。

```
<script type="text/vbscript">
if a <> Empty then
    x = a
else
    if b <> Empty then
        x = b
    else
        if c <> Empty then
            x = c
        else
            x = "变量怎么都没有赋值啦"
        end if
    end if
end if
document.write(x)
</script>
```

多层嵌套条件语句还可以添加更多的条件嵌套。不过, 也可以改写为下面条件结构。

```
<script type="text/vbscript">
if a <> Empty then
    x = a
elseif b <> Empty then
    x = b
elseif c <> Empty then
    x = c
else
    x = "变量怎么都没有赋值啦"
end if
```

```
document.write(x)
</script>
```

这种结构更容易理解，也不容易出错，因此，建议用户使用后面一种条件结构实现多层条件嵌套。用户可以添加任意多个 ElseIf 子句以提供多种选择。使用多个 ElseIf 子句经常会变得很累赘。在多个条件中进行选择的更好方法是使用 Select Case 语句。

4.8.2 Select Case 语句

Select Case 语句是一种更高效的多条件分支结构，常用它代替多条件嵌套语句，以便在多个语句中有选择地执行其中一个。多条件分支结构比多条件嵌套语句更容易阅读。

多条件分支结构的语法如下：

```
Select Case 测试表达式
    [Case 表达式1
      [语句体1]]
    [Case 表达式2
      [语句体2]]
    ...
    [Case Else
      [语句体n]]
End Select
```

VBScript 先计算测试表达式，然后将表达式的值与每个 Case 后面的表达式的值进行比较。若相等，就执行该 Case 语句下的语句体。每个 Case 的值是一个或几个值的列表。如果在一个列表中有多个值，则要用逗号把值隔开。如果不止一个 Case 与测试表达式匹配，那么只对第一个匹配的 Case 执行与之相关联的语句体。如果在表达式列表中没有一个值与测试表达式相匹配，且有 Case Else 子句，则执行 Case Else 子句中的语句体 n，否则，一条语句也不执行。

例如：

```
<script type="text/vbscript">
Dim n
x = InputBox("请输入一个数字 (1~7)：")
if not isnumeric(x) then x = ""
select case x
    case ""
        MsgBox "输入错误，请输入数字！"
    case 1
        MsgBox "星期一"
    case 2
        MsgBox "星期二"
    case 3
        MsgBox "星期三"
    case 4
        MsgBox "星期四"
    case 5
        MsgBox "星期五"
    case 6
        MsgBox "星期六"
    case 7
        MsgBox "星期日"
    case else
        MsgBox "请输入1-7的数字！"
End select
</script>
```

4.9 使用循环语句

循环是用于重复执行一组语句，循环结构可以分为以下 3 类。

- 在条件变为 False 之前重复执行语句。
- 在条件变为 True 之前重复执行语句。
- 按照指定的次数重复执行语句。

在 VBScript 中可以使用下列循环语句。

- Do...Loop: 当（或者直到）条件为 True 时循环。
- While...Wend: 当条件为 True 时循环。
- For...Next: 指定循环次数，使用计数器重复运行语句。
- For Each...Next: 对于集合中的每项或数组中的每个元素，重复执行一组语句。

4.9.1 Do 循环

使用 Do...Loop 语句可以多次（次数不定）运行语句块。当条件为 True 时或条件变为 True 之前，重复执行语句块。在程序设计中，如果不知道循环的准确次数，可以使用 Do 循环。

Do...Loop 语句的语法格式有如下两种形式。

- 当条件为 True 时重复执行语句。

```
Do [While 循环条件]
    循环体
[Exit Do]
Loop
或者
Do
    循环体
[Exit Do]
Loop [While 循环条件]
例如:
<script type="text/vbscript">
Sub ChkFirstWhile()
    Dim a, b
    a = 0
    b = 30
    Do While b > 10
        b = b - 1
        a = a + 1
    Loop
    MsgBox "循环重复了 " & a & " 次。"
End Sub
ChkFirstWhile()
</script>
```

浏览器执行上面代码时会弹出提示对话框，提示循环执行了 20 次。上面代码也可以改为：

```
<script type="text/vbscript">
Sub ChkFirstWhile()
    Dim a, b
    a = 0
    b = 30
    Do
        b = b - 1
```

```

        a = a + 1
    Loop While b > 10
    MsgBox "循环重复了 " & a & " 次。"
End Sub
ChkFirstWhile()
</script>

```

上面的示例运行后, 执行循环的次数都是一样的, 即都为 20 次。

- 重复执行语句直到条件变为 True。

```

Do [Until循环条件]
    循环体
[Exit Do]
Loop
或者
Do
    循环体
[Exit Do]
Loop [Until循环条件]

```

上面两种循环语句的区别是: 对于包含 While 关键字的 Do 循环, 在循环条件为真时一直重复执行循环体, 直到循环条件不满足时退出 Do 循环; 对于包含 Until 关键字的 Do 循环, 在循环条件不为真时一直重复执行循环体, 直到循环条件为真时退出 Do 循环。

例如:

```

<script type="text/vbscript">
Sub ChkFirstWhile()
    Dim a, b
    a = 0
    b = 30
    Do Until b < 11
        b = b - 1
        a = a + 1
    Loop
    MsgBox "循环重复了 " & a & " 次。"
End Sub
ChkFirstWhile()
</script>

```

同理, 用户也可以把 Until 条件放在循环结构的后面, 例如:

```

<script type="text/vbscript">
Sub ChkFirstWhile()
    Dim a, b
    a = 0
    b = 30
    Do
        b = b - 1
        a = a + 1
    Loop Until b < 11
    MsgBox "循环重复了 " & a & " 次。"
End Sub
ChkFirstWhile()
</script>

```

上面的示例运行后, 执行循环的次数都是一样的, 即都为 20 次。

4.9.2 退出 Do 循环

在 Do 循环体中, 可以使用 Exit Do 语句强制退出 Do...Loop 循环。因为 Exit Do 语句只

是在某些特殊情况下使用(如出现死循环时),所以可使用 If...Then...Else 语句选择使用 Exit Do 语句。如果在某个假设的条件为 True 时,执行 Exit Do 语句退出循环。

例如:

```
<script type="text/vbscript">
Sub ExitExample()
    Dim a, b
    a = 0
    b = 9
    Do Until b = 10
        b = b - 1
        a = a + 1
        If b < 20 Then Exit Do
    Loop
    MsgBox "循环重复了 " & a & " 次。"
End Sub
ExitExample()
</script>
```

在上面示例中, b 的初始值将导致死循环。If...Then...Else 语句检查此条件,防止出现死循环。

4.9.3 While 循环

While 循环与 Do 循环都具有相同的应用环境,针对循环次数不确定的循环体,While...Wend 语句和 Do...Loop 语句可以相互替换。

While...Wend 语句的语法格式如下:

```
While 循环条件
    循环体
Wend
```

While 循环是先判断循环条件,根据循环条件的值来决定是否执行循环体。如果循环条件为真时执行循环体,直到循环条件不满足时退出 While 循环。

While...Wend 语句是为那些熟悉传统循环用法的用户提供的。但是由于 While...Wend 缺少灵活性,所以用户最好使用 Do...Loop 语句代替 While...Wend 语句。

4.9.4 For 循环

For 循环用来完成已知循环次数的循环,也称计数循环。For 循环含有一个计数变量,每重复一次循环,计数变量的值就会增加或减少。For...Next 语句的语法格式如下:

```
For 循环变量=初值 To 终值[ Step 步长]
    循环体
[Exit For]
Next[循环变量]
```

For 循环按指定的次数执行循环体。执行 For 循环时,先将循环变量设为初值。测试循环变量是否小于(当步长值为正)或大于(当步长值为负)终值,若是,则执行循环体,否则退出循环,执行 For 循环后面的语句。如果省略 Step 子句,那么默认步长值是 1。

例如:

```
<script type="text/vbscript">
Sub Total()
    Dim n, total
```

```

    For n = 0 To 20
        total = total + n
    Next
    MsgBox "20内整数和为 " & total & "。"
End Sub
Total()
</script>

```

在上面示例中, For 循环的步长为默认 1, 执行代码则可以求 20 内整数和。用户也可以自定义 For 循环的步长, 例如:

```

<script type="text/vbscript">
Sub Total()
    Dim n, total
    For n = 0 To 20 Step 2
        total = total + n
    Next
    MsgBox "20内偶数和为 " & total & "。"
End Sub
Total()
</script>

```

在上面示例中, 计数器变量 n 每次加 2。循环结束后, total 的值为 2、4、6、8、10、12、14、16、18 的偶数和。

要使计数器变量递减, 可以将 Step 设为负值。此时计数器变量的终值必须小于初值。例如:

```

<script type="text/vbscript">
Sub Total()
    Dim n, total
    For n = 20 To 0 step -2
        total = total + n
    Next
    MsgBox "20内整数和为 " & total & "。"
End Sub
Total()
</script>

```

在上面的示例中, 计数器变量 n 每次减 2。循环结束后, total 的值为 18、16、14、12、10、8、6、4 和 2 的偶数和。



提示

在 For 循环中, 可以使用 Exit For 语句在计数器变量达到终值之前退出 For...Next 语句。它与 Exit Do 语句用法相同。

4.9.5 For Each 循环

在 VBScript 中, 用户可以使用 For Each 循环。For Each 循环与 For...Next 循环相似, 但 For Each 循环只针对数组或对象集合中的每个元素重复执行一组语句。如果不知道一个数组或集合内有多少个元素时, 用 For Each 循环会非常方便地遍历数组或集合, 实现快速获取数组或集合内的元素值, 也可以快速为数组或集合内的元素赋值。For Each...Next 语句的语法格式如下:

```

For Each 下标变量 In 数组|集合
    语句体
Next [元素]

```

例如, 在下面示例中使用 For...Next 语句循环为一个数组元素赋值, 然后再使用 For Each...Next 语句把数组中所有元素的值读出来显示在页面中。

```

<script type="text/vbscript">
Sub myFor()
    Dim a(10)
    For n = 0 to UBound(a)
        a(n) = n
    Next
    For Each m in a
        document.write(a(m) & "<br>")
    Next
End Sub
myFor()
</script>

```

4.10 VBScript 过程

在 VBScript 中, 可以通过定义过程完成特定的功能。VBScript 根据过程是否有返回值将过程分为 Sub 过程 (子过程) 和 Function 过程 (函数)。



提示

给过程传递数据的途径是使用参数。参数被作为要传递给过程的数据的占位符。参数名可以是任何有效的变量名。使用 Sub 语句或 Function 语句创建过程时, 过程名之后必须紧跟括号。括号中包含所有参数, 参数间用逗号分隔。

4.10.1 Sub 过程基本用法

Sub 过程是包含在 Sub 和 End Sub 语句之间的一组 VBScript 语句, Sub 过程被调用执行后不会返回值。Sub 过程可以使用参数 (由调用过程传递的常数、变量或表达式)。如果 Sub 过程无任何参数, 则 Sub 语句必须包含空括号()。

声明 Sub 过程时, 必须指明过程的名称、是否带有参数以及构成其主体的代码。Sub 过程的语法格式如下:

```

[Public [Default] | Private] Sub 过程名称[(参数列表)]
    [语句]
    [Exit Sub]
    [语句]
End Sub

```

其中 Public 关键字表示 Sub 过程可以被所有脚本中的所有其他过程访问, Default 关键字只能与类中的 Public 关键字连用, 用来表示 Sub 过程是类的默认方法。在一个类中, 用户只能指定一个 Default 过程, Private 关键字表示 Sub 过程只能被声明该过程的脚本中的其他过程访问。

参数列表代表在调用时要传递给 Sub 过程的参数的变量列表, 用逗号隔开多个变量。参数包含可以包含下列语法和部分:

```
[ByVal | ByRef] varname[( )]
```

其中 ByVal 关键字表示该参数是按值传递的, ByRef 关键字表示该参数按引用传递的, varname 表示参数的变量名。

所谓按值传递就是允许过程访问变量的一个副本, 但是不能更改变量的实际值, 所谓引用传递就是参数向过程传递的是变量的地址, 如果在过程中更改参数变量的值, 则原变量的值也随之发生变化。

例如,在下面示例中为过程 `mysub()` 定义一个值传递参数 `a`, 然后调用该过程, 则变量 `a` 原来的值依然保持不变, 显示为 1。

```
<script type="text/vbscript">
Sub mysub(byval a)
    a = 2
End Sub
a = 1
mysub a
MsgBox a
</script>
```

下面, 把上面示例中的参数改为按引用传递, 代码如下:

```
<script type="text/vbscript">
Sub mysub(byref a)
    a = 2
End Sub
a = 1
mysub a
MsgBox a
</script>
```

执行上面代码, `a` 的值显示为 2。说明在调用过程中, 通过按引用传递参数改变了原来变量的实际值。

4.10.2 Sub 过程使用技巧

如没有显式使用 `Public` 或 `Private` 关键字, 则 `Sub` 过程默认为公用的, 即它们对于脚本中的所有其他过程都是有效的。Sub 过程中局部变量的值在调用过程中不被保留。

不能在任何其他过程中定义 `Sub` 过程, 例如, 不能在 `Function` 函数中或 `Property Get` 属性中定义过程。

用户可以使用 `Exit Sub` 语句强制程序从 `Sub` 过程中退出, 然后继续执行调用 `Sub` 过程的语句之后的语句。可以在 `Sub` 过程中任意位置使用任意多个 `Exit Sub` 语句。

与 `Function` 过程相似之处是: `Sub` 过程是一个可以获取参数、执行一系列语句以及可改变其参数的值的独立过程。

与 `Function` 过程不同之处是: `Function` 过程可以返回值, 而 `Sub` 过程不能用于表达式中。



提示

Sub 过程是可以递归的, 即该过程可以调用自己来完成某个给定的任务。但是递归可能会导致堆栈溢出, 使用时一定要结合 `Exit Sub` 语句强制终止 `Sub` 过程无限递归。

例如:

```
<script type="text/vbscript">
Sub mysub(a)
    a = a + 1
    if a > 4 then
        exit sub
    end if
    MsgBox a
    mysub a
End Sub
mysub a
</script>
```

在上面示例中，通过定义一个值传递参数来监控 Sub 过程的递归，如果值传递参数大于 4，即当递归超过 4 次后，Exit Sub 语句就会强制 Sub 过程结束。

在 Sub 过程中使用的变量可以分为如下两类。

- 在过程内显式声明的变量。一般使用 Dim 语句或其他等效方法，在过程内显式声明的变量总是局部变量。
- 在过程内没有显式声明的变量。对于没有在过程中显式声明的变量，VBScript 会默认为局部变量，除非在该过程外更高级别的位置显式地声明它们。

过程可以使用没有在过程内显式声明的变量，但只要有任何脚本级定义的名称与之同名，就会产生名称冲突。如果过程中引用的未声明的变量与其他的过程、常数或变量的名称相同，则会认为过程引用的是脚本级的变量。例如，把上面示例中值传递参数改为脚本级变量来监控 Sub 过程的递归：

```
<script type="text/vbscript">
Dim a
a = 1
Sub mysub
    a = a + 1
    if a > 4 then
        exit sub
    end if
    MsgBox a
    mysub
End Sub
mysub
</script>
```

为了避免局部变量与脚本变量名称发生冲突，用户可以使用 Option Explicit 语句可强制显式声明变量。

4.10.3 调用 Sub 过程

调用 Sub 过程的方法有如下两种。

- 使用 Call 语句。
- 直接引用 Sub 过程名。

调用 Sub 过程时，调用语句必须是一个独立的语句。使用 Call 语句调用 Sub 过程的语法如下：

```
Call Sub进程名([参数列表])
```

使用 Call 语句时，必须将所有参数包含在括号之中。

直接使用 Sub 过程名的语法如下：

```
Sub进程名 [参数列表]
```

注意，直接调用 Sub 过程时，Sub 过程名与后面的参数列表之间要用空格隔开，不能使用小括号。例如，下面调用方法中，Sub 过程将不被 VBScript 执行，显示结果依然为 1，也就是说 Sub 过程 mysub 没有被执行。

```
<script type="text/vbscript">
Sub mysub(byref a)
    a = 2
End Sub
a = 1
```

```
mysub(a)
MsgBox a
</script>
```

4.11 VBScript 函数

Function 过程是包含在 Function 和 End Function 语句之间的一组 VBScript 语句。Function 过程与 Sub 过程类似，但是 Function 过程可以返回值。Function 过程可以使用参数（由调用过程传递的常数、变量或表达式）。如果 Function 过程无任何参数，则 Function 语句必须包含空括号()。

4.11.1 Function 过程基本用法

声明 Function 过程需要指定函数的名称、参数以及构成其主体的代码。其语法格式如下：

```
[Public [Default] | Private] Function name [(参数列表)]
    [语句]
    [name = expression]
    [Exit Function]
    [语句]
    [name = expression]
End Function
```

其中 Public 关键字表示 Function 过程可被所有脚本中的所有其他过程访问，Default 关键字只与 Class 中的 Public 关键字一起使用，表示 Function 过程是类的默认方法。在一个类中只能指定一个 Default 过程，Private 关键字表示 Function 过程只可被声明它的脚本中的其他过程访问，或者如果函数是一个数据类，那么 Function 过程只能被该类中的其他过程访问；name 表示 Function 的名称；expression 表示 Function 的返回值。

例如：

```
<script type="text/vbscript">
public function toCharArray(ByVal str)
    redim charArray(len(str))
    for i = 1 to len(str)
        charArray(i-1) = Mid(str,i,1)
    next
    toCharArray = charArray
end function
dim a
a = toCharArray("abcdefg")
Msgbox a(2)
</script>
```

在上面示例中，先定义一个公共函数 toCharArray()，该函数能够把传递给它的字符串转换为一个数组，字符串中的每个字符为数组的一个元素。最后调用该函数，然后显示字符串 "abcdefg" 的第 3 个字符。

Function 过程的参数类型也包括引用传递和值传递，具体使用可以参见上节 Sub 过程的用法。

4.11.2 Function 过程使用技巧

如果没有显式指定 Public 或者 Private 关键字，则 Function 过程默为公用，即它们对于脚

本中的所有其他过程是可见的。Function 中局部变量的值在对过程的调用中不被保留。

不能在任何其他过程中定义 Function 过程。例如，Sub 过程或者 Property Get 属性。

使用 Exit Function 语句可以从 Function 过程中强制退出。程序继续执行调用 Function 过程的语句之后的语句。可在 Function 过程的任何位置使用任意多个 Exit Function 语句。

与 Sub 过程类似，Function 过程是可以获取参数的。但与 Sub 过程的不同之处是，当要使用由函数返回的值时，可以在表达式的右边使用 Function 过程，这与内部函数的使用方式一样。

在表达式中，可以通过使用函数名，并在其后用圆括号给出相应的参数列表来调用 Function 过程。

Function 过程可以递归，即该过程可以调用自身以完成某个给定的任务，但是递归可能会导致堆栈溢出，使用时要结合 Exit Function 语句避免此类问题的发生。

在 Function 过程中使用的变量分为两类：一类是在过程内显式声明的，另一类则不是显式声明的。在过程内显式声明的变量总是过程的局部变量。被使用但没有在过程中显式声明的变量也是局部变量，除非在该过程外更高级别的位置显式声明它们。



提示

VBScript 可能会重新排列数学表达式以提高内部效率。当 Function 过程修改数学表达式中变量的值时，应避免在同一表达式中使用该函数。

4.11.3 Function 过程的返回值

Function 过程与 Sub 过程最大的不同就是函数能够返回值。要从函数返回一个值，只需将值赋给函数名。在过程的任意位置都可以出现任意多个这样的赋值。例如：

```
<script type="text/vbscript">
Public Function HtmlEncode(str)
    If Trim(Str)=" " Or IsNull(str) then
        HtmlEncode=" "
    else
        str=Replace(str,">","&gt;")
        str=Replace(str,"<","&lt;")
        str=Replace(str,Chr(32),"&nbsp;")
        str=Replace(str,Chr(9),"&nbsp;")
        str=Replace(str,Chr(34),"&quot;")
        str=Replace(str,Chr(39),"&#39;")
        str=Replace(str,Chr(13),"")
        str=Replace(str,Chr(10) & Chr(10), "</p><p>")
        str=Replace(str,Chr(10),"<br> ")
        HtmlEncode=str
    end if
End Function
Dim a
a ="<meta http-equiv='Content-Type' content='text/html; charset=gb2312' />"
a = HtmlEncode(a)
document.write a
</script>
```

在上面示例中，通过自定义函数将字符串中的 HTML 代码进行过滤，以便能在网页中显示 HTML 源代码。其中 Trim()（清除字符串左右空格）、Replace()（替换指定字符串中特定字符）和 Chr()（将指定 ANSI 字符编码转换为相对应的字符）都是 VBScript 都是内置函数。

如果没有给函数名赋值，则 Function 过程将返回一个默认值：数值函数返回 0；字符串

函数返回零长度字符串("")。

如果在 Function 中没有对象引用被指定给函数名(使用 Set), 则返回对象引用的函数将返回 Nothing。

VBScript 还为用户提供了大量的内置函数, 这些函数在网页编程中非常实用。熟悉并灵活掌握这些函数的使用, 能够提高动态网页编程的速度和质量。详细说明可以参考光盘电子书稿。

4.12 VBScript 内置函数

VBScript 还为用户提供了大量的内置函数, 这些函数在网页编程中非常实用。熟悉并灵活掌握这些函数的使用, 能够提高动态网页编程的速度和质量。

4.12.1 数学函数

VBScript 提供了众多数学函数和派生的数学函数, 这些函数在计算中都比较有用。常用数学函数见表 4.9。

表 4.9 常用数学函数

函 数	功 能
Abs	返回数字的绝对值
Atn	返回数值的反正切值
Cos	返回某个角的余弦值
Exp	返回 e (自然对数的底) 的幂次方
Fix	返回数字的整数部分 (如果数字为负, 返回大于或等于数字的第一个负整数), 如 $\text{Fix}(-5.4) = -5$
Int	返回数字的整数部分 (如果数字为负, 返回小于或等于数字的第一个负整数), 如 $\text{Int}(-5.4) = -6$
Log	返回数值的自然对数
Rnd	返回一个随机数
Sgn	返回表示数字符号的整数, 一般用来判断数值的正负, 如果返回 1, 则表示参数为正; 如果返回 -1, 则表示参数为负; 如果返回 0, 则表示参数为 0
Sin	返回某个角的正弦值
Sqr	返回数值的平方根
Tan	返回某个角的正切值

下面以 Rnd 函数介绍数学函数的使用。Rnd 函数的语法格式如下:

```
Rnd[ (number) ]
```

Rnd 函数能够返回一个小于 1, 但大于或等于 0 的值。其中 number 参数可以是任意有效的数值表达式, 该参数决定了 Rnd 生成随机数的方式:

如果参数小于 0, 则 Rnd 将使用 number 作为种子, 每次都生成相同的值;

如果参数大于 0, 则 Rnd 将生成序列中的下一个随机数;

如果参数等于 0, 则 Rnd 将生成最近生成的数。

Rnd 默认方式为序列中的下一个随机数。由于每一次连续调用 Rnd 函数时都用序列中的

前一个数作为下一个数的种子，所以对于任何最初给定的种子都会生成相同的数列。



在调用 Rnd 之前，先使用无参数的 Randomize 语句初始化随机数生成器，该生成器具有基于系统计时器的种子。

例如：

```
<script type="text/vbscript">
lowerbound = 1
upperbound = 10
Randomize
a = Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
document.write a
</script>
```

上面示例将产生一个指定范围的随机整数，其中最小值为 1，最大值为 10。用户可以通过 lowerbound 和 upperbound 变量指定最小值和最大值。

4.12.2 字符串函数

字符串函数在 VBScript 中最常用的一组函数，在脚本中常需要对一些字符串进行一些技术处理。例如，过滤掉字符串中的敏感字符以符合最终显示的要求、计算字符串的长度、查找字符串、替换字符串等。如果使用字符串函数，这些问题就会迎刃而解。常用字符串函数见表 4.10。

表 4.10 常用字符串函数

函 数	功 能
Split	返回基于 0 的一维数组，其中包含指定数目的子字符串。格式：Split(expression[, delimiter[, count[, start]]])。其中 expression 表示字符串表达式；delimiter 表示分界字符，默认为空格；count 表示返回子字符串的个数；Compare 表示在计算子字符串时使用的比较类型的数值，取值 0 则执行二进制比较，取值 1 则执行文本比较，默认为 0
Replace	返回字符串，其中指定数目的某子字符串被替换为另一个子字符串。格式：Replace(expression, find, replacewith[, compare[, count[, start]]])，其中 expression 表示要替代的子字符串；Find 表示被搜索的子字符串；Replacewith 表示用于替换的子字符串；Start 表示 expression 中开始搜索子字符串的位置，默认为 1；count 表示执行子字符串替换的数目，默认为所有；Compare 表示在计算子字符串时使用的比较类型的数值，取值 0 则执行二进制比较，取值 1 则执行文本比较，默认为 0
Mid	从字符串中返回指定数目的字符。格式：Mid(string, start[, length])，其中 string 表示字符串表达式；Start 表示从 string 中被提取的字符部分的开始位置；Length 表示返回的字符数，如果省略或 length 超过文本的字符数（包括 start 处的字符），将返回字符串中从 start 到字符串结束的所有字符
LTrim	返回不带前导空格的字符串副本。格式：LTrim(string)
RTrim	返回不带后续空格的字符串副本。格式：RTrim(string)
Trim	返回不带前导和后续空格的字符串副本。格式：Trim(string)
InStr	返回某字符串在另一字符串中第一次出现的位置。格式：InStr([start,]string1, string2[, compare])，其中 start 表示开始搜索的位置，默认为 1；string1 表示接受搜索的字符串表达式；string2 表示要搜索的字符串表达式；compare 表示在计算子字符串时使用的比较类型的数值，可参考 Split 函数说明
String	返回具有指定长度的、重复字符组成的字符串。格式：String(number, character)，其中 number 表示返回字符串的长度；character 表示指定字符或字符串表达式的字符代码
Space	返回由指定数目的空格组成的字符串。格式：Space(number)，其中 number 表示返回空格数
Left	返回指定数目的从字符串的左边算起的字符。格式：Left(string, length)，其中 string 表示字符串表达式；Length 表示返回字符串长度，如果是 0，返回零长度字符串 ("")；如果大于或等于 string 参数中的字符总数，则返回整个字符串

续表

函 数	功 能
Right	返回指定数目的从字符串的右边算起的字符。格式: Right(string, length), 其中 string 表示字符串表达式; Length 表示返回字符长度, 如果是 0, 返回零长度字符串 (""); 如果大于或等于 string 参数中的字符总数, 则返回整个字符串
Len	返回字符串内字符的数目, 或是存储一变量所需的字节数。格式: Len(string varname), 其中 string 表示任意有效的字符串表达式; varname 表示任意有效的变量名
Asc	返回与字符串的第一个字母对应的 ANSI 字符代码。格式: Asc(string)
Chr	返回与指定的 ANSI 字符代码相对应的字符。格式: Chr(charcode), 与 Asc 函数是一个相反操作函数, 参数是可以标识字符的数字。从 0~31 的数字表示标准的不可打印的 ASCII 代码
LCase	返回字符串的小写形式。格式: LCase(string)
UCase	返回字符串的大写形式。格式: UCase(string)
Join	返回一个字符串, 此字符串由包含在数组中的许多子字符串连接创建。格式: Join(list[, delimiter]), 其中 list 表示包含要联接的子字符串一维数组; Delimiter 表示在返回字符串中用于分隔子字符串的字符。如果省略, 将使用空字符 (""). 如果 delimiter 是零长度字符串, 则在同一列表中列出全部项, 没有分界符
StrComp	返回一个表明字符串比较结果的值。格式: StrComp(string1, string2[, compare]), 其中 string1 和 string2 表示任意字符串表达式; Compare 表示在计算字符串时使用的比较类型的数值, 可参见 Replace 函数说明
StrReverse	返回字符串, 此字符串与指定字符串顺序相反。格式: StrReverse(string1)

下面将运用字符串函数实现字符串的省略显示:

```
<script type="text/vbscript">
'' 计算源字符串Str的长度(一个中文字符为2个字节长)
Public Function StrLen(Str)
    If Trim(Str)="" Or IsNull(str) Then
        strlen=0
    else
        Dim x
        StrLen=0
        For x=1 To Len(Trim(Str))
            If Asc(Mid(Str,x,1))<0 Then
                StrLen=Int(StrLen) + 2
            Else
                StrLen=Int(StrLen) + 1
            End If
        Next
    end if
End Function
'' 截取源字符串Str的前LenNum个字符(一个中文字符为2个字节长)
Public Function CutStr(Str,LenNum)
    Dim P_num,I,X
    If StrLen(Str)<=LenNum Then
        Cutstr=Str
    Else
        P_num=0
        X=0
        Do While Not P_num > LenNum-2
            X=X+1
            If Asc(Mid(Str,X,1))<0 Then
                P_num=Int(P_num) + 2
            Else
                P_num=Int(P_num) + 1
            End If
            Cutstr=Left(Trim(Str),X)&"..."
        Loop
    End If
End Function
```

```
End Function
document.write CutStr("abcdefg",5)
</script>
```

在上面示例中，定义了两个函数：函数 `strLen` 能够精确计算字符串的长度，以单字节作为计算的单位，中文作为双字节来计算；函数 `CutStr` 能够根据指定长度，来截取并省略显示超过指定长度的字符串。在上面代码中，主要技巧涉及对字符进行判断，以确定每个字符是单字节还是双字节。

4.12.3 转换函数

转换函数也是很重要的一组函数，常用来转换子数据类型，以实现数据的特殊需求，例如把字符串类型的数字转换为整数，以方便进行数值比较等。常用转换函数见表 4.11。

表 4.11 常用转换函数

函数	功 能
CBool	将表达式转换为 Boolean 子类型的 Variant。格式：CBool(expression)，其中 expression 表示任意有效的表达式
Cbyte	将表达式转换为 Byte 子类型的 Variant，用法同 CBool 函数
Ccur	将表达式转换为 Currency 子类型的 Variant，用法同 CBool 函数
Cdate	将表达式转换为 Date 子类型的 Variant，用法同 CBool 函数
Cdbl	将表达式转换为 Double 子类型的 Variant，用法同 CBool 函数
Cint	将表达式转换为 Integer 子类型的 Variant，用法同 CBool 函数
Clng	将表达式转换为 Long 子类型的 Variant，用法同 CBool 函数
Csng	将表达式转换为 Single 子类型的 Variant，用法同 CBool 函数
Cstr	将表达式转换为 String 子类型的 Variant，用法同 CBool 函数
VarType	返回指定变量子类型的值。格式：VarType(varname)，VarType 函数返回值说明见表 4.12。利用该函数可以准确确定一个变量的数据类型，然后就可以根据数据类型执行相应操作

表 4.12 VarType 函数返回值列表

常数值	返回值	变量类型
vbEmpty	0	Empty（未初始化）
vbNull	1	Null（无有效数据）
vbInteger	2	整数
vbLong	3	长整数
vbSingle	4	单精度浮点数
vbDouble	5	双精度浮点数
vbCurrency	6	货币
vbDate	7	日期
vbString	8	字符串
vbObject	9	Automation 对象
vbError	10	错误
vbBoolean	11	Boolean
vbVariant	12	Variant（只和变量数组一起使用）
vbDataObject	13	数据访问对象
vbByte	17	字节
vbArray	8192	数组

4.12.4 日期/时间函数

如果要在脚本中设计与时间有关的操作，必须使用各种日期/时间函数。常用日期/时间函数见表 4.13。

表 4.13 常用日期/时间函数

函 数	功 能
Now	根据本地计算机系统设定的日期和时间，返回当前的日期和时间值
Date	返回当前计算机系统设定的日期值
Time	返回当前计算机系统设定的时间值
Year	返回一个代表某年的整数值。格式：Year(date)，其中 date 表示任意可以代表日期的参数，格式：year(date())，还可以直接指定具体的日期，如 year(#8/8/2008#)
Month	返回 1~12 之间的一个整数（包括 1 和 12），代表一年中的某月。格式：Month(Now)
Day	返回 1~31 之间的整数值，表示一个月中的某天。格式：day(date)，关于参数 date 的说明和 year 函数相同。同样要注意日期的正确性
Hour	返回 0~23 之间的整数值，表示一天中的某个小时。格式：hour(time)，其中参数 time 是任意的可代表时间的表达式，如 hour(#11:45:50#)
Minute	返回 0~59 之间的整数值，表示一小时中的某分钟。格式：minute(time)，其中 time 参数的说明和 hour 函数相同
Second	返回 0~59 之间的整数值，表示一分钟中的某秒。格式：second(time)，其中 time 参数的说明和 hour 函数相同
Weekday	返回星期中某天的整数。格式：weekday(date)，其中参数 date 与 year 函数相同。该函数返回值为 1~7 之间的数字，分别代表“星期日”、“星期一”、…、“星期六”，如返回值是 4 就表示是“星期三”
WeekDayName	返回星期中具体某天的字符串。格式：weekdayname(weekday)，其中参数 weekday 表示星期中具体某天的数值，如 weekdayname(weekday(date()))就表示当前是星期几，当然 weekdayname 函数最终显示的字符串还与当前操作系统语系有关，比如中文操作系统将显示“星期一”这类的中文字符，而英文操作系统则显示为“Mon”（Monday 简写）
DateAdd	返回已添加指定时间间隔的日期。格式：DateAdd(interval, number, date)，其中 interval 表示需要添加的时间间隔单位，以字符串的形式表示，如“yyyy”表示年，“q”表示季度，“m”表示月份，“d”表示天数，“ww”表示周数，“h”表示小时数，“n”表示分钟数，“s”表示秒数；number 表示添加的时间间隔数，以数值的形式表示，可以为负值；date 表示要添加 interval 的日期文字。例如，dateadd("d",100,"2007-9-9")就表示 2007 年 9 月 9 号以后 100 天的日期值，返回结果为“2007-12-18”
DateDiff	返回两个日期时间之间的间隔。格式：DateDiff(interval, date1, date2 [,firstdayofweek[,firstweekofyear]])，其中 interval 与 dateadd 函数参数相同；date1 和 date2 参数分别就是相互比较的两个日期时间；Firstdayofweek 表示指定星期中第一天的常数，如果没有指定，则默认为星期日；Firstweekofyear 表示指定一年中第一周的常数。如果没有指定，则默认为 1 月 1 日所在的星期

4.12.5 判断函数

判断函数与数据类型转换函数存在很多联系。结合使用这两类函数，用户可以编写出功能灵活的脚本。判断函数能够判断一个变量的数据类型，一般都返回一个逻辑值，如果返回 True，则表示变量是特定类型的数据，否则不是。常用判断函数见表 4.14。

表 4.14 常用判断函数

函 数	功 能
IsArray	判断是否为数组
IsDate	判断是否为日期
IsEmpty	判断变量是否已初始化，即已经赋值

续表

函 数	功 能
IsNull	判断表达式是否不包含任何有效数据。在某些情况下要想使表达式取值为 True, 例如, Var=Null 和 Var<>Null, 但它们通常总是为 False。这是因为任何包含 Null 的表达式本身就为 Null, 所以表达式的结果为 False
IsNumeric	判断表达式的值是否为数字
IsObject	判断表达式是否引用了有效的 Automation 对象

4.12.6 数组函数

VBScript 提供了 3 个比较实用的数组函数, 见表 4.15。

表 4.15 常用数组函数

函 数	功 能
LBound	返回指定数组维的最小可用下标。格式: LBound(arrayname[, dimension]), 其中 arrayname 表示数组变量; dimension 表示返回哪一维下界的整数, 使用 1 表示第一维, 2 表示第二维, 以此类推
UBound	返回指定数组维的最大可用下标。格式: UBound(arrayname[, dimension]), 其中 arrayname 表示数组变量; dimension 表示返回哪一维下界的整数, 使用 1 表示第一维, 2 表示第二维, 以此类推
Filter	返回下标从零开始的数组, 其中包含以特定过滤条件为基础的字符串数组的子集。格式: Filter(InputStrings, Value[, Include[, Compare]]), 其中 InputStrings 表示一维数组, 要在其中搜索字符串; Value 表示要搜索的字符串; Include 表示指定返回的子字符串是否包含 Value, 如果为 True, Filter 将返回包含子字符串 Value 的数组子集, 如果为 False, Filter 将返回不包含子字符串 Value 的数组子集; Compare 指出使用的比较字符串类型, 取值 0 则执行二进制比较, 取值 1 则执行文本比较, 默认为 0

4.13 VBScript 类

为了满足用户日益复杂的开发需求, 从 VBScript 5.0 版本开始支持类功能, 实现完整的 VB 类(class)模型。但 VBScript 类仅是一个简单的类模型, 不如 .Net、C++、Java 等语言那样完全以类为核心进行开发。因此, 学习和使用 VBScript 自定义类就相对简单了许多。



提示

VBScript 提供的 Scripting 运行时库实际上就是一个类库, 与 .Net、Java 所提供的类库相似, 但功能和规模都比较弱小。

4.14.1 认识类

使用任何一门面向对象的编程语言, 工程师自然会想到如何能把代码模块化, 模块化也是动态网站开发中常常面对的问题。VBScript 在本质上不是面向对象的语言, 但用户可以使用 VBScript 提供的类来实现代码的封装, 从而实现模块化开发的目的。

类和对象是面向对象开发中两个很重要的概念。类, 简单说就是对象的模板, 它定义了对象的特性; 而对象实际上就是类的实例化, 也是用户可以与之交互的内容。

对象包含属性、方法和事件。

- 属性: 作为一个名词, 描述对象的特征。
- 方法: 作为一个动词, 描述对象可以完成的工作, 或者希望它完成的工作。
- 事件: 描述了对象为响应某个动作而执行的操作。

类具有抽象、封装、多态和继承的特征。

- 抽象：用户可以不必要知道类的内部工作机制和运作方式，这正如看电视，用户不必要知道电视是如何工作的，只需打开电视机，搜索频道即可，开关按钮简化了实际的操作。
- 封装：在类中每个对象都包含完成操作所需要的所有信息，这个对象称为封装，因此对象不会依赖其他对象或者外部辅助来完成自己的操作。
- 多态：它表示不同的对象可以执行相同的动作，但要通过自己的实现代码来执行，虽然对象的名称一样，但底层实现的代码是不一样的。
- 继承：它定义了类如何相互关联、共享特性的。继承的工作方式是定义类和子类。其中子类继承了父类的所有特性。继承的重要性是可以扩展类的功能，实现不同类的一致性，并允许共享代码。如果决定创建一个新类，就不必定义父类的所有特性。

4.14.2 创建类

类的定义与函数和 Sub 过程非常类似。以 Class 语句开始，以 End Class 结束，所有的对象都被包含在其中，例如，类的变量、属性和方法。具体格式如下：

```
Class name
    包含对象、方法、属性
End Class
```

在 Class 中，成员对象通过相应的声明语句被声明为 Private 或者 Public。被声明为 Private 的成员将只在 Class 内有效，被声明为 Public 的成员不仅在 Class 的内部有效，而且在 Class 外的代码中也有效。

如果没有使用 Private 或者 Public 明确声明的成员，默认为 Public。在类的内部被声明为 Public 的过程（包括 Sub 和 Function）将作为类的方法，Public 变量将成为类的属性，同使用 Property Get、Property Let 和 Property Set 显式声明的属性一样。

类的缺省属性和方法是在它们的声明部分用 Default 关键字指定。

例如：

```
<%@Language="VBScript" Codepage="936" %>
<%
'声明一个名叫asp_class的类
Class asp_class
    '声明一个类的私有变量
    Private asp
    '初始化类（定义类的构造函数）
    Private Sub Class_Initialize
        asp="欢迎使用ASP类！<br>"
    End Sub
    '定义一个类的方法（有返回值）
    Public Function out_info()
        out_info=asp & "我是asp_class类的一个函数。<br>"
    End Function
    '定义一个类的方法（无返回值）
    Public Sub WriteStr(str)
        Response.Write(str)
    End Sub
End Class
%>
```

4.14.3 实例化类

创建类之后用户就可以实例化类，并在代码中进行引用。实例化类一般使用 Set 语句，

语法格式如下：

```
Set objectvar = {objectexpression | New classname | Nothing}
```

其中 objectvar 表示一个变量，objectexpression 表示一个对象。New 关键字用来创建一个类的新实例，但只限于创建一个 class 的实例；classname 表示创建的类名；Nothing 表示断开 objectvar 与任何指定对象或类的关联，一般当不再使用前面的引用对象时，应该为 objectvar 赋值为 Nothing，以释放与前面的引用对象相关联的系统或内存资源。

由于 VBScript 是一种后期约束型语言，下面的做法是不允许的：

```
Dim objectvar as New classname
```

或者

```
Dim objectvar  
objectvar = New classname
```

或者

```
Set objectvar = New Scripting.FileSystemObject
```

例如，可以使用下面代码对上节定义的类进行引用：

```
<%  
Dim myclass  
'定义一个名为myclass的asp_class对象的实例  
Set myclass=New asp_class  
'调用对象的方法  
Response.Write(myclass.out_info)  
'调用对象的方法  
Dim str  
str="我是asp_class类的一个方法。<br><span style='color:red;'>类是ASP应用开发中最高级别技术，  
快吃掉它>>>--==^</span>"  
myclass.WriteStr str  
'关闭对象，释放系统资源  
Set myclass = nothing  
%>
```

4.14.4 定义类的属性

属性和方法都是类的两个重要成员。用户可以通过为类定义公共变量来接收或向外部传递信息，但这种方式不是很安全，破坏了类的封闭性。为了保证类与外部的通信，例如，接收外部传递的值或者把内部值传递出去，一般可以使用 Property 语句来实现，但 Property 必须与 Let、Set 或 Get 语句配合使用。

- Let: 可以设置属性值。
- Get: 可以读取属性值。
- Set: 可以设置对象。

1. Property Let 语句

Property Let 语句可以在 Class 中设置属性的值。语法格式如下：

```
[Public | Private] Property Let name ([arglist,] value)  
[语句]  
[Exit Property]  
[语句]  
End Property
```

其中 Public 关键字表示 Property Let 过程可以被所有脚本中的其他所有过程访问；Private

关键字表示 Property Let 过程只能被定义的 Class 内的其他过程访问；name 表示 Property Let 过程的名称，在 Class 中的 Property Get 和 Property Set 过程名称要相同；arglist 是一个变量列表，它表示在调用时被传递到 Property Let 过程的参数，多个参数之间用逗号隔开，Property Let 过程的每个参数的名字必须与 Property Get 过程中的相应参数相同，而且 Property Let 过程的参数比相应的 Property Get 过程总要多一个，该参数为被赋予属性的值；value 表示要赋予属性的值。

每个 Property Let 语句必须为所定义的属性定义至少一个参数，该参数包含了当 Property Let 语句被调用时要赋予属性的值。

如果未明确地使用 Public 或 Private 进行指定，Property Let 过程被缺省设置为公有的，即它们对于脚本内的其他所有过程都是可见的。Property Let 过程中的局部变量的值在不同的过程调用之间是不被保存的。

在其他任何过程（如 Function 或 Property Get）的内部不能够定义 Property Let 过程。

Exit Property 语句可以强制从 Property Let 过程中退出。程序将从调用 Property Let 过程的语句之后的点继续执行。Exit Property 语句可以出现在 Property Let 过程中的任何位置，次数不限。

与 Function 和 Property Get 过程类似，Property Let 过程是一个单独的过程，它可以接受参数，执行一系列的语句，还可以改变参数的值。不过，与 Function 和 Property Get 过程不同的是，它们两者都返回一个值，而 Property Let 过程只能用于属性赋值表达式的左侧。

2. Property Get 语句

在 Class 中，Property Get 语句可以取得（返回）属性的值。其语法格式如下：

```
[Public [Default]] Private] Property Get name [(arglist)]
[语句]
[[Set] name = expression]
[Exit Property]
[语句]
[[Set] name = expression]
End Property
```

具体参数说明可以参见 Property Let 语句。下面看一个实例：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Class myClass
    '声明myClass类的内部变量
    Private strExample
    '定义类的输入属性，该属性是让用户初始化strExapmle变量
    Public Property Let setExapmle(ByVal strVar)
        strExample = strVar
    End Property
    '定义类的输出属性，将返回用户定义的属性
    Public Property Get getExapmle
        getExapmle = strExample
    End Property
End Class
%>
<%
'实例化该类
Dim oneNewClass
Set oneNewClass = New myClass
oneNewClass.setExapmle = "这是一个简单类的例子"
Response.Write oneNewClass.getExapmle
```

```
Set oneNewClass = Nothing
%>
```

4.14.5 定义类的方法

要定义类的方法，只需要把过程包含在类中就可以了，其中 **Function** 过程可以定义有返回值的方法，而 **Sub** 过程可以定义没有返回值的方法。过程的使用遵循前面介绍的所有语法规则。例如：

```
<%@Language="VBScript" Codepage="936" %>
<%
Class TV
    private channel '定义私有变量channel(电视频道)
    public function getChannel()
        getChannel=channel
    end function
    sub setChannel(c)
        channel=c
    end sub
End Class
'实例化类
set mytv=new TV
'设置类的方法(无返回值)
'实际上这与Property Let语句功能相似，但不支持使用
mytv.setChannel("频道5")
'调用类的方法(有返回值)
Response.write mytv.getChannel()
%>
```

4.14.6 定义类的事件

VBScript 的类事件主要包括初始化事件和结束事件。所谓类的初始化就是定义只要一开始使用该类，就首先会触发并执行初始化部分代码。一般使用 **Class_Initialize** 语句对类进行初始化，而使用 **Class_Terminate** 语句定义类的结束事件，结束事件就是只要一退出该类，就会触发该事件。例如：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Class myClass
    '声明myClass类的私有变量
    Private strAuthor
    Private strVersion
    '定义类的初始化事件，以实现初始化类
    '在下面代码中首先初始化该类的私有变量，以及在屏幕上提示该类已经开始了
    Private Sub Class_Initialize()
        strAuthor = "CSS8"
        strVersion = "1.0"
        Response.Write "<h2>myClass类开始了</h2>"
    End Sub
    '定义类的结束事件，只要一退出该类，就会触发该事件
    '下面代码提示退出该类，显示该类已结束
    Private Sub Class_Terminate()
        Response.Write "<h2>myClass结束了</h2>"
    End Sub
End Class
%>
<%
'实例化类
```

```
Dim oneNewClass
Set oneNewClass = New myClass
Response.Write "类处于引用状态……"
Set oneNewClass = Nothing
%>
```

4.14 VBScript 编码规范

任何一种语言都有一套规范，VBScript 虽然是一种脚本语言，但如何遵循编码规则仍然是用户要特别注意的问题。

4.14.1 代码注释方法

注释是非执行语句，它不会被系统解释和编译，也不在程序执行结果中显示出来，可通过在编辑器中打开程序的源代码查看程序中的注释，以方便开发人员阅读代码。

注释语句有两种表达形式，其语法为：

```
Rem 注释内容
'注释内容
```



提示

注释内容是什么文本，注释可以和语句在同一行并写在语句的后面，也可以单独占一行。



提示

编程的第一个好习惯就是给代码增加注释，以说明程序中用到的常量、变量或某段代码的用途。这样方便对程序的阅读，也方便对已编写好的程序进行修改。不论是程序设计者本人还是其他的程序员都可以根据注释了解程序的设计思路，方便程序的修改。

在编写程序代码时，有的语句可能很长，为了阅读和程序查错的直观、方便，可使用续行符“-”（由一个空格和一个下划线组成），将长语句分成多行书写。

4.14.2 注释规范

在开发脚本时，用户应该在所有过程的开始部分添加描述其功能的简要注释。这些注释并不描述细节信息（如何实现功能），这是因为细节有时要频繁更改。这样就可以避免不必要的注释维护工作以及错误的注释。细节信息由代码本身及必要的内部注释来描述。例如：

```
<%@Language="VBScript" Codepage="936" %>
<%
'*****
' @功能说明：将源字符串str中每个单词的第一个字母转换成大写
' @参数说明：- str [string]：源字符串，
' @返回值：- [string] 转换后的字符串
'*****
public function capitalize(str)
    words = split(str, " ")
    for i = 0 to ubound(words)
        if not i = 0 then
            tmp = " "
        end if
        tmp = tmp & ucase(left(words(i), 1)) & right(words(i), len(words(i))-1)
        words(i) = tmp
    next i
end function
```

```

        next
        capitalize = arrayToString(words)
    end function
%>

```

当传递给过程的参数的用途不明显，或过程对参数的取值范围有要求时，应加以说明。如果过程改变了函数和变量的返回值（特别是通过参数引用来改变），也应在过程的开始部分描述该返回值。

过程开始部分的注释建议应包含以下描述。

- 目的：过程的功能（不是实现功能的方法）。
- 假设：其状态影响此过程的外部变量、控件或其他元素的列表。
- 效果：过程对每个外部变量、控件或其他元素的影响效果的列表。
- 输入：每个目的不明显的参数的解释。每个参数都应占据单独一行并有其内部注释。
- 返回：返回值的解释。

另外，每个重要的变量声明都应有内部注释，描述变量的用途。

应清楚地命名变量、控件和过程，仅在说明复杂细节时需要内部注释。

4.14.3 常量命名规范

早期的 VBScript 版本不支持用户自定义常量。如果要使用常量，则通过变量的方式实现，且全部字母大写以便与其他变量区分，常量名中的多个单词用下划线（_）分隔。例如：

```
USER_LIST_MAX
```

这种标识常数的方法依旧可行，但用户还可以选择其他方法，如以“con”作为常量名的前缀。例如：

```
con UserListMax
```

4.14.4 变量命名规范

为提高易读性和一致性，用户应该在 VBScript 代码中使用表 4.16 中的变量命名规范。

表 4.16 变量命名规范

子类型	前 缀	示 例	子类型	前 缀	示 例
Boolean	bln	blnFound	Integer	int	intQuantity
Byte	byt	bytRasterData	Long	lng	lngDistance
Date (Time)	dtm	dtmStart	Object	obj	objCurrent
Double	dbl	dblTolerance	Single	sng	sngAverage
Error	err	errOrderNum	String	str	strFirstName

此外，变量名或过程名的主体应使用大小写混合格式，并且尽量完整地描述其目的。另外，过程名应以动词开始，例如，InitNameArray 或 CloseDialog。

对于经常使用的或较长的名称，推荐使用标准缩写以使名称保持在适当的长度内。通常多于 32 个字符的变量名会变得难以阅读。使用缩写时，应确保在整个脚本中保持一致。例如，在一个脚本中随意切换 Cnt 和 Count 就会引起混乱。

4.14.5 代码格式化

用户可以使用下面方式对代码格式进行排版。

标准嵌套块应缩进 4 个空格。

过程的概述注释应缩进 1 个空格。

概述注释后的最高层语句应缩进 4 个空格，每一层嵌套块再缩进 4 个空格。

例如：

```
<%@Language="VBScript" Codepage="936" %>
<%
'*****
' @功能说明：检查字符character是否是英文字符 A-Z or a-z
' @参数说明： - character [char]：检查的字符
' @返回值： - [bool] 如果是英文字符
'             返回TRUE,反之为FALSE
'*****
public function isAlphabetic(byVal character)
    asciiValue = cint(asc(character))
    if (65 <= asciiValue and asciiValue <= 90) or (97 <= asciiValue and asciiValue
    <= 122) then
        isAlphabetic = true
    else
        isAlphabetic = false
    end if
end function
%>
```

4.15 实践总结

本章详细介绍了 ASP 编程中默认的语言 VBScript，同时也介绍了 JavaScript 和 VBScript 语言的基本特性。完全掌握 VBScript 语言，例如可以自由使用变量、数组、函数、子程序和类，以及 VBScript 编码规范，能够编写各种条件、循环等逻辑结构，对于后面的动态网页开发至关重要。

虽然 VBScript 语法规则缺乏严谨，但对于初学者来说，能够快速掌握 VBScript 并应用于实践。希望读者认真吃透本章介绍的内容，为后面的实例学习奠定扎实的基础。

第 5 章

数据库访问、查询和操作

如果说网络是信息传输的媒介，Web 应用是信息发布的一种平台，那么数据库就应该是信息存储与管理的载体。要开发基于 B/S (Browser/Server, 浏览器/服务器) 模式的 Web 应用，首先要解决的问题就是要实现动态网页与数据库的连接。ASP 提供了强大的数据库支持功能，它主要通过 ADO 组件来完成所有与数据库的操作，例如，连接数据库、数据读取、数据添加、数据更新和数据删除等操作。本章将从数据库最基本知识讲解起，详细介绍 SQL 数据库查询语言的基本使用，同时讲解 ASP 为实现操作数据库而专门开发的 ADO 数据库操作组件。

技术要点

- 了解数据库相关知识
- 了解 SQL 语言
- 建立数据库连接
- 操作记录集

5.1 认识数据库

在第 2 章图 2.1 中，用户已经看到了服务器端后台数据库的位置和作用。那么什么是数据库？如何操作服务器端的数据库呢？

数据库是信息管理的常规方法，它可以处理各种各样的数据信息。数据库的优势就在于它可以将庞大而复杂的信息以有序的方式组织起来，便于修改和查询，以免管理人员手工处理这些枯燥的数据。数据库系统在算法、系统结构等方面采取了很多方法以提高对数据的处理能力，同时在数据的保护、存取控制、备份上附加了很多重要的功能。

数据库一般按照数据的组织和查询方式加以区分。早期的数据库常常是网状数据库或层次数据库。顾名思义，这些数据库是通过网状或树状的模型处理数据存储和记录数据之间的关系的。现在这种数据库在一些地方依然还在使用，但目前使用最多的都是基于关系代数的关系数据库管理系统 (RDBMS)。

在关系数据库管理系统中，数据按照表存放。一个数据库可以有多个数据表，每个表由行和列组成，每行是一个记录，每个记录的信息分为多段，一张表的每一行的段的组成都是相同的，而不同行中相同的列就称为字段，图 5.1 中就是一张 Access 数据表。RDBMS 中表的数据可以通过行和列查询，而且每个表之间的数据可以建立某种关系，这样可以从不同的

表中找到所要的信息。

联系人编号	公司编号	名字	姓氏	职称	头衔
1	4	Janet	Leverling	Janet	副代表
2	5	安德	玛	Andrew	销售代表
3	2	玛	玛	Margaret	采购运营
4	1	雷加	王	Nancy	技术联系人
5	3	Steven	Buchanan	Steve	
(自动编号)	0				

图 5.1 Access 数据表结构示意图

在关系数据库中使用的数据查询语言称为结构化查询语言 SQL (Structured Query Language), SQL 是数据库语言的标准。

例如, 微软公司的 SQL Server、IBM 公司的 DB2、Oracle 公司的 Oracle 等都是符合 SQL 标准的数据库管理系统, 但这些数据库价格都很昂贵。除此之外, 还有一些免费的数据库, 例如, MySQL 等。这些数据库除了遵循 SQL 标准外, 还加入了自己的特色。SQL 的标准在不断地变化, 因此就有数据库产品遵循哪个标准的问题。尽管这些不同版本标准相互差异不是很大, 而且有的实现了 SQL 标准的超集, 有的只实现了标准的一个子集, 在使用时还是要注意数据库之间相互兼容的问题。



提示

本书案例都以 Access 2000 及其以上版本数据库为基础进行介绍, 该软件捆绑在微软 Office 办公套件中, 安装 Office 2000 或以上版本软件时会默认安装到用户的操作系统中, 该软件使用方便, 非常符合个人用户学习和建站使用。

5.2 SQL 语言

SQL 语言是数据库的标准语言。在 ASP 中无论何时访问一个数据库, 都要使用 SQL 语言, 因此掌握好 SQL 语言对 ASP 动态网页制作是非常重要的。

SQL 最早是 IBM 的圣约瑟实验室为其关系数据库管理系统 System R 开发的一种查询语言, 它的前身是 SQUARE 语言。SQL 语言结构简洁, 功能强大, 简单易学, 所以自从 IBM 公司 1981 年推出以来, SQL 语言就得到了广泛的应用。如今无论是 Oracle、Sybase、Informix、SQL Server 等大型数据库管理系统, 还是像 Visual Foxpro、PowerBuilder 这些微机上常用的数据库开发系统, 都支持 SQL 语言作为查询语言。

SQL 语言包含以下 4 个部分。

- 数据查询语言 (DQL-Data Query Language), 包括 SELECT 等语句。
- 数据操纵语言 (DML-Data Manipulation Language), 包括 INSERT、UPDATE、DELETE 等语句。
- 数据定义语言 (DDL-Data Definition Language), 包括 CREATE、ALTER、DROP 等语句。
- 数据控制语言 (DCL-Data Control Language), 包括 COMMIT WORK、ROLLBACK WORK 等。



提示

本章示例将以 Access 示例数据库“NorthWind.mdb”（罗斯文数据库）为例进行介绍，主要围绕“产品”表的操作。为了完成本章学习，建议读者模仿第 2 章介绍的方法在本地计算机上建立一个动态站点，其中【服务器类型】为“ASP VBScript”，【访问】类型为“本地/网络”，然后就可以在该站点内对 SQL 语句进行验证。

5.2.1 SQL 语言规范

SQL 语言的代码都是以字符串的形式存在，需要时把这些字符串传递给后台数据库管理系统，由数据库管理系统执行 SQL 字符串并返回执行的结果。

SQL 语言本身不会自动执行，因此它没有严密的逻辑体系和语法结构。但在编写 SQL 字符串时也需要注意一些语言规范。

(1) SQL 语言不区分大小写，且不受所引用语言的影响。例如，如果在 JavaScript 语言中编写的 SQL 字符串（JavaScript 语言是严格区分大小写的），但在编写 SQL 字符串时，读者可以小写、大写或者大小写混合使用，都不会影响执行结果，且在 JavaScript 语言中编写的 SQL 字符串可以在 VBScript 语言中使用。

(2) 在 SQL 语法中，表名、字段名不需要加引号，不管是中文名称，还是英文名称。

(3) 但如果输入的字段值是字符串或备注等，则应该加单引号，如“地址='西直门大街 110 号'”。

(4) 如果要输入日期，使用 Access 数据库，需要在日期前后加上#号，例如，“date=#12/1/2007#”；使用 SQL Server 数据库，需要在日期前后加上单引号，例如，“date='12/1/2007'”。

(5) 在 SQL 字符串中，所有标点符号都必须是英文标点符号，即必须在英文状态下输入标点符号。



提示

由于不同数据库管理系统对 SQL 字符串解析机制不同，因此 SQL 字符串的写法也不完全相同，使用时应该参考所用数据库帮助文档。例如，插入数据操作，Access 必须使用 Insert Into，而 SQL Server 可以省略 Into。不过基本语法是相通的。

5.2.2 SQL 数据查询

在 SQL 语言中主要使用 SELECT 语句来实现数据查询。SELECT 语句构成了 SQL 语言的核心。它能够从数据库中检索行，并允许从一个或多个表中选择一个或多个行或列。虽然 SELECT 语句的完整语法比较复杂，但是它的主要子句语法格式如下：

```
SELECT select_list
[INTO new_table]
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_expression]
[HAVING search_condition]
[ORDER BY order_expression [ASC|DESC]]
```

整个 SELECT 语句能够根据 WHERE 子句的条件表达式，从 FROM 子句指定的表或视图中找出满足条件的记录，再根据 SELECT 子句中的指定列选出并形成结果记录集。

如果有 GROUP 子句，则将结果按指定的值进行分组，该属性列值相等的元素组被分为

一个组。如果 GROUP 子句带有 HAVING 短语,则只有满足指定条件的组才予以输出。如果有 ORDER 子句,则结果表还要按指定的值进行升序或降序排序。

下面是常用 SQL 查询语句(针对“产品”表进行操作),为了方便学习,读者可以在已建立的动态站点中新建一个文件,保存为 select.asp,然后输入下面代码,以便来验证自己编写的 SQL 语句:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<% '定义数据库连接
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '实例化Connection对象
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置数据驱动
conn.Open Server.MapPath("data/NorthWind.mdb") '数据库的相对路径
%>
<% '定义一个记录集
sql = "Select * from 产品" '在此处输入用户自己编写的SQL查询语句
Set rs = Server.CreateObject("ADODB.RecordSet") '实例化Recordset对象
rs.Open sql,conn,1,1 '打开记录集
%>
<% '显示记录集中所有记录
'显示记录集中标题
Dim fds
Set fds=rs.Fields '实例化记录集Recordset对象的数据集合Fields
Response.Write("<table border=1 width=90% align=center><tr>")
For i=0 To fds.Count-1 '循环显示记录集的标题
    Response.Write("<td class='thead'>"&fds.Item(i).Name"&"</td>")
Next
Response.Write("</tr><tr>")
'显示记录集中记录
rs.MoveFirst '把记录集的指针移到第1条记录
While Not rs.eof '循环读取记录集中每条记录
    For i=0 To fds.Count-1 '循环读取每条记录集中每个字段值
        Response.Write("<td class='tbody' ">"&fds.Item(i).Value"&"</td>")
    Next
    Response.Write("</tr><tr>")
    rs.MoveNext '把记录集指针移到下一条记录
Wend
Response.Write("</table>")
%>
<% '释放记录集
rs.close
set conn=nothing
%>
```

1. 查询指定表中所有数据

select * from 产品

说明: *号表示所有记录。上面语句表示返回产品表中所有数据。

2. 查询指定表中限定列中的数据

select 产品名称, 单价 from 产品

说明: 限定字段列表之间要用逗号隔开,表示返回这些字段中的数据。上面语句将把产品表中的产品名称和单价列的所有数据返回。读者还可以增加更多的列,只需要添加列标题即可,列标题之间用逗号隔开。

3. 查询指定表中前几条记录

Select top 10 产品名称, 单价 from 产品

说明: 查询前几条记录可以使用 TOP 关键字放在字段列表前面, 然后在 TOP 关键字后指定要返回多少条记录。上面语句将把产品表中前 10 条记录中的产品名称和单价列返回。

使用该语句可以实现限数查询, 或者排名查询, 但需要结合 WHERE 子句设置记录筛选的条件。

4. 查询派生字段

Select 产品名称, 单价, 订购量, 再订购量, (单价*(订购量+再订购量)) as 总订购资金 from 产品

说明: 派生字段就是利用表中现有字段经过计算产生一个新的字段, 派生字段常用来显示字段数据汇总。派生字段与表中原有字段可以混在一起使用。用关键字 AS 为派生字段起一个别名。注意, 派生字段仅是一个临时的字段, 数据库中不存在该字段, 但如果把查询数据回写入数据库时, 该字段也被写入。

上面语句将把订购量和在定购量字段值相加, 然后乘以单价字段值得到一个总订购资金的新字段。执行上面 SQL 查询字符串, 显示结果如图 5.2 所示。



公司名称	库存总量
百达	130
成记	86
大钰	62
德昌	98
德级	164
东海	10
福满多	23
涵合	58
宏仁	57
佳佳	136
佳佳乐	69
金美	20
菊花	207
康堡	74
康富食品	133

图 5.2 派生字段查询结果

5. 查询指定条件记录

Select * from 产品 where 再订购量=25

说明: 查询指定条件记录一般使用 WHERE 子句, 在 WHERE 后面加上要查询的条件。上面语句将查询再订购量等于 25 的所有产品记录。



注意

WHERE 子句可以包含任何逻辑运算, 只要返回值为 True 或 False 即可, SQL 能够支持的比较运算符和逻辑运算符见表 5.1、表 5.2。

表 5.1 SQL 语言比较运算符列表

运算符	说 明	运算符	说 明
=	等于	>=	大于等于
>	大于	<=	小于等于
<	小于	<>	不等于

表 5.2 SQL 语言逻辑运算符列表

运算符	说 明
And	逻辑与，表示两个条件都为 true 时才返回 true，否则返回 false
Or	逻辑或，表示只要有一个条件为 true，即返回 true，否则返回 false
Not	逻辑非，如果条件为 true，则返回 false；如果条件为 false，则返回 true

另外，SQL 还支持一些特殊的运算符，这些运算符在查询特殊条件的数据时非常有效，见表 5.3。

表 5.3 SQL 语言特殊运算符列表

运算符	说 明
Between	介于，如“订购量 between 10 and 40”表示查询订购量在 10~40 之间的值
Not Between	不介于
In	在指定值列表范围内，如“订购量 in (10,20,30,40)”表示查询订购量为 10、20、30 或 40 的所有记录
Not In	不在指定值范围内
Is Null	记录值为空的记录
Is Not Null	记录为非空的记录
Like	匹配查询，如“产品名称 like %奶%”表示查询在产品名称中包含“奶”的记录。Like 关键字包含的常用通配符见表 5.4

表 5.4 Like 常用通配符列表

通配符	说 明
%	任何长度的字符串
_ (下划线)	任意一个字符
[] (中括号)	某个范围内的一个字符，如[123]表示从 1、2 和 3 中取一个值，[1-9]表示从 1~9 中间取一个值，[abc]表示从 a、b 和 c 中取一个小写字母，[A-Z]表示从大写字母 A 到 Z 之间取一个大写字母

例如：

(1) 查询所有没有中止合同且单价大于 18 元的产品记录。

```
Select * from 产品 Where 中止=false And 单价>18
```

(2) 查询库存量从 52~100 之间的所有产品记录。

```
Select * from 产品 Where 库存量 between 52 and 100
```

(3) 查询订购量为 10、20、30、40 或 50 的所有产品记录。

```
Select * from 产品 Where 订购量 in (10,20,30,40,50)
```

说明：关键字 IN 与后面的括号可以添加空格，括号内值列表之间要用逗号隔开。

(4) 在供应商表中查询没有填写传真号码的记录。

```
Select * from 供应商 Where 传真 is Null
```

(5) 查询产品名称中包含肉字的产品记录。

```
Select * from 产品 Where 产品名称 like '%肉%'
```

(6) 查询产品名称中以肉字结尾的产品记录。

```
Select * from 产品 Where 产品名称 like '%肉'
```

(7) 查询产品单位数量为每箱 1~29 瓶的所有产品记录。

```
Select * from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

6. 查询记录排序

```
Select * from 产品 Where 单位数量 like '每箱[12][0-9]瓶' Order By 单价
```

说明：查询记录排序一般使用 ORDER BY 子句，在 ORDER BY 后面加上要进行排序的字段，默认为升序。上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后根据单价进行升序排列。

(1) 如果要降序排列，需要在子句后面加 DESC (descending) 关键字，例如：

```
Select * from 产品 Where 单位数量 like '每箱[12][0-9]瓶' Order By 单价 DESC
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后根据单价进行降序排列。

(2) 如果要升序排列，需要在子句后面加 ASC (ascending) 关键字，例如：

```
Select * from 产品 Where 单位数量 like '每箱[12][0-9]瓶' Order By 单价 ASC
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后根据单价进行升序排列。

5.2.3 Select 聚合查询

SELECT 聚合查询就是根据 SQL 提供的聚合函数来计算查询的记录。常用聚合函数包括 Sum 函数、Avg 函数、Count 函数、Max 函数和 Min 函数。

1. Sum

计算某个查询字段的总和。例如：

```
Select Sum(库存量) from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后计算这些产品库存量的总和。注意，聚合查询返回的是一个值，而不是一个表。执行上面语句后的结果如图 5.3 所示。



图 5.3 汇总库存量

2. Avg

计算某个查询字段的平均值。例如：

```
Select Avg(单价) from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后计算它们的平均单价。

3. Count

计算某个查询字段的记录个数。例如：

```
Select Count(单价) from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后计算它们的个数。

4. Max

计算某个查询字段的最大值。例如：

```
Select Max(单价) from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后查找其中的最大值。注意：使用 Max 函数查询返回的是一个最大值，而不是最大值所在的记录或字段值，如执行上面 SQL 字符串所得结果如图 5.4 所示。



图 5.4 查找单价最大值

5. Min

计算某个查询字段的最小值。例如：

```
Select Min(单价) from 产品 Where 单位数量 like '每箱[12][0-9]瓶'
```

说明：上面语句将查询产品单位数量为每箱 1~29 瓶的所有产品记录，然后查找其中的最小值。

5.2.4 Select 分组查询

分组查询就是使用 GROUP BY 子句对查询字段进行分组，或附带 HAVING 子句查询指定条件的组。例如：

```
Select 供应商ID, Sum(库存量) from 产品 Group By 供应商ID
```

说明：上面 SQL 字符串根据产品表中的供应商 ID 字段进行分组，然后计算每组的总库存量。执行上面 SQL 字符串，会返回如图 5.5 所示的结果。

供应商ID	Expr1001
1	69
2	133
3	141
4	64
5	108
6	98
7	68
8	74
9	207

图 5.5 分组计算总库存量

也可以给聚合函数字段起个别名, 例如:

Select 供应商ID, Sum(库存量) As 库存总量 from 产品 Group By 供应商ID

执行上面 SQL 字符串, 会返回如图 5.6 所示的结果。

供应商ID	库存总量
1	69
2	133
3	141
4	64
5	108
6	98
7	68
8	74
9	207
10	20
11	140

图 5.6 给聚合函数分组起别名

在使用分组查询时应注意:

- (1) WHERE 子句必须放在 GROUP BY 子句前面;
- (2) SELECT 子句只能选择分组字段或聚合函数;
- (3) HAVING 子句必须放在 GROUP BY 子句之后, 且只能包含分组字段或聚合函数。

例如:

Select 库存量, Avg(单价) from 产品 Group By 库存量 Having 库存量=112

说明: 上面 SQL 字符串根据产品表中的库存量字段进行分组, 然后计算库存量为 112 的所有记录的单价平均值, 并显示计算结果, 此时返回的结果不是一个值, 而是一条记录。执行上面 SQL 字符串, 会返回如图 5.7 所示的结果。

库存量	Expr1001
112	10.75

图 5.7 显示指定分组的平均单价

对于上面语句也可以用下面 SQL 字符串表示:

```
Select 库存量,Avg(单价) from 产品 Where 库存量=112 Group By 库存量
```

5.2.5 Select 多表查询

多表查询就是根据多个表之间的内在联系,使用一个 SELECT 子句同时从多个数据表中获取数据。例如:在 Northwind.mdb 数据库中,产品表通过供应商 ID 字段与供应商表的主键字段供应商 ID 建立表之间的连接,如图 5.8 所示。

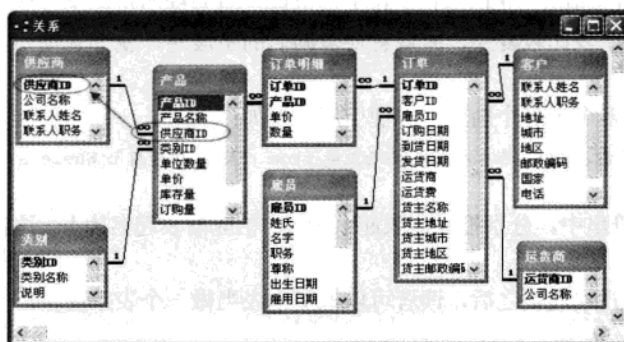


图 5.8 产品表和供应商表关系图

现在,把 5.2.4 小节中的 SQL 字符串:

```
Select 供应商ID,Sum(库存量) from 产品 Group By 供应商ID
```

改写为:

```
Select 公司名称,Sum(库存量) As 库存总量 from 产品,供应商 Where 产品.供应商ID=供应商.供应商ID Group By 公司名称
```

执行这个 SQL 字符串,查询结果如图 5.9 所示。

公司名称	库存总量
百达	130
成记	86
大柱	62
德昌	98
德级	164
东海	10
福满多	23
福合	58
宏仁	57
佳佳	136
佳佳成	50

图 5.9 多表查询结果

读者在使用多表查询时应注意以下问题。

(1) 首先必须保证要查询的多个表之间有连接,即存在公共字段,如产品表和供应商表都有供应商 ID 字段,且数据类型相同。

(2) 公共字段必须在一个表中是主键,即在一个表中该字段值是唯一的,不能出现重复或空值。

(3) 在 SQL 字符串的 FROM 子句中列出所有连接的表名,并用逗号隔开。

(4) 在 SQL 字符串的 WHERE 子句中用等号把各个表中的公共字段连接在一起,例如,产品.供应商 ID=供应商.供应商 ID,表名与字段名之间的点号表示所属关系。

(5) 在 SQL 字符串的 SELECT 子句中也可以用点号标明每个字段所属的表,例如供应商公司名称。当多个表比较复杂,或者有很多同名的字段时,就必须使用点号标明字段所属哪个表。

(6) 也可以为多个表起个别名,方便引用,例如:

```
Select b.公司名称,Sum(a.库存量) As 库存总量 from 产品 a,供应商 b Where a.供应商ID=b.供应商ID
Group By b.公司名称
```

上面 SQL 字符串中,分别把产品表起名为 a, 供应商表起名为 b。这样就方便在字符串中引用。

(7) 当多个表建立连接之后,读者可以把多个表当做一个表来操作。

5.2.6 Insert 插入数据

Insert 语句在 SQL 语言中表示向数据表中插入新的记录,语法如下:

```
Insert Into 表名(字段1,字段2,...) Values (字段1的值,字段2的值,...)
```

说明如下。

(1) 利用上面语句可以给表中全部或部分字段赋值,表名和 Values 后面括号中的列表必须是一一对应的,列表之间用逗号隔开。

(2) 如果字段值类型为字符串或备注类型,则应把字段值用单引号引起;如果字段类型为日期,则应在字段值前后加#;如果字段类型为布尔型,则字段值为 True 或 False;如果是自动编号,则不需要赋值。

(3) 如果保留默认值,则字段值为 Default;如果输入空值,则字段值为 Null。

(4) 如果某个字段为必填字段,而在 Insert 语句中没有插入该字段值,则就会报错。

(5) 该语法适合 Access 数据库,SQL Server 数据库中略有不同,它不需要使用 Into 关键字。

例如,在本地站点内新建一个文件保存为 operate.asp,然后输入下面代码来验证下面 SQL 字符串。

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<% '定义数据库连接
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '实例化Connection对象
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置数据驱动
conn.Open Server.MapPath("data/NorthWind.mdb") '数据库的相对路径
%>
<%
'在此处输入用户自己编写的SQL操作语句
sql = "insert into 产品(产品名称) values ('上海青') "
'执行SQL命令,没有返回值
conn.Execute sql
```

```
'销毁连接对象实例
set conn=nothing
%>
```

使用时只需要把 SQL 字符串替换上面代码中的 sql 变量即可。执行操作后在浏览器中会看不到任何执行结果，读者可以打开远程文件夹中的数据库查看插入的数据。

(1) 在产品表中插入一个产品名称。执行下面 SQL 字符串会在产品表中新插入一条记录，该记录仅在“产品名称”列中显示数据。

```
insert into 产品(产品名称) values ('上海青')
```

(2) 在产品表中插入一个产品名称和单价。

```
sql = "insert into 产品(产品名称, 单价) values ('海鱼', '¥34.80')"
```

(3) 在雇员表中插入一个姓名和出生日期。

```
insert into 雇员(姓氏, 名字, 出生日期) values ('张', '三', #12/12/1983#)
```

说明：在雇员表中，姓氏字段是必填字段，因此 Insert 语句中也必须赋值，同时如果在数据库中设置了各种规则，也会对插入值产生影响，例如，在雇员表中，因为出生日期字段设置了有效性规则为“<Date()”，所以赋值时出生日期必须小于现在日期，否则会报错。

5.2.7 Update 更新数据

Update 语句在 SQL 语言中表示修改数据表中一条或多条记录，语法如下：

```
Update 表名 Set 字段1=值, 字段2=值, ... [Where 条件]
```

说明：Update 语句可以更新一条或多条记录，主要根据 WHERE 子句指定的条件，凡是符合 WHERE 子句指定的条件的记录都将被更新，如果没有符合条件的记录，则不更新记录。因此，读者在操作时要务必小心，最好将数据库做好备份，避免更新了不该更新的记录。

例如，在 operate.asp 文件中用下面 SQL 字符串进行替换练习。

(1) 在产品表中把产品 ID 等于 78 的产品名称更改为“淡水鱼”。

```
Update 产品 Set 产品名称='淡水鱼' Where 产品ID=78
```

(2) 在供应商表中把供应商 ID 为 41 和 42 的公司名称更改为“戴尔”。

```
Update 供应商 Set 公司名称='戴尔' Where 供应商ID between 41 and 42
```

5.2.8 Delete 删除数据

Delete 语句在 SQL 语言中表示删除数据表中一条或多条记录，语法如下：

```
Delete from 表名 [Where 条件]
```

说明：Delete 语句和 Update 语句用法相同，根据 WHERE 条件来确定要删除的记录，如果没有符合 Where 条件，则不删除任何记录，但如果要省略 WHERE 条件，将删除所有数据。所以，读者在练习时务必小心，最好对数据库做好备份，避免删除全部数据或不该删除的数据。

例如，在供应商表中删除供应商 ID 为 41 和 42 的供应商记录。

```
Delete from 产品 Where 产品ID between 41 and 42
```

5.3 认识 ADO

ASP 对数据库的访问和操作主要通过内置组件 ADO 来实现的，ADO 是在微软公司的数

数据库应用开发接口 (API) OLE DB 技术基础上实现的, 这些技术都是基于 ODBC 数据引擎。ADO 组件主要包含 7 个对象: Connection、Command、Parameter、RecordSet、Field、Property 和 Error。详细说明如下。

- Connection 对象用于创建 ASP 脚本与指定数据库的连接。在使用任何数据库之前, 脚本必须首先要与数据库建立连接, 然后才能继续下一步的操作。使用 Connection 对象可以与各种类型的数据库相连接 (如微软公司的 SQL Server 或 Access 等), 也可以同 Microsoft Excel 建立连接。Connection 对象除了与数据库进行连接之外, 还可以通过事务 (Transaction) 确保在事务中所有数据库变更完全成功。
- Command 对象负责向数据库提供请求, 也就是传递指定的 SQL 命令。Command 对象查询数据库后, 将返回 RecordSet 对象记录, 以便执行大量操作, 例如, 增加记录 (INSERT)、删除记录 (DELETE)、更新记录 (UPDATE)、筛选记录 (SELECT) 等操作, 或处理数据库结构, 建立数据表 (CREATE TABLE)、删除数据表 (DELETE TABLE)、修改表结构 (ALTER TABLE) 等。
- Parameter 对象负责传递 Command 对象所需要的 SQL 命令参数。
- RecordSet 对象负责浏览与操作从数据库中查询的记录集 (即存储于内存中的临时数据表), 也就是说 RecordSet 对象可以用来表示来自表或命令执行结果的记录全集。在使用 ADO 时, 通过 RecordSet 对象几乎可以对所有数据进行操作, 所有 RecordSet 对象均使用记录 (行) 和字段 (列) 进行构造。
- Field 对象指定 RecordSet 对象的数据字段。每个 Field 对象对应于 RecordSet 中的一列, 可以使用 Field 对象的 Value 属性设置或返回当前记录的数据。
- Property 对象表示 ADO 的各项对象属性值, 每个 ADO 对象都会有一个描述或控制对象行的属性。
- Error 对象负责记录连接过程中所有发生的错误信息。每当错误出现时, 一个或多个 Error 对象将被放到 Connection 对象的 Errors 集合中。
- ADO 对象包含 Fields、Properties、Parameters 和 Error 这 4 个数据集合。
- Fields 数据集合。RecordSet 对象包括 Field 对象组成的 Fields 集合。每个 Field 对象对应 RecordSet 集中的一列。
- Properties 数据集合。某些 ADO 对象 (如 Connection 对象、Command 对象、RecordSet 对象、Field 对象等) 包含由 Property 对象组成的 Properties 集合, 每个 Property 对象与指定给提供者的 ADO 对象的特性相对应。
- Parameters 数据集合。包含了 Command 对象的所有 Parameter 对象。Command 对象具有由 Parameter 对象组成的 Parameters 集合。
- Errors 数据集合。包含响应单个失败 (涉及提供者) 时产生的所有 Error 对象。Errors 集合中的 Error 对象描述响应单个语句时产生的所有错误。

5.4 数据库连接

要实现与数据库的连接, 必须使用 Connection 对象。本节将详细介绍各种可视化操作和代码直接连接的方法, 帮助读者建立数据库的连接操作。

5.4.1 创建 ODBC 连接

ODBC (Open Database Connectivity, 开放式数据库连接) 是数据库服务器的一个标准协议, 它向访问网络数据库的应用程序提供一种通用的语言。只要系统中有相应的 ODBC 驱动程序, 任何程序都可以通过 ODBC 操纵数据库。

每个 ODBC 数据源都被指定一个名字, 即 DSN (Data Source Name, 数据源名称)。DSN 表示将应用程序和某个数据库建立连接的信息集合。ODBC 数据源管理器就是利用该信息来创建管理指向的数据库连接。通常 DSN 可以保存在文件或注册表中。

建立 ODBC 连接, 实际就是创建同数据源的连接, 也就是创建 DSN。一旦建立了与数据库的 ODBC 连接, 那么同该数据库的连接信息将被保存在 DSN 中, 程序的运行必须通过 DSN 来进行。

1. ODBC 数据源管理器

要在动态网页中使用 ADO 对象来操作数据库, 首先要创建一个指向该数据库的连接。在 Windows 系统中, ODBC 的连接主要是通过 ODBC 数据库资源管理器来完成。

在管理工具中, 可以看到【ODBC】、【32 位 ODBC】或【数据源 ODBC】的图标 (不同的 Windows 版本系统名称不同), 如图 5.10 所示。双击该图标便可启动 ODBC 数据源管理器, 在该管理器中包含了许多选项卡, 允许对该管理器进行多项 ODBC 操作。在默认状态下, ODBC 已经内置了多种数据驱动程序。可以单击【驱动程序】选项卡, 查看当前要连接的数据类型是否位于其中。如果没有, 需要下载并安装相应的驱动程序, 如图 5.11 所示。



图 5.10 Windows XP 控制面板

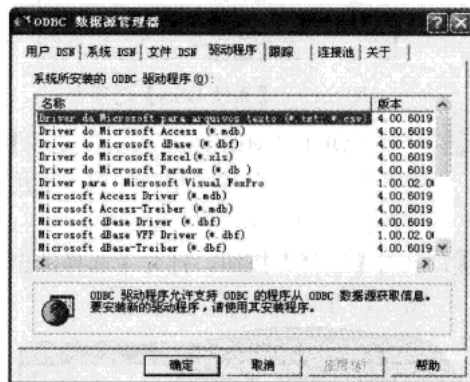


图 5.11 ODBC 数据源管理器

在【ODBC 数据源管理器】中可以看到【用户 DSN】、【系统 DSN】和【文件 DSN】, 表明可通过【ODBC 数据源管理器】创建 3 种类型的 DSN。

- 【用户 DSN】: 定义用户使用的 DSN, ASP 是不能使用的。用户 DSN 通常保存在注册表中。
- 【系统 DSN】: 定义系统进程所使用的 DSN, 系统 DSN 信息同用户 DSN 一样被存储在注册表中。ASP 与数据库的连接主要通过该选项卡来实现。
- 【文件 DSN】: 与系统 DSN 有所区别, 是保存在文件中, 而不是注册表中。

2. 创建系统 DSN

建立系统 DSN 具体操作步骤如下。

- 01** 在操作系统中启动【Windows 控制面板】(不同操作系统的启动方法有所不同, 本书主要针对 Windows XP 进行介绍), 然后切换到【经典视图】, 双击【管理工具】图标, 打开【管理工具】面板(如图 5.10 所示)。再双击【数据源(ODBC)】图标, 打开【ODBC 数据源管理器】对话框(如图 5.11 所示)。
- 02** 在【ODBC 数据源管理器】对话框中, 单击【系统 DSN】标签, 切换到【系统 DSN】选项卡, 如图 5.12 所示。
- 03** 单击【添加】按钮, 进入【创建新数据源】对话框, 选择数据源类型。如果使用 Access 2007 版本以前的数据库软件, 应在列表中选择 Microsoft Access Driver (*.mdb); 如果使用 Access 2007 版本, 应在列表中选择 Microsoft Access Driver (*.mdb, *.accdb), 然后单击【完成】按钮, 如图 5.13 所示。



只有当用户在本地计算机中安装了 Access 2007 软件或者 Access 2007 驱动程序, 才能在列表中选择 Microsoft Access Driver (*.mdb, *.accdb) 选项。



图 5.12 【系统 DSN】选项卡



图 5.13 【创建新数据源】对话框

- 04** 打开【ODBC Microsoft Access 安装】对话框, 如图 5.14 所示。在【数据源名】文本框中输入数据源的名称(此名称用于调用打开数据库时使用), 在【说明】文本框中可输入对该数据库的描述性文字。
- 05** 单击【选择】按钮, 打开【选择数据库】对话框, 选取要提供数据的 Access 数据库, 如图 5.15 所示。



图 5.14 【ODBC Microsoft Access 安装】对话框

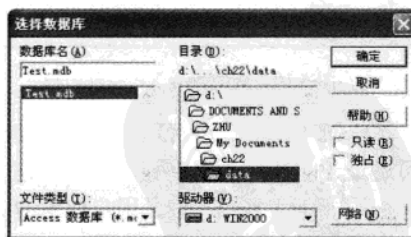


图 5.15 【选择数据库】对话框

- 06** 选择数据库后, 单击【确定】按钮, 返回到【ODBC Microsoft Access 安装】对话框时, 将可以看到新增了一个 ODBC 数据源, 如图 5.16 所示。
- 07** 单击【确定】按钮, 返回到【ODBC 数据源管理器】对话框, 在【系统 DSN】选项卡中可以看到刚建立的数据库连接, 如图 5.17 所示, 至此与数据库的连接就建立好了。



图 5.16 安装 ODBC Microsoft Access



图 5.17 显示安装完毕的数据源

5.4.2 使用 DSN 创建数据库连接


如果用户的动态站点服务器（或者本地服务器）和 Dreamweaver CS5 软件都运行在同一个 Windows 系统上，那么就可以使用系统 DSN（Data Source Name）来创建数据库连接，DSN 是指向数据库连接的一个快捷方式。



提示



如果 Dreamweaver CS5 没有安装在站点服务器上，那么需要使用自定义连接字符串（Connection string）建立一个连接。

如果 Dreamweaver CS5 软件安装在网页服务器所在计算机上，建立一个数据库连接可以使用 DSN 快速实现连接。具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中先建立一个动态站点（其中【服务器类型】为“ASP VBScript”，【访问】类型为“本地/网络”），选择【窗口】|【数据库】菜单命令，打开【数据库】面板，然后单击【加号】按钮，弹出下拉菜单，如图 5.18 所示。



注意

在建立数据连接之前，用户应该先建立一个动态服务器技术的站点，并打开站点内要运用数据库的网页文件，否则【加号】按钮显示无效，如图 5.19 所示。当然也可以按照列表框中提示的步骤新建站点，设置服务器文档类型，即该文档使用什么服务器技术和脚本语言支持。同时还要设置测试服务器。当各项前边显示一个对号✓时，说明可以建立数据库连接，【加号】按钮显示有效，如图 5.18 所示。

- 02** 在下拉菜单选择【数据源名称 (DSN)】项，打开【数据源名称 (DSN)】对话框，如图 5.20 所示。

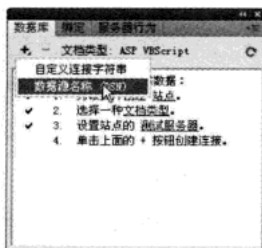


图 5.18 选择连接定义的方式

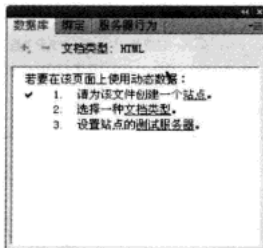


图 5.19 无效状态

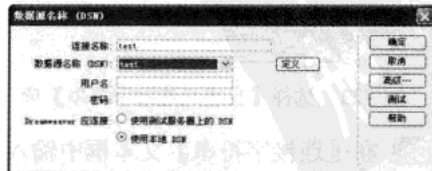


图 5.20 【数据源名称 (DSN)】对话框

- 03 在【连接名称】文本框处输入一个字符串作为连接名。为 DSN 命名时建议添加 com 前缀, 这样能够使程序更容易阅读。
- 04 在【数据源名称 (DSN)】下拉列表框中选择所需的 DSN, 从该下拉列表框中选择 test (在 5.4.1 节中已经定义好的 DSN), 其余项目就不用填写了。如果设置了数据库的用户名和密码, 还需要设置【用户名】和【密码】文本框。
- 05 单击【测试】按钮, 稍后如果看到如图 5.21 所示的对话框, 说明已经成功地建立了与数据库的连接。单击【确定】按钮关闭【数据源名称 (DSN)】对话框, 此时新建的连接会出现在【数据库】面板中, 如图 5.22 所示。



注意

默认情况下, 本书实例的数据库连接都以上面操作步骤实现, 在后面实例介绍中除非运用其他数据库连接方式, 否则就不再重复介绍。



图 5.21 提示对话框



图 5.22 【数据库】面板

5.4.3 使用自定义连接字符串创建数据库连接

使用自定义连接字符串创建数据库连接的操作步骤如下。

- 01 将数据库文件上传到远程服务器。记下它的虚拟路径, 例如, data/NorthWind.mdb。
- 02 在 Dreamweaver CS5 中打开已建立的动态网站, 选择【窗口】|【数据库】命令, 打开【数据库】面板。
- 03 单击【数据库】面板上的【加号】按钮 \oplus , 从弹出菜单中选择【自定义连接字符串】选项, 如图 5.23 所示。
- 04 打开【自定义连接字符串】对话框, 如图 5.24 所示, 在【连接名称】文本框中输入新连接的名称。

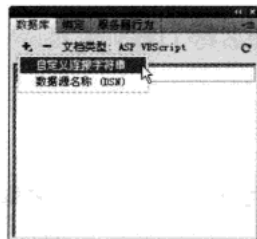


图 5.23 选择【自定义连接字符串】项

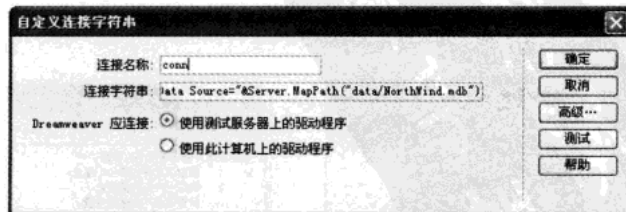


图 5.24 【自定义连接字符串】对话框

- 05 在【连接字符串】文本框中输入自定义连接字符串。由于用户只能知道数据库的相对路径, 如 data/NorthWind.mdb, 因此必须使用 MapPath()函数获取该相对路径的物理路径。详细的自定义连接字符串如下:

```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath("data/NorthWind.mdb")
```

或者

```
"Driver={Microsoft Access Driver (*.mdb)};DBQ=" & Server.MapPath("data/NorthWind.mdb ")
```

如果要连接到 Access 2007 数据库, 则自定义连接字符串表示如下:

```
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & Server.MapPath("罗斯文 2007.accdb")
```



Server.MapPath(path)方法是 ASP 中 Server 对象的一个方法, 该方法能够返回与 Web 服务器上的指定文件的虚拟路径相对应的物理路径。例如: Server.MapPath("/")可以返回应用程序根目录所在的位置, 如 C:\Inetpub\wwwroot\; Server.MapPath("./")可以返回所在页面的当前目录, 与 Server.MapPath("")相同都可以所在页面的物理文件路径, Server.MapPath("../")表示上一级目录, Server.MapPath("~/")表示当前应用级程序的目录, 如果是根目录, 就是根目录, 如果是虚拟目录, 就是虚拟目录所在的位置, 如 C:\Inetpub\wwwroot\mysite\。

05

- 06** 选择【使用测试服务器上驱动程序】复选框, 单击【测试】按钮, Dreamweaver CS5 尝试连接到数据库。如果连接成功, 会显示提示信息; 如果连接失败, 请复查连接字符串。

5.4.4 使用 Connection 对象创建数据库连接

从 ADO 存取数据源的步骤看, 数据库访问操作的第一步是建立数据库的连接。在前面介绍了几种数据库连接的可视化方法, 但都存在着兼容性差、容易出错的问题。如果用户熟悉 ASP 语言及其对象, 建议使用 Connection 对象直接建立数据库连接。假设要与站点根目录下的 data/NorthWind.mdb 数据库建立连接, 主要有下面几种方法。

- 在 Connection 对象中设置 ConnectionString 参数时, 可以通过设置系统数据源(System DSN), 即通过设置 ODBC 的方法连接数据库。关于 DSN 的设定请参阅前面的介绍。在 ASP 中利用 DSN 连接数据库的代码如下:

```
Set conn=Server.CreateObject("ADODB.Connection")
conn.Open "DSN=data;UID=sa;PWD=;Database=data"
```

在上面代码中, 第 1 行先定义一个 Connection 对象实例, 在第 2 行中调用 Connection 对象的方法 Open 打开数据库连接, 其中 DSN 参数用来设置用户在系统中创建的 DSN 名称, 创建方法可以参考 5.4.1 节内容。如果数据库设置了用户名和密码, 需要在 UID 参数中指明用户名, 在 PWD 参数中指明用户密码。参数 Database 用来设置数据库的物理路径, 可以结合 MapPath()方法把相对路径转换为物理路径。

- 如果不采用 DSN 连接数据库, 在设置 ConnectionString 参数时, 可以采用直接指定 ODBC 驱动程序来连接数据库, 程序代码如下:

```
Set conn=Server.CreateObject("ADODB.Connection")
strProvider="Driver={Microsoft Access Drier (*.mdb)};DBQ=" & Server.MapPath ("\"") & "\
data.mdb;UID=sa;PWD;"
conn.Open strProvider
```



说明

首先定义一个连接实例, 然后设置 Access 驱动打开数据库。

- 在 Connection 对象中可以通过设置 Connection 对象的 Provider 属性来设置连接信息, 代码如下:

```
Set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open Server.MapPath("data.mdb")
```



说明

首先定义一个连接实例，然后用 OLE DB 驱动程序打开数据库。

- 在其他对象中，也可以通过设置 ActiveConnection 参数完成对数据源连接的设置。以常用的 RecordSet 对象为例，可以使用该对象的 Open 方法，利用设置好的 ActiveConnection 参数完成对数据库的连接，程序代码如下：

```
Set conn=Server.CreateObject("ADODB.Connection")
DbPath=Server.MapPath("data.mdb")
conn.Open "Driver={Microsoft Access Driver (*.mdb)};DBQ=" & DbPath
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open "data", conn, 3
```



说明

首先定义一个连接实例，然后用 RecordSet 对象打开数据库。

5.5 Connection 对象

在上节中详细介绍了数据库连接的各种方法。实际上数据库连接主要依靠 Connection 对象来实现。其实 Connection 对象不仅仅用于数据库连接，它还提供了很多属性、方法和集合。充分利用这些属性、方法和集合能够帮助读者解决很多实际问题。

Connection 对象描述了 ASP 到数据源的物理连接。用户可以使用 ODBC 或 OLE DB 来连接数据源。当打开一个 Connection 对象时，State 属性会显示连接是否成功，通过 Connection 对象的 Execute 方法向数据源发送 SQL 语句或运行存储过程。如果向数据源发送的命令要求数据源返回记录集，那么记录集对象会自动被创建。当连接上数据库后用户可以使用 Close 方法关闭 Connection 对象。

5.5.1 Connection 对象的属性

Connection 对象的属性主要包括 Attributes、CommandTimeout、ConnectionString、ConnectionTimeout、CursorLocation、DefaultDatabase、IsolationLevel、Mode、Provider、State、Version 等，Connection 对象的属性详细说明见表 5.5。

表 5.5 Connection 对象的属性列表

属 性	说 明
Attributes	指示对象的一项或多项特性
CommandTimeout	设置 Connection 对象的 Execute 方法运行的最长执行时间，其属性值为一个长整型变量，默认值为 30 秒。如果将 CommandTimeout 的值设为 0，系统会一直等到运行结束为止
ConnectionString	包含用于建立连接数据源的信息
ConnectionTimeout	设置 Connection 对象的 Open 方法与数据库连接时的最长等待时间，其属性值为一个长整型变量，缺省值为 15 秒。如果设为 0，系统会一直等到连接成功为止
CursorLocation	设置服务器指针处理的位置，取值包括 adUseClient、adUserServer、adUseclientBatch
DefaultDatabase	设置或返回字符串来计算出从提供者处可用的数据库的名称
IsolationLevel	设置 Connection 对象运行事务的时机
Mode	设置修改数据库的权限，参数取值见表 5.6

续表

属 性	说 明
Provider	取得或者设置 Connection 对象的驱动程序（即提供者，内定数据库管理程序的名称），默认值为 MSDASQL（Microsoft OLE Db Provider For ODBC），它负责管理所有以 ODBC 连接的数据库
State	获取 Connection 对象的状态，取值包括 adStateClosed 和 adStateOpen
Version	获取 ADO 版权信息

表 5.6 Mode 属性取值列表

常 数	说 明
AdModeUnknown	默认值。表明权限尚未设置或无法确定
AdModeRead	表明权限为只写
AdModeWrite	表明权限为只读
AdModeReadWrite	表明权限为读/写
AdModeShareDenyRead	防止其他用户使用读权限打开连接
AdModeShareDenyWrite	防止其他用户使用写权限打开连接
AdModeShareExclusive	防止其他用户打开连接
AdModeShareDenyNone	防止其他用户使用任何权限打开连接



提示

所有 ADO 内置常量表都保存在 adovbs.inc（针对 VBScript 脚本语言）文件中，该文件位于系统 C:\Program Files\Common Files\SYSTEM\ADO 目录下，当读者在自己的脚本中引用了 ADO 内置常量时，必须把 adovbs.inc 文件复制到站点内，并用 `<!--#include virtual="根目录路径\adovbs.inc" -->` 或者 `<!--#include file="相对路径\adovbs.inc" -->` 语句把该文件包含到动态页面中。

在 Connection 对象的属性中比较常用的属性包括 CommandTimeout、ConnectionString、ConnectionTimeout、Mode 和 Provider。下面代码综合演示了 Connection 对象使用：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb ") '打开数据库
%>
<%
If conn.State = adStateOpen Then '检查是否完成连接
    Response.Write "<script>alert('数据库连接成功!');</script>"
Else
    Response.Write "<script>alert('数据库没有连接成功!');</script>"
End If
conn.Close '关闭connection对象
%>
```

5.5.2 Connection 对象的方法

在 ADO 中，Connection 对象是其他对象操作的基础，所有对象都必须依赖 Connection 对象的连接才能发挥其特有的功能。Connection 对象代表与数据源进行唯一会话，要实现这种会话，必须发挥 Connection 对象的方法的作用。Connection 对象的方法说明见表 5.7。

表 5.7 Connection 对象的方法列表

方 法	说 明
BeginTrans	开始新事务
CommitTrans	保存任何更改并结束当前事务。它也可能启动新事务
RollbackTrans	取消当前事务中所作的任何更改并结束事务。它也可能启动新事务
Cancel	取消执行挂起的异步 Execute 或 Open 方法的调用
Close	关闭打开的对象及任何相关对象
Execute	执行指定的查询、SQL 语句、存储过程或特定提供者的文本等内容
Open	打开到数据源的连接
OpenSchema	从提供者（驱动程序）获取数据库模式信息

在 Connection 对象的方法列表中, Open 和 Execute 方法是最常用也比较实用的 2 个方法, 下面结合这 2 个方法进行详细介绍。

1. Open

Open 方法负责打开到数据源的连接。其语法格式如下:

```
connection.Open ConnectionString, UserID, Password, Options
```

在上面语法中, 参数 ConnectionString 包含连接信息, 参数 UserID 表示建立连接时所使用的用户名, 参数 Password 表示建立连接时所使用的密码, 参数 Options 表示该方法是在连接建立之后（异步）还是连接建立之前（同步）返回, 取值见表 5.8。

表 5.8 参数 Options 取值列表

常 数	说 明
adConnectUnspecified	（默认）同步打开连接
adAsyncConnect	异步打开连接。ConnectComplete 事件可以用于决定连接何时可用

使用 Connection 对象的 Open 方法可以建立到数据源的物理连接。ConnectionString 参数是一个可选参数, 它指定连接字符串, 包含由分号分隔的一系列 argument=value 语句。ConnectionString 属性自动继承用于 ConnectionString 参数的值, 因此可在打开之前设置 Connection 对象的 ConnectionString 属性, 或在 Open 方法调用时使用 ConnectionString 参数设置或覆盖当前连接参数。

如果在 ConnectionString 参数和可选的 UserID 和 Password 参数中传送用户和密码信息, 那么 UserID 和 Password 参数将覆盖 ConnectionString 中指定的值。

在对打开的 Connection 的操作结束后, 可使用 Close 方法释放所有关联的系统资源。关闭对象并非将它从内存中删除; 可以更改它的属性设置并在以后再次使用 Open 方法打开它。要将对象完全从内存中删除, 可将对象变量设置为 Nothing。



注意

当在客户端的 Connection 对象上使用 Open 方法时, 在 Connection 对象上打开 Recordset 之前 Open 方法其实并未建立到服务器的连接。

下面将介绍如何使用 Connection 对象的 Open 方法具体连接一个数据源。通过前面章节建立的 DSN 连接 Microsoft Access 数据库, 该方法的连接字符串包括 3 个部分: 数据库的 DSN 名、连接到数据库所使用的用户名、该用户的密码。

```
<%  
Dim conn
```

```
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open("DSN=db1;UID=zhu;PWD=123;")
%>
```

下面的例子利用数据库本身的专用接口，即指定数据库驱动程序来连接。这里用的是 Microsoft Access 数据库，因此该专用接口为“{Microsoft Access Driver (*.mdb)}”，连接的数据库与上面相同。该方法的连接字符串可以包括 4 个部分：专用数据库接口名称、数据库位置及名称、用户名、用户口令。

```
<%
Dim conn
Dim strProvider
Set conn=Server.CreateObject("ADODB.Connection")
strProvider="Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("data/NorthWind.mdb") & ";UID=;PWD=;"
conn.Open strProvider
%>
```

通过 Connection 对象的 Provider 属性，指定 OLE DB 驱动

```
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
```

2. Execute

Execute 方法执行指定的查询、SQL 语句、存储过程或特定提供者的文本等内容，执行 Execute 方法之后，会返回 Recordset 对象引用。

对于非按行返回的命令字符串，语法格式如下：

connection.Execute CommandText, RecordsAffected, Options

对于按行返回的命令字符串，语法格式如下：

Set recordset = connection.Execute (CommandText, RecordsAffected, Options)

在上面语法中，参数 CommandText 包含要执行的 SQL 语句、表名、存储过程或特定提供者的文本，参数 RecordsAffected 表示提供者向其返回操作所影响的记录数目，参数 Options 指示提供者应如何计算 CommandText 参数，取值见表 5.9。

表 5.9 参数 Options 取值列表

常 数	说 明
AdCmdText	指示提供者应按命令的文本定义计算 CommandText
AdCmdTable	指示 ADO 应生成 SQL 查询以便从 CommandText 命名的表中返回所有行
AdCmdTableDirect	指示提供者应从 CommandText 命名的表中返回所有行
AdCmdTable	指示提供者应按表名计算 CommandText
AdCmdStoredProc	指示提供者应按存储过程计算 CommandText
AdCmdUnknown	指示 CommandText 参数中的命令类型未知
adAsyncExecute	指示命令应该异步执行
adAsyncFetch	指示对在 CacheSize 属性指定的初始数量之后的剩余行使用异步提取

使用 Connection 对象的 Execute 方法，可执行任何在指定连接的 CommandText 参数中发送给方法的查询。如果 CommandText 参数指定按行返回查询，执行产生的任何结果将存储在新的 Recordset 对象中。如果命令不是按行返回的查询，则提供者返回关闭的 Recordset 对象。

返回的 Recordset 对象始终为只读、仅向前的游标。如需要具有更多功能的 Recordset 对象，应首先创建具有所需属性设置的 Recordset 对象，然后使用 Recordset 对象的 Open 方法执行查询并返回所需游标类型。

CommandText 参数的内容对提供者是特定的，并可以是标准的 SQL 语法或提供者支持的任何特殊命令格式。

Execute 方法可以执行标准的 SQL 语句命令，例如，Select（查询提取数据）、Insert（插入数据）、Delete（删除数据）、Update（更新数据）、Create Table（创建数据表）和 Drop Table（删除数据表）操作等。

下面尝试使用这些 SQL 语句实现数据的插入、查看、更新和删除操作。具体操作步骤如下。

01 新建 connection_execute_insert.asp 文件，利用 Execute 方法向 NorthWind.mdb 数据库的产品表中插入一条记录，实现代码如下：

```
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim connString
connString = "Insert into 产品(产品名称, 单价) values('火龙果', 30)"
conn.Execute connString
conn.close
set conn=nothing
%>
```

执行上面代码，如果读者在远程服务器目录下打开 NorthWind.mdb 数据库，会发现新插入的记录。在使用 Execute 方法时可以省略其他参数，直接把数据库操作的 SQL 字符串赋予给该方法即可，SQL 将按默认参数值对其进行操作。

02 新建 connection_execute_select.asp 文件，并输入下面代码，利用 Execute 方法可以查看新插入的记录，代码如下：

```
<%'定义数据库连接
Dim conn
Set conn=Server.CreateObject("ADODB.Connection")'实例化Connection对象
conn.Provider = "Microsoft.Jet.OLEDB.4.0"'设置数据驱动
conn.Open Server.MapPath("data/NorthWind.mdb")'数据库的相对路径
%>
<%
sql = "Select * from 产品 Where 产品名称 = '火龙果'"
Set rs = conn.execute(sql)
%>
<%'显示记录集中所有记录
If not Rs.eof then
Dim fds
Set fds=rs.Fields'实例化记录集Recordset对象的数据集合Fields
Response.Write("<table border=1 width=90% align=center><tr>")
For i=0 To fds.Count-1'循环显示记录集的标题
Response.Write("<td class='thead'>"&fds.Item(i).Name&"</td>")
Next
Response.Write("</tr><tr>")
rs.MoveFirst '把记录集的指针移到第1条记录
While Not rs.eof' 循环读取记录集中每条记录
```

```

        For i=0 To fds.Count-1 '循环读取每条记录集中每个字段值
            Response.Write("<td class='tbody' >"&fds.Item(i).Value&"</td>")
        Next
        Response.Write("</tr><tr>")
        rs.MoveNext '把记录集指针移到下一条记录
    Wend
    Response.Write("</table>")
else
    Response.Write("你没有插入该记录")
End if
%>
<% '释放记录集和数据库连接
rs.close
set conn=nothing
%>

```

- 03** 新建 connection_execute_update.asp 文件，再利用 Execute 方法更新新插入记录的单价，代码如下：

```

<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim connString
connString = "Update 产品 Set 单价 = 60 Where 产品名称='火龙果'"
conn.Execute connString
conn.close
set conn=nothing
%>

```

- 04** 新建 connection_execute_delete.asp 文件，使用 Execute 方法删除新插入的记录，代码如下：

```

<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim connString
connString = "Delete from 产品 Where 产品名称='火龙果'"
conn.Execute connString
conn.close
set conn=nothing
%>

```

3. BeginTrans、CommitTrans 和 RollbackTrans

BeginTrans、CommitTrans 和 RollbackTrans 方法都与 Connection 对象的事务管理有关。如果希望以独立单元保存或取消对源数据所做的一系列更改，则可以使用这些 Connection 对象的方法。例如，在网上银行转账时，必须从账户中减去某个数额并将对等数额添加到另一个账户中。无论其中哪个更新失败，都将导致账户收支不平衡。在打开的事务中进行这些更改可确保只能选择进行全部更改或不作任何更改。

BeginTrans 表示开始新事务，CommitTrans 表示保存任何更改并结束当前事务，它也可能启动新事务，RollbackTrans 表示取消当前事务中所作的任何更改并结束事务，它也可能启动新事务。其语法格式如下：

```
level= connection.BeginTrans()
connection.BeginTrans
connection.CommitTrans
connection.RollbackTrans
```

请注意，并非所有的 DBMS 都支持事务。需要验证 DBMS 定义的属性 Transaction DDL 是否出现在 Connection 对象的 Properties 集合中，如果存在，则表示提供者支持事务。如果 DBMS 不支持事务，调用其中的某个方法将返回错误。

一旦调用了 BeginTrans 方法，调用 CommitTrans 或 RollbackTrans 结束事务之前，提供者将不再立即提交所作的任何更改。

使用 Connection 对象的事务管理结构代码如下：

```
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection")
conn.Provider = "Microsoft.Jet.OLEDB.4.0"
conn.Open Server.MapPath("data/NorthWind.mdb")
%>
<%
sql = "Select * from 产品"
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1
%>
<%
Sub conntrans() '事务处理子过程
    On Error GoTo Err_trans
    conn.BeginTrans '开始事务
    '.....
    conn.execute("sql1")
    '.....
    conn.execute("sql2")
    conn.CommitTrans '结束事务
    Set conn = Nothing
    Exit Sub
Err_trans: '错误操作
    conn.RollbackTrans '滚回事务
    Set conn = Nothing
    Exit Sub
End Sub
%>
```

5.5.3 Connection 对象的集合

Connection 对象提供了两个数据集合：Errors 和 Properties。Errors 数据集合表示 Connection 对象运行时最近一次的错误或警告信息，Properties 数据集合表示 Connection 对象所定义的相关属性。

例如，新建文件并输入下面代码，执行该文件，可以查看新建的 Connection 对象实例的所有属性。

```
<%
Set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.jet.Oledb.4.0"
conn.open Server.MapPath("data/NorthWind.mdb")

Response.Write "<table border=3>"
Response.Write "<tr><td>属性</td>"
Response.Write "<td>属性值</td>"
Response.Write "</tr>"
```

```

For Each item In conn.properties
    Response.Write "<tr><td>"&item.name&"</td>"
    Response.Write "<td>"&item.value&"</td></tr>"
Next
Response.Write "</table>"

Set conn=nothing
%>

```

5.6 Command 对象

05

Command 对象可以查询数据库或者向数据库发送请求, 当查询数据库时可以返回 Recordset 对象中的记录, 以便执行大量操作或处理数据库结构。这些操作都取决于提供者(驱动程序)的功能, 某些 Command 集合、方法或属性被引用时可能会产生错误。

5.6.1 Command 对象的属性

Command 对象提供了 6 个属性, 包括 ActiveConnection、CommandText、CommandType、CommandTimeout、State、Prepared, 见表 5.10。

表 5.10 Command 对象的属性列表

属 性	说 明
ActiveConnection	指示指定的 Command 对象当前所属的 Connection 对象
CommandText	包含要根据提供者发送的命令的文本
CommandTimeout	指示在终止尝试和产生错误之前执行命令期间需等待的时间
CommandType	指示 Command 对象的类型, 取值见表 5.11
Prepared	指示执行前是否保存命令的编译版本。使用 Prepared 属性可使提供者在首次执行 Command 对象前保存 CommandText 属性中指定的已准备好(已编译)的查询版本。该属性会降低命令首次执行的速度, 但提供者对命令进行编译后, 在后继的命令执行中提供者可使用已编译好命令版本, 这样可以提高执行性能
State	对所有可应用对象, 说明其对象状态是打开或是关闭。该属性是只读的

表 5.11 CommandType 属性取值列表

常 数	说 明
AdCmdText	将 CommandText 作为命令或存储过程调用的文本化定义进行计算
AdCmdTable	将 CommandText 作为其列全部由内部生成的 SQL 查询返回的表格的名称进行计算
AdCmdTableDirect	将 CommandText 作为其列全部返回的表格的名称进行计算
AdCmdStoredProc	将 CommandText 作为存储过程名进行计算
AdCmdUnknown	默认值。CommandText 属性中的命令类型未知
adCmdFile	将 CommandText 作为持久 Recordset 文件名进行计算
AdExecuteNoRecords	指示 CommandText 为不返回行的命令或存储过程(例如, 插入数据的命令)。如果检索任意行, 则将丢弃这些行且不返回。它总是与 adCmdText 或 adCmdStoredProc 进行组合

5.6.2 Command 对象的方法

Command 对象提供了 3 个方法, 见表 5.12。

表 5.12 Command 对象的方法列表

方 法	说 明
Cancel	取消执行挂起的异步 Execute 或 Open 方法的调用
CreateParameter	使用指定属性创建新的 Parameter 对象
Execute	执行在 CommandText 属性中指定的查询、SQL 语句或存储过程

1. CreateParameter

CreateParameter 方法能够新建 Parameter 对象。使用 CreateParameter 方法指定参数名称、类型、方向、大小和值，创建新的 Parameter 对象，在参数中传送的所有值都将写入相应的 Parameter 属性。其语法格式如下：

```
Set parameter = command.CreateParameter(Name, Type, Direction, Size, Value)
```

在上面语法中，Parameter 变量表示返回的 Parameter 对象，参数 Name 代表 Parameter 对象名称，参数 Type 表示新建 Parameter 对象数据类型，参数 Direction 用来指定 Parameter 对象类型（见表 5.13），参数 Size 用来指定参数值最大长度（以字符或字节数为单位），参数 Value 用来指定 Parameter 对象的值。

该方法无法自动将 Parameter 对象追加到 Command 对象的 Parameters 集合，这样就可以设置附加属性。如果将 Parameter 对象追加到集合，则 ADO 将使该附加属性的值生效。

表 5.13 参数 Direction 取值列表

常 数	说 明
AdParamUnknown	指示参数方向未知
AdParamInput	默认值。指示输入参数
AdParamOutput	指示输出参数
AdParamInputOutput	同时指示输入参数和输出参数
AdParamReturnValue	指示返回值

如果在 Type 参数中指定可变长度的数据类型，那么在将它追加到 Parameters 集合之前必须指定 Size 参数或者设置 Parameter 对象的 Size 属性，否则将会产生错误。

例如，在下面代码中为 Command 对象实例创建了一个新的 Parameter 对象，并为该对象赋值，然后把该对象绑定到 Command 对象实例上。由于代码中使用了 ADO 内置常量，使用前应引入 adovbs.inc 文件。

```
<!--#include File="adovbs.inc"-->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim comm
Set comm=Server.CreateObject("ADODB.Command")
Set param=comm.CreateParameter("myParameter",adVarChar,adParamInput,20,"myValue")
'使用Append方法把新参数添加到Parameters集合中，使其成为集合中的一员，这样才可以使用该参数。
comm.Parameters.append param
'获取新定义参数值
Response.Write comm.parameters.item(0).name&"<br>"
Response.Write comm.parameters.item(0).value&"<br>"
%>
```

2. Execute

该方法与 Connection 对象的 Execute 方法相似，都是负责运行指定的 SQL 命令或存储过程。其语法格式如下：

- 按行返回结果

```
Set recordset = command.Execute( RecordsAffected, Parameters, Options )
```

- 无返回结果

```
command.Execute RecordsAffected, Parameters, Options
```

在上面语法中，参数 RecordsAffected 表示提供者向其返回操作所影响的记录数目，它仅应用于操作查询或存储过程。RecordsAffected 不返回有返回结果的查询或存储过程所返回的记录数目。

Parameters 表示使用 SQL 语句传送的参数值。

Options 指示提供者如何计算 Command 对象的 CommandText 属性。该参数取值见表 5.8。

使用 Command 对象的 Execute 方法可执行在对象的 CommandText 属性中指定的查询。如果 CommandText 属性指定按行返回查询，执行所产生的任何结果都将存储在新的 Recordset 对象中。如果该命令不是按行返回查询，则提供者返回关闭的 Recordset 对象。某些应用程序语言允许忽略该返回值（如果不需要任何 Recordset）。

如果查询带有参数，将使用 Command 对象中参数的当前值，除非通过 Execute 调用传送的参数覆盖它们。可以在调用 Execute 方法时通过省略某些参数的新值来覆盖参数子集。指定参数的次序与其在方法中被传送的次序相同。例如，如果有 4 个（或更多）参数并且希望只为第 1 个和第 4 个参数传送新值，则可以将 Array(var1,,,var4) 作为 Parameters 参数传送。

例如，下面这个实例利用 Command 对象的 Execute 方法实现在产品表中插入一条记录。注意，先包含 adovbs.inc 文件，该文件存储了所有 ADO 内置的常量表。

```
<!--#include File="adovbs.inc"-->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim comm,strSQL,m
Set comm=Server.CreateObject("ADODB.Command")
comm.ActiveConnection=conn
strSQL="Insert into 产品(产品名称, 单价) values('大兴西瓜', 1.5)"
comm.CommandText=strSQL
comm.Execute m,,adCmdText

if m = 1 Then
    Response.Write("数据插入成功!")
else
    Response.Write("插入失败!")
end if
%>
```

下面实例显示了如何从产品表中用 Command 对象的 Execute 方法读取所有产品名称。

```
<!--#include File="adovbs.inc"-->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
```

```

conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Dim comm
Set comm=Server.CreateObject("ADODB.Command")
comm.ActiveConnection=conn
comm.Prepared=True
comm.CommandType=adCmdText
comm.CommandText="Select * from 产品"
Set rs=comm.Execute

While Not rs.EOF '循环读取所有产品的名称
    Response.Write(rs(1)&"<br>")
    rs.MoveNext
Wend
%>

```

5.7 RecordSet 对象

RecordSet 对象是 ADO 组件中最灵活、功能最强大的一个对象。Recordset 对象表示的是来自基本表或命令执行结果的记录全集。任何时候, Recordset 对象所指的当前记录均为集合内的单个记录。

利用 RecordSet 对象可以方便地操作数据库中的记录, 熟练地掌握和灵活运用 RecordSet 对象可以在 ASP 中完成对数据库的几乎所有操作。实际上 RecordSet 对象就是内存内临时存放的一个数据表, 该数据表是经过 SQL 查询产生的结果记录集, 它像一个二维数组, 数组的每一行表示数据库中的一个记录, 每列就是一个数据字段, 每个数据字段就表示一个 Field 对象。创建 RecordSet 对象实例的方法如下:

```
set rs=server.createobject("adodb.recordset")
```

其中 rs 变量就表示一个临时记录集。

5.7.1 RecordSet 对象的属性

RecordSet 对象的属性非常多, 灵活掌握这些属性的使用可以提高读者对数据库的应用能力。RecordSet 对象的属性详细说明见表 5.14。

表 5.14 RecordSet 对象的属性列表

属 性	说 明
AbsolutePage	指定当前记录所在的页
AbsolutePosition	指定 Recordset 对象当前记录的序号位置
ActiveConnection	指示指定的 Recordset 对象当前所属的 Connection 对象
BOF	指示当前记录位置位于 Recordset 对象的第一个记录之前
EOF	指示当前记录位置位于 Recordset 对象的最后一个记录之后
Bookmark	返回唯一标识 Recordset 对象中当前记录的书签, 或者将 Recordset 对象的当前记录设置为由有效书签所标识的记录
CacheSize	指示缓存在本地内存中的 Recordset 对象的记录数
CursorLocation	设置或返回游标服务的位置

续表

属 性	说 明
CursorType	指示在 Recordset 对象中使用的游标类型。这是一个非常重要的属性，取值见表 5.15
EditMode	指示当前记录的编辑状态
Filter	为 Recordset 中的数据指定筛选条件
Index	指示对 Recordset 对象当前生效的索引的名称
LockType	指示编辑过程中对记录使用的锁定类型。这是一个非常重要的属性，取值见表 5.16
MarshalOptions	指示要被调度返回服务器的记录
MaxRecords	指示通过查询返回 Recordset 的记录的最大数目
PageCount	指示 Recordset 对象包含的数据页数
PageSize	指示 Recordset 中一页所包含的记录数
RecordCount	指示 Recordset 对象中记录的当前数目
Sort	指定一个或多个 Recordset 以之排序的字段名，并指定按升序还是降序对字段进行排序
Source	指示 Recordset 对象中数据的来源（Command 对象、SQL 语句、表的名称或存储过程）
State	对所有可应用对象，说明其对象状态是打开或是关闭
Status	指示有关批更新或其他大量操作的当前记录的状态

表 5.15 CursorType 属性取值列表

常 数	参数值	说 明
AdOpenForwardOnly	0	仅向前游标，默认值。除了只能在记录中向前滚动外，与静态游标相同。当只需要在记录集中单向移动时，使用它可提高性能
AdOpenKeyset	1	键集游标。尽管从自己的记录集不能访问其他用户删除的记录，但除无法查看其他用户添加的记录外，键集游标与动态游标相似。仍然可以看见其他用户更改的数据
AdOpenDynamic	2	动态游标。可以看见其他用户所作的添加、更改和删除。允许在记录集中进行所有类型的移动，但不包括提供者不支持的书签操作
AdOpenStatic	3	静态游标。可以用来查找数据或生成报告的记录集合的静态副本。另外，对其他用户所作的添加、更改或删除不可见

表 5.16 LockType 属性取值列表

常 数	参数值	说 明
adLockReadOnly	0	默认值，只读。无法更改数据
adLockPessimistic	1	保守式记录锁定（逐条）。提供者执行必要的操作确保成功编辑记录，通常采用编辑时立即锁定数据源的记录的方式
adLockOptimistic	2	开放式记录锁定（逐条）。提供者使用开放式锁定，只在调用 Update 方法时锁定记录
adLockBatchOptimistic	3	开放式批更新。用于与立即更新模式相反的批更新模式

5.7.2 RecordSet 对象的方法

同 RecordSet 对象的属性一样，RecordSet 对象的方法也非常多。详细说明见表 5.17。

表 5.17 RecordSet 对象的方法列表

方 法	说 明
AddNew	创建可更新 Recordset 对象的新记录
Cancel	取消执行挂起的异步 Execute 或 Open 方法的调用
CancelBatch	取消挂起的批更新
CancelUpdate	取消在调用 Update 方法前对当前记录或新记录所作的任何更改
Clone	创建与现有 Recordset 对象相同的复制 Recordset 对象
Delete	删除当前记录或记录组
Move	移动 Recordset 对象中当前记录的位置
MoveFirst	将当前记录位置移动到 Recordset 中的第一个记录
MoveLast	将当前记录位置移动到 Recordset 中的最后一个记录
MoveNext	将当前记录向前移动一个记录（向 Recordset 的底部）
MovePrevious	将当前记录位置向后移动一个记录（向记录集的顶部）
NextRecordset	清除当前 Recordset 对象并通过提前执行命令序列返回下一个记录集
Open	打开游标
Requery	通过重新执行对象所基于的查询，更新 Recordset 对象中的数据
Resync	从基本数据库刷新当前 Recordset 对象中的数据
Save	将 Recordset 保存（持久）在文件中
Seek	搜索 Recordset 的索引，快速定位与指定值相匹配的行，并将当前行更改为该行
Supports	确定指定的 Recordset 对象是否支持特定类型的功能
Update	保存对 Recordset 对象的当前记录所做的所有更改
UpdateBatch	将所有挂起的批更新写入磁盘

下面介绍 RecordSet 对象的常用方法和属性。

1. 增加记录

利用 AddNew 方法增加一条记录，然后用 UpdateBatch 方法把数据写回到数据库，代码如下：

```
<%LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="adovbs.inc" -->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Set rs=Server.CreateObject("ADODB.RecordSet") '定义记录集实例
rs.Open "产品",conn,adOpenKeySet,adLockOptimistic '打开产品表

rs.AddNew '增加记录
rs("产品名称").value = "广州荔枝"
rs("单价") = 20
rs.UpdateBatch '把新增加的记录保存到数据库中

rs.close
set conn=nothing
%>
```

2. 打开记录

Open 方法是 RecordSet 对象中使用频率最高的一个方法，它的语法规则如下：

```
recordset.Open Source, ActiveConnection, CursorType, LockType, Options
```

其中参数 Source 表示计算 Command 对象的变量名、SQL 语句、表名、存储过程调用或持久 Recordset 文件名。

ActiveConnection 表示计算有效 Connection 对象变量名或字符串，包含ConnectionString 参数。

CursorType 表示打开 Recordset 时应该使用的游标类型。详细说明见表 5.15。

LockType 表示打开 Recordset 时应该使用的锁定（并发）类型。详细说明见表 5.16。

Options 用于指示提供者如何计算 Source 参数，取值见表 5.18。

表 5.18 Options 参数取值列表

常 数	说 明
adCmdText	指示提供者应该将 Source 作为命令的文本定义来计算
adCmdTable	指示 ADO 生成 SQL 查询以便从在 Source 中命名的表中返回所有行
adCmdTableDirect	指示提供者更改从在 Source 中命名的表中返回所有行
adCmdStoredProc	指示提供者应该将 Source 视为存储过程
adCmdUnknown	指示 Source 参数中的命令类型为未知
adCmdFile	指示应从在 Source 中命名的文件中恢复保留（保存的）Recordset
adAsyncExecute	指示应异步执行 Source
adAsyncFetch	指示在提取 Initial Fetch Size 属性中指定的初始数量后，应该异步提取所有剩余的行。如果所需的行尚未提取，主要的线程将被堵塞直到行重新可用
adAsyncFetchNonBlocking	指示主要线程在提取期间从未堵塞。如果所请求的行尚未提取，当前行自动移到文件末尾

例如：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="adovbs.inc" -->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Set rs=Server.CreateObject("ADODB.RecordSet") '实例化RecordSet对象
'为RecordSet对象设置必要属性
rs.Source = "产品"
rs.ActiveConnection=conn
rs.CursorType=adOpenStatic
rs.LockType=adLockreadOnly
rs.Open '利用上面的属性直接打开产品表
'显示记录集中的"产品名称"列
Do while not rs.eof
    Response.Write rs(1) &"<br>"
    rs.MoveNext
Loop

rs.close
set conn=nothing
%>
```

对于其中的记录集属性设置和打开记录集，可以全部在 Open 方法中完成，主要代码如下：

```
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open "产品",conn,dOpenStatic,adLockreadOnly,adCmdTable
```

读者甚至可以简写为如下代码,省略参数将被赋予默认值。如果读者对打开的记录集没有特殊的读写操作,不妨使用这种简单的打开方式,也可以为第1个参数指定一个SQL查询语句。

```
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open "产品",conn
```

3. 循环显示记录

当打开记录集之后,读者需要把打开的记录集显示在页面中,这里需要用到 eof、bof 属性来判断记录集当前记录指针的位置,以及是否存在记录。然后利用 MoveNext 方法移动当前记录指针到下一条记录,实现循环读取当前记录并显示出来。例如:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="adovbs.inc" -->
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open "产品",conn

if rs.eof then '如果当前记录指针指到最后一条记录下面,说明当前记录集没有记录
    Response.Write("没有记录")
else
    Do while not rs.eof '循环读取并显示记录中的数据
        Response.Write rs("产品名称") & " : " & rs("单价") & "<br>"
        rs.MoveNext '向下移动记录指针
    Loop
end if

rs.close
set conn=nothing
%>
```

4. 记录集分页显示

记录集分页显示相对要复杂多了,它涉及 PageSize 属性。下面以上述实例为基础,实现每页显示 10 条记录。代码如下:

```
<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
sql="Select * from 产品"
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1

if rs.eof then
    Response.Write("没有记录")
else
    pages=10 '定义每页显示的记录数
    rs.PageSize=pages
```

```

Do while Not rs.EOF and pages>0
    Response.Write rs("产品名称") & " : " & rs("单价") & "<br>"
    rs.MoveNext
    pages=pages-1
    rs.MoveNext
Loop
Response.Write("</table>")
end if

rs.close
set conn=nothing
%>

```

5. 记录集分页导航

上面代码仅能显示当前记录集的前 10 条记录, 后面记录无法阅读, 如果要翻阅不同页面记录, 必须使用记录集分页导航条。记录集分页导航条的实现方法有多种, 但基本方法是相同的: 用 `PageSize` 属性设置每页显示的记录数, 用 `PageCount` 属性获取记录集共有多少页, 用 `AbsolutePage` 属性设置当前显示页。然后, 用一个变量 `page` 来实现当前页数的显示和传递。实现的代码如下:

```

<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
sql="Select * from 产品"
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1

if rs.eof then
    Response.Write("没有记录")
else
    pageNum=10 '定义每页显示的记录数
    rs.PageSize=pageNum '把每页显示的记录数传递给PageSize属性
    allPages=rs.PageCount '获取当前记录集的页数
    page=Request.QueryString("page") '从URL查询字符串中获取page参数值, 即当前页的页数
    '基本排错处理
    If isEmpty(page) or Cint(page)<1 Then '如果page变量为空或者小于1, 则赋值为1
        page=1
    ElseIf Cint(page)>allPages Then '如果page变量大于记录集总页数, 则赋值为最大页数
        page=allPages
    End If
    rs.AbsolutePage=page '把当前页数 (page变量的值) 赋予给AbsolutePage属性

    Do while Not rs.EOF and pageNum>0
        Response.Write rs("产品名称") & " : " & rs("单价") & "<br>"
        rs.MoveNext
        pageNum=pageNum-1
        rs.MoveNext
    Loop
    Response.Write("</table>")
end if

rs.close
set conn=nothing
%>
<% '定义一个简单导航条
Response.Write "<a href='?page=1' >首页</a> "

```

```

Response.Write "<a href='?page=" & page-1 & "' >上一页</a> "
Response.Write "<a href='?page=" & page+1 & "' >下一页</a> "
Response.Write "<a href='?page=" & allPages & "' >尾页</a>"
%>

```

运行效果如图 5.25 所示。但是，读者会发现该导航条在显示上还存在一些问题，例如，如果当前记录为第 1 条记录时，“首页”和“上一页”超链接依然有效，这个不符合常规习惯。因此，还可以增加两个条件，判断如果当前页数为第 1 页时，使“首页”和“上一页”超链接失效；反之，如果当前页数为最后一页时，则使“下一页”和“尾页”超链接失效。实现的代码如下，显示效果如图 5.26 所示。

```

<%'定义简单条件导航条
If page = 1 Then
    Response.Write "首页 "
    Response.Write "上一页 "
Else
    Response.Write "<a href='?page=1' >首页</a> "
    Response.Write "<a href='?page=" & page-1 & "' >上一页</a> "
End If
If page = allPages Then
    Response.Write "下一页 "
    Response.Write "尾页"
Else
    Response.Write "<a href='?page=" & page+1 & "' >下一页</a> "
    Response.Write "<a href='?page=" & allPages + 1 & "' >尾页</a>"
End If
%>

```



图 5.25 记录集分页导航条



图 5.26 具有显示条件的分页导航条

5.7.3 Field 对象和 Fields 集合

RecordSet 对象有一个重要的集合 Fields，它包含 Recordset 对象的所有 Field 对象。每个 Field 对象对应于 RecordSet 中的一列，多个 Field 对象可以构成 Fields 数据集合。使用 Field 对象的 Value 属性可以设置或返回当前记录的数据，使用 Name 属性可以设置或返回当前记录的字段名称。

在 Fields 数据集合中，利用 Count 属性可以获取该集合中 Field 对象的数目，利用 Refresh 方法可以刷新记录，重新载入 Fields 数据集合中所包含的 Field 对象。

利用 Item 方法结合 Field 对象的索引值 (Index) 可以得到某一个 Field 对象，这个索引值一般介于 0 到 Count-1 之间。例如：

```

<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
sql="Select * from 产品"
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1

'读取记录集中的记录
Response.Write rs("产品名称") & "<br>" '法1: 直接引用字段名
Response.Write rs(1) & "<br>" '法2: 直接引用记录集的索引值
Response.Write rs.Fields("产品名称") & "<br>" '法3: 引用记录集的数据集合, 并把字段名赋予集合
Response.Write rs.Fields(1) & "<br>" '法4: 引用记录集的数据集合, 并引用集合的索引值
Response.Write rs.Fields.Item("产品名称") & "<br>" '法5: 引用记录集的数据集合的Item方法, 并把字段名赋予方法
Response.Write rs.Fields.Item(1) & "<br>" '法6: 引用记录集的数据集合的Item方法, 并引用集合的索引值

%>

```

Field 对象包含实用的属性, 详细说明见表 5.19。

表 5.19 Field 对象的属性列表

属 性	说 明
ActualSize	指示字段的值的实际长度
Attributes	指示对象的一项或多项特性
DefinedSize	指示 Field 对象所定义的大小
Name	指示对象的名称
NumericScale	指出 Parameter 或 Field 对象中数字值的范围
OriginalValue	指示发生任何更改前已在记录中存在的 Field 的值
Precision	指示在 Parameter 对象中数字值或数字 Field 对象的精度
Type	指示 Parameter、Field 或 Property 对象的操作类型或数据类型
UnderlyingValue	指示数据库中 Field 对象的当前值
Value	指示赋给 Field、Parameter 或 Property 对象的值

下面看一个综合示例, 该示例显示 Field 对象的所有属性及其取值。代码如下, 显示效果如图 5.27 所示。

```

<%
Dim conn
Set conn=Server.CreateObject("ADODB.Connection") '定义Connection对象实例
conn.Provider = "Microsoft.Jet.OLEDB.4.0" '设置Connection对象的驱动程序
conn.Open Server.MapPath("data/NorthWind.mdb") '打开数据库
%>
<%
sql="Select * from 产品"
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,3
%>
<%
'显示记录集中Field对象的各个属性及其值
Dim fds
Set fds=rs.Fields
Response.Write("<center>当前的RecordSet对象有"&fds.Count&"个字段。<br>")
Response.Write("<table border=3><tr>")

```

```

Response.Write("<td>Name属性</td>")
Response.Write("<td>Value属性</td>")
Response.Write("<td>Type属性</td>")
Response.Write("<td>Attributes属性</td>")
Response.Write("<td>NumericScale属性</td>")
Response.Write("<td>Precision属性</td>")
Response.Write("<td>ActualSize属性</td>")
Response.Write("<td>DefinedSize属性</td>")
Response.Write("<td>OriginalValue属性</td>")
Response.Write("</tr>")
For i=0 To fds.Count-1
    Response.Write("<tr><td>"&fds(i).Name&"</td>")
    Response.Write("<td>"&fds(i).Value&"</td>")
    Response.Write("<td>"&fds(i).Type&"</td>")
    Response.Write("<td>"&fds(i).Attributes&"</td>")
    Response.Write("<td>"&fds(i).NumericScale&"</td>")
    Response.Write("<td>"&fds(i).Precision&"</td>")
    Response.Write("<td>"&fds(i).ActualSize&"</td>")
    Response.Write("<td>"&fds(i).DefinedSize&"</td>")
    Response.Write("<td>"&fds(i).OriginalValue&"</td>")
    Response.Write("</tr>")
Next

rs.close
set conn=nothing
%>

```

Name	Value	Type	Attributes	NumericScale	Precision	ActualSize	DefinedSize	OriginalValue
产品ID	1	3	90	255	10	4	4	1
产品名称	苹果汁	202	102	255	255	6	40	苹果汁
供应商ID	1	3	118	255	10	4	4	1
类别ID	1	3	118	255	10	4	4	1
单位	每箱24	202	102	255	255	10	20	每箱24瓶
数量	18	6	118	255	19	8	8	18
单价	39	2	118	255	5	2	2	39
库存量	0	2	118	255	5	2	2	0
订购量	10	2	118	255	5	2	2	10
再订购量	10	2	118	255	5	2	2	10
中止	True	11	86	255	255	2	2	True

图 5.27 显示 Field 对象的属性

5.8 实践总结

SQL 语言的主要功能是实现数据库查询和操作。同时数据库连接是动态网页的基础，如果不能建立与数据库的连接，后面实例学习就无法进行。同时，要注意不同类型的数据库的连接方法有些区别，主要表现在参数设置上。刚开始学习时，用户可先从最简单的 Access 入手，然后不断深入。

本章重点介绍了数据库的基本概念、创建数据库连接，以及操作数据库的基本方法。通过本章学习，相信用户完全可以掌握数据库，从而真正实现动态数据页面显示。

用户注册模块

模块化是 Web 应用程序开发的一种方式,这种方式能够提高工作效率,缩短开发周期。网站模块化也是网络开发工程师们一致认可的标准。从本章开始,将讲解常用网站模块的开发过程,帮助读者快速构建实用的网站和 Web 应用项目。

用户注册和登录犹如居家的门锁,有了锁才能防止无证明身份的人混入。注册相当于用户配钥匙的行为,而登录相当于使用钥匙打开家门的行为。看似通俗易懂的比喻,但对于网站开发来说就非常重要了,它是网站建设的第一步。Dreamweaver CS5 提供了实现该项功能的可视化操作方法,只需要简单的两步就可以帮助你的网站设计了一道安全门。

技术要点

- 客户端行为验证
- 自定义代码验证
- 插入记录
- 服务器端检查新用户
- 自动发送邮件确认信息
- 用户密码加密

6.1 模块设计分析

动态网页的核心功能就是要实现数据的动态交流,为了了解交流对象的基本信息,用户注册就成为很多网站要求网友进入的第一步操作。当然,为了稳定站点流量,或者出于其他目的,也可以设置用户注册后才可以成为网站会员,才能访问网站内部资源或服务。

要实现用户注册功能,读者应该学习和掌握下面技术。

- 用户数据表的设计。用户表用来存储用户信息,字段设置可根据需要酌情增设。但必须要有“用户名”和“密码”两项字段。
- 根据数据表来设计注册表单,同时要设计注册成功和失败页面,以便提供更人性化的服务。
- 要保证注册者输入信息的合法性,还需要对用户输入信息进行验证。
- 要把注册者输入的信息提交到服务器端,并保存到数据库中。
- 要保证注册的信息是否与数据库中注册用户重复,应该检查新用户是否出现重复。
- 能够在用户注册成功之后,把该用户的注册信息发送到用户的邮箱中。
- 把用户输入的密码进行加密。



提示

学习本实例之前，读者应该准备的工作：

在【Internet 信息服务】窗口中建立一个虚拟目录，如 mysite（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】为“本地/网络”（详细说明可以参阅第 2.5.2 节介绍）。

把本书光盘中的“6”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构，数据库结构，并帮助读者建立了数据库连接。

6.2 设计和连接数据库

数据库的结构设计并没有十分严格的要求，由于项目的不同，也不可能有固定的模式和结构，但是设计不合理的数据库会造成很多功能难以实现，甚至无法实现。所以，即使是一个简单的功能，读者也应该好好规划。


6.2.1 设计数据库

当用户注册成功后，用户信息将被提交到服务器端，存储在数据表中，并以记录的形式保存。根据设计分析，本实例建立一个名为 users 的 Access 数据库，保存在 database 文件夹中。并新增一个 users 数据表，各字段设计见表 6.1。

表 6.1 users 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
id	自动编号	-	-	-	自动编号
name	文本	30	是	否	用户名
pass	文本	16	是	否	密码
email	文本	50	是	否	邮箱
sex	是/否	-	是	否	性别

建立数据库的具体操作步骤如下。

- 01 启动 Microsoft Access 2007，在主窗口左上侧单击“Office”按钮，在弹出的下拉菜单中选择【新建】选项，然后在打开的【新建】面板中选择并创建空白数据库，如图 6.1 所示。保存数据库为 users.mdb，路径为本地站点下的 mysite/database 文件夹。
- 02 在新建的 users.mdb 数据库窗口中，单击【使用设计器创建表】选项，打开表设计器，在【字段名称】中分别输入表 6.2 中的字段名称，然后在每个字段的【数据类型】下拉列表框中选择该字段的数据类型。在下面的【常规】选项卡中设置字段大小、格式、必填字段、是否允许空字符串等属性。
- 03 按照表 6.1 列出的字段名称和属性，设置全部字段，然后保存表为 user，具体设置如图 6.2 所示。

注册信息的主要表单域项在数据表中都有某个字段与其对应，并且该字段的属性限制了用户输入数据的格式和长度等。一旦输入的注册信息与这些字段属性不符，数据库就不会接受此信息，而显示出错信息。



图 6.1 Microsoft Access 2007 窗口



图 6.2 设置字段属性



提示

name 字段的【字段大小】定义为“30”，就是规定了用户输入的用户名不能超过 30 个字符，【必填字段】选择为“是”，规定此字段为必填项，【允许空字符串】选择为“否”，说明该项不允许使用 null。


- 04** 关于用户名与密码字段的设计，为了节约数据库空间，可根据实际需要限制字段大小，对于密码字段需要进行加密，即设置【输入掩码】属性。单击【输入掩码】栏右边的  按钮，在打开的【输入掩码向导】对话框中进行设置，如图 6.3 所示。这样，在数据表中密码就全部以“*”来表示，避免被别人看见，如图 6.4 所示。



图 6.3 【输入掩码向导】对话框

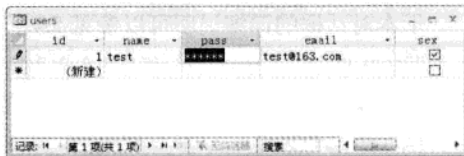


图 6.4 掩码数据显示效果

- 05** 选择 User 字段，然后右击鼠标，在弹出的菜单中选择【主键】命令，将该字段设置为主键。

6.2.2 连接数据库

设计好数据库之后，读者就可以在本地站点内建立 ASP 与数据库的连接。具体操作步骤如下。

- 01** 启动 Adobe Dreamweaver CS5，在【文件】面板中打开本实例站点，如 mysite。选择【窗口】|【数据库】菜单命令，打开【数据库】面板。如果在 Dreamweaver CS5 中没有打开网页文档，则【数据库】面板显示图 6.5 所示效果；如果打开的页面是静态页面，则【数据库】面板会显示图 6.6 所示效果。不过读者只要单击“文档类型”和“测试服务器”文本超链接，系统会自动刷新站点，如果已经建立了动态站点，则会在选项前面显示对

号标记✓；如果在动态站点内打开一个.asp 文件，则【数据库】面板显示图 6.7 所示效果。

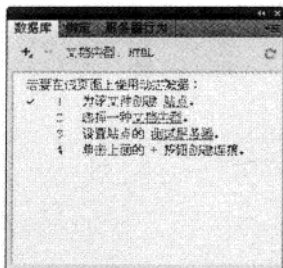


图 6.5 【数据库】面板无效

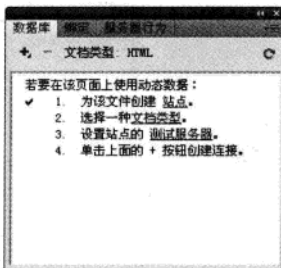


图 6.6 【数据库】面板条件不够

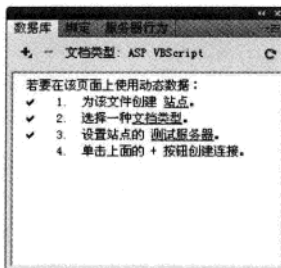


图 6.7 【数据库】面板可以连接

02 单击【数据库】面板上的【加号】按钮 \oplus ，从弹出菜单中选择【自定义连接字符串】选项，如图 6.8 所示。

03 打开【自定义连接字符串】对话框，如图 6.9 所示，在【连接名称】文本框中输入新连接的名称，如 conn，它相当于 Connection 对象的一个实例名称。

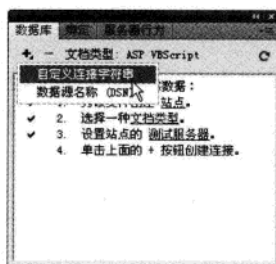


图 6.8 选择【自定义连接字符串】项

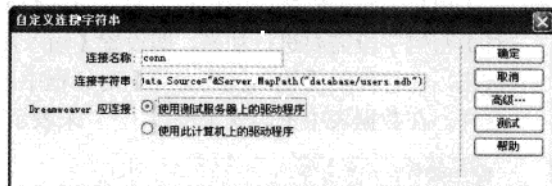


图 6.9 【自定义连接字符串】对话框

04 在【连接字符串】文本框中输入自定义连接字符串。详细的自定义连接字符串如下（关于本对话框的详细说明可以参见 5.4.3 节介绍）：

"Provider=Microsoft.Jet.OLEDB.4.0;Data Source="&Server.MapPath("database/users.mdb")

05 选择【使用测试服务器上驱动程序】复选框，单击【测试】按钮，Dreamweaver CS5 尝试连接到数据库。如果连接成功，则会显示如图 6.10 所示的提示对话框。

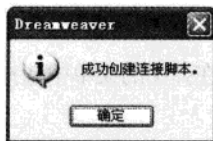


图 6.10 数据库连接测试和连接成功提示信息



提示

由于使用 DSN 建立的数据库被上传到远程服务器之后，会无法工作。因此不建议读者使用 DSN 来连接数据库。在后面实例中将全部采用自定义连接字符串的方法建立与数据库的连接。

Dreamweaver CS5 在建立数据库连接时，会在站点根目录下生成一个临时目录_mmServerScripts，在该目录中默认有 MMHTTPDB.asp 和 MMHTTPDB.js 两个文件。注意，该文件夹

在【文件】面板中是看不见的，读者可以在【资源管理器】窗口中的本地远程目录中查看。此时，使用自定义连接字符串进行数据库连接时，如果读者进行测试，则会显示图 6.11 所示提示对话框。

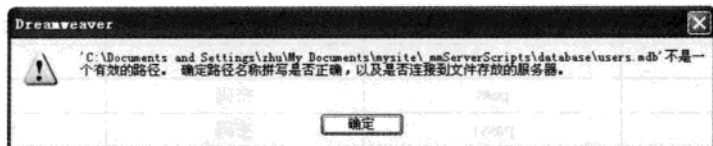


图 6.11 数据库连接错误提示信息

这些错误提示信息并不是说上面的自定义连接字符串出现了错误，只是说目前在 Dreamweaver CS5 中无法使用数据库连接，因此，Dreamweaver CS5 的所有服务器行为操作也就无法进行了。

解决的方法：假设 _mmServerScripts 为站点根目录，把 database 文件夹以及其内部的数据库文件 user.mdb 全部复制到 _mmServerScripts 目录中。由于 _mmServerScripts 目录是一个临时文件夹，运行站点文件时它是不起作用的，当上传站点到远程服务器上时，完全可以删除该临时目录及其内部文件。

6.3 设计交互表单

本实例主要包括 3 个文件，包括 index.asp、fail.asp、ok.asp。

- index.asp 主要是注册表单和注册功能实现。
- fail.asp 是注册失败提示页面。
- ok.asp 是注册成功提示页面。

关于前台 3 个文件的设计，本书已提供整个实例模板。在本章“实例模板”文件夹中，用户可以把它们复制到本地站点文件夹中为下一步工作做好准备。下面重点介绍 index.asp 文件中的表单设计。

index.asp 文件是一个表单页面，本页表单主要包括 4 个文本框、1 个单选按钮和 1 个提交按钮，设计如图 6.12 所示。下表列出本页面表单中 4 个文本框的属性设置，见表 6.2。

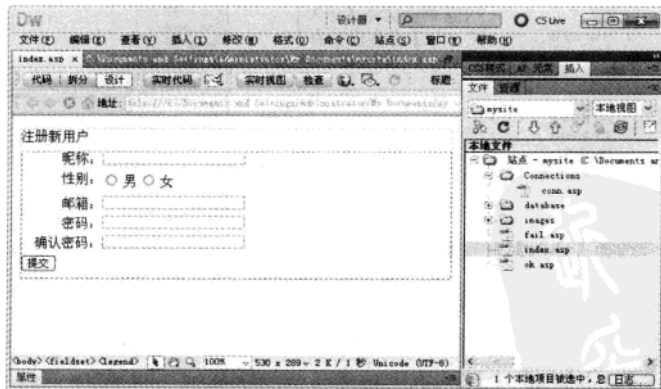


图 6.12 注册表单设计效果

在本表单中还包含两个单选按钮，分别命名为 sex，【选定值】分别设置为 1 和 0。最后，插入一个提交按钮，设置【值】为“提交”。

表 6.2 index.asp 中文本框对象设置


对 象	名 称	类 型	备 注
文本框	name	单行	用户名
文本框	pass	密码	密码
文本框	pass1	密码	确认密码
文本框	email	单行	邮编

6.4 实现基本的注册功能

Dreamweaver CS5 提供了【插入】和【检查新用户】服务器行为，读者利用这两个服务器行为可以快速完成用户注册基本功能。

6.4.1 添加【插入】服务器行为

要把用户在表单中输入的信息保存到服务器端的数据库中，需要使用【插入】服务器行为来实现，具体步骤如下。

- 01 在 Dreamweaver CS5 中，打开 index.asp 页面，在【数据库】面板中设置好连接数据源，如果在该面板中没有任何数据源显示，需要先定义站点数据源。
- 02 在菜单中选择【窗口】|【服务器行为】命令，打开【服务器行为】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【插入】菜单命令，如图 6.13 所示。
- 03 打开【插入】对话框，设置如图 6.14 所示。在【连接】下拉列表中选择已定义好的数据源 conn；在【插入到表格】下拉列表中选择要插入的数据表 users；在【插入后，转到】文本框中设置插入记录成功后要跳转到 ok.asp 页面。

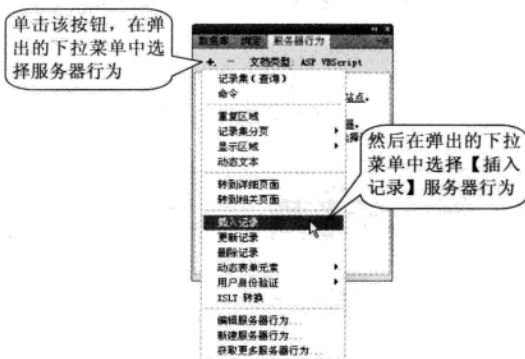


图 6.13 选择【插入】服务器行为



图 6.14 设置【插入】对话框

- 04 在【表单元素】列表框中分别设置每个表单对象中输入数据要插入数据库中 users 表中哪个字段。方法是在【表单元素】列表框中选中一个选项（即表单中的指定表单域，如文本框），在下面的【列】下拉列表中选择对应字段选项（即 users 数据库的 users 表中对应字段名）；在【提交为】下拉列表中选择提交数据的类型，可根据字段类型进行设置。

- 05 设置完毕，单击【确定】按钮，在【服务器行为】面板中显示该行为，如图 6.15 所示。

如果用户使用字段名称与 SQL 保留字发生冲突，应该使用中括号把字段名括起来，否则会显示“Microsoft JET Database Engine (0x80040E14) INSERT INTO 语句的语法错误”错误信息。例如，在 index.asp 页面中，切换到【代码】视图，找到下面代码：

```
MM_editCmd.CommandText = "INSERT INTO users ([name], sex, email, pass) VALUES (?, ?, ?, ?)"
```

在 SQL 字符串中，Dreamweaver CS5 会自动把 user、key 等关键字用中括号括起来了。但是，name 字段名也是一个 SQL 预留字，因此执行上面 SQL 时会显示错误信息。读者可以把 name 用中括号括起来，修改后的代码如下所示。




图 6.15 插入的服务器行为



在【服务器行为】面板中双击该服务器行为，可以再次打开【插入】对话框对新建立的服务器行为进行重新编辑。

6.4.2 添加【检查新用户】服务器行为

为了避免读者输入的用户名与数据库中的记录重复，需要使用【检查新用户名】服务器行为来验证读者输入的用户名是否与数据库中的用户名冲突，具体步骤如下。

- 01 单击【服务器行为】面板左上角的  按钮，在弹出的下拉菜单中选择【用户身份验证】|【检查新用户名】菜单命令，如图 6.16 所示。
- 02 打开【检查新用户名】对话框，设置如图 6.17 所示。在【用户名字段】下拉列表中选择要检查的字段 name，在【如果已存在，则转到】文本框中设置如果已经存在该用户名，则要跳转到 fail.asp 页面。
- 03 设置完毕，单击【确定】按钮，此时在【服务器行为】面板中显示【检查新用户名】行为，如图 6.18 所示。

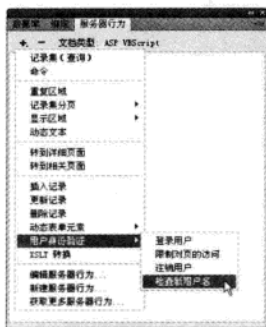


图 6.16 选择【检查新用户
名】服务器行为

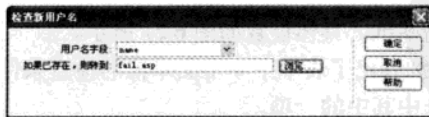


图 6.17 【检查新用户名】对话框



图 6.18 插入【检查新用户
用户名】服务器行为

6.5 表单验证

经过上节操作就可以实现基本注册功能。但有些问题还需要解决，例如，读者在登录时，


可能输入的登录密码和确认密码不一致,或者输入的邮箱地址非法等。因此,还必须验证用户输入信息的合法性,避免非法信息的存入。

表单验证一般可分为客户端验证和服务器端验证两种。客户端验证指的是在客户端的浏览器中运行有关验证的代码,服务器端验证指的是在服务器端运行有关验证的代码。客户端验证一般使用 Dreamweaver CS5 的行为来实现,该行为产生的全部是客户端的 JavaScript 代码;服务器端的验证一般使用 Dreamweaver CS5 的服务器行为来实现。一般情况下, Dreamweaver CS5 的服务器行为产生的都是服务器端代码。

显然,在服务器端验证需要把用户待验证的数据传到服务器上,然后服务器把验证结果再传回浏览器,这样会浪费时间。所以除非有特殊需要,或者考虑到安全原因,一般均采用客户端代码验证。

6.5.1 注册信息验证

关于验证用户输入信息的合法性,由于不涉及安全或隐私,完全可以放在客户端完成。这里利用 Dreamweaver CS5 的行为来解决,具体操作步骤如下。

- 01 在 Dreamweaver CS5 中,打开 index.asp 页面,准确选中表单。方法是将光标置于表单内任何位置,在 Dreamweaver 编辑窗口底部的标签选择栏中单击选中<form>标记即可,如图 6.19 所示。
- 02 在菜单中选择【窗口】|【行为】命令,打开【行为】面板。单击面板左上角的  按钮,在弹出的下拉菜单中选择【检查表单】菜单命令,如图 6.20 所示。

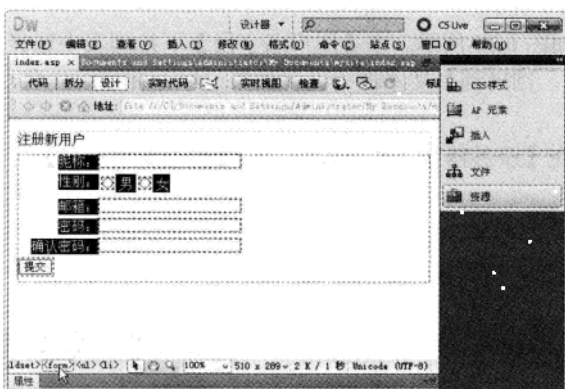


图 6.19 选中整个表单

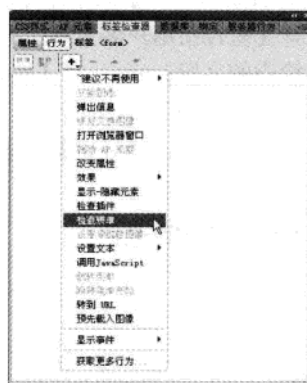


图 6.20 选择【检查表单】行为

- 03 打开【检查表单】对话框,具体设置如图 6.21 所示。
在【域】列表框中详细列出了表单中 4 个文本框的检查条件。设置检查表单的步骤如下。
 - 01 在【域】列表框中选中其中的一项。
 - 02 在列表框下面的【值】选项右边的【必需的】复选框中,设置该表单对象是不是必须填写,勾选该复选框即可。
 - 03 在【可接受】选项区右边的单选按钮组中选择设置输入值的范围。
 - 【任何东西】单选按钮设置输入内容为任意字符。
 - 【数字】单选按钮设置输入内容为数字。
 - 【电子邮件地址】单选按钮设置输入内容为 E-mail 格式。

- 【数字从】单选按钮则设置输入内容为某个区间数字。

本例中除了与输入 E-mail 地址相关的选项应选中【电子邮件地址】单选按钮之外，其余的都应选中【任何东西】单选按钮，同时要全部勾选【必需的】复选框。

04 单击【确定】按钮保存设置。设置完毕后，返回【行为】面板，并在事件选项栏中设置 onsubmit 事件选项，表示当用户单击【提交】按钮时，会触发该事件。方法是单击选项栏右边的三角按钮，在弹出的下拉列表框中选择 onsubmit 事件选项即可，如图 6.22 所示。

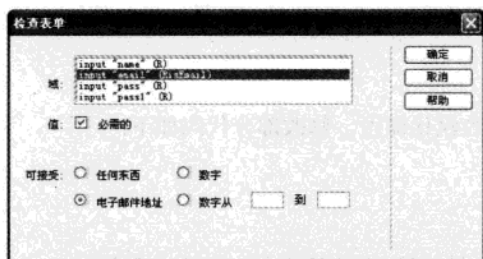


图 6.21 【检查表单】对话框



图 6.22 设置检查表单触发事件

6.5.2 完善信息验证

增加【检查表单】行为之后，对该表单进行信息输入验证。发现该行为存在两个漏洞。

- 错误提示对话框全部以英文显示，不符合中文用户浏览阅读，如图 6.23 所示。
- 当登录密码和确认密码不一致时，不能给出错误提示信息。

要解决上面两个漏洞，先切换到【代码】视图，找到插入的【检查表单】行为代码（一般位于</html>结束标记的上面）。代码如下：

```
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
    if (document.getElementById){
        var i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
        for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=document.getElementById
(args[i]);
            if (val) { nm=val.name; if ((val=val.value)!="") {
                if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
                    if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an e-mail address.\n';
                } else if (test!='R') { num = parseFloat(val);
                    if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
                    if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
                        min=test.substring(8,p); max=test.substring(p+1);
                        if (num<min || max<num) errors+='- '+nm+' must contain a number between '+min+'
                        and '+max+'.\n';
                    } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is required.\n'; }
            }
        }
    }
}
```

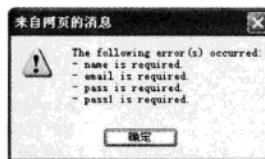


图 6.23 错误提示信息

```

    } if (errors) alert('The following error(s) occurred:\n'+errors);
    document.MM_returnValue = (errors == '');
  } }
  //-->
</script>

```

该段代码利用 JavaScript 函数实现对输入数据进行验证, 并给出错误提示。MM_validateForm() 函数被表单中的 onSubmit 事件调用, 判断输入信息的合法性, 并给出错误提示。它实现的功能就是遍历表单中的每个表单域, 检测表单域中的值是否与事先设置的条件相符合。当判断条件不成立时, 把错误信息增加到 errors 变量中, 最后再全部显示 errors 变量内容。

1. 关于英文提示问题

解决英文提示问题, 可以把 errors 变量内容汉化即可, 修改部分代码如下:

```

<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=document.getElementById(
(args[i]);
      if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
          if (p<1 || p==(val.length-1)) errors+='- '+nm+' 邮件格式输入不正确! \n';
        } else if (test!='R') { num = parseFloat(val);
          if (isNaN(val)) errors+='- '+nm+' 应该输入一个数字! \n';
          if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
            if (num<min || max<num) errors+='- '+nm+' 必须填入 '+min+' 和 '+max+' 之间的数据! \n';
          } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' 必须填! \n'; }
      } if (errors) alert('错误提示:\n\n'+errors);
      document.MM_returnValue = (errors == '');
    } }
  //-->
</script>

```

2. 关于密码输入不一致问题

可以在上面表单检查行为代码段中增加 2 行代码, 来验证密码输入, 代码插入位置和增加代码如下:

```

<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2); i+=3) { test=args[i+2];
val=document.getElementById(args[i]);
      if (val) { nm=val.name; if ((val=val.value)!="") {
        if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
          if (p<1 || p==(val.length-1)) errors+='- '+nm+' 邮件格式输入不正确! \n';
        } else if (test!='R') { num = parseFloat(val);
          if (isNaN(val)) errors+='- '+nm+' 应该输入一个数字! \n';
          if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
            min=test.substring(8,p); max=test.substring(p+1);
            if (num<min || max<num) errors+='- '+nm+' 必须填入 '+min+' 和 '+max+' 之间的数据! '+max+'.\n';
          } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' 必须填! \n'; }
      }
    }
  }
}

```

```

//检查两次密码是否相同
if(document.getElementById("pass").value != document.getElementById("pass1").value)
{
    errors+="- 登录密码和确认密码不一致! ";
}
if (errors) alert('错误提示:\n\n'+errors);
document.MM_returnValue = (errors == '');
} }
//-->
</script>

```

通过上面的代码修改和完善,再次注册,提示信息如图 6.24 所示。

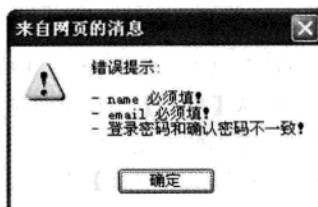



图 6.24 汉化后的错误提示信息

6.6 Spry 表单验证

从 Dreamweaver CS3 版本开始, Dreamweaver 新增加了强大的 Spry 框架,利用该框架可以实现 Ajax 动态数据显示功能。Spry 包含了强大的表单验证工具,利用这些工具可以实现更人性化的表单验证功能。

下面以本实例的表单验证为例,介绍 Spry 表单验证的实现方法。为了避免与上节介绍的 Dreamweaver 行为发生冲突,建议用户先在【行为】面板中删除已添加的表单验证行为。然后再添加 Spry 表单验证行为,具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中,打开 index.asp 页面,选中“昵称”文本框。选择【插入】|【Spry】|【Spry 验证文本域】命令,或者在【插入】面板中选择【Spry】选项,然后在其中单击【Spry 验证文本域】按钮,如图 6.25 所示。

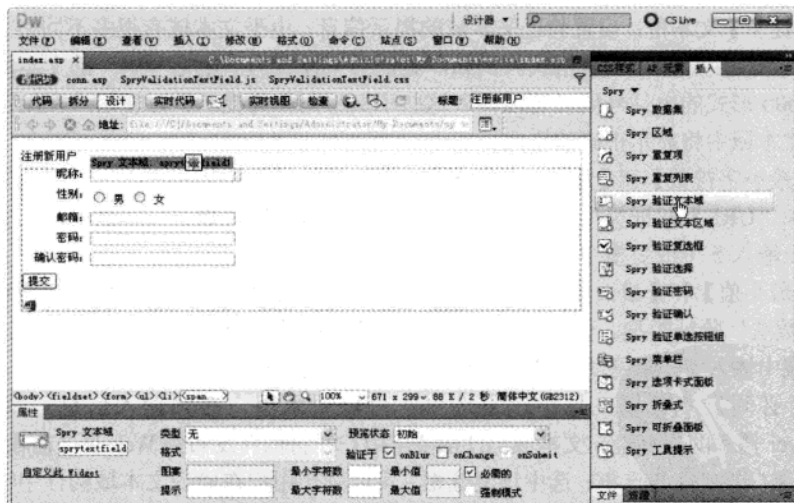


图 6.25 插入 Spry 验证文本域

- 02** 在【属性】面板中可以详细设置验证文本域的相关验证选项。说明如下。

(1) 【类型】下拉列表:在该菜单中指定验证类型,用户可以为验证文本域构件指定不同的验证类型。例如,如果文本域将接收电子邮件地址,则可以指定电子邮件地址验证类型。

(2) 【格式】下拉列表:当选择一种格式类型之后,大多数验证类型都会使文本域要求采

用标准格式。例如，如果用户向文本域应用整数验证类型，那么，除非用户在该文本域中输入数字，否则该文本域构件将无法通过验证。但是，某些验证类型允许选择文本域将接受的格式种类。下面详细说明验证文本域使用的验证类型和格式。

①无：无须特殊格式。②整数：文本域仅接受数字。③电子邮件：验证文本域接受包含@和句点(.)的电子邮件地址，而且@和句点的前面和后面都必须至少有一个字母。④日期：格式可变。可以从【格式】下拉菜单中选择日期的格式。⑤时间：格式可变。可以从【格式】下拉菜单中进行选择，其中“tt”表示 am/pm 格式，“t”表示 a/p 格式。⑥信用卡：格式可变。可以从【格式】下拉菜单中进行选择。可以选择接受所有信用卡，或者指定特定种类的信用卡，文本域不接受包含空格的信用卡号，例如，4321 3456 4567 4567。⑦邮政编码：格式可变。可以从【格式】下拉菜单中进行选择。⑧电话号码：文本域接受美国和加拿大格式，例如，(000)000-0000 或自定义格式的电话号码。⑨社会安全号码：文本域接受 000-00-0000 格式的社会安全号码。⑩货币：文本域接受 1,000,000.00 或 1.000.000,00 格式的货币。⑪实数/科学记数法：验证各种数字，例如，数字（如 3）、浮点值（如 12.123）、以科学记数法表示的浮点值（如 1.212e+12，其中 e 用做 10 的幂）。⑫IP 地址：格式可变。可以从【格式】下拉菜单中进行选择。⑬URL：文本域接受 http://xxx.xxx.xxx 或 ftp://xxx.xxx.xxx 格式的 URL。⑭自定义：可用于指定自定义验证类型和格式。

(3)【预览状态】下拉列表框：设置要查看的状态。例如，如果要查看处于“有效”状态的构件，请选择“有效”。

(4)【验证于】选项：设置验证何时发生。可以选择所有选项，也可以都不选。

①onBlur：当用户在文本域的外部单击时验证。②onChange：当用户更改文本域中的文本时验证。③onSubmit：当用户提交表单时验证。

(5)【提示】文本框：设置验证文本域的提示信息。由于文本域有很多不同格式，因此提示信息可以帮助用户输入正确格式。例如，验证类型设置为“电话号码”的文本域将只接受(000) 00-0000 形式的电话号码。可以输入这些示例号码作为提示，以便用户在浏览器中加载页面时，文本域中将显示正确的格式。

(6)【最小字符数】和【最大字符数】文本框：该选项仅适用于“无”、“整数”、“电子邮件地址”和“URL”验证类型。例如，如果用户在【最小字符数】文本框中输入 5，那么，只有当用户输入 5 个或更多个字符时，该构件才通过验证。

(7)【最小值】和【最大值】文本框：该选项仅适用于“整数”、“时间”、“货币”和“实数/科学记数法”验证类型。例如，如果用户在【最小值】文本框中输入 5，那么，只有当用户在文本域中输入 5 或者更大的值时，该构件才能够通过验证。

(8)【必须】复选框：选中该复选框，可以要求用户必须输入信息。默认情况下 Dreamweaver 插入的所有验证文本域构件都要求用户在将构件发布到 Web 页之前输入内容。

(9)【强制模式】复选框：选中该复选框，可以禁止用户在验证文本域构件中输入无效字符。例如，如果对具有“整数”验证类型的构件集选择此选项，那么，当用户尝试键入字母时，文本域中将不显示任何内容。

03 分别为不同文本框插入 Spry 验证，如图 6.26 所示。插入并设置 Spry 构件之后保存 index.asp 页面时，Dreamweaver CS5 会在同目录下新建 SpryAssets 文件夹，在其中保存 SpryValidationTextField.css 和 SpryValidationTextField.js 系统文件，该文件应该同页面一同上传到远程服务器上。

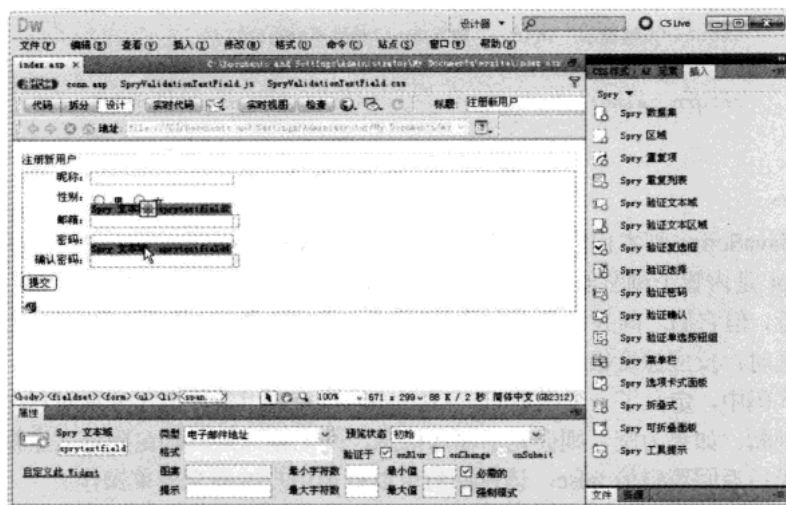


图 6.26 插入 Spry 验证文本域后的表单页面

6.7 自定义 JavaScript 脚本验证

不管是用 Dreamweaver CS5 的【检查表单】行为，还是使用 Spry 验证表单，它们都是使用客户端 JavaScript 脚本来实现的。如果读者了解 JavaScript 脚本的简单写法，使用自定义 JavaScript 脚本验证表单将是件很轻松的事情，而且自定义 JavaScript 脚本验证表单功能更贴切、操作更灵活。建议读者在条件允许的情况下，自己编写 JavaScript 脚本来验证表单。

例如，打开 index.asp 页面，切换到【代码】视图，在代码中任何位置输入下面 JavaScript 脚本代码：

```
<script type="text/javascript">
function checkform() {
    var $ = document.getElementById;
    if($("#name").value == "") {
        alert("友情提示：\n\n  请输入用户名！");
        $("#name").focus();
        return false;
    }
    if($("#pass").value == "") {
        alert("友情提示：\n\n  请输入密码！");
        $("#pass").focus();
        return false;
    }
    if($("#pass").value != $("#pass1").value) {
        alert("友情提示：\n\n  两次输入的密码不一致，请重新输入！");
        $("#pass").focus();
        return false;
    }
    var reg = /^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]{2,3}){1,2}$/;
    var flag = reg.test($("#email").value);
```

```

        if(!flag){
            alert("友情提示: \n\n    输入Email地址不合法!");
            $("email").focus();
            return false;
        }
    }
</script>

```

客户端 JavaScript 脚本代码一般都嵌入在<script type="text/javascript">和</script>标记之间。JavaScript 是内置于浏览器中的脚本语言,读者可能会觉得它与 Java 很相近,虽然它们的名字很相近,但它们之间并没有太大的关系。JavaScript 通过浏览器解析并执行的,所以执行起来没有延时,反应速度很快。

在上面代码中,定义了一个函数 checkform(),该函数用条件语句分别对每个文本框中的输入值进行判断。如果为空,则会提示输入信息,并把输入焦点确定到该文本框中,以方便用户输入,最后返回逻辑值 false,该返回值可以取消用户的提交表单操作。

在电子邮件文本框验证时,脚本中使用了正则表达式来验证输入文本是否符合 E-mail 地址的要求。即是否包含@和点号(.)。其中 var reg = /^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]{2,3}){1,2}\$/;定义一个变量 reg,并把一个 E-mail 正则表达式赋值给它。然后调用正则表达式的方法 test 来验证用户输入的字符串是否匹配正则表达式的模式。如果匹配,则返回一个逻辑值 true,并把该值赋值给变量 flag;如果不匹配,则返回一个逻辑值 false,并把该值赋值给变量 flag。最后用一个条件语句来判断 flag 变量的逻辑值,然后这个条件不为真,则提示用户输入正确的 E-mail 地址格式。



提示

JavaScript 脚本与 VBScript 脚本在语法上存在很大的区别。JavaScript 语言是严格区别大小的,且每句结尾要加分号(;). JavaScript 语言结构体也与 VBScript 不同,JavaScript 是用大括号({})来表示结构体的开始和结束。

由于正则表达式实现容易,格式简单,而且功能强大,很多编程语言都采用正则表达式来验证用户输入格式是否有效。JavaScript 语言的正则表达式内容位于“/”定界符之间。用户只要把希望查找匹配对象的模式内容放入“/”定界符之间即可。为了能够使用户更加灵活地定制字符模式内容,正则表达式提供了专门的“元字符”。所谓元字符,就是指那些在正则表达式中具有特殊意义的专用字符,可以用来规定其前导字符(即位于元字符前面的字符)在目标对象中的出现模式。例如,[a-zA-Z0-9_-]表示任意个 a~z 和 A~Z 之间的字母与 0~9 之间数字的组合。“\.”用来验证其中是否存在一个点符号。通过上面的组合就可以得到了“字母数字组合+@+字母数字组合+.+字母数字组合”的 E-mail 验证模式,该模式可以有效地验证用户输入的 E-mail 格式是否正确(后面实例中将介绍更多有关正则表达式的学习内容和参考信息)。

最后,读者只需要把该函数加入到<form>标记中的 onSubmit 事件属性中即可,代码如下:

```

<form ACTION="<%=MM_editAction%>" METHOD="POST" name="reg_frm" id="reg_frm"
onSubmit="javascript:return checkform()" >

```

其中 onSubmit 属性事件表示表单被提交时将触发该函数内的代码。javascript:表示后面的脚本是 JavaScript 脚本,return checkform()表示返回该函数的返回值,这样当返回 false 时,会自动取消用户的表单提交操作。

6.8 注册信息确认

为了防止用户注册的邮箱作假，一般通过发送注册信息确认邮件，以辨认注册的邮箱是否真实。但是 ASP 本身没有发送邮件的能力，如果希望用 ASP 来发送电子邮件信息，那么需要安装一个 ASP 组件，并借助这些组件来帮助实现邮件的自动发送、回复等工作。

6.8.1 使用 cdonts 组件发送确认邮件

在默认情况下，Windows XP 和 Windows 2003 版本操作系统没有附带 cdonets.dll 组件，必须手动安装。读者可以从 Windows 2000 的 system32 文件夹中得到该文件，或者从本书光盘中获得该组件文件，然后复制到 windows 的 system32 目录下，即可安装组件 cdonets.dll。

选择【开始】|【运行】命令，在【运行】对话框中输入“regsvr32 cdonets.dll”，回车即可注册该组件，注册成功 cdonets 组件后就可以在当前操作系统中使用了。

打开 index.asp 文件，切换到【代码】视图，找到“插入记录”服务器行为代码段，在 Response.Redirect(MM_editRedirectUrl)代码前输入下面代码，即当插入记录成功并准备跳转页面之前执行下面代码：

```
'发送注册确认邮件-----
    dim objCDOMail
    Set objCDOMail = Server.CreateObject("CDONTS.NewMail")
    objCDOMail.From = "zhuyinhong@css8.cn"
    objCDOMail.To = trim(Request.Form("email"))
    objCDOMail.Subject = "感谢注册本站会员"
    objCDOMail.Body = Request.Form("User") & ":感谢您注册本站会员，请记住你的用户名：" &
Request.Form("User") & "，密码：" & Request.Form("Password")
    objCDOMail.Send
    set objCDOMail = Nothing
'-----
    Response.Redirect(MM_editRedirectUrl)
End If
End If
%>
```

NewMail 对象不是建立在普通的 API 体系结构上。它能够快速产生通告邮件，而且不提供任何用户接口，在邮件的产生和发送过程中也不提供任何的交互。因此 NewMail 对象的属性没有被设计成为可以回读和检查。仅除 Version 外，其他属性只能被写入。

NewMail 对象的主要优点在于它产生和传送邮件的简单性和易用性。用户不必登录到一个会话，也不必与文件夹和邮件集合打交道。只需创建 NewMail 对象，将其发送出去，然后将其设置为 Nothing 即可。用户可以在 Send 方法的参数中提供关键信息。

6.8.2 使用 JMail 组件发送确认邮件

使用 cdonets 组件发送确认邮件虽然比较简便，但是功能还比较弱，有些类型的邮箱无法实现邮件接发。w3 JMail 邮件组件是 Dimac 公司开发的用来完成邮件的发送、接收、加密和集群传输等工作的组件。w3 JMail 组件是国际最为流行的邮件组件之一，大多数 ASP 程序员都是使用 w3 JMail 组件构建邮件收发系统。因此，下面再来介绍如何使用 w3 JMail 组件技术实现自动发送注册确认邮件。

w3 JMail 组件发送邮件速度快, 功能丰富, 并且是免费的。用户可以访问主页: <http://tech.dimac.net> 下载 w3 JMail 组件。本书光盘中也提供该组件。

与别的组件一样, 在使用 w3 JMail 邮件组件之前需要先安装和注册该组件 Jmail.dll。w3 JMail 组件安装和一般软件的安装方法相同, 不需要特别设置, 按默认设置单击 Next 按钮即可。安装该组件后即可自动注册。

w3 JMail 组件主要包括以下 15 个对象。

- POP3 对象: 主要是对 POP3 服务器进行操作。
- Messages 对象: 是与邮件集合有关的对象。
- Message 对象: 是与邮件相关的对象, 既可以是接收的邮件信息, 也可以是发送邮件的信息。
- Headers 对象: 是与邮件头有关的对象。
- Recipients 对象: 是与接收的邮件集合有关的对象。
- Recipient 对象: 是与接收邮件信息有关的对象。
- Attachments 对象: 是与邮件附件的集合有关的对象。
- Attachment 对象: 是与邮件附件有关的对象。
- MailMerge 对象: 是与邮件模板有关的对象。
- PGPKeys 对象: 是与 PGP 密钥集合有关的对象。
- PGPKeyInfo 对象: 是与 PGP 密钥信息有关的对象。
- SpeedMailer 对象: 是与快速发送邮件的操作有关的对象。
- PGPEncodeResults 对象: 是对每封邮件进行 PGP 加密操作后得到的独立的结果集合的对象。
- PGPEncodeResult 对象: 是 PGP 加密操作的结果的对象。
- PGPEncodeResultCollection 对象: 是 PGPEncodeResult 对象集合的对象。

在本例中要用到 Message 对象, 如果发送的邮件还带有附件, 则还需要使用 Attachments 对象和 Attachment 对象。有关 w3 JMail 组件各个对象的详细说明可以参考该组件的用户手册。

打开 index.asp 文件, 切换到【代码】视图, 找到“插入记录”服务器行为代码段, 在 Response.Redirect(MM_editRedirectUrl)代码前输入下面代码, 即当插入记录成功并准备跳转页面之前执行下面代码:

```
'发送注册确认邮件
'-----
dim JMail
set JMail=server.CreateObject("Jmail.message")

JMail.Silent=true
JMail.Charset = "gb2312"           '邮件字符编码
JMail.FromName = "zhuyinhong@css8.cn" '发送人姓名
JMail.from = "zhuyinhong@css8.cn"   '发送人的邮箱地址
JMail.AddRecipient Request.Form("email"), Request.Form("User")
                                   '收信人的邮箱地址和姓名
JMail.Subject = "感谢注册本站会员" '邮件标题
jmail.Body= Request.Form("User") & ":感谢您注册本站会员, 请记住你的用户名: " &
Request.Form("User") & ", 密码: " & Request.Form("Password") '邮件主要内容
JMail.MailServerUserName = "zhuyinhong" '发信人的邮箱登录用户名
```

```

JMail.MailServerPassWord = 111111      '发信人的邮箱登录密码,读者应该修改为
                                         自己邮箱系统的系统配置
JMail.MailDomain = "263.net"           '发送人的接收邮件服务器
sendok=JMail.send("smtp.263.net")      '利用send方法从发送人的发送邮件服务器
                                         发送邮件

if sendok then                          '发送结果,返回一个逻辑值,为true表示成功,否则失败
    msg="已成功发送了邮件!"
else
    msg="发送邮件失败,可能是您填写了错误的邮件地址.<br>" & _
        "如果您确信没有写错邮件地址,则可能是发信程序出了<br>" & _
        "故障,请与管理员联系"
end if

'-----
Response.Redirect(MM_editRedirectUrl)
End If
End If
%>

```

在上面代码中,用户应该根据自己的邮箱管理系统进行相应的配置,才能进行邮件发送。

06

6.9 注册信息加密

用户密码是最敏感的数据,也是用户最关心的隐私信息,如何保证该信息不被任何人(包括网站管理员)窃取是网站开发中一项重要技术。在 ASP 应用程序中一般常用 MD5 加密方法。

MD5 的全称是 Message-digest Algorithm 5 (信息-摘要算法),在 20 世纪 90 年代初由 Mit laboratory for Computer Science 和 Rsa Data Security Inc 的 ronald l. rivest 开发出来,经 MD2、MD3 和 MD4 发展而来。它的作用是让大容量信息在用数字签名软件签署私人密钥前被“压缩”成一种保密的格式(就是把一个任意长度的字节串变换成一定长的大整数)。

MD5 的最大特点就是加密的数据是不可逆的。传统的加密算法都是可逆的,如果知道了一个加密后的字段所用的算法,就可以进行解密。但是用户不能用 MD5 算法来解密被加密的数据。要验证被加密的密码,用户就只能把登录密码用 MD5 加密之后与数据库中已加密的密码进行比较,如果相同则说明输入密码正确,否则就不对。

由于 MD5 是一种复杂的数学算法,对于初学者来说没有必要去研究这种算法的实现过程,因此本书也不打算介绍该算法的具体代码。读者可以在本书光盘的实例模板中把 md5.asp 文件复制到站点内即可。然后用#include 语句包含到需要加密的文件中:

```
<!--#include file="md5.asp" -->
```

再用 md5.asp 文件定义的函数 md5()把用户输入的密码进行直接加密即可,例如,在 index.asp 文件中把用户输入的密码进行 MD5 加密:

```
MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param4", 202, 1, 16,
md5(Request.Form("pass"))) ' adVarChar
```

6.10 实践总结

本章重点介绍了用户注册功能的实现,从客户端注册验证到服务器端数据库插入,共用了 4 项技术要点。先在客户端设置信息验证,确保用户输入注册信息的合法性。当在注册页

中输入注册信息并按下【提交】按钮后，这些信息首先在客户端被验证是否合法，只有这些信息在客户端被验证合法才能被提交到服务器端验证。服务器端验证如果没有与之重复的用户名时，这些信息就会被插入到数据库中，最后打开提示用户注册成功页面。*Dreamweaver* CS5 提供的行为和服务器行为为初学者提供了方便，可视化操作使熟悉代码的用户能轻松完成复杂的工作，但要想深入掌握网页编程，建议用户进一步学习脚本语言，这样会使自己的网页编程更随心所欲。



用户登录模块

本章与上一章模块在功能上紧密联系，因此本章实例将在上一章实例的基础上进行介绍。虽然用户注册和登录是两个不同的模块，但是它们之间的关系比较紧密。用户要登录必须先注册，只有注册之后才能登录，注册的目的是为了登录。在后台设计中，注册模块和登录模块一般都使用同一个数据表。随着 Web 技术的不断发展演化，各种复杂运用令人眼花缭乱，但其核心功能是相同的，即使用 Session 会话变量实现用户信息的站内传递。

技术要点

- 用户登录
- 会话变量
- 注销用户
- 客户端变量存储
- 图形验证码
- 防止 SQL 注入式攻击

7.1 程序设计分析

当用户注册之后，下一步的操作就是用注册的用户名和密码登录网站了。在用户注册时，输入的用户名和密码被提交到服务器，服务器处理程序首先要检验用户名是否合法和密码是否正确，然后才能打开相应的页面，表示登录成功；若登录不成功，则会给出错误的提示。

要实现用户登录功能，读者应该学习和掌握以下 ASP 服务器技术。

- 设计后台数据库，读者可以使用上一章的 user.mdb 数据库。
- 设计登录表单，以及登录成功和失败时显示的页面，读者可以使用实例模板。
- 利用客户端行为验证登录信息的合法性。
- 利用服务器端行为完成用户注册。
- 使用 Session 会话变量在站内传递用户信息。
- 利用服务器端行为注销用户。
- 手动编写代码实现注册信息的存储和显示。
- 使用图形验证码防止不良用户的伪注册和攻击。
- 防止恶意 SQL 注入式攻击。
- 防止用户绕开登录。



学习本实例之前,读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录,如 site (详细说明可以参阅第 2.4.2 节介绍)。

启动 Dreamweaver CS5 创建一个动态站点,远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”,【访问】应为“本地/网络”(详细设置可以参阅第 2.5.2 节介绍)。

把本书光盘中的“7”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构,并帮助读者建立了数据库连接。

7.2 数据库设计

本章用户登录模块和第 6 章用户注册模块关系密切,因此本章实例以第 6 章实例的数据库为基础进行操作,用户可把该数据库复制到本地文件夹中,具体创建可参见第 6.2 小节。

7.3 页面设计

本实例主要包括 3 个文件,即 index.asp、ok.asp、fail.asp。

- index.asp 为登录页面,页面包含一个登录表单,登录模块一般包括客户端和服务端行为代码。
- ok.asp 是登录成功提示页面,也可以从该页退出登录。
- fail.asp 是登录失败提示页面。

在 index.asp 文件中包含了一个登录表单。登录页面设计如图 7.1 所示。



图 7.1 登录表单设计效果

本页登录表单主要包括 2 个文本框和 1 个提交按钮。选中整个表单,在【属性】面板中设置【表单名称】为 login,【方法】为“POST”,在【动作】文本框中可以设置要提交表单数据要到的页面地址,用户可以先不设置,当插入服务器行为后会自动设置一个变量,如 <%=MM_LoginAction%>,该变量存储着要提交的 URL。

下面列出本页面表单中 2 个文本框的属性,设置见表 7.1。

表 7.1 index.asp 中文本框对象设置

对 象	名 称	类 型	最多字符数	字符宽度	备 注
文本框	name	单行	-	-	用户名
文本框	pass	密码	-	-	密码


最后，插入一个提交按钮，设置【值】为“提交”。

7.4 登录基本功能实现

登录功能主要包括登录信息的验证和登录信息的存储。其中登录信息验证是一个关键技术，它可以在客户端实现，也可以在服务器端完成，读者应该结合前后台技术来实现，防止不良用户的注册攻击。

7.4.1 客户端验证

客户端验证主要使用 Dreamweaver 提供的“检查表单”行为实现，上一章已经详细介绍了，操作步骤如下。

- 01 在 Dreamweaver CS5 中，打开 index.asp 页面，准确选中表单。
- 02 在菜单中选择【窗口】|【行为】命令，打开【行为】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【检查表单】菜单命令。
- 03 打开【检查表单】对话框，具体设置如图 7.2 所示。在【域】列表框中详细列出了页面中全部表单文本框的检查条件。设置 name 和 password 文本框的【值】为“必须的”，【可接受】选项为“任何东西”。
- 04 单击【确定】按钮保存设置，返回【行为】面板，并在事件选项栏中设置 onsubmit 事件选项，表示当用户单击提交按钮时，会触发该事件，如图 7.3 所示。

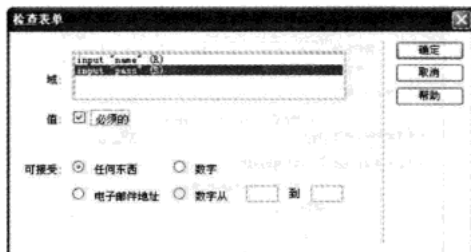


图 7.2 【检查表单】对话框



图 7.3 设置检查表单触发事件

- 05 切换到【代码】视图，在</head>标记上面找到 function MM_validateForm()函数代码段，汉化 errors 变量的内容，使错误提示内容显示为中文（详细介绍可参见第 6.5.2 小节）。修改代码如下：

```
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
    if (document.getElementById){
        var i,p,q,nm,test,num,min,max,errors='',args=MM_validateForm.arguments;
        for (i=0; i<(args.length-2); i+=3) { test=args[i+2]; val=document.getElementById
(args[i]);
            if (val) { nm=val.name; if ((val=val.value)!="") {
                if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
                    if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an e-mail address.\n';
                } else if (test!='R') { num = parseFloat(val);
                    if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
                    if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
                        min=test.substring(8,p); max=test.substring(p+1);
```


```

        if (num<min || max<num) errors+='- '+nm+' must contain a number between '+min+'
and '+max+'.\n';
    } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' 必须输入!\n'; }
    } if (errors) alert('错误提示:\n'+errors);
    document.MM_returnValue = (errors == '');
} }
//-->
</script>

```

7.4.2 服务器端用户登录

在客户端初步完成用户输入信息的验证,接着在服务器端增加服务器行为验证用户输入的用户名和密码是否正确,具体操作步骤如下。

- 01 选择【窗口】|【服务器行为】菜单命令,打开【服务器行为】面板。
- 02 单击【服务器行为】面板左上角的按钮,在弹出的下拉菜单中选择【用户身份验证】|【登录用户】菜单命令,如图 7.4 所示。
- 03 打开【登录用户】对话框,设置如图 7.5 所示。

在【登录用户】对话框的前 3 个选项中,分别选择与输入登录信息表单中相关的 3 个选项。从【从表单获取输入】下拉列表中选择表单 login 项;从【用户名字段】下拉列表中选择用户名文本框 name 项;从【密码字段】下拉列表中选择密码输入文本框 pass 项。

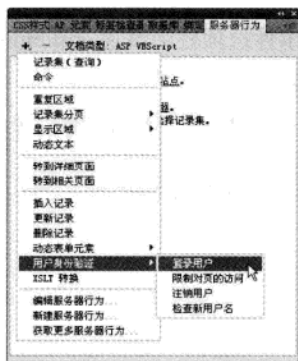


图 7.4 选择【登录用户】服务器行为

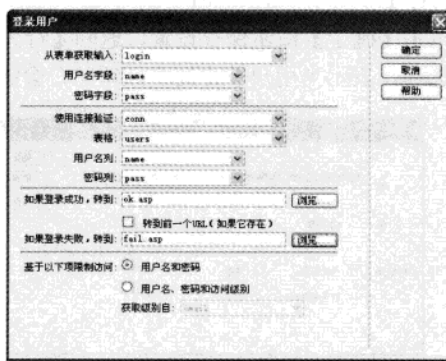


图 7.5 【登录用户】对话框

- 04 在【使用连接验证】下拉列表中选择已建立的连接 conn,从【表格】下拉列表中选择数据库中存储用户信息的表 users,然后分别在【用户名列】和【密码列】下拉列表中选择表 users 中相对应字段,如 name 字段和 pass 字段。利用这两个字段数据在服务器端验证用户输入的用户名和密码是否合法。

- 05 在【如果登录成功,转到】文本框中输入登录成功后跳转的页面,如 ok.asp。在【如果登录失败,转到】文本框中输入登录失败后跳转的页面,如 fail.asp。也可以单击这两个文本框右边的【浏览】按钮进行选择,在这两个文本框中设置了登录成功时转向 ok.asp 页,登录失败时转向 fail.asp 页。

- 06 单击【确定】按钮保存设置即可。此时在【服务器行为】面板中显示登录用户的服务器行为,如图 7.6 所示。

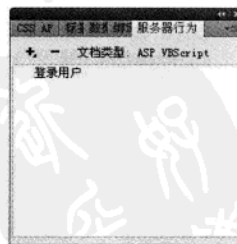


图 7.6 【服务器行为】面板



提示

用 Request 对象读取用户输入信息是开发中经常进行的操作。但是 Request 是通过读取 HTTP 头并进行查找来获取用户输入的, 这一过程要占用一定的系统资源。如果用户在同一个页面中多次调用到同一个用户信息, 则可以先将 Request 获取的信息保存在变量中, 然后在 ASP 代码中直接调用这个变量就可以了, 这样能够提高程序运行效率。

7.4.3 传递会话变量

完成 index.asp 页面的用户登录工作之后, 打开 ok.asp 文件, 设置当登录成功后准备显示的动态提示信息。这里面要用到 Session 会话变量, 把 index.asp 页面的用户名传递过来, 并显示在页面上, 显示效果如图 7.7 所示。



提示

在 ASP 中共有两大类页面之间的变量: Session (会话变量) 和 Application (应用程序变量)。Session 变量针对单个用户, 也就是说, 当某个用户在网站的页面之间切换时, 只能访问自己的 Session 变量, 而无法访问其他用户的 Session 变量。

Application 变量是一个应用程序级的变量, Application 包含的数据可以在整个 Web 站点中被所有用户使用, 并且可以在网站运行期间持久保存数据。Application 对象是针对所有用户, 可以被多个用户共享; 而且一旦建立了 Application 变量, 就会一直存在, 直到网站关闭或这个 Application 变量被卸载 (Unload)。而 Session 变量会随着用户离开网站而被自动删除。

完成会话传递的具体操作步骤如下。

- 01 打开 ok.asp 文件后, 选择【窗口】|【绑定】菜单命令, 打开【绑定】面板。
- 02 单击【绑定】面板左上角的 按钮, 在弹出的下拉菜单中选择【阶段变量】命令, 如图 7.8 所示。



图 7.7 传递会话变量显示效果

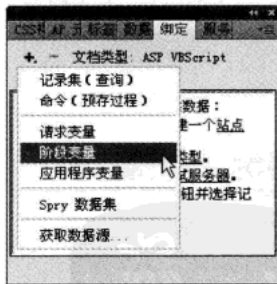


图 7.8 选择【阶段变量】命令



提示

在【绑定】面板的下拉菜单中有 3 个变量类型, 即请求变量、阶段变量和应用程序变量。请求变量主要用来获取 Form (表单)、Cookies (客户端信息)、QueryString (查询字符串)、ClientCertificate (客户端验证字段) 和 ServerVariables (服务器环境变量) 数据集合内部数据。

阶段变量就是 Session 变量, 通俗说是私有变量, 不同用户有不同的阶段变量, 并不能永久保存。

应用程序变量就是 Application 变量, 通俗说就是公有变量, 所有用户都可以使用, 并能长久保存。

- 03** 打开【阶段变量】对话框，在【名称】文本框中输入 Session 变量名 MM_Username，如图 7.9 所示。该变量名是在 index.asp 页面中定义“用户登录”服务器行为时系统自动定义的，读者可以在 index.asp 文档中切换到【代码】视图，然后在文档顶部找到下面代码：

```
.....
If Not MM_rsUser.EOF Or Not MM_rsUser.BOF Then
    ' username and password match - this is a valid user
    Session("MM_Username") = MM_valUsername
    If (MM_fldUserAuthorization <> "") Then
        Session("MM_UserAuthorization") = CStr(MM_rsUser.Fields.Item(MM_fldUserAuthorization).Value)
    Else
        Session("MM_UserAuthorization") = ""
    End If
    if CStr(Request.QueryString("accessdenied")) <> "" And false Then
        MM_redirectLoginSuccess = Request.QueryString("accessdenied")
    End If
    MM_rsUser.Close
    Response.Redirect (MM_redirectLoginSuccess)
End If
.....
```

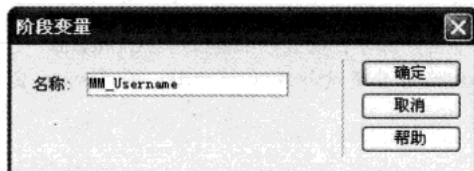


图 7.9 【阶段变量】对话框

- 04** 单击【确定】按钮，在【绑定】面板中会显示该绑定变量，如图 7.10 所示。
- 05** 在 ok.asp 页面中，把光标置于要显示用户名的地方，然后在【绑定】面板中选择 MM_Username，在面板底部单击【插入】按钮，此时页面如图 7.11 所示。




图 7.10 【绑定】面板



图 7.11 插入变量

7.4.4 注销用户

定义 Session 变量后（登录用户），如果用户要离开页面或退出站点，应注销用户（即释放 Session 变量），避免别人利用用户信息进入网站进行操作，具体操作步骤如下。

- 01 打开 ok.asp 文件，插入“注销”超链接文本。选择【窗口】|【服务器行为】菜单命令，打开【服务器行为】面板。
- 02 单击【服务器行为】面板左上角的  按钮，在弹出的下拉菜单中选择【用户身份验证】|【注销用户】命令。此时会打开【注销用户】对话框，如图 7.12 所示。

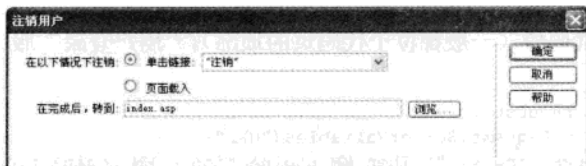


图 7.12 【注销用户】对话框

- 03 在【在以下情况下注销】选项中选择【单击链接】单选按钮，然后单击【在完成后，转到】文本框右侧的【浏览】按钮，从中选择 index.asp 文件。

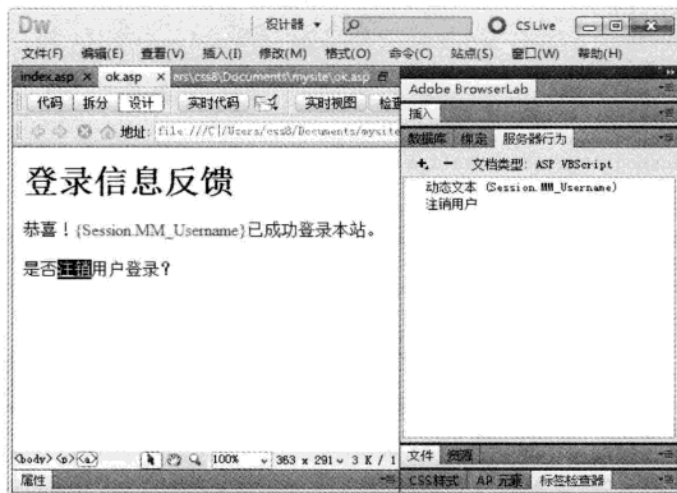


图 7.13 设置注销用户服务器行为

- 04 单击【确定】按钮保存设置。这时会在页面光标处插入一个【注销】字样的链接，如图 7.13 所示。单击该链接会跳转到 index.asp 文件，同时释放 MM_User 变量。

7.5 自动记忆输入密码

很多网站的用户登录表单中常提供一个密码记忆功能，当用户在第一次登录时输入用户名和密码，下一次登录时就不再输入密码而直接进入。

一般服务器是从 Form 表单中获取用户名和密码信息，如果把用户名和密码存储到客户端计算机内的 Cookie 变量中（在客户端的 Cookie 中存储着用户个人信息，供浏览器和服务端之间交换信息使用），登录时就可以不用输入用户名和密码了。当下一次用户登录时就直接从 Cookie 变量中读取登录信息，这样就不需要用户直接在表单中输入了。

具体实现自动记忆输入密码的操作步骤如下。

01 在 Dreamweaver CS5 中打开 index.asp 文件, 切换到【代码】视图, 添加如下复选框代码:

```
<li>
    <label for="auto">开启密码记忆: </label>
    <input name="auto" type="checkbox" id="auto" value="1" />
</li>
```

02 找到用户登录代码段 (一般都位于代码页的顶部)。“用户登录”服务器行为代码如下:

```
<%
' *** Validate request to log in to this site.
MM_LoginAction = Request.ServerVariables("URL")
If Request.QueryString <> "" Then MM_LoginAction = MM_LoginAction + "?" + Server.
HTMLEncode(Request.QueryString)
MM_valUsername = CStr(Request.Form("name"))
If MM_valUsername <> "" Then
    Dim MM_fldUserAuthorization
    Dim MM_redirectLoginSuccess
    Dim MM_redirectLoginFailed
    Dim MM_loginSQL
    Dim MM_rsUser
    Dim MM_rsUser_cmd

    MM_fldUserAuthorization = ""
    MM_redirectLoginSuccess = "ok.asp"
    MM_redirectLoginFailed = "fail.asp"

    MM_loginSQL = "SELECT name, pass"
    If MM_fldUserAuthorization <> "" Then MM_loginSQL = MM_loginSQL & "&" & MM_fldUserAuthorization
    MM_loginSQL = MM_loginSQL & " FROM users WHERE name = ? AND pass = ?"
    Set MM_rsUser_cmd = Server.CreateObject ("ADODB.Command")
    MM_rsUser_cmd.ActiveConnection = MM_conn_STRING
    MM_rsUser_cmd.CommandText = MM_loginSQL
    MM_rsUser_cmd.Parameters.Append MM_rsUser_cmd.CreateParameter("param1", 200, 1, 30,
MM_valUsername) ' adVarChar
    MM_rsUser_cmd.Parameters.Append MM_rsUser_cmd.CreateParameter("param2", 200, 1, 16,
Request.Form("pass")) ' adVarChar
    MM_rsUser_cmd.Prepared = true
    Set MM_rsUser = MM_rsUser_cmd.Execute

    If Not MM_rsUser.EOF Or Not MM_rsUser.BOF Then
        ' username and password match - this is a valid user
        Session("MM_Username") = MM_valUsername
        If (MM_fldUserAuthorization <> "") Then
            Session("MM_UserAuthorization") = CStr(MM_rsUser.Fields.Item
(MM_fldUserAuthorization).Value)
        Else
            Session("MM_UserAuthorization") = ""
        End If
        End If
        if CStr(Request.QueryString("accessdenied")) <> "" And false Then
            MM_redirectLoginSuccess = Request.QueryString("accessdenied")
        End If
        MM_rsUser.Close
        Response.Redirect(MM_redirectLoginSuccess)
    End If
    MM_rsUser.Close
    Response.Redirect(MM_redirectLoginFailed)
End If
%>
```

上面代码段是逐条从数据库中取出用户名和密码，并与输入的用户名和密码进行比较验证，然后根据条件的真假分别跳转到不同的页面。



注意

“密码记忆”代码功能就是实现把用户名和密码存储到本地计算机的 Cookie 中，以便下一次登录时直接从本地硬盘上获得用户名和密码。记忆的用户名和密码必须是合法的，否则登录时就要自动转向错误页了。所以这样的语句应该在客户端验证和服务器端验证之后，即在转向 ok.asp 页之前执行。

- 03** “密码记忆”代码功能是在客户端验证和服务器端验证都通过后，根据是否选中【开启密码记忆】复选框来判断在登录前是否记住密码。因此，在上面代码段中 Response.Redirect(MM_redirectLoginSuccess)语句前面插入下面一段代码：

```
.....
MM_rsUser.Close
'在客户端Cookie变量中存储用户名和密码，以便下次访问时页面自动登录
'-----
if Request.Form("auto")="1" then
    Response.Cookies("user")=Request.Form("name")
    Response.Cookies("pass")=Request.Form("pass")
    Response.Cookies("user").Expires=Date+365
    Response.Cookies("pass").Expires=Date+365
end if
'-----
Response.Redirect (MM_redirectLoginSuccess)
End If
MM_rsUser.Close
Response.Redirect (MM_redirectLoginFailed)
End If
%>
```

上面新增加的代码会首先判断用户是否选中【开启密码记忆】复选框，如果选中复选框，则把表单中输入的用户名和密码存储到 Cookie 变量中，并设置存储时间为 1 年，Date 代表当前日期。

- 04** 把用户名和密码存储到本地 Cookie 变量，还需要从 Cookie 变量中读取用户名和密码。

在上面的“用户登录”代码段中的第 7 行代码：

```
MM_valUsername = CStr(Request.Form("name"))
```

是说明从登录表单中获取用户名，并把该用户信息赋予给变量 MM_valUsername。

- 05** 在上面的“用户登录”代码段中的第 25 行代码中：

```
MM_rsUser_cmd.Parameters.Append MM_rsUser_cmd.CreateParameter("param1", 200, 1, 30,
MM_valUsername) ' adVarChar
```

这行代码是从登录表单中获取密码，并把该密码赋值给 Command 对象的参数变量集合中。如果修改该句则可以不需要每次都输入密码，代码如下：

```
'读取Cookies("password")变量的值，并根据是否为空执行不同的赋值操作
'-----
if Request.Cookies("password")<>" " then
    MM_rsUser_cmd.Parameters.Append MM_rsUser_cmd.CreateParameter("param2", 200, 1,
16, Cstr(Request.Cookies("password"))) ' adVarChar
else
    MM_rsUser_cmd.Parameters.Append MM_rsUser_cmd.CreateParameter("param2", 200, 1,
16, Request.Form("pass")) ' adVarChar
end if
'-----
```

该段代码也是先判断 Cookies 集合中是否已存储了用户密码,如果存储则从本地 Cookies 变量中获取密码,否则从登录表单中获取密码。



注意

在本步骤操作前,建议用户先修改“检查表单”行为,取消对 pass 文本框的【值】选项中的【必须的】复选框,否则当不输入密码时,系统会提示输入密码。

7.6 增加图形验证码

现在网站的注册和登录模块中都增加图形验证码功能。使用图形验证码可以防止恶意用户通过代码或者注册机等方式批量注册或登录网站,这样会严重威胁网站的安全。

图形注册码的实现原理不是很复杂。所谓验证码,就是将一串随机生成的数字或字符符号生成一幅图片,有时还可以在图片里加上一些干扰像素(防止 OCR 扫描),强迫用户肉眼识别其中的验证码信息,并输入表单提交系统验证,验证成功后才能进入使用。早期的验证码是利用随机函数生成一组随机数字作为验证码,但很容易被黑客破解,破解者可以利用类似于小偷程序的技术,先把用户注册或登录页面读取,然后分析出验证码,最后进行暴力猜解。这样的话,验证码就没有任何实际意义了。现在很少使用这种方式了,多使用图形验证码,增加验证码的识别难度,防止暴力猜解密码。

增加图形验证码功能的具体实现具体步骤如下。

- 01 要实现在 ASP 文件中显示图形验证码,需要 3 个文件: getcode.asp (用于生成图片验证码)、head.fix (图片头) 和 body.fix (图片主体)。首先,读者可以把“添加图形验证码”文件夹中的 head.fix 和 body.fix 文件复制到站点根目录下。
- 02 新建 getcode.asp 文件,利用该文件生成图片验证码。然后输入下面代码:

```
<% '生成图片验证码
Option Explicit '强制声明变量
Response.buffer=true '启动缓存
NumCode '调用图片验证码函数
Function NumCode() '图片验证码函数
    Response.Expires = -1
    Response.AddHeader "Pragma","no-cache"
    Response.AddHeader "cache-ctrol","no-cache"
    On Error Resume Next
    Dim zNum,i,j
    Dim Ados,Ados1
    Randomize timer
    zNum = cint(8999*Rnd+1000) '生成随机四位数字
    Session("GetCode") = zNum '把随机数传递给session 变量
    Dim zimg(4),NStr
    NStr=cstr(zNum)
    For i=0 To 3 '利用for循环将随机数字放入一个下标为3的数组中,便于提供给后面的阵列变换
        zimg(i)=cint(mid(NStr,i+1,1))
    Next
    '定义2个 ADODB.Stream binary对象,作为图像数据操作之用
    Dim Pos
    Set Ados=Server.CreateObject("Adodb.Stream")
    Ados.Mode=3
    Ados.Type=1
    Ados.Open
    Set Ados1=Server.CreateObject("Adodb.Stream")
    Ados1.Mode=3
    Ados1.Type=1
```

```

Ados1.Open
'载入0~9的数字数据10*100的Gbr的阵列数据,每个320字节,10个数字3200byte
'BGR一个点,10*10个点表示一个数字,一个点2个字节(二进制8位,16进制 00~FF)
'一行10个点 30字节 + 行结束标记 00 00 二字节 32字节,所以一个10*100宽小于长的图像每个数字
10*10是320字节
'长大于宽的则无行结束标记 0000,直接是300字节
'这些就是BMP 24bit的数据详细信息了
'至于头部,也很简单,包含长宽,图像开始标记等于54字节
'注意, getcode.asp、 body.fix和head.fix应该在同一目录中, 否则要修改相对位置
Ados.LoadFromFile(Server.mappath("body.fix"))
'第1个for循环,按生成的随机数字顺序从10*100的数字阵列中提取出相应的4个数字
Ados1.write Ados.read(1280)
For i=0 To 3
    Ados.Position=(9-zimg(i))*320
    Ados1.Position=i*320
    Ados1.write ados.read(320)
Next
'清空已经用完的Ados的数据,调入替换新的图像头54字节的头文件
Ados.LoadFromFile(Server.mappath("head.fix"))
Pos=lenb(Ados.read())
Ados.Position=Pos '指定Pos位置,即可再偏移54字节的位置添加图形数据
'第2个for循环,进行数字的阵列变换,由竖排的块转换为横排的数字块 方法是隔320字节抽取4次30字节
写入Ados对象,再抽取偏移第2行的图像数据 30字节是因为bmp 宽大于长时无00 00的行结束标记
For i=0 To 9 Step 1
    For j=0 To 3
        Ados1.Position=i*32+j*320
        Ados.Position=Pos+30*j+i*120
        Ados.write ados1.read(30)
    Next
Next
Response.ContentType = "image/BMP"
Ados.Position=0
Response.BinaryWrite Ados.read() '直接向客户端发送图像数据
Ados.Close:set Ados=nothing
Ados1.Close:set Ados1=nothing
If Err Then Session("GetCode") = 9999
End Function
%>

```

打开 index.asp, 然后切换到【代码】视图, 找到表单代码区域, 在“登录密码”行下面添加如下代码:

```

<li>
    <label for="auto">验证码: </label>
    <input name="useryz" type="text" id="useryz" size="6" /><img src=getcode.asp border=0>
</li>

```

在上面结构代码中插入一个文本框(用来接收验证码输入)和一个标记, 然后定义该标记的 src 属性为 getcode.asp, 即调用图形验证码。

03 每当调用 getcode.asp 之后, ASP 服务器会自动生成一个 Session 变量, 这个 Session 变量存储着和验证图片上显示的一样的 4 位数字。读者只需把从表单中接收到的验证码与这个 Session 值进行比较, 然后判断验证码是否输入正确。在 index.asp 文件的顶部输入下面代码段, 该代码段的目的是把用户输入到表单中的验证码保存在 Session("useryz"), 以便在 ok.asp 文件中比较使用。

```

<% '把表单中的验证码保存到Session变量中
if Request.Form("useryz")<> "" Then
    Session("useryz") = Request.Form("useryz")
end if
%>

```



提示

有的读者可能有这样的疑问：为什么不直接在 ok.asp 文件中用 Request.Form("useryz") 获取表单中输入的验证码呢？这是因为当读者插入“用户登录”服务器行为后，Dreamweaver CS5 把表单中的数据转换为 URL 字符串方式传递，所以 Request.Form("useryz") 就不能获取有效表单数据。

04 打开 ok.asp 文件，切换到【代码】视图，在文档代码顶部输入下面的代码。这段代码将把 getcode.asp 文件生成的验证码与用户在表单中输入的数字进行比较，如果不一致则提示重新输入，如图 7.14 和图 7.15 所示。

```
<% '把getcode.asp文件生成的验证码与用户在表单中输入的数字进行比较
if cstr(session("getcode"))<>cstr(Session("useryz")) then
    response.codePage = 65001
    response.charset="UTF-8"
    response.write "<script language='javascript'"
    response.write "alert('验证码输入有误，请核对！');"
    response.write "history.go(-1);"
    response.write "</script>"
    response.end
end if
%>
```



图 7.14 验证码显示效果



图 7.15 输入验证码不一致的错误提示信息

7.7 防止 SQL 注入式攻击

登录环节控制着用户访问网站的权限。当用户的登录信息被提交到服务器端，一般都会直接用来构造动态的 SQL 字符串命令，或者直接用做存储过程的参数。此时如果读者的安全意识不强，很可能会被恶意者钻空子。例如，在下面 SQL 查询字符串中：

```
string sql = "select * from userinfo where usertext=" & pu & " and userpass=" & pw
```

那么用户就有可能通过输入“usertext”和“userpass”登录信息骗过服务器的验证。如果用户在用户名和密码输入框中输入“or 1=1”之类的登录信息，也同样可以骗过服务器。

实际上简单分析这个 SQL 字符串也很好理解，由于用户输入的信息非常特殊，所以最后得到的 SQL 字符串就变成了：

```
string sql = "select * from userinfo where usertext= '' or 1=1 and userpass= '' or 1=1;
```

由于 $1=1$ 是一个恒等式, 所以这个 SQL 字符串查询命令实际上就形成了一个无条件查询, 当然能够通过服务器的验证了。

如果用户在表单中输入一些 SQL 关键字或者命令, 甚至会对后台数据库的数据造成破坏, 严重威胁到网站的安全。因此如何防止 SQL 注入式攻击是读者必须重视的问题。

要解决此类问题, 可以定义一个函数, 在该函数中过滤掉用户输入的所有可能存在的不良关键字。然后当用户登录时, 首先调用该函数过滤用户输入信息。具体实现过程如下。

01 新建 `safe_sql.asp` 文件, 在其中定义一个函数 `safe_sql()`, 该函数的代码如下:

```
<%
function safe_sql(strGet)
    strTemp=strGet
    dim sqlstr(16)
    sqlstr(0)=""
    sqlstr(1)=""
    sqlstr(2)="select"
    sqlstr(3)="insert"
    sqlstr(4)="delete"
    sqlstr(5)="from"
    sqlstr(6)="drop"
    sqlstr(7)="truncate"
    sqlstr(8)="asc"
    sqlstr(9)="char"
    sqlstr(10)="mid"
    sqlstr(11)="count"
    sqlstr(12)="net user"
    sqlstr(13)="xp_cmdshell"
    sqlstr(14)="/add"
    sqlstr(15)="exec%20master.dbo.xp_cmdshell"
    sqlstr(16)="net localgroup administrators"
    for i=1 to ubound(sqlstr)
        if instr(strTemp,sqlstr(i)) then
            response.codePage = 65001
            response.charset="UTF-8"
            response.write "<script language='javascript'>"
            response.write "alert('你输入的内容含有非法字符, 请返回重试!');"
            response.write "history.go(-1);"
            response.write "</script>"
            response.end
        end if
    next
    save_sql=strTemp
end function
%>
```

在上面代码中定义一个数组 `sqlstr`, 然后把一些危险的字符都保存在该数组中, 读者也可以无限增加该数组, 然后用该数组与用户输入的信息进行逐个比较, 如果发现相同则提示返回。

02 打开 `index.asp` 文件, 在文档代码顶部利用 `#include` 语句包含该函数文件 `safe_sql.asp`, 代码如下:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="65001"%>
<!--#include file="Connections/conn.asp" -->
<!--#include file="safe_sql.asp" -->
```

03 打开 `index.asp` 文件, 在“登录用户”服务器行为代码段中找到下面代码行, 调用 `safe_sql()` 函数过滤危险字符:

```
MM_valUsername = CStr(Request.Form("name"))
```

修改为:

```
MM_valUsername = safe_sql(CStr(Request.Form("name")))
```

- 04 其他地方的修改与上面类似,读者可以自己动手修改。这时,如果读者再在登录表单中输入了危险字符,系统就会提示警告信息并返回,如图 7.16 和图 7.17 所示。



图 7.16 输入危险字符




图 7.17 警告提示信息

7.8 防止用户绕过登录

为了防止用户不经过登录页面而直接进入其他页面,读者还应该对每位访问者进行身份验证,判断用户是否已登录,如果没有登录则直接跳转到登录页面强制用户登录。

判断用户是否登录,只需要判断 Session("MM_Username")变量是否为空。也可以直接使用服务器行为进行验证。具体操作步骤如下。

- 01 新建一个 validate.asp 文件。打开该文件,切换到【代码】视图,清除该文档中的所有 HTML 代码,然后再切换到【设计】视图。
- 02 选择【窗口】|【服务器行为】菜单命令,打开【服务器行为】面板。
- 03 单击【服务器行为】面板左上角的  按钮,在弹出的下拉菜单中选择【用户身份验证】|【限制对页的访问】菜单命令。
- 04 打开【限制对页的访问】对话框,设置如图 7.18 所示。在【基于以下内容进行限制】选项中选择【用户名和密码】项;在【如果访问被拒绝,则转到】文本框中输入“index.asp”文件。

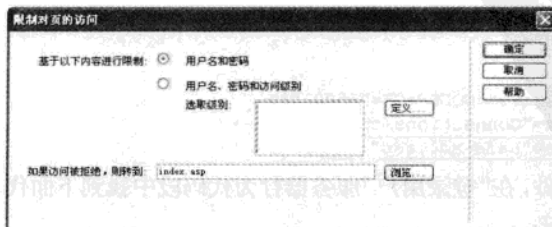


图 7.18 【限制对页的访问】对话框

05 单击【确定】按钮，关闭对话框。然后切换到【代码】视图，删除第1行代码：

```
<%@LANGUAGE="VBSCRIPT"%>
```

06 读者可以打开其他任何文件，然后在文件顶部包含该验证文件（代码如下），这样当用户访问站点内每个文件时，都要先对用户身份进行验证，如果通过则允许用户浏览，否则就会跳转到 index.asp 文件要求用户登录。例如，打开 ok.asp 文件，然后在顶部输入下面代码行：

```
<!--#include file="validate.asp" -->
```



提示

模块化代码更容易编写、调试、理解和重用，大段的代码会互相影响，而且会引发各种奇怪的错误。程序模块化在开发中可能要付出更多的时间，但可以为后期的修改与维护节省大量的时间。更重要的是拆分后的每个函数的执行过程都相对简单，删除与更新也就更容易一些。ASP 的 include 相当于把指定文件中的代码全部引用。如果一个 ASP 文件使用 include 包含其他 ASP 文件，那么这些文件就相当于一个文件，include 部分会被指定文件的代码所代替。使用 include 是实现模块化的一个好方法。例如，本实例中的防止 SQL 注入式攻击和防止用户绕过登录都是采用这种方式，建议读者多多采用。

07

7.9 实践总结

本章重点介绍登录页的制作过程和需要添加的行为，同时也介绍了登录页的其他功能。登录和注册是动态网页操作的第1步，两者关系比较密切，在学习本章时可以结合前一章内容，使它们能紧密结合在一起。当然要制作更加完善的登录页面还需要用户增加手写代码，在后面章节中还会介绍各种形式的注册和登录页面设计及功能。



第 8 章

Dreamweaver

用户管理模块

用户管理在网站管理中是一项核心功能模块，其重要性不言而喻。一般网站对于用户管理都要求能够实现分级管理，不同级别的用户将拥有不同的权限。这正如 Windows 操作系统的用户账户管理功能类似。大型网站的用户管理都会把权限分为普通用户、高级用户（VIP 用户）、管理员、高级管理员、系统管理员等不同级别用户。普通用户只能管理个人基本信息，高级用户可以拥有更大个人管理能力；而普通管理员可以管理本组或者所有用户信息（根据网站规模而定），高级管理员可以管理全部用户信息和普通管理员的权限设置等；而系统管理员能够实现对整个网站的所有前后台事务的管理，拥有最大权限。

当然，不同类型的网站由于规模和风格不同决定了这种分级管理的形式不尽相同，当然现实中的网站也很难有一个统一的管理模式。不过，读者只要掌握网站管理的基本设计思路，也就能以不变应万变了。

技术要点

- 多身份用户登录
- Cookies 技术
- URL 查询字符串多参数传递
- 信息模糊查询
- 记录集智能筛选
- 记录集高级显示

8.1 程序设计分析

实际上用户管理的内容包括很多，大到网站结构的管理，小到内容的管理，所以比较复杂。本章将演示用户管理的基本模式，限于篇幅本书没有设计更复杂的管理功能。本实例只把用户权限分为 2 个级别：普通用户和管理员用户，以实现不同的内容管理。用户管理系统的基本功能包括用户信息的增加、修改和删除，以及查询浏览用户信息等。

本实例模块分为两个独立的功能模块，根据用户登录的级别，验证合法性之后，分别进入两个独立的操作空间，然后完成用户信息的管理操作。普通用户只能管理个人信息，而管理员为多人制，且级别相同，任一合法管理员均可对管理员账号进行添加、删除及修改操作，设计流程如图 8.1 所示。

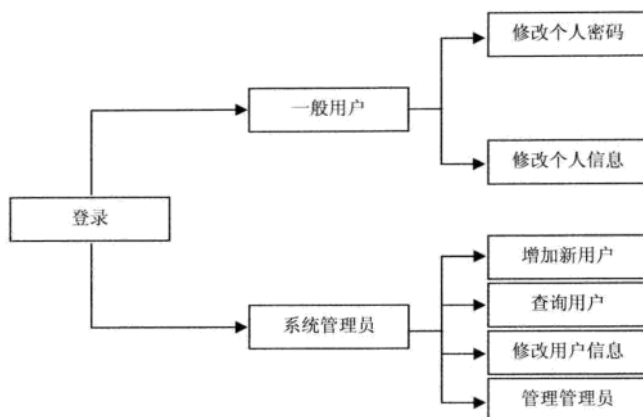


图 8.1 用户管理流程图

本实例涉及用户管理模块的基本功能，读者在学习过程中应该掌握和解决下面技术问题。

- 多级用户登录。可通过代码实现对用户身份的判断并进行登录。
- 登录用户的信息传递。使用 Cookies 技术并结合 QueryString 变量实现。
- 查询用户。需要灵活使用 SQL 查询字符串。
- 记录集的分页显示。可利用前面一章介绍的服务器行为技术来实现。
- 利用“插入记录”、“删除记录”和“更新记录”服务器行为实现对记录的一般操作。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，如 site（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”（详细设置说明可以参阅第 2.5.2 节介绍）。

把本书光盘中的“8”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

8.2 数据库设计

本实例模块所连接的数据库名称为 admin.mdb，该数据库包含两张表：admin 和 infor。

8.2.1 admin 表

admin 表用来记录管理员的用户信息，包括管理员姓名和密码。该表数据结构设计见表 8.1。

表 8.1 admin 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
admin_id	自动编号	-	-	-	自动编号
admin_name	文本	50	是	否	管理员姓名
admin_pwd	文本	50	是	否	密码

最后，在 admin 表中输入几组用户名和密码，以方便在练习时使用。

8.2.2 infor 表

infor 表主要用来存储普通用户信息，该表数据结构设计见表 8.2。

表 8.2 infor 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
tc_id	自动编号	-	-	-	自动编号
tc_name	文本	50	是	否	用户姓名
tc_sex	文本	4	是	否	性别
tc_birt	文本	50	是	否	出生年月
tc_indate	文本	20	是	否	上班时间
tc_code	文本	20	是	否	身份证号
tc_address	文本	200	是	否	家庭住址和联系电话
tc_zhy	文本	50	是	否	所学专业
tc_dw	文本	50	是	否	所在单位
tc_xl_bydate1	文本	50	是	否	第 1 次学历、教育、毕业时间
tc_xl_bydate2	文本	50	是	否	第 2 次学历、教育、毕业时间
tc_xl_bydate3	文本	50	是	否	第 3 次学历、教育、毕业时间
tc_xl_byschool1	文本	50	是	否	第 1 次学历、教育、毕业学校
tc_xl_byschool2	文本	50	是	否	第 2 次学历、教育、毕业学校
tc_xl_byschool3	文本	50	是	否	第 3 次学历、教育、毕业学校
tc_xl_zhy1	文本	50	是	否	第 1 次学历、教育、所学专业
tc_xl_zhy2	文本	50	是	否	第 2 次学历、教育、所学专业
tc_xl_zhy3	文本	50	是	否	第 3 次学历、教育、所学专业
tc_work_jx	备注	-	是	否	工作成绩、工作成果
tc_work_ky	备注	-	是	否	工作成绩、科研成果
tc_work_xsh	备注	-	是	否	工作成绩、学术专著
tc_shh_xsh_zhw1	文本	50	是	否	第 1 次社会兼职、学术团体、职务
tc_shh_xsh_zhw2	文本	50	是	否	第 2 次社会兼职、学术团体、职务
tc_shh_xsh_zhw3	文本	50	是	否	第 3 次社会兼职、学术团体、职务
tc_shh_xsh_date1	文本	50	是	否	第 1 次社会兼职、学术团体、任职时间
tc_shh_xsh_date2	文本	50	是	否	第 2 次社会兼职、学术团体、任职时间
tc_shh_xsh_date3	文本	50	是	否	第 3 次社会兼职、学术团体、任职时间
tc_shh_xsh_name1	文本	50	是	否	第 1 次社会兼职、学术团体、团体名称
tc_shh_xsh_name2	文本	50	是	否	第 2 次社会兼职、学术团体、团体名称
tc_shh_xsh_name3	文本	50	是	否	第 3 次社会兼职、学术团体、团体名称
tc_shh_shh_zhw1	文本	50	是	否	第 1 次社会兼职、社会团体、职务
tc_shh_shh_zhw2	文本	50	是	否	第 2 次社会兼职、社会团体、职务
tc_shh_shh_zhw3	文本	50	是	否	第 3 次社会兼职、社会团体、职务
tc_shh_shh_date1	文本	50	是	否	第 1 次社会兼职、社会团体、任职时间
tc_shh_shh_date2	文本	50	是	否	第 2 次社会兼职、社会团体、任职时间
tc_shh_shh_date3	文本	50	是	否	第 3 次社会兼职、社会团体、任职时间

续表

字段	类型	字段大小	必填字段	允许空字符串	说明
tc_shh_shh_name1	文本	50	是	否	第1次社会兼职、社会团体、团体名称
tc_shh_shh_name2	文本	50	是	否	第2次社会兼职、社会团体、团体名称
tc_shh_shh_name3	文本	50	是	否	第3次社会兼职、社会团体、团体名称
tc_bzh	备注	-	是	否	备注
tc_pwd	文本	20	是	否	用户密码

infor 表在 Microsoft Access 2007 中的实现如图 8.2 所示。注意：不要忘记设置一个主键。



图 8.2 设置 infor 表字段属性

8.3 页面设计

本实例包括 12 个文件，具体说明见表 8.3。

表 8.3 程序页面列表说明

文件名	说明
index.asp	登录页面
check.asp	用户登录验证，并启动用户操作环境，同时保存用户登录信息
admin.asp	管理员操作页面
admin_admin.asp	账号管理页面，在该页面中实现对管理员账号的修改、删除和新增管理员操作
admin_modify.asp	查询用户页面，在该页面中可以进行删除、修改和浏览用户信息操作
admin_edit.asp	修改用户信息页面
admin_writeto.asp	新增用户信息
admin_del.asp	删除管理员信息、账号
user.asp	用户操作页面
user_admin.asp	修改用户账号和密码页面
user_edit.asp	修改用户个人信息页面
conn.asp	连接后台数据库

admin_edit.asp、admin_writeto.asp 和 user_edit.asp 这 3 个文件的表单设计相同，所以可先设计一个页面，复制表单来完成另外两个网页的页面设计，admin.asp 和 user.asp 的页面导航部分设计也相同，设计出其中一个，复制修改即可完成另一个。下面就其中 5 个主要页面设计进行介绍。读者也可以使用 Dreamweaver CS5 提供的模板功能进行快速设计。

8.3.1 登录页面

登录页面是用户管理系统的大门，任何用户要想进入系统都必须先登录。登录页面(index.asp)的设计效果如图 8.3 所示。该页面使用了 2 个文本框和 1 个下拉菜单对象，具体属性设置见表 8.4。

表 8.4 登录表单设置列表

对 象	名 称	类 型	属性面板设置说明
表单	frm1	表单	【动作】为“check.asp”，【方法】为“POST”
文本框	txt_id	单行	【最多字符数】为 15，【字符宽度】为 20
文本框	txt_pwd	密码	【最多字符数】为 15，【字符宽度】为 20
列表/菜单	shf	菜单	列表值为“一般用户、系统管理员”，【初始化时选定】为“一般用户”
注册	Submit	按钮	【值】为“确定”，【动作】为“提交表单”
取消	Submit2	按钮	【值】为“重置”，【动作】为“重设表单”

在该页面中，“使用帮助”为一个文本链接，单击可弹出一个对话框，显示帮助信息，如图 8.4 所示。它主要是利用 JavaScript 脚本语言的 alert() 函数来实现的，具体代码如下：

```
<a href="#" onClick="window.alert('.....提示信息.....')" class="4">使用帮助</a>
```

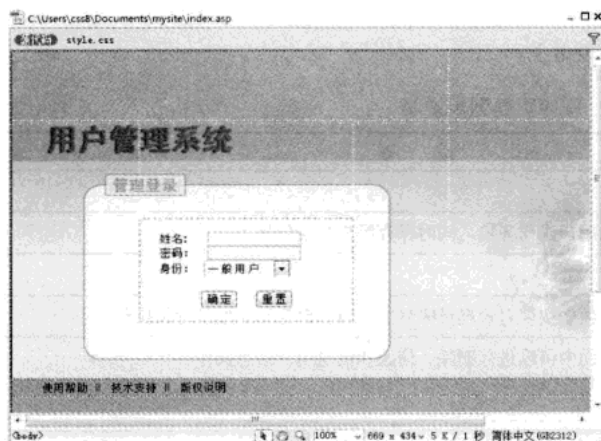


图 8.3 登录页面设计效果



图 8.4 帮助提示信息

8.3.2 导航页面

导航页面包括普通用户导航和管理员导航，其中管理员导航页面(admin.asp)的设计效果如图 8.5 所示，用户导航页面(user.asp)的设计效果如图 8.6 所示。

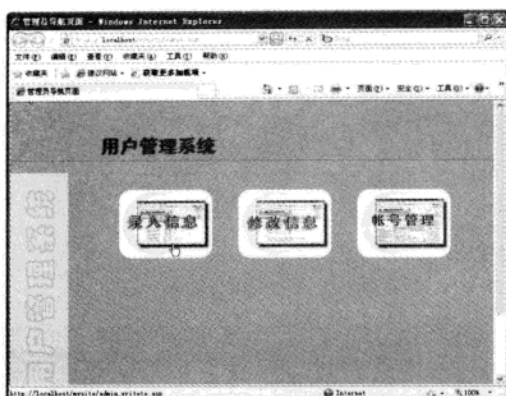


图 8.5 管理员操作页面



图 8.6 用户操作页面

在本页面中利用了 Dreamweaver CS5 提供的“交换图像”行为来实现动态效果。当光标移到导航缩微图上面时，将显示另外一张图像。具体操作步骤如下。

- 01 事先制作两张大小相同的图像，可以用图像编辑器把一张图像的色彩设计成两种效果，即可实现当光标移动图上时，图像立即变色的效果。
- 02 先插入正常状态下显示的图像，然后选中该图像。
- 03 选择【窗口】|【行为】菜单命令，打开【行为】面板，增加【交换图像】行为，设置如图 8.7 所示。

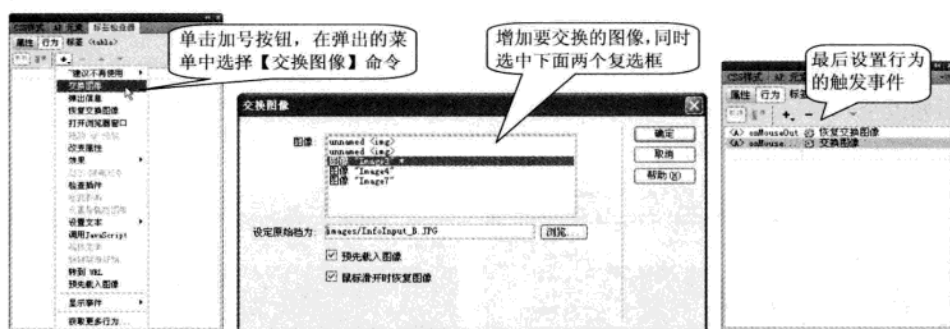


图 8.7 设置交换图像行为

- 04 在【行为】面板中设置触发事件分别为 onMouseOut 和 onMouseOver 即可。

8.3.3 信息操作页面

admin_edit.asp、admin_writeto.asp 和 user_edit.asp 这 3 个文件都是用户信息操作页面。这些页面中的表单设计相同，如图 8.8 所示。在设计表单对象时，建议设置对象名称与 infor 表中各个字段名称相对应，最好名称保持相同，这样即方便操作，也容易识记。由于该表单设计比较简单，这里就不作详细说明，读者可参考光盘中的实例模板。

8.3.4 信息查询页面

管理员信息查询页面 (admin_modify.asp) 分上下两个功能区，上半部分为记录查

询区域, 下半部分为查询记录显示区域, 在 Dreamweaver CS5 中设计如图 8.9 所示。

图 8.8 表单设计效果

图 8.9 查询页面设计效果

8.3.5 用户管理页面

用户管理包括一般用户管理和管理员管理两个页面。一般用户管理主要涉及一般用户登录后修改个人账号, 如图 8.10 所示。管理员管理主要包括对管理员注册信息的修改、删除和添加管理员, 如图 8.11 所示。

图 8.10 一般用户管理页面

图 8.11 管理员管理页面

8.4 用户管理基本功能实现

用户管理基本功能主要包括用户信息的增加、删除和修改等操作, 由于这些基本功能没有复杂的逻辑设计和算法分析, 因此读者只需要在 Dreamweaver CS5 中利用服务器行为即可

快速实现。下面进行具体介绍，同时还将介绍网页查询技巧。

8.4.1 用户登录验证和引导

在第7章中曾经详细介绍了用户登录功能的实现，本例在此基础上介绍如何实现分级用户登录。当用户在 index.asp 登录页面中选择不同的身份，并输入用户名和密码之后，单击【确定】按钮提交表单，系统会跳转到 check.asp 文件中。

check.asp 文件是一个纯代码文件，具体代码如下：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="conn.asp" -->
<%
dim uid ,upwd
'获取并简单处理用户输入的信息
uid=trim(Request.Form("txt_id"))
upwd=trim(Request.Form("txt_pwd"))
uid = Replace(uid, "'", "") '清除单引号
upwd = Replace(upwd, "'", "") '清除单引号
ushf=request("shf")
'判断用户是否登录，否则提示登录
if uid="" or upwd="" then
    response.write "<script language=JavaScript>window.alert('账号和密码不能为空!');window.
location.href='index.asp';</script>"
    response.end
end if
'定义一个记录集对象rs
set rs=server.createobject("adodb.recordset")
'根据登录身份设置不同的SQL字符串和身份标志变量，并引导到不同页面
if ushf="admin" then
    geturl="admin.asp"
    flag="admin"
    sqltext="select * from admin where admin_name='" & uid & "' and admin_pwd='" & upwd & "'"
else if ushf="user" then
    geturl="user.asp"
    flag="user"
    sqltext="select tc_id from infor where tc_name='" & uid & "' and tc_pwd='" & upwd & "'"
end if
end if
'利用上面定义的SQL字符串打开记录集
rs.open sqltext,conn,1,1
'判断记录集是否存在记录
if rs.recordcount >= 1 then
'根据不同身份，定义uid变量的值，该值为表中的主键字段
    if ushf="user" then
        uid = rs("tc_id")
    elseif ushf="admin" then
        uid = rs("admin_id")
    End if
'定义cookie对象分别保存用户输入的信息
    response.cookies("flag") = flag
    response.cookies("uid") = uid
    response.cookies("upwd") = upwd
'跳转到相应的页面
    Response.Redirect geturl
'关闭该记录集
    rs.close
else
```

```

'当用户名和密码不对时,给出提示信息,并返回首页要求重新登录
response.write "<script language=JavaScript>window.alert('账号和密码错误!');window.
location.href='index.asp';</script>"
response.end
rs.close
end if
%>

```

上面代码的第 1 行定义脚本语言为 VBScript,而不是 JavaScript 脚本语言。不同的脚本语言语法规则不同,但是它们的编程方法和设计思路则相同,因此相互之间可以进行转换。

第 2 行代码意思是导入另外一个文件 conn.asp,在 conn.asp 文件中定义并打开了一个数据库连接,详细代码如下:

```

<!--#include file="Connections/conn.asp" -->
<%
'定义变量
dim conn
'定义错误处理机制,如果发生错误则跳转到下一行继续执行
on error resume next
'定义数据库连接对象
set conn=server.createobject("ADODB.CONNECTION")
'打开该数据库
conn.open MM_conn_STRING
%>

```

在上面代码中先用#include 命令包含 conn.asp 文件,该文件是由 Dreamweaver CS5 在【数据库】面板中创建的。然后定义一个 Connection 对象,利用该对象以及数据库连接字符串 MM_conn_STRING (由 Dreamweaver CS5 定义) 打开数据库连接。

8.4.2 用户搜索

搜索实际上就是查询的意思,查询就是利用 SQL 字符串在数据库中筛选满足一定条件的记录,并汇集为一个记录集,存储在内存中供系统随时调用。当搜索某个用户时,可能会存在对要搜索的用户信息不清楚或者模糊现象,因此在设计搜索功能时必须考虑到模糊查询问题。

模糊查询就是给出的查询条件不是非常明晰,例如,当想查询“张三”这个用户的信息时,而不知张三的名字,只知道用户姓“张”,这时就可以利用模糊查询先查找出全部姓张的用户,再进行选择。

在 admin_modify.asp 文件中需要读者设计模糊查询功能,以方便管理员快速找到某个或部分准备操作的用户列表。实现用户搜索功能的具体操作步骤如下。




- 01 在 Dreamweaver CS5 的【文件】面板中,打开 admin_modify.asp 文件。在菜单中选择【窗口】|【绑定】命令,打开【绑定】面板。单击面板左上角的按钮,在弹出的下拉菜单中选择【记录集(查询)】菜单命令。
- 02 打开【记录集】对话框。如果出现的是高级【记录集】对话框,单击【简单】按钮可以切换到简单【记录集】对话框,先在该对话框中简单设置要查询的表和排列顺序。【记录集】对话框具体设置如图 8.12 所示。
- 03 单击【记录集】对话框右侧的【高级】按钮切换到高级【记录集】对话框,如图 8.13 所示。然后修改其中的 SQL 查询字符串,以实现模糊查询功能。
 - 在【记录集】对话框的【参数】设置区域上面单击按钮,可以增加一个新的参数。
 - 单击按钮,可以删除当前选中的参数。



图 8.12 设置简单查询条件

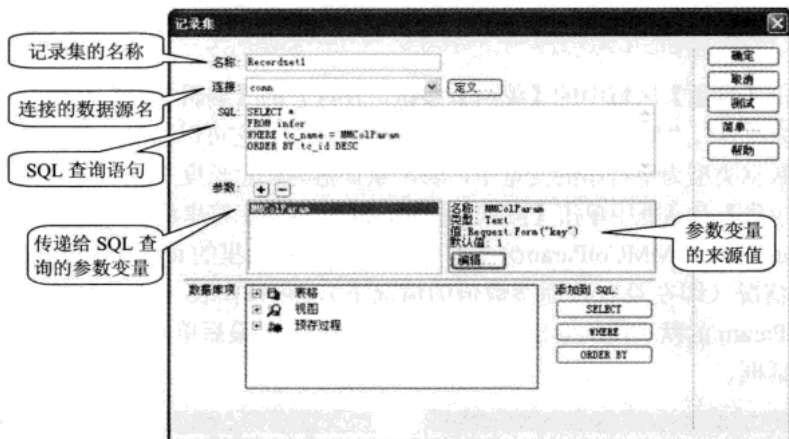


图 8.13 高级【记录集】对话框

- 在【参数】设置区域内单击【编辑】按钮，可以打开【编辑参数】对话框，如图 8.14 所示。在【名称】文本框中可以输入参数的名称，在【类型】下拉列表中可以指定参数的数据类型，其中包括 Numeric（数值型）、Boolean（布尔型）、Text（字符串型）和 Date（日期型）。由于表单和查询字符串所传递的参数值一般都是字符串类型，所以可以设置为 Text；在【值】文本框中，可以输入静态的值，也可以输入 ASP 代码，获取各种参数值，例如 Request.Form("key")就表示从名称为 key 的表单对象中获取值；在【默认值】文本框中可以输入默认值，例如 1，这个值是为了防止表单提交的参数值为空，会造成 SQL 查询错误。

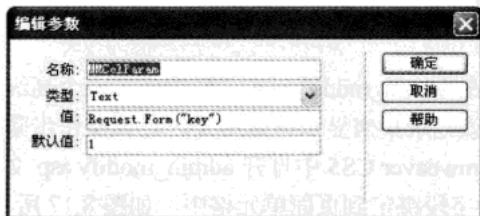


图 8.14 【编辑参数】对话框



当变量对应的数据库值为文本时应该用单引号括起来，而当变量对应的数据库值为数字，则该变量就不用单引号括起来。

- 04** 在【记录集】对话框中修改【SQL】文本框中的 SQL 字符串，修改后的字符串如下：

```
SELECT tc_id, tc_name, tc_dw
FROM infor
WHERE tc_name like %MMColParam%
ORDER BY tc_id DESC
```

在上面 SQL 查询字符串中，把变量 MMColParam 和通配符连接在一起，组合成一个匹配模式，它表示搜索字段中包含 MMColParam 值的记录。修改 SQL 语句后，【记录集】对话框设置如图 8.15 所示。



提示

在 SQL 中百分号 (%) 表示通配符，它可以匹配任意长度的字符串，关键字 LIKE 表示模糊搜索。更详细说明可以参阅第 5.2.2 节介绍。

- 05** 再次单击【参数】区域中的【编辑】按钮，在打开的【编辑参数】对话框中，设置【默认值】为“%”。“%”是标准 SQL 语言中的通配符，它可以匹配任意长度的字符串。通常用在数据类型为字符串的变量中，表示默认值为任意长度字符。
- 06** 在【记录集】对话框中单击【测试】按钮，SQL 查询字符串将会根据 WHERE 子句指定的 tc_name like %MMColParam% 条件搜索数据表 infor 里的 tc_id、tc_name、tc_dw 字段。在默认情况（即在没有传递参数值的情况下），它将在表 infor 中筛选所有等于变量 MMColParam 的默认值的全部记录，如图 8.16 所示。最后单击【确定】按钮关闭【记录集】对话框。

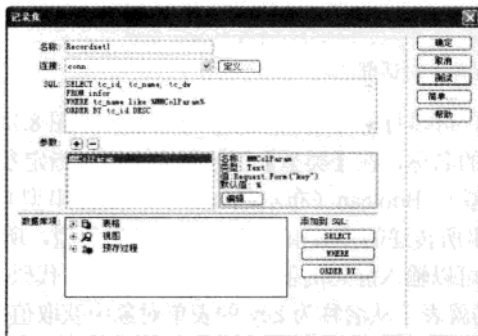


图 8.15 修改模糊查询语句

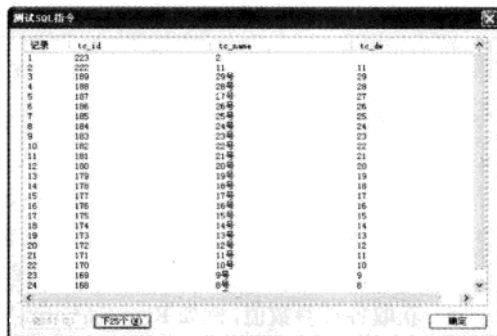


图 8.16 测试 SQL 查询字符串

8.4.3 显示搜索结果

上一小节详细介绍了在 admin_modify.asp 文件中定义一个高级记录集，以实现用户搜索功能。下面介绍如何利用该记录集来显示搜索结果。具体操作步骤如下。

- 01** 绑定记录集。在 Dreamweaver CS5 中打开 admin_modify.asp 文件，把 Recordset1 记录集中的 tc_name、tc_dw 字段绑定到页面单元格中，如图 8.17 所示。



提示

关于本步骤的具体操作方法可以参阅第 9.4.2 节介绍。

- 02** 定义重复显示区域。选择插入单元格所在的行（包含<tr>和</tr>标记），然后选择【窗口】|【服务器行为】菜单命令，打开【服务器行为】面板。单击【服务器行为】面板左上角

的 \oplus 按钮，在弹出的下拉菜单中选择【重复区域】菜单命令，打开【重复区域】对话框，在【记录集】下拉列表中选择所要显示的记录集 Recordset1，在【显示】区域设置每页只显示 10 条记录，如图 8.18 所示。单击【确定】按钮，即可在页面中插入“重复区域”服务器行为。



图 8.17 插入记录集字段



提示

关于本步骤的具体操作和【重复区域】对话框参数介绍可以参阅第 9.4.3 节内容。

- 03** 定义记录集分页显示。逐个选择页面上的导航文本，然后单击【服务器行为】面板左上角的 \oplus 按钮，在弹出的下拉菜单中的【记录集分页】子菜单中选择相应的导航命令，如图 8.19 所示。

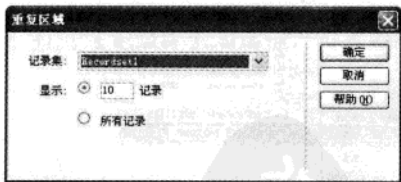


图 8.18 【重复区域】对话框

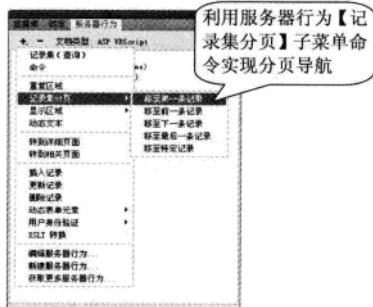


图 8.19 【记录集分页】子菜单



提示

关于本步骤的具体操作和对话框参数介绍可以参见第 9.4.4 节内容。

- 04** 定义分页导航按钮的显示区域。分别选择导航文本（要注意选取区域的完整性，建议切换到【代码】视图进行选取），然后单击【服务器行为】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【显示区域】子菜单命令，如图 8.20 所示，利用这些命令可实现区域的条件显示。



图 8.20 【显示区域】子菜单



关于本步骤的具体操作和对话框参数介绍可以参阅第 9.4.5 节内容。

- 05** 设置无记录时不显示导航条。选取两个导航条所在行 (包含<tr>和</tr>标记), 单击【服务器行为】面板左上角的 \oplus 按钮, 在弹出的下拉菜单中选择【显示区域】|【如果记录集不为空则显示区域】菜单命令, 设置当没有数据显示时就不再显示导航条。
- 06** 设置记录集分页提示信息。切换到【代码】视图, 然后插入下面代码, 也可以利用【绑定】面板中的特定值来插入, 如图 8.21 所示。

```
<td colspan="3">共计<%= (Recordset1 total) %>条记录, 当前显示第<%= (Recordset1 first) %></i>&nbsp;&nbsp;&nbsp;条到第<%= (Recordset1 last) %></i>&nbsp;&nbsp;&nbsp;条</td>
```



图 8.21 设置记录集分页提示信息



如果读者在显示记录集总记录数时显示为-1, 说明当前 ADO 无法确定记录数, 或者 DBMS 或记录集游标类型不支持 RecordCount 属性, 因此读者可以重新定义一个记录集, 在【记录集】对话框中切换到高级状态, 然后在【SQL】文本框中输入下面 SQL 字符串来统计数据表 infor 中的记录数。

```
SELECT Count(tc_id)
FROM infor
```



最后在【绑定】面板中把该记录集的“Expr1000”字段绑定到页面中，或者在页面中输入下面代码：

```
共计<%= (rscount.Fields.Item("Expr1000").Value) %>条记录
```

- 07** 设置表单提交 URL。在前面实例中介绍过在表单的【动作】文本框中输入要提交的 URL 和传递参数。但是在本页中，当浏览者输入查询字符串并提交表单之后，希望在本页面下面显示查询结果，因此在【动作】文本框中不需要输入任何 URL 信息，代码如下：

```
<form name="form1" method="post" action="">
```

这样就表示表单将提交给本文件自己，表单数据将在页内传输，这与页面刷新功能有点相似。


- 08** 实现记录修改功能。选中页面中“修改”超链接文本，在【属性】面板中单击【链接】文本框后边的【浏览文件】图标，打开【选择文件】对话框，选择要跳转到 admin_edit.asp 文件；然后单击【URL】文本框右边的【参数】按钮，打开【参数】对话框，在【参数】对话框中单击按钮，增加一条参数，在【名称】栏中输入“adminid”，该参数用来传递记录的 ID 信息（相当于一个自定义变量）；然后在【值】栏右边单击【动态数据】按钮，打开【动态数据】对话框，在【域】列表框中选择 tc_id 字段（要注意该字段必须被设置为主键，这样才能保证该记录的唯一性）。单击【确定】按钮返回。定义的超链接代码如下：

```
<a href="admin_edit.asp?adminid=<%= (Recordset1.Fields.Item("tc_id").Value) %>" class="list1">修改</a>
```

- 09** 实现记录删除功能。有关删除功能的实现将在后面小节内详细介绍。

8.4.4 修改用户信息

当用户在 admin_modify.asp 文件中单击每条记录后面的“修改”超链接，站点将会跳转到 admin_edit.asp 文件，并通过 adminid 参数把该记录的 ID 值传递过来。下面就介绍如何在 admin_edit.asp 文件中实现对用户信息的修改。具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中，打开 admin_edit.asp 页面，在菜单中选择【窗口】|【绑定】命令，打开【绑定】面板。单击面板左上角的按钮，在弹出的下拉菜单中选择【记录集（查询）】命令。
- 02** 打开【记录集】对话框，设置如图 8.22 所示。

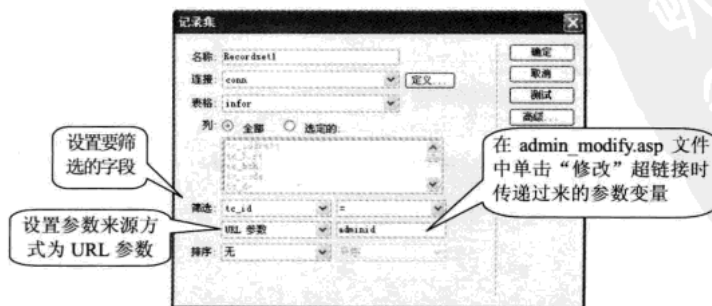


图 8.22 设置【记录集】对话框

03 把记录集中各个字段绑定到页面中对应表单对象上，如图 8.23 所示。由于“性别”项是一个下拉列表，为了使其正确显示，可以在【代码】视图下“列表/菜单”对象上增加下面的条件语句，以实现智能控制。

```
select name="txt_sex">
<% if Recordset1.Fields.Item("tc_sex").Value = "男" then %>
    <option value="男" selected>男</option>
    <option value="女">女</option>
<% else %>
    <option value="男">男</option>
    <option value="女" selected>女</option>
<% end if %>
</select>
```



关于本步骤的具体操作和参数设置请参阅第 9.4.2 节详细介绍。



图 8.23 绑定记录集


04 打开【服务器行为】面板，单击面板左上角的按钮，在弹出的下拉菜单中选择【更新记录】菜单命令，打开【更新记录】对话框，设置如图 8.24 所示。由于表单中各个对象的名称与数据表 infor 中各个字段名称是相同的，所以系统会自动在【表元素】列表中选择每个表元素相对应的数据表字段。



图 8.24 【更新记录】对话框



提示

关于【更新记录】对话框参数说明和设置请参阅第9.5.1节介绍内容。

- 05 单击【确定】按钮，即可插入“更新记录”服务器行为到页面中。如果用户修改完数据，单击【提交】按钮，就可以把修改的数据写入数据库中。

user_edit.asp 文件也是用于实现对一般用户的信息修改，它和 admin_edit.asp 文件的功能实现完全相同（包括表单设计和要操作的数据表），唯一的区别就是参数值来源文件不同，用户可以自己动手试一试，具体操作将不再介绍。



提示

读者也可以把 admin_edit.asp 文件复制一份，然后简单修改页面顶部导航区域的显示内容即可，其他内容不需要修改，最后覆盖 user_edit.asp 文件即可。

8.4.5 用户账号管理

user_admin.asp 文件作为普通用户管理页面，主要功能包括对普通用户账号的基本管理（即修改用户密码），设计效果如图 8.25 所示。

08



图 8.25 普通用户管理页面

具体操作步骤如下。

- 01 打开 user_admin.asp 页面，在菜单中选择【窗口】|【绑定】命令，打开【绑定】面板。单击面板左上角的 按钮，在弹出的下拉菜单中选择【记录集 (查询)】命令，打开【记录集】对话框，设置如图 8.26 所示。
- 02 把记录集中 tc_name 和 tc_pwd 字段绑定到页面相应位置上，其中 tc_name 字段插入到“用户名：”后面，而 tc_pwd 字段绑定到文本框的 value 属性上，如图 8.25 所示。
- 03 插入“更新记录”服务器行为。在【服务器行为】面板中选择【更新记录】命令，在打开的【更新记录】对话框中根据图 8.27 所示进行设置。详细说明可以参阅第 9.5.1 节（注意，要插入“更新记录”服务器行为，必须先定义一个记录集）。
- 04 单击【确定】按钮关闭【更新记录】对话框即可。



图 8.27 设置【更新记录】对话框

用户管理的复杂性主要体现在逻辑分析上，一个简单的功能可能很容易实现，但是如果把几个简单的逻辑组合在一起就会形成复杂的功能。复杂的功能会让初学者感觉高不可攀。学会逻辑分析就是要把复杂的问题简单化，把复杂的功能拆解成很多个单一的功能逐一实现，那么一切难题也就迎刃而解了。一般高级查询都要涉及多种条件的组合和嵌套查询，因此管理页面往往需要建立多个记录集，并相互牵制，初看会很难，也很容易一错皆错，所以用户在设计时要特别小心，思路要清晰。

在 `admin_admin.asp` 文件中利用服务器行为实现对管理员账户进行管理，例如，修改、增加和删除账户。具体步骤如下。

记录集

名称: rs_del

连接: conn

表格: aMain

列: ☒ 全部 ☐ 选定的:

aMain.id
aMain.pname
aMain.ppt

筛选: 无

排序: 无 下移

确定
取消
测试
高级...
帮助

记录集

名称: rs_mod

连接: Conn

表格: admin

列: ☒ 全部 ☐ 选定的

id	name	password
000001	admin	123456
000002	admin	123456
000003	admin	123456

筛选: admin_id = Cookie

排序: 无

确定 取消 测试 高级 帮助

图 8.29 定义 rs mod 记录集

02 在修改区域中，分别绑定 admin 记录集中的 admin_name 和 admin_pwd 字段，如图 8.30 所示。

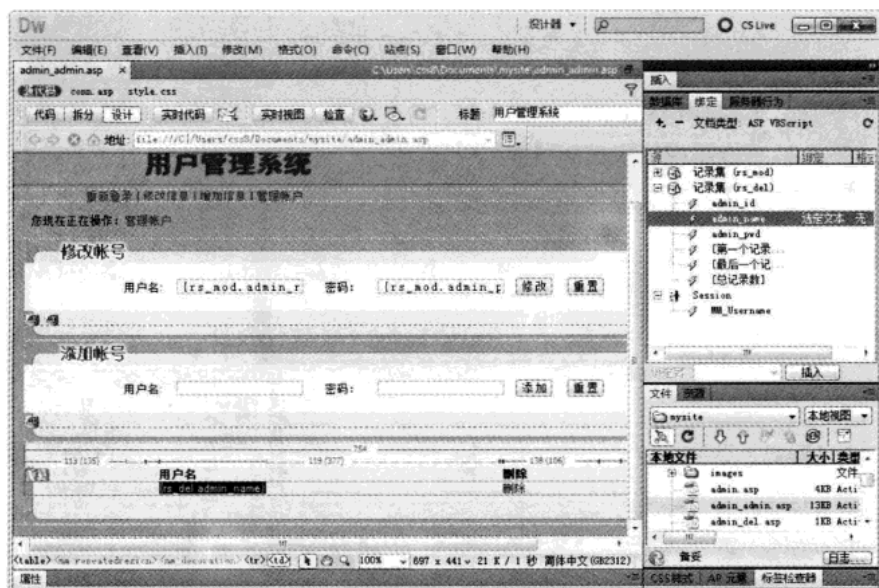


图 8.30 绑定修改功能字段

- 03 在【服务器行为】面板中增加“更新记录”服务器行为，以实现在 admin_admin.asp 页面中能够修改管理员账户。【更新记录】对话框设置如图 8.31 所示。



提示

关于【更新记录】对话框参数说明和设置可以参阅第 9.5.1 节内容介绍。

- 04 在【服务器行为】面板中增加“插入记录”服务器行为，以实现在 admin_admin.asp 页面中可以设置新的管理员账户，【插入记录】对话框设置如图 8.32 所示。

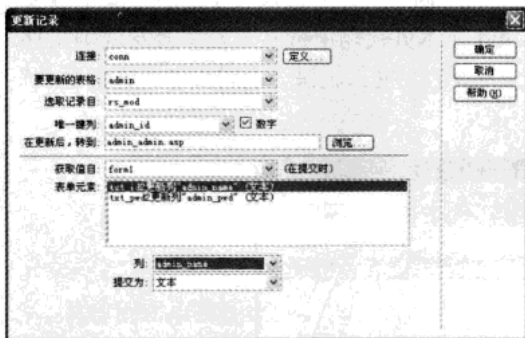


图 8.31 增加更新记录服务器行为

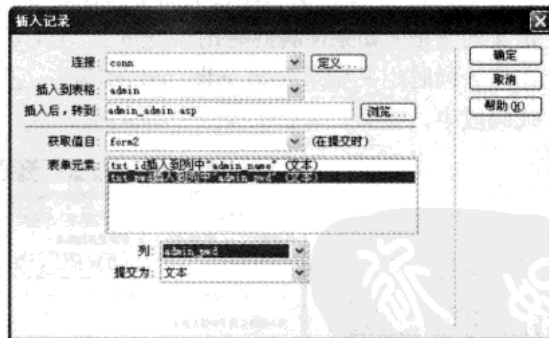


图 8.32 增加插入记录服务器行为



提示

在设置对话框时，读者一定要注意【获取值自】选项的设置，更新记录使用的表单是 form1，插入记录使用的表单是 form2。

关于【插入记录】对话框参数设置可以参阅第 9.4.6 节详细介绍。

- 05 把 rs_del 记录集中的 admin_name 字段插入到删除区域中的“用户名”单元格。然后选中该行，在【服务器行为】面板中增加“重复区域”服务器行为，设置显示所有记录，如图 8.33 所示。要注意在【重复区域】对话框中选择正确的记录集名称。



图 8.33 删除用户功能实现

8.5.2 纠正 Dreamweaver 服务器行为存在的 Bug

再聪明的机器人也赶不上人的大脑，因为机器人始终是按着事先定义好的逻辑进行各种操作。Dreamweaver CS5 虽然是功能非常完善的网页编辑器，但是当读者不断深入学习时，就会发现它也很机械，甚至愚蠢。

在上节操作中，我们完成了高级账户的管理设计，现在如果读者按 F12 键在浏览器中预览 admin_admin.asp 文件，会发现页面显示如图 8.34 所示的错误信息。

“名称重定义”说明在该页面中存在变量重复定义的问题，在“错误类型”后面显示重复定义的变量，如果读者此时用该变量名 MM_editCmd 在本页代码中搜索一下，会发现确实存在两处同时定义变量 MM_editCmd 的情况。它们分别是在“插入记录”代码段和“更新记录”代码段中。



图 8.34 浏览器显示错误信息

由于 Dreamweaver CS5 在插入服务器行为时，它都会自动定义变量，当在一个页面插入一个服务器行为时，可能还没事，但是如果在一个页面内插入多个服务器行为时，就会出现这个问题。因为按照 VBScript 脚本语言的语法规则，同一个变量在同一个页面内是不能够重复定义的，而 Dreamweaver 不能够灵活地解决这个问题。

要解决此类问题，读者只能切换到【代码】视图，查找到第 2 处变量定义的语句，删除即可。以后如果再出现此类错误页面，读者也可以按这种方法来解决。在后面综合实例中经常会在同一个页面定义多个服务器行为，也不断在一个页面内使用#include 语句包含另一个文件，多个文件一旦包含在同一个文件中，它们就属于同一个程序级，因此重复定义变量问题就更加突出。所以读者在包含文件时，一定要小心这个问题。笔者坚持的原则是：包含文件不定义变量，除非特殊专用变量外，同时包含文件不使用@命令，因为同一个程序级内只能够有使用一个@命令，最常使用的@命令是：<%@LANGUAGE="VBSCRIPT"%>。

另外，本页还有一个重复定义变量 MM_editRedirectUrl，读者可以搜索删除第 2 处变量定义。

请注意，一旦读者删除这些自动定义的变量，Dreamweaver CS5 就会不能识别该变量所在代码段表示的服务器行为，如果读者在【服务器行为】面板中，就会发现第 2 处删除变量代码段所表示的服务器行为已经在【服务器行为】面板中消失，如图 8.35 所示。如果能让 Dreamweaver CS5 能够识别该服务器行为（只有这样读者才能够可视化修改该服务器行为），必须在对应的位置补全该变量的定义。因此，为了方便日后修改服务器行为，建议读者不要直接删除重复定义的变量，而是在该变量的前面加上单引号（'），使该变量成为注释的一部分，同时要适当加上补充说明，方便日后能够看得懂。例如：

```
' execute the insert
''' Dim MM_editCmd '系统重复定义变量
' append the query string to the redirect URL
'''Dim MM_editRedirectUrl '系统重复定义变量
```

在上面代码中，在前面加入 3 个单引号，其目的是方便日后能够快速搜索到。



图 8.35 消失的“插入记录”服务器行为




在设计动态网页时，读者经常会遇到各种各样的错误，学会阅读浏览器显示的“错误类型”提示信息，该信息是解决错误的最好最实用的线索，希望读者重视。

8.5.3 账户高级删除

使用“删除记录”服务器行为可以删除记录，但是它只能删除简单的记录。现在，我们要把整个用户管理系统内的所有用户的删除操作都集中到一个文件中完成。初看起来，这是一个非常复杂的操作，因为不同的用户信息保存在不同数据表中，如何进行判断，怎么能够保证删除不会出现误操作。这些都是本节要解决的难题。为了能够分辨用户的身份，可以这样进行设计：为不同身份的用户删除操作附加一个查询字符串参数，根据该参数变量值来分辨删除操作来自哪个文件，以及删除账户的身份。

具体操作步骤如下。

- 01 在 Dreamweaver CS5 中，打开 admin_admin.asp 页面，选中“删除”超链接文本，在【属性】面板中单击【链接】文本框后边的【浏览文件】图标，打开【选择文件】对话框。
- 02 选择要链接到的文件，这里选择 admin_del.asp 文件，设置【URL】文本框要传递的参数如下：

```
admin_del.asp?adminid=<%=rs_del.Fields.Item("admin_id").Value)%>&grade=admin
```

该 URL 中包含两个参数，分别是 adminid 和 grade，其中 adminid 参数存储将要删除记录的 ID 值，grade 参数存储用户的身份或级别。两个变量之间用“&”符号连接起来，设置如图 8.36 所示。

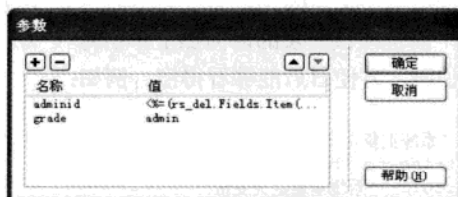


图 8.36 【参数】对话框

- 03 打开 admin_modify.asp 页面，选中“删除”超链接文本，在【属性】面板中的【链接】文本框中输入下面代码：

```
admin_del.asp?adminid=<%(Recordset1.Fields.Item("tc_id").Value)%>&grade=user
```

- 04 在 admin_del.asp 文件中输入下面代码，实现删除指定数据表中指定记录：

```
<%@LANGUAGE="VBScript" CODEPAGE="936" %>
<!--#include file="Connections/conn.asp" -->
<%
dim adminid, grade, sql, reurl, comm
'获取传递过来的id参数值
adminid = Request.QueryString("adminid")
grade = Request.QueryString("grade")
'用if条件判断用户身份，并根据身份定义不同的SQL查询字符串
if grade = "admin" then
    sql = "delete from admin where admin_id = " & adminid
    reurl = "admin_admin.asp"
elseif grade = "user" then
    sql = "delete from infor where tc_id = " & adminid
    reurl = "admin_modify.asp"
end if
Set comm = Server.CreateObject("ADODB.Command")
comm.ActiveConnection = MM_conn_STRING
comm.CommandText = sql
'删除查询到的记录
```

```

On error resume next
comm.Execute()
comm.close
set comm = nothing
'然后跳转到相应的文件
Response.Write("<script language=javascript>window.alert('记录删除成功!');window.
location.href='& reurl & ';</script>")
Response.End()
%>

```

8.5.4 录入用户信息

打开 admin_writeto.asp 文件, 然后在【服务器行为】面板中选择“插入记录”服务器行为, 【插入记录】对话框设置如图 8.37 所示。

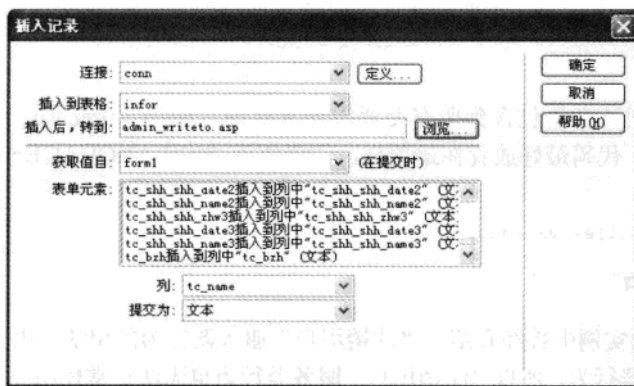


图 8.37 【插入记录】对话框

单击【确定】按钮, 关闭【插入记录】对话框即可在 admin_writeto.asp 文件中实现插入记录的功能, 也即实现普通用户信息录入功能。

8.5.5 用户身份分级限制

Dreamweaver CS5 提供了“限制对页的访问”服务器行为, 但是在使用该服务器行为之前, 读者应先定义“登录用户”服务器行为。由于没有定义“登录用户”服务器行为, 使用“限制对页的访问”服务器行为时系统会显示错误。

为了增强用户管理系统的安全性, 下面用手写代码的方式来实现。本系统涉及多级用户的管理, 不同级别的用户只能进入自己权限范围内的页面。所以在设计时, 可以定义两个文件, 分别定义针对不同级别用户的限制, 然后用#include 语句分别把这两个文件包含到不同文件中。具体实现步骤如下。

01 新建 validate_admin.asp 文件, 在该文件中输入下面代码, 用来限制非管理员用户不能进入:

```

<%
dim flag
flag = Request.Cookies("flag")
if flag <> "admin" then
    Response.Write("<script language=javascript>window.alert('你不是管理员, 没有权限进入!');window.location.href='index.asp';</script>")
    Response.End()

```

```
End if
%>
```

- 02** 用下面语句把该文件包含到所有管理员文件中，例如，admin.asp、admin_admin.asp、admin_del.asp、admin_edit.asp、admin_modify.asp 和 admin_writeto.asp，代码最好放在最顶部，一般位于<%@LANGUAGE="VBSCRIPT"%>命令行的下面。

```
<!--#include file="validate_admin.asp" -->
```

- 03** 新建 validate_user.asp 文件，在该文件中输入下面代码，用来限制非登录用户不能进入。

```
<%
dim flag
flag = Request.Cookies("flag")
if flag <> "user" and flag <> "admin" then
    Response.Write("<script language=javascript>window.alert('你不是登录用户，没有权限进入!');window.location.href='index.asp';</script>")
    Response.End()
End if
%>
```

- 04** 用下面语句把该文件包含到所有普通用户文件中，例如，user.asp、user_admin.asp 和 user_edit.asp，代码最好放在最顶部，一般位于<%@LANGUAGE="VBSCRIPT"%>命令行的下面。

```
<!--#include file="validate_user.asp" -->
```

8.5.6 注销用户

在前面章节的实例中曾经介绍了“注销用户”服务器行为的用法。由于本实例没有定义“登录用户”服务器行为，所以“注销用户”服务器行为也无法正常使用。不过用手写代码也比较简单，实现步骤如下。

- 01** 新建 exit.asp 文件，输入下面代码：

```
<%@LANGUAGE="VBSCRIPT"%>
<%
'清空cookie对象保存的用户信息
response.cookies("flag") = ""
response.cookies("uid") = ""
response.cookies("upwd") = ""
Response.Redirect("index.asp")
%>
```

- 02** 在每页导航条中增加退出系统超链接即可。

8.5.7 管理员分级管理

在初步设计时，管理员为多人制且级别相同，任一合法管理员均可对管理员账号进行添加、删除及修改操作。这样的设计存在一个很大的弊端，例如，在图 8.38 所示的界面中，由于多名管理员都拥有相同的权限，这样就会出现：A 管理员删除 B 管理员的账户，而 C 管理员又错删了 B 管理员的账户，如此一来，整个系统就会出现管理混乱。因此，读者有必要增加管理员分级管理功能。定义一个高级管理员，让高级管理员拥有增加和删除普通管理员账户的权限，而普通管理员只能管理自己的账户。

具体实现步骤如下。

- 01** 在数据库 admin.mdb 的 admin 数据表中新增字段 admin_grade，【数据类型】为“数值”，【字段大小】为“整型”，并设置【默认值】为“0”。

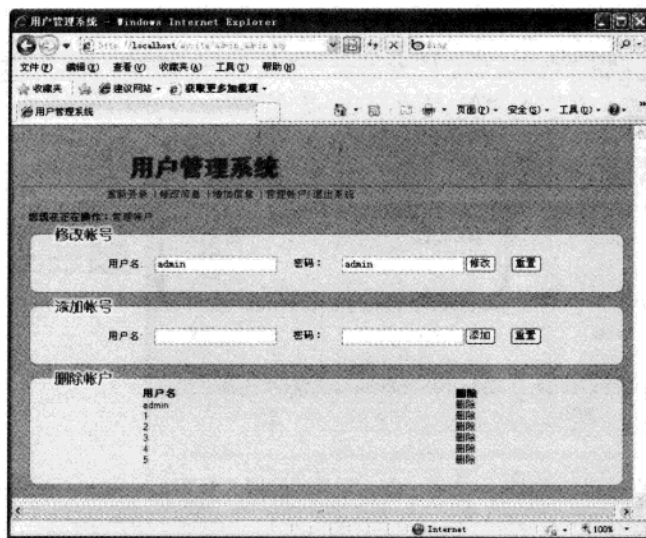


图 8.38 管理员账户管理界面

- 02 在 admin 数据表中增设一个高级管理员“admin”，设置该记录的“admin_grade”字段值为 9，其他管理员的该字段值都设置为 0。



提示

这样读者就可以利用该字段为管理员划分出很多级别，细化每个级别所拥有的权限。

- 03 打开 admin_admin.asp 文件，在该文件中重新定义一个记录集 rs_9，具体设置如图 8.39 所示。该记录将根据用户登录的 ID 信息，以及该字段 admin_grade 值是否为 9 筛选数据表 admin 中的数据。



图 8.39 定义记录集

- 04 选中“增加账号”和“删除账号”功能区域，注意要选中完整的表格，建议在【代码】下精确选取。然后在【服务器行为】面板中增加“如果记录集不为空则显示”服务器行为，在打开的【如果记录集不为空则显示】对话框中选择“rs_9”记录集，如图 8.40 所示。
- 05 单击【确定】按钮，关闭【如果记录集不为空则显示】对话框，这样如果登录管理员的级别不为 9，就不会显示“增加账号”和“删除账号”功能区域，也就无法操作了。



图 8.40 定义显示区域

8.6 实践总结

本章介绍了用户管理系统功能的实现，有关用户管理系统技术细节还有很多，本章仅就基本功能进行探讨。另外，使用服务器行为虽然很方便，但从代码编写的角度分析，实际是不经济的，也缺乏灵活性和适应能力。例如，设置当用户没有登录时就不能访问其他页面，实际上在每页的最顶部输入下面代码即可：

```
<%
//判断是否登录，该变量是在登录时赋予的一个值
if request.Cookies("flag") <> "admin" then
response.write "<script language=JavaScript>{window.alert('网络超时或你还未登录，请重新登
录!');window.location.href='index.asp';}</script>"
response.end
end if
%>
```

用户管理的核心是角色管理，角色划分是否恰当，权限分配是否合理，直接影响一个网站管理水平，有关更多技术细节还会在后面实例中继续探讨，希望用户去慢慢试验琢磨。



留言板是动态网站中最基本的组成模块，它是客户与网站之间进行交流的一种最简洁、最有效的方式。本章将详细地介绍如何制作具有基本功能的留言板，同时对留言板后台管理和一些常用扩展功能展开介绍。读者也可以利用本章介绍的方法在自己的网站内组合更具个性的留言板功能。留言板涉及各种服务器行为的使用技巧，例如，从定义记录集、重复显示、记录集分页到各种记录操作，因此，本章也是读者全面学习使用 Dreamweaver CS5 的服务器行为最好的一章。

技术要点

- 定义和绑定记录集
- 定义重复区域
- 记录集分页
- 定义显示区域
- 设置动态属性
- URL 参数应用
- 删除记录
- 更新记录

9.1 程序设计分析

留言板是网站与客户进行交流的一种基本方式，早期的互联网上，留言板很受欢迎，功能也相对简单。随着 Web 技术的发展，留言板已演变成为多种形式，如论坛、新闻发布、博客等。特别是博客在中国的发展比较迅速，博客技术被称为是第 4 次互联网革命。

留言板形式多样，小巧方便，它可以附在主题后面，用来收集浏览者的个人评价或反馈信息；也可以附在站点底部，实现与站长的交流；甚至可以开发一个功能完善的留言板系统。一般留言板都包含写留言、显示留言和管理留言等几个模块，管理又可以包含删除、修改、回复等功能。本章留言板的设计流程如图 9.1 所示。

本章实例涉及留言板的基本功能，包括书写留言、存储留言、显示留言、留言回复、留言审核、删除留言、修改留言和退出留言等功能。要完成本章用户留言系统，读者应该学习和掌握下面服务器技术。

- 书写留言，利用“插入记录”服务器端行为来实现。

- 显示留言，可以先创建记录集，然后用“重复区域”、“记录集分页”和“显示区域”服务器端行为来实现留言显示功能。
- 删除留言，需要把删除记录信息传递给删除文件，然后利用该文件中的删除代码删除记录。
- 修改留言，需要把修改的记录信息传递给修改表单，在表单中显示出来，修改之后再“更新记录”服务器行为来更新数据库。
- 利用服务器行为“用户登录”和“注销用户”来实现用户登录与退出功能。

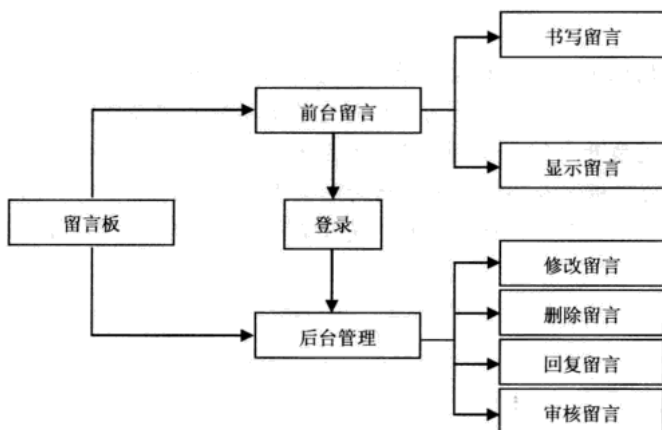


图 9.1 留言板流程图



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，如 site（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”（详细设置可以参阅第 2.5.2 节介绍）。

把本书光盘中的“9”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

9.2 数据库设计

数据库设计是 Web 开发中非常重要的环节，由于大量的网站信息都会存储在数据库中，并且使用数据库来完成对信息的读取和处理等操作，所以读者需要注意数据库的设计和结构优化。本示例数据库 feedback.mdb 包含两个数据表：admin 和 feedback。

9.2.1 admin 表

admin 表用来记录用户登录信息，包括用户名和密码。该表数据结构设计见表 9.1。

表 9.1 admin 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
id	自动编号	-	-	-	自动编号
admin_name	文本	100	是	否	用户名
admin_pass	文本	200	是	否	密码

admin 表在 Microsoft Access 中实现的效果如图 9.2 所示。

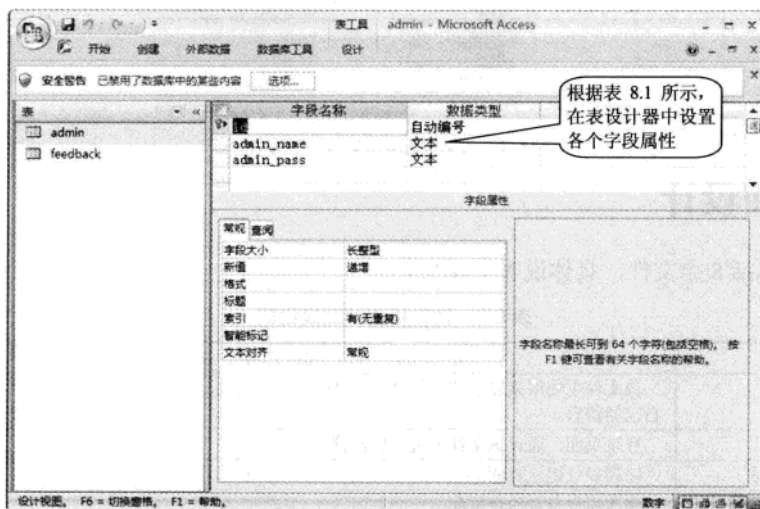


图 9.2 设置 admin 表字段属性



提示

不要忘记设置 id 字段为主键, 方法是: 右键单击 id 字段左侧的边框, 在快捷菜单中选择【主键】命令即可。

最后, 在 admin 表中输入用户名为 admin, 密码为 admin。

9.2.2 feedback 表

feedback 表主要存储用户留言内容, 该表数据结构设计见表 9.2。

表 9.2 feedback 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
id	自动编号	-	-	-	自动编号
title	文本	200	是	否	留言标题
content	备注	-	是	否	留言内容
weather	文本	50	是	否	当天天气
date	日期/时间	-	是	否	发表日期
week	文本	50	是	否	星期

feedback 表在 Microsoft Access 中设置如图 9.3 所示。由于 date 字段是日期/时间类型数据, 可以在【默认值】中输入内置函数 Date()。实现当增加一条记录时, 如果不明确指明该字段的日期, 则系统就会自动以当前日期填写该字段。



图 9.3 设置 feedback 表字段属性

9.3 页面设计

本实例包括 8 个文件，具体说明见表 9.3

表 9.3 留言板文件列表说明

文件名	说 明
index.asp	该文件主要用来显示留言信息，用户可以查看浏览，或者单击“增加留言”导航菜单增加自己的留言
admin.asp	登录页面，通过该文件可进入后台管理
admin_error.asp	错误登录提示页面
rewrite_admin.asp	修改用户登录信息页面
edit_diary.asp	后台管理主页面，在该页面中可以进行删除、修改和浏览留言操作
add_diary.asp	后台书写留言页面，主要涉及表单设计
del_diary.asp	该文件包含删除留言记录的代码
rewrite_diary.asp	修改留言页面，主要涉及到表单设计

下面就几个主要页面设计进行介绍。

9.3.1 显示留言页面

index.asp 页面模板设计效果如图 9.4 所示。启动 Dreamweaver CS5，选择【窗口】|【文件】菜单命令，在【文件】面板中打开 index.asp 文件，页面设计如图 9.5 所示。在光盘实例模板文件夹中可以直接引用。

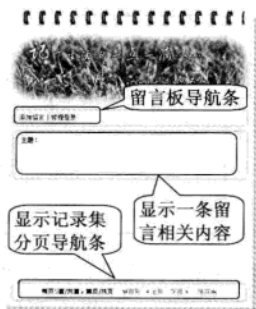


图 9.4 index.asp 模板文件浏览效果

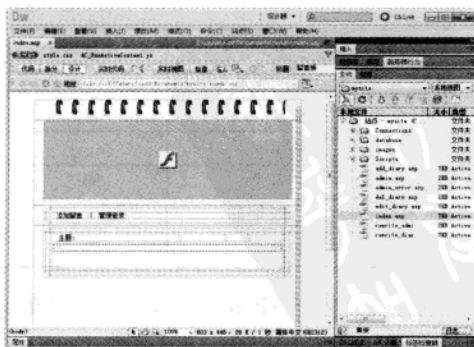


图 9.5 index.asp 页面设计

9.3.2 编辑留言页面

edit_diary.asp 模板页面设计最后效果如图 9.6 所示。在【文件】面板中打开 edit_diary.asp 文件，读者可以修改该页面的设计效果，如图 9.7 所示。

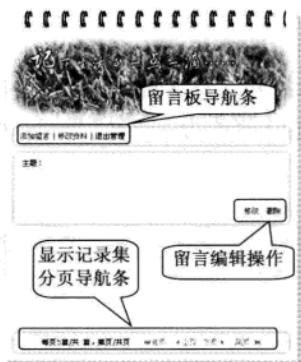


图 9.6 edit_diary.asp 模板文件浏览效果

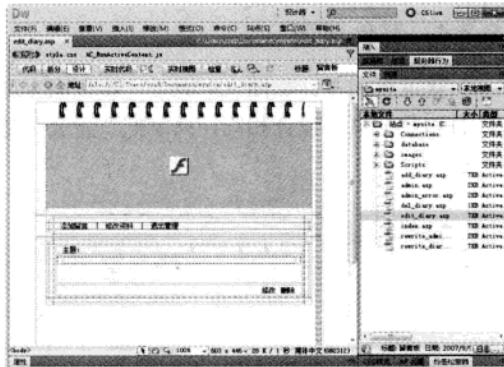


图 9.7 edit_diary.asp 页面设计

9.3.3 增加留言页面

add_diary.asp 页面主要涉及表单的设计，本页面表单包括 2 个文本框、2 个列表/菜单和 2 个按钮。下面列出本页表单中各个对象属性设置，见表 9.4。

表 9.4 add_diary.asp 页面表单对象设置

对象	名称	类型	属性面板设置说明
表单	form	表单	设置【方法】为“POST”，在【动作】文本框中可以设置要提交表单数据到的页面地址，用户可以先设置，当插入服务器行为后会自动设置一个变量 <code><%MM_editAction%></code>
文本框	title	单行	标题文本框的【字符宽度】设置为 20，【最多字符数】为 20
文本框	textarea	多行	内容文本框的【字符宽度】设置为 60，【行数】为 12，【换行】为“默认”
列表/菜单	weather	菜单	列表值设置可参见下面代码
列表/菜单	week	菜单	列表值设置可参见下面代码
按钮	Submit	提交	【值】为“发布”，【动作】为“提交表单”
按钮	Submit	重设	【值】为“清空”，【动作】为“重设表单”

有关下拉菜单的列表值设置代码如下，整个表单设计效果如图 9.8 所示。

```
&nbsp;天气:
<select name=weather class=yecao>
  <option value="晴天" selected>晴天</option>
  <option value="阴天">阴天</option>
  <option value="下雨">下雨</option>
  <option value="下雪">下雪</option>
  <option value="雷雨">雷雨</option>
  <option value="刮风">刮风</option>
</select>
&nbsp;星期:
<select name=week class=yecao>
  <option value="星期一" selected>星期一</option>
  <option value="星期二">星期二</option>
```

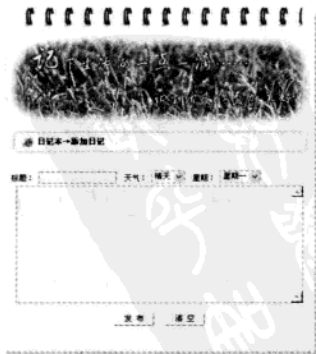


图 9.8 表单设计效果

```

<option value="星期三">星期三</option>
<option value="星期四">星期四</option>
<option value="星期五">星期五</option>
<option value="星期六">星期六</option>
<option value="星期日">星期日</option>
</select>

```

9.3.4 其他页面

登录页面设计如图 9.9 所示。该页面主要由 2 个文本框和 2 个按钮组成。登录信息修改页面和登录页面设计相同，只要把“登录”提交按钮的【值】修改为“修改”即可，如图 9.10 所示。



图 9.9 登录页面设计效果



图 9.10 登录信息修改页面设计效果

登录错误提示页面设计如图 9.11 所示。修改留言页面和增加留言相同，可把 add_diary.asp 另存为 rewrite_diary.asp 文件，然后把提交按钮的【值】改为“修改”，如图 9.12 所示。



图 9.10 登录错误提示页面设计效果

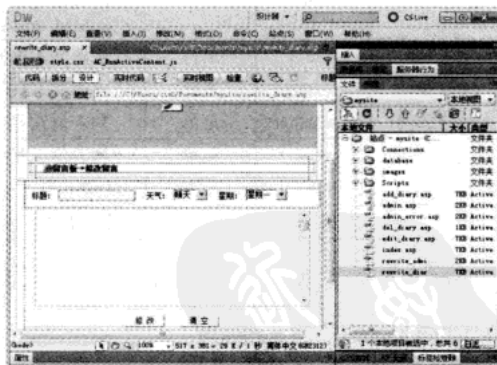


图 9.12 修改留言页面设计效果

9.4 留言板基本功能实现

用 Dreamweaver CS5 设计动态网页时，操作的核心是如何建立页面与数据库的连接，然后从中查询需要显示的数据，并把它绑定到页面中预显示的区域。这个过程被 Dreamweaver 分为定义记录集、绑定记录集、控制记录集的显示和管理记录集等几个主要操作步骤。下面详细介绍这个过程的实现方法。

9.4.1 定义记录集

动态网页的实质就是把后台数据库和前台页面捆绑在一起,那么数据究竟是怎么绑定在网页上呢?其实数据是通过记录集(Recordset, 查询)对象来实现的,而不是页面直接读取数据库中的数据,用户的操作也不是直接来读写数据库。记录集是一个临时的数据表,它是根据 SQL 查询字符串从数据库中查询所得到的数据,并临时保存在内存中。



提示

记录集可包括数据库表的所有记录和字段。由于服务器一般把记录集暂时存在内存中,所以应使记录集尽可能地小,定义记录集的基本策略是仅仅查询应用程序所需要的数据。当不再使用时应立即释放记录集在内存中占据的空间,以便提高服务器的性能。

本实例有 4 个文件需要定义记录集,包括 index.asp、edit_diary.asp、rewrite_diary.asp 和 rewrite_admin.asp,其中 edit_diary.asp 需要定义两个记录集。下面以 index.asp 文件为例介绍记录集的定义方法,具体操作步骤如下。

- 01 在 Dreamweaver CS5 中,打开 index.asp 页面。如果在【数据库】面板中如果没有任何数据源显示,要先定义站点数据源。
- 02 在菜单中选择【窗口】|【绑定】命令,打开【绑定】面板。单击面板左上角的 按钮,在弹出的下拉菜单中选择【记录集(查询)】菜单命令,如图 9.13 所示。
- 03 打开简单【记录集】对话框,如图 9.14 所示。如果显示的是高级【记录集】对话框,单击【简单】按钮可以切换到简单的【记录集】对话框。

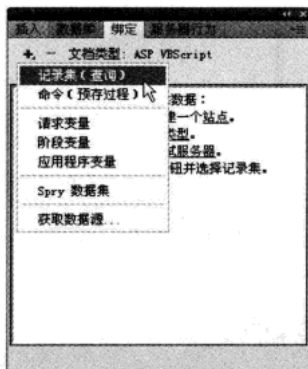


图 9.13 选择记录集

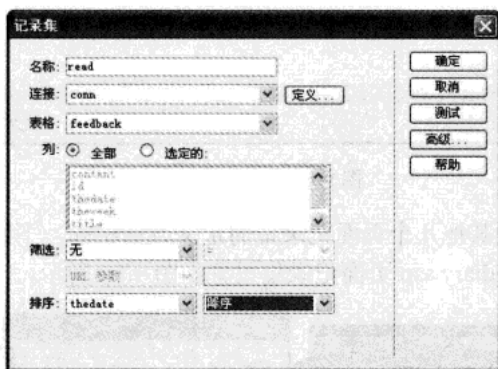


图 9.14 【记录集】对话框

- 04 在【名称】文本框中输入记录集的名称,如 read,该记录集名称可以方便在其他地方引用记录集,实际上它就是 Recordset 对象的实例变量名。然后在【连接】下拉列表框中选择一个连接(该连接将向记录集提供数据的数据源),如 conn。
- 05 在【表格】下拉列表框中选择数据源,即数据库中的要查询的表,如 feedback。
- 06 在【列】区域中,选择要包括在记录集中的表格字段,由于本例比较小,不会对内存有多大影响,因此选择【全部】单选按钮,选择 feedback 表中全部字段。通常,记录集的字段越多,可以显示更多的详细信息。



提示

若要使记录集中只包括某些表字段,请单击【已选定】单选按钮,然后按住 Ctrl 键或者 Shift 键并单击列表中的字段名,以选择所需字段。

- 07 在【筛选】下拉列表选择一个字段作为查找和显示的条件字段。例如，如果 URL 查询字符串参数中包含一个记录的 ID 号，则选择包含记录 ID 号的字段。筛选实质上是 SQL 查询字符串中的 WHERE 子句的设置。本例没有涉及条件查询，因此将在后面实例中详细介绍。
- 08 在【排序】下拉列表选择一个字段作为排序的依据，然后在其右边选择排序的方式，是升序还是降序。本页记录集设置 thedate 字段进行降序排序，这样就把最新留言显示在最前面。



提示

如果单击【记录集】对话框右边的【测试】按钮，可以测试记录集是否查询成功，并可查看结果，如图 9.15 所示。

- 09 设置完毕，单击【确定】按钮，记录集随即出现在【绑定】面板中，如图 9.16 所示。



图 9.15 测试记录集

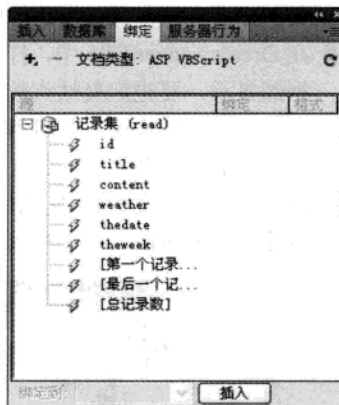


图 9.16 定义记录集

有关其他几个页面记录集的定义方法相同，下面列出各个页面记录集的设置。
edit_diary.asp 页面记录集设置如图 9.17 所示。



图 9.17 edit_diary.asp 页面记录集定义

rewrite_admin.asp 页面记录集设置如图 9.18 所示。在 rewrite_admin.asp 页面记录集中用到了条件筛选，它是根据用户单击编辑留言 (edit_diary.asp) 页面中的“修改”超链接文本

传送的查询字符串参数来确定要修改的登录用户记录。

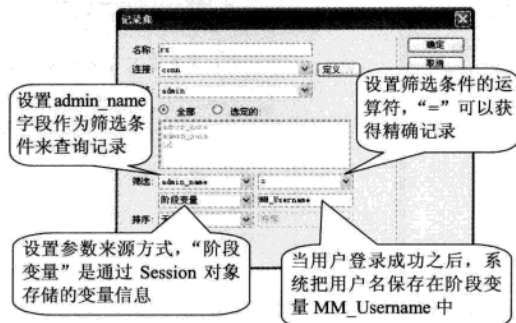


图 9.18 rewrite_admin.asp 页面记录集定义

rewrite_diary.asp 页面记录集定义结果如图 9.19 所示。

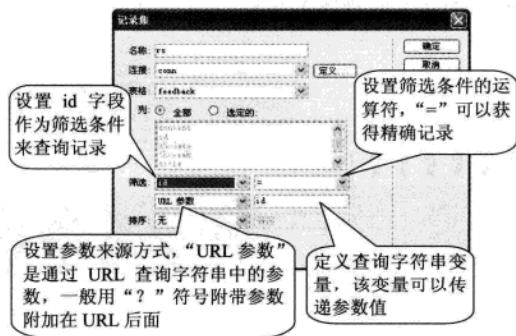


图 9.19 rewrite_diary.asp 页面记录集定义



提示

当用户在表单的【属性】面板中, 设置<form>标记的【方法】为“GET”选项, 则该表单被提交时, 表单域中被提交的值也是通过 URL 查询字符串参数来传递。例如, http://localhost/mysite/rewrite_diary.asp?id=35, 边框中的字符就是 URL 查询字符串参数。

9.4.2 绑定记录集

绑定记录集就是把用户定义记录集中的字段插入到页面中, 或者作为 HTML 标记的属性值以实现页面的动态显示效果。

1. 把记录集直接插入页面

定义完记录集, 下一步就需要把这些记录集的字段绑定到页面中, 仍然以 index.asp 文件为例讲解, 具体操作步骤如下。

- 01 在 Dreamweaver CS5 中, 打开 index.asp 文件。在菜单中选择【窗口】|【绑定】命令, 打开【绑定】面板。单击 read 记录集左边的加号展开记录集, 如图 9.20 所示。
- 02 把光标定位到要插入标题的单元格, 然后在【绑定】面板中选定 read 记录集中的 title 字段, 单击面板底部的【插入】按钮, 即可把 title 字段插入到页面中, 如图 9.21 所示。



图 9.20 展开记录集

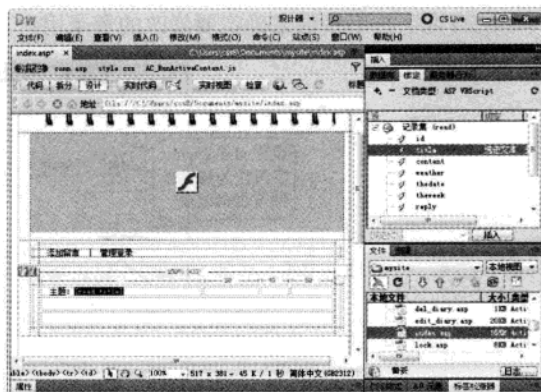



图 9.21 绑定标题字段

03 以同样的方式把日期、天气、星期和内容绑定到页面中，如图 9.22 所示。

04 单击【文档】工具栏中的【实时视图】按钮就可以查看结果，如图 9.23 所示。

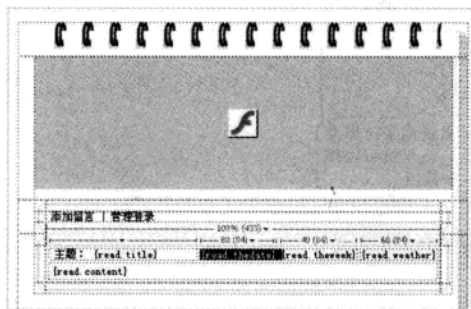


图 9.22 绑定其他字段



图 9.23 动态数据视图

在 edit_diary.asp 网页中插入记录集到页面的方法与和 index.asp 网页基本相同，如图 9.24 所示。

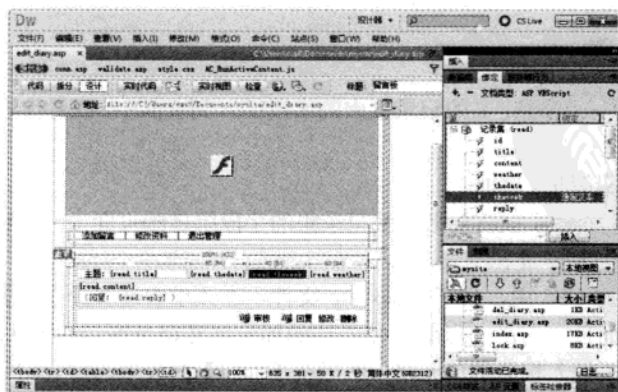


图 9.24 edit_diary.asp 网页记录集绑定

2. 把记录集绑定到文本框

rewrite_diary.asp 网页和 rewrite_admin.asp 网页的记录集不是直接插入到页面上，而是绑定到表单对象的属性上。具体操作步骤如下。

- 01 在 Dreamweaver CS5 中，打开 rewrite_diary.asp 文件。在菜单中选择【窗口】|【绑定】命令，打开【绑定】面板。单击 Recordset1 记录集左边的加号展开记录集。
- 02 在页面中选中标题文本框，然后在【绑定】面板底部【绑定到】下拉列表中选择输入值属性 input value，如图 9.25 所示。

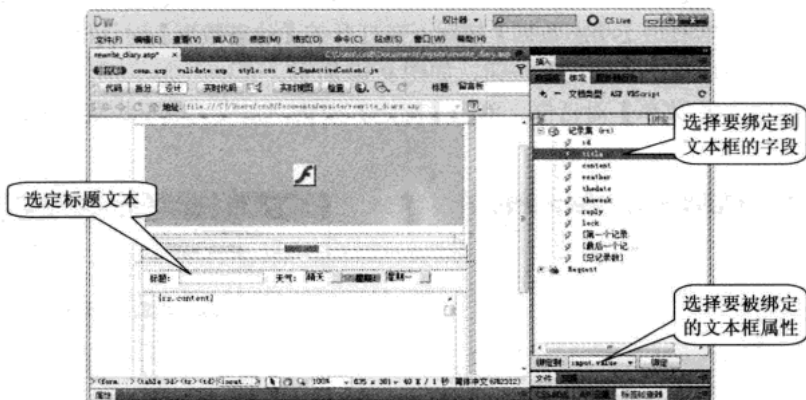


图 9.25 绑定动态数据

- 03 在 rs 记录集中选择 title 字段，单击【绑定】按钮，绑定后如图 9.26 所示。
- 04 以同样的方式把 rs 记录集中的 content 字段绑定到 textarea 文本域，如图 9.27 所示。



图 9.26 绑定标题文本框

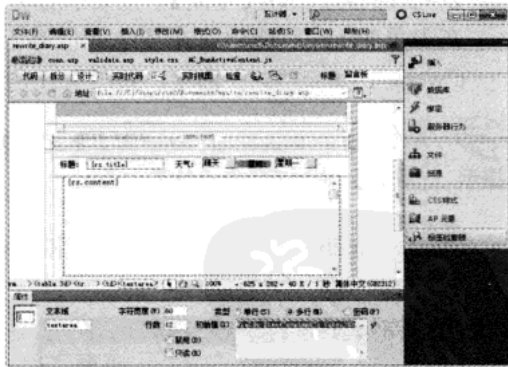


图 9.27 绑定内容文本框

3. 把记录集绑定到列表/菜单

把记录集绑定到列表/菜单中的操作步骤如下。

- 01 在 Dreamweaver CS5 中，打开 rewrite_diary.asp 文件。在菜单中选择【窗口】|【绑定】命令，打开【绑定】面板。单击 rs 记录集左边的加号展开记录集。
- 02 在页面中选中天气下拉菜单，如图 9.28 所示，然后在【属性】面板中单击【动态】按钮，打开【动态列表/菜单】对话框。

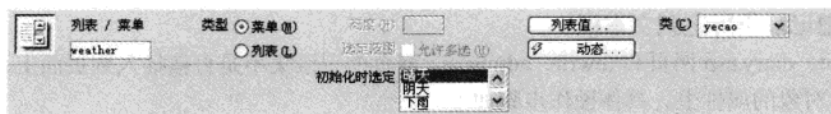


图 9.28 列表/菜单【属性】面板

- 03 在【动态列表/菜单】对话框的【菜单】下拉列表中选择“weather”在表单“form”选项，设置要绑定动态数据的菜单，如图 9.29 所示。
- 04 在【选取值等于】文本框中输入“<%= (Recordset1.Fields.Item("weather").Value)%>”，也可以单击文本框右边【动态数据】按钮，在打开的【动态数据】对话框中选择【weather】字段选项，然后单击【确定】按钮，如图 9.30 所示。
- 05 回到【动态列表/菜单】对话框，单击【确定】按钮，完成天气下拉菜单的动态数据绑定工作。

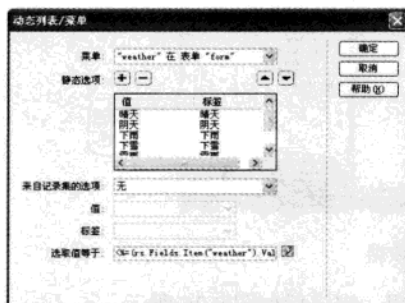


图 9.29 【动态列表/菜单】对话框



图 9.30 【动态数据】对话框

- 06 以同样的方式绑定【星期】下拉菜单的动态数据。要注意在【菜单】下拉列表中选择“week”在表单“form”选项，在【选取值等于】文本框中输入“<%= (Recordset1.Fields.Item("week").Value)%>”选项，如图 9.31 所示。



提示

由于 Recordset1 记录集需要进行筛选才能创建，因此此时单击标题栏中的【动态数据视图】按钮不能查看结果。

rewrite_admin.asp 网页的记录集绑定和 rewrite_diary.asp 文件操作基本相同，如图 9.32 所示。



图 9.31 【动态列表/菜单】对话框

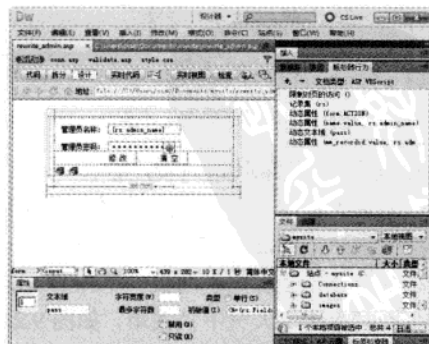



图 9.32 rewrite_admin.asp 网页的记录集绑定

9.4.3 重复显示记录

把记录集绑定到页面中，它只能显示一条当前指针所在位置的记录。这对于要显示多条记录的页面来说，显然是不合适的。因此，读者需要利用“重复区域”服务器行为来实现一页显示多条记录的功能，具体操作步骤如下。

- 01 在 Dreamweaver CS5 中打开 `edit_diary.asp` 文件。在页面中选中动态记录所处的独立表格，如图 9.33 所示。在选择表格时要注意选择完整，不能只选中表格的几行或跨表格选取，这样在显示时就会造成页面的错乱。
- 02 选择【窗口】|【服务器行为】菜单命令，打开【服务器行为】面板。单击【服务器行为】面板左上角的  按钮，在弹出的下拉菜单中选择【重复区域】菜单命令。
- 03 打开【重复区域】对话框，在【记录集】下拉列表中选择所要显示的记录集，如 `read`；在【显示】区域设置每页只显示 5 条记录，如图 9.34 所示。

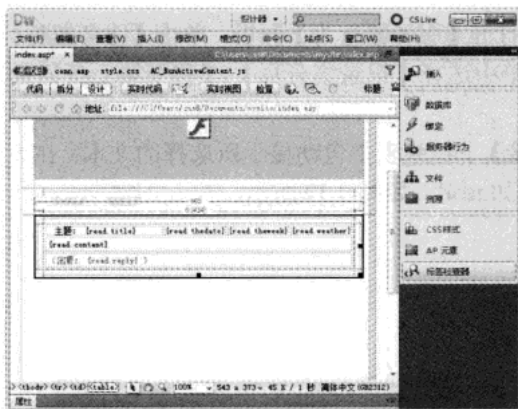


图 9.33 选中整个表格

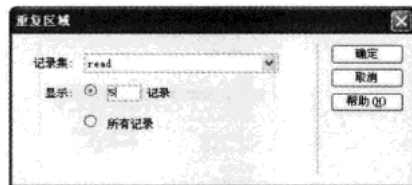


图 9.34 【重复区域】对话框

- 04 单击【确定】按钮，即可在页面中插入重复区域服务器行为，如图 9.35 所示。此时单击标题栏中的【动态数据视图】按钮可以看到图 9.36 所示的结果。

在 `index.asp` 网页中也需要每页显示多条记录，方法与上面操作步骤相同，读者可以自己动手试一试，这里就不再重复。



图 9.35 插入重复区域服务器行为



图 9.36 重复区域显示效果

9.4.4 记录集分页

如果按每页 5 条的记录显示记录集中的数据，还有很多条记录无法显示，这时可以通过插入“记录集分页”服务器行为来实现记录集多页显示，具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中，打开 edit_diary.asp 文件，把光标置于要显示导航条的区域。在本例中，选择第 1 个导航文本“首页”，如图 9.37 所示。

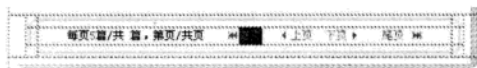


图 9.37 选中导航文本



提示

其中导航文本【首页】前边的 9 是 webdings 字体符号，源代码是9，显示效果是 9。

- 02** 选择【窗口】|【服务器行为】菜单命令，打开【服务器行为】面板。单击【服务器行为】面板左上角的【+】按钮，在弹出的下拉菜单中选择【记录集分页】|【移至第一条记录】菜单命令，如图 9.38 所示。

- 03** 打开【移至第一条记录】对话框，在【链接】下拉列表中自动显示所选择的文本；在【记录集】下拉列表中选择要读取的记录集，如 read，如图 9.39 所示。



图 9.38 选择“记录集分页”服务器行为

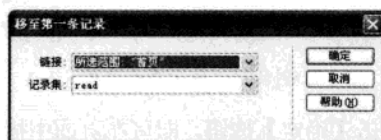


图 9.39 【移至第一条记录】对话框

- 04** 以同样的方式把【记录集分页】子菜单中的其他 3 个导航命令增加到相应的导航文本上，效果如图 9.40 所示。

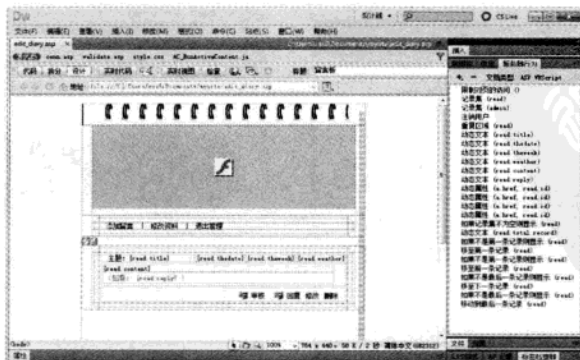


图 9.40 插入“记录集分页”服务器行为效果

在 index.asp 网页也需要插入记录集分页服务器行为，读者可以自己模仿练习。

9.4.5 控制导航条

完成导航条的制作后，在浏览器中浏览时会发现这样一个问题：当记录显示到第 1 条记录时，“首页”超链接文本依然可以点击执行，这会给浏览者造成一种错觉，即前边好像还有记录，其实已经没有记录了。

下面利用“显示区域”服务器行为来控制导航条的显示，实现当记录集显示到第 1 页时，使“首页”和“上页”超链接文本及相关提示符号不显示；而当记录集显示到最后 1 页时，使“下页”和“尾页”超链接文本及相关提示符号不显示，具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中，打开 edit_diary.asp 文件。选择第 1 个导航文本“首页”，同时要把前边的提示符号选中，要注意选取完整。



建议在【代码】视图下选取会比较保险，被选择的代码如下：`9<A HREF=""<%=MM_moveFirst%>">首页`。如果在【设计】视图下用光标选取，有时会在很大的误差。

- 02** 选择【窗口】|【服务器行为】菜单命令，打开【服务器行为】面板。单击【服务器行为】面板左上角的 按钮，在弹出的下拉菜单中选择【显示区域】|【如果不是第一条记录则显示区域】菜单命令，如图 9.41 所示。

- 03** 打开【如果不是第一条记录则显示区域】对话框，在【记录集】下拉列表选择一个选项，如 read，如图 9.42 所示。单击【确定】按钮，即可增加一条显示区域服务器行为。

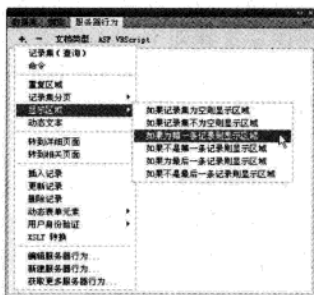


图 9.41 【显示区域】子菜单

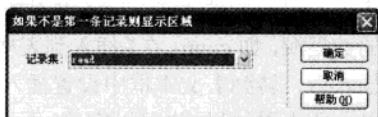


图 9.42 【如果不是第一条记录则显示区域】对话框

- 04** 以同样的方式为“上页”增加“如果不是第一条记录则显示区域”服务器行为，为“下页”和“尾页”增加“如果不是最后一条记录则显示区域”服务器行为。增加后的效果如图 9.43 所示。

- 05** 选中整个导航条表格，如图 9.44 所示。单击【服务器行为】面板左上角的 按钮，在弹出的下拉菜单中选择【显示区域】|【如果记录集不为空则显示区域】菜单命令，设置当没有数据显示时就不显示导航条。



图 9.43 增加的显示区域服务器行为



图 9.44 选取整个导航条表格

- 06** 此时【服务器行为】面板中列出了全部增加的服务器行为，如图 9.45 所示。单击其中任意一条服务器行为可以重新打开相应的对话框，修改参数。如果按 F12 键，则在浏览器中预览效果，如图 9.46 所示。



图 9.45 增加行为后的【服务器行为】面板



图 9.46 记录集分页效果

在 index.asp 网页也需要对记录集分页导航条进行控制，读者可以自己模仿练习。

9.4.6 增加记录

在第 6.4.1 小节中曾经详细介绍了利用“插入记录”服务器行为向数据库中插入数据的方法。下面再简要介绍在本例中实现插入记录的功能。具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中，打开 add_diary.asp 页面，在菜单中选择【窗口】|【服务器行为】命令，打开【服务器行为】面板。单击面板左上角的 按钮，在弹出的下拉菜单中选择【插入记录】菜单命令。
- 02** 打开【插入记录】对话框，具体设置如图 9.47 所示。在【连接】下拉列表中选择前面定义好的数据源，如 conn；在【插入到表格】下拉列表中选择要插入的数据表，如 feedback；在【插入后，转到】文本框中设置插入记录成功后要跳转到哪里，即插入后在编辑页面中可以立即查看结果。也可以空着，不设置该项表示插入成功之后依然显示 add_diary.asp 页面。

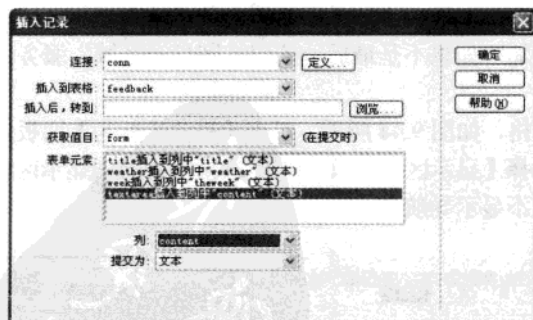


图 9.47 设置【插入记录】对话框

- 03 在【表单元素】列表框中分别设置每个表单对象中要插入数据库中 feedback 表的哪个字段。
 04 设置完毕, 单击【确定】按钮即可实现插入记录功能, 运行效果如图 9.48 所示。



图 9.48 增加留言效果

9.5 留言板后台管理功能实现

09

对于普通访问者来说, 用户只能在留言板中浏览和发表留言, 但不能修改、删除留言或者执行其他高级操作。只有网站管理人员登录后台之后才可以对所有留言进行修改、删除、审核或者回复等操作, 这些功能都属于网站后台管理功能。下面将重点介绍如何实现这些后台功能。

记录的删除和修改相对复杂, 由于涉及对数据库的写操作, 因此在学习的过程中可能会出现失败的现象, 存在的原因可能是数据库锁定、环境锁定, 或者操作失误, 所以初学者不要性急, 一方面有心理准备, 另一方面要积极探索问题所在, 提高自己的编程能力。

9.5.1 管理登录

在第7章中曾经详细介绍了用户登录的各种功能的实现方法。在本例中用户登录设计得比较简单, 仅需要添加登录成功与失败后跳转的页面, 打开 admin.asp 文件, 在页面中增加插入记录服务器行为。对话框设置如图 9.49 所示。

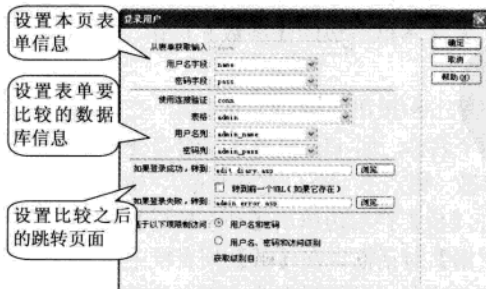



图 9.49 【登录用户】对话框

关于该对话框设置的详细说明可以参考第 7.4.2 节内容。

9.5.2 限制访问

留言板管理页面只允许后台登录成功的用户访问，为了禁止没有权限的用户访问后台页面，读者需要使用“限制对页的访问”服务器行为监视每位后台访问者。实现的方法如下：

01 新建一个 validate.asp 文件。打开该文件，切换到【代码】视图，清除该文档中的所有 HTML 代码。

02 单击【服务器行为】面板左上角的  按钮，在弹出的下拉菜单中选择【用户身份验证】|【限制对页的访问】菜单命令。为该页面增加“限制对页的访问”服务器行为。

03 切换到【代码】视图，删除第 1 行代码：

```
<%@LANGUAGE="VBSCRIPT"%>
```

04 使用下面一句代码把该文件包含到所有后台文件代码行的顶部，例如，edit_diary.asp、rewrite_admin.asp 和 rewrite_diary.asp。


```
<!--#include file="validate.asp" -->
```

9.5.3 修改留言

在动手实现修改留言页面的功能之前，不妨先理清该技术实现的基本思路。

- 当在编辑页面中单击某条留言中的“修改”超链接后，要准确获取这条留言的 ID 信息，并把该信息传递给修改页面。
- 在修改页面中定义一个记录集，该记录集根据传递过来的 ID 参数信息筛选出数据表中的记录（当然只有一条记录，该记录正是前面浏览要修改的记录）。
- 利用第 9.4.2 节介绍的方法把该记录集信息显示在修改页面中，让浏览者感觉是把修改的信息复制过来的一样。
- 这时用户可以在修改页面表单中修改该记录的信息，最后，利用“更新记录”服务器行为实现对后台数据库中数据的更新修改。

具体操作步骤如下。

01 在 Dreamweaver CS5 中，打开 edit_diary.asp 页面，选中“修改”超链接文本，在【属性】面板中单击【链接】文本框后边的【浏览文件】图标 ，打开【选择文件】对话框，如图 9.50 所示。

02 选择要跳转到的文件，这里选择 rewrite_diary.asp 文件，然后单击【URL】文本框右边的【参数】按钮，打开【参数】对话框。


03 在【参数】对话框中单击  按钮，增加一条参数，在【名称】栏中输入一个名称，如 id，该查询字符串变量用来存储和传递参数（相当于一个自定义变量），如图 9.51 所示。然后在【值】栏右边单击【动态数据】按钮，打开【动态数据】对话框，在【域】列表框中选择 id 字段（要注意该字段必须被设置为主键，这样才能保证该记录的唯一性），如图 9.52 所示。



图 9.50 【选择文件】对话框

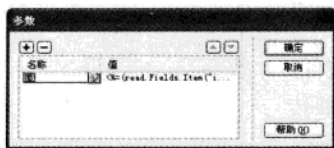


图 9.51 【参数】对话框

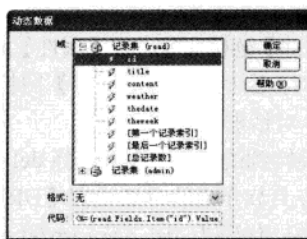


图 9.52 【动态数据】对话框


- 04 单击【确定】按钮返回，在【参数】对话框中单击【确定】按钮返回到【选择文件】对话框。此时会发现【URL】文本框中变成了 `rewrite_diary.asp?id= <%= (read.Fields.Item("id").Value)%>`，实现的功能就是把单击的记录 id 字段参数附加到 URL 后面，以便传递给 `rewrite_diary.asp` 文件。
- 05 打开 `rewrite_diary.asp` 文件，在新建的记录集中根据传递过来的参数变量 id 来筛选数据，如图 9.53 所示，详细介绍可参见第 9.4.1 小节。
- 06 把筛选出的记录集（只有一条记录）绑定到表单对象上，具体方法可参见第 9.4.2 小节。
- 07 打开【服务器行为】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【更新记录】菜单命令。
- 08 打开【更新记录】对话框，具体设置如图 9.54 所示。在【连接】下拉列表中选择前面定义好的数据源 `conn`；在【要更新的表格】下拉列表中选择要更新的数据表 `feedback`；在【选取记录自】下拉列表中选择定义的记录集 `Recordset1`；在【唯一键列】下拉列表中选择被定义为主键的字段 `id`；在【在更新后，转到】文本框中设置更新记录成功后要跳转到 `edit_diary.asp`，即更新后在编辑页面中可以立即查看结果；在【获取值自】下拉列表中选择表单 `form`；在【表单元素】列表框中设置每个表单对象所要对应的数据表中的字段。
- 09 单击【确定】按钮，即增加“更新记录”服务器行为到页面中，这样就可以在该页面中修改指定留言的内容，当单击“修改”按钮，提交表单后会把修改后的数据写入到数据库中。




图 9.53 利用传递参数筛选数据



图 9.54 设置【更新记录】对话框

9.5.4 删除留言

删除记录的实现方法和修改记录的思路相同，具体操作步骤如下。

01 在 Dreamweaver CS5 中, 打开 edit_diary.asp 页面, 选中【删除】链接文字, 在【属性】面板中单击【链接】文本框后边的【浏览文件】图标, 打开【选择文件】对话框。

02 选择要跳转到的文件, 这里选择 del_diary.asp 文件, 然后模仿上节介绍的方法设置【URL】文本框要传递的参数, 代码为 del_diary.asp?id=<%= (read.Fields.Item("id").Value)%>, 如图 9.55 所示。

03 在 del_diary.asp 文件中输入下面代码, 实现对指定记录执行删除操作, 删除之后跳转到 edit_diary.asp。

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<!--#include file="validate.asp" -->
<% '定义数据库连接实例
Dim conn
Set conn=Server.CreateObject("ADODB.Connection")
conn.Open MM_conn_STRING
'定义SQL字符串
dim theid,connString
theid =Cint(trim(Request.QueryString("id")))
connString = "delete from feedback where id = " & theid
'利用Connection对象执行删除操作
conn.Execute connString
'关闭数据库连接
conn.close
set conn=nothing
'跳转到edit_diary.asp页面
Response.Redirect("edit_diary.asp")
%>
```

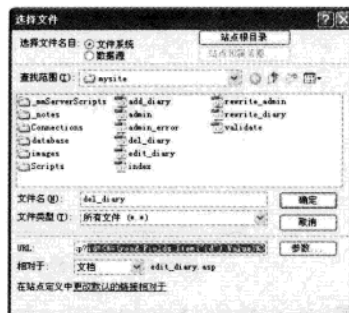


图 9.55 设置【选择文件】对话框



提示

读者也可以使用“删除记录”服务器行为来删除留言记录, 但是要使用该服务器行为, 必须要定义一个记录集, 同时要在页面中增加一个表单, 把记录集绑定到表单中。虽然不用手工编写代码, 但操作起来比较麻烦, 不如上面代码简洁高效。

9.5.5 增加记录集分页提示信息

当记录集中的记录非常多时, 用户在浏览留言时常会发现不知当前浏览到第几页, 总共多少留言, 分了多少页? 因此, 有必要在页面底部添加相关记录集提示信息, 方便用户浏览。解决这个问题的具体操作步骤如下。

01 打开 edit_diary.asp 文件, 在页面底部的导航条中设计一栏提示信息, 如图 9.56 所示。



图 9.56 设计动态提示牌版面

02 在“每页”文本后面输入 5, 表示每页要显示 5 篇留言 (根据重复区域服务器行为中设置数来设置), 并设为红色显示。在“共篇”两个文字之间输入代码: <%= (read_total)%>, 如图 9.57 所示, 其中 read_total 是一个系统变量, 用来获取记录集的记录总数。

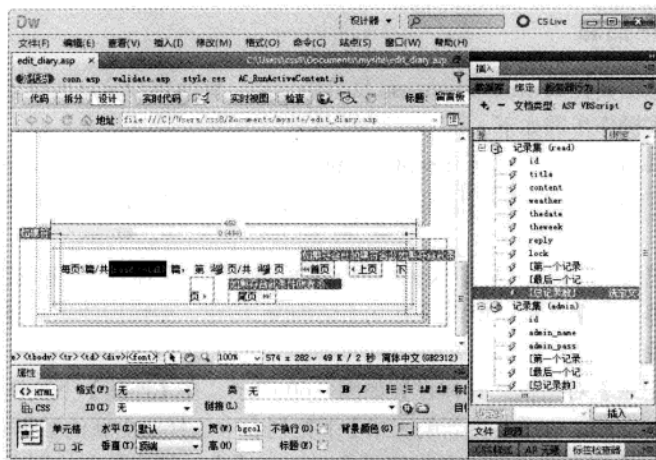


图 9.57 绑定记录集总数

03 在【设计】视图下，把光标置于“第页”两字之间，然后切换到【代码】视图，在光标处输入下面代码：

```
<% 利用每页最后一条记录数变量read_last和每页记录数，计算当前页数
if read_last > 0 then
    page = read_last / 5
    if page <> Int(page) then
        page = Int(page) + 1
    end if
    Response.Write(page)
Else
    Response.Write( "?")
end if
%>
```

在上面代码中，read_last 是系统变量，记录当前页最后一条记录在数据集中是第几条记录。然后根据每页记录数来计算当前页数。

04 在【设计】视图下，把光标置于“共页”两文字之间，然后切换到【代码】视图，在光标处输入下面代码：

```
<% 利用总记录数变量read_total和每页记录数，计算分页数
if read_total > 0 then
    page = read_total / 5
    if page <> Int(page) then
        page = Int(page) + 1
    end if
    Response.Write(page)
Else
    Response.Write( "?")
end if
%>
```

在上面代码中，read_total 是系统变量，记录当前记录集总记录数。然后根据每页记录数来计算记录集总页数。

05 按 F12 键，在浏览器中的浏览效果如图 9.58 所示。


在 index.asp 网页也需要插入记录集分页提示信息，读者可以自己模仿练习。



图 9.58 记录集分页信息效果

9.5.6 修改用户信息

修改用户信息主要在 `rewrite_admin.asp` 文件中实现, 修改用户信息的操作与修改留言的操作基本相同。基本操作步骤如下。

- 01 在 Dreamweaver CS5 中打开 `rewrite_admin.asp` 文件, 新建记录集, 并根据阶段变量 `Session("MM_Username")` 筛选用户记录, 记录集设置如图 9.59 所示, 详细介绍可参见第 9.4.1 小节。
- 02 把筛选出的记录集 (只有一条记录) 绑定到表单对象上, 具体方法可参见第 9.4.2 小节。
- 03 打开【服务器行为】面板。单击面板左上角的  按钮, 在弹出的下拉菜单中选择【更新记录】菜单命令。
- 04 打开【更新记录】对话框, 具体设置如图 9.60 所示。在【连接】下拉列表中选择前面定义好的数据源 `conn`; 在【要更新的表格】下拉列表中选择要更新的数据表 `admin`; 在【选取记录自】下拉列表中选择定义的记录集 `rs`; 在【唯一键列】下拉列表中选择被定义为主键的字段 `admin_name`; 在【在更新后, 转到】文本框中设置更新记录成功后要跳转到 `edit_diary.asp`; 在【获取值自】下拉列表中选择表单 `form`; 在【表单元素】列表框中设置每个表单对象所要对应的数据表中的字段。

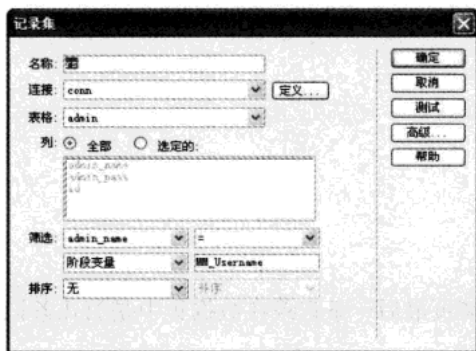


图 9.59 利用阶段变量筛选数据

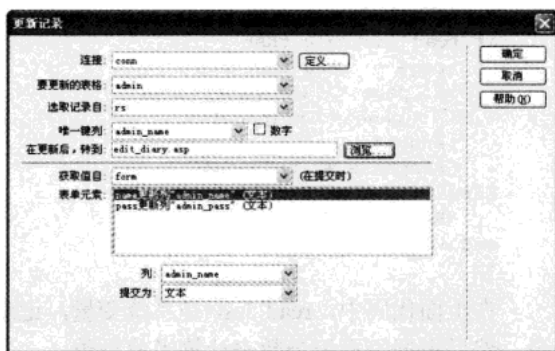


图 9.60 设置【更新记录】对话框

- 05 单击【确定】按钮, 即增加“更新记录”服务器行为到页面中, 这样就可以在该页面中修改用户信息了, 当单击“修改”按钮, 提交表单后会把修改后的数据写入到数据库中。
- 06 读者应该切换到【代码】视图, 找到“更新记录”服务器行为代码段, 在更新成功并准备跳转页面之前 (语句 `Response.Redirect(MM_editRedirectUrl)` 前面) 插入下面代码, 用来同时更新阶段变量 `MM_Username` 的值。

```
'修改用户名之后, 同时更新阶段变量MM_Username的值
'-----
Session("MM_Username") = trim(Request.Form("name"))
'-----
Response.Redirect(MM_editRedirectUrl)
End If
End If
%>
```

9.5.7 注销用户登录

当用户登录之后, 可能会中途退出管理, 这时就应该为用户提供一个退出管理的出口, 以

便把该用户信息注销掉，防止被其他用户利用。在第 7.4.4 节中曾经详细介绍过注销用户登录的方法。下面结合本实例简单介绍一下基本方法。具体操作步骤如下。

- 01 打开 edit_diary.asp 文件，选中导航菜单中的“退出管理”文本。在【服务器行为】面板中选择【用户身份验证】|【注销用户】命令。
- 02 在打开的【注销用户】对话框中进行简单设置，设置如图 9.61 所示。最后单击【确定】按钮即可。

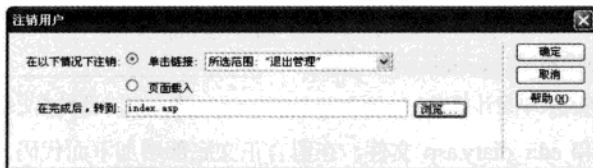


图 9.61 设置【注销用户】对话框

9.6 扩展留言板功能

留言板虽然作用单一，但用户可以通过扩展功能来增强留言板的作用。实际上 BBS、论坛、Blog 等 Web 应用都是从最初的留言板发展、演化而来。下面将重点介绍如何实现留言板的审核和回复功能。

9.6.1 留言回复

留言回复是管理员（登录用户）的权限，因此读者应该把该功能设计在后台页面中（edit_diary.asp 文件中）。由于留言回复涉及数据表结构的改动，所以实现时稍微麻烦点。

留言回复的设计思路是：在 feedback 数据表中增加 reply 字段，用来保存留言信息，然后新建 reply 文件，用来写回复或者更新回复，然后把留言回复显示在页面中即可。具体操作步骤如下。

- 01 在数据库 feedback.mdb 的 feedback 数据表中增加一个字段，字段名为 reply，数据类型为“文本”，字段大小可以设置“255”（根据实际情况而定，一般 255 个字符基本上满足回复需要）。
- 02 在 edit_diary.asp 文件中“修改”文本前面添加一个“回复”超链接，定义超链接 URL 为 reply.asp 文件，传递参数为 id（与“修改”超链接的设置完全相同，可以参阅第 9.5.3 节介绍），具体代码如下：


```
<a href="reply.asp?id=<%= (read.Fields.Item("id").Value) %>">回复</a>
```
- 03 新建 reply.asp 文件（读者可以复制 rewrite_diary.asp 文件，然后改名为 reply.asp，并简单修改表单页面和服务器行为设置）。
- 04 在该文件中定义一个记录集，查询符合参数变量 id 值的记录，然后把该记录集字段绑定到表单中，用来显示数据库中已经存在的留言回复信息。
- 05 增加“更新记录”服务器行为（操作与第 9.5.3 节介绍的修改留言操作基本相同），实现把新写或者修改的回复信息保存到数据库中。页面中表单设计效果如图 9.62 所示。定义的“更新记录”服务器行为如图 9.63 所示。



图 9.62 留言表单设计效果



图 9.63 设置【更新记录】对话框

- 06** 打开 index.asp 和 edit_diary.asp 文件，在留言正文底部增加下面代码。注意，在增加代码时一定要准确确定代码插入点，即应该在留言正文所在表格行

```
<%= (read.Fields.Item("content").Value) %>
<% If read.Fields.Item("reply").Value <> "" then %>
<TR><TD style="padding: 4px; margin-top: 6px; line-height: 120%; color: red; border-top: solid
#ddd 1px;" colSpan=4>
(回复: <%= (read.Fields.Item("reply").Value) %> )
</TD></TR>
<% end if %>
```

上面代码首先判断记录中是否存在留言回复，如果有则显示回复内容。

- 07** 设计完毕，如果用户登录后后台，单击“回复”超链接文本，将打开 reply.asp 文件，如图 9.64 所示。在其中写入新留言回复或者修改已经存在的留言回复，然后单击【回复】按钮，即可在 index.asp 和 edit_diary.asp 文件中看到新留言回复信息，如图 9.65 所示。



图 9.64 回复留言



图 9.65 显示留言回复

9.6.2 留言审核

留言审核可以说是留言板的防火墙，它能够预防有害或者不良信息的发布。当用户在网站内留言时，只有经过网站管理人员审核后才可以显示出来，这样就可以避免有些用户发表与主题不一致的内容，或者发布一些虚假、黄赌毒类的信息。虽然网站管理员可以通过删除功能删除这些留言信息，但有时当管理员发现时，这些不良信息已经在留言板中显示了很长

时间。审核具有预防效果，而删除只能是亡羊补牢，效果当然就不同了。

留言审核的设计思路是：在 feedback 数据表中增加 lock 字段，用来保存审核信息，然后新建 lock 文件，用来对留言信息进行审核操作，审核之后把审核结果写入数据库。在 index.asp 文件中根据 lock 字段的信息决定是否显示每条留言信息。具体操作步骤如下。

01 在数据库 feedback.mdb 的 feedback 数据表中增加一个字段，字段名为 lock，数据类型为“是/否”，该字段默认值为 0，表示“否”（即 false），如果取值为非 0（默认为 1），则表示“是”（即 true）。

02 在 edit_diary.asp 文件中“回复”文本前面添加一个“审核”超链接，定义超链接 URL 为 lock.asp 文件，传递参数为 id（与“修改”超链接的设置完全相同，可以参阅第 9.5.3 节介绍），具体代码如下：

```
<%if read.Fields.Item("lock").Value = 0 then %>
<a href="lock.asp?id=<%=read.Fields.Item("id").Value%>">审核</a>&#8195;
<% end if %>
```

在上面代码中，通过一个条件语句判断当前记录是否通过审核，如果没有通过审核，则显示“审核”超链接，单击该超链接将跳转到 lock.asp 文件，并传递 id 参数值。

03 新建 lock.asp 文件（读者可以复制 reply.asp 文件，然后改名为 lock.asp，并简单修改表单页面和服务行为设置）。

04 在该文件中定义一个记录集，查询符合参数变量 id 值的记录，然后把该记录集字段绑定到表单中，用来显示数据库中该记录的审核信息。代码如下：

```
<h3 align="center">审 核</h3>
<% if rs.Fields.Item("lock").Value <> 0 then %>
    <input name="ok" type="radio" value="true" checked>通 过<br>
    <input name="ok" type="radio" value="false">没通过
<% else %>
    <input name="ok" type="radio" value="true" >通 过<br>
    <input name="ok" type="radio" value="false" checked>没通过
<% end if %>
```

在上面代码中，利用一个条件语句判断该记录是否通过审核，如果通过审核，则显示“通过”单选按钮为选中状态，否则显示“没通过”单选按钮为选中状态。页面中表单设计效果如图 9.66 所示。

05 增加“更新记录”服务器行为，定义的“更新记录”服务器行为如图 9.67 所示。在【更新记录】对话框中，要选择【提交为】下拉菜单为“复选框 1、0”选项。

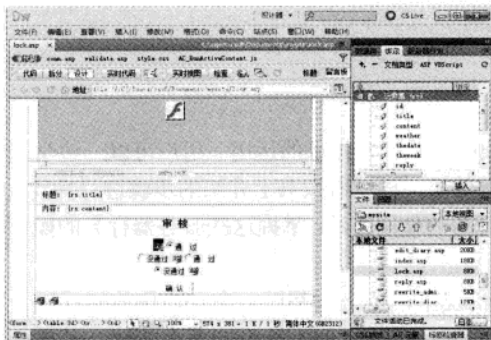


图 9.66 审核表单设计效果



图 9.67 设置【更新记录】对话框

- 06** 打开 index.asp 文件，修改记录集 read 的筛选条件为字段 lock 不等于 0。方法是在【绑定】面板中双击“read”记录集，打开【记录集】对话框，修改其中的【筛选】选项，如图 9.68 所示。

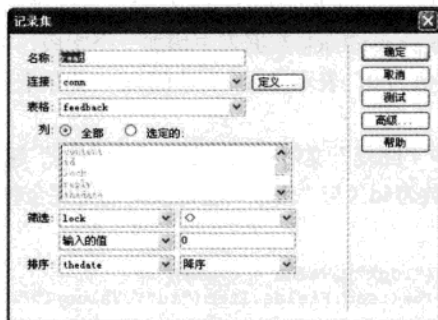


图 9.68 修改【记录集】对话框

- 07** 设计完毕，如果用户登录后后台，单击“审核”超链接文本，将打开 lock.asp 文件，如图 9.69 所示。在其中选择是否通过审核，然后单击【确认】按钮，即可在 index.asp 文件中看到该留言信息，如图 9.70 所示。

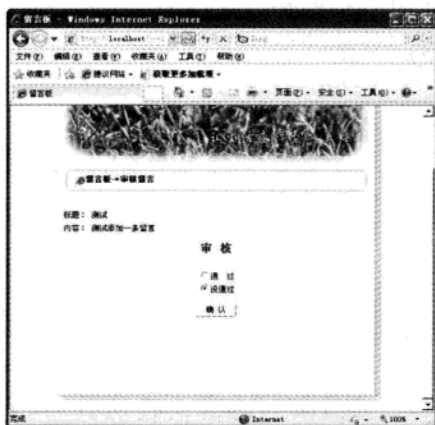


图 9.69 留言审核



图 9.70 显示审核通过的留言

9.7 实践总结

本章重点介绍了留言板的设计方法，涉及多项服务器行为技术的运用。用户可以一边学习一边操作，先把本书光盘实例复制到本地站点看看效果，再试着按书上介绍的步骤进行操作。同时本章介绍了 Dreamweaver CS5 大部分服务器行为的使用方法，掌握这些服务器行为的操作方法，能够提高用户的动态网站开发速度。

目前，留言板的显示形式多种多样。在很多情况下，留言板都以一个小模块的样式融合到其他页面中，例如，新闻页面的评论、调查页面的反馈等。但不管形式怎么变化，其核心技术是一样的，用户不妨从最基本的技术入手，然后逐步深入。

网站流量统计与分析

把统计学应用于网站功能已日渐成为网站开发的重要扩展领域,在实际应用中经常会发现很多统计应用的新功能,例如,圈子、同城、小纸条、搜索中各种关联信息提示、RSS 聚合信息分析、网站 PV 值等。流量统计和分析是网站中一个非常实用的功能模块,利用该模块可以了解网站被访问的详细信息。一个好的网站统计分析模块要比简单的网站访问计数器要复杂得多。由于计算机高效的数据处理能力,统计系统不仅应当收集尽可能多的客户端信息,而且更承担着对所收集数据进行一定后续分析的功能。对于大型商业站点来说,统计信息对于网站开发和更新具有重要的指导意义,因此对于统计系统的高效性和安全性的要求也是非常高的。

技术要点

- Application 对象技术
- 用代码编写数据库连接
- 用代码定义和绑定记录集
- 自定义函数
- 图形化流量显示

10.1 程序设计分析

流量统计和分析模块可以分为两个部分:信息收集、统计和信息分析、汇总。当客户端用户通过浏览器向远程服务器发送请求指令时,例如,输入一个网址、单击一个超级链接,或者是在网页代码中包含请求远程文件的代码,浏览器会向远程服务器发送一个数据包,这个数据包除了包含所请求的文件位置信息外,还包括许多其他信息,例如,在 HTTP 头信息中可以看到用户的信息(如浏览器类型、用户的操作系统、用户安装的组件等),这时如果在服务器端使用 ASP 的 Request 对象,可以访问 ServerVariables 数据集合来获取这些信息。

ASP 获取这些信息之后,可以将其分类保存到数据库中,然后服务器通过统计和分析当前网站的所有用户信息,就可以得出当前网站的流量信息,如用户的来源、用户访问时间段分布、用户访问密度、网站历史流量曲线等。

要实现网站流量统计和分析功能,读者需要完成 3 步工作:第一,在被统计的网站首页内嵌入一行 JavaScript 代码,由该脚本调用后台 ASP 信息统计文件;第二,在 ASP 信息统计文件中获取客户端用户的信息,然后把客户端原始信息经过简单的加工存储到服务器端数据

库中；第三，设计后台管理页面，用来显示网站流量的分析数据，方便网站管理者参考。

本实例页面设计比较简单，但要实现流量统计和分析功能，需要解决下面技术问题。

- 利用 Request.ServerVariables() 语句获取客户端信息。
- 利用 Application 技术实现对数据库的更新，避免多位浏览者同时读写数据库时，发生数据错误。
- 利用 Execute 命令实现对数据库的快速操作。
- 用代码编写数据库连接，并定义绑定记录集。
- 自编函数实现对流量信息的图形化显示。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，如 mysite（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”（详细设置说明可以参阅第 2.5.2 节介绍）。

把本书光盘中的“10”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

10.2 数据库设计

本例数据库 stat.mdb 共包括 3 个表，stat、userandip 和 history。要设计功能强大的网站流量统计和分析模块，数据库的结构和字段设计很重要，不合理的数据结构会为数据统计和分析带来麻烦，甚至有些功能无法实现。

10.2.1 stat 表

stat 表用来记录网站流量明细信息，结构见表 10.1。

表 10.1 stat 表字段列表

字段	类型	字段大小	必填字段	说明
v_sum	数字	长整型	否	总访问人数
y_ips	数字	长整型	否	昨日 IP 访问量
t_ips	数字	长整型	否	今日 IP 访问量
t_date	日期/时间	-	否	今天日期
top_ips	数字	长整型	否	最高 IP 访问量
v_days	数字	长整型	否	统计天数
t_clicks	数字	长整型	否	今日浏览量

stat 表主要用来对流量信息的汇总，不用做记录的保存，因此就不需要设置主键字段和必填字段。

10.2.2 userandip 表

userandip 表主要用来存储网名和 IP 地址，该表数据结构设计见表 10.2。

表 10.2 userandip 表字段列表

字段	类型	字段大小	必填字段	允许空字符串	说明
id	自动编号	-	-	-	自动编号
User	文本	100	否	否	访问地址
Ip	文本	50	否	否	访问 IP

10.2.3 history 表

history 表主要用来存储每日的流量信息，该表数据结构设计见表 10.3。

表 10.3 history 表字段列表

字段	类型	字段大小	必填字段	默认值	说明
id	自动编号	-	-	-	自动编号
h_date	日期/时间	-	否	-	访问日期
h_ips	数字	长整型	否	0	每天的 IP 数
h_clicks	数字	长整型	否	0	每天的访问量

10.3 页面设计

本实例共有 index.asp 和 counter.asp 这 2 个文件。

- Index.asp 文件主要用来分析和显示流量信息，设计如图 10.1 所示。

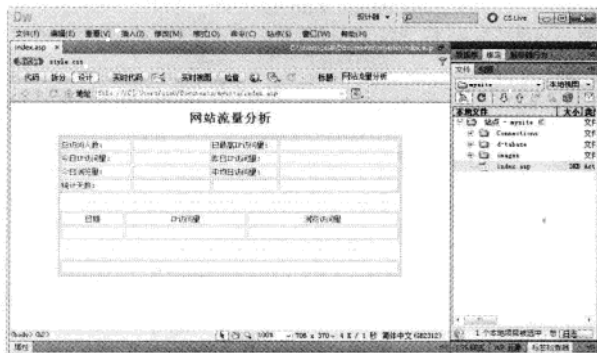


图 10.1 流量显示设计页面效果

- counter.asp 文件主要用来获取和统计用户信息，并存储到数据库中。

10.4 网站流量统计功能实现

为了对用户的访问进行跟踪和分析，读者应该先抓取每位用户的访问信息，这个信息可以包括客户端浏览器信息、系统信息、地区来源信息、IP 信息等，同时为了方便信息汇总和分析，读者还需要把抓取的信息存入数据库，以方便日后浏览和比较分析。下面就用代码的方式来实现流量的统计功能，具体介绍如下。

10.4.1 流量统计

新建 connection.asp 文件，该文件专门用来定义与数据库的连接，代码如下：

```

<!-- #include file= "Connections/conn.asp" -->
<% '打开数据库连接
Dim conn
Set conn = Server.CreateObject("ADODB.Connection")
on error resume next '定义错误处理方式,如果发现错误则跳转到下一行代码继续执行
conn.open MM_conn_STRING
%>

```

在上面代码中,第1行代码包含了 Connections/conn.asp 文件,该文件是使用【数据库】面板建立与数据库的连接时定义的数据库连接字符串(有关具体介绍可以参考第 5.4.3 节内容)。

打开 counter.asp 文件,编写下面代码用来获取用户名和用户 IP 信息,然后简单分析并存储到数据库中。详细代码如下:

```

<!-- #include file="connection.asp" -->
<%
'定义自变量
dim rs,Userip,User,t_date,history,stat,v_sums,t_ips,v_days,y_ips,Top,Stats,t_clicks
'获取用户名和IP地址
User=Request.QueryString("User")
Userip = Request.ServerVariables("HTTP_X_FORWARDED_FOR")
If Userip = "" Then Userip = Request.ServerVariables("REMOTE_ADDR")
'定义一个记录集
Set rs=Server.CreateObject("adodb.RecordSet")
Sql="Select * from stat"
rs.open Sql,conn,1,3
'获取数据库stat表中的t_date字段值
t_date=rs("t_date")
If t_date<>date() then '判断是否是当天日期
    application.lock '独占应用程序操作
    conn.Execute "Update stat Set t_ips=0,t_clicks=0,t_date=date(),v_days=v_days+1,
y_ips=" & rs("t_ips") & "" '更新数据库表stat记录
    conn.Execute "Insert into history(h_date,h_ips,h_clicks) values ('" & t_date & "','" & rs
("t_ips") & "','" & rs("t_clicks") & "')" '在history表相应字段插入新记录
    application.unlock '解除应用程序独占
    '定义记录集删除UserandIP表中记录
    Set rs=Server.CreateObject("adodb.recordSet")
    Sql="delete from UserandIP"
    rs.open Sql,conn,1,3
    rs.close
    Set rs=nothing
Else '如果是当天则增加stat表中流量字段t_clicks值
    application.lock
    conn.Execute "Update stat Set t_clicks=t_clicks+1"
    '定义新记录集,向UserandIP表中增加信息
    Set rs=Server.CreateObject("adodb.recordSet")
    Sql="Select * from UserandIP where IP='" & Userip & "' order by Id desc"
    rs.open Sql,conn,1,3
    If rs.bof and rs.eof then
        rs.addnew
        rs("IP")=Userip
        rs("User")=User
        rs.update
        conn.Execute "Update stat Set v_sum=v_sum+1,t_ips=t_ips+1"
    End If
    rs.close
    Set rs=nothing
    application.unlock
End If
'统计最高IP访问量
Set rs=Server.CreateObject("adodb.RecordSet")
Sql="Select * from stat"

```

```

rs.open Sql,conn,1,3
If rs("top_ips")<rs("t_ips") then
    conn.Execute "Update stat Set top_ips=" &rs("t_ips") &"
End If
'关闭RecordSet和Connection对象,释放资源
rs.close
Set rs=nothing
conn.close
Set conn=nothing
%>

```

上面代码的设计思路是这样的: 首先用 Request 对象获取客户端用户信息和 IP 地址, 再查询数据库统计表 stat, 获取该表中的字段 t_date, 该字段说明当前统计的信息的更新时间。用这个时间与现在时间比较, 如果相同, 说明表 stat 中统计数据今天的, 就不需要删除, 只需要累计更新部分数据即可; 如果不同, 说明表 stat 中统计数据不是今天的, 就需要全部删除记录, 重新进行统计, 同时把被删除记录中的重要数据转移保存到表 UserandIP 和表 history 中, 实现流量信息的不断存储, 以方便日后进行汇总、分析。

10.4.2 流量显示

index.asp 是一个后台文件, 为使网站管理者方便地浏览网站流量而设计。在该页面中可以显示当前网站不同的流量信息, 例如, 网站流量中主要指标值的明细表 (总访问人数、今日 IP 数、浏览量、最高一天的 IP 访问量、统计天数、平均 IP 访问量和平均浏览量)、每日浏览访问量, 以及今日最新的 IP 列表。实现流量信息显示的具体操作步骤如下。

01 在【文件】面板中, 打开 index.asp 文件。切换到【代码】视图, 在代码顶端输入下面代码, 包含 connection.asp 文件, 实现与后台数据库建立连接并打开这个连接, 然后定义一个记录集, 查询 stat 数据表中的数据。

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!-- #include file="connection.asp" -->
<%
'定义一个记录集, 查询stat表中所有字段记录
Set rs=Server.CreateObject("adodb.recordSet")
Sql="Select * from stat"
rs.open Sql,conn,1,3
%>

```

02 在流量明细表中绑定记录集中的字段, 如图 10.2 所示。代码如下:

```

<table border="0" cellpadding="0" style="border-collapse: collapse" width="100%">
    <tr onMouseOver="javascript:this.bgColor='#F4F4F4';" onMouseOut="javascript:
this.bgColor='#F9F9F9';">
        <td width="112">总访问人数: </td>
        <td><%=rs("v_sum") %></td>
        <td width="106">日最高IP访问量: </td>
        <td width="191"><%=rs("top_ips") %></td>
    </tr>
    <tr onMouseOver="javascript:this.bgColor='#F4F4F4';" onMouseOut="javascript:
this.bgColor='#F9F9F9';">
        <td>今日IP访问量: </td>
        <td><%=rs("t_ips") %></td>
        <td>昨日IP访问量: </td>
        <td><%=rs("y_ips") %></td>
    </tr>
    <tr onMouseOver="javascript:this.bgColor='#F4F4F4';" onMouseOut="javascript:
this.bgColor='#F9F9F9';">
        <td>今日浏览量: </td>

```

```
 <%=rs("t_clicks") %></td>  平均日访问量: </td>  <%=int(rs("v_sum")/rs("v_days")) %></td> </tr> |  |  |  | | --- | --- | --- | |统计天数: </td>  <%=rs("v_days") %></td>  &nbsp;&nbsp;&nbsp;</td>  &nbsp;&nbsp;&nbsp;</td> </tr> </table> | | | | | |
```

访问人数:		最高访问量:	
今日访问量:		昨日访问量:	
今日浏览量:		平均访问量:	
统计天数:			

图 10.2 绑定记录集字段

03 以同样的方法在上面表格前再定义一个记录集，在定义记录集之前，不要忘记关闭前面定义的记录集，释放资源。代码如下：

```

<%
rs.close '关闭记录集
dim h_date, h_ips, h_clicks
Sql="Select top 7 * from history order by h_date desc"
rs.open Sql,conn,1,3 '用新的SQL查询字符串打开记录集
%>

```

04 下面用一个条件语句判断该记录集是否为空，如果为非空，则利用循环语句循环显示 history 数据表中的所有记录。代码如下：

```

<%
If rs.eof and rs.bof then '判断记录集是否为空
    Response.Write "<p align=center>暂时没有往日统计记录</p>"
else
    rs.movefirst '把记录集指针移到第1条记录
    while not rs.eof '用循环语句把记录集循环显示出来
        h_date=rs("h_date")
        h_ips=rs("h_ips")
        h_clicks=rs("h_clicks")
    %>

<table border="0" cellpadding="0" style="border-collapse: collapse" width="100%">
<tr onMouseOver="javascript:this.bgColor='#F4F4F4';" onMouseOut="javascript:
this.bgColor='#F9F9F9';" align=center>
<td>日期</td>
<td>IP访问量</td>
<td>浏览访问量</td>
</tr>
|
|  |

```

把字段绑定到页面的效果如图 10.3 所示。

日期	IP访问量	浏览访问量
9/13	0	0
9/12	0	0
9/11	9	0
9/10	0	0
9/10	1	80

图 10.3 绑定 rs 记录集字段到页面

- 05 定义一个记录集，用来查询数据表 userandip 中的记录。代码如下：

```
<%
rs.close           '关闭前面定义的记录集
sql = "Select * from Userandip order by id desc"
rs.open sql,conn,1,1 '用上面定义的查询字符串打开记录集
If Not(rs.eof and rs.bof) then '判断如果记录集为非空，则显示记录
%>
```

显示记录的代码如下：

```
<%
For I = 1 to 10
    Response.Write "<tr>"
    For J=1 to 3
        Ip=rs("Ip")
        Response.Write "<td><font color='#000000'>"&ip&"</font></td>"
        rs.movenext
        If rs.eof then exit for
    Next
    Response.Write "</tr>"
    If rs.eof then exit for
Next
%>
```

- 06 关闭记录集，并销毁记录集对象实例。

```
<%
    End If
End If
rs.close
set rs = nothing
%>
```

- 07 输入代码完毕，按 F12 键，在浏览器中的浏览效果如图 10.4 所示。



图 10.4 流量信息显示效果

10.4.3 指定要统计的站点

前面主要介绍了网站流量和分析的核心代码设计，但是它还不能够工作，这正如 ASP 自定义对象一样，读者只有实例化后才可以在代码中使用。具体实现方法如下。

新建一个被统计的页面 `state.asp`，然后把下面脚本代码插入到被统计的页面任何位置即可：

```
<SCRIPT LANGUAGE=JavaScript src="counter.asp"></SCRIPT>
```

上面代码利用 JavaScript 脚本调入 `counter.asp` 文件，以实现流量统计功能的实现。这样每当一位用户打开该页面，都会自动调用 `counter.asp` 文件并执行其中的代码，实现网站流量的原始数据的抓取和保存。

如果管理者想具体了解网站流量，只需要在后台登录 `index.asp` 页面查看统计结果。

10.5 数字图形化显示技术

在互联网上，很多页面都以图形形式显示流量信息，视觉效果非常好，如图 10.5 所示。图形流量信息实际上就是用事先制作好的数字图片拼接成一串数字，具体操作步骤如下。

01 准备 10 张数字图片，最好大小一致，样式一致，如图 10.6 所示。

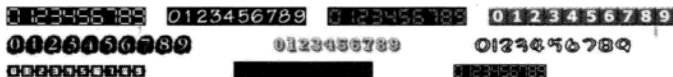


图 10.5 图形计数器效果

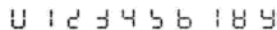


图 10.6 设计好的数字图片

02 打开 `index.asp` 文件，另存为 `picnum.asp`，然后在【代码】视图下代码底部定义一个自定义函数。

```
<%
'把数字替换为图形
Function picnum(counter)
    Dim S, i, G
    S = CStr(counter)
    For i = 1 to Len(S)
        If Mid(S, i, 1) = "-" Then '如果是字符 "-", 则替换为字符 "-", 并定义蓝色显示
            G = G & "<font color='blue'>-</font>"
        ElseIf Mid(S, i, 1) = "." Then '如果是字符 ".", 则替换为字符 ".", 并定义蓝色显示
            G = G & "<font color='blue'>.</font>"
        Else
            '如果是数字, 则用图片进行替换, 请注意图片路径设置要准确
            G = G & "<IMG SRC=images/" & Mid(S, i, 1) & ".gif Align=middle>"
        End If
    Next
    Response.write G
End Function
%>
```



提示

要注意引用的图片文件路径是否正确，请查看上面代码中的方框内语句。

03 用 `picnum()` 函数套用各个绑定的字段，例如：`rs("top_ips"))` 替换为 `<% =picnum(rs("top_ips")) %>`。其他绑定字段也是如此。

04 输入完毕，按 F12 键，在浏览器中的浏览效果如图 10.7 所示。



图 10.7 图形流量信息显示效果

10.6 实践总结

本章详细介绍了网站流量的设计方法，同时介绍了把记数数字转换成图形的方法，这样就使计数器的设计显示出了很强的灵活性。

流量统计的设计还有一种更为简捷的办法，即使用免费的计数器。其方法是将所申请的计数器代码加入到所制作的页面源代码中。

第 11 章

Dreamweaver

友情链接管理模块

友情链接 (Link exchange), 也称为网站交换链接, 是具有一定资源互补优势的网站之间的简单合作形式, 即分别在自己的网站上放置对方网站的 Logo 或网站名称, 并设置对方网站的超链接, 使得用户可以从合作网站中发现自己的网站, 达到互相推广的目的。因此常作为一种网站推广手段, 是互联网上宣传自己主页时常使用的一种方法。通过与其他站点交换链接, 可以吸引更多的用户点击访问。同时搜索引擎会根据交换链接的数量, 以及交换链接网站质量等对一个网站作出综合评价, 这也将影响网站在搜索引擎的排名。因此, 交换链接在吸引更多用户访问的同时也起到搜索引擎优化的作用。据权威调查显示, 全球 80% 的网站, 它们的访问量 70% ~ 90% 都是来自搜索引擎。如何让搜索引擎收录更多的网页, 就是提高网站访问量的最有效办法。

一般网站都会有很多友情链接, 而且随着网站不断发展, 友情链接的数量和内容会随时发生变化, 如果手工去修改就会相当麻烦。因此, 用户可以把友情链接功能集成一个模块, 实现模块化管理, 为以后网站的友情链接管理提供方便。本例将介绍如何智能管理网站内友情链接的思路和方法。

技术要点

- 增加和显示文本链接
- 增加和显示图片链接
- 多条件查询
- 网页嵌入技术
- 验证码技术
- 输入信息验证
- 链接管理

11.1 程序设计分析

本实例是一个开放式的网站链接交换和管理模块。要实现友情链接的交换功能, 就需要为网站提供一个增加链接的页面, 方便其他站长提交友情链接申请, 如果本站管理员审核通过后, 会自动在主页或者其他页面显示该友情链接。用户还可以对友情链接进行修改、删除、再审核等后台管理操作。本例友情链接浏览效果如图 11.1 所示, 设计流程如图 11.2 所示。

要实现本例功能, 读者应该学习和掌握下列技术。

PDG

- 显示链接。通过动态记录集绑定实现图标或文字的链接效果。
- 增加链接。利用表单技术，并结合插入记录服务器行为来实现。
- 管理链接。利用 Request 对象实现对链接信息的管理，主要包括删除、修改、审核等功能。
- 如何实现验证码功能。可以利用随机函数生成。

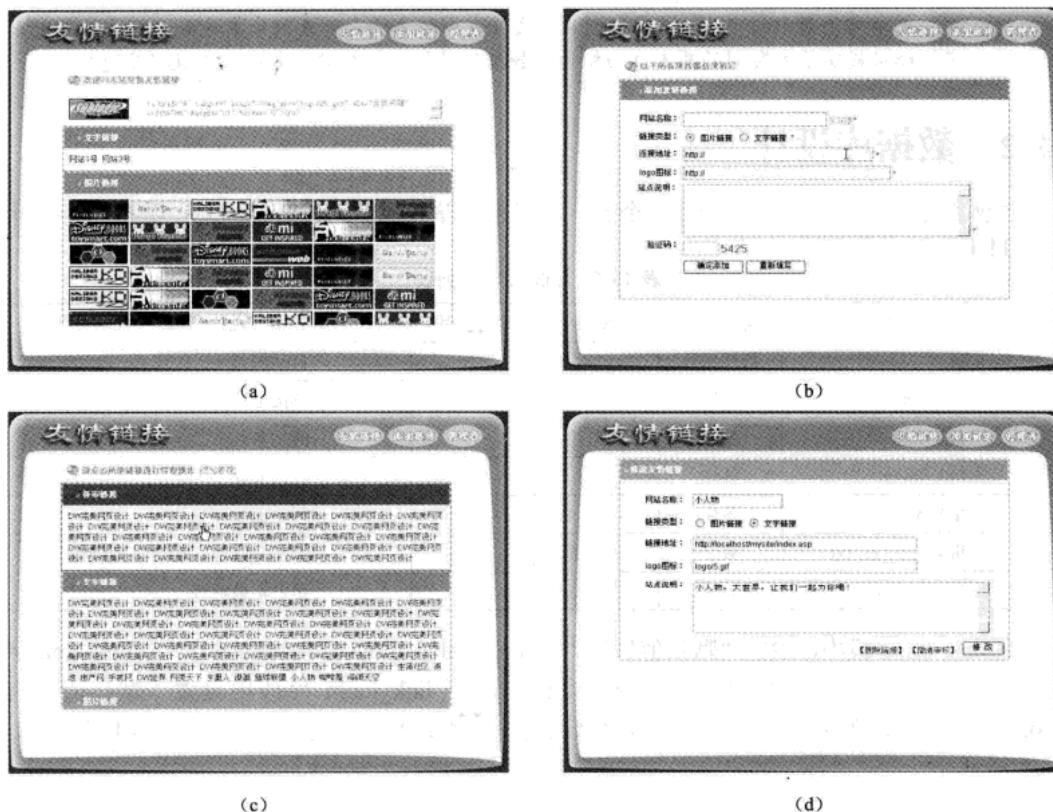


图 11.1 友情链接浏览效果

(a) 显示链接; (b) 增加链接; (c) 管理链接; (d) 审核并修改链接

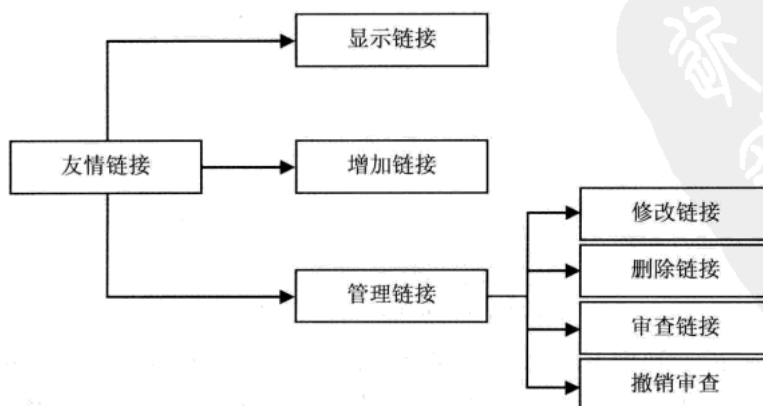


图 11.2 友情链接设计流程图



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，如 mysite（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”（详细设置说明可以参阅第 2.5.2 节介绍）。

把本书光盘中的“11”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

11.2 数据库设计

本例数据库 db.mdb 只有 link 一个表，它用来存储其他网站的链接信息，数据结构设计见表 11.1。

表 11.1 link 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
link_url	备注	-	-	链接地址
link_font	备注	-	-	网站名称
link_login	备注	-	-	网站 logo 存放网址
link_alt	备注	-	-	网站提示信息
link_logo_or_txt	数字	长整型	否	标示是图标，还是文本链接
link_yes_no	数字	长整型	否	是否审核通过

在 link_url 字段中，设置“默认值”为“http://localhost/mysite/index.asp”，在 link_font 字段中设置“默认值”为“DW 完美网页设计”，在 link_login 字段中设置“默认值”为“logo/logo.gif”，在 link_alt 字段中设置“默认值”为“DW 完美网页设计”，如图 11.3 所示。

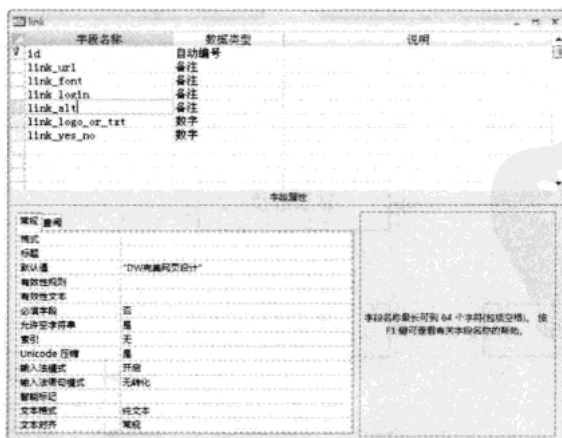


图 11.3 设置 link_alt 字段的默认值

link_logo_or_txt 字段用来判断是什么链接，值为 1 时表示文本链接，值为 0 时表示图标链接。link_yes_no 字段用来判断是否审核通过，值为 1 时表示该链接通过审核，值为 0 时表示该链接没有被审核。

11.3 页面设计

本实例共有 8 个文件，具体说明见表 11.2。

表 11.2 系统文件列表

文件名	说 明
index.asp	友情链接显示页面
add.asp	友情链接增加页面
admin.asp	友情链接管理主页面
links.asp	友情链接显示页面内嵌套页面，用来显示所有链接，可以把它作为一个功能模块移植到其他页面
admin_edit.asp	后台管理中针对选定的链接进行具体操作的页面
admin_links.asp	友情链接管理主页面内嵌套页面，用来显示所有链接，并可实现单击不同的链接进入操作页面 admin_edit.asp 进行具体操作。可以把它作为一个功能模块移植到其他页面
admin_save.asp	代码文件，根据具体的管理操作来删除、修改和审核链接
conn.asp	数据库连接文件，提供给 admin_save.asp 文件使用

下面就几个主要页面设计进行介绍。

11.3.1 友情链接显示页面

index.asp 文件主要用来显示友情链接，该页面设计效果如图 11.4 所示。当然在实际应用中，读者可以把友情链接做成一个.asp 文件，然后用#include 语句包含到其他页面中，这样就可以实现批量引用，而不需要在每个页面中去修改链接。

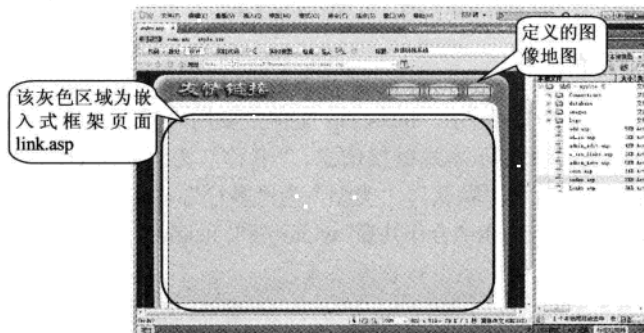


图 11.4 index.asp 文件设计效果

在该页面中，主要技术是嵌入框架的制作，这需要编写代码，如下所示：

```
<td align="center">
  <iframe id="datamain" src="links.asp" width=674 height=390 marginwidth=0
  marginheight=0 hspace=0 vspace=0 frameborder=0 scrolling=no ></iframe>
</td>
```

在上面方框内代码中，利用<iframe>标记实现网页的嵌入，并设置嵌入网页的宽为 674 像素，高为 390 像素，边框、边距、空隙都为 0，滚动条为无。

然后，在<iframe>标记下面再增加一个<div>标记，为<div>标记增加动态控制脚本，代码如下：

```

<td align="center">
<iframe id="datamain" src="links.asp" width=674 height=390 marginwidth=0 marginheight=0
hspace=0 vspace=0 frameborder=0 scrolling=no ></iframe>
<layer visibility=hide>
<div style="width:674px;" align="right"> <a href="#" onmouseover="scrollspeed=1"
onmouseout="scrollspeed=0" style='TEXT-DECORATION:
none;color:#ccc;font-family:Webdings'>5</a>
<a href="#" onmouseover="scrollspeed=1" onmouseout="scrollspeed=0" style='TEXT-
DECORATION: none;color:#ccc;font-family:Webdings'>6</a>
</div>
</layer>
</td>

```

在浏览器中浏览该页时，当鼠标移到该符号上，可实现页面的上下滚动，效果如图 11.5 所示。



(a)



(b)

图 11.5 实现动态滚动效果

(a) 嵌入页面上滚；(b) 嵌入页面下滚

11.3.2 链接显示嵌入页面

links.asp 页面设计效果如图 11.6 所示。links.asp 文件主要用来显示链接信息，布局比较简单，其中的文本域表单对象用来显示本站链接信息，“名称”为“textarea”，“字符宽度”为 70，“行数”为 2，“换行”设置为“默认”，“类型”为“多行”，“初始值”为“”。

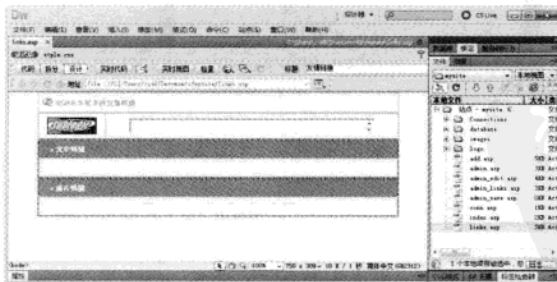


图 11.6 links.asp 页面设计效果

11.3.3 增加链接页面

add.asp 文件是一个表单域页面，表单设计如图 11.7 所示，其中表单属性设置见表 11.3。



图 11.7 add.asp 页面设计效果

表 11.3 增加链接表单设置列表

对象	名称	类型	设置说明
表单	form1	表单	“动作”设置为“add.asp?action=add”; “方法”设置为“默认”
文本框	t1	单行	“字符宽度”设置为 36, “最多字符数”为 36
文本框	t2	单行	“字符宽度”设置为 50, “最多字符数”为 100, “初始值”为“http://”
文本框	t3	单行	“字符宽度”设置为 55, “最多字符数”为 160, “初始值”为“http://”
文本框	t4	多行	“字符宽度”设置为 58, “行数”为 5, “换行”为“默认”
文本框	keys	单行	“字符宽度”设置为 4, “最多字符数”为 4
选择框	logoortxt	单选按钮	“选定值”为 1、0; 当“选定值”为 0 时, 设置“初始状态”为“已勾选”
按钮	Submit2	提交	“值”为“提交”, “动作”为“提交表单”
按钮	Submit	重置	“值”为“重新填写”, “动作”为“重置表单”

admin_edit.asp 页面设计和 add.asp 页面设计相同, 用户可以复制 add.asp 文件, 然后适当修改即可。

11.3.4 后台管理主页面

admin.asp 页面设计效果如图 11.8 所示。

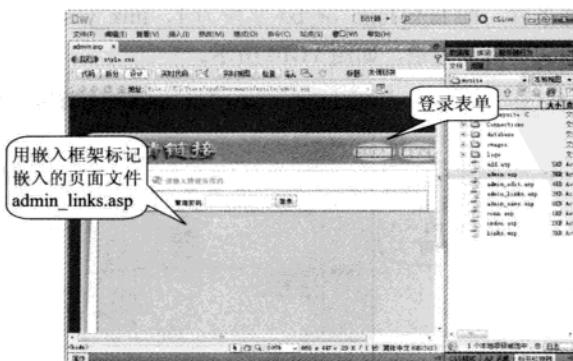


图 11.8 admin.asp 页面设计效果

admin_links.asp 页面设计比较简单, 如图 11.9 所示。

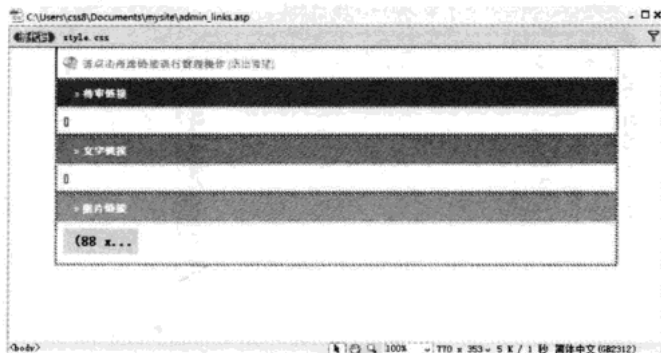



图 11.9 admin_links.asp 页面设计效果

11.4 前台功能实现

本实例模块的前台主要包括友情链接的显示和增加功能。下面就详细介绍这些功能的实现步骤。

11.4.1 显示友情链接

links.asp 文件主要用来显示友情链接, 实现该功能的操作步骤如下。

- 01** 定义 2 个记录集, 分别查询数据库中的所有文本链接和图标链接。启动 Dreamweaver CS5, 打开 links.asp 文件。选择【窗口】|【绑定】菜单命令, 打开【绑定】面板。
- 02** 单击【绑定】面板左上角的  按钮, 在弹出的下拉菜单中选择【记录集 (查询)】命令, 打开【记录集】对话框, 在【记录集】对话框中单击【高级】按钮, 切换到【记录集】对话框的高级状态, 设置【名称】为 Recordset1, 【连接】为 conn, 然后在【SQL】文本框中输入下面查询代码:

```
SELECT *
FROM link
WHERE link_logo_or_txt = 1 AND link_yes_no = 1
ORDER BY id DESC
```

其中设置 2 个筛选条件, 即 link_logo_or_txt 字段为 1, link_yes_no 字段为 1, 这样就可以筛选出所有审查过的文本链接记录, 如图 11.10 所示。



图 11.10 高级定义记录集

- 03 以同样的方式再定义1个记录集，也可以复制上面定义的记录集，方法是在【绑定】面板中右键单击 Recordset1，在弹出的菜单中选择【拷贝】命令，如图 11.11 所示。然后选择【编辑】|【粘贴】菜单命令，粘贴记录集 Recordset1 为 Recordset2。
- 04 在【绑定】面板中双击 Recordset2，在打开的【记录集】对话框中，修改 link_logo_or_txt 字段的变量【默认值】为 0，如图 11.12 所示。



图 11.11 复制记录集

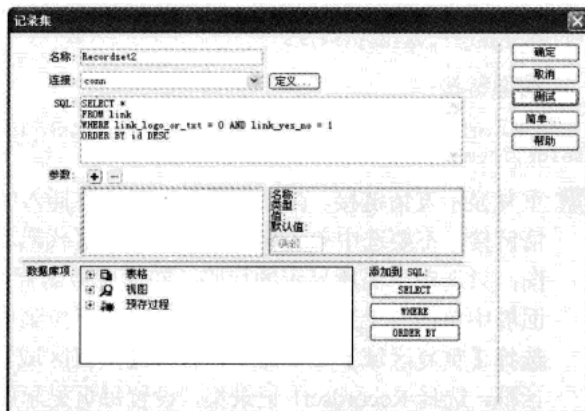


图 11.12 修改记录集查询参数

- 05 定义完记录集后，分别把 Recordset1 和 Recordset2 记录集绑定到“文字链接”栏目内和“图片链接”栏目内，如图 11.13 所示。



提示

“文字链接”栏目要绑定 Recordset1 记录集中的 link_font 字段，而“图片链接”栏目要把记录集绑定到图像的 src 属性上，方法是选择【插入】|【图像】菜单命令，在打开的【选择图像源文件】对话框中选择【选取文件名自】中的【数据源】单选按钮，然后在【域】文本框中选择“link_login”字段，如图 11.14 所示。



图 11.13 插入记录集

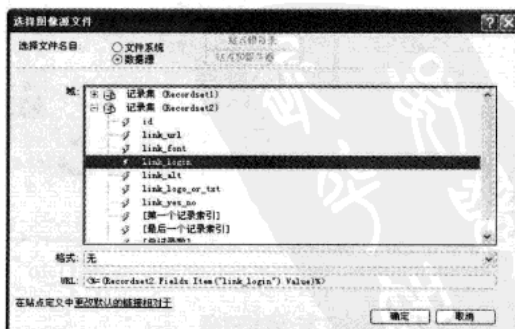



图 11.14 绑定图像的源文件数据源

- 06 定义文本和 Logo 图标的超链接。方法是选中绑定的字段和插入的图像，在属性面板中单击【链接】文本框右侧的【选择文件】按钮, 然后选择记录集 Recordset1 中的字段


link_url。显示的代码如下。注意，不要多选或少选，否则会出现错误。同时，为避免图像因为增加了超链接会显示粗蓝色的线框，可以为

文本链接：

```
<a href="<%= (Recordset1.Fields.Item("link_url").Value) %>"><%= (Recordset1.Fields.Item("link_font").Value) %></a>
```

图像链接：

```
<a href="<%= (Recordset2.Fields.Item("link_url").Value) %>">"></a>
```

- 07** 重复显示友情链接。首先在页面内选中要插入的友情链接，不要选中单元格，可以切换到【代码】视图，只选中上面所显示的代码。在【服务器行为】面板中单击左上角的按钮，在弹出的下拉菜单中选择【重复区域】菜单命令，打开【重复区域】对话框，选择 Recordset1 记录集，设置每页显示所有记录，如图 11.15 所示。

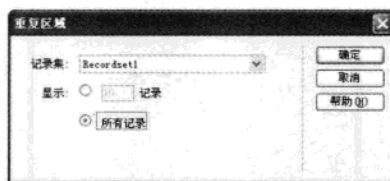


图 11.15 设置重复区域

- 08** 以同样的方法定义图片链接的重复显示区域。其中在【重复区域】对话框中选中 Recordset2 记录集，设置每页显示所有记录。单击【确定】按钮关闭对话框。按 F12 键浏览，效果如图 11.16 所示。

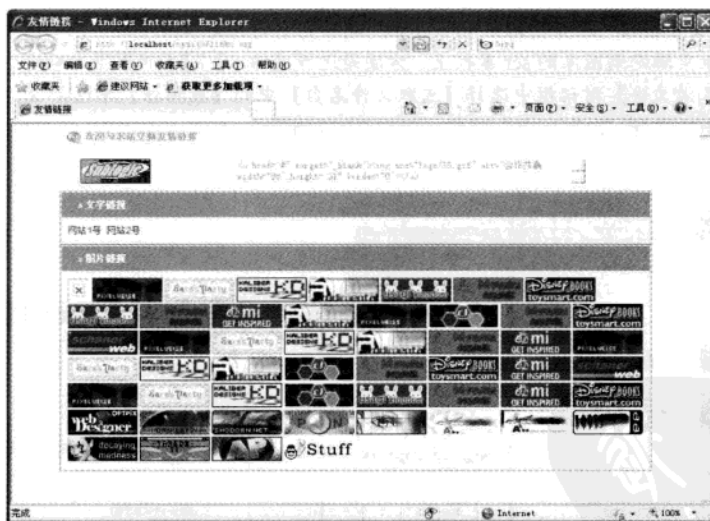


图 11.16 显示友情链接效果

11.4.2 增加友情链接

add.asp 文件可以实现友情链接的申请功能，以方便其他网站的站长能够快速提交链接信息，这样本站管理员可以在后台审核链接申请并决定是否确定接受该申请。实现增加友情链接功能的具体操作步骤如下。

01 要实现插入记录，应先定义一个记录集。在 Dreamweaver CS5 中，打开 add.asp 文件，选择【窗口】|【绑定】菜单命令，打开【绑定】面板，定义记录集 Recordset1，该记录集的设置如图 11.17 所示。

02 插入记录。在【服务器行为】面板中单击左上角的 \oplus 按钮，在弹出的下拉菜单中选择【插入记录】菜单命令，打开【插入记录】对话框，设置如图 11.18 所示。

在【连接】下拉列表中选择前面定义好的数据源 conn；在【插入到表格】下拉列表中选择要插入的数据表 link；在【插入后，转到】文本框中设置插入记录成功后要跳转到哪里，本例可以空着，则系统自动跳回本页。在【表单元素】列表框中设置表单对象值要插入到的数据库中的哪个字段。

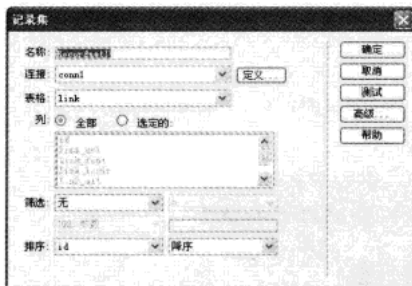


图 11.17 定义记录集

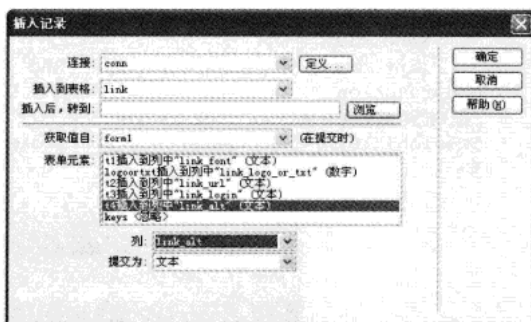


图 11.18 设置【插入记录】对话框

03 单击【确定】按钮，关闭对话框。然后返回到【设计】视图，选中表单 form1，在【属性】面板中修改【动作】文本框中的值“<%=MM_editAction%>”为“add.asp?action=add”，如图 11.19 所示。修改该值的目的是方便用户控制传递的参数，把系统插入记录时默认的值“<%=MM_editAction%>”废除，避免出错。

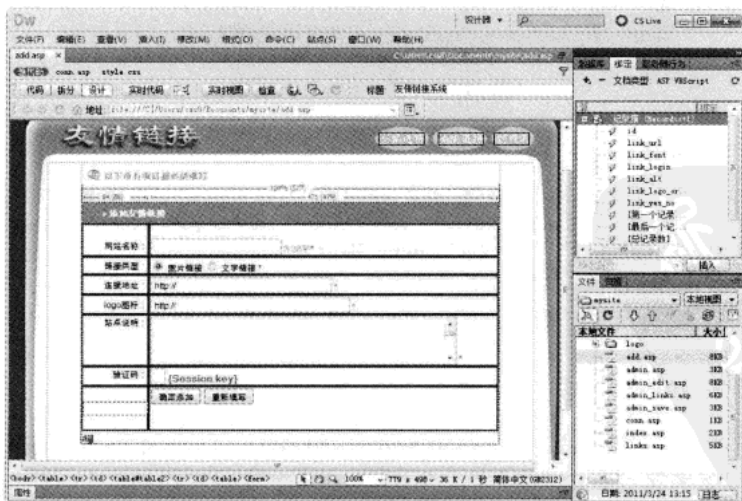


图 11.19 修改表单传递参数

本节完成了增加友情链接的主要功能，但还有许多细节技术没有介绍，如验证码技术实现、输入信息的验证、插入记录后的处理等操作，在下面几节中将详细介绍。

11.4.3 随机数字验证码功能实现

在第 7.6 节中曾经详细介绍了图形验证码功能的实现,下面再就随机数字验证码功能展开介绍。虽然随机数字验证码功能实现起来要简单得多,但它能够帮助读者开拓解题思路。实现本例验证码功能的具体操作步骤如下。

01 打开 add.asp 文件,在【设计】视图下,把光标置于图 11.20 所示的地方。

02 切换到【代码】视图下,然后输入下面代码:

```
<% '定义一个函数,获取一个0到9之间的随机数字
function linkaddkey()
    dim a, b
    a = 0
    b = 9
    Randomize
    linkaddkey = Int((b - a + 1) * Rnd + a)
end function
'把4位随机数保存到Session变量key中
Session("key")=linkaddkey() & linkaddkey() & linkaddkey() & linkaddkey()
%>
```

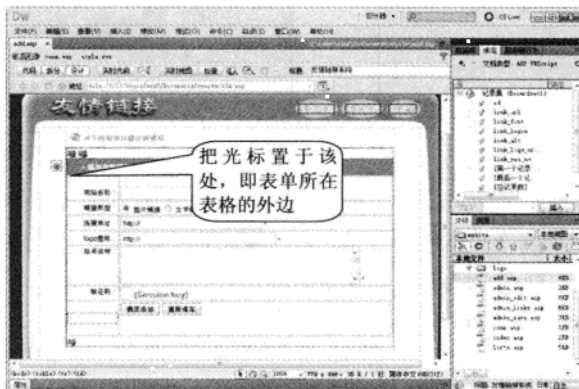


图 11.20 置于光标在页面中

03 在验证码文本框后面输入下面代码:

```
<td align="right" valign="top">验证码: </td>
<td width="471"><input name="keys" type="text" id="keys" size="4" maxlength="4">
<b><font color="#666666" size="3" face="Microsoft Sans
Serif"><%=Session("key")%></font></b></td>
```

这样就实现了随机产生一个验证码,并把它显示出来。

11.4.4 客户端信息验证

为了确保用户输入的信息合法,读者需要在 add.asp 文件中添加信息验证行为。方法有多种,读者可以参考第 6.5 节详细介绍。下面在 add.asp 文件中添加以下代码以实现更强大功能的智能验证。

```
<script type="text/javascript">
function checkform() {
//定义4个字符串变量,获取表单中的值
var str_t1 = document.form1.t1.value;
```

```

var str_t2 = document.form1.t2.value;
var str_t3 = document.form1.t3.value;
var str_t4 = document.form1.t4.value;
//判断输入的各个文本框中的信息是否合法
if ((str_t1=="") || ((str_t1.length)>36) || ((str_t1.length)<2) || (str_t2=="")
|| (str_t3=="") || (str_t2.substr(0,7)!="http://") || ((str_t2.length)<12) ||
((str_t1.length)>100) || (str_t3.substr(0,7)!="http://") || ((str_t3.length)>160) ||
((str_t1.length)<16) || (str_t4=="") || ((str_t4.length)>255) || ((str_t4.length)<2)) {
    alert('很抱歉, 请正确、完整的填写链接信息! ');
    return false;
}
}
</script>

```

上面代码为 JavaScript 客户端验证代码。先定义 4 个变量用来获取表单中的值, 然后通过一个条件语句分别判断每个文本框中输入的信息是否为空、包含的字符是否符合长度, 以及 URL 字符串中是否包含特定字符 “http://”。

最后, 在<form>标签中使用 onSubmmit="javascript:return checkform()"引用该函数即可。这样当用户单击提交按钮实现表单信息提交时, 首先会调用该脚本函数对用户输入信息进行验证, 如果合法则通过, 否则返回 false 值, 取消提交操作。

11.4.5 服务器端验证和表单提交处理

当用户在 add.asp 文件中输入友情链接信息后, 设想当单击【确定增加】按钮后, 通过了客户端 JavaScript 脚本的验证, 还需要在服务器端对用户输入的验证码进行确认, 如果一致则把信息插入到数据库中, 如果成功则显示提示信息, 否则返回客户端, 要求用户继续输入验证码。

实现该功能的操作步骤如下。

01 打开 add.asp 文件, 切换到【代码】视图, 在顶部输入下面代码, 最好位于“插入记录”服务器行为代码的上面。

```

<% '页面显示控制变量, 为1则显示表单, 为2则显示插入信息成功后提示信息
dim flag
flag = 1
'验证验证码是否正确
if Request("action")="add" then
    key1=Session("key")
    key2=Request("keys")
    if key1<> key2 then
        Response.write("<script>alert('很抱歉, 验证码输入错误! ');</script>")
        Response.write("<script>history.go(-1);</script>")
        Response.End()
    end if
%>

```

把该段代码放在页面最上面, 是避免插入记录到数据库后再验证信息, 以保证先验证信息的正确性后再插入记录。为了避免先插入记录, 设置表单 form1 的【动作】为“add.asp?action=add”, 这在上节已经详细介绍过。

这样在单击【提交】按钮后, 页面又跳转到本页, 并从最上面开始执行代码。程序先判断传递参数变量 action 是否是“add”, 如果判断为真, 则执行下面的代码, 先对用户输入的验证码进行比较, 如果一致则执行“插入记录”服务器行为代码。

02 在【代码】视图找到“插入记录”服务器行为代码，在其后输入下面代码：

```
<% 定义flag变量开关为2
flag = 2
end if
%>
```

03 在【设计】视图选中表单所在单元格，然后找到表单所包含的表格后面，如图 11.21 所示。输入下面代码：

```
<%
if flag = 1 then
%>





```

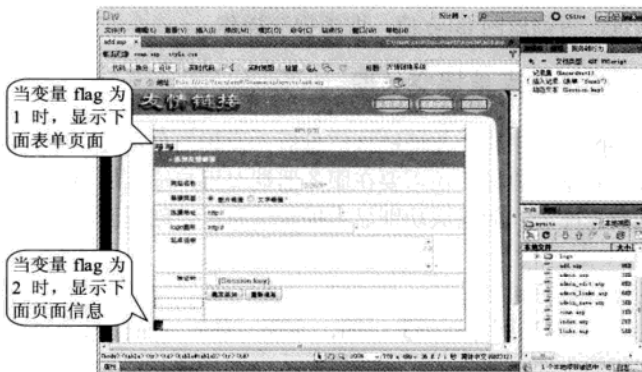


图 11.21 插入条件代码段

如果在运行时，浏览器显示下面错误提示信息：

错误类型：

Microsoft VBScript 运行时错误 (0x800A01A8)

缺少对象：''

/mysite/add.asp, 第 193 行

这说明在执行上面操作时，没有考虑到条件语句与 Recordset1 记录集对象的关系。因为定义的 Recordset1 记录集对象的代码包含在第 1 个条件语句结构中，而关闭记录集对象的语句处于第 2 个条件语句结构中，所以服务器就会报错，提示关闭对象时找不到 Recordset1 记录集对象。

解决的方法是为关闭记录集 Recordset1 的代码添加一个条件语句，判断只有用户执行插入记录操作时才会显示该语句（一般当读者在定义记录集时，生成的关闭记录集代码都放在 </html> 标签后面）。修改前的代码如下：

```
<%
Recordset1.Close()
Set Recordset1 = Nothing
%>
```

修改后的代码如下：

```
<%
if Request("action")="add" then
    Recordset1.Close()
    Set Recordset1 = Nothing
end if
%>
```

11.5 后台管理功能实现

后台功能主要包括友情链接的审核、修改和删除等操作。在编辑页面中可以实现对友情链接的增减和修改功能，以及对增加的友情链接进行审核和降级处理。

11.5.1 管理链接主页面

管理链接主页面（admin.asp）和显示链接页面（index.asp）在设计上有点相似，都有一个嵌入页面。管理链接主页面主要功能是链接显示，在操作之前首先要检验用户是否登录，如果没有登录则显示登录表单，如果登录成功则显示链接详细信息。

实现该页面功能的操作步骤如下。

- 01** 打开 admin_links.asp 文件，实现链接信息的显示。在 admin_links.asp 文件中定义 3 个记录集，分别查询没有经过审核的记录集、经过审核的图标链接记录集和文本链接记录集，如图 11.22 所示。

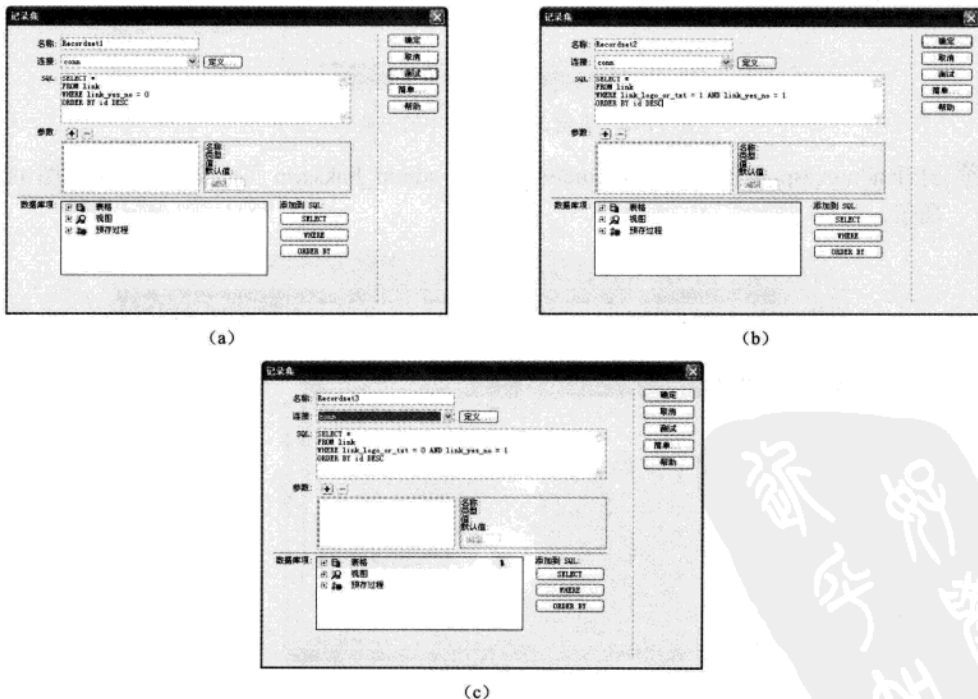


图 11.22 设置 3 个记录集

(a) 未经审核的记录集；(b) 经审核的图标链接记录集；(c) 经审核的文本链接记录集

- 02** 把每个记录集分别绑定到页面相应位置中, 并应用重复区域服务器行为, 操作后的页面如图 11.23 所示。具体操作方法可参照第 11.4.1 小节介绍。



注意

本节操作和第 11.4.1 小节操作主要区别在于设置链接的数据源不同。第 11.4.1 小节中链接的数据源为记录集中 link_url 字段, 如 `<%= (Recordsetn.Fields.Item("link_url").Value)%>`; 而本节中链接的文件为 admin_edit.asp, 没有绑定动态数据源, 但提供了传递参数, 该参数绑定了记录集中 id 字段, 如 `<a href="admin_edit.asp?id= <%= (Recordset1.Fields.Item("id").Value)%>" target="_top">`。

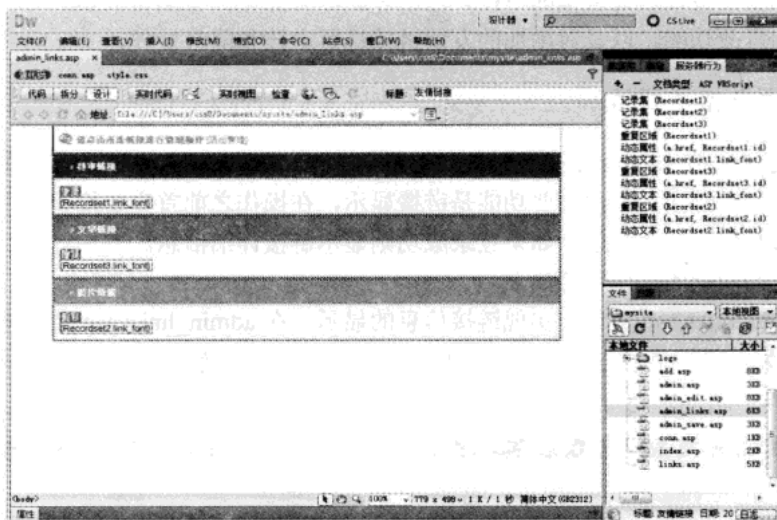


图 11.23 绑定 3 个记录集中的记录到页面中

- 03** 打开 admin.asp 页面, 利用 `<iframe>` 标记嵌入 admin_links.asp 页面文件, 效果如图 11.24 所示。

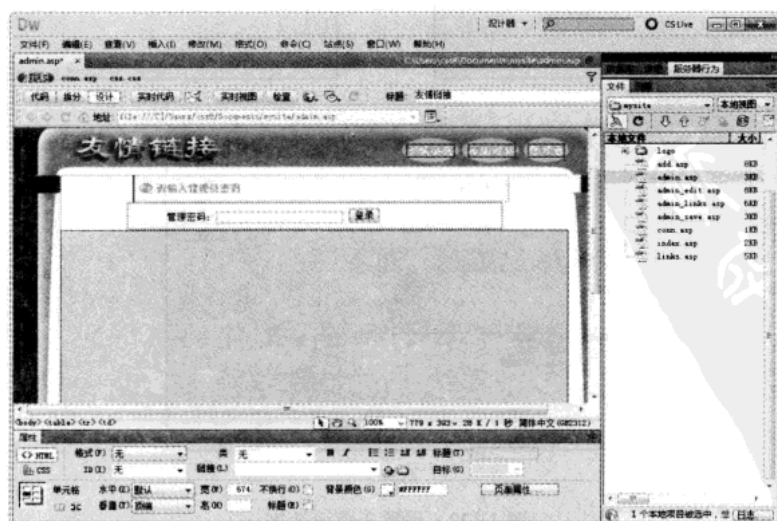


图 11.24 嵌入 admin_links.asp 页面文件

其详细代码如下:

```
<iframe id="datamain" src="admin_links.asp" width=674 height=380 marginwidth=0
marginheight=0 hspace=0 vspace=0 frameborder=0 scrolling=no ></iframe>
<layer visibility=hide>
  <div style="width:674px;" align="right"> <a href="#" onmouseover="scrollspeed=-1"
onmouseout="scrollspeed=0" style='TEXT-DECORATION:
none;color:#ccc;font-family:Webdings'>5 </a>
  <a href="#" onmouseover="scrollspeed=1" onmouseout="scrollspeed=0" style='TEXT-
DECORATION: none;color:#ccc;font-family:Webdings'>6</a> </div>
</layer>
```

04 下面利用 if 条件结构整合整个页面, 判断用户是否登录并显示不同的页面内容。首先把光标置于登录表单所在表格的前边, 如图 11.25 所示。

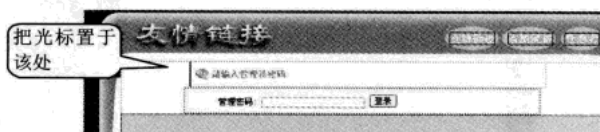


图 11.25 插入光标位置

05 切换到【代码】视图, 在顶部输入下面代码:

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<%
'获取用户输入信息, 并判断输入信息是否正确, 如果正确设置Session变量admin为admin, 如果错误, 弹出错误信息, 并返回
if Request.QueryString("action")="login" then
  if Request.Form("adminpass")="admin" then
    Session("admin")="admin"
  elseif Request.Form("adminpass") <> "admin" then
    Response.write("<script>alert('很抱歉, 密码输入错误, 请重新输入! ');</script>")
  end if
end if
%>
```

06 同样, 在登录表单前后包含一个条件语句, 用来控制没有登录用户进行登录。代码如下:

```
<%
'当Session变量admin不等于admin时, 也就是用户没有登录或者登录错误时, 显示登录表单页面, 让用户继续登录
if Session("admin") <> "admin" then
%>
  <table width="74%" border="0" align="center" cellpadding="8" cellspacing="0">
    <tr>
      <td>
      <b><font color="#999999">请输入管理员密码</font></b></td>
    </tr>
  </table>
  <table width="74%" height="17" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#F5F5F5">
    <form name="form1" method="post" action="admin.asp?action=login">
      <tr>
        <td height="36" bgcolor="#F9F9F9">管理密码:
        <input type="password" name="adminpass">
        <input type="submit" name="Submit" value="登录">
      </td>
    </tr>
  </form>
  </table>
<%
else
%>
```

- 07 在嵌入页面文件的后面插入条件结束标志代码:

```
<%  
end if  
%>
```

- 08 切换到【设计】视图, 可以看见插入的 ASP 代码段, 如图 11.26 所示。

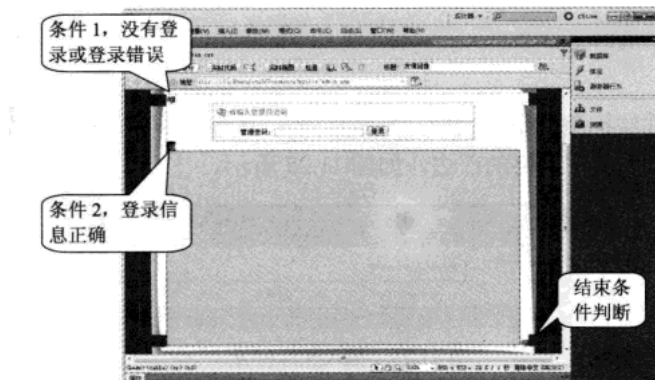


图 11.26 管理链接主页面

11.5.2 编辑主页面

当用户在 admin.asp 页面中选择一个要编辑的链接, 然后跳转到 admin_edit.asp 文件中, 这时可利用传递过来的参数 id 在表中显示该链接信息, 从而实现修改、删除和审核功能, 如图 11.27 所示。实现修改功能的具体步骤如下。

- 01 定义一个记录集, 以查询要编辑的记录。打开【绑定】面板, 单击面板左上角的 按钮, 在弹出的下拉菜单中选择【记录集 (查询)】命令, 然后设置【记录集】对话框, 如图 11.28 所示。

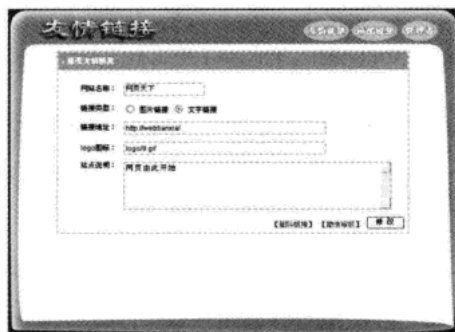


图 11.27 显示要编辑的记录

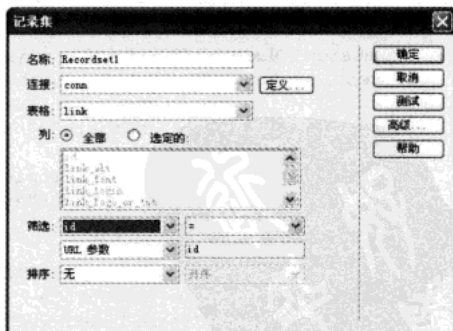


图 11.28 设置【记录集】对话框

- 02 把记录集 Recordset1 中各个字段分别绑定到表单中, 效果如图 11.29 所示。文本框绑定可先选中文本框, 然后在【属性】面板中单击【初始值】文本框右边的【绑定到动态数据源】按钮, 在打开的数据源对话框中选择要绑定的字段。单选按钮组绑定可先选中某个单选按钮, 然后在【属性】面板中单击【动态】按钮, 打开【动态单选按钮】对话框, 然后在【选取值等于】文本框中设置动态数据源, 如图 11.30 所示。

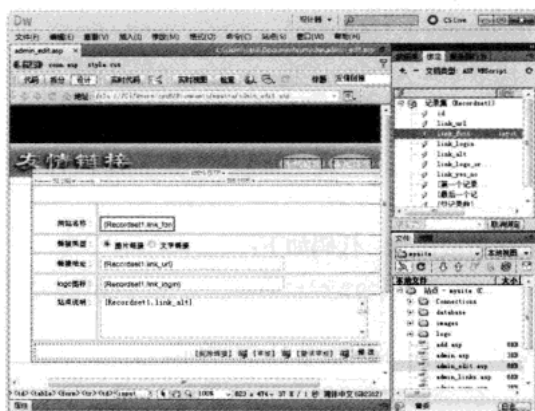


图 11.29 绑定记录集中各个字段

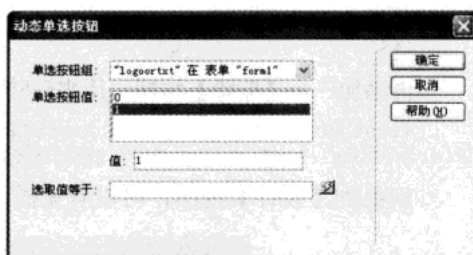


图 11.30 设置单选按钮的数据源

- 03 打开【服务器行为】面板，单击面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【更新记录】命令，打开【更新记录】对话框。
- 04 在【连接】下拉列表中选择前面定义好的数据源，如 conn；在【要更新的表格】下拉列表中选择要更新的数据表，如 link；在【选取记录自】下拉列表中选择定义的记录集，如 Recordset1；在【唯一键列】下拉列表中选择被定义为主键的字段，如 id；在【在更新后，转到】文本框中设置更新记录成功后要跳转到哪里，如 admin.asp；在【获取值自】下拉列表中选择表单名称，如 form1；在【表单元素】列表框中设置每个表单对象所要对应的数据表中的字段，如图 11.31 所示。



图 11.31 设置【更新记录】对话框

- 05 单击【确定】按钮，即可增加更新记录服务器行为到页面中，这样就可以把修改的数据写入数据库中。
- 06 在“修改”提交按钮前面增加一个条件语句，判断当前链接是否通过审核，并根据条件显示不同的操作按钮。代码如下：

```
<% '如果操作的记录是没有审核通过的记录，则显示【审核】文本按钮
if Recordset1.Fields.Item("link_yes_no").Value=0 then
%>
<a
href="admin_save.asp?action=ok&id=<%= (Recordset1.Fields.Item("id").Value) %>"><font
color="#FF0000">【审核】</font></a>
<%
else
```

```

'如果操作的记录为审核通过的记录,则显示【撤销审核】文本按钮
if Recordset1.Fields.Item("link_yes_no").Value =1 then
%>
<a
href="admin_save.asp?action=no&id=<%= (Recordset1.Fields.Item("id").Value) %>"><font
color="#FF0000">【撤销审核】</font></a>
<%
end if
%>

```

再添加一个【删除】按钮,用来删除当前友情链接信息。代码如下:

```

<font color="#FF0000"> <a href="admin_save.asp?action=delete&id= <%= (Recordset1.
Fields.Item ("id").Value) %>">【删除链接】</a></font>

```

切换到【设计】视图,可以看见插入的 ASP 代码图标,如图 11.32 所示。

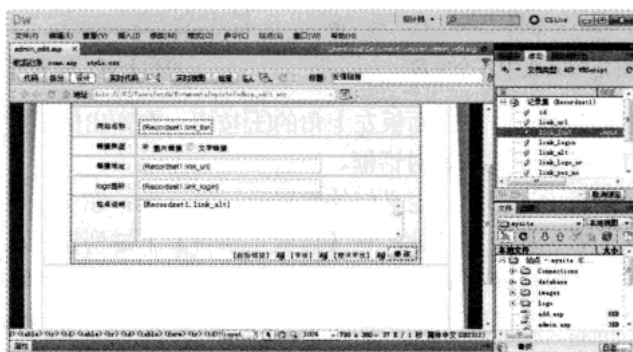


图 11.32 添加操作按钮

11.5.3 删除、审核和退出管理

admin_edit.asp 文件主要用来显示从 admin.asp 文件传递过来的 id 编号的链接信息,然后在该页面进行具体修改、删除和审核功能。其中修改功能在页面中利用服务器行为直接进行修改,而删除和审核功能要在 admin_save.asp 文件中用代码来进行操作完成。

打开 admin_save.asp 文件,输入下面代码:

```

<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<script language=javascript>
function confirm_button(){
    if (confirm("真的要删除吗,添加上去很辛苦的呢,考虑一下吧?")){
        return true;
    }
    return false;
}
function goHist(a) {
    history.go(a);
}
</script>
<%'打开数据库连接
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open MM_conn_STRING
%>
<%'根据查询字符串参数action的取值执行不同的操作
if request("action")="ok" then '审核通过
    id=htmls(encodestr(request("id")))
    StrSql = "select * from link where id=" & id

```

```

Set rs = server.CreateObject("ADODB.RecordSet")
rs.Open StrSql,conn,1,2
rs("link_yes_no")=1
rs.update
rs.close
response.Redirect("admin.asp")
elseif request("action")="no" then '撤销审核
id=htmls(encodestr(request("id")))
StrSql = "select * from link where id=" & id
Set rs = server.CreateObject("ADODB.RecordSet")
rs.Open StrSql,conn,1,2
rs("link_yes_no")=0
rs.update
rs.close
response.Redirect("admin.asp")
elseif request("action")="delete" then '删除友情链接
id=htmls(encodestr(request("id")))
conn.execute("delete * from link where ID=" & id)
response.Redirect("admin.asp")
elseif Request("action")="logout" then '退出用户登录
Session.Abandon()
Response.Redirect("index.asp")
end if
%>
<%
'定义两个函数，实现对字符串的转换
Function htmls(shtml)
Dim objRegExp, shtmls
Set objRegExp = New RegExp
objRegExp.IgnoreCase = True
objRegExp.Global = True
objRegExp.Pattern = "<.+?>"
shtmls = objRegExp.Replace(shtml, "")
htmls = shtmls
Set objRegExp = Nothing
End Function
function encodestr(str)
dim i
str=Trim(str)
str=Replace(str,"<","&lt;")
str=Replace(str,">","&gt;")
str=Replace(str,"'","''")
str=Replace(str,vbCrLf&vbCrLf,"</p><p>")
str=Replace(str,vbCrLf,"<br>")
HTMLEncode = str
encodestr=replace(str," ","")
encodestr=replace(str,"&nbsp;","")
if IsNumeric(encodestr) then
str=encodestr
else
str=0
end if
end function
%>

```

11.6 实践总结

本章详细介绍了友情链接系统的设计方法，系统兼顾前台链接显示和后台链接管理，是一个功能比较完善的友情链接管理系统。设计时考虑链接显示和管理链接的双重功能，因此，实例分设了几个功能不同的模块页面，组合成一个完整的系统功能。

第 12 章

Dreamweaver

网站搜索模块

互联网是一个巨大的信息海洋，每个网站就是这个信息汪洋中的一个孤岛。网上信息浩瀚无边且毫无秩序，所有的信息象汪洋上的一个个小岛，网页链接是这些小岛之间纵横交错的桥梁。搜索引擎能够自动从互联网上搜集信息，经过一定整理以后，把所有需要的信息有秩序地提供给用户，以方便快速浏览。

虽然搜索引擎是一项比较高级的 Web 应用技术，现在已经发展到人工智能、模糊查询等高级阶段。但是作为一名初学者能够掌握简单的搜索技巧并应用于自己的网站内也是一个不错的选择。站内搜索是网站开发中必不可少的一个功能模块。本例将在前面学习基础上介绍高级查询技巧的实现方法，主要包括模糊查询、范围查询等查询方式。

技术要点

- SQL 查询方法
- WHERE 条件设计
- 数据排查
- 字符串处理方法
- 获取客户端系统信息

12.1 程序设计分析

搜索技术的实现实际上就是如何设计并优化 SQL 查询，SQL 查询是动态网站使用最频繁的一项技术，灵活运用 SQL 查询可以实现各种实用功能，它在网页编程中作用非常大。本例将在前面实例基础上深入探索 SQL 查询方法。

由于一个网站会有很多信息，限于篇幅以及方便读者学习，本书以如何快速查询手机和 IP 地址号段信息，并以此为处理对象，介绍如何快速准确地找到手机或 IP 号码信息，效果如图 12.1 所示。可能实际设计中的站内搜索功能略有差别，但如果掌握了基本的 SQL 查询方法，相信读者只要简单地进行方法迁移就可以实现更多搜索功能。

要实现本例功能，读者需要掌握和解决下面技术问题。

- 如何处理复杂模糊的信息，使其符合电脑查询的要求，可以使用字符串函数进行处理。
- 如何在数据库中查询符合条件的记录。可以定义不同的记录集，利用条件结构进行排查筛选。
- 如何获取客户端系统信息。可以利用 Server 对象的系统变量获取。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“12”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。



(a)



(b)



(c)



(d)

图 12.1 高级查询浏览效果

(a) 查询 IP 地址；(b) 反馈 IP 地址信息；(c) 查询手机地址；(d) 反馈手机地址信息

12.2 数据库设计

本例数据库 db.mdb 包括 ip 和 tel 两个表。ip 表用来存储 IP 号码的全球基本信息，数据结构设计见表 12.1。tel 表用来存储所有手机号段信息，数据结构设计见表 12.2。

表 12.1 ip 表字段列表

字 段	类 型	字段大小	必填字段	说 明
startip	文本	16	否	IP 号段的起始号码
endip	文本	16	否	IP 号段的结束号码
country	文本	14	否	IP 号段所属国家
local	文本	80	否	IP 号段所属具体区域

表 12.2 tel 表字段列表

字 段	类 型	字段大小	必填字段	说 明
start	数字	长整型	否	手机号段的起始号码
end	数字	长整型	否	手机号段的结束号码
city	文本	20	否	手机号段所属省市
city1	文本	20	否	手机号段所属城市
url	文本	20	否	手机号段所在区号和种类

12.3 页面设计

本实例共有 5 个文件，页面具体功能说明见表 12.3。

表 12.3 文件列表

文件名	说 明
index.asp	高级查询主页面
search_tel.asp	查询手机号码信息页面
search_ip.asp	查询 IP 号码信息页面
search_result.asp	查询反馈页面
conn.asp	定义数据库连接页面

下面就几个主要页面设计进行简单介绍。

12.3.1 高级查询主页面

index.asp 页面主要由两部分组成，即顶部的标题、导航栏和下半部分嵌入的框架页面，页面设计效果如图 12.2 所示。



图 12.2 index.asp 页面设计效果

在该页面中使用了嵌入框架技术和浮动层，详细代码如下。

```
<iframe id="main" name="main" src="#" width=100% height=100% marginwidth=0 marginheight=0 hspace=0 vspace=0 frameborder=0 scrolling=no ></iframe>
```

12.3.2 查询页面

search_tel.asp 和 search_ip.asp 页面分别用来查询手机和 IP 信息, 整个页面设计如图 12.3 所示。

在 search_tel.asp 和 search_ip.asp 页面中设计为简单的表格, 在表格单元格中插入一个文本框和按钮对象, 以方便用户输入信息进行查询。

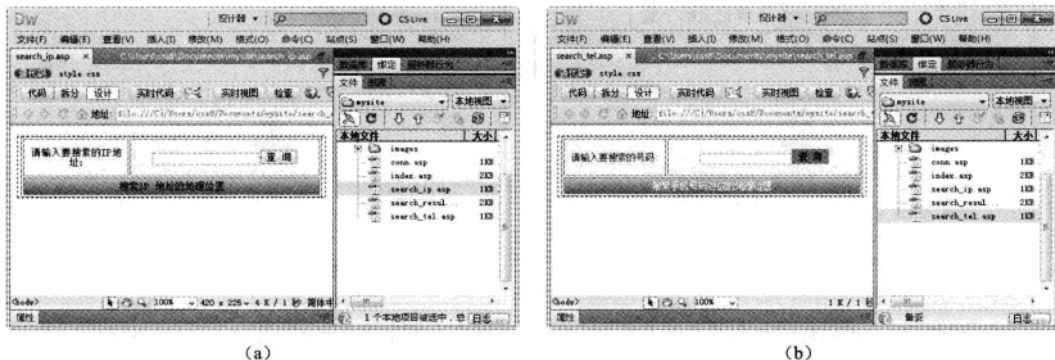


图 12.3 查询页面设计效果

(a) 查询 IP 地址页面; (b) 查询手机地址页面

12.3.3 查询结果页面

search_result.asp 页面用来显示查询结果, 设计效果如图 12.4 所示。该页面包括两个结构相同的表格, 分别用来显示 IP 和手机查询信息。利用条件语句判断用户查询的信息, 然后根据条件显示其中一个表格。

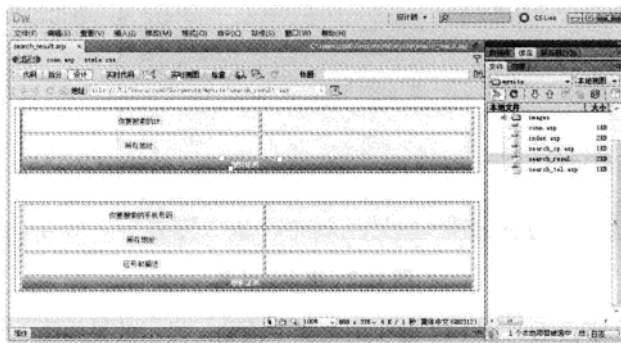


图 12.4 查询结果页面设计效果

12.4 基本搜索功能实现

下面从三个方面介绍基本搜索功能的实现, 包括计算查询数据库中的总记录数、利用服务器端的环境变量获取客户端的 IP 地址信息、利用用户输入的信息查询相近信息。

12.4.1 查询数据库记录数

当数据库中的记录非常多时, 利用 SQL 语句可以快速查询数据库中的记录数, 具体操作

步骤如下。

01 在 Dreamweaver CS5 中, 打开 index.asp 文件。选择【窗口】|【绑定】菜单命令, 打开【绑定】面板, 定义记录集 Recordset1, 设置如图 12.5 所示。

02 以同样的方式再定义一个记录集, 用来查询表 tel 中的记录数, 如图 12.6 所示。

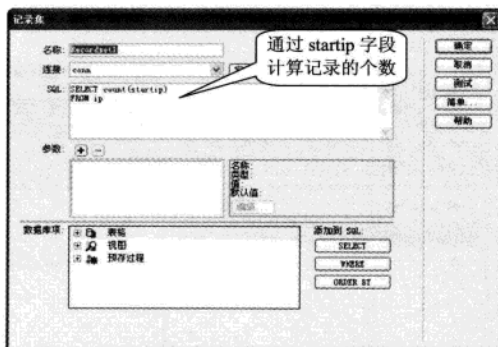


图 12.5 查询数据表 ip 记录个数集

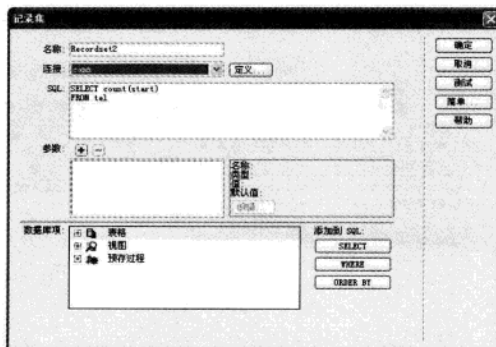


图 12.6 查询数据表 tel 记录个数集

03 切换到【设计】视图, 把记录集的 Expr1000 字段绑定到页面中, 该字段是一个特殊的字段, 它能够统计当前记录集的记录个数, 如图 12.7 所示。

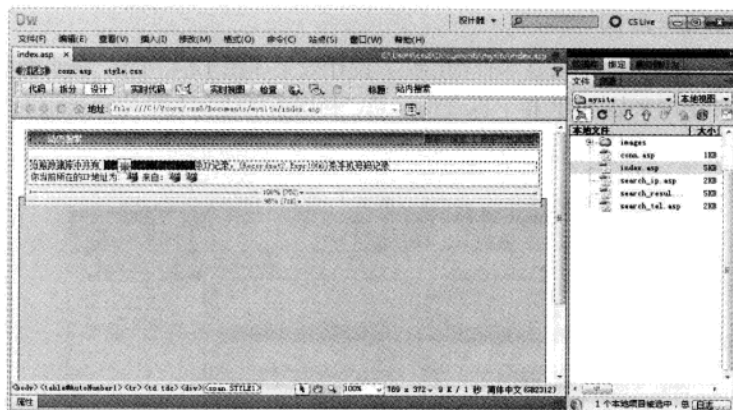


图 12.7 绑定记录集

04 绑定 Expr1000 字段后, 按 F12 键在浏览器中预览, 显示效果如图 12.8 所示。

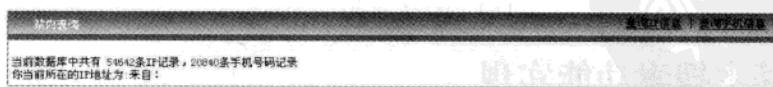


图 12.8 查询数据表中记录个数



提示

也可以使用另外一种方法来查询数据表中记录的个数。先定义一个记录集, 查询数据表中所有的记录, 即不设置 WHERE 条件语句, 并设置记录集的指针类型 CursorType 属性为 1 或者 2, 再利用记录集对象的 RecordCount 属性也可以获取定义记录集的个数, 不过这种方法效率比较低。

12.4.2 获取客户端系统信息

前面介绍过使用 Form、QueryString、Cookies 等数据集合获取客户端所传递的数据。ServerVariables 数据集合负责取得 Web 服务器端的环境变量 (Server Environment Vailable), 环境变量描述见表 12.4 (更详细的说明请参考第 3.6.8 节详细介绍)。

表 12.4 常用 ServerVariables 环境变量

环境变量名称	说 明
ALL_HTTP	客户端发送的所有 HTTP 标题文件
AUTH_TYPE	这是用户访问受保护的脚本时, 服务器用于检验用户是否为合法用户
CONTENT_LENGTH	发送到客户端的文件长度
CONTENT_TYPE	发送到客户端的内容的数据类型
GATEWAY_INTERFACE	服务器使用的 CGI 规格的修订。格式为 CGI/revision
LOCAL_ADDR	返回接受请求的服务器地址。如果在绑定多个 IP 地址的多宿主机器上查找请求所用的地址时, 这条变量非常重要
LOCAL_USER	用户登录 Windows NT 的账号
PATH_INFO	客户端提供的额外路径信息。可以使用这些虚拟路径和 PATH_INFO 服务器变量访问脚本
QUERY_STRING	查询 HTTP 请求中间号 (?) 后的信息
REMOTE_ADDR	发出请求的远程主机的 IP 地址
REMOTE_HOST	发出请求的主机名称
REMOTE_USER	用户发送的未映射的用户名字符串。该名称是用户实际发送的名称, 与服务器上验证过滤器修改过后的名称相对
REQUEST_METHOD	该方法用于提出请求。对 HTTP 的请求方式, 可以是 GET、HEAD、POST 等方法
SCRIPT_NAME	执行脚本的虚拟路径。用于自引用的 URL
SERVER_NAME	服务器主机名或 IP 地址
SERVER_PORT	发送请求的端口号
URL	提供 URL 的相对网址

下面利用 HTTP_X_FORWARDED_FOR 和 REMOTE_ADDR 变量获取客户端 IP 地址。在 index.asp 文件的【代码】视图中插入下面代码:

```
<%
'获取用户ip信息
myip = Request.ServerVariables("HTTP_X_FORWARDED_FOR")
if myip = "" or myip="unknown" then myip = Request.ServerVariables("REMOTE_ADDR")
'拆分ip, 目的是为了补足3位的补0, 同时也判断ip地址是否合法
myip1 = Split(myip, ".")
if Ubound(myip1)=3 then
    For i=0 To Ubound(myip1)
        if len(myip1(i))=2 then
            myip1(i)="0"&myip1(i)
        elseif len(myip1(i))=1 then
            myip1(i)="00"&myip1(i)
        end if
    next
    myip2=myip1(0)& "." &myip1(1)& "." &myip1(2)& "." &myip1(3)
```

'重新组合ip后,与数据库的ip地址比较,获取地址,采用三种不同方法的目的是为了当前两个ip段相同时,得到的地址比较具体

```

sql1="select * from ip where startip<='"&myip2&"' and endip>='"&myip2&"'and
left(startip,3)='"&myip1(0)&"' and left(endip,3)='"&myip1(0)&"' and
mid(startip,5,3)='"&myip1(1)&"' and '"&myip1(1)&"=mid(endip,5,3) and
mid(startip,9,3)='"&myip1(2)&"' and '"&myip1(2)&"=mid(endip,9,3) "
set rsl=conn.execute(sql1)
if not rsl.eof then
    al=rsl("country")
    bl=rsl("local")
else
    sql="select * from ip where startip<='"&myip2&"' and endip>='"&myip2&"'and left
(left(startip,3)='"&myip1(0)&"' and left(endip,3)='"&myip1(0)&"' and mid(startip,5,3)=
'"&myip1(1)&"' and '"&myip1(1)&"=mid(endip,5,3) "
set rs=conn.execute(sql)
if not rs.eof then
    al=rs("country")
    bl=rs("local")
set rs=nothing
else
    sql2="select * from ip where startip<='"&myip2&"' and endip>='"&myip2&"'
and left(startip,3)='"&myip1(0)&"' and left(endip,3)='"&myip1(0)&"'
set rs2=conn.execute(sql2)
if not rs2.eof then
    al=rs2("country")
    bl=rs2("local")
set rs2=nothing
else
    sql3="select * from ip where startip<='"&myip2&"' and endip>='"&myip2&"'
set rs3=conn.execute(sql3)
if not rs3.eof then
    al=rs3("country")
    bl=rs3("local")
set rs3=nothing
else
    a="找不到地址! "
end if
end if
end if
end if
else
    al="抱歉,找不到地址! "
end if
%>

```

在上面代码段中,方框中的代码可以实现获取客户端 IP 信息。接下来的代码是对 IP 地址进行规范化处理,然后与数据库中的 IP 地址进行比较,以便获取该 IP 的地理位置信息。有关这些代码的详细介绍可参见下面几节。最后,把获取的详细信息变量插入到页面中,如图 12.9 所示。详细代码如下:

```

<td height=22 bgcolor="#EFF1F3" class=tdc><br />
<div align="left">当前数据库中共有 <span
class="STYLE1"><%= (Recordset1.Fields.Item("Expr1000").Value)%></span>条IP记录, <span
class="STYLE1"><%= (Recordset2.Fields.Item("Expr1000").Value)%></span>条手机号码记录</div>
    你当前所在的IP地址为: <span class="STYLE1"><%=myip%></span>来自: <span class="STYLE1">
<%=al%> <%=bl%></span>
</td>

```

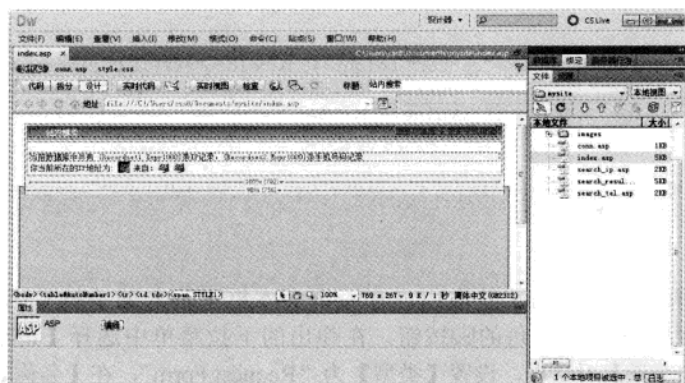



图 12.9 显示客户端 IP 详细信息

12.4.3 查询手机号码

当用户在表单中输入要查询的手机号码或者 IP 地址信息时,运用 SQL 语句可以实现数据库搜索该号码所在的区段位置。

下面以手机号码查询为例进行介绍,具体操作步骤如下。

- 01** 打开 search_result.asp 文件,选择【窗口】|【绑定】命令,打开【绑定】面板,单击面板左上角的  按钮,在弹出的下拉菜单中选择【记录集(查询)】命令。

- 02** 打开【记录集】对话框,切换到高级的【记录集】对话框,设置【记录集】对话框查询,如图 12.10 所示。切换到代码视图,修改其中的查询字符串代码:

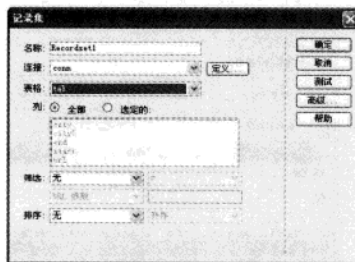


图 12.10 定义手机号码查询记录集

```
<%
Dim Recordset1
Dim Recordset1_cmd
Dim Recordset1_numRows
```

```
Set Recordset1_cmd = Server.CreateObject ("ADODB.Command")
Recordset1_cmd.ActiveConnection = MM_conn_STRING
Recordset1_cmd.CommandText = "SELECT * FROM tel where [start]<= "&c&" And "&c&"<= [end]"
Recordset1_cmd.Prepared = true
Set Recordset1 = Recordset1_cmd.Execute
Recordset1_numRows = 0
%>
```

- 03** 设置完毕,单击【确定】按钮返回【设计】视图,把记录集字段绑定到页面表格中,如图 12.11 所示。具体代码如下:

```
<table >
<tr>
<td width="28%" align="center" height="40" bgcolor="#EFF1F3" class=tdc>你要搜索的手机号码:</td>
<td width="75%" align="center" height="40" bgcolor="#EFF1F3" class=tdc><%=Request.Form("tel") %></td>
</tr>
<tr>
<td width="28%" align="center" height=40 bgcolor="#EFF1F3" class=tdc>所在地址:</td>
<td width="75%" align="center" height=22 bgcolor="#EFF1F3" class=tdc><%=Recordset1.Fields.Item("city").Value %> <%=Recordset1.Fields.Item("city1").Value %></td>
```

```

</tr>
<tr>
  | |
```


- 04** 定义一个请求变量，用来获取用户在查询文本框中输入的手机号码。在【绑定】面板中，单击面板左上角的按钮，在弹出的下拉菜单中选择【请求变量】命令。
- 05** 打开【请求变量】对话框，设置【类型】为“Request.Form”，在【名称】文本框中输入 tel，即查询文本框的名称，如图 12.12 所示。



图 12.11 绑定记录集中字段

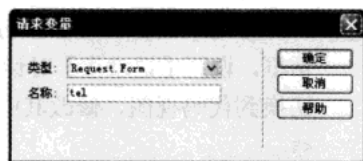


图 12.12 定义【请求变量】对话框

- 06** 在【绑定】面板中，单击“Request”定义变量选项，把其中的 Form.tel 绑定到页面单元格中，如图 12.13 所示。

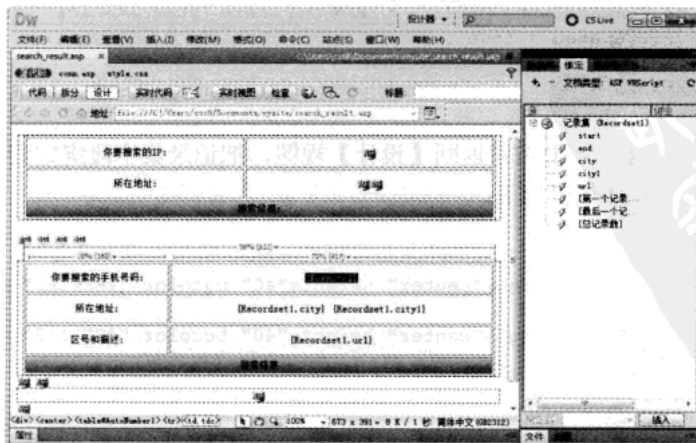


图 12.13 插入变量

- 07 由于用户输入的手机号码一般是 11 位数字,而数据库中的手机号码区段是以前 7 位进行存储的,因此还需要对用户输入的手机号码进行处理,即去掉后 4 位数字。下面通过编写代码的方式加以实现。切换到【代码】视图,在代码的顶部输入下面语句:

```
<%
'手机号码处理
if request.form("search_tel")<>" " then
    d= request.form("tel")
'判断是否为数字或者大于7位的数字
    if len(d)>=7 and IsNumeric(d) then
        c=left(d,7)
    else
        a="错误:请确保你输入的是大于7位的数字!"
    end if
end if
%>
```

- 08 上面的语句实现了对手机号码的判断和处理,但是已定义的记录集 Recordset1 变量参数是直接来自表单文本框中获取的,所以应修改其中的参数,把上面的变量 C 传递给记录集参数。方法是在【代码】视图下,找到定义记录集 Recordset1 的代码段,修改其中的代码如下:

```
<!--#include file="Connections/conn.asp" -->
<%
Dim Recordset1_MMColParam
Recordset1_MMColParam = "1"
If (Request.Form("tel") <> " ") Then
'原代码
'Recordset1_MMColParam = Request.Form("tel")
'修改后的代码
Recordset1_MMColParam = c
End If
%>
<%
Dim Recordset1
Dim Recordset1_numRows
Set Recordset1 = Server.CreateObject("ADODB.Recordset")
Recordset1.ActiveConnection = MM_conn_STRING
Recordset1.Source = "SELECT * FROM tel WHERE start <= " + Replace(Recordset1_MMColParam,
"", "'') + " AND end >= " + Replace(Recordset1_MMColParam, "''", "") + ""
Recordset1.CursorType = 0
Recordset1.CursorLocation = 2
Recordset1.LockType = 1
Recordset1.Open()
Recordset1_numRows = 0
%>
```

- 09 保存并在浏览器中预览,效果如图 12.14 所示。



图 12.14 查询预览效果

12.5 完善搜索功能

利用上节介绍的方法实现的查询功能还存在很多缺点,例如,用户输入的信息不符合要求,这是由于记录集找不到所指内容而出错,由于电话号码或 IP 地址不是规范的数字,查询难度会非常大,还需要用代码的形式对电话号码或 IP 地址进行处理,使其符合规范要求。

12.5.1 完善手机号码查询

对于上一节中的介绍过程在实践中还存在缺陷,例如,如果用户输入的号码不是标准的手机号码,页面就会显示错误。如果稍稍修改其中的代码,功能会更完善,具体步骤如下:

01 在页面代码的顶部增加下面代码:

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="conn.asp" -->
<% '手机号码处理
if request.form("search_tel")<>" then
d= request.form("tel")
'判断手机号码前3位是否合法
g = left(d,3)
if g<130 or g>139 then
a = "错误:你输入的号码非法! <br><a href='search_tel.asp'>返回</a>重新输入"
else
'判断是否为数字或者大于7位的数字
if len(d)>=7 and IsNumeric(d) then
c=left(d,7)
c=Cstr(c)
%>
```

02 把条件判断的结束语句移到页面代码的最末尾。具体代码如下:

```
<%
Dim Recordset1
Dim Recordset1_cmd
Dim Recordset1_numRows

Set Recordset1_cmd = Server.CreateObject ("ADODB.Command")
Recordset1_cmd.ActiveConnection = MM_conn_STRING
Recordset1_cmd.CommandText = "SELECT * FROM tel where [start]<="&c&" And "&c&"<=[end]"
Recordset1_cmd.Prepared = true

Set Recordset1 = Recordset1_cmd.Execute
Recordset1_numRows = 0
If Not Recordset1.EOF Then
%>
<div align="center">
<center>
<table border="1" cellpadding="4" style="border-collapse: collapse" bordercolor
="#0092f2" width="98%" id="AutoNumber1">
<tr>
<td width="28%" align="center" height="40" bgcolor="#EFF1F3" class=tdc>你要搜索的手机号码:</td>
<td width="75%" align="center" height="40" bgcolor="#EFF1F3" class=tdc><%=Request.Form("tel") %></td>
</tr>
<tr>
<td width="28%" align="center" height=40 bgcolor="#EFF1F3" class=tdc>所在地址:</td>
<td width="75%" align="center" height=22 bgcolor="#EFF1F3" class=tdc><%=Recordset1.Fields.Item("city").Value%>
```

```

<%= (Recordset1.Fields.Item("city1").Value) %></td>
</tr>
<tr>
<td width="28%" align="center" height=40 bgcolor="#EFF1F3" class=tdc>区号和描
述:</td>
<td width="75%" align="center" height=22 bgcolor="#EFF1F3" class=tdc><%= (Recordset1.
Fields.Item("url").Value) %></td>
</tr>
<tr>
<td width="100%" colspan="2" align="center" background="images/b2.gif" height=
"22" class=tdc1>搜索结果:</td>
</tr>
</table>
</center>
</div>
<%
End if
Recordset1.Close()
Set Recordset1 = Nothing
%>
<%
else
a="错误:请确保你输入的是大于7位的数字! <br><a href='search_tel.asp'>返回</a>重新输入"
end if
end if
'显示错误信息
%>
<div align="center" class="tdc"><%=a%></div>
<%
end if
%>

```

12.5.2 查询 IP 地址信息

IP 地址信息的查询相对于手机号码查询要复杂些,因为它涉及字符串的处理,具体操作步骤如下。

01 建立一个连接文件 conn.asp,输入下面代码,定义一个数据库连接,并自定义一个函数过滤非法字符。

```

<%
dim connstr,conn
const connpath="db.mdb" 'ip数据库路径
connstr="DBQ="+server.mappath("&connpath&")+";DefaultDir=;DRIVER={Microsoft
Access Driver (*.mdb)};"
set conn=server.createobject("ADODB.CONNECTION")
conn.open connstr
'过滤非法字符
function chktopic(str)
if str<>"" then
chktopic=replace(str,"'","''")
end if
end function
pages=20%>

```

02 打开 search_result.asp 文件,切换到【代码】视图,在代码顶部输入下面代码,实现导入 conn.asp 连接文件。

```

<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="conn.asp" -->
<!--#include file="css.css" -->

```

```
<!--#include file="Connections/conn.asp" -->
```

- 03** 输入下面代码，获取用户输入的 IP 地址，然后利用字符串函数 Split() 拆分该 IP 地址，用 Ubound() 判断该 IP 地址是否是 4 组，再用 len() 函数判断 IP 字符串中每组是否是 3 位，不是的补足 0 使其符合要求，为后面的比较提供方便。

```
<%
if chktopic(request.form("search_ip"))<>" then
  fip=chktopic(request.form("ip"))
  c=fip
  '拆分ip，目的是为了不足3位的补0，同时也判断ip地址是否合法
  fip = Split(fip, ".")
  if Ubound(fip)=3 then
    For i=0 To Ubound(fip)
      if len(fip(i))=2 then
        fip(i)="0"&fip(i)
      elseif len(fip(i))=1 then
        fip(i)="00"&fip(i)
      end if
    next
    oip=fip(0)& "." & fip(1) & "." & fip(2) & "." & fip(3)
```

- 04** 输入下面代码重新组合 IP，再与数据库的 IP 地址进行比较，以获取地址。

```
sql1="select * from ip where startip<="&oip&" and endip>="&oip&" and
left(startip,3)= "&fip(0)&" and left(endip,3)="&fip(0)&" and
mid(startip,5,3)="&fip(1)&" and "&fip(1)&"=mid(endip,5,3) and
mid(startip,9,3)="&fip(2)&" and "&fip(2)&"=mid(endip,9,3) "
set rs1=conn.execute(sql1)
if not rs1.eof then
  a=rs1("country")
  b=rs1("local")
  set rs1=nothing
else
  sql="select * from ip where startip<="&oip&" and endip>="&oip&" and
left(startip,3)="&fip(0)&" and left(endip,3)="&fip(0)&" and
mid(startip,5,3)="&fip(1)&" and "&fip(1)&"=mid(endip,5,3) "
set rs=conn.execute(sql)
if not rs.eof then
  a=rs("country")
  b=rs("local")
  set rs=nothing
else
  sql2="select * from ip where startip<="&oip&" and endip>="&oip&" and left(startip,3)
=&fip(0)&" and left(endip,3)="&fip(0)&" "
set rs2=conn.execute(sql2)
if not rs2.eof then
  a=rs2("country")
  b=rs2("local")
  set rs2=nothing
else
  sql3="select * from ip where startip<="&oip&" and endip>="&oip&" "
set rs3=conn.execute(sql3)
if not rs3.eof then
  a=rs3("country")
  b=rs3("local")
  set rs3=nothing
else
  a="对不起，找不到地址！"
end if
end if
end if
```

```

end if
else
a="错误:您输入的ip地址非法,请重新输入!"
end if
%>

```

在上面代码段中利用判断结构分步比较,目的是为了在当前两个 IP 段相同时,得到的地址比较具体。

05 把变量 a 、 b 输出到页面中,如图 12.15 所示。

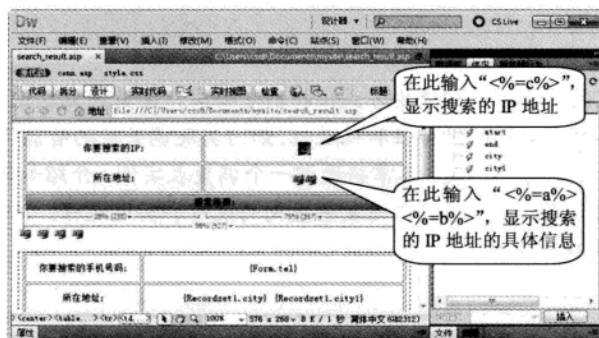


图 12.15 绑定变量

12.6 实践总结

本章详细介绍了 SQL 查询的一些高级使用技巧。实际上信息查询技术运用比较宽泛,大到专门提供搜索服务的大型网站,小至站点内部信息的查询,几乎每个网站都提供了不同的查询服务。能够设计一个高级查询功能对于提高网站服务水平是非常重要的。

查询的核心技术就是 SQL 语句的灵活运用。SQL 语言不仅查询数据,还可以建立数据表,操作数据表,建立视图和过程,以及处理复杂数据。如果想提高数据查询技术,建议参阅 SQL 方面的书籍,并在实践不断运用。

第 13 章

Dreamweaver

网站调查模块

调查表是网上信息交互的一种重要方式，特别受公司、企业网站欢迎。与留言板相比，调查表所获取的信息具有更强的针对性和权威性。如何实现调查表的智能化管理和灵活显示，是一般网站开发者必须思考的问题。本章将通过一个调查表实例，介绍如何利用 SQL 查询语句动态控制数据的表格化显示，同时讲述复杂查询的设计和实现过程。另外，还将介绍如何实现动态 CSS 样式表技术。

技术要点

- 条件查询
- 动态表单属性
- 重复区域横向显示
- 重复区域换行显示
- 条件显示
- 动态样式表
- 数据更新
- 动态信息提示

13.1 程序设计分析

调查表与投票管理系统关系比较紧密。它们由两个功能模块组成：提供输入调查信息的生成模块和显示调查信息与结果的显示模块。调查表不仅要统计网上调查的结果，而且还要把每一位参加调查的个人信息记录下来。所以在调查系统中对数据库的操作就是插入记录。

本例主要介绍调查表的前台技术，即研究如何在网上动态、个性地显示调查信息，并能根据用户的不同需求生成不同的调查表，效果如图 13.1 所示，而投票系统则重在探索后台管理以及数据

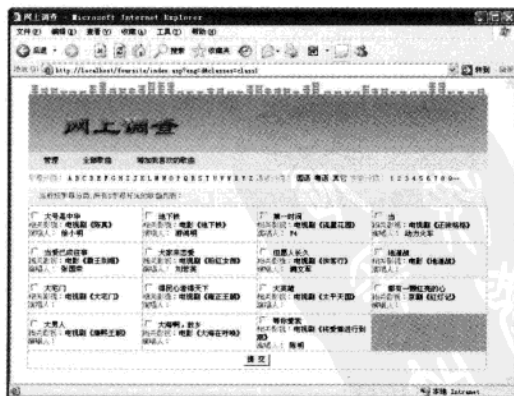


图 13.1 网上调查浏览效果

的统计分析技术的实现。

使用调查表之前应先选择调查的内容,由服务器根据用户的选择生成不同的调查表,如果用户想要增加调查信息,则需要单击“增加”链接进入另一个页面,提交个人喜爱的信息,当然如果进入管理页面,则会有更多的操作选择。因此,本例流程图如图 13.2 所示。

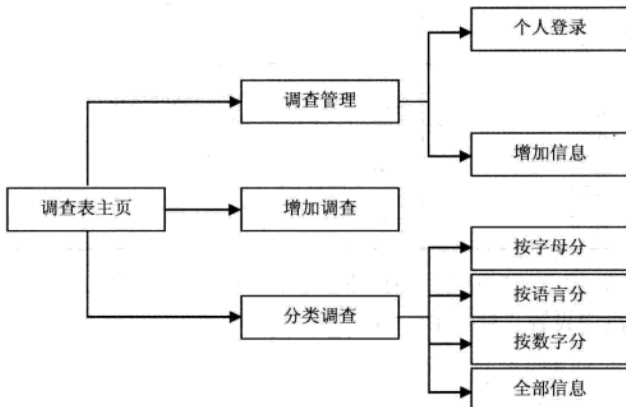


图 13.2 调查表流程图

本实例主要涉及两个页面设计,但要实现调查表功能,应需要解决下面技术问题。

- 如何实现记录集横向重复显示,需要手动输入两句代码。
- 如何动态控制单元格背景色的显示,这需要使用动态样式表。
- 根据用户的要求快速定义不同的记录集,实现一页内多条记录集的显示。
- 如何分类显示不同条件的记录集。
- 如何实现动态提示信息。



提示

学习本实例之前,读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录,如mysite。启动 Dreamweaver CS5 创建一个动态站点,远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”,【访问】应为“本地/网络”。

把本书光盘中的“13”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面内容、数据库结构,并帮助读者建立了数据库连接。

13.2 数据库设计

本例数据库 mdb.mdb 只有 musiclist 一个表。musiclist 表用来记录调查表要显示的所有信息和用户个人信息,数据结构设计见表 13.1。

表 13.1 musiclist 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	音乐名称
music	文本	50	否	信息发布者
ycr	文本	50	否	演唱人
tv	文本	50	否	电视名

续表

字 段	类 型	字段大小	必填字段	说 明
eng	文本	50	否	字母分类
guoyue	数字	长整型	否	国语粤语分类
zs	数字	长整型	否	字数分类
user	文本	20	否	添加人
tel	文本	20	否	电话
mail	文本	50	否	邮箱
add	文本	200	否	地址
htc	数字	长整型	否	票数 1
thtc	数字	长整型	否	票数 2
tdate	日期/时间	-	-	增加时间
key	是/否	-	-	是否作为投票项显示

在 tdate 字段中, 设置“默认值”为“Now()”, 如图 13.3 所示, 使发布信息自动增加现在时间。在 key 字段中设置“默认值”为“True”。

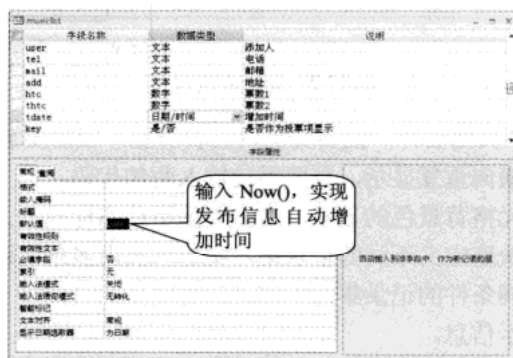


图 13.3 设置 tdate 字段为自动增加时间

musiclist 表的字段可以分为 4 大类, 即调查显示信息, 包括 music、music、ycr 和 tv 共 4 个字段; 调查分类根据, 包括 eng、guoyue 和 zs 共 3 个字段; 增加信息人的个人信息, 包括 user、tel、mail、add、tdate 和 key 共 6 个字段; 调查投票结果, 包括 htc 和 thtc 共 2 个字段。

13.3 页面设计

本实例共有 8 个文件, 具体说明见表 13.2。

表 13.2 系统文件列表

文件名	说 明
index.asp	调查表页面
musicadd.asp	增加歌曲信息页面
save.asp	保存歌曲信息文件
Musicadd1.asp	增加歌曲信息页面
save1.asp	保存歌曲信息文件
admin.asp	用户登录与管理信息页面
adminsava.asp	保存用户信息文件
Conn.asp	数据库连接文件

下面就几个主要页面设计进行介绍。

13.3.1 调查表页面

调查表页面主要考虑调查项目布局问题,使其在页面上分布更趋合理、更适合用户阅读。本例调查表的记录项要显示的信息不多,而记录却非常多,因此不适合单列显示,而是需要一行显示多条记录,因此 index.asp 页面设计如图 13.4 所示。

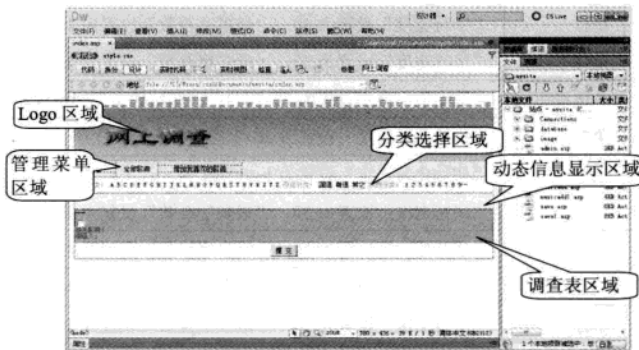


图 13.4 index.asp 页面设计效果

在该页面中,信息显示表格中只有一个单元格,以便使用动态服务器制作重复显示区域。本页面使用 2 个表单对象,属性设置见表 13.3。

表 13.3 调查表表单设置列表

对象	名称	类型	设置说明
表单	form1	表单	“动作”设置为“save.asp?action=htc”;“方法”设置为“POST”
选择框	id	复选框	“选定值”先空着,等建立记录集后插入动态数据,“初始状态”为“未选中”
按钮	Submit	提交	“值”为“提交”,“动作”为“提交表单”,设置“类”为“input1”

13.3.2 增加信息页面

musicadd.asp 页面设计效果如图 13.5 所示。



图 13.5 musicadd.asp 网页设计效果

musicadd.asp 文件用来允许用户向数据库增加信息,以收集用户关于某个主题的想法,因此涉及表单的制作,表单对象属性设置见表 13.4。

表 13.4 增加信息表单设置列表

对象	名称	类型	设置说明
表单	form1	表单	“动作”设置为“save.asp?action=add”;“方法”设置为“POST”
文本框	music	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
文本框	tv	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
文本框	ycr	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
选择框	guoyue	单选按钮	“选定值”分别设置为 1、2、3;“初始状态”都设置为“未选中”;
列表/菜单	eng	菜单	“初始化时选定”设置为 26 个英文字母
列表/菜单	zs	菜单	“初始化时选定”设置为 1~9 个数字
文本框	user	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
文本框	tel	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
文本框	mail	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
文本框	add	单行	“字符宽度”设置为 30;“最多字符数”设置为 45
按钮	Submit	提交	“值”为“提交”,“动作”为“提交表单”,设置“类”为“input1”

musicadd1.asp 文件也是一个表单页面,用来允许管理人员输入信息,然后增加到数据库中,设计如图 13.6 所示。

由于是专门为管理人员设计的表单,因此就不再设计个人信息输入表单区域。具体制作和设置可以参照表 13.4 所示。

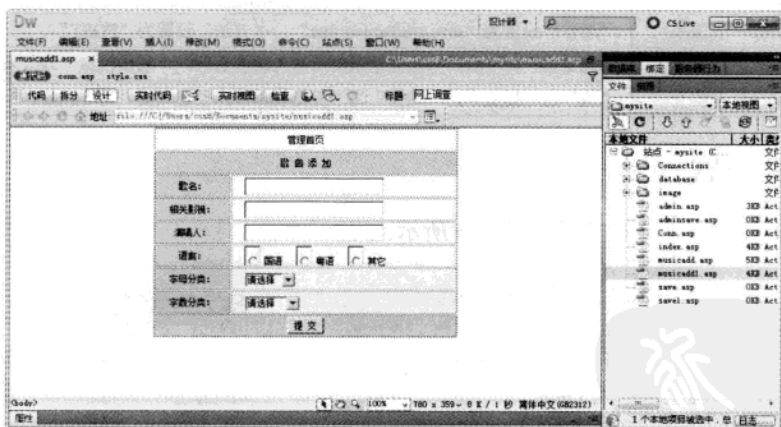


图 13.6 musicadd1.asp 页面设计效果

13.3.3 管理页面

admin.asp 页面先判断用户是否登录,然后判断数据库中是否已经存在该记录,如果存在则显示该记录,反之则增加新记录,并把这些记录写入数据库,其页面设计如图 13.7 所示。其中表单属性设置见表 13.5。

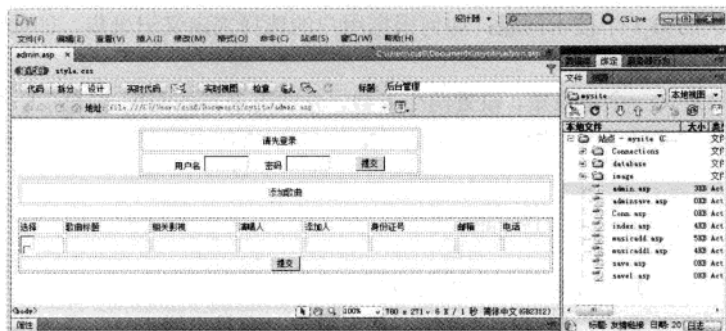


图 13.7 admin.asp 文件设计效果

表 13.5 管理页面表单设置列表

对 象	名 称	类 型	设 置 说 明
表单	form1	表单	“动作”设置为“adminsave.asp?action=login”；“方法”设置为“POST”
文本框	1	单行	“字符宽度”设置为 10
文本框	2	单行	“字符宽度”设置为 10
按钮	Submit	提交	“值”为“提交”，“动作”为“提交表单”，设置“类”为“input1”
表单	form1	表单	“动作”设置为“adminsave.asp?action=htc”；“方法”设置为“POST”
选择框	id	复选框	设置“初始状态”为“未选中”
按钮	Submit	提交	“值”为“提交”，“动作”为“提交表单”，设置“类”为“input1”

注意，本页面中设计了两个名称相同的表单对象，这是因为后面还将在该页面中插入判断条件，进行条件显示，实现一页多种显示功能的目的。

同时要设置“增加歌曲”链接文本的链接地址为“musicadd1.asp”。

13

13.4 前台功能实现

该调查表将以表格化形式显示，即显示为多行和多列，因此需要两个条件语句控制数据显示的列数和行数，同时用脚本动态控制 CSS 样式的显示效果，详细操作如下。

13.4.1 多条件查询

本实例最大技术特点就是允许网民自由选择不同的调查分类形式，以及在大类中选择不同的小类分别生成一个调查表，然后在这个表上进行投票，如图 13.8 所示。

先选择大类别，再选择小类别，最后在生成的表中投票

管理 全部歌曲 增加我喜欢的歌曲

字母分类: ABCDEFGHIJKLMNOPQRSTUVWXYZ 语言分类: 国语 粤语 其它 字数分类: 1 2 3 4 5 6 7 8 9...

当前按数字分类, 所有1个字符的歌曲列表

<input type="checkbox"/> 当 相关影视: 电视剧《王保长》 演唱人: 动力火车	<input type="checkbox"/> 飘 相关影视: 电视剧《非常公民》 演唱人: 沙宝亮	<input type="checkbox"/> 谁 相关影视: 电视剧《将爱情进行到底》 演唱人: 谢雨欣
---	---	--

提交

图 13.8 多级查询

现在要解决的第 1 个技术问题：如何根据用户的选择来定义记录集，即如何实现多条件查询？

方法一：只定义一个记录集，设置两个参数，分别用来判断网民选择的大类别和小类别，再根据参数值来定义记录集。也就是说在记录集的 SQL 语句中需要增加两个变量，代码如下：

```
SELECT id, music, ycr, tv, htc, thtc
FROM musiclist
WHERE MMColParam1 = MMColParam2
ORDER BY id DESC
```

设置的【记录集】高级对话框如图 13.9 所示。

此方法比较高效简洁，但运行时系统不是太稳定，容易出错。

方法二：根据不同的大类，分别定义 4 个记录集，再用一个变量来判断不同的小类别，然后生成记录集，这种方法稳定性比较好，本例选择了此方法。

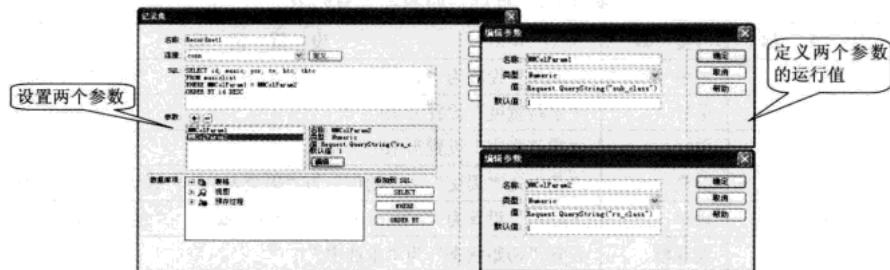


图 13.9 设置多条件查询

方法三：根据不同的小类别定义所有的记录集，然后判断用户选择的小类别，调用不同的记录集。这种方法比较原始，工作量大，一般不建议使用。

当然，在 ASP 代码环境中还可以编写更高效、更简洁且稳定的设计方法，但需要读者熟悉 ASP 编程，不能使用 Dreamweaver CS5 可视化操作，因此本例就不再涉及。

下面以方法二为例进行介绍，具体步骤如下。

- 01 在 Dreamweaver CS5 中，打开 index.asp 文件。选择【窗口】|【绑定】菜单命令，打开【绑定】面板。
- 02 单击【绑定】面板左上角的 按钮，在弹出的下拉菜单中选择【记录集（查询）】命令，在打开的【记录集】对话框中设置要查询的信息，如图 13.10 所示。



图 13.10 定义记录集

- 03 在【记录集】对话框中单击【高级】按钮，切换到【记录集】对话框的高级状态，改变变量的默认值为“a”，如图 13.11 所示。为后面设置重复显示区域提供一个条件，在接下来的小节中会详细介绍。

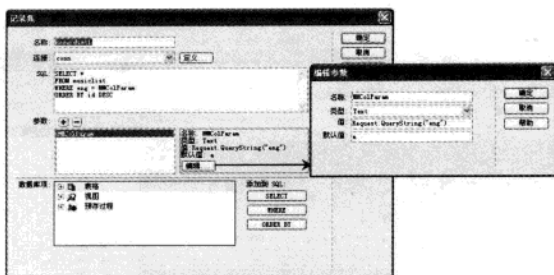


图 13.11 高级定义记录集

04 以同样的方式再定义 3 个记录集，分别设置不同的大类筛选条件，设置如图 13.12 所示。

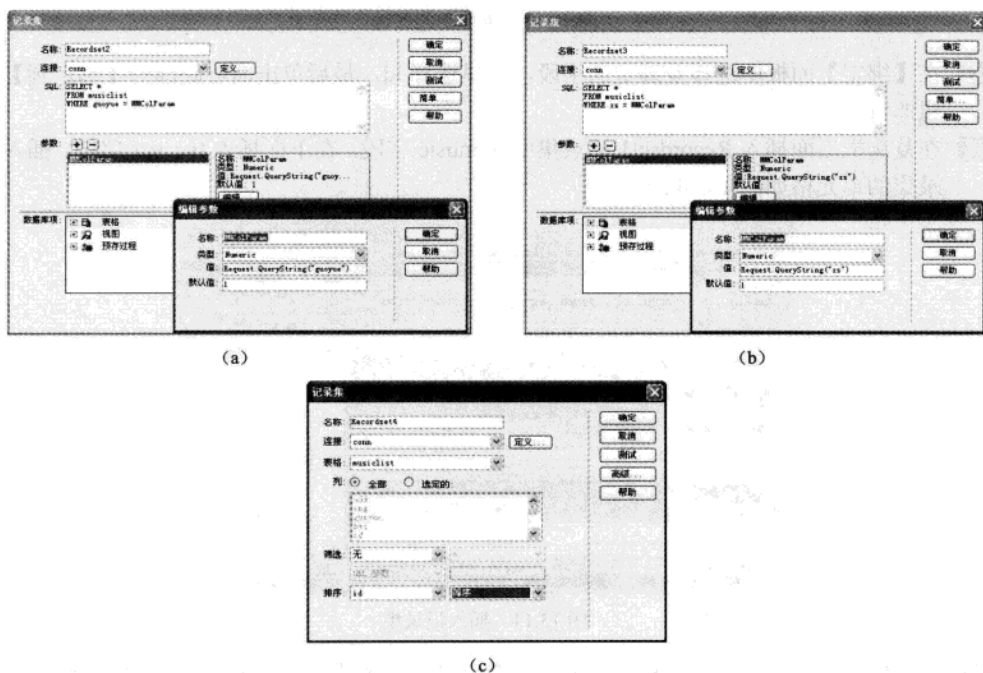


图 13.12 定义其他记录集

(a) 以语言类为筛选条件，并设置默认值为 222；(b) 以数字类为筛选条件，并设置默认值为 333；
(c) 查询所有记录

13.4.2 记录集横向重复显示

利用 Dreamweaver CS5 提供的重复区域服务器行为可以实现在页面内显示多条记录，但在前面的实例中，重复区域为单列垂直显示。下面介绍如何使多条记录多列沿横向显示，具体操作步骤如下。

01 在 Dreamweaver CS5 中打开 index.asp 文件。选择【窗口】|【绑定】菜单命令，打开【绑定】面板，把 Recordset1 记录集中的 id 字段绑定到复选框的 value 值上。方法是选中复选框，在【绑定】面板底部的【绑定到】下拉列表中选择“input.value”选项，如图 13.13 所示。

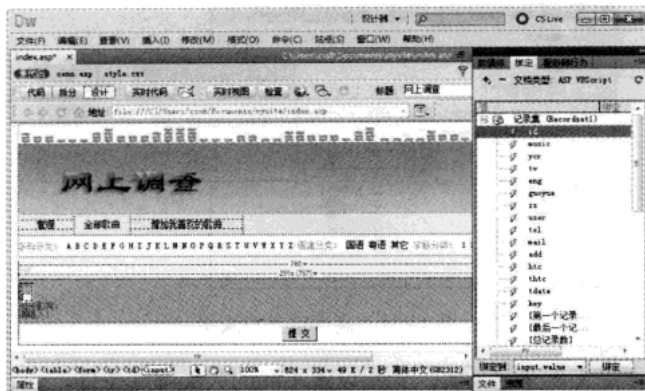


图 13.13 绑定动态属性

- 02** 在【绑定】面板中选择要绑定的字段，这里选择 id，最后单击面板底部的【绑定到】按钮即可。
- 03** 在复选框后面插入 Recordset1 记录集中的 music 字段，在下面插入 tv、ycr 字段，插入记录集的单元格如图 13.14 所示。

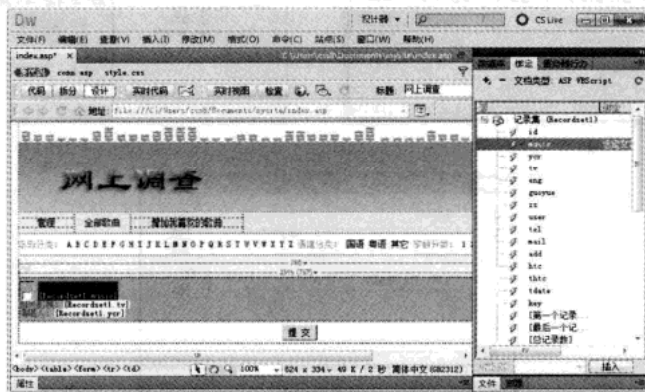


图 13.14 插入记录集

- 04** 定义横向重复显示记录。首先在页面中选中调查表区域的单元格，如图 13.15 所示。

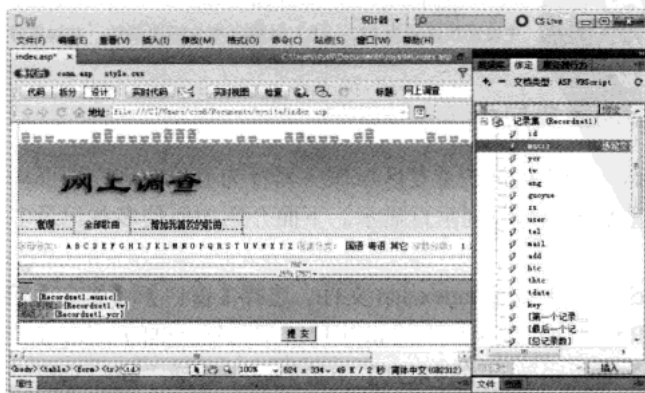


图 13.15 选中单元格

注意不要选中行，如果选中行（即<tr>标签）就会使记录纵向重复显示。

05 在【服务器行为】面板中单击左上角的 \oplus 按钮，在弹出的下拉菜单中选择【重复区域】命令，打开【重复区域】对话框，选择 Recordset1 记录集，设置每页显示 4 条记录，如图 13.16 所示。

06 单击【确定】按钮，关闭对话框。按 F12 键浏览，效果如图 13.17 所示。

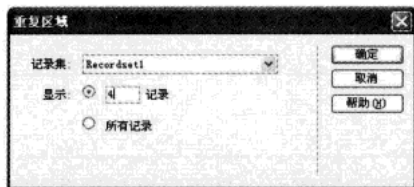


图 13.16 设置重复区域

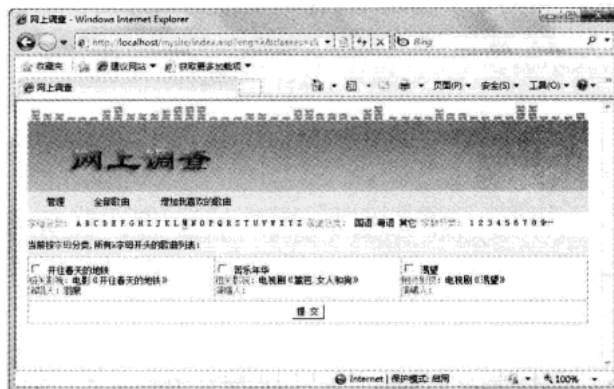


图 13.17 设置记录横向显示效果

13.4.3 记录集横向换行显示

在上一节实现了记录集横向显示的初步功能，但这不符合实际需要，因为记录集往往不是几条记录，而是成百上千条记录。如果让它们在一行内显示，记录会紧挤在一起，将无法观看。下面介绍如何用几行代码实现记录集横向换行显示，具体操作步骤如下。

01 打开 index.asp 文件，切换到【代码】视图，找到实现重复区域代码段，显示如下：

```
<%
While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF))
%>
<td width=25% height=60;><input type="checkbox" name="id" style="border:1px ">
<%= (Recordset1.Fields.Item("music").Value) %><br>
<font color=#666666>相关影视:</font><%= (Recordset1.Fields.Item("tv").Value) %><br>
<font color=#666666>演唱人:</font><%= (Recordset1.Fields.Item("ycr").Value) %> </td>
<%
Repeat1__index=Repeat1__index+1
Repeat1__numRows=Repeat1__numRows-1
Recordset1.MoveNext()
Wend
%>
```

仔细观察上面代码段，会发现使记录重复显示的代码就那么 5 行，如上面代码段中方框区域中的代码。

第 1 行代码是一个循环结构的语句，当变量 Repeat1__numRows（该变量用来保存用户设置每页要显示记录的条数）减 1 后不等于 0，同时记录集还没有到末尾，则系统会重复显示。

02 在上面代码中再加一个循环判断，来实现当一行 4 条显示完毕后自动换行显示，代码如下：

```
<%
While NOT Recordset1.EOF
Repeat1__index = 4
%>
```

```

<tr>
<%
While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF))
%>
    <td width=25% height=60;><input type="checkbox" name="id" style="border:1px ">
    <%= (Recordset1.Fields.Item("music").Value) %><br>
    <font color=#666666>相关影视: </font><%= (Recordset1.Fields.Item("tv").
    Value) %><br>
    <font color=#666666>演唱人: </font><%= (Recordset1.Fields.Item("ycr").
    Value) %> </td>

    <%
    Repeat1__index=Repeat1__index+1
    Repeat1__numRows=Repeat1__numRows-1
    Recordset1.MoveNext()
    Wend
    %>
    <input type="hidden" name="htc" id="htc" >
    <input type="hidden" name="thtc" id="thtc" >
</tr>

<%
Wend
%>

```

在上面代码段中，在<tr>标签外围增加了一个循环语句。把前面定义重复区域时生成的 While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF)) 语句复制到<tr>前边，并修改为 While NOT Recordset1.EOF。这样两个循环结构嵌套在一起，当自定义变量 Repeat1__numRows 不等于 0 时，执行内部循环命令，即横向显示 4 条记录，如果变量 Repeat1__numRows 小于或等于 0，则跳出内部循环，执行外部循环。外部循环把<tr>和</tr>行标签包括在内，这样就可以实现换行的目的。在外部循环开始定义变量 Repeat1__numRows 初始值为 4，然后在内部循环内设置每循环一次减 1，这样就实现了记录集横向换行的目的。

03 按 F12 键预览该页面，显示效果如图 13.18 所示。



图 13.18 设置记录横向换行显示效果

13.4.4 记录集条件显示

下面将解决另一个技术难题：如何根据用户的要求条件显示不同记录集结果，即根据不同的选择生成不同的调查表，具体操作步骤如下。

01 应定义好分类栏目内各个链接要传递的参数，设置代码如下：

```
<tr bgcolor="#FFFFCC">
  <td height=30><span class="STYLE1">字母分类: </span>
  <a href=index.asp?eng=a&classes=class1>A</a>
  <a href=index.asp?eng=b&classes=class1>B</a>
  <a href=index.asp?eng=c&classes=class1>C</a>
  <a href=index.asp?eng=d&classes=class1>D</a>
  <a href=index.asp?eng=e&classes=class1>E</a>
  <a href=index.asp?eng=f&classes=class1>F</a>
  <a href=index.asp?eng=g&classes=class1>G</a>
  <a href=index.asp?eng=h&classes=class1>H</a>
  <a href=index.asp?eng=i&classes=class1>I</a>
  <a href=index.asp?eng=j&classes=class1>J</a>
  <a href=index.asp?eng=k&classes=class1>K</a>
  <a href=index.asp?eng=l&classes=class1>L</a>
  <a href=index.asp?eng=m&classes=class1>M</a>
  <a href=index.asp?eng=n&classes=class1>N</a>
  <a href=index.asp?eng=o&classes=class1>O</a>
  <a href=index.asp?eng=p&classes=class1>P</a>
  <a href=index.asp?eng=q&classes=class1>Q</a>
  <a href=index.asp?eng=r&classes=class1>R</a>
  <a href=index.asp?eng=s&classes=class1>S</a>
  <a href=index.asp?eng=t&classes=class1>T</a>
  <a href=index.asp?eng=u&classes=class1>U</a>
  <a href=index.asp?eng=v&classes=class1>V</a>
  <a href=index.asp?eng=w&classes=class1>W</a>
  <a href=index.asp?eng=x&classes=class1>X</a>
  <a href=index.asp?eng=y&classes=class1>Y</a>
  <a href=index.asp?eng=z&classes=class1>Z</a>
  <span class="STYLE1">语言分类: </span>
  <a href=index.asp?guoyue=1&classes=class2>国语</a>
  <a href=index.asp?guoyue=2&classes=class2>粤语</a>
  <a href=index.asp?guoyue=3&classes=class2>其他</a>
  <span class="STYLE1">字数分类: </span>
  <a href=index.asp?zs=1&classes=class3>1</a>
  <a href=index.asp?zs=2&classes=class3>2</a>
  <a href=index.asp?zs=3&classes=class3>3</a>
  <a href=index.asp?zs=4&classes=class3>4</a>
  <a href=index.asp?zs=5&classes=class3>5</a>
  <a href=index.asp?zs=6&classes=class3>6</a>
  <a href=index.asp?zs=7&classes=class3>7</a>
  <a href=index.asp?zs=8&classes=class3>8</a>
  <a href=index.asp?zs=9&classes=class3>9...</a></td>
</tr>
```

02 在管理栏目中设置“全部”超级链接为“全部歌曲”。

03 在【代码】视图中找到重复区域代码段，然后在最前面输入下面代码：

```
<%
if Request.QueryString("classes") = "class1" then
%>
```

该行代码的作用是用 Request.QueryString 获取查询字符串传递过来的参数 classes 的值，即判断是否是按字母分类进行查询，如果是则执行该段代码，否则跳到下面继续执行。

04 在代码段的最后一行加上<%end if%>，就会形成一个条件结构，在该结构内嵌套了两个循环结构的代码段，代码如下：

```
<%
if Request.QueryString("classes") = "class1" then
```

```

%>
<%
While (NOT Recordset1.EOF)
Repeat1__numRows = 4

%>
<tr>
<%
While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF))
%>
  |
```

05 把上面的代码段全部复制，并在下面连续粘贴3次，这样在【代码】视图中就会有4个相同的代码段，然后选中第2个代码段，把其中的 Recordset1 替换为 Recordset2，把条件中的 Request.QueryString("classes")="class1" 替换为 Request.QueryString("classes")="class2"。

06 以同样的方式选中替换第3个代码段和第4个代码段。注意第4个代码段中要把 Request.QueryString("classes")="class1" 替换为 Request.QueryString("classes")="class4"。

至此就完成了本例的主要功能，即调查表的分类显示，设计页面如图 13.19 所示。

13.4.5 动态样式表

在网上会经常看见网页换肤技术，用户可以选择个人喜爱的页面风格。其实质就是动态样式表的运用，下面结合本例介绍这种技术实现的方法。

从上面的图例中可以看到，调查表的背景颜色单调而没有个性。下面使用动态样式表让调查表每列背景色隔列交替变化显示，如图 13.20 所示。

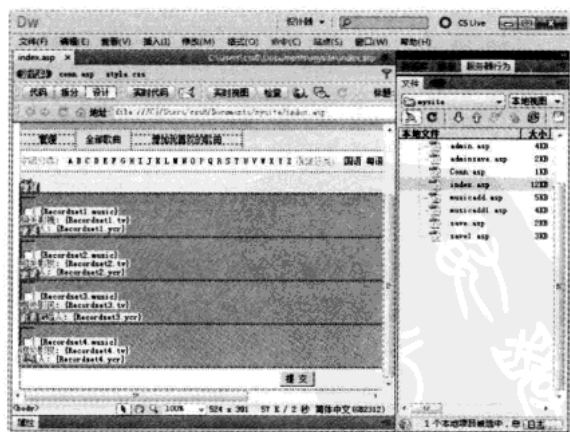


图 13.19 记录集按条件显示

「大号是中华 相关影视：电视剧《陈真》 演唱人：徐小明	「地下铁 相关影视：电影《地下铁》 演唱人：游鸿明	「第一时间 相关影视：电视剧《流星花园》 演唱人：F4	「当 相关影视：电视剧《还珠格格》 演唱人：动力火车
「当爱已成往事 相关影视：电影《霸王别姬》 演唱人：张国荣	「大家来恋爱 相关影视：电视剧《粉红女郎》 演唱人：刘若英	「但愿人长久 相关影视：电视剧《侠客行》 演唱人：满文军	「地道战 相关影视：电影《地道战》 演唱人：

图 13.20 设置交替背景颜色

要实现上图效果，就需要定义两个样式表，分别设置不同的背景颜色，代码如下：

```
td.bg1 {
    background:#ffffff;
}
td.bg2 {
    background:#FFF5EC;
}
```

把上面代码放在 style.css 文件中（位于 images 文件夹中），回到 index.asp 代码视图，设置单元格背景色为：

```
<td width=25% height=60 class="<%= "bg" & n% ">">
```

其中 bg 表示背景前缀，n 为变量，设置 n 的值为 1 或 2，并让它们交替变化，即可实现图 3.20 所示的动态效果。

切换到【代码】视图，找到第 1 个条件结构代码段，修改后的代码如下：

```
<% '按字母分类进行显示
if Request.QueryString("classes") = "class1" then
%>
<%
While (NOT Recordset1.EOF)
Repeat1__numRows = 4
n = 1
%>
<tr>
<%
While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF))
%>
<td width=25% height=60 class="<%= "bg" & n% ">"><input type="checkbox" name="id"
style="border:1px ">
<%= (Recordset1.Fields.Item("music").Value) %><br>
<font color=#666666>相关影视:
</font><%= (Recordset1.Fields.Item("tv").Value) %><br>
<font color=#666666>演唱人: </font><%= (Recordset1.Fields.Item("ycr").Value) %></td>
<%
n = n+1
if n > 2 then n = 1
%>
<%
Repeat1__index=Repeat1__index+1
Repeat1__numRows=Repeat1__numRows-1
Recordset1.MoveNext()
Wend
%>
<input type="hidden" name="htc" id="htc" >
<input type="hidden" name="thtc" id="thtc" >
</tr>
<%
Wend
%>
<%
end if
%>
```

先在外部循环体内开始区域定义一个变量 n 为 1, 然后在 `<td>` 标签内插入样式表属性, 设置 `class` 类的值为一个动态值。

再在内部循环结构的尾部增加 3 行代码, 实现变量 n 的自动递增; 然后用一个条件结构判断, 如果变量 n 大于 2, 则使该变量的值又恢复到 1, 如此循环即可实现 n 的值交替变化, 也就实现了样式表的交替变化。在 IE 中运行网页, 效果如图 13.21 所示。



图 13.21 动态样式表的设计效果

13.4.6 动态信息提示

完成上面的工作后, 读者基本上可以使用 `index.asp` 文件动态生成不同调查内容的调查表了。但在使用时会存在这样一个问题: 当单击某个选项生成一个调查表后, 有时会忘记当前调查表是根据什么生成的。为了使程序更人性化, 下面增加一行动态信息提示, 具体操作步骤如下。

要实现动态信息提示功能, 必须捕获用户选择的大类和小类名称, 然后通过 ASP 输出代码输出信息。当用户单击某个分类选项的文字链接时, 会在 URL 后面的查询字符串中传递 2 个参数 (可参见第 13.4.4 小节介绍):

`eng=u&classes=class1`

第 1 个参数传递小类的参数值, 第 2 个参数是大类参数值。简单分析会发现, 第 1 个参数的变量名是不同的, 分别为 `eng`、`guoyue`、`zs`。这就为程序设计带来难度, 不过可以使用 `Request.QueryString` 数据集合的索引值来获取第 1 个参数的变量值, 如 `Request.QueryString(1)` 就可以获取第 1 个参数的变量值, 而不管其变量名是什么。



提示

与 Form 数据集合一样, 在 `QueryString` 数据集合中, 可以利用参数的名称, 也可以用索引值取得参数的值。一般当一个参数的名称不是唯一的时候, 与 Form 数据集合一样处理: 先取得名称相同的参数的总数, 然后再加上一个索引值就可以分别访问各个参数的值了。

取得相同参数的总数的语法如下:

名称相同的参数的总数=`Request.QueryString("参数名称").Count`

名称相同的参数的总数=`Request.QueryString(索引值).Count`

取得名称相同的内容值语法如下:

名称相同的参数的内容值=`Request.QueryString(索引值)`。

在代码中找到动态信息栏，插入下面代码：

```
<%
'先定义变量a的默认值，定义变量subclass和classes获取参数值
a = "当前<font color=red>没有</font>列表！"
classes = Request.QueryString("classes")
subclass = Request.QueryString("eng")
subclass = Request.QueryString("guoyue")
subclass = Request.QueryString("zs")
'如果选择字母分类显示歌曲
if classes="class1" then
    if Not Recordset1.EOF then '判断记录集是否为空，为空则不执行下面语句
        a = "当前按<font color=red>字母</font>分类，所有<font color=red>" & subclass &
"</font>字母开头的歌曲列表："
    end if
end if
'如果选择语言分类显示歌曲
if classes="class2" then
    if Not Recordset2.EOF then '判断记录集是否为空，为空则不执行下面语句
        '判断选择哪个小类，再赋予不同的显示信息
        if subclass=1 then
            a = "当前按<font color=red>语言</font>分类，所有<font color=red>国语</font>歌曲列表："
        elseif subclass=2 then
            a = "当前按<font color=red>语言</font>分类，所有<font color=red>粤语</font>歌曲列表："
        elseif subclass=3 then
            a = "当前按<font color=red>语言</font>分类，所有<font color=red>其他语言</font>歌曲列表："
        end if
    end if
end if
'如果选择数字分类显示歌曲
if classes="class3" then
    if Not Recordset3.EOF then '判断记录集是否为空，为空则不执行下面语句
        a = "当前按<font color=red>数字</font>分类，所有<font color=red>" & subclass
        &"</font>个字标题的歌曲列表："
    end if
end if
'如果选择“全部”歌曲
if classes="class4" then
    if Not Recordset4.EOF then '判断记录集是否为空，为空则不执行下面语句
        a = "下面列出所有歌曲："
    end if
end if
Response.Write a
%>
```

在上面的代码段中，先定义3个自变量，变量a为要显示的动态信息字符串；变量subclass为用来获取查询字符串中的小类参数值；变量classes用来获取大类参数值。

先利用条件语句判断变量classes的类别值，然后分别进行处理。再根据变量subclass判断所单击的选项是哪个选项。最后根据上面的条件判断，分别赋予变量a不同的显示信息。

13.5 后台管理

在调查的过程中，可能存在用户喜欢的歌曲，但在后台数据库中却没有。为实现动态交互性，丰富数据库中的歌曲，本实例提供给用户增加歌曲的机会，然后经过网站管理员的审核后再显示出来。

13.5.1 后台控制台设计

在 admin.asp 文件中, 先判断用户是否登录, 如果没有登录, 则显示登录表单, 让用户进行登录。如果已经登录, 则判断在数据库中是否已经存在管理人员增加的记录: 如果存在则显示已输入记录, 并要求选择提交; 如果没有记录, 则显示一个增加记录的超级链接, 让管理人员进入增加记录页面进行操作。

实现本页面功能的操作步骤如下。

- 01** 定义记录集。在 Dreamweaver CS5 中, 打开 admin.asp 页面, 在【数据库】面板中要设置数据源, 如果在该面板中没有任何数据源显示, 需要先定义站点数据源。

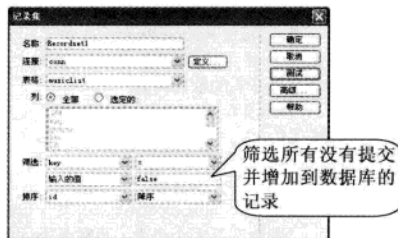


图 13.22 定义记录集

- 02** 在菜单中选择【窗口】|【绑定】命令, 打开【绑定】面板。单击面板左上角的 按钮, 在弹出的下拉菜单中选择【记录集 (查询)】命令, 设置【记录集】对话框, 如图 13.22 所示。

- 03** 然后把记录集内的字段插入到相应的表格单元格中, 如图 13.23 所示。注意, 要把 id 字段绑定到复选框的 value 属性上, 代码如下:

```
value="<%= (Recordset1.Fields.Item("id").Value) %>"
```

- 04** 定义记录集显示的重复区域。选中绑定字段所在的行 (包含<tr>和</tr>标签)。选择【窗口】|【服务器行为】菜单命令, 打开【服务器行为】面板。单击【服务器行为】面板左上角的 按钮, 在弹出的下拉菜单中选择【重复区域】命令, 打开【重复区域】对话框。在【记录集】下拉列表中选择所要显示的记录集, 如 Recordset1; 在【显示】区域设置“所有记录”, 如图 13.24 所示。



图 13.23 插入记录集

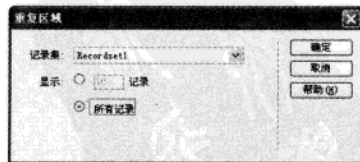


图 13.24 设置【重复区域】对话框

- 05** 单击【代码】视图按钮, 切换到代码视图, 在<body>标签后面输入下面代码:

```
<% '判断是否登录, 否则要求先登录
if (Session("admin")="") then
%>
```

06 在第一个表格的后面输入下面代码:

```
<% '如果登录,则可以执行下面操作
else
%>
```

07 在第二个表格的后面输入下面代码:

```
<% '判断是否存在记录,如果存在记录则循环显示
if not Recordset1.EOF then
%>
```

08 在</body>标签前面输入下面代码,此时的设计视图如图 13.25 所示:

```
<%
else
Response.Write("<h3>当前还没有添加记录! </h3>")
end if
end if
%>
```

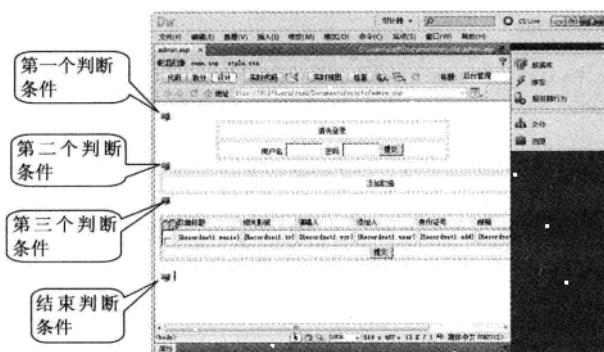


图 13.25 多条件显示结构视图

13.5.2 登录验证和信息审核

在 admin.asp 中,如果用户没有登录,则会显示登录表单,要求输入用户信息。当提交表单后,则会跳转到 adminsave.asp 文件,系统将对用户输入的信息进行验证。

当用户已经登录,同时数据库中还存在没有提交的记录(即字段 key 为 False 的记录),则会在页面中显示这些记录(登录用户名和密码都为 admin),如图 13.26 所示。



图 13.26 显示没有提交的记录

这时如果用户选择某些记录前边的复选框，然后单击【提交】按钮，提交 admin.asp 文件的表单时，也就把选定歌曲记录的 id 号传递过去，同时用查询字符串参数 action=htc 指示 adminsave.asp 文件执行相应的数据库更新操作。具体操作步骤如下。

01 建立一个数据库连接文件 conn.asp，用来打开数据库连接。代码如下：

```
<!--#include file="Connections/conn.asp" -->
<%
dim conn
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open MM_conn_STRING
%>
```

02 打开 adminsave.asp 文件，输入下面代码，用来执行用户登录验证和相关数据操作。

```
<!--#include file="conn.asp"-->
<% '用户身份验证
if request("action")="login" then '判断用户是否从admin.asp文件的登录页面状态跳转过来
    if request.form("user")="admin" and request("pass")="admin" then '如果是，则再判断
        登录输入的信息是否合法
        session("admin")="admin" '然后，把这些合法信息保存到session变量中
        response.Redirect "admin.asp" '再次跳转到admin.asp文件，以便用户对未提交记录进行处理
        或增加记录
    else
        response.write"<script>alert('对不起，用户名或密码不正确');history.go(-1);</script>"
        '登录错误提示
        response.end()
        response.Redirect "admin.asp" '再次跳转到admin.asp文件，要求重新登录
    end if
'更新数据库中相关记录的key字段值
elseif request("action")="htc" then '判断用户是否从admin.asp文件的记录处理页面状态跳转过来
    id=split((request.form("id")),",") '获取用户选择的要处理记录编号，并转换为数组
    for i=0 to UBound(id) '循环改写数据库中key字段值
        sql="select * from musiclist where id=" & id(i)
        set rs=server.createobject("adodb.recordset") '定义一个记录集，便于下面的对数据库
        的改写
        rs.open sql,conn,1,3
        rs("key")=true '改写key字段的值
        rs.update
        rs.close '关闭记录集
        set rs=nothing
    next
    response.Redirect "admin.asp" '再次跳转到admin.asp，继续进行新的处理
end if
%>
```

13.5.3 增加信息

save1.asp 文件是专门为 musicadd1.asp 文件的表单服务的。它以代码的形式快速把用户在 musicadd1.asp 页面表单中输入的信息保存到数据库中。

在前面实例中多次介绍使用“增加记录”服务器行为来实现数据的写入。为了帮助读者提高编程能力，这里就不再介绍如何用可视化操作实现插入记录的方法，而是通过编写代码来实现。打开 save.asp 文件，输入下面代码：

```
<!--#include file="conn.asp"-->
<%
if request("action")="add" then '判断是否从musicadd1.asp文件跳转过来的
    '定义变量，并保存musicadd1.asp文件中表单传递过来的数据
    dim music,tv,ycr,guoyue,zs,eng
```

```

music=request("music")
tv=request("tv")
ycr=request("ycr")
guoyue=request("guoyue")
zs=request("zs")
eng=request("eng")
user=request("user")
if user="" then user=null else user=user
tel=request("tel")
if tel="" then tel=null else tel=tel
add=request("add")
if add="" then add=null else add=add
mail=request("mail")
if mail="" then mail=null else mail=mail
if eng="" or guoyue="" or zs="" or tv="" or music="" then '判断musicadd1.asp文件
中表单是否输入信息, 否则返回要求重新输入信息
    response.write"<script>alert('对不起, 请输入完整的歌曲资料');history.go(-1);</script>"
    response.end()
end if
'打开记录集, 查询插入的记录是否存在, 如果存在则返回要求重新输入
Set rs=Server.CreateObject("Adodb.RecordSet")
rs.Open "Select * from musiclist where music='&music&' and tv='&tv&'",conn
if not rs.EOF then
    Response.Write "<script>alert('对不起, 此歌曲已经存在!');history.go(-1);</Script>"
    Response.End
end if
'定义敏感字符列表, 并验证用户输入的信息中是否包含这些字符
ping="&|<|>|*|;|'|'""
pingb = split((ping), "|") '转换为数组
for i=0 to UBound(pingb) '循环验证用户输入的信息是否存在敏感字符
    if instr(user,pingb(i)) or instr(mail,pingb(i)) or instr(tel,pingb(i)) or
instr(add,pingb(i)) or instr(eng,pingb(i)) or instr(ycr,pingb(i)) or instr(zs,pingb(i)) or
instr(music,pingb(i)) or instr(tv,pingb(i)) or instr(guoyue,pingb(i)) then
        response.write"<script>alert('对不起, 你的歌曲资料中含有敏感字符
');history.go(-1);</script>"
        response.end()
    end if
next
'定义记录集, 把用户输入的数据插入到数据表中
sql="select * from musiclist"
set rs=server.createobject("adodb.recordset")
rs.open sql,conn,1,3
rs.addnew
rs("music")=music
rs("tv")=tv
rs("ycr")=ycr
rs("guoyue")=guoyue
rs("zs")=zs
rs("eng")=eng
rs("user")=user
rs("tel")=tel
rs("add")=add
rs("mail")=mail
rs.update
rs.close
set rs=nothing
Response.Redirect "musicadd1.asp"
end if
%>

```

save.asp 文件是专门为 musicadd.asp 文件的表单服务的, 其内容和 save1.asp 基本相同, 用户可参照上面代码段, 根据 musicadd1.asp 文件的表单结构修改要读写的记录。要注意, 由于 save.asp 文件是针对所有用户添加歌曲的, 因此所添加的信息并没有得到管理员的同意, 所以要把字符 key 的值设置为 false。其中插入记录的代码如下:

```
<% '定义记录集,把用户输入的数据插入到数据表中
sql="select * from musiclist"
set rs=server.createobject("adodb.recordset")
rs.open sql,conn,1,3
rs.addnew
rs("music")=music
rs("tv")=tv
rs("ycr")=ycr
rs("guoyue")=guoyue
rs("zs")=zs
rs("eng")=eng
rs("user")=user
rs("tel")=tel
rs("add")=add
rs("mail")=mail
rs("htc")=rs("htc")+1
rs("thtc")=rs("thtc")+1
rs("tdate")=date()
rs("key")=false
rs.update
rs.close
set rs=nothing
%>
```

其他代码都与 save1.asp 文件的代码相同,所以就不再重复介绍。

13.6 实践总结

本章重点介绍了网上调查表的实现过程,包括数据的多行多列动态表格显示、动态样式表控制页面显示等,这些技术在网页编程中经常使用,希望用户能熟练掌握运用。同时,多条件查询也是非常实用的技术,经常在页面中使用。

在本例功能实现过程中,没有设计保存投票人信息功能,仅把投票信息累计保存在原调查信息表中。这也带来一个问题,如果想汇总分析投票结果就无法实现,因此还应该增加用户登录功能并能在数据库中同时保存用户投票信息,这样就可以实现更多功能。



博客被誉为是继 E-mail、BBS、ICQ 之后出现的第四种网络交流方式。作为一种新的网络应用方式和一种新的文化现象,博客的出现和繁荣,真正凸显网络的知识价值,标志着互联网发展开始步入更高的阶段。博客秉承了个人网站的自由精神,在网络世界里体现个人的存在,张扬个人的社会价值,拓展个人的知识视野,建立属于自己的交流沟通的群体。从这个意义上说,博客将会变得越来越普及,越来越为更多的人接受。本章将详细介绍个人博客的设计和开发方法。

● 技术要点

- 日志前台显示
- 图片共享
- 音乐共享
- 留言设计
- 评论和友情链接
- 博客后台综合管理

14.1 程序设计分析

博客(Weblog, 缩写为 Blog, 中文翻译为网络日志)是一种新的 Web 应用,用户可以在其中即时发布个人日志与他人交流,提供网络资源,实现信息共享。博客为每一个人提供了一个信息发布、知识交流的窗口,博客使用者可以很方便地用文字、链接、影音、图片建立起自己个性化的网络世界。最初的 Blog 只是个人通过网络发布自己写的一些文字或者收藏相关的文章和评论等,目前, Blog 已成为家庭、公司、部门和社会群体之间越来越重要的沟通工具。互联网的发展经历了以门户为代表的 Web1.0 阶段,并已经步入了以博客、播客、维客和 RSS 等应用为代表的 Web2.0 阶段。

Blog 主要包括以下 7 种类型。

(1) 基本的博客。Blog 中最简单的形式。某个作者对于特定的话题提供相关的资源,发表简短的评论。这些话题几乎可以涉及所有领域。

(2) 小组博客。基本的博客的简单变型,一些小组成员共同完成博客日志,有时候作者不仅能编辑自己的内容,还能够编辑别人的条目。这种形式的博客能够使得小组成员就一些共同的话题进行讨论,甚至可以共同协商完成同一个项目。

(3) 亲朋之间的博客。这种类型博客的成员主要由亲属或朋友构成,他们是一种生活圈、

一个家庭或一群项目小组的成员。

(4) 协作式博客。与小组博客相似，其主要目的是通过共同讨论使得参与者在某些方法或问题上达成一致，通常把协作式的博客定义为允许任何人参与、发表言论和讨论问题的博客日志。

(5) 公共社区博客。公共出版在几年以前曾经流行过一段时间，但是因为没有持久有效的商业模式而销声匿迹了。廉价的博客与这种公共出版系统有着同样的目标，但是使用更方便，所花的代价更小，所以也更容易生存。

(6) 商业、企业广告型博客。对于这种类型博客的管理类似于通常网站的 WEB 广告管理。

(7) 知识库博客。或者叫 K-Log。基于博客的知识管理将越来越广泛，使得企业可以有效地控制和管理那些原来只是由部分工作人员拥有的、保存在文件档案或者个人电脑中的信息资料。知识库博客提供给了新闻机构、教育单位、商业企业和个人一种重要的内部管理工具。

博客出现之前，BBS 是主要网络应用形势。网友从论坛发言，到做个人主页，再到成为博客的转变过程，其中可显示这 3 种个人意见发布方式的不同之处。

在公共论坛上张贴帖子，是网上发言的常用形式。但是，在论坛上发帖要受论坛主题和版主的管理限制，因此那些希望享受更多自由表达权并集中保存自己意见和发言内容的网络用户，就自然转向个人主页的解决方案。

然而，个人主页的难点在于，个人是否有足够的信息资源来长久维持其主页的魅力系数居高不下。从思索和解决这一难题入手，一些网络个人用户逐渐转向博客样式。

博客网站是个人主页的集中化。个人网站由于其对网民的技术水准、资金条件和维护资源的高层次要求，而注定只能少数人拥有，而博客网站的低门槛，则使其具有很高的开放性。

Blogger 的软件是免费的，要成为博客，用户只需在网上免费或付费的博客网站上申请注册，就可以拥有属于自己的一方领地。博客网站还具有个人主页所缺乏的公共性。如果说，个人网站只是一对多的传播出口，其涉及的多是有关个人生活、事业和情感方面的私人话题和私人资讯的话，那么博客网站则是无数人对无数人的传播社区，这里既有个人爱好，也有公众趣味，既有私人观点，也有群体争论。博客的价值一方面体现为个性化，而另一方面，这种个性化又要被多数人认可。

本实例是针对个人用户而打造的功能强大的博客应用系统，用户只需要申请一个域名和购买一个支持 ASP 技术的空间，即可建立自己的博客网站梦想。本博客流程设计如图 14.1 所示。本系统预览效果如图 14.2 所示。

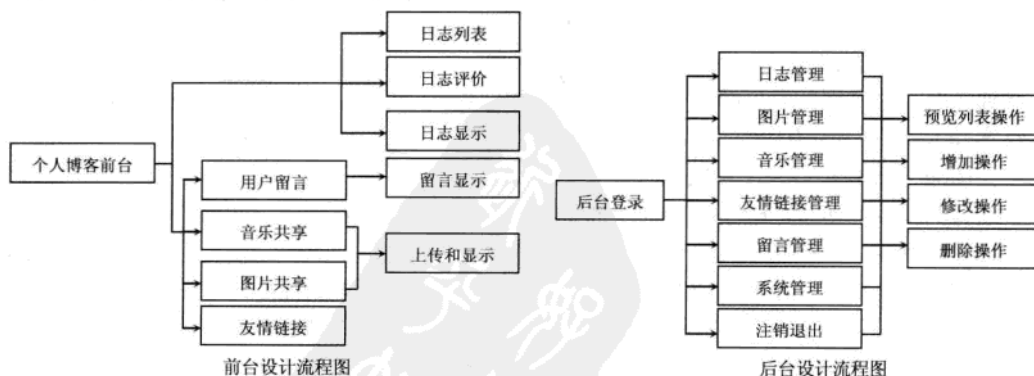


图 14.1 博客设计流程图

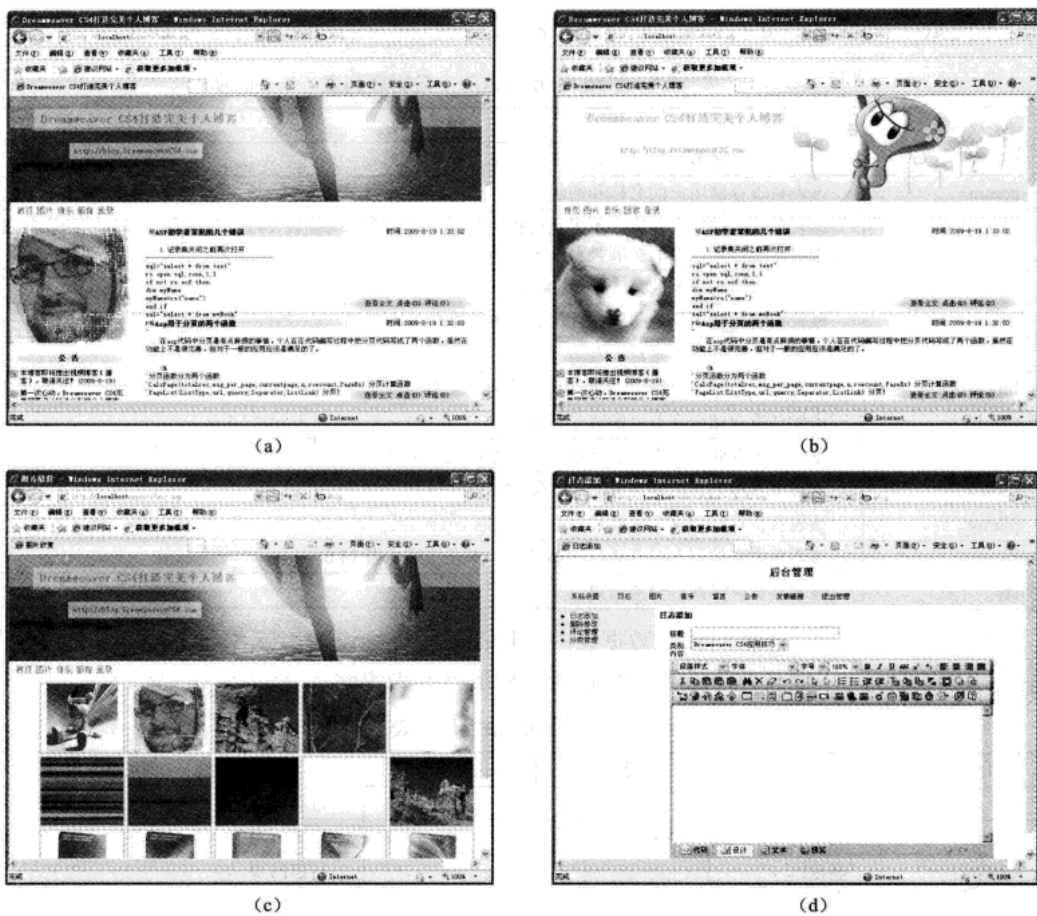


图 14.2 个人博客浏览效果

(a) 博客首页皮肤1; (b) 博客首页皮肤2; (c) 共享图片; (d) 超强的后台管理功能



提示

学习本实例之前,读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录,如mysite。启动 Dreamweaver CS5 创建一个动态站点,远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP/JavaScript”,【访问】应为“本地/网络”。

把本书光盘中的“14”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构,并帮助读者建立了数据库连接。

14.2 数据库设计

在 Web 应用开发中,数据库设计是一项重要的基础性工作。为了便于管理,设计师常会把前台数据与后台数据分离开来,根据系统的规模决定是否把前台、后台数据分别存储于不同数据库或数据表中。前台数据库专用于信息的存储和显示,而后台数据库专用于系统管理中各种配置参数的设置和存储。本例 Blog 应用程序是一个功能相对完整和独立的系统,而不是一个孤立的模块,因此对于数据库的设计也相对复杂些。它涉及博客信息的存储和系统配置的管理。

本例数据库 blog.mdb 共包括 13 个数据表，这些数据表的功能说明见表 14.1。

表 14.1 blog.mdb 数据库中数据表列表

数据表	说 明
ad	该表负责存储 Blog 的名称和 URL 信息
admin	该表负责存储用户的名称和密码
cont	该表负责存储日志内容，本表结构相对复杂些，详细说明见表 14.2。该表结构设计是否合理和完善对整个博客功能的扩展具有重要影响
gg	该表负责博客系统的公告信息的存储和管理
img	该表负责存储用户上传的图片信息，本表结构见表 14.3
link	该表负责存储博客站点的友情链接
liuyan	该表负责存储用户留言信息，本表结构见表 14.4
muc	该表负责存储博客首页的背景音乐
music	该表负责存储用户上传的音乐，本表结构见表 14.5
pichead	该表负责存储页面顶部导航区域的背景图片信息
picmy	该表负责存储首页个人照片信息
re	该表负责存储网友对具体日志的评价，本表结构见表 14.6
sty	该表负责存储博客分类信息

限于篇幅，本节将不再对每个数据表的结构都进行详细说明，读者可以参阅本书光盘中提供的实例模板。下面就几个重要的数据表结构进行列表说明。

表 14.2 cont 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
title	文本	50	否	日志标题
content	备注	-	-	日志内容
re	备注	-	-	日志回复信息
time	日期/时间	-	-	发布时间，【默认值】设置为“Now()”
sty	文本	50	否	日志分类
hitnum	数字	长整型	否	日志被点击的次数
renum	数字	长整型	否	日志被回复评价的次数

表 14.3 img 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
name	文本	50	否	图片的名称（或者 url 的名称）
title	文本	50	否	图片提示信息
text	备注	-	-	图片说明内容
sty	文本	50	否	图片分类

表 14.4 liuyan 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
name	文本	50	否	留言名称
liuyan	备注	-	-	留言信息
time	日期/时间	-	-	留言时间，【默认值】设置为“Now()”
web	文本	50	否	留言者的个人站点

表 14.5 music 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
url	备注	-	-	音乐的 URL 地址（绝对路径）
name	文本	50	否	音乐名称
sty	文本	50	否	音乐分类

表 14.6 re 表字段列表

字 段	类 型	字段大小	必填字段	说 明
idd	自动编号	-	-	自动编号
re	备注	-	-	评价内容
id	数字	长整型	否	评价的日志编号
tmie	日期/时间	-	-	评价时间，【默认值】设置为“Now()”
name	文本	50	否	评价者的姓名
qq	文本	50	否	评价者的 QQ 号码

14.3 前台公共模块功能实现

一个复杂的系统都由很多模块组合而成，不同模块之间协同工作共同实现整个系统的功能。模块可以包含大的综合模块，例如一个大型网站的不同频道和服务区域；也可以包含某个单一功能的模块，例如注册表单、留言板、导航条等。

本实例就是由很多个这样的小模块组成，它们之中有专用模块，例如显示评价和留言等，也有很多共同模块，例如页面顶部的导航区域、底部的版权信息区域、左侧的公共功能模块。下面详细介绍这些公共模块的实现方法。

14.3.1 导航模块

由于每个页面都需要顶部导航区域，所以把它作为一个单独的模块来设计是非常必要的，这样能够提高开发速度，充分利用现有代码，避免重复劳动。本例导航模块主要功能包括顶部背景图像的动态设计，以及导航条设计。具体实现步骤如下。

01 打开 header.asp 文件，在【绑定】面板中定义两个记录集 rs 和 rs1。其中记录集 rs1 查询数据表 pichead 中的顶部区域背景图像的 URL 信息，以实现背景图像的动态控制效果；记录集 rs1 查询数据表 ad 中的信息，以获取博客站点的名称和链接地址。设置如图 14.3 和图 14.4 所示。

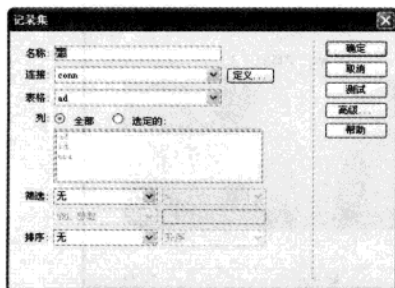


图 14.3 定义 rs 记录集

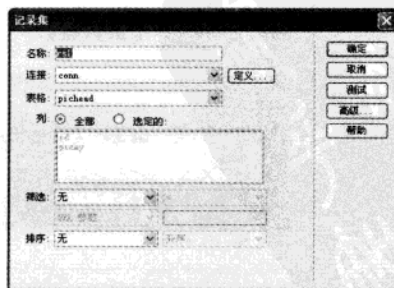


图 14.4 定义 rs1 记录集

- 02** 把这两个记录集中的字段绑定到页面中，读者可以切换到【代码】视图，准确绑定字段到页面内绝对定位的标签内，以及绑定到

```
<div id="header">
  <div id="header1">" width=
"800" height="183" />
  <ul>
    <li><a href="index.asp" target="_parent">首页</a>
    <li><a href="pic.asp" target="_blank">图片</a>
    <li><a href="muc.asp" target="_blank">音乐</a>
    <li><a href="liuyan.asp" target="_blank">留言</a>
    <li><a href="admin/index.asp" target="_blank">登录</a></li>
  </ul>
  <div class="h-titile"><%= (rs.Fields.Item("ad").Value) %></div>
  <div class="h-url"><%= (rs.Fields.Item("url").Value) %></div>
</div>
</div>
```

- 03** 用CSS绝对定位博客名称和URL信息在页面中的显示位置，代码如下：

```
.h-titile { /* 定义标题样式 */
  font-size:20px; /* 标题字体大小 */
  font-weight:bold; /* 标题加粗 */
  color:#D52E8A; /* 字体颜色 */
  position:absolute; /* 绝对定位，这样就可以不受其他元素位置的影响 */
  top:20px; /* 距离最近包含块 (<div id="header">元素) 的顶部距离 */
  left:40px; /* 距离最近包含块 (<div id="header">元素) 的左侧距离 */
  padding:0.4em 0.6em; /* 调整标题的显示位置 */
  background:#DBF789; /* 定义背景色 */
  filter:alpha(opacity=50); /* 设置在IE浏览器中的透明度 */
  opacity:.5; /* 设置在其他浏览器中的透明度 */
}
.h-url { /* 定义URL样式，具体规则与上面类似，就不再说明 */
  font-size:14px;
  color:red;
  position:absolute;
  top:80px;
  left:100px;
  padding:0.4em 0.6em;
  background:#F5FCDB;
  filter:alpha(opacity=60);
  opacity:.6;
}
```

- 04** 按F12键在浏览器中预览，顶部导航区域的设计效果如图14.5所示。



图 14.5 顶部导航区域设计效果

14.3.2 脚部版权模块

脚部版权模块设计比较简单,都是一些简单的静态信息显示,打开 footer.asp 文件,设计如图 14.6 所示。读者可以根据需要不断增加其他帮助信息或者辅助信息。

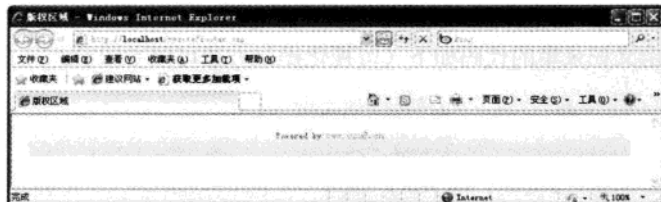


图 14.6 脚部版权区域设计效果

14.3.3 通用功能模块

在博客站点内,设计师常会把一些重要信息或者最新信息按功能简单罗列在一侧,以方便网友快速浏览,例如最新日志、最新评价、最新留言、日志分类等。在本实例中,通用功能模块被放置在页面的左侧(当然读者也可以放置在页面的右侧),主要功能包括博客照片、博客公告、最新文章、博文分类、日历、最新评论、最新留言、友情链接、博客统计、背景音乐控制等功能模块。

为了方便开发,可根据功能不同分别在不同页面中进行设计,最后组合在一起,成为一个完整的通用模块。打开 left.asp 文件,输入下面代码可以实现功能的组合:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>左侧通栏</title>
<link href="images/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<!--#include file="left_photo.asp" --><!--博客照片-->
<!--#include file="left_gg.asp" --><!--博客公告-->
<!--#include file="left_doc.asp" --><!--最新文章-->
<!--#include file="left_class.asp" --><!--日志分类-->
<!--#include file="left_diary.asp" --><!--日历-->
<!--#include file="left_talk.asp" --><!--最新评论-->
<!--#include file="left_message.asp" --><!--最新留言-->
<!--#include file="left_link.asp" --><!--友情链接-->
<!--#include file="left_count.asp" --><!--博客统计-->
<!--#include file="left_music.asp" --><!--背景音乐控制-->
</body>
</html>
```

读者可以根据需要增加或者减少通用模块内的小模块显示数目,也可以设计一个模块动态显示功能,把通用模块的具体显示权力交给网友自己(后面还将就这个问题进行介绍)。

下面按节逐个介绍这些功能模块的实现方法。

14.3.4 博客照片动态控制

博客照片一般位于站点首页的第一屏的某个重要位置,以方便网友相互认识。为了方便

用户管理，本例把博客照片的显示设计为动态控制，用户可以在后台随时更换个人照片。为了实现这个功能，需要后台数据库的支持，在数据库中设计一个表专门用存储博客照片的 URL 信息，这样用户可以在后台随时更新该 URL，以更换照片，在前台就可以实现动态显示效果，预览效果如图 14.7 所示。具体实现步骤如下。

01 打开 `left_photo.asp` 文件，在该页面中定义一个记录集，来查询 `picmy` 数据表中的照片 URL 信息。定义记录集的代码如下（可视化操作方法可以参阅前面几章介绍的方法）：

```
<%
Dim rs
Dim rs_cmd
Dim rs_numRows

Set rs_cmd = Server.CreateObject ("ADODB.Command")
rs_cmd.ActiveConnection = MM_conn_STRING
rs_cmd.CommandText = "SELECT * FROM picmy ORDER BY id DESC"
rs_cmd.Prepared = true

Set rs = rs_cmd.Execute
rs_numRows = 0
%>
```

02 在【绑定】面板中，把 `picmy` 字段绑定到 `` 标签的 `src` 属性上，代码如下：

```
<dl class="left">
  <dt class="nodisplay">首页头片</dt>
  <dd class="bg_none">
    " width="200" />
  </dd>
</dl>
```

03 为了防止照片过大而撑破栏目的宽度，可以定义 `width="200"` 属性限制照片的显示宽度，该宽度为左侧栏目的宽度，高度可以随系统按比例自由伸展。本模块以及后面其他模块都使用 `<dl>`、`<dt>` 和 `<dd>` 这 3 个配套标签来控制页面的结构，然后利用 CSS 来控制模块的显示样式，具体布局代码可以参阅模板实例中的 `style.css` 样式表文件，显示效果如图 14.7 所示。

04 删除 ASP 中冲突代码。为了避免在模块组合中发生代码冲突，最后应该删除下面这些代码（可以参阅“实例效果”/“第 2 步-模块组合”文件夹中的该文件代码）：

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="Connections/conn.asp" -->
<%
Dim rs
Dim rs_cmd
Dim rs_numRows
%>
```



提示

删除上面代码之后，读者就无法直接预览该页面了，也无法在【绑定】面板中操作已定义的记录集，如果要修改记录集或者预览页面效果，应该补加上面被删除代码。

05 删除 HTML 代码。为了优化 HTML 代码，读者还需要删除 HTML 的页面结构代码（可以参阅“实例效果”/“第 3 步-优化代码”文件夹中的该文件代码），例如：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>首页头片</title>
<link href="images/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
...
</body>
</html>

```

- 06** 因为多个文件组合后，会在同一个文件中产生多个相同而又重复的页面结构标签，这些重复标签虽然对于预览没有太大影响，但容易造成代码冗余，甚至可能产生错误。

14.3.5 博客公告

博客公告功能与博客照片功能实现方法基本相同，具体操作步骤如下。

- 01** 打开 left_gg.asp 文件，在【绑定】面板中定义一个记录集，查询 gg 数据表中的所有字段信息，设置记录集按 id 字段降序排列，这样就可以保证最新发布的公告能够排列在前面。该记录集核心代码如下：

```

<%
Set rs_cmd = Server.CreateObject ("ADODB.Command")
rs_cmd.ActiveConnection = MM_conn_STRING
rs_cmd.CommandText = "SELECT * FROM gg ORDER BY id DESC"
rs_cmd.Prepared = true

Set rs = rs_cmd.Execute
%>

```

- 02** 把字段 gonggao 和 time 绑定到页面<dd>标签中，代码如下：

```

<dl class="left">
<dt>公告</dt>
<dd><%= (rs.Fields.Item("gonggao").Value) %><%= (rs.Fields.Item("time").Value) %></dd>
</dl>

```

- 03** 选中<dd>和</dd>标签，以及其内部绑定的字段，在【服务器行为】面板中添加“重复区域”服务器行为，在【重复区域】对话框中设置重复记录为 5 条，如图 14.8 所示。



图 14.7 博客照片



图 14.8 定义并绑定记录集

- 04** 删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅上节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅上节介绍）。

14.3.6 最新日志列表

最新日志列表显示用户最新发布的日志，该模块能够方便网友了解本博客最新动态。具体操作步骤如下。

- 01** 打开 left_doc.asp 文件，在【绑定】面板中定义一个记录集，查询 cont 数据表中的所有字段信息，设置记录集按 id 字段降序排列，这样就可以保证最新发布的日志能够排列在前面。该记录集核心代码如下：

```
<%
Set rs_cmd = Server.CreateObject ("ADODB.Command")
rs_cmd.ActiveConnection = MM_conn_STRING
rs_cmd.CommandText = "SELECT id, title FROM cont ORDER BY id DESC"
rs_cmd.Prepared = true

Set rs = rs_cmd.Execute
%>
```

- 02** 把字段 title 绑定到页面<dd>标签中，同时为该绑定字段定义一个超链接，单击该链接将会跳转到 show.asp 文件，并把该日志的 id 编号利用查询字符串的方式传递给 show.asp 文件，然后利用该 id 编号在该页面中显示日志详细内容。代码如下：

```
<dl class="left">
  <dt>公告</dt>
  <dd><A HREF="show.asp?id=<%= (rs.Fields.Item("id").Value)%%" target=" blank"><%=
(rs.Fields.Item("title").Value)%></A></dd>
</dl>
```

- 03** 选中<dd>和</dd>标签，以及其内部绑定的字段，在【服务器行为】面板中添加“重复区域”服务器行为，在【重复区域】对话框中设置重复记录为 5 条。

- 04** 删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅第 14.3.4 节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍）。

- 05** 按 F12 键预览该文件，显示效果如图 14.9 所示。

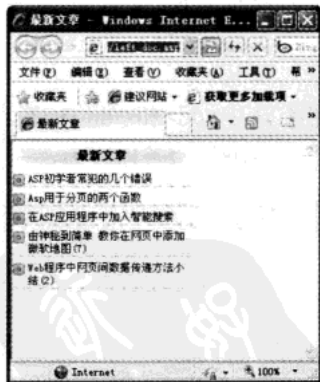


图 14.9 最新文件效果

14.3.7 日志分类

日志分类是博客站点内一个重要的功能模块，通过分类可以方便日志的管理与浏览。实现日志分类的具体操作步骤如下。

- 01** 打开 left_class.asp 文件，在【绑定】面板中定义一个记录集，查询 sty 数据表中的所有字段信息，设置记录集按 id 字段降序排列，这样可以保证新增加的分类显示在上面。该记录集核心代码如下：

```
<%
Set rs_cmd = Server.CreateObject ("ADODB.Command")
rs_cmd.ActiveConnection = MM_conn_STRING
rs_cmd.CommandText = "SELECT * FROM sty ORDER BY id DESC"
```

```
rs_cmd.Prepared = true

Set rs = rs_cmd.Execute
%>
```

- 02** 把字段 `sty` 绑定到页面 `<dd>` 标签中, 同时为该绑定字段定义一个超链接, 单击该链接将跳转到 `title.asp` 文件, 并把该分类的 `id` 编号传递过去, 然后在 `title.asp` 文件中详细显示该分类的所有日志, 代码如下:

```
<dl class="left">
  <dt>公告</dt>
  <dd><A HREF="title.asp?id=<%=rs.Fields.Item("id").Value%>"
target=" _blank"><%=rs.Fields.Item("sty").Value%></A></dd>
</dl>
```

- 03** 选中 `<dd>` 和 `</dd>` 标签, 以及其内部绑定的字段, 在【服务器行为】面板中添加“重复区域”服务器行为, 在【重复区域】对话框中设置重复记录为所有, 即重复显示所有分类。
- 04** 删除文件组合后可能发生冲突的 ASP 代码 (具体被删除的 ASP 代码请参阅第 14.3.4 节介绍)。同时优化清理 HTML 页面结构代码 (具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍)。
- 05** 按 F12 键预览该文件, 显示效果如图 14.10 所示。



图 14.10 日志分类

14.3.8 设计日历

在博客网站中一般都会设计一个日历, 这样便于网友查看时间。有些博客网站还能够在日历中智能显示日志信息, 即当用户在某天写过日志后, 会在日历中的具体日期上显示一个超链接, 单击该日期, 可以快速跳转到当日所写的具体日志所在的页面, 这样能够方便网友按日历查看日志 (在后面扩展功能中会详细介绍)。

打开 `left_diary.asp` 文件, 切换到【代码】视图, 在 `<dd>` 和 `</dd>` 标签之间添加下面代码, 显示效果如图 14.11 所示。

```
<table width="180" cellpadding="0" cellspacing="1"
align=center>
  <% '获取当前日期
  If Trim(Request("ReqDate"))="" or ( not IsDate(Trim(Request("ReqDate"))) ) then
    CurrentDate=Date
  else
    CurrentDate=Trim(Request("ReqDate"))
  end if
  PreviousMonthDate=DateAdd("m",-1,CurrentDate) '获取上个月前的时间
  NextMonthDate=DateAdd("m",1,CurrentDate) '获取下一个月前的时间
  %>

  <tr align="LEFT">
    <td width="14%" height="19" align="center"></td>
    <td colspan="5" align="center"><%=year(CurrentDate) & "年" & month(CurrentDate)
& "月"%></td>
    <td width="14%" align="center"></td>
  </tr>
```



图 14.11 日历设计效果

```
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 日</td>  一</td>  二</td>  三</td>  四</td>  五</td>  六</td> | | | | | | |

```

```

<%
ym=year(CurrentDate) & "-" & month(CurrentDate) & "-" '组合当前年、月时间
i=1
do while i<33 '从1开始循环判断一个月中所有日期，逐个判断每个数字属于星期几并进行排列显示
    j=1
    response.write("<tr height=19>")
    do while j<8 '循环判断一周内日期显示，即单元格所在行的显示日期
        If IsDate(ym&i) then
            CurrentWeekDay=weekday(ym&i)
            if j=CurrentWeekDay then
                If Datediff("d",ym&i,now)>-1 then
                    LinkText="&i"</a>"
                else
                    LinkText=i
                end if
                if i<>Day(now) then '如果不是今天，则按普通格式显示，否则为今日添加背景色
                    response.write("<td>&nbsp;&nbsp;&nbsp;& "&LinkText"</td>")
                else
                    response.write("<td bgcolor=#e0e594>&nbsp;&nbsp;&nbsp;& "&LinkText"</td>")
                end if
                i=i+1
            else '如果循环日期的星期数与循环数不相等，则显示空白单元格
                response.write("<td></td>")
            end if
            else '如果不是日期则显示空白单元格
                response.write("<td></td>")
                i=i+1
            'exit do
        end if
        j=j+1
    loop
    response.write("</tr>")
loop
%>
</table>

```

14.3.9 最新评论列表

最新评论列表显示网友最新的对日志的评论，实现该功能模块的具体操作步骤如下。

- 01** 打开 left_talk.asp 文件，在【绑定】面板中定义一个记录集，查询 re 数据表中的 id、idd、re 字段信息，设置记录集按 id 字段降序排列，这样就可以保证最新评论能够排列在前面。该记录集核心代码如下：

```

<%
Set rs_cmd = Server.CreateObject ("ADODB.Command")
rs_cmd.ActiveConnection = MM_conn_STRING
rs_cmd.CommandText = "SELECT id, idd, re FROM re ORDER BY idd DESC "
rs_cmd.Prepared = true

Set rs = rs_cmd.Execute
%>

```

- 02 把字段 re 绑定到页面<dd>标签中,同时为该绑定字段定义一个超链接,单击该链接将会跳转到 show.asp 文件,并把该日志的 id 编号利用查询字符串的方式传递给 show.asp 文件,然后利用该 id 编号在该页面中显示日志详细内容以及网友对该日志的评论。代码如下:

```
<dl class="left">
  <dt>公告</dt>
  <dd><a href="show.asp?id=<%=rs.Fields.Item("id").Value%>" target="blank"><%=
(rs.Fields.Item("re").Value)%></a></dd>
</dl>
```

- 03 选中<dd>和</dd>标签,以及其内部绑定的字段,在【服务器行为】面板中添加“重复区域”服务器行为,在【重复区域】对话框中设置重复记录为8条。

- 04 删除文件组合后可能发生冲突的 ASP 代码(具体被删除的 ASP 代码请参阅第 14.3.4 节介绍)。同时优化清理 HTML 页面结构代码(具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍)。

- 05 按 F12 键预览该文件,显示效果如图 14.12 所示。



图 14.12 最新评论列表

14.3.10 最新留言列表

实现最新留言列表模块的具体操作步骤如下。

- 01 打开 left_message.asp 文件,在【绑定】面板中定义一个记录集,查询 liuyan 数据表中的所有字段信息,设置记录集按 id 字段降序排列,这样就可以保证最新留言能够排列在前面。其中的 SQL 字符串代码如下,其他都与上面定义的记录集操作和代码都相同:

```
<%
rs_cmd.CommandText = "SELECT * FROM liuyan ORDER BY id DESC"
%>
```

- 02 把字段 liuyan 绑定到页面<dd>标签中,同时为该绑定字段定义一个超链接,单击该链接将会跳转到 liuyan.asp 文件,并把该留言的 id 编号利用查询字符串的方式传递给 liuyan.asp 文件,然后利用该 id 编号在该页面中显示留言的详细内容。代码如下:

```
<dl class="left">
  <dt>公告</dt>
  <dd><a href="liuyan.asp?id=<%=rs.Fields.Item("id").Value%>" target="blank"><%=
(left(rs.Fields.Item("liuyan").Value,25))%>...</a> </dd>
</dl>
```

- 03 选中<dd>和</dd>标签,以及其内部绑定的字段,在【服务器行为】面板中添加“重复区域”服务器行为,在【重复区域】对话框中设置重复记录为8条。

- 04 删除文件组合后可能发生冲突的 ASP 代码(具体被删除的 ASP 代码请参阅第 14.3.4 节介绍)。同时优化清理 HTML 页面结构代码(具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍)。

- 05 按 F12 键预览该文件,显示效果如图 14.13 所示。



图 14.13 最新留言列表

14.3.11 友情链接列表

实现友情链接列表模块的具体操作步骤如下。

- 01** 打开 left_link.asp 文件，在【绑定】面板中定义一个记录集，查询 link 数据表中的所有字段信息，设置记录集按 id 字段降序排列，其中的 SQL 字符串代码如下，其他都与上面定义的记录集操作和代码都相同：

```
<%
rs_cmd.CommandText = "SELECT * FROM link ORDER BY id DESC"
%>
```

- 02** 把字段 link 绑定到页面<dd>标签中，同时为该绑定字段定义一个超链接，单击该链接将会跳转到相应的站点中，代码如下：

```
<dl class="left">
  <dt>公告</dt>
  <dd><a href="<%=rs.Fields.Item("url").Value)%>" target="blank"><%=rs.Fields.
Item("link").Value)%></a></dd>
</dl>
```

- 03** 选中<dd>和</dd>标签，以及其内部绑定的字段，在【服务器行为】面板中添加“重复区域”服务器行为，在【重复区域】对话框中设置重复记录为 8 条。

- 04** 最后删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅第 14.3.4 节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍）。

- 05** 按 F12 键预览该文件，显示效果如图 14.14 所示。



图 14.14 友情链接列表

14.3.12 博客统计

博客统计功能也是一个常用模块，在第 10 章中曾经详细介绍过流量统计系统功能的实现。但博客统计主要是针对数据库中的数据进行统计分析。实现博客统计功能的具体操作步骤如下。

- 01** 打开 left_count.asp 文件，在【绑定】面板中定义 5 个记录集，分别查询 5 个数据表中的 id 字段。记录集不需要特殊设置，只需要指定相应的数据表即可，如图 14.15 所示。5 个记录集所用的 SQL 代码如下：

```
<%
rs1_cmd.CommandText = "SELECT id FROM cont"
rs2_cmd.CommandText = "SELECT id FROM re"
rs3_cmd.CommandText = "SELECT id FROM liuyan"
rs4_cmd.CommandText = "SELECT id FROM img"
rs5_cmd.CommandText = "SELECT id FROM music"
%>
```

- 02** 把每个记录集的 total 字段绑定到页面相应提示标签内，如图 14.16 所示。

- 03** 删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅第 14.3.4 节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍）。

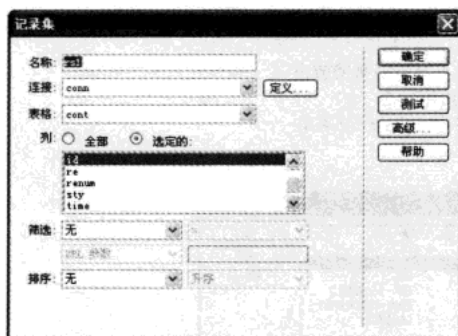


图 14.15 定义记录集



图 14.16 绑定记录集的总数

14.3.13 博客动态背景音乐

博客作为个人在网上公开的私人日记本, 用户都喜欢把自己博客装扮得更酷、更个性些。背景音乐是博客扮酷的一种选择方式。本实例设计的首页背景音乐是动态背景音乐, 用户可以随时在后台进行更新, 这样就能够避免长期听单一背景音乐的枯燥感。实现动态背景音乐功能的具体操作步骤如下。

- 01** 打开 left_music.asp 文件, 在【绑定】面板中定义一个记录集, 查询 muc 数据表中的所有字段信息。其中的 SQL 字符串代码如下:

```
<%
rs_cmd.CommandText = "SELECT * FROM muc"
%>
```

- 02** 在页面<dd>标签中插入一个音乐播放控件。方法是: 在 Dreamweaver CS5 中选择【插入】|【媒体】|【ActiveX】菜单命令, 插入一个 ActiveX 控件, 然后在属性面板中简单设置播放参数, 如图 14.17 所示。

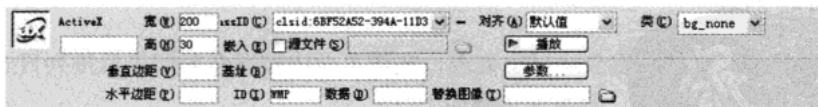


图 14.17 ActiveX 属性面板

- 03** 单击【参数】按钮, 在打开的【参数】对话框中设置音乐播放参数, 其中【URL】属性被绑定为前面定义记录集的 url 字段, 如图 14.18 所示。读者可以切换到【代码】视图简单播放参数 (即【参数】对话框中设置的参数)。

```
<OBJECT id="WMP" height="30" width="200" classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" VIEWASTEXT>
<PARAM NAME="URL" VALUE="<%= (rs.Fields.Item("url").Value) %>">
<PARAM NAME="rate" VALUE="1">
<PARAM NAME="balance" VALUE="0">
<PARAM NAME="currentPosition" VALUE="0">
<PARAM NAME="defaultFrame" VALUE="">
<PARAM NAME="playCount" VALUE="1" > <!--播放一次 -->
```

```

<PARAM NAME="autoStart" VALUE="-1"> <!--自动播放 -->
<PARAM NAME="currentMarker" VALUE="0">
<PARAM NAME="invokeURLs" VALUE="-1">
<PARAM NAME="volume" VALUE="50"> <!--控制音量 -->
<PARAM NAME="mute" VALUE="0">
<PARAM NAME="uiMode" VALUE="full">
<PARAM NAME="stretchToFit" VALUE="0">
<PARAM NAME="windowlessVideo" VALUE="0"> <!--不显示视频播放窗口 -->
<PARAM NAME="enabled" VALUE="-1"> <!--播放控件可用状态 -->
<PARAM NAME="enableContextMenu" VALUE="-1"> <!--显示控制菜单 -->
</OBJECT>

```

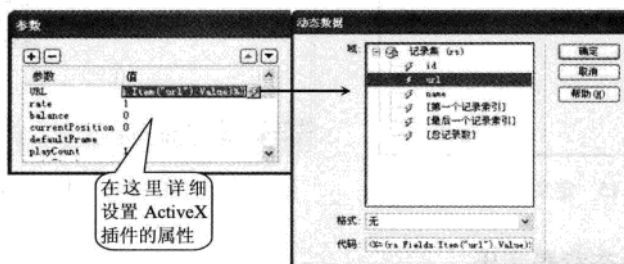


图 14.18 高级定义记录集

- 04** 删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅第 14.3.4 节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍）。

14.4 前台日志功能模块实现

前台日志模块包括日志列表、日志内容显示，以及对日志的评价等几个页面组成。具体说：index_main.asp 文件负责显示日志列表；show.asp 文件是日志详细页，负责组合后面 3 个文件；show_content.asp 负责显示日志详细内容；show_form.asp 文件负责发表评论；show_talks.asp 文件负责显示日志评论列表。

14.4.1 日志列表

index_main.asp 文件负责显示日志列表，在该页面中按发布时间先后排列日志标题、日志部分内容，以及相关信息。其设计步骤如下。

- 01** 打开 index_main.asp 文件，在【绑定】面板中定义一个记录集，查询 cont 数据表中的所有字段信息，设置记录集按 id 字段降序排列，这样就可以保证最新发布的日志能够排列在前面。其中记录集所用 SQL 字符串代码如下：

```

<%
rs_cmd.CommandText = "SELECT * FROM cont ORDER BY id DESC"
%>

```

- 02** 把该记录集相关字段绑定到页面<dd>标签中，同时为该绑定字段定义一个超链接，单击该链接将会跳转到 show.asp 文件，并把该日志的 id 编号利用查询字符串的方式传递给 show.asp 文件，然后利用该 id 编号在该页面中显示日志详细内容。代码如下：

```

<dl class="main">

```

```

<dt> <%=rs.Fields.Item
("title").Value)%> <span class="titl2">时间:<%=rs.Fields.Item("time").Value)%></span>
</dt>
<dd class="sub1"> <%=left(rs.Fields.Item("content").Value,240)%>... </dd>
<dd class="sub2"> <A HREF="show.asp?id=<%=rs.Fields.Item("id").Value %>" target=
" blank">查看全文</A> 点击<%=rs.Fields.Item("hitnum").Value)%>
评论<%=rs.Fields.Item("renum").Value)%> </dd>
</dl>

```

03 选中整个<dl>和</dl>标签, 以及其内部嵌套标签和绑定的字段, 在【服务器行为】面板中添加“重复区域”服务器行为, 在【重复区域】对话框中设置重复记录为10条, 即设置每页显示10条记录。

04 为了防止当没有找到要查询的记录时产生错误, 可以增加“显示区域”服务器行为。选中整个<dl>和</dl>标签以及内部包含的所有内容, 然后在【服务器行为】面板中增加“如果记录集不为空则显示区域”服务器行为(具体操作可以参阅第8.4.5节介绍)。然后在文档的后面增加“暂无文章!”文本, 选中该文本, 为其增加“如果记录集为空则显示区域”服务器行为。

05 删除文件组合后可能发生冲突的 ASP 代码(具体被删除的 ASP 代码请参阅第14.3.4节介绍)。同时优化清理 HTML 页面结构代码(具体被删除的 HTML 代码可以参阅第14.3.4节介绍)。

06 按 F12 键预览该文件, 显示效果如图14.19所示。



图 14.19 日志列表显示效果

14.4.2 日志详细页

show.asp 文件是日志详细页, 它包含了 show_content.asp、show_form.asp 和 show_talks.asp 这3个文件。详细代码如下:

```

<!--#include file="header.asp" -->
<table align="center" width="800" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="200" align="middle" valign="top">
      <!--#include file="left.asp" -->
    </td>
    <td align="center">
      <!--#include file="show_content.asp" -->
      <!--#include file="show_talks.asp" -->
      <!--#include file="show_form.asp" -->
    </td>
  </tr>
</table>
<!--#include file="footer.asp" -->

```

14.4.3 日志内容显示

show_content.asp 文件专门负责显示日志内容, 在该页面中主要定义一个记录集, 记录集

将根据查询字符串中传递的 id 参数来查询指定的日志记录,【记录集】对话框设置如图 14.20 所示。

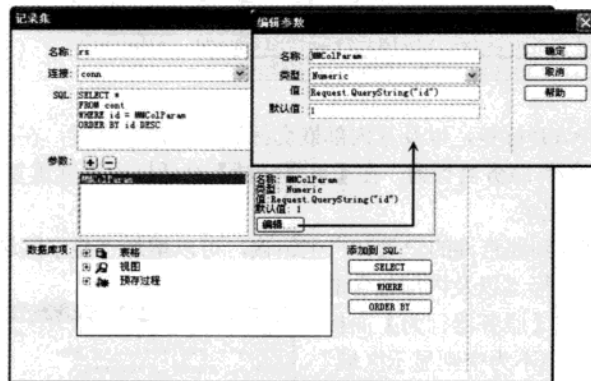


图 14.20 设置记录集

最后把记录集中的字段分别绑定到页面中,代码如下:

```
<dl class="main">
  <dt> <%=rs.Fields.Item
  ("title").Value%> <span class="titl2">时间:<%=rs.Fields.Item("time").Value%></span>
</dt>
  <dd class="sub1"><%=rs.Fields.Item("content").Value%></dd>
</dl>
```

14.4.4 发表评论

发表评论涉及两个文件,在 show_form.asp 文件中提供一个表单域,为用户发表评论提供一个交互接口。然后表单信息被提交到 reaction.asp 文件,该文件将对信息进行验证,如果验证通过把信息插入到 re 数据表中,同时更新 cont 数据表中的 renum 字段的值。具体操作步骤如下。

01 打开 reaction.asp 文件,切换到【代码】视图,首先在代码顶部输入下面代码,用来验证用户输入的验证码是否正确。

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<% '验证用户输入的验证码是否正确
If Trim(Request.Form("validatecode"))=Empty Or Trim(Session("cnbruce.com_ValidateCode"))
<>Trim( Request.Form("validatecode")) Then
  response.write"<script language='javascript'" & chr(13)
  response.write "alert('验证码错误,点确定后再次输入验证码');" & Chr(13)
  response.write "history.go (-1);" & Chr(13)
  response.write "</script>" & Chr(13)
  Response.End
end if
%>
```

02 在【服务器行为】面板中增加一个命令,用来把表单中输入的信息插入到数据库中。代码如下:

```
<% '获取表单值
if(Request("id") <> "") then Command1_newid = Request("id")
if(Request("re") <> "") then Command1_newre = Request("re")
```

```

if(request("name") <> "") then Command1_newname = request("name")
%>
<%判断表单输入值是否为空
if(Request("id") <> "") then Command2_newnum = Request("id")
%>
<%执行向数据库中插入命令
set Command1 = Server.CreateObject("ADODB.Command")
Command1.ActiveConnection = MM_conn_STRING
Command1.CommandText = "INSERT INTO re (id, re, name) VALUES ( '" + Replace(Command1_
newid, "'", "'') + "', '" + Replace(Command1_newre, "'", "'') + "', '" + Replace(Command1_
newname, "'", "'') + "'" )
Command1.CommandType = 1
Command1.CommandTimeout = 0
Command1.Prepared = true
Command1.Execute()
%>

```

03 在【服务器行为】面板中增加一个命令，更新 cont 数据表中的 renum 字段的值。代码如下：

```

<%
set Command2 = Server.CreateObject("ADODB.Command")
Command2.ActiveConnection = MM_conn_STRING
Command2.CommandText = "UPDATE cont SET renum=renum+1 WHERE " + Replace(Command2_newnum,
"'", "'') + "=id"
Command2.CommandType = 1
Command2.CommandTimeout = 0
Command2.Prepared = true
Command2.Execute()
%>

```

14.4.5 显示评论

评论在 show_talks.asp 文件中显示。实现该功能的具体操作步骤如下。

01 打开 show_talks.asp 文件，在【绑定】面板中定义一个记录集，查询 re 数据表中的字段信息，并根据日志 id 编号筛选属于指定日志的评论，【记录集】设置见图 14.20 所示。

02 把记录集相应字段绑定到页面中。代码如下：

```

<dd> 网友:<strong><%=rs.Fields.Item("name").Value)%></strong>发表于<%=rs.Fields.
Item("tmie").Value)%>
<p><%=HTML(rs.Fields.Item("re").Value)%></p>
</dd>

```

03 选中<dd>和</dd>标签，以及其内部绑定的字段，在【服务器行为】面板中添加“重复区域”服务器行为，在【重复区域】对话框中设置重复记录为所有。

04 增加“显示区域”服务器行为。选中整个<dl>和</dl>标签以及内部包含的所有内容，然后在【服务器行为】面板中增加“如果记录集不为空则显示区域”服务器行为（具体操作可以参阅第 8.4.5 节介绍）。在文档的后面增加“暂无评论！”文本，并选中该文本，为其增加“如果记录集为空则显示区域”服务器行为。

05 删除文件组合后可能发生冲突的 ASP 代码（具体被删除的 ASP 代码请参阅第 14.3.4 节介绍）。同时优化清理 HTML 页面结构代码（具体被删除的 HTML 代码可以参阅第 14.3.4 节介绍）。

06 按 F12 键预览 index.asp 首页文件，然后在日志列表中单击“查看全文”超链接，会打开 show.asp 文件，显示效果如图 14.21 所示。

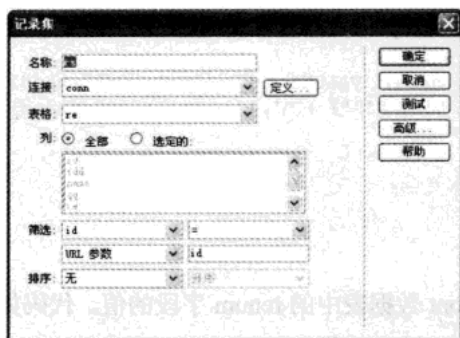


图 14.20 设置【记录集】对话框



图 14.21 日志评论及列表显示效果

14.5 图片分享功能实现

图片分享是博客功能的扩展，它允许用户把个人喜爱的图片上传到博客网站中供网友欣赏。本实例图片分享功能主要由 pic.asp 和 showpic.asp 文件实现。其中 pic.asp 图片负责显示图片列表，而 showpic.asp 文件负责图片的欣赏。

14.5.1 图片列表

pic.asp 文件负责显示图片列表，主要操作包括定义一个记录集显示图片列表，利用“重复区域”服务器行为显示所有的记录，并利用“显示区域”服务器行为控制记录集没有找到记录 i 时如何处理，最后还需要增加“记录集导航条”服务器行为。其设计步骤如下。

01 打开 pic.asp 文件，在【绑定】面板中定义一个记录集，查询 img 数据表中的所有字段信息，设置记录集按 id 字段降序排列，这样就可以保证最新发布的图片能够排列在前面。其中 SQL 查询字符串代码如下：

```
<%
r3.Source = "SELECT * FROM img ORDER BY id DESC"
%>
```

02 将该记录集相关字段绑定到页面中，同时为该绑定字段定义一个超链接，单击该链接将会跳转到 showpic.asp 文件，并把该图片的 id 编号利用查询字符串的方式传递给 showpic.asp 文件，然后利用该 id 编号在该页面中显示大图。代码如下：

```
<A class="box1" HREF="showpic.asp?<%= Server.HtmlEncode(MM_keepNone) & MM_joinChar
(MM_keepNone) & "id=" & r3.Fields.Item("id").Value %>" target="_blank">" height="120" border="0"></A>
```

03 选中上面代码，在【服务器行为】面板中添加“重复区域”服务器行为，在【重复区域】对话框中设置重复记录为所有。

04 增加“显示区域”服务器行为。选中整个重复区域代码，然后在【服务器行为】面板中增加“如果记录集不为空则显示区域”服务器行为（具体操作可以参阅第 8.4.5 节介绍）。

然后在文档的后面增加“暂无图片!”文本,选中该文本,为其增加“如果记录集为空则显示区域”服务器行为。

- 05 把光标置于页面底部,选择【插入】|【数据对象】|【记录集分页】|【记录集导航条】菜单命令,增加一个完整的记录集导航条,在打开的【记录集】对话框中设置如图 14.22 所示。
- 06 同时把记录集 rs3 的 r3_first、r3_last 和 r3_total 这 3 个特殊的字段绑定到页面底部,分别表示记录集分页中当前的第 1 条记录和最后一条记录,以及整个记录集的所有记录数。代码如下:

```
<div align="center">记录 <%= (r3 first) %> 到 <%= (r3 last) %> 总共 <%= (r3 total) %> </div>
```

- 07 按 F12 键预览该文件,显示效果如图 14.23 所示。

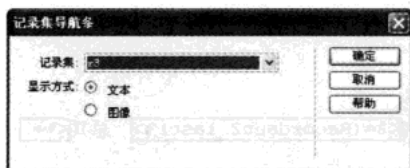


图 14.22 设置【记录集导航条】对话框



图 14.23 图片列表显示效果

14.5.2 欣赏图片

showpic.asp 文件是指定图片放大页面,它主要根据 pic.asp 文件传递过来的 id 编号来查询数据库表 img 中的记录,并把字段 name 绑定到标签的 src 属性上。详细代码如下:

```
">
```

14.6 音乐分享功能实现

音乐分享主要在 muc.asp 文件中实现。其设计步骤如下。

- 01 打开 muc.asp 文件,在【绑定】面板中定义一个记录集,查询 music 数据表中的所有字段信息,设置记录集按 id 字段降序排列,这样就可以把最新发布的音乐排列在前面。其中 SQL 查询字符串代码如下:

```
<%  
Recordset2.Source = "SELECT * FROM music ORDER BY id DESC"  
%>
```

- 02 把该记录集相关字段绑定到页面中单元格所在的行中,其中把 id 字段绑定到 JavaScript 脚本中,当单击“播放”超链接,将打开一个新窗口,播放指定 id 编号的音乐。而当单击“下载”超链接,则根据绑定的 url 字段所记录的 URL 详细信息下载音乐。具体代码

如下:

```
<tr>
  <td width="67%" align="left"><%=Recordset2.Fields.Item("name").Value%></td>
  <td width="16%" align="center"><A HREF="#" target="_parent" onClick="MM_openBr-
Window('play.asp?id=<%=Recordset2.Fields.Item("id").Value %>', '', 'width=300,height=50')
">播放</A></td>
  <td width="17%" align="center"><a href="#<%=Recordset2.Fields.Item("url").
Value%>" target=" _blank">下载</a></td>
</tr>>
```

- 03 选中上面代码, 在【服务器行为】面板中添加“重复区域”服务器行为, 在【重复区域】对话框中设置重复记录为所有。
- 04 增加“显示区域”服务器行为。选中整个重复区域代码, 然后在【服务器行为】面板中增加“如果记录集不为空则显示区域”服务器行为(具体操作可以参阅第 8.4.5 节介绍)。然后在文档的后面增加“暂无音乐”文本, 选中该文本, 为其增加“如果记录集为空则显示区域”服务器行为。
- 05 把光标置于页面底部, 选择【插入】|【数据对象】|【记录集分页】|【记录集导航条】菜单命令, 增加一个完整的记录集导航条。
- 06 同时把记录集 Recordset2 的 Recordset2_first、Recordset2_last 和 Recordset2_total 这 3 个特殊的字段绑定到页面底部, 分别表示记录集分页中当前的第 1 条记录和最后一条记录, 以及整个记录集的所有记录数, 选中<p>和</p>标签内所有代码, 增加“如果记录集不为空则显示区域”服务器行为。代码如下:

```
<% If Not Recordset2.EOF Or Not Recordset2.BOF Then %>
  <p align="center">记录 <%= (Recordset2 first) %>到<%= (Recordset2 last) %> 总共<%=
(Recordset2 total) %></p>
<% End If ' end Not Recordset2.EOF Or NOT Recordset2.BOF %>
```

- 07 按 F12 键预览该文件, 显示效果如图 14.24 所示。



图 14.24 音乐列表显示效果

14.7 留言功能模块实现

留言功能模块实际上就是一个简单的留言板系统, 为了方便网友快速留言。本博客中留言模块主要包含发布留言和显示留言两个基本功能, 共包括 4 个文件, 其中 liuyan.asp 文件

是一个容器文件，负责包含其他 3 个文件；liuyan_form.asp 文件显示一个表单，当提交表单时，将跳转到 liuyanaction.asp 文件；liuyanaction.asp 文件主要负责表单的后台处理，例如，验证验证码是否正确，以及把表单数据插入到数据库中；liuyan_main.asp 文件负责留言的列表显示。主要操作步骤如下。

- 01 在 liuyan_form.asp 文件中设计一个表单，该表单包含 3 个文本框。并为表单增加“验证表单”行为（具体操作方法可以参阅第 6 章介绍），设置表单的 action 属性为“liuyanaction.asp”，即提交表单后将跳转到 liuyanaction.asp 文件，设置 method 方法为“post”。
- 02 打开 liuyanaction.asp 文件，切换到【代码】视图，在代码的顶部添加验证码验证代码和插入记录命令（详细说明可以参阅第 14.4.4 节介绍）。注意，在本页面中插入记录命令所使用的 SQL 代码如下：

```
Command1.CommandText = "INSERT INTO liuyan (name, liuyan, web) VALUES ('" + Replace(Command1_newname, "'", "'') + "','" + Replace(Command1_newliuyan, "'", "'') + "','" + Replace(Command1_newweb, "'", "'') + '")"
```

- 03 在 liuyan_main.asp 文件中，专门用来显示用户留言。主要操作包括定义一个记录集显示图片列表，利用“重复区域”服务器行为显示所有的记录，并利用“显示区域”服务器行为控制记录集没有找到记录 i 时如何处理，最后还需要增加“记录集导航条”服务器行为（详细介绍可以参考第 14.5.1 节内容），整个留言模块的显示效果如图 14.25 所示。



图 14.25 留言功能模块显示效果

14.8 后台管理

一个功能强大的应用系统，其后台功能也一定是非常完善的。甚至在一些大型 Web 应用中，后台管理体系要远远比前台庞杂得多。在本例博客系统中，后台管理主要包括系统管理、日志管理、图片管理、音乐管理、留言管理、公告管理、友情链接管理，如图 14.26 所示。所有后台管理文件全部存放在根目录下 admin 文件夹中。



图 14.26 博客后台管理体系

由于后台管理涉及的内容和文件比较多,因此读者不妨把公共功能抽取出来进行单独设计,这样可以降低程序开发的难度。例如,后台控制台的导航条完全可以提取出来,在一个单独文件中进行设计,然后使用#include 命令包含到所有后台管理文件中。

设计后台控制台导航条。打开 admin_nav.asp,在其中输入下面代码即可:

```
<h2 class="admin_title">后台管理</h2>
<ul class="admin_nav">
  <li><a href="blog.asp">系统设置</a>
  <li><a href="rizhi.asp">日志</a>
  <li><a href="pic.asp">图片</a>
  <li><a href="music.asp">音乐</a>
  <li><a href="liuyan.asp">留言</a>
  <li><a href="gonggao.asp">公告</a>
  <li><a href="link.asp">友情链接</a>
  <li><a href="exit.asp">退出管理</a>
</li>
</ul>
```

为了避免与前台数据库连接文件发生冲突,读者可以在 admin 中新建一个数据库连接文件 Connections/conn.asp,然后输入下面代码,这样读者就可以在 Dreamweaver CS5 中使用各种服务器行为了,否则系统会无法实现可视化添加服务器行为。

```
<%
Dim MM_conn_STRING
MM_conn_STRING = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" & Server.MapPath("../database/blog.mdb")
%>
```



关于后台开发的详细页面和代码,请读者参阅本书光盘示例,这里就不再赘述。

14.9 实践总结

本章重点介绍了博客网站的相关知识和创建博客网站的方法。博客网站实质上是一种复杂的日记本,相对于个人网页而言,它又比较容易建设和维护,因此,它已成为目前网络新宠。博客网站是广大网民心灵的栖息地,自然为广大用户所欢迎。用户可以在网上申请一个免费的博客网站,但如果用户能自己构建一个简单的个人博客发布到网上,会感到更加兴奋和珍惜。

如何实现网站的智能化管理,是 Web 应用中一个重要课题,也是读者在学习 ASP 应用开发中必须面临和思考的问题。网上投票是网站开发中重要的组成模块,本例是在第 13 章调查表实例的基础上进一步深化(第 13 章主要介绍网上调查表的前台设计),重点研究投票管理功能的实现。相信通过本章实例的学习,读者能够更透彻地理解网站智能化管理的实现方法和途径,从而举一反三在独立开发中设计出功能更完善、逻辑更严密的 Web 应用和管理系统。本例将重点介绍账户管理、投票管理以及投票选项管理 3 个大的功能模块,同时还将介绍如何实现后台调查表的生成和调查数据汇总显示。

技术要点

- 投票中的单选与多选混合处理
- 投票标题和选项管理
- 账号管理
- 投票预览和网上投票
- 投票统计和图形化显示

15.1 程序设计分析

在第 13 章中曾经详细介绍过网上调查表的前台设计和一些简单功能的实现方法。相对于前台设计,后台管理功能会复杂许多,它包括 4 大功能模块:投票标题增设、修改和删除功能,投票选项的设置与修改、类型设置,具体选项的增减、修改删除、是否默认等功能,投票统计与结果显示等功能,账号管理等。本实例投票管理系统浏览效果如图 15.1 所示,程序设计流程如图 15.2 所示。

要实现本例功能,读者应该学习和解决下面技术难点。

- 用户管理与登录。该项功能模块可以结合前面实例会更容易理解。
- 标题和选项管理。可利用“更新记录”和“插入记录”服务器行为来实现。
- 投票、统计与显示。读者需要编写简单的代码来实现,同时利用动态属性实现投票的图表化显示。
- 账户管理。可以使用服务器行为或者 Session 变量来实现。



提示

学习本实例之前,读者应该准备如下工作。

启动 Dreamweaver CS5 创建一个动态站点,远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”,【访问】应为“本地/网络”。把本书光盘中的“15”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。

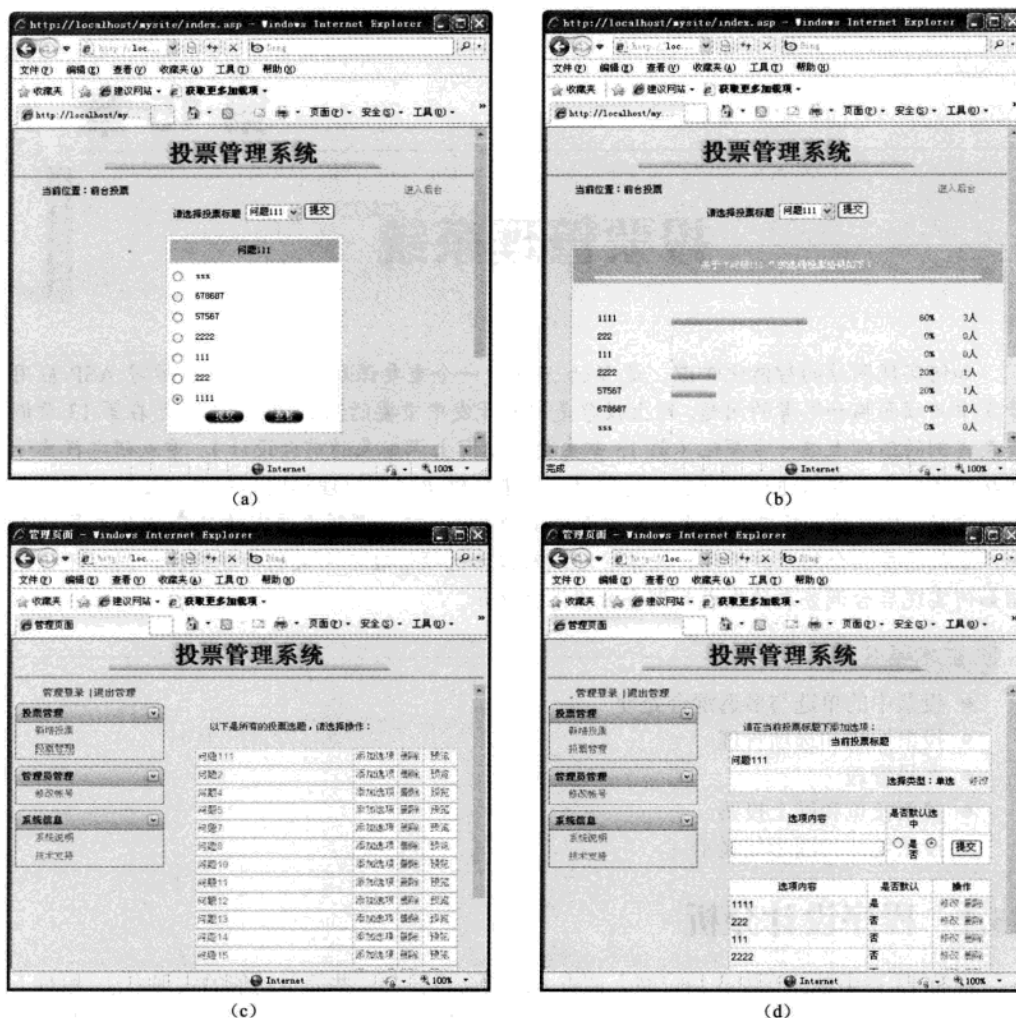


图 15.1 投票管理浏览效果

(a) 前台投票演示; (b) 投票图形化统计; (c) 投票管理; (d) 选项管理

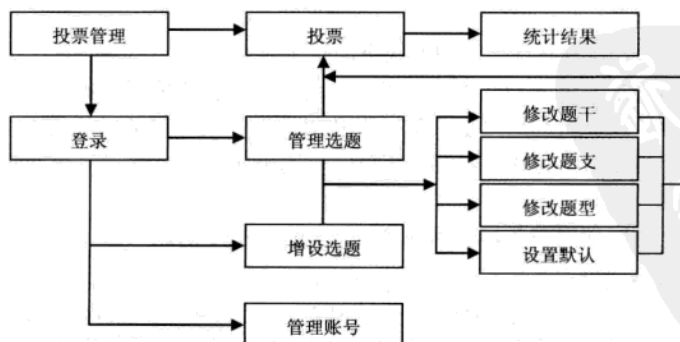


图 15.2 投票管理设计流程图

15.2 数据库设计

本例数据库 data.mdb 包括 choice、manage 和 title 三个表。choice 表用来存储选项，数据结构设计见表 15.1；manage 表用来存储用户信息，数据结构设计见表 15.2；title 表用来存储投票标题信息，数据结构设计见表 15.3。

表 15.1 choice 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
choice	文本	200	否	选项具体内容
pass	文本	30	否	用户密码
extends	数字	长整型	-	该选项所对应的投票标题编号
IsDefault	是/否	-	-	是否是默认勾选项
num	数字	长整型	-	该项得票数

表 15.2 manage 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
names	文本	50	否	用户名
password	文本	50	否	用户密码

表 15.3 title 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
title	文本	200	否	选项题干
current	是/否	-	-	设置当前状态
choice	是/否	-	-	设置题型，包括单投票标题/复投票标题

15.3 页面设计

本应用系统共包含 21 个文件，具体说明见表 15.4。

表 15.4 系统文件列表

文 件 名	说 明	文 件 名	说 明
index.asp	网上投票前台显示	top.asp	后台管理主页面的标题栏
vote.asp	投票生成与处理	bottom.asp	后台管理主页面的状态栏
show.asp	投票统计与显示投票结果	admin.asp	后台登录文件
deal.asp	投票处理与监测文件	edit_manage.asp	账号管理文件
main.asp	后台管理主页面	logout.asp	退出管理文件
menu.asp	后台管理主页面的菜单栏	manage.asp	投票标题管理主页面

续表

文 件 名	说 明	文 件 名	说 明
add.asp	增加投票标题内容	del.asp	删除投票标题文件
edit.asp	投票标题内容与属性修改页面	del_choice.asp	删除选项文件
edit1.asp	投票标题内容与属性修改页面	right.htm	静态提示信息页面
add_choice.asp	增加与修改选项主页面	conn.asp	打开数据库连接文件
edit_choice.asp	选项内容与属性修改页面		

下面就几个主要页面设计进行简单介绍。

15.3.1 导航菜单页面

menu.asp 文件是一个导航菜单页面,用来在后台管理中起导航作用。页面设计效果如图 15.3 所示。



图 15.3 menu.asp 页面设计效果

在该页面中利用动态 HTML 实现折叠式菜单特效,运行效果如图 15.4 所示。

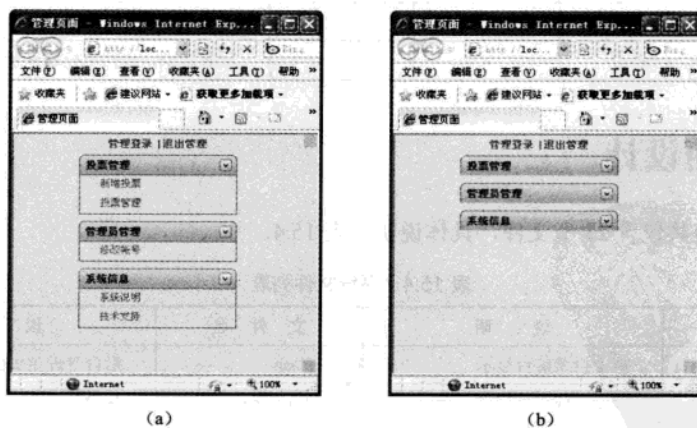


图 15.4 菜单运行效果

(a) 展开菜单; (b) 收缩菜单

该特效主要通过对象的 style.display 属性结合 JavaScript 脚本,来实现隐藏或显示菜单效果,具体代码如下:

```

<SCRIPT language=javascript1.2>
'定义一个函数，设置CSS样式表的display属性在显示和隐藏之间切换
function showsubmenu(sid)
{
    whichEl = eval("submenu" + sid);
    if (whichEl.style.display == "none")
    {
        eval("submenu" + sid + ".style.display=\"\";");
    }
    else
    {
        eval("submenu" + sid + ".style.display=\"none\";");
    }
}
</SCRIPT>
...
<TBODY>
    <TR>
        <TD class=menu title id=menuTitle1 style="CURSOR: hand"
            onclick=showsubmenu(0) onMouseOut="this.className='menu title';"
            width=195 background=images/headtitle.gif bgColor=#cccccc
            height=24><SPAN>投票管理</SPAN> </TD></TR>
        <TR>
            <TD id=submenu0
                style="BORDER-RIGHT: #53876d 1px solid; BORDER-TOP: #53876d 1px solid; BORDER-
                LEFT: #53876d 1px solid; BORDER-BOTTOM: #53876d 1px solid"
                bgColor=#cde7bb>
                <DIV style="WIDTH: 183px; HEIGHT: 20px">
                    <TABLE cellSpacing=0 cellPadding=0 width=135 align=center>
                        <TBODY>
                            <TR height=24>
                                <TD height=23><FONT color=white><a href="add.asp" target="right">新增投
                                票</a></FONT></TD>
                            </TR>
                            <TR height=24>
                                <TD height=23><FONT color=white><A
                                    href="manage.asp"
                                    target=right>投票管理</A></FONT></TD>
                            </TR></TBODY></TABLE>
                        </DIV></TD></TR>
            <TR>
                <TD height=8></TD></TR></TBODY>

```



提示

用户可以利用该实例为模板，增加修改伸缩下拉菜单特效。

15.3.2 标题页面

top.asp 文件是一个标题页面，它主要利用一个 CSS 样式表滤镜实现文字的特效，浏览效果如图 15.5 所示。



图 15.5 标题页面浏览效果

CSS 样式表代码如下:

```
<style type="text/css">
.STYLE1 {
    filter: Shadow(Color=#A0D37F, Direction=-45);
}
</style>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr> <td bgcolor="#CDE7BB">
<div align="center" class="STYLE1">投票管理系统</div>
</td></tr>
</table>
```



提示 注意要把文字包含在<div>层标签中, 否则滤镜特效将失去其作用。

15.3.3 后台管理主页面

main.asp 文件是一个后台管理主页面, 由嵌入式框架结构组成, 设计效果如图 15.6 所示。

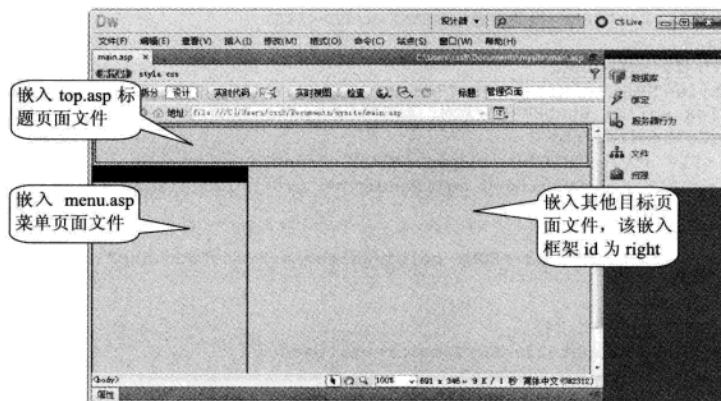


图 15.6 后台管理平台页面设计效果

下面列出该页面的一个嵌入式框架代码, 其他 2 个嵌入框架代码相同, 只修改其中的 id、name 和 src 属性即可。

```
<IFRAME id=right
    style="Z-INDEX: 1; VISIBILITY: inherit; WIDTH: 100%; HEIGHT: 100%"
    name=right src="right.htm" frameBorder=0
    target="_self">
</IFRAME>
```

在页面设计上可以使用表格进行布局。

15.3.4 登录页面

admin.asp 文件是一个登录页面, 设计如图 15.7 所示。该页面中包含两个文本框和一个图像域。图像域的作用和提交按钮功能相同, 用来提交表单。使用图像域可以很容易设计出符合页面设计效果的表单。

在登录页面中包含一个简单的表单, 属性设置见表 18.5。

表 15.5 登录表单属性设置列表

对 象	名 称	类 型	设置说明
表单	form1	表单	【方法】设置为“POST”
文本框	names	单行	【字符宽度】设置为 27
文件域	password	密码	【字符宽度】设置为 14
按钮	imageField	图像域	【源文件】为“images/button.gif”



图 15.7 登录页面设计效果

15.3.5 增加选项页面

add_choice.asp 文件主要包含 3 个独立的表格。在表格边线设计上利用表格属性的组合设置实现细边框效果，如图 15.8 所示。

选中表格，在属性面板中设置“背景颜色”为“#A0D37F”，该颜色将显示为表格边线颜色，同时设置“边框”为 0、“间距”为 1，单元格之间的间距值将成为表格边框的宽度，如果在扩展表格模式下（按 F6 快捷键可快速切入）会看到单元格间距所显示的边框效果，如图 15.9 所示。

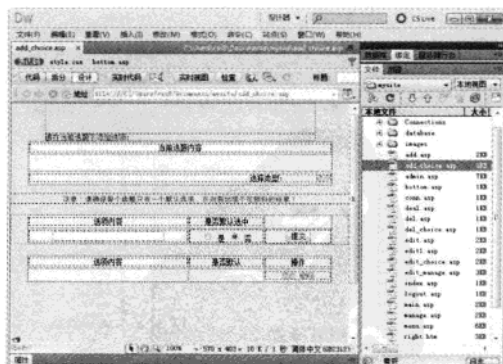


图 15.8 细边框表格浏览效果

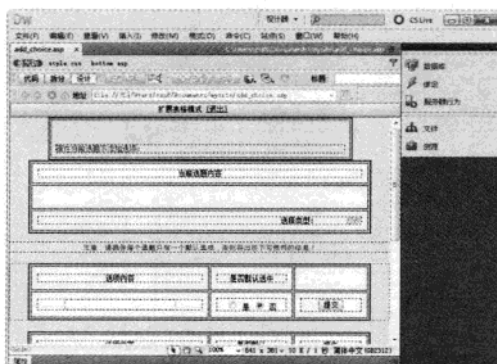


图 15.9 扩展表格模式下设计效果

15.3.6 其他页面

上面介绍了本系统中技术较复杂的几个页面设计，其他页面设计相对简单。下面仅列出几个表单页面的属性设置。

edit.asp 页面设计如图 15.10 所示，其表单属性设置见表 15.6。

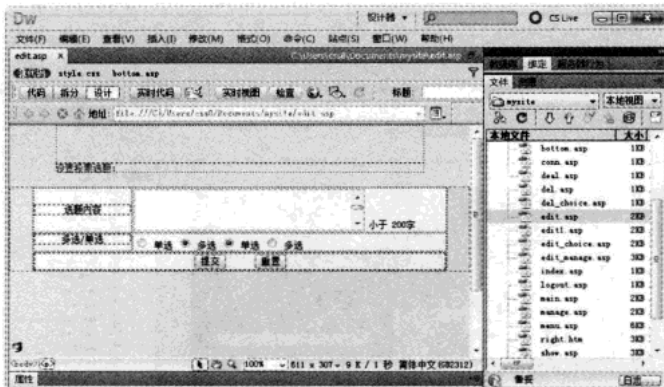


图 15.10 edit.asp 页面设计效果

表 15.6 投票标题编辑表单属性设置列表

对 象	名 称	类 型	设置说明
表单	form1	表单	“方法”设置为“默认”
文本框	topic	多行	“字符宽度”设置为 40，“行数”为 4，“换行”为“默认”
文件域	password	密码	“字符宽度”设置为 14
选择框	choice	单选按钮	设置两对重复的单选按钮，设置“初始状态”为 2 个勾选
按钮	Submit	提交	“值”为“提交”，“动作”为“提交表单”
按钮	Submit2	重置	“值”为“重置”，“动作”为“重置表单”

edit_manage.asp 文件是一个账号管理页面，页面设计如图 15.11 所示。表单属性设置基本相同。

vote.asp 文件是一个投票生成页面，设计如图 15.12 所示。在该页面中插入了两个个性化的按钮，其中“提交”按钮是一个图片域，执行提交功能；“查看”按钮是一个图片，在该图片上附加一行脚本，实现页面跳转。代码如下：



图 15.11 edit_manage.asp 页面设计效果

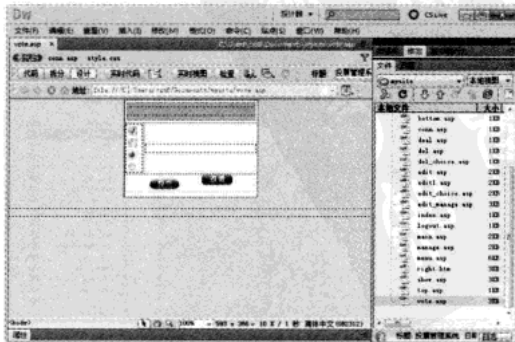


图 15.12 vote.asp 页面设计效果

```
<a href=# onclick=javascript:openwin('show.asp?id=<%=request("id")%>','600','400')>
<img src=images/button2.gif width=50 height=20 border=0></a>
```

15.4 投票管理功能实现

下面就按照账户管理、投票标题管理和投票选项管理 3 个关键模块展开介绍, 实现常规的增加、删除和修改管理。

15.4.1 用户登录

本系统操作的第 1 步就是要先登录, 登录功能的实现方法有多种, 借用服务器行为能快速完成用户登录功能。下面着重介绍用代码直接灵活地实现用户登录, 其设计步骤如下。

- 01** 新建一个数据库连接文件 conn.asp, 输入下面代码, 定义数据库连接对象 Connection, 然后利用数据库连接字符串打开数据库连接。

```
<!--#include file="Connections/conn.asp"-->
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.open MM_conn_STRING
%>
```

- 02** 打开 admin.asp 文件, 导入数据库连接文件 conn.asp, 代码如下:

```
<!--#include file="conn.asp"-->
```

- 03** 输入用户身份验证代码, 验证用户输入信息是否合法, 并及时给出提示, 如果正确则保存用户信息到 Session("login_id")变量中, 同时设置 session("manage")条件变量为 1, 最后跳转到 main.asp 文件。

```
<% '获取表单中输入的用户名和密码, 并用ASP函数trim()去除两边空格
names=trim(request("names"))
pass=trim(request("password"))
'判断, 当提交表单时, 则定义记录集读取数据库中的用户信息
if request("actions")="1" then
    set rs=server.CreateObject("adodb.recordset")
    sql="select * from manage where id=1"
    rs.open sql,conn
    Mname=rs("names")
    Mpass=rs("password")
    '判断输入的用户信息与数据库中的信息是否相同, 不同则提示错误并返回
    if names<>Mname then
        response.write "<script language='javascript'>alert('账号错误!');history.go(-1);
</script>"
    elseif pass<>Mpass then
        response.write "<script language='javascript'>alert('密码错误!');history.go(-1);
</script>"
    else '相同则保存用户信息到session变量中, 并跳转到manage.asp文件
        session("manage")=1
        '获取用户名变量
        session("login_id") = rs("id")
        response.Redirect "main.asp"
        rs.close
        set rs=nothing
    end if
else '如果不是本文件内的表单提交进入, 则显示表单域, 让用户输入登录信息
%>
```

- 04** 在文档的底部输入<%end if%>语句, 结束条件语句。


15.4.2 账户管理

账户管理功能实现的具体操作步骤如下。

01 打开 `edit_manage.asp` 文件，在该文件中实现对用户账号的管理，主要功能包括对账号的修改重设，设计效果如图 15.13 所示。

02 上面页面由 2 个表单域构成，分别用来完成更新用户名和更新密码。因此，也可以定义两个记录集分别执行“更新记录”服务器行为操作。

在菜单中选择【窗口】|【绑定】命令，

打开【绑定】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【记录集（查询）】命令，打开【记录集】对话框，分别定义两个记录集。【记录集】对话框具体设置如图 15.14 所示。

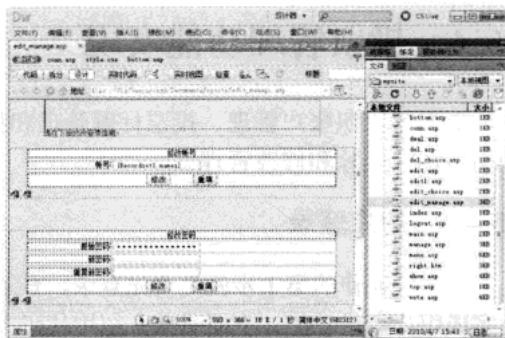


图 15.13 `edit_manage.asp` 页面设计效果

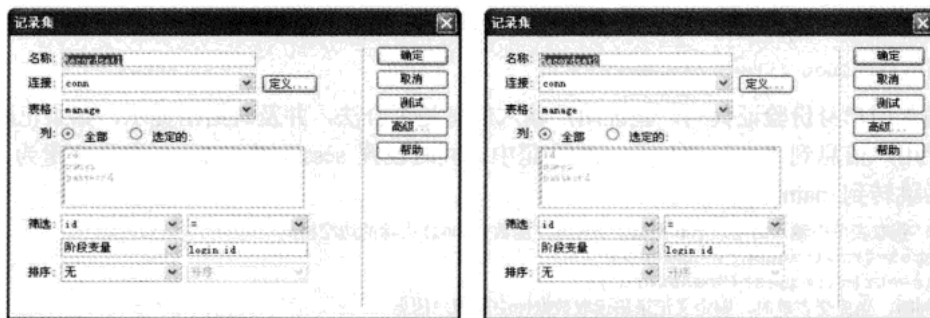


图 15.14 设置【记录集】对话框

上面两个记录集设置相同，根据用户登录时所保存的 Session 变量（即阶段变量）筛选出该用户的记录。

03 把记录集 Recordset1 中的 `names` 字段绑定在 `form1` 表单中的 `users` 文本框上，如图 15.15 所示。把记录集 Recordset2 中的 `password` 字段绑定在 `form2` 表单中的 `pass1` 文本框上，如图 15.16 所示。



图 15.15 绑定用户名字段

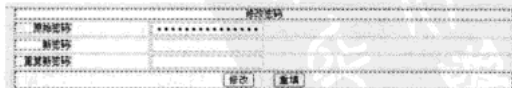



图 15.16 绑定密码字段

04 定义更新记录。打开【服务器行为】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【更新记录】命令，打开【更新记录】对话框，设置如图 15.17 所示。

在【连接】下拉列表中选择前面定义好的数据源，如 `conn`；在【要更新的表格】下拉列表中选择要更新的数据表，如 `manage`；在【选取记录自】下拉列表中选择定义的记录集，如 `Recordset1`；在【唯一键列】下拉列表中选择被定义为主键的字段，如 `id`；在【获取值自】

下拉列表中选择表单名称, 如 form1; 在【表单元素】列表框中设置每个表单对象所要对应的数据表中的字段。

05 以同样的方式更新表单 form2, 设置如图 15.18 所示。

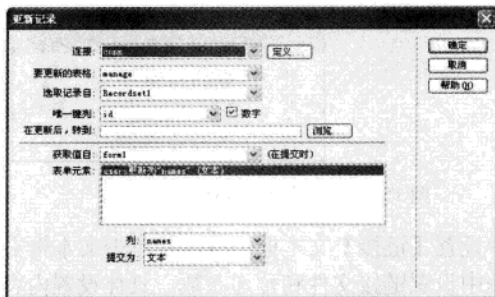


图 15.17 更新记录集 Recordset1

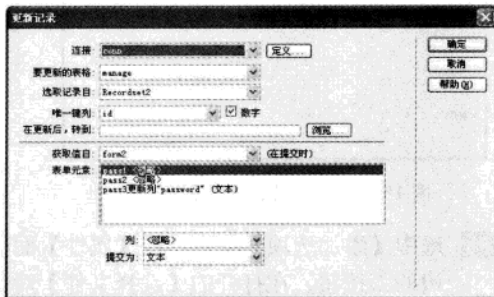


图 15.18 更新记录集 Recordset2

06 为了避免用户在修改密码时出现两次输入不一致现象, 可在表单 form2 中插入一个隐藏域, 代码如下:

```
<input type="hidden" name="bijiao" id="bijiao" value="pass" />
```

07 切换到【代码】视图, 找到更新表单 form2 的记录服务器行为代码, 在其前面输入下面方框内代码:

<% 先获取隐藏域值, 由该值判断当前提交行为是否是表单form2提交, 然后获取两次密码输入, 比较是否一致, 如果一致则执行中间的“更新记录”服务器行为脚本

```
pass = Request.Form("bijiao")
pass22 = Request.Form("pass2")
pass33 = Request.Form("pass3")
```

```
if (pass="pass") then
    if pass22<>pass33 then
        Response.write("<script language='javascript'>alert('密码输入不一致, 请重新输入!');history.go(-1);</script>")
    else
```

```
%>
中间省略更新form2表单的“更新记录”服务器行为代码
<%
```

```
end if
end if
```

```
%>
```

15.4.3 管理投票标题

投票标题管理主要包括投票标题的增加、修改、删除和属性设置。实现该模块需要多个文件协同工作, 主要包括 manage.asp 文件管理投票标题, add.asp 文件增加投票标题, del.asp 文件删除投票标题, edit.asp 和 edit1.asp 文件修改投票标题。下面重点介绍投票标题的管理功能, 具体操作步骤如下。

01 打开 manage.asp 文件, 在该页面中主要实现记录集定义与显示。先定义一个记录集, 查询所有的投票标题记录, 设置【记录集】对话框如图 15.19 所示。


02 把 Recordset1 记录集中的 title 字段插入页面, 然后选中插入所在行, 定义一个重复区域服务器行为, 在页面中显示所有记录, 如图 15.20 所示。



图 15.19 设置【记录集】对话框



图 15.20 设置记录重复显示区域

- 03** 选中【添加选项】文本，在【属性】面板中设置【链接】文本框，方法是单击文本框右边的按钮，在打开的【选择文件】对话框中设置链接文件和传递参数。具体设置内容如下：

```
add_choice.asp?id=<%= (Recordset1.Fields.Item("id").Value) %>
```

- 04** 以同样的方法设置【删除】文本的链接，设置为“del.asp?id=<%= (Recordset1.Fields.Item("id").Value) %>”，设置“预览”文本的链接为“vote.asp?id=<%= (Recordset1.Fields.Item("id").Value) %>”。

- 05** 切换到【代码】视图，在代码顶部输入下面代码，检测用户是否登录，如果没有登录则强制返回并要求登录。


```
<% 判断是否登录，否则返回到admin.asp文件
if Session("manage") <> 1 then
    Response.Redirect("admin.asp")
end if
%>
```

15.4.4 增加投票标题

投票标题增加功能主要在 add.asp 文件中完成，具体操作步骤如下。

- 01** 打开 add.asp 文件，在该文件中实现增加投票标题功能。增加投票标题主要利用插入记录服务器行为。在使用服务器行为之前，先插入 2 个隐藏域对象，具体方法是把光标置于表单中，选择【插入】|【表单】|【隐藏域】命令，然后在【属性】面板中设置名称和要传递的值。代码如下：

```
<input type="hidden" name="current_id" value=titles>
<input type="hidden" name="current1" value= <%=cu%> >
```

- 02** 在【服务器行为】面板。单击面板左上角的按钮，在弹出的下拉菜单中选择【插入记录】命令，打开【插入记录】对话框，设置如图 15.21 所示。

在【连接】下拉列表中选择前面定义好的数据源 conn；在【插入到表格】下拉列表中选择要插入的数据表 title；在【插入后，转到】文本框中输入“manage.asp”，即插入后跳转到后台管理页面查看结果。在【表单元素】中设

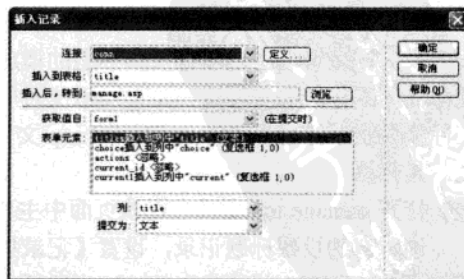


图 15.21 设置【插入记录】对话框

置各个表单对象所对应的数据表中的字段,在设置复选框时,要注意在【提交为】下拉列表中选择“复选框 1,0”。



注意在设置表单中单选按钮组时,要设 Value 的值分别为“True”和“false”,否则将无法写入 Access 数据库中的【是/否】类型的数据列。

03 切换到【代码】视图,在代码顶部输入下面代码,实现对用户输入的值进行验证,并设置变量 cu 的值,该值将作为隐藏域的值写入到数据库。

```
<%
if Session("manage") <> 1 then
    Response.Redirect("admin.asp")
end if
if Request("current")=1 then
    cu="true"
else
    cu=""
end if
If (CStr(Request("MM_insert")) = "form1") Then
    title=Request("titles")
    if title = "" or len(title) > 200 then
        response.write "<script language='javascript'>alert('主题内容不能为空或大于指定字数!');window.location.href='manage.asp';</script>"
    end if
    Response.End()
    '获取、更新用户名变量
    Response.cookies("name") = Request.Form("titles")
end if
%>
```

15.4.5 编辑投票标题

投票标题编辑功能主要包括投票标题内容的修改和投票标题类型的设置,其设计步骤如下。

01 打开 edit.asp 文件,在该页面中实现对选定投票标题进行修改编辑操作。首先打开【绑定】面板,单击左上角的按钮,在弹出的下拉菜单中选择【记录集 (查询)】命令,在打开的【记录集】对话框中设置查询条件,如图 15.22 所示。

02 把记录集 Recordset1 中的 title 字段绑定到文本框上,如图 15.23 所示。

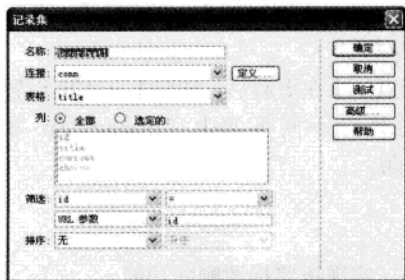


图 15.22 设置【记录集】对话框

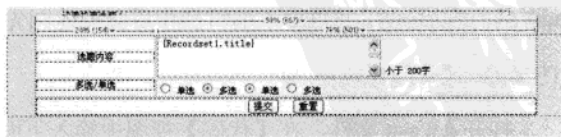


图 15.23 绑定动态文本框初始值

03 下面来实现选择不同类型的投票标题,单选按钮组会动态显示该记录类型。方法是切换到【代码】视图,找到单选按钮组代码段,然后在其中加入一段条件语句。代码如下:

```

<% '动态设置单选按钮组的初始值
danduo = (Recordset1.Fields.Item("choice").Value)
if danduo then
%>
    <input name="choice" type="radio" value="" >
    单选
    <input type="radio" name="choice" value="true" checked>
    多选
    <%
else
%>
    <input name="choice" type="radio" value="" checked>
    单选
    <input type="radio" name="choice" value="true" >
    多选
    <%
end if
%>

```

- 04** 当修改单选类型投票标题或者多选类型投票标题时，会显示相应选中状态，如图 15.24 所示。

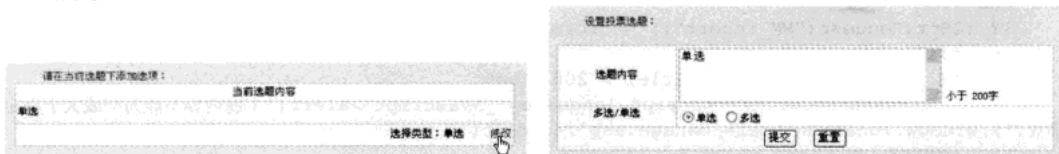



图 15.24 修改一个单选类型投票标题后显示状态

- 05** 打开【服务器行为】面板。单击面板左上角的  按钮，在弹出的下拉菜单中选择【更新记录】命令，打开【更新记录】对话框，设置如图 15.25 所示。

在【连接】下拉列表中选择前面定义好的数据源 conn；在【要更新的表格】下拉列表中选择要更新的数据表 title；在【选取记录自】下拉列表中选择定义的记录集 Recordset1；在【唯一键列】下拉列表中选择被定义为主键的字段 id；在【在更新后，转到】文本框中输入 add_choice.asp；在【获取值自】下拉列表中选择表单 form1；在【表单元



图 15.25 设置更新记录对话框

素】列表框中设置每个表单对象所要对应的数据表中的字段。

editl.asp 文件的所有设置与 edit.asp 文件相同，不同点是【在更新后，转到】文本框设置不同，在 editl.asp 文件中输入为“manage.asp”，这样在不同页面中（manage.asp 和 add_choice.asp）中实现修改编辑操作后会能返回先前页面。

15.4.6 删除投票标题

打开 del.asp 文件，在【代码】视图输入下面代码，实现投票标题删除功能。

```

<!--#include file="conn.asp"-->
<% '判断用户是否登录
if session("manage") <> "1" then
    response.Redirect "admin.asp"
end if

```

```

%>
<% '定义一个记录集删除title表中字段id为传递过来的参数
set rs=server.CreateObject("adodb.recordset")
sql="delete * from title where id=&request("id")
rs.open sql,connstr
'定义一个记录集删除choice表中字段extends为传递过来的参数
set rs=server.CreateObject("adodb.recordset")
sql="delete * from choice where extends=&request("id")
rs.open sql,connstr
response.write "<script language='javascript'>alert('删除成功!');window. location.
href='manage.asp';</script>"
%>

```

15.4.7 增加选项

增加了投票标题后,在后台管理主页面中单击该投票标题后的“增加选项”链接文本,如图 15.26 所示,这样就会跳转到 add_choice.asp 文件,此时就可以在该页面中增加该问题的选项,如图 15.27 所示。



图 15.26 选择要增加选项的投票标题



图 15.27 增加选项

实现上述功能的具体操作步骤如下。

- 01 打开 add_choice.asp 文件,在【绑定】面板中定义一个记录集 Recordset1,根据其他页面传递过来的投票标题编号筛选出该投票标题的记录,具体设置如图 15.28 所示。
- 02 将该记录集 Recordset1 中的 title 字段插入到页面中的【当前投票标题内容】表格中,如图 15.29 所示。

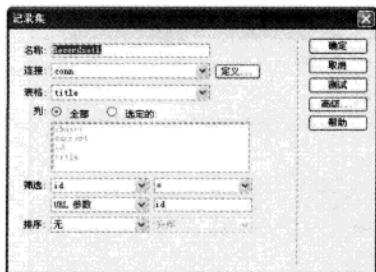


图 15.28 定义记录集 Recordset1

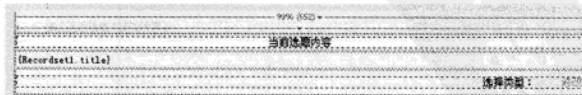


图 15.29 插入记录集 Recordset1 中的 title 字段

08 把记录集 Recordset2 中的 choice 字段插入页面中, 选中插入字段所在行, 定义一个【重复区域】服务器行为, 如图 15.33 所示。

09 选中“修改”文本, 在【属性】面板的【链接】文本框中输入下面代码传递投票标题编号:

```
edit_choice.asp?idc=<%= (Recordset2.Fields.Item("id").Value) %>&id=<%= (Recordset2.Fields.Item("extends").Value) %>
```

10 选中“删除”文本, 在【属性】面板的【链接】文本框中输入下面代码:

```
del_choice.asp?idc=<%= (Recordset2.Fields.Item("id").Value) %>&id=<%= (Recordset2.Fields.Item("extends").Value) %>
```

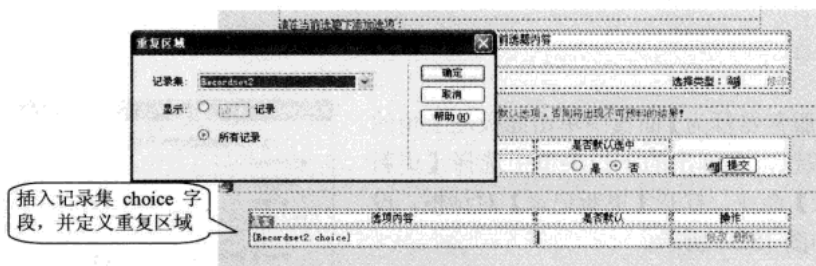


图 15.33 插入记录集

11 切换到【代码】视图, 找到【是否默认】列代码, 然后输入下面代码实现显示选项是否默认被选中状态:

```
<% '第1个条件先判断记录集Recordset2是否为空, 为空则跳过该行不显示
if not Recordset2.EOF then
    mr = (Recordset2.Fields.Item("IsDefault").Value)
    if mr then '第2个条件判断该选项是否为默认选中, 是则赋予变量com值为“是”, 否则为“否”
        com="是"
    else
        com="否"
    end if
end if
%>
<tr>
    <td height="21" valign="middle" bgcolor="#D7F2FF"><%= (Recordset2.Fields.Item("choice").Value) %></td>
    <td valign="middle" bgcolor="#D7F2FF"><%= com %></td>
    <td valign="middle" bgcolor="#D7F2FF"><div align="center"><a href="edit_choice.asp?idc=<%= (Recordset2.Fields.Item("id").Value) %>&id=<%= (Recordset2.Fields.Item("extends").Value) %>">修改</a> <a href="del_choice.asp?idc=<%= (Recordset2.Fields.Item("id").Value) %>&id=<%= (Recordset2.Fields.Item("extends").Value) %>" onClick="return test()">删除</a></div></td>
</tr>
```

15.4.8 编辑选项

编辑选项与编辑投票标题在表单设计和功能实现上都非常相似, 其设计步骤如下。

01 打开 edit_choice.asp 文件, 首先定义一个记录集, 根据其他页面传递过来的选项编号筛选出该选项的记录, 设置如图 15.34 所示。

02 把该记录集中 choice 字段绑定到页面文本框中, 然后切换到【代码】视图, 在其后面单

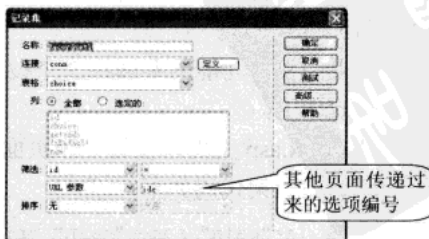


图 15.34 定义记录集

元格中输入下面代码:

```
<%
if (Recordset1.Fields.Item("IsDefault").Value) then
%>
☐

```


- 03** 在【服务器行为】面板中单击面板左上角的  按钮, 在弹出的下拉菜单中选择【更新记录】命令, 打开【更新记录】对话框, 设置如图 15.35 所示。



图 15.35 设置【更新记录】对话框

15.4.9 删除选项

打开 del_choice.asp 文件, 切换到【代码】视图, 在其中输入下面代码:

```
<!--#include file="conn.asp"-->
<% 判断是否登录
if session("manage")<>"1" then
    response.Redirect "admin.asp"
end if%>
<% 定义记录集, 并删除其中的选项
cid=request("idc")
id=request("id")
set rs=server.CreateObject("adodb.recordset")
sql="delete * from choice where id=&cid"
rs.open sql,connstr
response.write "<script language='javascript'>alert('删除成功!');window.location.href=
'add_choice.asp?id=&id&';</script>"
%>
```

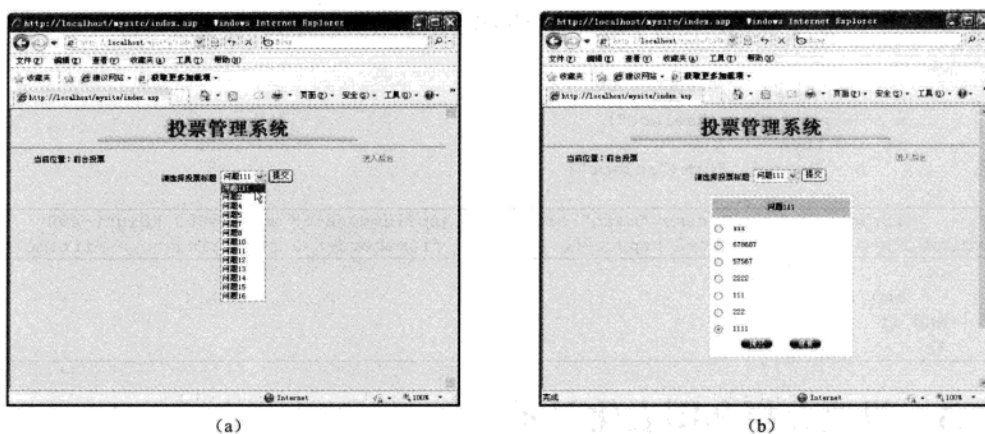
15.4.10 选择投票

用户在后台管理中设计的所有投票主题在 index.asp 文件中都可以预览并进行投票, 浏览效果如图 15.36 所示。

实现上面显示功能需要解决 3 个技术问题: 动态下拉菜单、动态初始值和参数传递, 具体操作步骤如下。

- 01** 打开 index.asp 文件, 设计一个页面, 包含一个下拉菜单和一个嵌入框架, 如图 15.37 所示。其中框架代码如下:

```
<iframe id="main" name="main" src="vote.asp?id=<%=n%>" width=400 height=400
marginwidth=0 marginheight=0 hspace=0 vspace=0 frameborder=0 scrolling=no ></iframe>
```



(a)

(b)

图 15.36 选择投票

(a) 选择主题; (b) 浏览投票

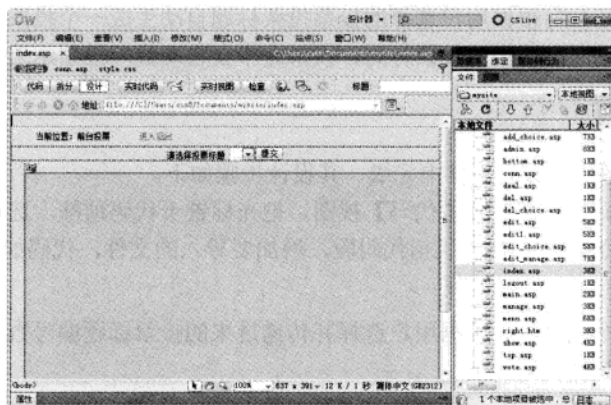


图 15.37 设计页面效果

15

02 定义一个记录集，筛选出所有的投票标题，具体设置如图 15.38 所示。

03 绑定下拉菜单。首先选中下拉菜单，在【属性】面板中单击【动态】按钮，打开【动态列表/菜单】对话框，设置如图 15.39 所示。设置下拉菜单要传递的值为投票标题记录集的编号，要显示的下拉标签为投票标题的题目。



图 15.38 定义记录集



图 15.39 绑定记录集

04 切换到【代码】视图，在其中输入下面代码实现动态初始值和动态参数传递：

```
<%
  if not Recordset1.EOF then
    n = Request.Form("select")
    if n>0 then
      n = Request.Form("select")
    %>
    <iframe id="main" name="main" src="vote.asp?id=<%=n%>" width=600 height=400
marginwidth=0 marginheight=0 hspace=0o vspace=0 frameborder=0 scrolling=no ></iframe>
    <%
      end if
    end if
  %>
```

15.5 投票分析和图表化显示

投票管理是本例的重点，但如何汇总和分析用户的投票，并进行形象化显示则是本例的难点。下面重点介绍如何根据用户选择的投票投票标题自动生成一个投票页面，并允许用户在其中进行投票，同时监视用户投票行为，投票完毕会自动显示投票结果。

15.5.1 生成投票

生成投票主要在 vote.asp 文件中完成，其设计步骤如下。

01 打开 vote.asp 文件，切换到【代码】视图，把光标置于代码顶部，选择【插入】|【ASP 对象】|【包括】命令，插入包括代码段，增加要导入的文件，代码如下：

```
<!--#include file="conn.asp"-->
```

02 在下面定义一个记录集，根据用户选择并传递过来的投票标题编号查询指定投票标题的选项记录集，具体代码如下：

```
<!--#include file="conn.asp"-->
<%
set rs=server.CreateObject("adodb.recordset")
sql="select * from title where id=" & request.QueryString("id")
rs.open sql,conn
if rs("choice")=true then
  types=true
else
  types=false
end if
%>
```

03 把投票标题插入标题单元格内，然后再定义一个记录集，筛选出在 choice 表中该投票标题编号下的所有选项，代码如下：

```
<%
rs.close
sql="select * from choice where extends=" & request("id")
rs.open sql,conn
if rs.eof and rs.bof then
  response.write "您还没有添加投票选项!"
else
  i=1
  do while not rs.eof
  %>
```

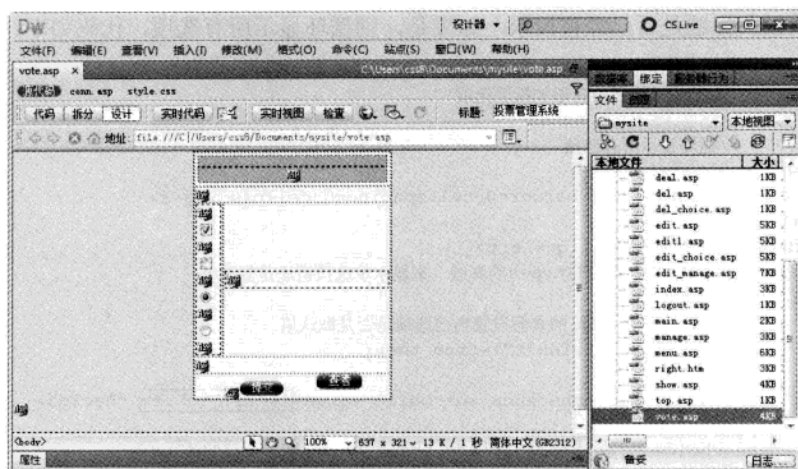



图 15.40 vote.asp 页面效果

15.5.2 投票监测

当在 vote.asp 页面内的表单中选择相应选项后,单击【提交】按钮,就会跳转到 deal.asp 文件,实现对用户的监督,主要包括是否重复投票,或者空投票。

打开 deal.asp 文件,在【代码】视图中输入下面代码:

```
<!--#include file="conn.asp"-->
<%判断是否投票
if isempty(session("go")) then
'判断是否空投票
choice=request("choice")
if isempty(choice) then
response.write"<script language='javascript'>alert('您还没选择投票内容!');history.go(-1);</script>"
else
'定义记录集,更新choice表中的投票数
set rs=server.CreateObject("adodb.recordset")
sql="update choice set num=num+1 where id in ('&choice&')"
rs.open sql,connstr
'设置session("go")变量为1,表示当前登录用户已经投票了
session("go")=1
response.write "<script language='javascript'>alert('投票成功');</script>"
response.write "<meta http-equiv='refresh' content='1;url=show.asp?id="&request("id")&">"
end if
else
response.write "<script language='javascript'>alert('您已经投过票了!');history.go(-1);</script>"
end if
%>
```

15.5.3 显示投票结果

图形化显示投票结果主要利用<td>标签单元格的 width 属性动态显示其宽度值,同时设置该单元格的背景图像为事先制作好的图形效果截面,这样就可以实现动态图形化显示用户投票结果,如图 15.41 所示。

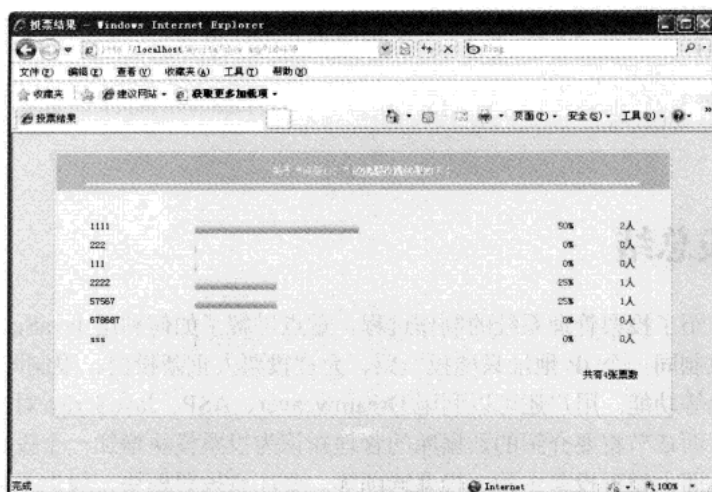


图 15.41 图形化动态投票显示效果

实现该功能的代码如下。

01 定义两个记录集，查询指定投票标题编号的投票标题内容和所有选项内容。代码如下：

```
<!--#include file="conn.asp"-->
<% 查询指定编号的投票投票标题记录信息
set rsTp=server.CreateObject("adodb.recordset")
sqlTp="select * from title where id=&request("id")
rsTp.open sqlTp,connstr
%>
<% 查询指定编号的投票投票标题所属的所有选项记录信息
set rs=server.CreateObject("adodb.recordset")
sql="select * from choice where extends=&request("id")&" order by id desc"
rs.open sql,connstr%>
<% 查询并计算投票的总数
set rsTotle=server.CreateObject("adodb.recordset")
sqlTotle="select SUM(num) from choice where extends=&request("id")
rsTotle.open sqlTotle,connstr
'判断当前投票标题是否已经被投票
if rsTotle(0)=0 then
    response.write "目前还没有投票数!"
else%>
```

02 把记录集相应字段绑定在单元格宽度属性上，代码如下：

```
<td height="179">
<%do while not rs.eof%>
<table width="100%" border="0" cellspacing="4" cellpadding="0"><tr>
<td width="19%"><%=rs("choice")%></td>
<td width="61%">%" height="15%"></td>
<td width="9%"><div align="right"><span class="style4">
<%=int(100*rs("num")/rsTotle(0))%></span>%</div></td>
<td width="11%"><div align="right"><span class="style4"><%=rs("num")%></span>人
</div></td>
</tr></table>
<%rs.movenext
loop
rs.close
set rs=nothing%>
</td>
```

03 在页面代码的末尾输入下面代码，结束记录集和条件语句。

```
<%  
rsTotle.close  
set rsTotle=nothing  
end if  
%>
```

15.6 实践总结

本章详细介绍了投票管理系统的制作过程，重点讲解了如何利用 JavaScript 和 ASP 的知识，增加诸如限制同一个 IP 地址只能投一票、允许投票人重新投票、限制同一个 IP 地址的投票的时间间隔等功能。用户还可以利用 Dreamweaver、ASP、JavaScript 对本系统进一步完善，例如，以后面章节将要介绍的数据库的管理知识为投票模块增加一个投票管理系统，让管理员可以在页面上设置投票主题、投票选项等。要注意投票模块一般也不会单独存在，而是与其他页面相结合的。



新闻发布系统是信息发布系统的一个特定分支,是将某些需要经常变动的信息,例如,新闻、新产品发布和业界动态等更新信息集中管理,并根据信息的性质和特征进行分类,最后系统化、标准化发布到网站上的一种网站应用程序。在后台管理员可以通过一个操作简单的界面把新闻信息加入数据库,然后通过已有的网页模板格式与审核流程发布到网站上显示出来。

新闻信息发布与管理能够减轻网站更新维护的工作量,利用后台数据库,管理员可以将网站的更新维护工作简化到只需录入文字和上传图片,从而使网站的更新速度大大缩短。在大型专业的新闻站点,例如,新浪的新闻中心、百度、Google 新闻咨询频道等,新闻的更新速度已经缩短到五分钟,大大加快了信息的传播速度,也吸引了更多的长期用户群,时时保持网站的活动力和影响力。

本章将介绍新闻发布系统的设计思路和实现技巧。新闻站点主要功能是汇集众多新闻信息并能够快速发布,方便网民浏览查阅。新闻站点的性质决定了新闻站点的管理显得非常重要。因此,本章在演示前台新闻发布方法之后,还要重点探讨新闻管理的功能实现。

技术要点

- 新闻站点设计思路和方法
- 信息发布的操作步骤
- 栏目导航技术
- 信息分类管理
- 多级目录技术

16.1 程序设计分析

新闻发布与管理系统是网上最实用也是最经典的一类功能模块,它可以包括狭义的新闻板块、文章管理、综合资料管理等不同内容形式。新闻发布属于前台性质,它的主要功能是利用网页对大量文章信息进行分类组织和呈现,方便浏览者查看、阅读。新闻管理属于后台性质,它主要包括类别管理、信息管理和用户管理。

类别管理为整个系统提供了灵活高效的操作性,它使网站管理员可随时调整各类别(包括总类别下的二级类别、新闻专题),可以根据需要增加、修改或删除类别。这对于网站上新闻信息的分类调整以及网站发展规划中第二步的实现具有很大的作用,可以极大地减少网站二次开发的工作量。类别管理提供的具体功能包括:增加、修改、删除新闻类别和专题的功

能；更改类别顺序以确定新闻类别和专题在网站页面上显示的排序的功能。

信息管理主要针对网站内容的更新与维护，提供在后台输入、查询、修改、删除各种新闻类别和专题中的具体信息的功能，选择本信息是否出现在栏目的首页、网站的首页等一系列完善的信息管理功能。具体包括：增加、修改、删除各个栏目的信息（包括文字与图片）的功能；修改信息状态以确定信息是否出现在类别首页或网站首页的功能。

用户管理包括系统用户和管理员。系统用户具有管理网站的权限（即可进入后台管理界面）的用户。其下又分为管理员与一般操作员。管理员具有管理系统管理员的功能，可增加、删除系统管理员账号，分配与修改一般操作员的权限，并拥有一般操作员的所有权限。

本实例简单演示了新闻发布和管理的基本功能，重点介绍如何快捷地管理和发布新闻，管理新闻类别。演示效果如图 16.1 所示，程序设计流程如图 16.2 所示。

要实现本例功能，读者需要学习和掌握下面技术难点。

- 分类和分级发布新闻。
- 新闻多功能查询。
- 登录管理。
- 栏目的增加、删除和修改。
- 文章的增加、删除和修改。



新闻发布与管理主页



新闻标题分类列表



浏览具体新闻



管理新闻菜单

图 16.1 新闻发布与管理浏览效果

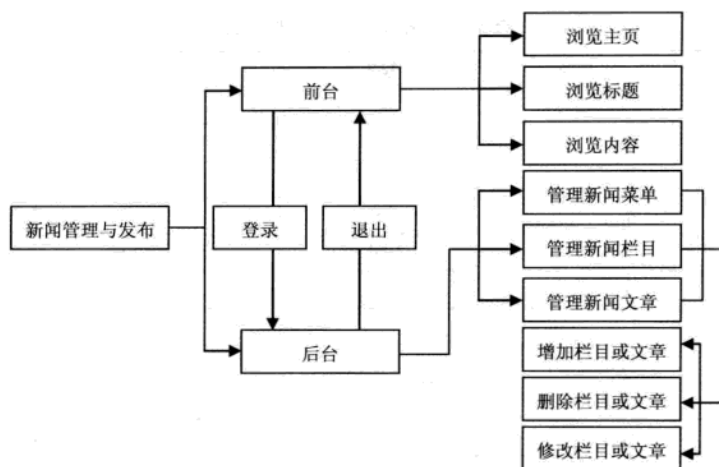


图 16.2 新闻发布与管理设计流程图



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“16”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

16.2 数据库设计

本例数据库 db.mdb 包括 3 个表，其中，user 表用来储存管理信息，class 表用来储存分类信息，message 表用来储存具体的新闻文章。

16.2.1 class 表

class 表用来记录新闻分类信息，用于文章的分类和组织，数据结构设计见表 16.1。

表 16.1 class 表字段列表

字 段	类型	字段大小	必填字段	说 明
id	自动编号	-	-	分类信息编号
class_name	文本	50	否	栏目名称
sup_class_id	数字	长整型	否	上一级栏目编号
big_class_id	数字	长整型	否	菜单栏目 ID
navigation	文本	255	否	栏目导航
orders	数字	长整型	否	栏目序号

16.2.2 message 表

message 表用来记录新闻分类信息，用于文章的分类和组织，数据结构设计见表 16.2。

表 16.2 message 表字段列表

字段	类型	字段大小	必填字段	说明
id	自动编号	-	-	新闻文章编号
title	文本	200	否	文章标题
new	数字	长整型	否	最新文章
content	备注	-	-	文章内容
author	文本	50	否	文章作者
add_person	文本	50	否	添加人
add_date	日期/时间	-	-	添加日期
add_time	日期/时间	-	-	添加时间
small_class_id	数字	长整型	否	栏目 ID
big_class_id	数字	长整型	否	菜单列表类 ID
clicks	数字	长整型	否	点击次数

16.3 页面设计

本实例共有 21 个文件，各个文件的具体功能说明见表 16.3。

表 16.3 新闻发布与管理系统文件列表

文 件 名	说 明
admin_addbigclass.asp	新闻菜单栏目管理页面
admin_addfile.asp	新闻文章管理页面
admin_addsmallclass.asp	新闻分类栏目管理页面
admin_conter.asp	增加新闻文章页面
admin_editfile.asp	新闻文章修改页面
bottom.asp	系统页脚页面
check_user.asp	用户身份验证文件
conn.asp	数据库连接文件
content.asp	主体新闻列表页面
dele_class.asp	删除新闻栏目文件
disp.asp	新闻文章内容浏览页面
index.asp	首页页面
list.asp	新闻栏目列表页面
login.asp	登录页面
menu.asp	前台导航菜单页面
news.asp	最新文章列表页面
quit.asp	退出后台管理文件
rd.asp	热点文章列表页面
seach_lr.asp	搜索表单页面
search.asp	站内文章搜索文件
top.asp	系统标题栏页面

有关页面的设计可查看光盘中的实例文件。其中在页面设计中，把功能相对独立的页面分为独立的文件，然后利用`<!--#include file="文件名" -->`命令导入到需要的地方，这样能节省

代码,提高工作效率,方便修改和更新。例如,首页 index.asp 就是全部利用各个功能模块文件组装起来的,代码如下:

```
<%LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<!--#include file="conn.asp" -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>新闻发布与管理</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<!--#include file="css.css" -->
</HEAD>
<BODY link=#000000 background=images/index_bg.gif topMargin=0>
<TABLE class=tabletblg cellSpacing=0 cellPadding=0 width=760 align=center border=0>
  <TR>
    <TD>
      <!--#include file="top.asp" -->
    </TD>
  </TR>
</TABLE>
<TABLE class=tablelrb cellSpacing=0 cellPadding=0 width=760 align=center border=0>
  <TR>
    <TD background=images/top-d.gif height=28>
      <!--#include file="menu.asp"-->
    </TD>
  </TR>
</TABLE>
<TABLE class=tablelrb cellSpacing=0 cellPadding=0 width=760 align=center border=0>
  <TR>
    <TD valign=top width=181 background=images/bocolor03.gif>
      <TABLE cellSpacing=0 cellPadding=0 width=160 border=0>
        <TR><TD class=fontsize10></TD></TR></TABLE>
      <!--#include file="login.asp"-->
      <!--#include file="news.asp"-->
    </TD>
    <TD valign=top width=400>
      <TABLE width=100% border=0 cellPadding=0 cellSpacing=0 bgcolor="#CFE3F3">
        <TR><TD class=fontsize10></TD></TR></TABLE>
      <!--#include file="content.asp"-->
    </TD>
    <TD valign=top width=180 background=images/bocolor04.gif>
      <TABLE cellSpacing=0 cellPadding=0 width=160 border=0>
        <TR>
          <TD class=fontsize10> </TD></TR></TABLE>
      <!--#include file="seach_lr.asp"-->
      <!--#include file="rd.asp"-->
    </TD>
  </TR>
</TABLE>
<table width="760" border="0" align="center" cellpadding="0" cellspacing="0"
background="images/bocolor02.gif" class="tabletblg">
  <tr><td>
    <!--#include file="bottom.asp" -->
  </td></tr>
</table>
</BODY></HTML>
```

整个 index.asp 文件共导入了 11 个文件,除了前 3 个导入文件是常用的以外,后面 8 个文件拼成了整个页面,这样整个页面代码看起来非常简洁,只需要用几个表格适当地组织一下即可,如图 16.3 所示。

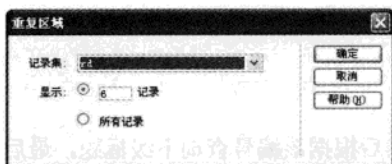


图 16.5 定义重复区域



图 16.6 浏览热点文章显示效果

16.4.2 新闻列表页面

当在首页菜单栏中单击选择一个菜单时，会打开 list.asp 文件，在该文件中包含该新闻大类下所有二级分类显示、最新文章和热点文章，以及栏目导航，设计如图 16.7 所示。

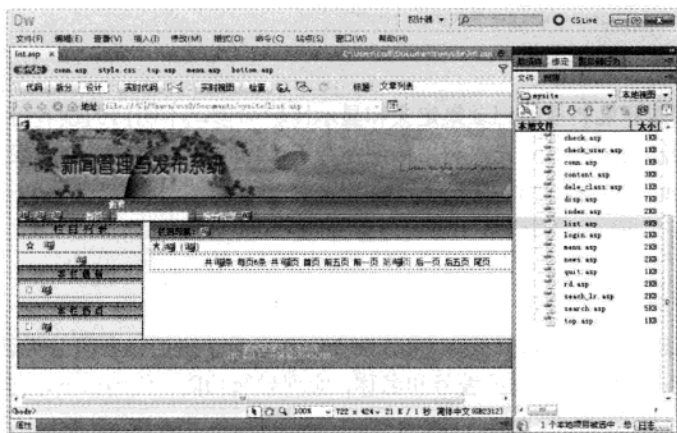


图 16.7 新闻列表设计效果

实现该功能的具体操作步骤如下。

- 01 打开 list.asp 文件，切换到【代码】视图，首先在代码顶部输入下面代码，实现与数据库的连接，同时获取单击菜单时传递过来的新闻大类编号：

```
<!--#include file="conn.asp" -->
<%listid=Cint(Request.QueryString("listid"))%>
```

- 02 制作新闻分类列表。下面来实现新闻多级分类列表浏览功能，先来浏览一下效果，如图 16.8 所示。

首先，根据传递过来的 listid 参数，来查询数据库中的该大类下的所有二级分类记录。代码如下：

```
<%str="select id,class_name from class where sup_class_id=" &listid &"
set rs0=conn.execute(str)%>
```

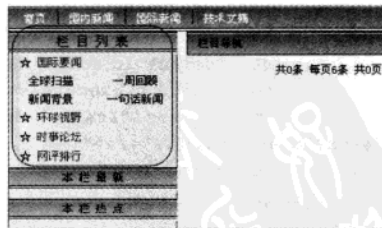


图 16.8 多级分类列表浏览效果

用“if not rs0.eof then”条件语句先判断该记录集是否存在记录,如果存在则用 do while not rs0.eof 语句循环显示该记录集中的分类标题。

然后,在该循环结构中再定义一个记录集,用来查询该二级分类中是否还存在下一级标题。代码如下:

```
<%nextid=rs0("id")
str="select id,class_name from class where sup_class_id='"&nextid&'"
set rs1=conn.execute(str)
if not rs1.eof then
m=0
do while not rs1.eof%>
```

上面代码首先获取 rs0 记录集中的记录编号,然后根据该编号查询下级标题,最后判断是否存在有记录,有则循环显示出来。其中定义一个 m 变量,用来实现一行显示 2 个记录的目的。代码如下:

```
<%
rs1.movenext
m=m+1
if m mod 2=0 and not rs1.eof then response.write "</tr><tr>"
loop
if m=1 then response.write "<td></td>"
end if
rs1.close%>
```

有关热点新闻和最新新闻记录集定义和显示比较简单,不明白的读者可以参考光盘。

03 栏目导航功能的实现。首先,利用传递参数 listid 来查询数据库 class 表中的该类记录中的 navigation 字段值。代码如下:

```
<%str="select id,navigation from class where id='"&listid&'"
set rs=conn.execute(str)
lmdh=rs("navigation")
rs.close%>
```

然后利用 ASP 函数 mid() 和 instr() 来分别查询该导航字段值中各段分类编号,利用该编号再查询 class 表中各个分类标题的编号和名称,实现导航功能。代码如下,浏览效果如图 16.9 所示:

```
<%lmmcc="<a href=index.asp class=fontsize12wubai>首页</a>"
do while lmdh<>""
lmda=Cint(left(lmdh,instr(lmdh,"|")-1))
lmdh=mid(lmdh,instr(lmdh,"|")+1)
str="select id,class_name from class where id='"&lmda&'"
set rsx=conn.execute(str)
lmmcc=lmmcc "&rarr; <ahref=list.asp?listid='"&rsx("id")&" class=fontsize12wubai>"&rsx("class_name")&"</a>"
rsx.close
loop%>
```



提示

mid() 表示返回特定长度的字符串,语法为 Mid(string, start [, length]), 其中从 start 开始,长度为 length。instr() 返回字符或字符串在另一个字符串中第一次出现的位置,语法为 Instr([start,] strToBeSearched, strSearchFor [, compare]), 其中 start 为搜索的起始值, strToBeSearched 为接受搜索的字符串, strSearchFor 为要搜索的字符, compare 为比较方式。



图 16.9 栏目导航功能浏览效果

04 分页功能的实现。先定义一个记录集 rs，利用传递参数 listid 查询 message 中所有该类下的文章记录。代码如下：

```
<%str="select id,title,add_date from message where small_class_id=" &listid &" order by
id desc"
rs.open str,conn,3,2
rs.pagesize=6
if pageno=0 or pageno="" then pageno=1 end if
if pageno<0 then pageno=1 end if
if pageno>rs.pagecount then pageno=rs.pagecount end if
if not rs.eof then
rs.absolutePage=pageno
for n=1 to rs.pagesize>
```

同时，在上面代码中设置每页要显示的最大记录数，定义当前页变量 pageno 的各种条件处置，然后循环显示一页中的各个记录。

05 在页面底部定义一个导航条。代码如下：

```
<tr>
<td height="30" colspan="2"> <div align="center" class="fontsize12h20wu">共<%=rs.
recordcount%>条 每页6条 共<%=rs.pagecount%>页 <a href=list.asp?listid=<%=listid%>&pageno=1
class="fontsize12wu">首页</a>
<a href=list.asp?listid=<%=listid%>&pageno=<%=pageno-5%> class="fontsize12wu">前五页</a>
<a href=list.asp?listid=<%=listid%>&pageno=<%=pageno-1%> class="fontsize12wu">前一页</a>
<font color="#FF0000">第<%=pageno%>页</font> <a href=list.asp?listid=<%=listid%>
&pageno=<%=pageno+1%> class="fontsize12wu">后一页</a>
<a href=list.asp?listid=<%=listid%>&pageno=<%=pageno+5%> class="fontsize12wu">后
五页</a>
<a href=list.asp?listid=<%=listid%>&pageno=<%=rs.pagecount%> class="fontsize12wu">尾
页</a></div></td>
</tr>
```

上面代码主要使用记录集的几个属性，其中 pagesize 表示记录集每页显示记录数，pagecount 表示记录集的总页数，absolutePage 表示记录集的当前页位置，recordcount 表示记录集的总记录数。设计效果如图 16.10 所示。

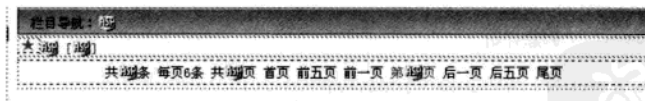


图 16.10 导航条功能浏览效果



提示

当要用到记录集分页属性时，如果使用上面代码实现查询数据库中的数据，这时要注意设置光标类型为静态，即设置为 3。

16.5 新闻管理

本实例新闻管理主要涉及到类别管理和信息管理，由于用户管理在前面实例中做过专题

介绍, 所以本实例就不再重复。当然, 读者在实际网站开发中应该加入用户管理模块。

16.5.1 栏目管理

栏目管理主要包括 `admin_addbigclass.asp` 和 `admin_addsmallclass.asp` 两个文件, 其设计基本相同, 如图 16.11 所示。

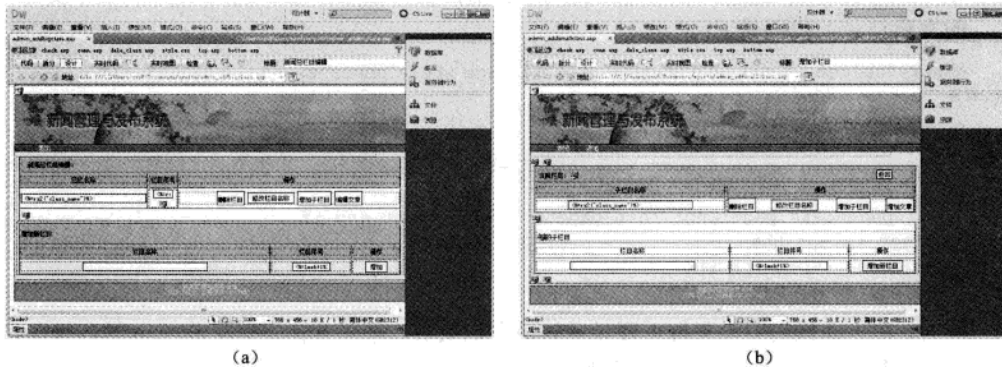


图 16.11 栏目管理页面设计

(a) `admin_addbigclass.asp` 页面; (b) `admin_addsmallclass.asp` 页面

下面以 `admin_addbigclass.asp` 页面为重点进行介绍, 其设计步骤如下。

01 获取传递参数, 包括菜单和栏目分类编号, 以及在该页面中的操作行为。具体代码如下:

```
<%lmid=Cint(Request.QueryString("lmid"))
zlid=Cint(Request.QueryString("zlid"))
doing=Trim(Request.QueryString("doing"))%>
```

02 定义删除栏目行为, 代码如下:

```
<%'删除一个栏目
if doing="delelm" then
    dele_id=Cint(Request.QueryString("dele_id"))
    lmide=dele_id%>
    <str="select id from class where sup_class_id=" & dele_id & ""
    set rs=conn.execute(str)
    if not rs.eof then
        rs.close
        msg1="该栏目存在有子栏目, 请先删除子栏目后再删除该栏目!"
    else
        rs.close '删除开始
        %>
        <!--#include file="dele_class.asp" --><%
    end if
end if
%>
```

03 定义修改栏目操作行为, 代码如下:

```
<%'修改栏目设置
if doing="editlm" then
    edid=Cint(Request.QueryString("edid"))
    lmmc=Trim(Request.Form("lmmc"))
    if lmmc<>" " then
        str="select id from class where class_name=" & lmmc & " and sup_class_id=" & lmid & " and
id<" & edid & ""
```

```

set rs0=conn.execute(str)
if rs0.eof then
    lmxh=Trim(Request.Form("lmxh"))
    lmwz=Request.Form("lmwz")
    if lmwz="" then lmwz="不显示"
    cdwz=Request.Form("cdwz")
    if cdwz="" then cdwz="无"
    lmlx=Request.Form("lmlx")
    glz=Trim(Request.Form("glz"))

    str="select * from class where id=&edid&"
    rs.open str,conn,2,3
    rs("class_name")=lmmc
    rs("orders")=lmxh
    rs.update
    rs.close
end if
rs0.close
end if
end if
%>

```

04 定义增加栏目行为，在增加栏目时，要注意 navigation 字段的写入代码。整个代码如下：

```

<% 增加总栏目
if doing="addzlm" then
    newlmmc=Trim(Request.Form("newlmmc"))
    newlmxh=Cint(Request.Form("newlmxh"))
    newlmwz=Trim(Request.Form("newlmwz"))
    newcdwz=Trim(Request.Form("newcdwz"))
    newxslx=Trim(Request.Form("newxslx"))
    newglz=Trim(Request.Form("newglz"))
    if newlmmc<>"" then
        str="select id from class where class_name='"&newlmmc&'" and sup_class_id=0"
        set rs0=conn.execute(str)
        if rs0.eof then
            str="select * from class"
            rs.open str,conn,2,3
            rs.addnew
            rs("class_name")=newlmmc
            rs("sup_class_id")=0
            rs("big_class_id")=0
            rs("orders")=newlmxh
            if newglz="" then newglz="|"
            rs.update
            rs.close
            str="select id from class where sup_class_id=0 and class_name='"&newlmmc&'"
            set rs=conn.execute(str)
            temp2=rs("id")
            rs.close
            temp3=temp2&"|"
            str="update class set big_class_id='"&temp2&',"navigation='"&temp3&'" where id=
"&temp2&'"
            conn.execute(str)
        end if
        rs0.close
    end if
end if
%>

```

05 定义一个记录集 rs2，查询所有的菜单分类，代码如下。然后绑定到表单 value 属性中，如图 16.12 所示。

```
<%str="select * from class where sup_class_id=0 order by orders"
set rs2=conn.execute(str) %>
```

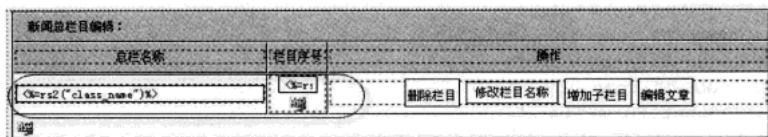


图 16.12 绑定记录集

06 判断该记录集 rs2 是否存在有记录，如果存在则重复显示所有记录。代码如下：

```
<% if not rs2.eof then
    do while not rs2.eof%>
... '表单域
<%
    rs2.movenext
    loop
end if
rs2.close%>
```

07 正确设置各个操作按钮的传递参数。代码如下：

```
<a href="admin_addbigclass.asp?doing=delelm&dele_id=<%=rs2("id")%>&lmid=<%=rs2("sup
class_id")%>" class="inputbox">删除栏目</a>
<input name="Submit" type="submit" class="inputbox" value="修改栏目名称">
<a href="admin_addsmallclass.asp?lmid=<%=rs2("id")%>&zlid=<%=rs2("big_class_id")%>"
class="inputbox">增加子栏目</a>
<a href="admin_addfile.asp?lmid=<%=rs2("id")%>&zlid=<%=rs2("big_class_id")%>" class=
"inputbox">编辑文章</a>
```

16.5.2 信息管理

文章管理页面的设计思路和栏目管理页面的设计思路基本相同，设计如图 16.13 所示。其制作步骤如下。

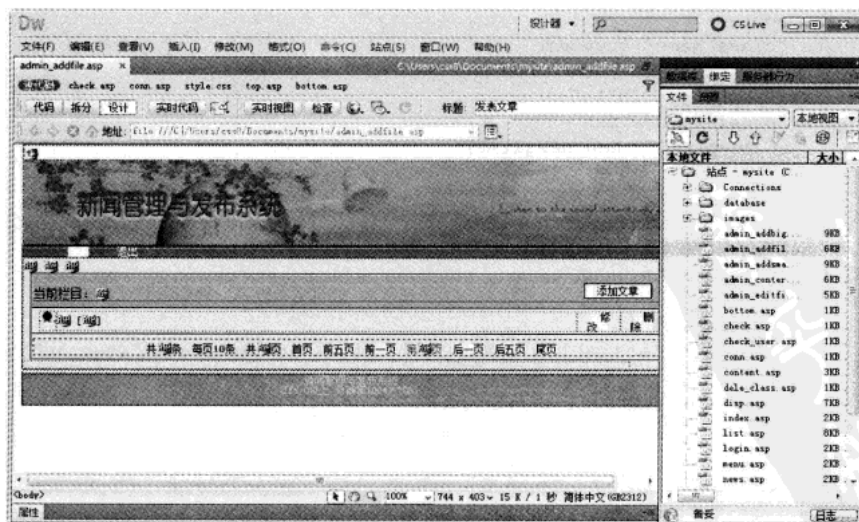


图 16.13 文章管理页面设计效果

01 用 Request.QueryString 语句获取传递参数，以便确定当前页面要显示的栏目文章和将要

执行的操作。

```
<%lmid=Cint(Request.QueryString("lmid"))
    zlid=Cint(Request.QueryString("zlid"))%>
<%doing=Trim(Request.QueryString("doing"))%>
```

02 定义一个删除文章的命令，即当接到要删除文章参数后，即执行该段语句。代码如下：

```
<% 删除一个文章
if doing="dele_tf" then
    deleid=Cint(Request.QueryString("deleid"))
    ide=deleid
    str="delete from message where id=" & deleid & ""
    conn.execute(str)
end if
%>
```

03 参照上节介绍的方法制作栏目导航功能，具体代码可参见上面一节介绍。

04 定义一个记录集查询所有的该栏目中文章，并设置记录集分页属性设置。代码如下：

```
<%str="select id,title,add_date from message where small_class_id=" & lmid & " order by id
desc"
rs.open str,conn,3,2
rs.pagesize=10
pageno=Cint(Request.QueryString("pageno"))
if pageno=0 or pageno="" then pageno=1 end if
if pageno<0 then pageno=1 end if
if pageno>rs.pagecount then pageno=rs.pagecount end if
if not rs.eof then
rs.absolutePage=pageno%>
```

05 在重复单元格中插入要显示的文章标题 title 字段，并模仿上节介绍方法制作导航条，效果如图 16.13 所示。

16.6 实践总结

本章重点介绍了新闻发布与管理系统的思路和制作步骤，其中还有很多细节问题限于篇幅不能拓展，用户可以在光盘中的实例文件的帮助下，仔细琢磨各个页面的实现方法。

新闻发布与管理系统在网上运用比较广泛，形式多样，而且根据站点类型和对象需要其功能也不尽相同，用户可以浏览新浪、搜狐等综合门户网站的新闻栏目设计，从中可获得不少有益的启示。

第 17 章

dreamweaver

网站即时聊天模块

即时聊天是一种简单、快捷的网上即时交流方式，深受网友喜爱。早期的即时聊天形式多为聊天室。聊天室一般都分主题，加入相应主题的聊天室，就可以与相同喜好的网友一起聊天，也可以选择一个单独的用户进行私聊。为了保证聊天室的秩序，踢除恶意用户，甚至屏蔽用户使其无法继续登录也是必要的功能。

早期的聊天室依赖于不断刷新屏幕来实现信息的即时更新和显示，但是频繁刷新屏幕会影响人的视力，聊天时间久了就会伤害眼睛。现在随着浏览器和服务端技术的不断改进，用户可以实现无刷新聊天室。不过受各种即时通信工具的挤压，聊天室这种曾经很受网友喜爱的 Web 应用形式也日渐萎缩，一般形成规模并积累了固定用户群的聊天室还能继续存在，但必须不断与其他技术和 Web 应用程序进行融合，以增强网站的吸引力。

技术要点

- 聊天室设计思路和方法
- 网页框架技术
- 网页实时更新
- Session 技术
- 隐藏域应用
- 限量查询
- 动态表单技术

17.1 程序设计分析

聊天室是一种网络即时通信工具，作为网上信息交流的一种方式，聊天室很受用户欢迎。当登录进入聊天室后，便可以自由发言，自由指定聊天对象。聊天室内所有发言信息都会即时显示在页面上，每个用户都可以看到，如同在一个屋内聊天，因此就形象地叫它聊天室。

与其他 Web 应用不同的是，聊天室要求聊天信息可以实现即时共享，即要求有一个应用程序级的变量，可以在整个应用程序中随时被访问。ASP 提供了 Application 对象，该对象提供了完善的访问、保护机制。但应用这个对象的变量之前，必须保证聊天室是运行在一个独立的程序域之内，享有独立的变量。

聊天室里的用户及聊天内容是不断变化的，Application 变量也会随之而发生变化。为了让客户端及时知道这些变化，就要求客户端浏览器定时刷新，不断地去读取最新的 Application 信息。刷新时间的间隔越短，客户就可以越快得到聊天室最新的信息；但是刷新越快，就

会影响视力对屏幕的阅读。

聊天室技术的核心就是利用框架技术实现在一个页面内同时显示聊天内容、浏览和选择聊天对象、发布聊天信息，如图 17.1 所示。

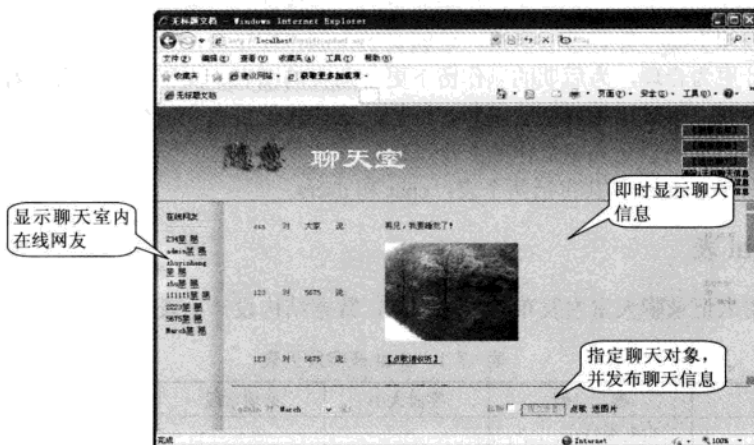


图 17.1 本例聊天室浏览效果

使用聊天室应先登录，一般还应该获取一个账号，才能进入聊天室。一般大型聊天室系统内部又包括很多分级聊天室，网友被分成不同级别的室友，关系复杂，涉及的技术难度大。本例相对简单，仅演示聊天室基本设计思路和方法。另外，在前面章节中曾介绍过注册功能的实现，所以在本例中用户可以不注册，直接登录即可进入聊天室主页，然后选择聊天对象进行交流。本例设计流程如图 17.2 所示。

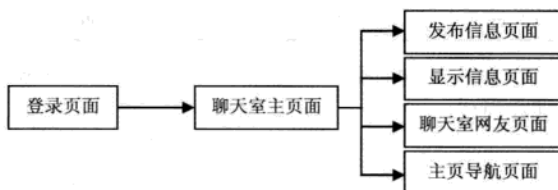


图 17.2 聊天室流程图

要实现本实例聊天室功能，读者需要学习和掌握下面技术。

- 登录进入聊天室，用 Session 技术保存用户登录信息。
- 发布信息，利用动态下拉菜单绑定登录聊天网友。
- 显示信息，显示页面要实时刷新并能即时显示聊天内容。
- 按发布时间顺序更新聊天信息，涉及限量查询。
- 发布图片和点歌。
- 聊天室常用管理功能实现。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“17”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

17.2 数据库设计

在程序设计中,实现一个功能往往是简单的。但保证程序易用、健壮、有良好的可扩展性和可维护性却不是一件容易的事情。这就要求在制作程序前,认真地分析程序的功能,使数据库的设计更为合理,为后期的制作留下更为翔实的图表、文档。虽然表面看起来前期耗费了更多的时间,不过比起后期的制作、维护,这些代价是完全值得的。特别是在团队合作中,这些图表、文档也是大家合作的一个基础。

本例数据库 chat.mdb 共包括两个表: chat 和 users。

17.2.1 chat 表

chat 表用来记录聊天室内发布的所有信息,数据结构设计见表 17.1。

表 17.1 chat 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	聊天信息编号
user	文本	50	否	信息发布者
to_who	文本	50	否	指定的聊天对象
with_what	文本	255	否	聊天内容
post_time	日期/时间	-	否	发布时间
with_url	文本	200	否	点歌的地址
with_gif	文本	200	否	发布的图片地址

在 post_time 字段中,设置“默认值”为“Now()”,如图 17.3 所示。使发布信息自动增加现在时间。

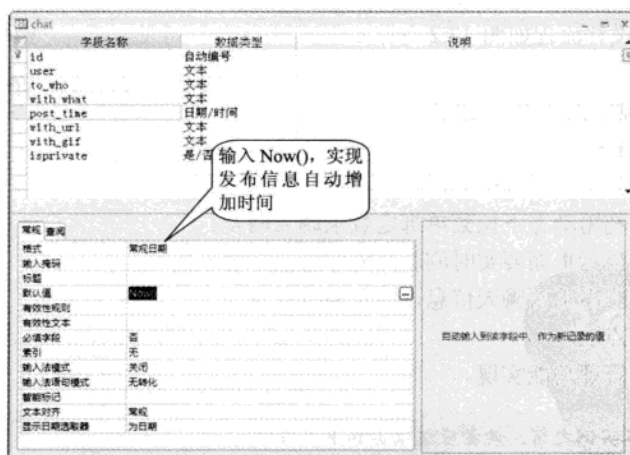


图 17.3 设置 post_time 字段为自动增加时间

17.2.2 users 表

users 表主要用来存储登录的聊天室友,该表数据结构设计见表 17.2。

表 17.2 users 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
user	文本	20	否	登录用户名
password	文本	20	否	登录密码
sex	文本	10	否	用户性别
logintime	日期/时间	-	否	用户登录时间
isonline	是/否	-	否	是否在线

在 logintime 字段中, 也设置“默认值”为“Now()”, 自动增加用户登录的时间。

17.3 页面设计

本实例共有 10 个文件, 具体说明见表 17.3。

表 17.3 系统文件列表

文件名	说 明
index.asp	聊天室登录页面
index1.asp	聊天室主框架集页面
Index2.asp	聊天室主体框架集页面
postmsg.asp	信息发布框架页面
Showmsg.asp	信息显示框架页面
showname.asp	在线网友显示框架页面
topmsg.asp	导航框架页面
mf.asp	发布图片和点歌页面
Loginout.asp	聊天退出文件
Conn.asp	数据连接文件

下面就几个主要页面设计进行介绍。

17.3.1 聊天信息发布页面

postmsg.asp 文件是专门用来发送聊天信息的表单, 其页面效果如图 17.4 所示。

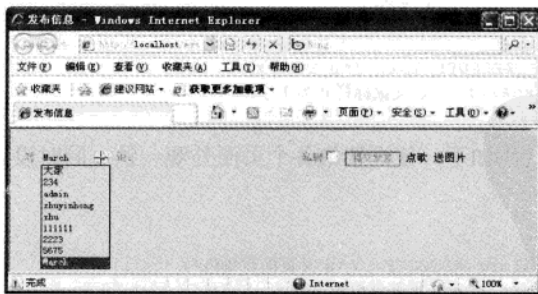


图 17.4 postmsg.asp 文件浏览效果

启动 Dreamweaver CS5, 选择【窗口】|【文件】菜单命令, 打开【文件】面板。然后打开 postmsg.asp 文件, 页面设计如图 17.5 所示。

本页面使用 4 个表单对象, 属性设置见表 17.4。

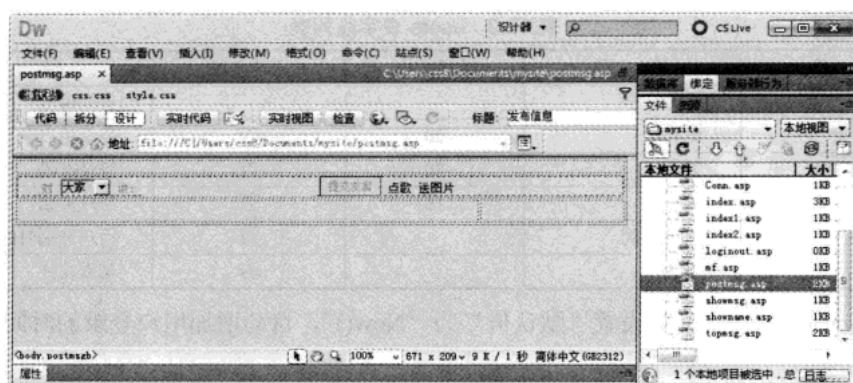


图 17.5 postmsg.asp 文件页面设计

表 17.4 登录表单设置列表

对 象	名 称	类 型	设置说明
表单	form1	表单	【动作】先不设置, 设置服务器行为后, 系统自动插入; 【方法】为“POST”
列表/菜单	to_who	下拉菜单	【类型】为“菜单”, 【初始化设置】为“大家”, 在后面功能实现中, 再介绍如何绑定动态数据
文本框	with_what	单行文本	【字符宽度】为 30
按钮	Submit	提交	【值】为“提交发言”, 【动作】为“提交表单”

为了美化表单外观, 这里利用 CSS 样式表修饰表单。代码如下:

```
input {
    font-size: 12px; /*字体大小*/
    font-weight: lighter; /*定义字体较轻显示*/
    color: #0000FF; /*定义字体颜色*/
    text-decoration: none; /*设置不带下划线*/
    background-color: #CCCCFF; /*定义背景色*/
    height: auto; /*定义行高为自动*/
    border: thin #FF00FF 1px; /*定义边框*/
    cursor: hand; /*定义鼠标样式为手形*/
}
select {
    font-size: 12px; /*定义字体大小*/
    font-weight: lighter; /*定义字体较轻显示*/
    color: #0000FF; /*定义字体颜色*/
    text-decoration: none; /*设置不带下划线*/
    background-color: #CCCCFF; /*定义背景色*/
    border: thin #FF00FF 1px; /*定义边框*/
    cursor: crosshair; /*定义鼠标样式为十字形*/
}
```

设置页面属性, 代码如下, 使框架集各个页面外观一致, 同时设置上、左边距为 0, 使框架页面能紧密结合:

```
body {
    background-color: #CCCCFF; /*定义页面背景色*/
    margin-top: 0px; /*定义顶部页边距为0*/
    margin-left: 0px; /*定义左侧页边距为0*/
}
```

17.3.2 聊天信息显示页面

Showmsg.asp 文件专门用来显示用户聊天信息, 页面显示效果如图 17.6 所示。

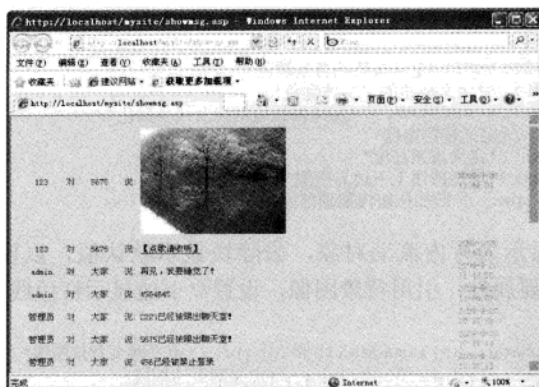


图 17.6 Showmsg.asp 文件浏览效果

Showmsg.asp 页面设计简单，只需要插入一个单行多列的表格即可。为了保持页面的美观，应设置滚动条颜色和页面颜色一致。实现方法是在页面头部区域输入下面的 CSS 代码：

```
<style type="text/css">
body {
background-color:#CCCCFF; /*定义页面背景色*/
scrollbar-face-color:#9999FF; /*以下将定义页面滚动条样式*/
scrollbar-highlight-color:#CCCCFF;
scrollbar-3dlight-color:#CCCCFF;
scrollbar-darkshadow-color:#CCCCFF;
scrollbar-Shadow-color:#CCCCFF;
scrollbar-arrow-color:#9999FF;
scrollbar-track-color:#CCCCFF;
}</style>
```



提示

scrollbar-face-color 设置滚动条凸出部分的颜色。
 scrollbar-highlight-color 设置滚动条空白部分的颜色。
 scrollbar-3dlight-color 设置滚动条亮边的颜色。
 scrollbar-darkshadow-color 设置滚动条暗边的颜色。
 scrollbar-Shadow-color 设置立体滚动条阴影的颜色。
 scrollbar-arrow-color 设置上下按钮上三角箭头的颜色。
 scrollbar-track-color 设置滚动条的背景颜色。
 scrollbar-base-color 设置滚动条的基本颜色。

17.3.3 导航页面

topmsg.asp 文件主要用来显示聊天室顶部导航条，页面主要涉及用一个<div>标签控制 3 个按钮表单对象。代码如下：

```
<body topmargin="0">
<div style="display:inline; vertical-align:middle; float:right; margin-bottom:auto; margin-top:25px;">
<form>
<input type="button" class="button" name="refreshonline" value="【刷新名单】" onClick=
"top.showname.location.reload()"><br>
<input type="button" class="button" name="refreshonline" value="【刷新屏幕】" onClick=
"top.showmsg.location.reload()"><br>
<input type="button" class="button" name="refreshonline" value="【退出聊天】" onClick=
"top.window.location.href='logout.asp'"><br>
</form>
</div>
```

同时设置其中的按钮样式为:

```
.button{
    background:#00aeef; /*定义按钮的背景颜色*/
    color:#000090; /*定义按钮的中字体颜色*/
    border:solid 1px #ffffff; /*定义按钮边框线及线条颜色*/
    width:100px; /*定义按钮宽度*/
    height: 20px; /*定义按钮高度*/
    margin:2px 0; /*增加按钮上下边界空隙*/
    padding-top:1px; /*增加按钮内部顶部空隙*/
}
```

设置<div>标签显示为行内流动对象, 去除块显示默认值, 以便其向右流动。同时在<body>标记中设置主要属性, 引用背景图像, 设置页下边显示红色线条, 代码如下:

```
body {
    background-image: url(images\logo.jpg); /*定义网页背景图像*/
    border-bottom:solid #FF0000 1px; /*定义网页边框线*/
}
```

整个页面显示效果如图 17.7 所示。



图 17.7 topmsg.asp 文件浏览效果

17.3.4 主体框架集

index2.asp 文件是一个嵌套框架集文件, 框架结构如图 17.8 所示。嵌套框架集具体实现步骤如下。

01 在 Dreamweaver CS5 中新建一个动态文件, 然后选择【插入】|【HTML】|【框架】|【左侧及下方嵌套】菜单命令, 插入一个框架集, 如图 17.9 所示, 保存页面为 index2.asp。

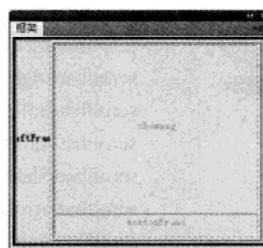


图 17.8 index2.asp 框架集结构

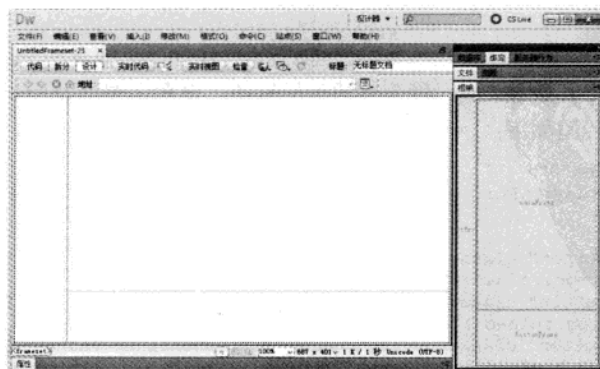


图 17.9 插入框架集

02 设置各个框架页属性和页面源文件见表 17.5。

表 17.5 框架页属性设置列表

框架名称	滚 动	不能调整大小	源 文 件
showname	否	是	showname.asp
showmsg	是	否	showmsg.asp
postmsg	否	是	postmsg.asp
按钮	Submit	提交	【值】为“提交发言”，【动作】为“提交表单”

03 设置左框架的宽度【列值】为 100 像素，右侧框架【单位】设置为“相对”，【列值】为“1”，如图 17.10 所示。

04 利用【框架】面板，选中其中的上下结构的嵌套框架集，以同样的方法设置下框架的宽度【列值】为 100 像素；上框架的【单位】设置为“相对”，【列值】为“1”。

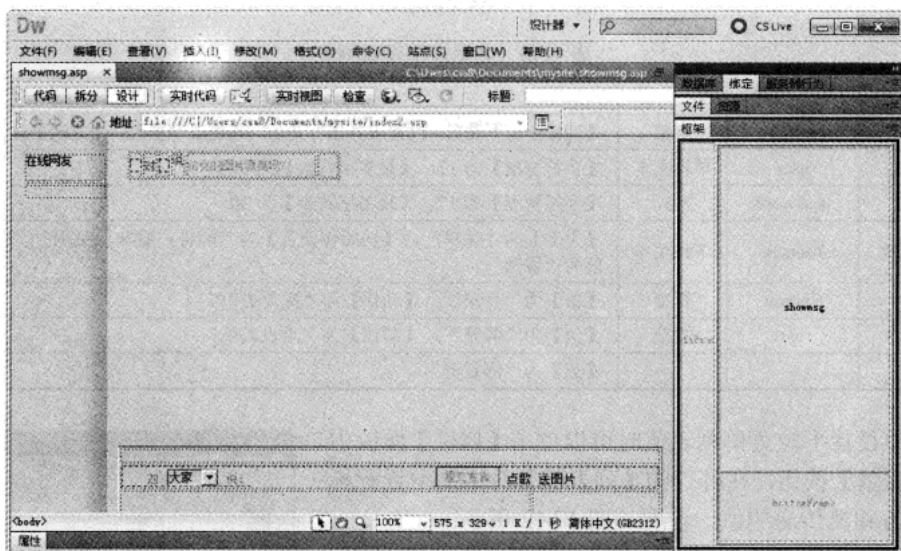


图 17.10 index2.asp 框架集效果

17.3.5 聊天室框架集

index1.asp 也是一个框架集文件，它是一个上下结构的框架集，操作方法同上，实现代码如下：

```
<frameset rows="145,*" frameborder="NO" border="0" framespacing="0">
  <frame src="topmsg.asp" name="topmsg" scrolling="NO" noresize >
  <frame src="index2.asp" name="include">
</frameset>
<noframes><body>
</body></noframes>
```

17.3.6 登录页面

登录页面设计如图 17.11 所示。该页面主要由 6 个表单对象组成，各个表单对象属性设置如表 17.6 所示。

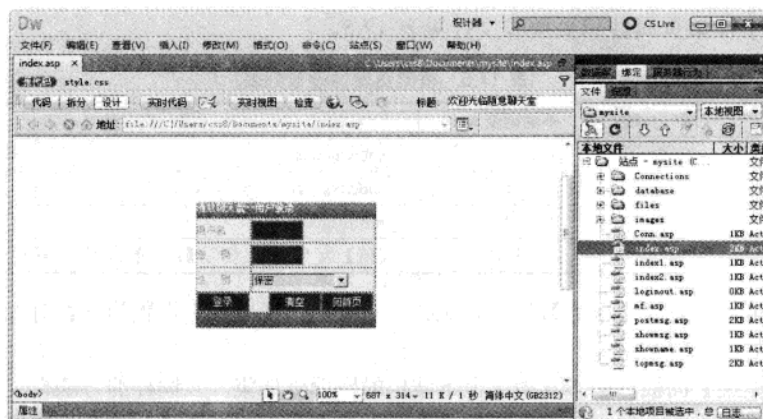


图 17.11 登录页面设计效果

表 17.6 登录表单设置列表

对 象	名 称	类 型	设置说明
表单	form1	表单	【动作】先不设置, 设置服务器行为后, 系统自动插入; 【方法】为“POST”
文本框	user	单行文本	【字符宽度】为 17, 【最多字符数】为 10
文本框	password	密码	【字符宽度】为 17, 【最多字符数】为 30
列表/菜单	usersex	下拉菜单	【类型】为“菜单”, 【初始化设置】为“帅哥、靓妹、保密”, 设置选定值为“保密”
按钮	Submit	提交	【值】为“登录”, 【动作】为“提交表单”
按钮	reset	提交	【值】为“清空”, 【动作】为“重设表单”
按钮	-	-	【值】为“回首页”

其中设置下拉菜单列表值时可以单击【属性】面板中的【列表值】按钮, 在打开的【列表值】对话框中设置标签显示值和要传递的值, 设置如图 17.12 所示。

在 3 个按钮标记中增加按钮动态控制属性:

```
onClick="window.location.href='#'"
onMouseOver="this.style.backgroundColor='#ff0000'"
onMouseOut="this.style.backgroundColor='#006898'"
```

第 1 行代码定义了单击按钮可以跳转到某页, 第 2、3 行代码设置鼠标经过和移出时, 按钮的背景颜色发生变化。

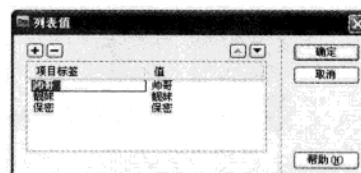


图 17.12 设置下拉菜单列表值

17.4 聊天室基本功能实现

下面分别介绍框架集中不同的页面功能的实现, 其中 postmsg.asp 框架页用来发布聊天内容, showmsg.asp 框架页用来显示聊天内容, showname.asp 框架页用来显示在线用户名单。

17.4.1 用户登录和用户信息存储

index.asp 的主要功能是利用服务器行为把用户登录的信息插入数据库, 并把用户名传递给下一页。为了实现信息的完整保存, 需要插入隐藏域, 补全信息。具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中, 打开 index.asp 文件, 把光标置于表单内, 在菜单栏中选择【插入】|【表单】|【隐藏域】命令, 插入一个隐藏域对象。
- 02** 选中该隐藏域, 在属性面板中设置隐藏域对象的名称和值, 如图 17.13 所示, 代码如下。

```
<input name="isonline" type="hidden" id="isonline" value="True">
```

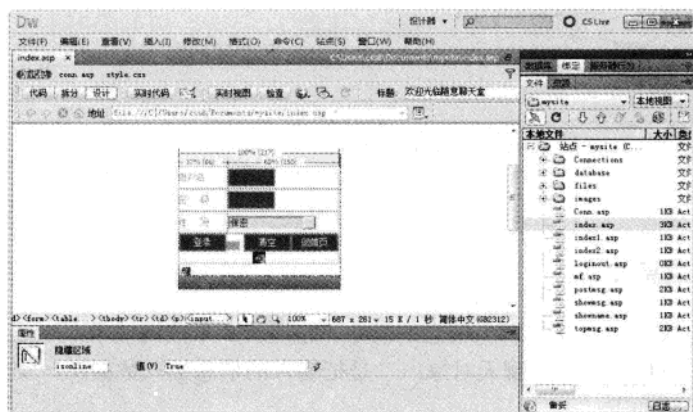


图 17.13 设置隐藏域属性



提示

隐藏域的主要功能是传递一些其他表单对象无法或没能传递的值, 一般在页面中不能显示隐藏域, 但在源代码中可以查看隐藏域要传递的值。

也可以直接输入代码来实现插入隐藏域, 代码如下: `<input name="isonline" type="hidden" id="isonline" value="True">`。

- 03** 设置表单的方法为 `method="get"`, 即通过查询字符串参数传递表单中的信息。在菜单中选择【窗口】|【服务器行为】命令, 打开【服务器行为】面板。
- 04** 单击面板左上角的【+】按钮, 在弹出的下拉菜单中选择【插入记录】命令, 打开【插入记录】对话框。在【连接】下拉列表中选择前面定义好的数据源 conn; 在【插入到表格】下拉列表中选择要插入的数据表 users; 在【插入后, 转到】文本框中设置插入记录成功后要跳转到 index1.asp, 如图 17.14 所示。

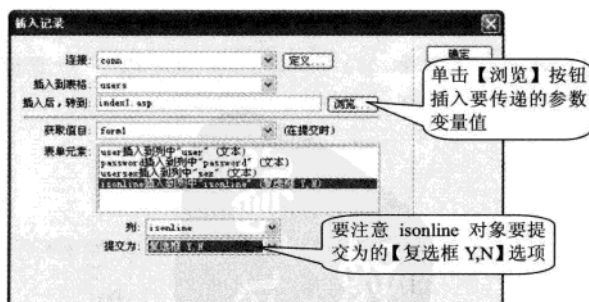


图 17.14 【插入记录】对话框

- 05** 在【表单元素】列表框中分别设置每个表单对象中的数据, 要插入数据库中 users 表中哪些字段。设置完毕, 单击【确定】按钮即可实现插入记录功能。由于 password 字段名

与 VBScript 预保留字存在冲突,读者还应该在【代码】视图下把“插入记录”服务器行为中的 SQL 操作字符串中的 password 用中括号括起来,代码如下:

```
MM_editCmd.CommandText = "INSERT INTO users ([user],[password], sex, isonline) VALUES ( ?, ?, ?, ?) "
```

- 06 切换到【代码】视图,在“插入记录”服务器行为代码段中插入下面代码,用来保存用户名信息到阶段变量 Session 中:

```
'-----
'定义会话变量存储用户名
Session("user") = Request.Form("user")
'-----


Response.Redirect(MM_editRedirectUrl)
End If
End If
%>
```

17.4.2 聊天信息发布

postmsg.asp 文件用于实现聊天信息的发布,它涉及 2 个关键技术:一是把动态数据绑定到下拉菜单中,方便用户选择聊天对象;二是把输入的信息插入数据库,实现信息的间接发布。具体操作步骤如下。

1. 实现登录用户名动态显示

- 01 在 Dreamweaver CS5 中打开 postmsg.asp 文件。选择【窗口】|【绑定】菜单命令,打开【绑定】面板。

- 02 单击【绑定】面板左上角的  按钮,在弹出的下拉菜单中选择【阶段变量】命令。打开【阶段变量】对话框,输入变量 user,如图 17.15 所示。该会话变量已在 index1.asp 文件中定义过。代码如下:

```
<%
//定义会话变量存储用户名
Session("user") = Request.QueryString("user")
%>
```

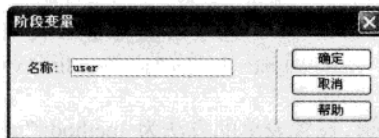


图 17.15 获取用户信息

- 03 在【绑定】面板中展开 Session 对象,然后把 user 变量插入到页面中要显示用户名的位置,如图 17.16 所示。

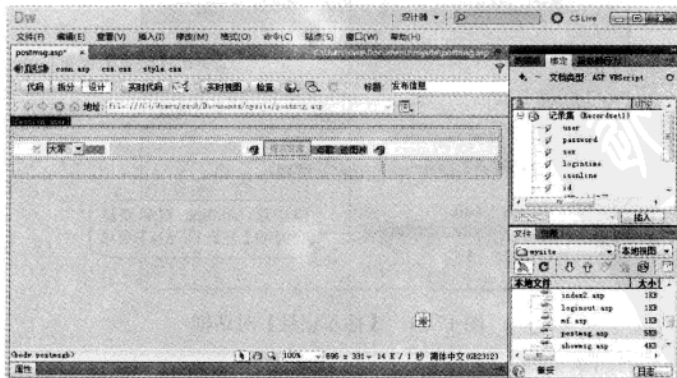


图 17.16 绑定用户会话变量

2. 实现动态下拉菜单功能

当不同的用户登录进入聊天室后,在信息发布页面的“对谁说”下拉菜单中如果能动态显示所有登录用户名,就会使页面更智能化,方便选择聊天对象,其设计步骤如下。

01 在【绑定】面板中定义一个记录集,用来筛选所有在线用户信息。选择【窗口】|【绑定】菜单命令,打开【绑定】面板。单击【绑定】面板左上角的 \oplus 按钮,在弹出的下拉菜单中选择【记录集(查询)】命令。

02 在打开的【记录集】对话框中设置各选项如图 17.17 所示。

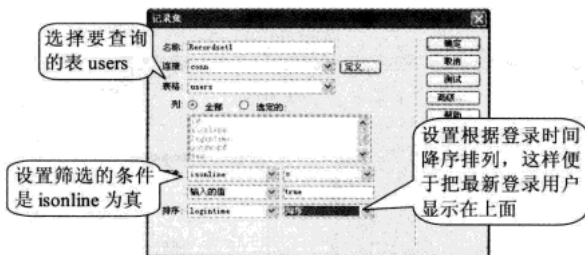


图 17.17 设置记录集查询条件

03 定义记录集完毕,回到信息发布页面,选中下拉菜单,在【属性】面板中单击【动态】按钮,打开【动态列表/菜单】对话框,设置如图 17.18 所示。

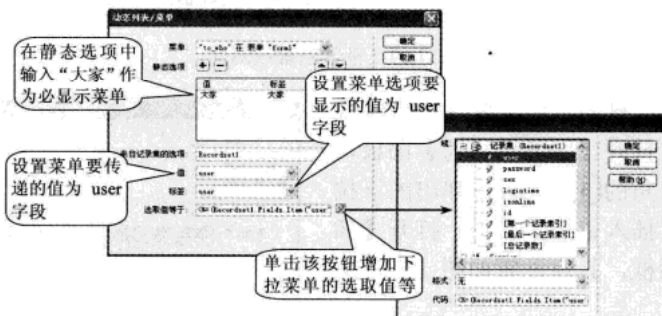


图 17.18 绑定动态下拉菜单选项

04 切换到【代码】视图,找到动态下拉菜单代码段,在“<option value="大家">大家</option>”代码中输入 selected, 目的是设置【大家】为初选值。

```
<select name="to_who" id="to_who">
  <option value="大家" selected <%If (Not isNull((Recordset1.Fields.Item
("user").Value))) Then If ("大家" = CStr((Recordset1.Fields.Item("user").Value))) Then
Response.Write("selected="selected") : Response.Write(">>大家</option>
  <%
While (NOT Recordset1.EOF)
  <%
    <option value="&=(Recordset1.Fields.Item("user").Value)" <%If (Not isNull
((Recordset1.Fields.Item("user").Value))) Then If (CStr(Recordset1.Fields.Item("user").
Value) = CStr((Recordset1.Fields.Item("user").Value))) Then Response.Write("selected=
selected") : Response.Write(">>&=(Recordset1.Fields.Item("user").Value)">></option>
  <%
  Recordset1.MoveNext()
Wend
If (Recordset1.CursorType > 0) Then
```

```

Recordset1.MoveFirst
Else
Recordset1.Requery
End If
%>
</select>

```

- 05** 插入一个隐藏域，用来向数据库中传递用户名。把光标置于表单中，在菜单栏中选择【插入】|【表单】|【隐藏域】命令，插入一个隐藏域对象。选中该隐藏域，在属性面板中设置隐藏域对象的名称和值，如图 17.19 所示。



图 17.19 设置隐藏域属性

- 06** 把输入的聊天信息插入到数据库中，实现信息的发布。打开【服务器行为】面板，单击面板左上角的【+】按钮，在弹出的下拉菜单中选择【插入记录】命令。打开【插入记录】对话框，具体设置如图 17.20 所示。

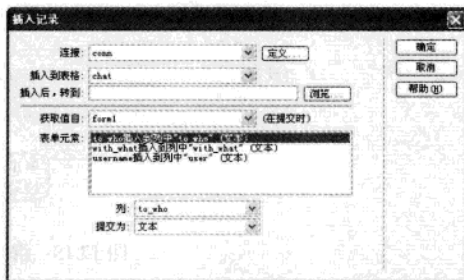


图 17.20 插入记录

17.4.3 聊天信息即时显示

聊天信息显示页面只需要定义一个记录集，并把记录集字段绑定到页面上，然后通过代码实现页面的实时刷新即可，具体操作步骤如下。

- 01** 打开 showmsg.asp 页面，在【绑定】面板中定义一个记录集。在【记录集】对话框中设置要查询的信息，如图 17.21 所示。
- 02** 在【记录集】对话框中，单击【高级】按钮，切换到【记录集】对话框的高级状态，然后在 SQL 查询语句中增加下面语句，如图 17.22 所示。

```

SELECT top 30 *
FROM chat
ORDER BY id DESC

```

插入“top 30”的目的就是指定查询前 30 位的记录，其他记录不再显示。同时要以降序排列记录，以便使最新聊天信息显示在最前面。

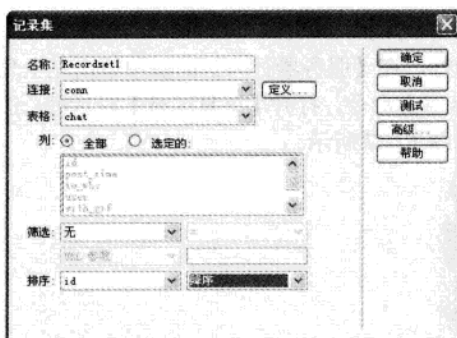


图 17.21 设置查询的基本条件

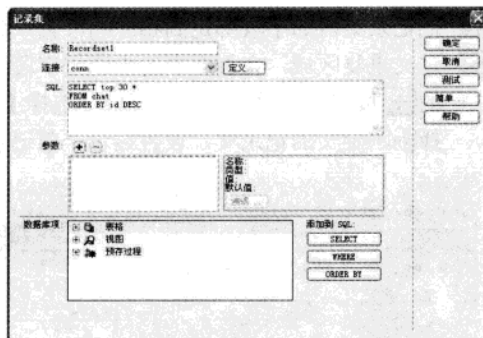



图 17.22 设置查询的高级条件

03 把记录集内的字段绑定到相应的位置，如图 17.23 所示。

04 定义重复区域。在页面中选中绑定字段的那一行，在【服务器行为】面板中单击左上角的  按钮，在弹出的下拉菜单中选择【重复区域】命令。打开【重复区域】对话框，选择刚刚定义的记录集，设置显示全部记录，如图 17.24 所示。

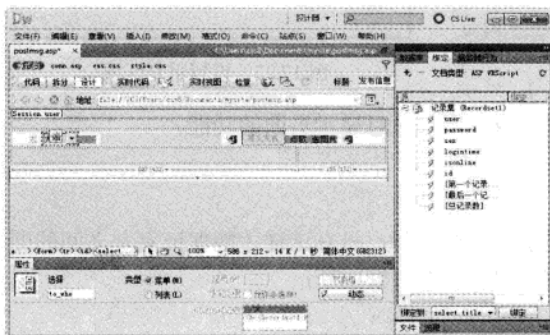


图 17.23 绑定记录集字段

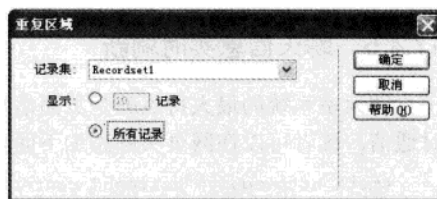


图 17.24 设置重复区域

05 单击【确定】按钮关闭对话框，这样就完成了聊天信息的显示功能。

17.4.4 在线用户即时显示

在线用户信息显示页面和聊天信息显示页面的功能基本相似，它需要定义一个记录集，并把记录集字段绑定到页面上，然后定义绑定字段的重复区域，具体操作步骤如下。

01 打开 showname.asp 页面，在【绑定】面板中定义一个记录集。在【记录集】对话框中设置要查询的信息，如图 17.25 所示。



图 17.25 设置查询条件


- 02 把记录集内的字段绑定到相应的位置, 如图 17.26 所示。
- 03 定义重复区域。在页面中选中绑定字段的那一行, 在【服务器行为】面板中单击左上角的  按钮, 在弹出的下拉菜单中选择【重复区域】命令。打开【重复区域】对话框, 选择刚刚定义的记录集, 设置显示全部记录, 如图 17.27 所示。



图 17.26 绑定记录集字段

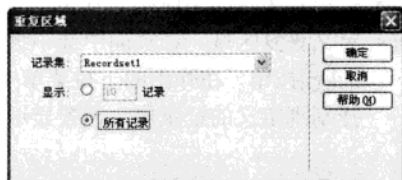


图 17.27 设置重复区域

- 04 单击【确定】按钮关闭对话框, 这样就完成了用户信息显示页面的功能。

17.4.5 聊天信息实时刷新

聊天室系统的最大特点就是页面信息显示的实时更新, 只有这样才能保证室友之间的即时通信。读者可以在网页头部添加下面代码实现页面的频繁刷新:

```
<meta http-equiv="Refresh" content="1;URL=index.asp" />
```

下面介绍如何使用脚本代码来实现页面的实时更新, 步骤如下。

- 01 打开 showmsg.asp 页面, 首先定义 2 个隐藏域, 分别命名为 chart 和 percent, 利用这两个隐藏域实现向状态栏传递信息。代码如下:

```
<table>
<tr>
<td align="center"><form name=loading>
<table border=0 cellpadding=0 cellspacing=0 width="100%" height="100%">
<tr>
<td align="center" valign="top">
<input type="hidden" name="chart" size=46 style="font-family:Arial; font-weight:bold; color:black; background-color:#EAEAEA; padding:0px; border-style:none;">
<br>
<input type="hidden" name="percent" size=46 style="font-family:Arial; color:black; background-color:#EAEAEA; text-align:center; border-width:medium; border-style:none;">
</td>
</tr>
</table>
</form></td>
</tr>
</table>
```

- 02 切换到【代码】视图, 在底部输入下面代码:

```

<script>
var bar = 0
var line = "||"
var amount = "||"
count()
function count(){
    bar= bar+1
    amount =amount + line
    document.loading.chart.value=amount
    document.loading.percent.value=bar+"%"
    if (bar<99)
    {
        setTimeout("count()",20);
    }
    else
    {
        window.location = "showmsg.asp";
    }
}
</script>

```

这样就可以实现不断刷新页面，并动态向状态栏传递信息。

- 03** 打开 topmsg.asp 页面，切换到【代码】视图，设置【刷新名单】和【刷新屏幕】按钮的附带脚本。输入代码如下：

```



```

上面方框内的代码定义当单击按钮时，调用 reload()函数刷新页面。

- 04** 打开 showname.asp 页面，切换到【代码】视图，在头部输入下面代码：

```

<meta http-equiv="Refresh" content="10;URL=''" />

```

上面对 3 个不同页面使用了 3 种方式来实现页面的实时刷新，这些不同实现方法功能是相同的，用户可以选择使用。

17.4.6 退出聊天室

退出聊天室并不是简单的关闭页面，它涉及两个问题：一是用户退出后，但该用户的 Session 信息依然存在，假设用户再次进入，就会发生冲突；二是用户退出之后，数据库中依然记录着该用户在线的信息。因此当退出时需要清除数据库中的在线记录和 Session 存储信息。

退出聊天室的具体实现方法就是当单击【退出】按钮后，跳转到 logout.asp 文件中，执行相应的代码，并返回首页。

首先，新建一个 conn.asp，在该文件中用代码打开数据库连接，并定义一个记录集变量。代码如下：

```

<!--#include file="Connections/conn.asp" -->
<%
dim conn,rs
set conn=server.createobject("adodb.connection")
conn.open MM_conn_STRING
set rs=server.createobject("adodb.recordset")
%>

```

在上面代码中先用 #include 命令把数据库连接字符串包含到本文件中，然后利用这个字符串打开数据库连接，最后再定义一个记录集实例。

把 conn.asp 文件包含到 loginout.asp 文件中，然后清除用户信息，具体代码如下：

```
<!--#include file="conn.asp" -->
<%
sql="update users set isonline = false where user='"&session("user")&"'"
conn.execute sql
conn.close
set conn=nothing
Session("user")=""
Response.redirect "index.asp"
%>
```

该代码段主要功能就是设置 Session("user") 变量为空，设置 user 数据表中的 isonline 字段为假。

17.5 扩展聊天室功能

一个完善的聊天室，其功能是非常复杂的，限于篇幅本书只能探索聊天室的基本功能，下面就聊天室的常用功能展开介绍。

17.5.1 发布图片和点歌

在聊天室内如果能给网友发去一张漂亮的图片或者给对方点首歌，聊天的兴趣会更浓，下面就这个问题进行介绍。发布图片和点歌技术的核心就是把图片和歌曲的地址发布到数据库，在信息显示页面判断数据库中是否有存在图片和歌曲，然后把它显示出来，效果如图 17.28 所示。



图 17.28 发布图片和点歌

其设计步骤如下。

- 01 先打开 postmsg.asp 页面，输入 2 个链接文本，如图 17.29 所示。
- 02 切换到【代码】视图，设置链接代码如下：

```
<a href="#" onClick="MM_openBrWindow('mf.asp?cat=歌曲','','width=600, height= 10')">点歌</a>
```

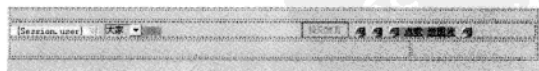


图 17.29 设置链接文本

```
<a href="#" onClick="MM_openBrWindow('mf.asp?cat=图片','','width=600,height= 10')">
送图片</a>
```

方框代码表示当单击按钮时将用 MM_openBrWindow() 系统函数打开 mf.asp 文件, 设置打开窗口大小为宽 600, 高为 10, 并传递单击按钮参数。


- 03 打开 mf.asp 页面, 在【绑定】面板中定义一个在线用户记录集, 在【记录集】对话框中设置要查询在线用户的信息, 如图 17.30 所示。
- 04 在【绑定】面板中定义一个请求变量来获取传递过来的参数变量。在【绑定】面板中单击面板左上角的  按钮, 在弹出的下拉菜单中选择【请求变量】命令, 打开【请求变量】对话框。
- 05 在【类型】下拉菜单中选择“Request.QueryString”选项, 获取传递变量 cat 的值, 如图 17.31 所示。



图 17.30 设置用户查询信息

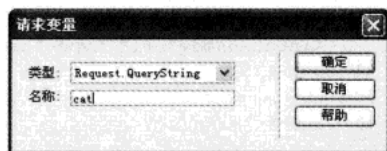



图 17.31 设置请求变量

- 06 把记录集和请求变量绑定到页面中, 参考第 17.4.4 节方法和步骤绑定用户下拉菜单的动态信息, 设置后的页面如图 17.32 所示。
- 07 在该页面中定义一个隐藏变量 user, 用来向数据库中插入用户名, 并绑定动态值为 <%= Session("user") %>, 操作方法可参照第 17.4.2 节。代码如下:

```
<input name="user" type="hidden" id="user" value="<%= Session("user") %>" />
```

- 08 用服务器行为向数据库中插入记录。打开【服务器行为】面板, 单击面板左上角的  按钮, 在弹出的下拉菜单中选择【插入记录】菜单命令, 打开【插入记录】对话框, 设置如图 17.33 所示。

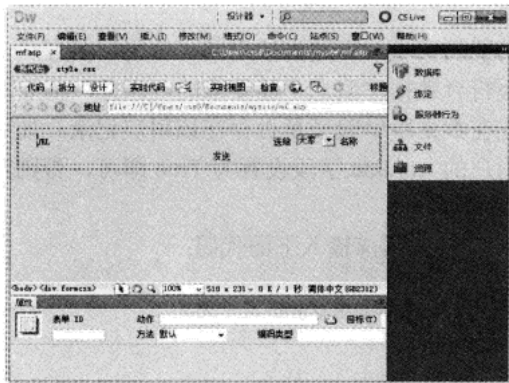


图 17.32 设置 mf.asp 页面

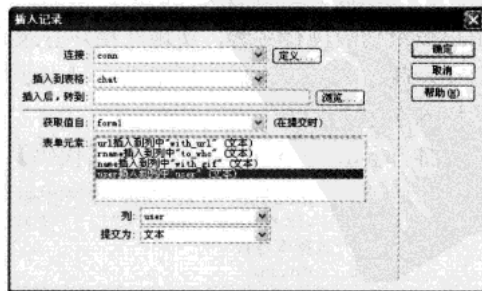


图 17.33 插入记录

- 09** 切换到【代码】视图，找到“插入记录”服务器行为代码段，然后在 SQL 字符串变量语句前后添加下面代码：

```

'-----
'根据用户提交的信息，判断是发送图片，还是点歌，然后分别定义不同的SQL字符串内容
if Request.QueryString("cat") = "歌曲" then
    MM_editCmd.CommandText = "INSERT INTO chat (with_url, to_who, with_what, [user])
VALUES (?, ?, ?, ?)"
elseif Request.QueryString("cat") = "图片" then
    MM_editCmd.CommandText = "INSERT INTO chat (with_gif, to_who, with_what, [user])
VALUES (?, ?, ?, ?)"
end if
'-----

```

- 10** 找到表单代码区域，用相同的条件结构修改文本域显示的默认值。代码如下：

```

<%
if Request.QueryString("cat") = "歌曲" then
%>
    <input name="url" type="text" value="files/1.mp3" />
<%
elseif Request.QueryString("cat") = "图片" then
%>
    <input name="url" type="text" value="files/1.jpg" />
<%
end if
%>

```

- 11** 打开 showmsg.asp 文件，切换到【代码】视图，在“说”所在单元格后面插入下面代码段。

```

<% if Recordset1.Fields.Item("with_url").Value<>" then %>
<a href="<%= (Recordset1.Fields.Item("with_url").Value) %>" target="_blank">
<%Response.Write("【点歌请收听】") %></a>
<%elseif Recordset1.Fields.Item("with_gif").Value<>" then%>
<a href="<%= (Recordset1.Fields.Item("with_gif").Value) %>" target="_blank"><img src=
"<%= (Recordset1.Fields.Item("with_gif").Value) %>" width="200" border="0" ></a>
<%end if %>

```

该代码段对 with_url 和 with_gif 字段进行判断，如果存在路径，即用 HTML 代码显示相应的信息，并提供超链接操作。

17.5.2 用户登录功能完善

在第 12.4.1 节中曾经详细介绍了用户登录功能的实现，但是如果用同一个用户名登录，则系统会在数据库中两次存储用户信息，这是不符合数据库信息存储要求的。因此，需要对用户登录功能进行完善。实现在用户登录之前先在数据库中验证是否存在该用户记录。如果存在，就不需要再次插入新记录，只需要把该用户的 isonline 字段设置为 True 即可，否则再次插入新记录。

打开 index.asp 文件，切换到【代码】视图，在代码顶部输入下面代码：

```

<% '完善用户登录功能，避免重复向数据库中插入记录
if Request.Form("MM_insert") = "form1" then '判断是否是提交表单
    '打开数据库连接
    dim conn,rs
    set conn=server.createobject("adodb.connection")
    conn.open MM_conn_STRING

```

```

set rs=server.createobject("adodb.recordset")
'用表单中用户输入的用户名和密码查询数据库
user = Request.Form("user")
password = Request.Form("password")
sql = "Select * from users where [user] = '" & user & "' and [password] = '" &
password & "'"
rs.open sql ,conn, 1,2
if not rs.eof then '判断是否存在该用户名和密码
    Session("user") = user '保存用户名到阶段变量中
    rs("isonline")= True '更新isonline字段为True
    rs.update
    rs.close '关闭记录集
    Set rs = nothing '注销记录集
    Response.Redirect("index1.asp") '跳转到index1.asp
else '判断是否存在用户名, 如果存在, 说明密码输入错误
    sql = "Select * from users where [user] = '" & user & "'"
    Set rs1 = Server.CreateObject("ADODB.RecordSet")
    rs1.open sql ,conn, 1,1
    if not rs1.eof then '如果存在该记录, 提示返回要求重新输入密码
        Response.Write("<script>alert('你输入的密码不正确,请重新输入密码!');history.
go(-1);</script>")
        Response.end()
    end if
end if
end if
%>

```

17.5.3 禁止恶意用户登录

聊天室是一个公共场所, 由于面向所有用户, 因此不可避免地会混入一些恶意用户。为了防止恶意用户对聊天室公共秩序的破坏, 管理员应该拥有对恶意用户禁止登录的权力。为了实现这个功能, 读者可以通过下面几个主要步骤来实现。

- 01** 在数据库 chat.mdb 的 users 表中增加一个 isclose 字段, 数据类型为“是/否”, 默认值为 0, 表示默认用户不禁止登录。
- 02** 打开 showname.asp 文件, 找到“重复区域”服务器行为控制显示的代码, 用一个条件语句判断当前用户是否是管理员, 本例默认管理员为 admin。如果是管理员, 则显示管理操作, 否则不显示管理操作链接。代码如下:

```

<%
If Session("user") <> "admin" then
%>

<%
While ((Repeat1_numRows <> 0) AND (NOT Recordset1.EOF))
%>
    <tr>
        <td><%= (Recordset1.Fields.Item("user").Value) %></td>
    </tr>
    <%
        Repeat1_index=Repeat1_index+1
        Repeat1_numRows=Repeat1_numRows-1
        Recordset1.MoveNext()
    <%
        Wend
    %>
    <%
    elseif Session("user") = "admin" then
    %>
    <%

```

```

While ((Repeat1__numRows <> 0) AND (NOT Recordset1.EOF))
%>
    <tr>
        <td><%= (Recordset1.Fields.Item("user").Value) %>
        <a href="admin.asp?action=close&id=<%= (Recordset1.Fields.Item("id").Value) %>"
target="_top" >禁</a>
    </td>
</tr>
<%
Repeat1__index=Repeat1__index+1
Repeat1__numRows=Repeat1__numRows-1
Recordset1.MoveNext()
Wend
%>

<%
end if
%>

```

03 新建 admin.asp 文件, 该文件将根据管理员在 showname.asp 页面内单击某个用户的“禁”超链接而进行后台操作。代码如下:

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="conn.asp" -->
<% '禁止用户登录
if Request.QueryString("action") = "close" then
    id = Request.QueryString("id")
    sql = "select * from users where id = " & id
    rs.open sql, conn, 1, 3
    theuser = rs("user")
    rs("isclose") = true
    rs.update
    rs.close '关闭记录集
    Set rs = nothing '注销记录集
    sql = "Insert Into chat([user], to_who, with_what) values('管理员','大家','"& theuser
    &"已经被禁止登录') "
    conn.execute sql
    conn.close
    set conn=nothing
    Response.Redirect "index1.asp"
end if
%>

```

在上面代码中, 先判断用户的行为是什么, 如果是禁止某个用户登录操作, 则进入下面代码执行。在条件语句中先获取禁止用户的 id 编号, 然后根据该 id 编号查询记录集, 并设置字段 isclose 为 True。同时在 chat 表中插入一条记录, 提示禁止某位用户登录。最后跳转到 index1.asp 文件。

04 打开 showname.asp 和 postmsg.asp 文件, 在【绑定】面板中修改前面定义的记录集。方法是双击打开 Recordset1 记录集, 在打开的【记录集】对话框中切换到高级状态, 然后在 SQL 字符串中添加一个条件。代码如下:

```

SELECT *
FROM users
WHERE isonline = MMColParam and isclose = false
ORDER BY logintime DESC

```

这样就会禁止被禁止用户无法继续发言。

05 打开 index.asp 文件, 切换到【代码】视图, 增加下面代码判断登录用户是否被禁止登录, 如果是则提示错误并返回。代码如下:

```

<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
<% '限制被禁止用户登录
if Request.Form("MM_insert") = "form1" then '判断是否是提交表单
    '打开数据库连接
    set conn=server.createobject("adodb.connection")
    conn.open MM_conn_STRING
    set rs=server.createobject("adodb.recordset")
    '用表单中用户输入的用户名以及字段isclose为true查询数据库
    user = Request.Form("user")
    sql = "Select * from users where [user] = '" & user & "' and isclose = true"
    rs.open sql ,conn, 1,1
    if not rs.eof then '判断该用户名是否被禁止
        Session("user") = "" '清空该用户的阶段变量信息
        Response.Write("<script>alert('你已经被禁止登录!');history.go(-1);</script>")
        Response.End()
    end if
    conn.close
    set conn = nothing
end if
%>

```

17.5.4 踢人

踢人也是聊天室管理员一项常规管理工作，它主要针对在聊天中说话不恰当，或者存在恶意攻击行为的用户，管理员临时把该用户从聊天室中踢出去，禁止该用户再继续发言。踢出去不等于禁止登录，如果踢出去之后，该用户下次还可以继续登录。

踢人功能实现比较简单，只需要在 showname.asp 文件中添加一个超链接。代码如下：

```

<td>
<%= (Recordset1.Fields.Item("user").Value) %>
<a href="admin.asp?action=close&id=<%= (Recordset1.Fields.Item("id").Value) %>"
target=" blank" >禁</a>
<a href="admin.asp?action=kick&id=<%= (Recordset1.Fields.Item("id").Value) %>"
target=" _top" >踢</a>
</td>

```

然后，打开 admin.asp 文件，输入下面代码即可。

```

<% '踢人
if Request.QueryString("action") = "kick" then
    id = Request.QueryString("id")
    sql = "select * from users where id = " & id
    rs.open sql, conn, 1, 3
    theuser = rs("user")
    rs("isonline") = false
    rs.update '更新记录集
    rs.close '关闭记录集
    Set rs = nothing '注销记录集
    sql = "Insert Into chat([user], to_who, with_what) values('管理员','大家','"& theuser
    &"已经被踢出聊天室!')"
    conn.execute sql
    conn.close
    set conn=nothing
    Response.Redirect "index1.asp"
end if
%>

```

在上面代码中，先根据被踢用户的 id 编号查询记录集，然后把字段 isonline 设置为 false，强制该用户下线，离开聊天室，最后在聊天室内发出一个公告，告诉大家该用户已经被踢出

聊天室。演示效果如图 17.34 所示。



图 17.34 禁止登录和踢人功能演示效果

17.5.5 私聊

私聊是聊天室功能的深化，因为聊天室是一个公共聊天场所，如果用户想与某位用户单独聊天，就涉及聊天信息的处理问题，下面就来解决这个问题。

01 在 chat 表中新建一个字段 isprivate，数据类型为“是/否”，默认值为 0，表示默认用户不进行私聊。

02 打开 postmsg.asp 文件，在聊天信息发布页面中增加一个复选框，方便用户选择是否私聊，代码如下：

```
私聊☐
```

03 在【服务器行为】面板中修改“插入记录”服务器行为，修改如图 17.35 所示。

04 打开 showmsg.asp 文件，切换到【代码】视图，找到聊天信息显示所在的单元格行，增加一个条件语句判断每条聊天信息是否是私聊信息，如果是则根据 to_who 字段与用户的 Session 变量是否一致，来判断是否将该条私聊信息向当前用户公开。详细代码如下：

```
<%
While ((Repeat1_numRows <> 0) AND (NOT Recordset1.EOF))
    if Recordset1("isprivate") = false then '判断是不是公聊记录
%>
|  |  |
| --- | --- |
| <div align="center" class="STYLE2"><%= (Recordset1.Fields.Item("user").Value) %></div></td>   | |

```

```

"<%= (Recordset1.Fields.Item("with_gif").Value) %>" width="200" border="0" ></a>
<%end if %>
</td>
<td width="14%" align="left"><%= (Recordset1.Fields.Item("post_time").Value) %></td>
</tr>

<%
elseif Recordset1("isprivate") = true and Recordset1("to_who") = Session("user") then
'判断是不是私聊记录, 且是针对当前用户说的记录
%>

<tr>
<td width="8%" height="24" align="left"><div align="center" class="STYLE2"><%=
(Recordset1.Fields.Item("user").Value) %></div></td>
<td width="2%" align="left" ><span class="style1">对</span></td>
<td width="8%" align="left"><div align="center" class="STYLE2"><%= (Recordset1.
Fields.Item("to_who").Value) %></div></td>
<td width="12%" align="left" ><span class="style1">悄悄说:</span></td>
<td width="50%" align="left" class="STYLE2"><%= (Recordset1.Fields.Item("with_
what").Value) %>

<% if Recordset1.Fields.Item("with_url").Value<>" then %>
<a href="<%= (Recordset1.Fields.Item("with_url").Value) %>" target="_black">
<%Response.Write("【点歌请收听】") %></a>
<%elseif Recordset1.Fields.Item("with_gif").Value<>" then%>
<a href="<%= (Recordset1.Fields.Item("with_gif").Value) %>" target="_blank">" width="200" border="0" ></a>
<%end if %>
</td>
<td width="14%" align="left"><%= (Recordset1.Fields.Item("post_time").Value) %>
</td>
</tr>

<%
End If
%>

<%
Repeat1_index=Repeat1_index+1
Repeat1_numRows=Repeat1_numRows-1
Recordset1.MoveNext()
Wend
%>

```

05 这样, 当用户想与某位室友私聊时, 该聊天信息只能被指定室友看见, 演示效果如图 17.36 所示。



图 17.35 修改【插入记录】对话框



图 17.36 私聊效果

17.5.6 离开提示

当某位用户离开聊天室之后，为了避免其他用户还以为该用户呆在聊天室，不妨在离开时，提示以下其他用户。要实现该功能只需打开 `logout.asp` 文件，在其中增加下面代码。

```
<!--#include file="Conn.asp" -->
<%
sql="update users set isonline = false where user='"&session("user")&"'"
conn.execute sql
'离开提示
sql = "Insert Into chat([user], to_who, with_what) values('"& Session("user") &"', '
大家','再见,我要睡觉了! ')"
conn.execute sql
conn.close
set conn=nothing
Session("user")=""
Response.redirect "index.asp"
%>
```

17.5.7 清除过时聊天信息

聊天室是一个大型信息制造工厂，由于这些即时聊天信息没有长时间保存的价值，因此管理员可以根据情况随时清除过时的聊天信息。本例设置了 3 个超链接，方便用户快速清除 1 天前、2 天前或者 3 天前所有的聊天信息。实现该功能的操作步骤如下。

01 打开 `topmsg.asp` 文件，在其中增加 3 个超链接，并设置操作权限。限制该操作链接只对管理员才可见。代码如下：

```
<%
If session("user") = "admin" Then
%>
<a href="admin.asp?action=c1e1" target="_top" >清除1天前聊天信息</a><br>
<a href="admin.asp?action=c1e2" target="_top" >清除2天前聊天信息</a><br>
<a href="admin.asp?action=c1e3" target="_top" >清除3天前聊天信息</a><br>
<%
end if
%>
```

02 打开 `admin.asp` 文件，添加下面代码实现聊天信息清除操作。

```
<%清除1天前所有聊天信息
if Request.QueryString("action") = "c1e1" then
    sql = "select * from chat"
    rs.open sql, conn, 1, 3
    while not rs.eof
        if DateDiff("d", rs("post_time"), Now) > 1 then
            rs.delete
        end if
        rs.movenext
    wend
    rs.close
    set rs=nothing
    Response.Redirect "index1.asp"
end if
%>
<%清除2天前所有聊天信息
if Request.QueryString("action") = "c1e2" then
    sql = "select * from chat"
    rs.open sql, conn, 1, 3
    while not rs.eof
        if DateDiff("d", rs("post_time"), Now) > 2 then
```

```

        rs.delete
    end if
    rs.movenext
wend
rs.close
set rs=nothing
Response.Redirect "index1.asp"
end if
%>
<% 清除3天前所有聊天信息
if Request.QueryString("action") = "cle3" then
    sql = "select * from chat"
    rs.open sql, conn, 1, 3
    while not rs.eof
        if DateDiff("d", rs("post_time"), Now) > 3 then
            rs.delete
        end if
        rs.movenext
    wend
    rs.close
    set rs=nothing
    Response.Redirect "index1.asp"
end if
%>

```

在上面代码中运用了 VBScript 的 DateDiff 时间函数, 该函数可以循环判断每条记录的发布时间与现在时间之间的天数, 该函数的使用可以参阅第 4.12.4 节详细介绍。

17.6 实践总结

本章详细介绍了聊天室的设计方法, 并对一些技术难点进行了突破。在实际应用中, 一般的聊天室还会提供更多功能, 例如, 悄悄话、分屏、清屏、设置文字显示属性、设置心情图片等功能。其实要实现这样的功能很简单, 就是将相关的 HTML 标签插入到要说的话中, 并在聊天页面显示出来即可。由于设计时需要频繁地读取数据库数据, 可能会造成网络堵塞, 在实际应用中, 一般采用的是将聊天记录写入到数据库, 但显示的时候并不是从数据库中读取记录, 而是利用 Application 方法来显示聊天记录。

通过本章的学习, 相信读者对于聊天室功能的实现有了基本的掌握, 在此基础结合 ASP 技术定能制作功能更完善的聊天室系统。

第 18 章

Dreamweaver

BBS 论坛模块

论坛简称 BBS (Bulletin Board Service, 公告牌服务), 是互联网上的一种电子信息服务系统。它提供了一块公共电子白板, 每个用户都可以在上面自由发布信息, 或者提出问题。BBS 最早产生于 20 世纪 70 年代末, 它为用户提供了四大主要功能: 发布新闻; 发布有价值的资源、服务信息; 发布个人思想、感悟、心情描述和经历等内容; 针对其他互联网用户的提问, 提供者根据自身知识、技能经验作出的回答。BBS 这四大主要功能经过不断演化和整合, 衍生出现在流行的网络应用系统: 新闻发布系统、电子商务系统、博客系统和威客系统 (Witkey, 指能够让智慧体现价值的人)。

论坛是一个很大的应用系统, 目前国内有很多大型、成熟的论坛网站, 而且很多论坛网站功能还在不断扩展, 有的已经形成了一个庞大的社区, 例如, 西陆论坛、天涯社区、新浪论坛、中华网论坛、网易社区、搜狐社区、21CN 社区、CSDN 技术社区等。本实例将演示论坛的基本设计思路和基本功能的实现方法。

技术要点

- 动态图片和图片多条件显示
- 记录集导航条
- 多表查询
- 页面信息交互

18.1 程序设计分析

BBS 最早是用来公布股市价格等信息的, 早期的 BBS 功能比较简陋, 连文件传输的功能都没有, 而且只能在苹果计算机上运行。这与街头和校园内的一般公告板性质相同, 只不过是通过电脑来传播或获得消息而已。一直到个人计算机开始普及之后, 开始有人尝试将苹果计算机上的 BBS 转移到个人计算机上, BBS 才开始渐渐普及开来。近些年来, BBS 的功能得到了很大的扩充。目前, 通过 BBS 系统可以帮助用户随时了解国际最新的咨询信息, 也可以通过 BBS 系统来和别人讨论计算机软件、硬件、Internet、多媒体、程序设计以及医学等各种有趣的话题, 更可以利用 BBS 系统来刊登一些征友、买卖、声明等启事。

本章将介绍论坛的基本结构和设计思路, 包括论坛主题发布和管理、帖子发布和管理。整个实例设计流程如图 18.1 所示, 实例显示效果如图 18.2 所示。

要实现本例功能, 读者应该学习和掌握下面技术难点。



图 18.1 论坛流程图

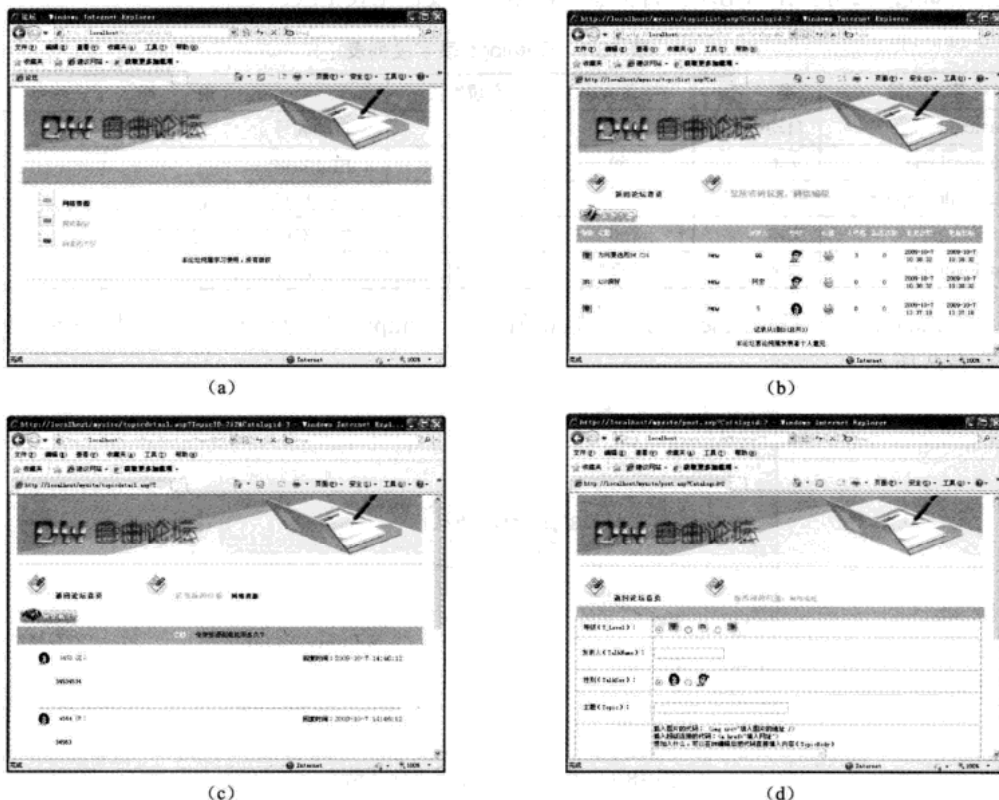


图 18.2 论坛效果图

(a) 论坛栏目页面；(b) 论坛主题页面；(c) 帖子显示页面；(d) 发布主题页面

- 页面之间跳转和参数传递。可以利用超链接参数传递技术，并通过 URL 参数定义记录集。
- 动态显示图片。利用动态表单技术加以实现。
- 多表查询。利用 SQL 语句实现多表查询，也可以定义多个记录集来实现多表信息的查询。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，如 mysite（详细说明可以参阅第 2.4.2 节介绍）。

启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”（详细设置说明可以参阅第 2.5.2 节介绍）。

把本书光盘中的“18”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

18.2 数据库设计

本例数据库 data.mdb 共包括 3 个数据表: Catalogs、TalkArea 和 topic。

18.2.1 Catalogs 表

Catalogs 表用来存储论坛栏目信息, 数据结构设计见表 18.1。

表 18.1 Catalogs 表字段列表

字 段	类 型	字段大小	必填字段	说 明
Catalogid	自动编号	-	-	自动编号
CatalogName	文本	50	否	栏目名称
Catalogdescription	备注	-	-	描述栏目备注信息
pic	文本	50	否	栏目对应的显示图片地址
Catalogorder	数字	长整型	否	栏目的显示顺序

在 Catalogs 表中设置 pic 字段的“默认值”为“"http://\"", 要注意用引号引起表示文本, 如图 18.3 所示。

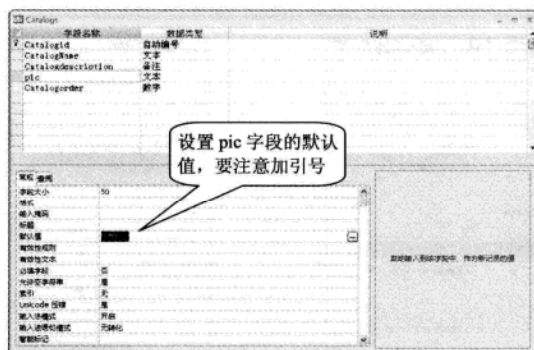


图 18.3 设置 pic 字段的“默认值”

18.2.2 TalkArea 表

TalkArea 表用来存储发布的帖子信息, 该表数据结构设计见表 18.2。

表 18.2 TalkArea 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
TalkId	自动编号	-	-	-	自动编号
TalkBody	备注	-	-	-	帖子内容
TalkName	文本	50	否	是	发言人姓名
TalkSex	文本	50	否	是	发言人的性别
TalkEmail	文本	50	否	是	发言人的邮箱
TalkURL	文本	50	否	是	发言人的主页地址
TalkDate	日期/时间	-	-	-	帖子发布时间
WhichTopic	数字	长整型	否	-	主题编号
Catalogid	数字	长整型	否	-	栏目编号

在该表中设置 TalkSex 字段的默认值为 “images/”，TalkURL 字段的默认值为 “http://”。

18.2.3 topic 表

topic 表用来存储主题信息，该表数据结构设计见表 18.3。

表 18.3 topic 表字段列表

字 段	类 型	字段大小	必填字段	允许空字符串	说 明
TopicID	自动编号	-	-	-	自动编号
Topic	文本	50	否	是	主题标题
TopicBody	备注	-	-	-	主题详细内容
TalkName	文本	50	否	是	发布人的姓名
TalkSex	文本	50	否	是	发布人的性别
TalkEmail	文本	50	否	是	发布人的邮箱
TalkURL	文本	50	否	是	发布人的主页地址
TalkMood	文本	50	否	是	发布人的表情
T_Level	文本	50	否	是	主题等级 PostDate
PostDate	日期/时间	-	否	-	发布时间
UpdateDate	日期/时间	-	否	-	更新时间
ReplyTimes	数字	长整型	否	-	回复次数
ClickTimes	数字	长整型	否	-	点击次数
Catalogid	数字	长整型	否	-	栏目编号

分别设置 TalkMood 和 TalkSex 字段的默认值为 “images”，设置 TalkURL 字段的默认值为 “http://”，设置 PostDate 和 UpdateDate 的默认值为 “Now()”，实现自动插入当前时间。

18.3 页面设计

本实例共有 7 个文件，具体说明见表 18.4。

表 18.4 系统文件列表

文 件 名	说 明
index.asp	论坛栏目页面
topiclist.asp	论坛主题列表页面
topicdetail.asp	论坛帖子显示页面
Reply.asp	发布帖子页面
post.asp	发布主题页面
post_success.asp	发布主题成功页面
Reply_success.asp	发布帖子成功页面

下面就几个主要页面设计进行介绍。

18.3.1 论坛栏目页面

index.asp 页面设计效果如图 18.4 所示。它只需要建立一个简单的表格，并插入 logo 图像

占位符即可。

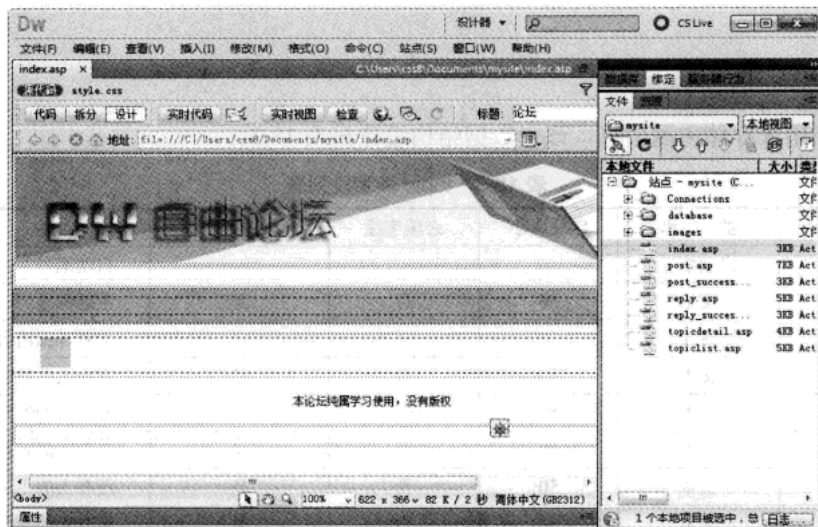


图 18.4 index.asp 页面设计效果

18.3.2 主题列表页面

topiclist.asp 页面设计效果如图 18.5 所示。在设计表格时用户要考虑周全，每行单元格的个数以及每个单元格的宽度都要设计好，避免修改麻烦，同时要在页面底部留出一个表格用于插入导航条。



图 18.5 topiclist.asp 页面设计效果

18.3.3 帖子列表页面

topicdetail.asp 文件主要用来显示某个主题里发布的帖子，设计效果如图 18.6 所示。



图 18.6 topicdetail.asp 页面设计效果

18.3.4 发布页面

post.asp 文件主要用来发布主题，设计效果如图 18.7 所示。

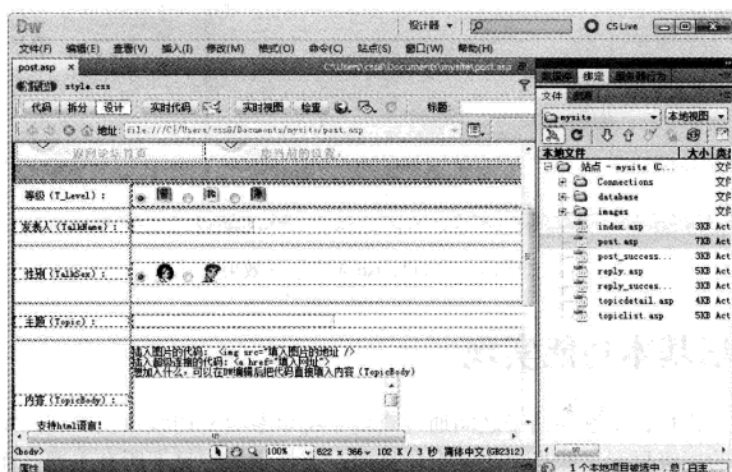


图 18.7 post.asp 页面设计效果

本页面涉及表单制作，主要包括 8 个表单对象，属性设置见表 18.5。

表 18.5 发布主题表单设置列表

对 象	名 称	类 型	设置说明
表单	form1	表单	【动作】先不设置，设置服务器行为后，系统自动插入；【方法】为“默认”
单选/菜单组	T_Level	单选按钮	选定值分别设置为“images/LEVEL01.GIF”、“images/LEVEL02.GIF”和“images/LEVEL03.GIF”
文本框	TalkName	单行文本	-
单选/菜单组	TalkSex	单选按钮	选定值分别设置为“images/male.gif”和“images/Female.gif”

续表

对 象	名 称	类 型	设置说明
文本框	Topic	单行文本	设置【字符宽度】为 40
文本框	TopicBody	多行文本	设置【字符宽度】为 50,【行数】为 10,【换行】为“默认”,
单选/菜单组	TalkMood	单选按钮	选定值分别设置为“images/MOOD01.gif”、“images/MOOD02.gif”、 images/MOOD03.gif、images/MOOD04.gif、images/MOOD05.gif、 images/MOOD06.gif 和 images/MOOD07.gif”
按钮	Submit	提交按钮	【值】设置为“发送”,【动作】为“提交表单”
按钮	Submit2	重置按钮	【值】设置为“重写”,【动作】为“重置表单”

Reply.asp 文件主要用来发布帖子,如图 18.8 所示。本页面涉及表单制作,属性设置见表 18.5。



图 18.8 Reply.asp 页面设计效果

18.4 论坛基本功能实现

本例论坛的栏目、主题和帖子之间通过查询字符串参数密切联系在一起,实现栏目、主题和帖子之间的层级显示,即当单击某个栏目后,会显示该栏目下所有主题;而当单击某个主题,则会显示该主题下所有帖子。该功能是论坛首先应该解决的技术点,本节将详细介绍如何实现栏目、主题和帖子之间的这种联系。

18.4.1 定义自定义连接字符串

Dreamweaver CS5 提供了两种可视化数据库连接方法:DSN(数据源名称)建立数据库连接和定义自定义连接字符串实现数据库连接。DSN 方法必须保证远程服务器与本地机同属一台计算机,否则就会出现错误。如果读者在一台计算机中建立的 DSN 数据连接,转移到另一台计算机中就不能使用,因此 DSN 建立数据库连接的方法灵活性较差。

下面介绍用自定义连接字符串实现数据库连接的方法,具体步骤如下。

01 在 Dreamweaver CS5 中,选择【窗口】|【绑定】菜单命令,打开【绑定】面板。

- 02 单击【绑定】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【自定义连接字符串】命令，如图 18.9 所示。
- 03 打开【自定义连接字符串】对话框，如图 18.10 所示。在【连接名称】文本框中输入连接名称，如 conn；在【连接字符串】文本框中输入连接字符串，字符串代码如下（更详细说明可以参见第 5.4.3 节介绍）：

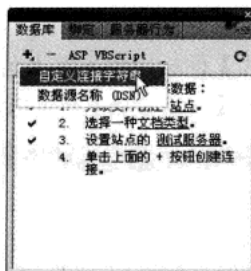


图 18.9 选择【自定义连接字符串】命令

```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath("database/data.mdb")
```



提示

Provider 指定数据库的 OLE DB 提供程序。如下面分别是 Access、SQL Server 和 Oracle 数据库的常用 OLE DB 提供程序的参数：

```
Provider=Microsoft.Jet.OLEDB.4.0;
Provider=SQLOLEDB;
Provider=OraOLEDB.
```

- 04 单击【测试】按钮可以测试连接是否成功，当弹出如图 18.11 所示的提示对话框，表示连接成功。

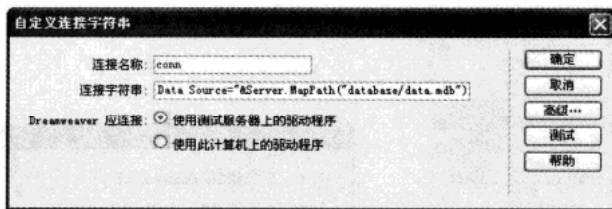


图 18.10 【自定义连接字符串】对话框

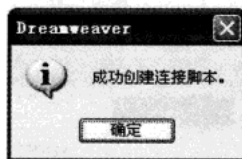


图 18.11 提示对话框

- 05 单击【确定】按钮，完成与指定数据库的连接。

18.4.2 栏目显示

在论坛中，一个栏目就相当于一个频道，或者是一类话题模块，即相同主题模块。下面介绍栏目显示的制作方法，具体操作步骤如下。

- 01 定义一个记录集。打开 index.asp 文件，选择【窗口】|【绑定】菜单命令，打开【绑定】面板。
- 02 单击【绑定】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【记录集（查询）】菜单命令，打开【记录集】对话框，设置如图 18.12 所示，并设置 Catalogorder 字段以降序排列。
- 03 把光标置于要显示栏目图片的位置，选择【插入记录】|【图片】菜单命令，打开【选择图像源文件】对话框，在对话框中的【选取文件名自】选项中选择【数据源】单选按钮，如图 18.13 所示。
- 04 在【域】列表框中选择要插入的动态数据字段，如图 18.13 所示。同时，在【URL】文本框中立即显示绑定的代码。
- 05 在 catalog 记录集中选择 CatalogName 字段，并把它插入到页面中栏目名称的位置，如图 18.14 所示。

- 06** 定义重复区域。在页面中选中绑定字段的那一行，要注意不要多选或少选。在【服务器行为】面板中单击左上角的 \oplus 按钮，在弹出的下拉菜单中选择【重复区域】菜单命令，打开【重复区域】对话框，选择刚刚定义的记录集，设置显示全部记录，如图 18.15 所示。

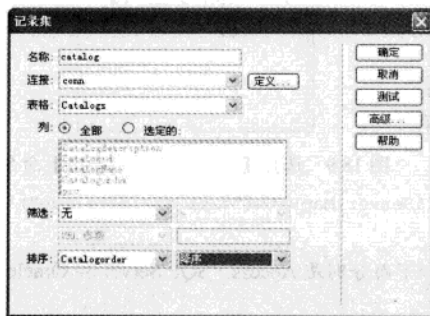


图 18.12 设置查询条件

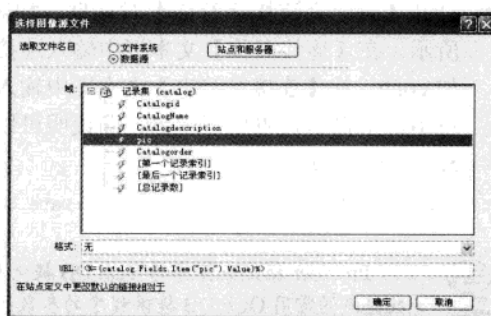


图 18.13 选择图像动态数据源



图 18.14 插入栏目动态数据源

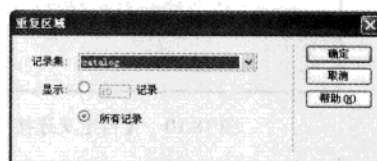


图 18.15 设置重复区域

- 07** 单击【确定】按钮关闭对话框，完成栏目页面的显示功能。
- 08** 选择动态栏目名称，然后在【属性】面板中单击【链接】文本框右边的【选择文件】按钮 $\left[\text{选择文件} \right]$ ，在打开的【选择文件】对话框中选择要跳转的文件 topiclist.asp，同时设置传递参数为“?Catalogid=<%= (catalog.Fields.Item("Catalogid").Value) %>”。方法是在【选择文件】对话框中单击【参数】按钮打开【参数】对话框，然后在【名称】列中增加传递变量，并在【值】中增加动态数据源，如图 18.16 所示。

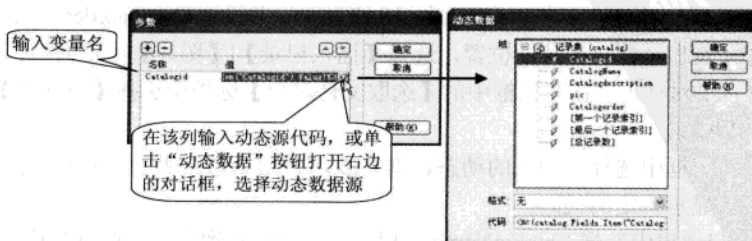

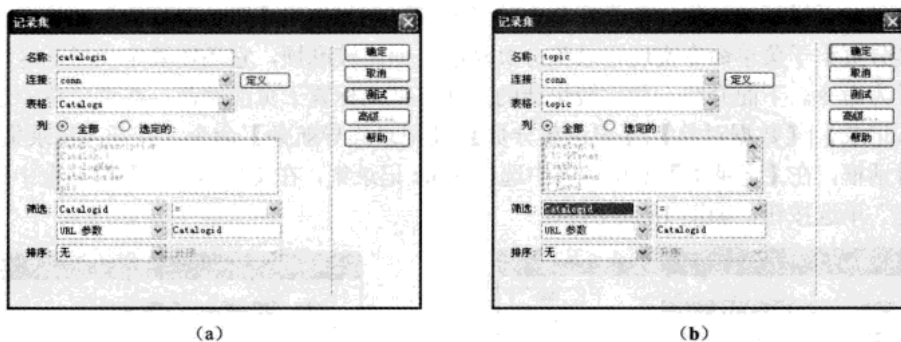


图 18.16 设置动态数据源传递参数

18.4.3 主题列表

topiclist.asp 页面主要显示主题列表。当用户选择具体的论坛栏目之后, 利用传递过来的 id 编号定义记录集, 查询相关主题记录, 然后绑定记录集到页面并进行分页显示。具体操作步骤如下。

- 01** 工作的第 1 步还是定义记录集, 本页需要定义两个记录集, 一个是查询指定栏目的记录, 另一个是查询指定栏目下的所有主题。选择【窗口】|【绑定】菜单命令, 打开【绑定】面板。单击【绑定】面板左上角的  按钮, 在弹出的下拉菜单中选择【记录集 (查询)】菜单命令。
- 02** 打开【记录集】对话框, 分别设置 catalogin 和 topic 记录集, 如图 18.17 所示。



(a)

(b)

图 18.17 定义记录集

(a) catalogin 记录集; (b) topic 记录集

- 03** 在页面各栏目中分别插入记录集字段, 如图 18.18 所示。然后切换到【代码】视图, 在主题名称列后面插入下面代码, 实现根据主题发布时间和点击次数来判断当前帖子是否显示为最新帖子和热门帖子。如果帖子发布的时间是今天, 则显示“最新”提示小图标, 如果该帖子点击超过 50 次, 则显示“热门”提示小图标, 说明该帖子为热门帖子。

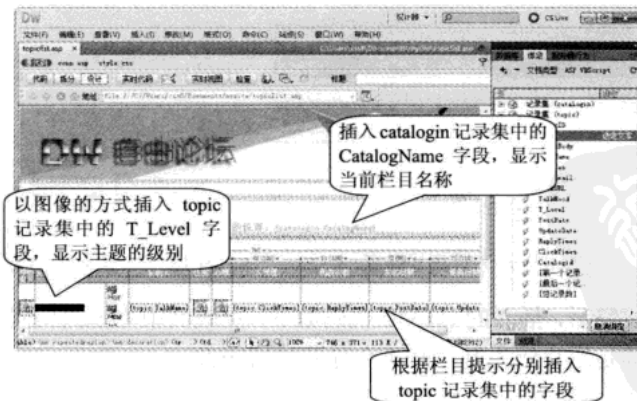


图 18.18 插入记录集


```
<td width="56" valign="middle" class="style2"><p align="left">
<%
clicknum = (topic.Fields.Item("ClickTimes").Value)
postday = (topic.Fields.Item("PostDate").Value)
today =day(date())
if (clicknum>50) then
%>
```

```


<%
end if
if (day(postday)=today) then
%>

<%end if %>
</p></td>

```

04 定义重复区域。在页面中选中绑定字段所在的行，在【服务器行为】面板中单击左上角的  按钮，在弹出的下拉菜单中选择【重复区域】菜单命令，打开【重复区域】对话框。选择刚刚定义的记录集，设置每页显示 10 条记录，如图 18.19 所示。

05 单击【确定】按钮关闭对话框，即可实现记录集重复显示。

06 定义分页显示记录集。前面章节中曾经介绍过如何使用【服务器行为】面板中的【记录集分页】子菜单命令实现记录集分页功能，但比较麻烦，它还需要定义显示区域服务器行为命令。下面介绍一种比较快捷的方法：把光标置于页面底部，在菜单栏中选择【插入记录】|【数据对象】|【记录集分页】|【记录集导航条】命令，打开【记录集导航条】对话框，在【记录集】下拉列表中选择 topic 记录集，在【显示方式】对话框中选择“文本”单选按钮，如图 18.20 所示。

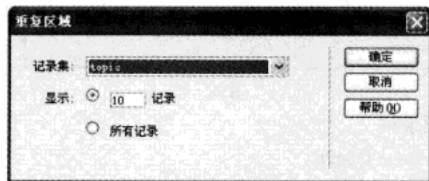


图 18.19 设置重复区域

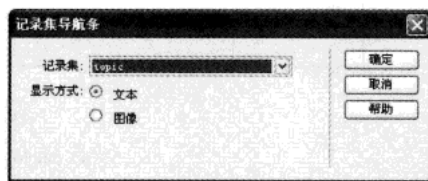


图 18.20 【记录集导航条】对话框

07 设置完毕，单击【确定】按钮，即可在指定位置插入一个导航条，如图 18.21 所示。

如果选择【显示方式】中的“图像”单选按钮，则显示如图 18.22 所示的效果。当然，用户也可以切换到【代码】视图，改变导航条显示的图像样式和效果。

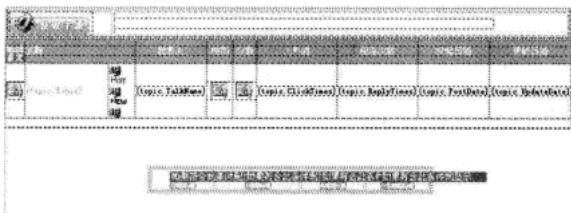


图 18.21 插入的文本样式记录集导航条



图 18.22 插入的图像样式记录集导航条

插入记录集导航条后，插入的 HTML 代码如下，更改其中方框内代码即可实现导航条图像的不同效果，前提是用户应先制作好 4 个导航按钮图像。

```

<table border="0">
<tr>
<td><% If MM_offset <> 0 Then %>
<a href="<%=MM_moveFirst%>">
<% End If ' end MM_offset <> 0 %>
</td>
<td><% If MM_offset <> 0 Then %>
<a href="<%=MM_movePrev%>">
<% End If ' end MM_offset <> 0 %>
</td>

```

```

<td><% If Not MM_atTotal Then %>
    <a href="<%=MM_moveNext%>"></a>
<% End If ' end Not MM_atTotal %>
</td>
<td><% If Not MM_atTotal Then %>
    <a href="<%=MM_moveLast%>"></a>
<% End If ' end Not MM_atTotal %>
</td>
</tr>
</table>

```

08 设置完毕,单击【确定】按钮,即可在指定位置插入一个导航条,如图 18.22 所示。然后在选中插入的记录集导航条表格,在属性面板中设置其居中显示即可。

09 在页面的底部插入记录集分页提示信息,如图 18.23 所示。方法是切换到【代码】视图,输入下面代码。

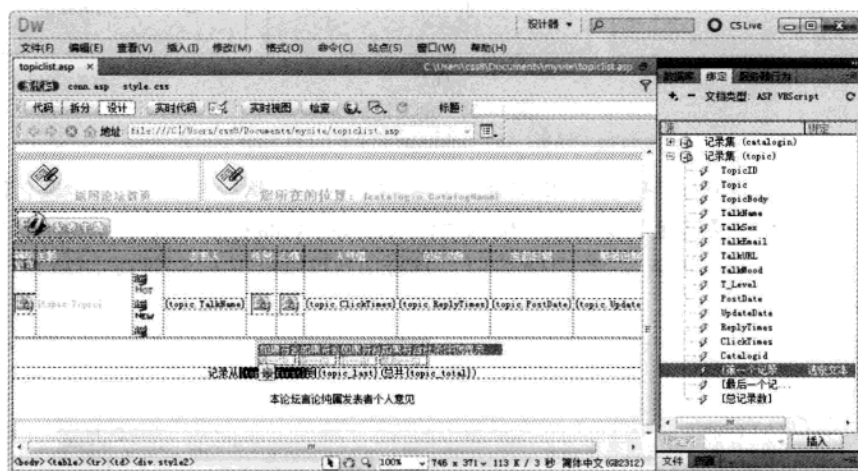


图 18.23 输入页面提示信息

```

div align="center" class="style2" >记录从<%=topic_first%>到<%=topic_last%>(总共<%=
topic_total%>)</div>

```

topic_first、topic_last 和 topic_total 这 3 个变量是在插入记录集导航条时系统定义的。topic_first 和 topic_last 分别用来存储每页的第 1 条记录和最后一条记录在记录集中的位置,topic_total 主要记录记录集的记录数。

10 选中绑定在页面中的 Topic 字段,然后在属性面板的【链接】文本框中设置要跳转的文件,具体设置如下:

```

topicdetail.asp?TopicID=<%= (topic.Fields.Item("TopicID").Value) %> & Catalogid=<%= (to
pic.Fields.Item("Catalogid").Value) %>

```

其中第 1 个方框内为目标文件,第 2 个方框内为要传递的第 1 个参数变量,第 3 个方框内为要传递的第 2 个参数变量。

18.4.4 帖子列表

topicdetail.asp 文件主要显示某个主题下的所有帖子列表。该页面操作与主题列表页面的操作方法相同,具体操作步骤如下。

- 01** 定义记录集。本页需要定义 3 个记录集，一个是查询指定栏目的信息，一个是查询指定栏目下的所有主题，另一个是要查询指定主题下所有帖子。

选择【窗口】|【绑定】菜单命令，打开【绑定】面板。单击【绑定】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【记录集（查询）】菜单命令。

- 02** 打开【记录集】对话框，分别设置 catalog、topicname 和 topic 记录集，如图 18.24 所示。

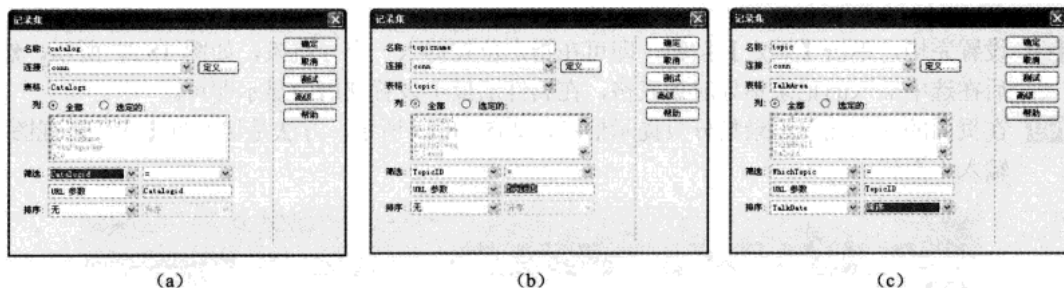


图 18.24 定义记录集

(a) catalog 记录集; (b) topicname 记录集; (c) topic 记录集

- 03** 在页面各栏目中分别插入记录集字段，如图 18.25 所示。然后选中绑定的主题名称字段，在属性面板的【链接】文本框中输入“topiclist.asp?Catalogid=<%= (catalog.Fields. Item ("Catalogid").Value)%>”，详细代码如下，这样用户可以单击主题名称返回到上一级主题列表中。

您当前的位置: <a href="topiclist.asp?Catalogid=<%= (catalog.Fields. Item ("Catalogid"). Value) %>"><%= (catalog.Fields. Item ("CatalogName"). Value) %>

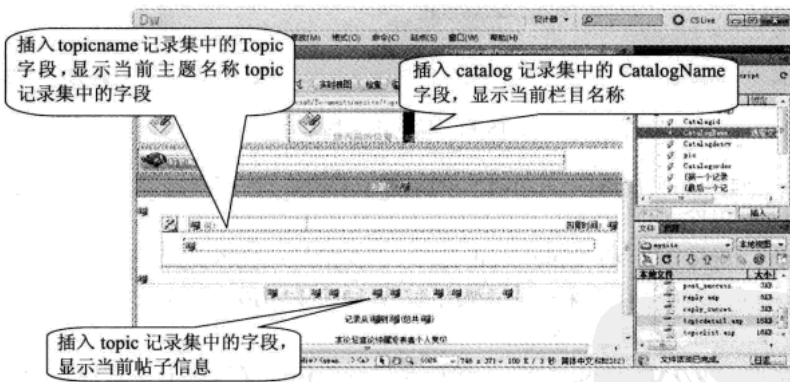


图 18.25 插入记录集

- 04** 定义重复区域。在页面中选中要显示帖子的表格，如图 18.26 所示。在【服务器行为】面板中单击左上角的 \oplus 按钮，在弹出的下拉菜单中选择【重复区域】菜单命令，打开【重复区域】对话框，选择定义的 topic 记录集，设置每页显示 10 条记录。

- 05** 定义分页显示记录集。在菜单栏中选择【插入记录】|【数据对象】|【记录集分页】|【记录集导航条】命令，打开【记录集导航条】对话框，在【记录集】下拉列表中选择 topic 记录集，在【显示方式】对话框中选择【文本】单选按钮，如图 18.27 所示。

- 06** 在页面的底部插入记录集分页提示信息。切换到【代码】视图，输入下面代码即可。

```
<p align="center" class="style2"> 记录从<%= topic_first %>到<%= topic_last %>(总共<%=
=topic_total %>) </p>
```

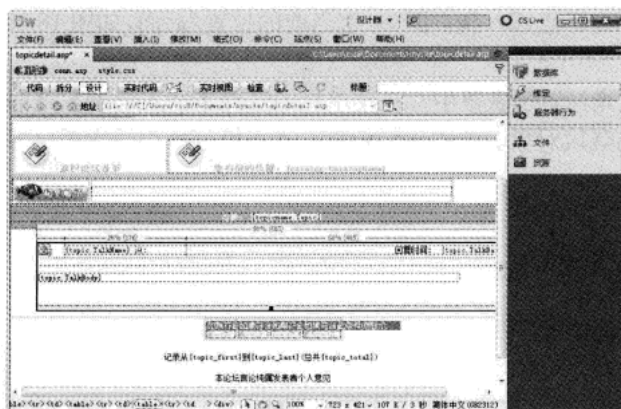


图 18.26 选择要重复的区域

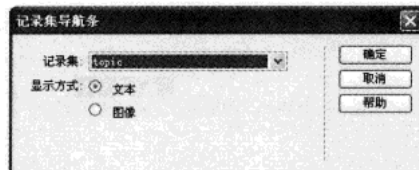


图 18.27 【记录集导航条】对话框

18.4.5 发布主题

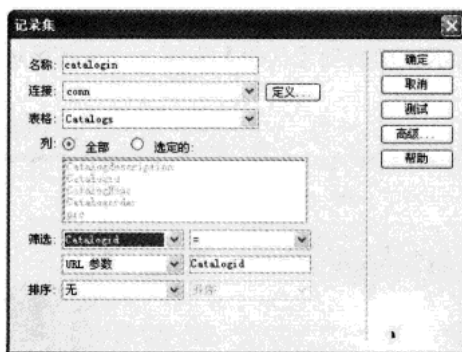
发布主题主要利用“插入记录”服务器行为来实现，但是读者必须根据从主题列表页跳转过来时传递的栏目参数变量定义记录集，把用户输入的信息插入到数据库的 topic 表中，并指定 Catalogid 字段的栏目编号，具体操作步骤如下。

- 01** 先打开 topiclist.asp 文件，选中标题栏中的“发表主题”图片，然后在属性面板中的【链接】文本框中输入下面链接路径。

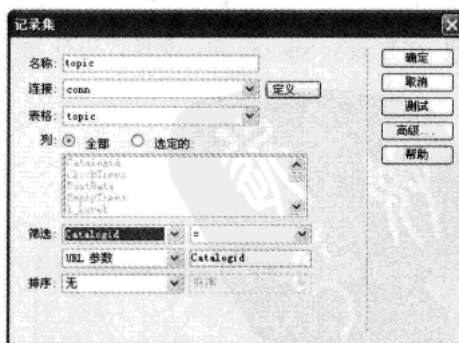
```
post.asp?Catalogid=<%= (catalogin.Fields.Item("Catalogid").Value) %>
```

其中变量 catalogin 用来传递栏目编号，方便目标网页定义并筛选记录集。

- 02** 打开 post.asp 文件，定义 2 个记录集，查询 topic 表和 Catalogs 表中的指定栏目的记录信息。选择【窗口】|【绑定】菜单命令，打开【绑定】面板。
- 03** 单击【绑定】面板左上角的 按钮，在弹出的下拉菜单中选择【记录集（查询）】菜单命令，打开【记录集】对话框，分别设置 catalogin 和 topic 记录集，如图 18.28 所示。



(a)



(b)

图 18.28 定义记录集

(a) catalogin 记录集; (b) topic 记录集

定义 catalogin 记录集的目的就是在标题处“你的位置……”中显示栏目标题，定义 topic 记录集主要是获取 topic 表中主题编号。

- 04** 把 catalogin 记录集中的 CatalogName 字段插入到页面中“您当前的位置”后面,如图 18.29 所示。
- 05** 在页面表单中插入一个隐藏域。把光标置于表单中,在菜单栏中选择【插入记录】|【表单】|【隐藏域】命令,插入一个隐藏域对象。选中该隐藏域,在属性面板中设置隐藏域对象的名称和值,如图 18.30 所示。

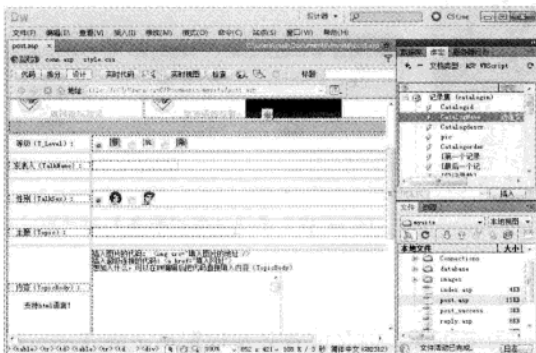


图 18.29 绑定记录集

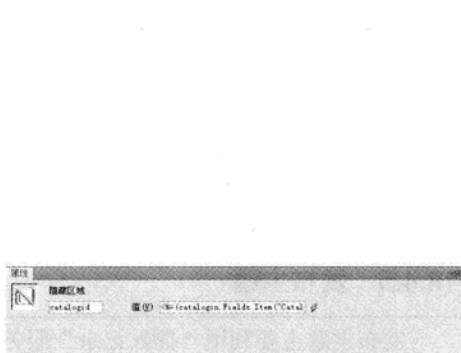


图 18.30 插入隐藏域

插入 Catalogid 隐藏域主要是用来传递栏目编号,这样当向数据库中插入记录时,能保证该插入记录的 Catalogid 字段信息正确,便于其他页面的筛选。

- 06** 在【服务器行为】面板中单击左上角的 按钮,在弹出的下拉菜单中选择【插入记录】菜单命令。
- 07** 打开【插入记录】对话框,在【连接】下拉列表中选择前面定义好的数据源 conn,在【插入到表格】下拉列表中选择要插入的数据表 topic,在【插入后,转到】文本框中设置插入记录成功后要跳转到 post_success.asp 文件,如图 18.31 所示。
- 08** 在【表单元素】列表框中分别设置每个表单对象中的数据,要插入到数据库中 topic 表的哪个字段。在设置【表单元素】列表时,要注意表单对象和数据表字段的对应性。
- 09** 设置完毕,单击【确定】按钮即可实现插入记录功能。
- 10** 打开 post_success.asp 文件,设计如图 18.32 所示。切换到【代码】视图,在头部区域输入下方框内代码。

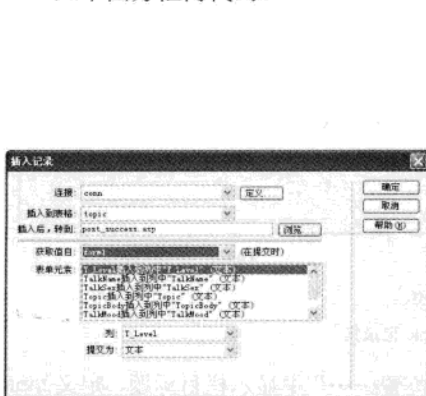


图 18.31 【插入记录】对话框

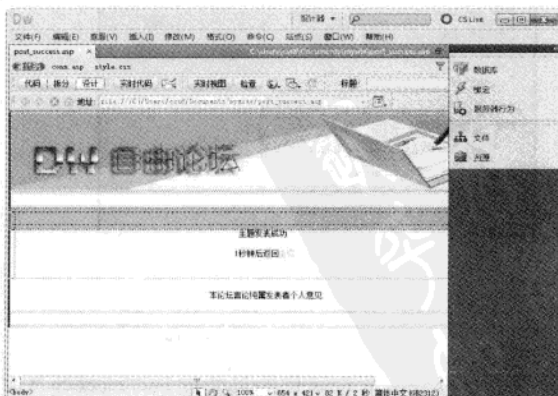


图 18.32 发布主题成功页面设计效果

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title></title>
<link href="css.css" type="text/css" rel="stylesheet" media="all" title="css1"/>
<meta http-equiv="Refresh" content="1;URL=index.asp" />
</head>
```

上面的边框内代码可以自动实现页面跳转,其中 content 变量的第 1 个参数表示几秒后,发生刷新或跳转,URL 参数表示要跳转的页面,如果其值为空,则页面会实现自动刷新,但不跳转。

18.4.6 发帖子

发帖子和发布主题的实现方法基本相同。要注意发布帖子是从帖子列表文件 topicdetail.asp 跳转过来,并传递参数相关的主题编号,如图 18.33 所示。



图 18.33 设置发布帖子图像按钮

其链接代码如下,该代码主要通过一个参数来传递主题编号的动态信息。

```
<a href="Reply.asp?topicid=<%= (topicname.Fields.Item("TopicID").Value) %>"></a>
```

在定义记录集时,最麻烦的问题是条件跨表查询,这需要用户在【记录集】的高级对话框中手写 SQL 代码。

如果要在同一页面内显示数据库中多个表的记录,会涉及从多个表中进行查询,即需要定义多个记录集。例如,在本例的 topicdetail.asp 页面中,就涉及同时要显示栏目标题、主题标题和帖子信息,如图 18.34 所示。为了实现本页所设计的功能,在第 15.4.4 节中同时定义了 3 个记录集,分别从 3 个表中查询记录,如图 18.35 所示。



图 18.34 同时显示多表记录

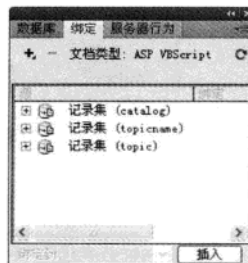


图 18.35 定义多个记录集

下面就来具体介绍如何在 reply.asp 文件中实现跨表查询。具体操作步骤如下。

- 01 在 Dreamweaver CS5 中打开 reply.asp 文件，然后打开【绑定】面板。
- 02 单击【绑定】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【记录集（查询）】菜单命令，打开【记录集】对话框。单击【高级】按钮，切换到高级状态，在 SQL 文本框中输入下面代码，详细设置如图 18.36 所示。

```
SELECT *
FROM Catalogs INNER JOIN topic
ON Catalogs.CatalogId = topic.CatalogId
WHERE TopicID = MMColParam
```

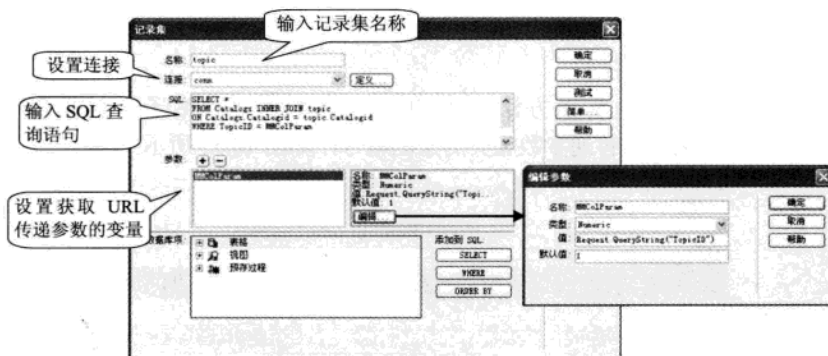


图 18.36 定义记录集



上面 SQL 代码中，SELECT 关键字表示要查询的字段，FROM 关键字表示从哪些表中进行查询，INNER JOIN 关键字表示建立表之间的联系，ON 关键字表示两个表是通过哪个字段建立联系的，且必须保证这些连接字段类型和值要一致，WHERE 关键字表示查询的条件。

- 03 定义混合记录集之后，把记录集 topic 中的 CatalogName 字段绑定到标题位置，如图 18.37 所示。

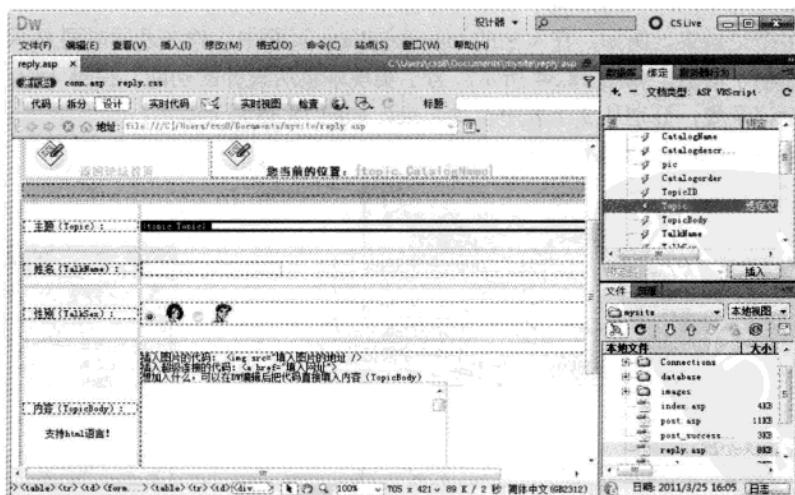


图 18.37 插入记录集

04 模仿上节介绍方法在表单中插入 2 个隐藏域，代码如下：

```
<input type="hidden" id="whichtopic" name="whichtopic" value="<%= (topic.Fields.Item
("TopicID").Value) %>" />
<input type="hidden" id="catalog" name="catalog" value="<%= (topic.Fields.Item
("Catalogs.Catalogid").Value) %>" />
```

05 以同样的方式，利用“插入记录”服务器行为把表单中的数据插入到数据库中，【插入记录】对话框设置如图 18.38 所示。

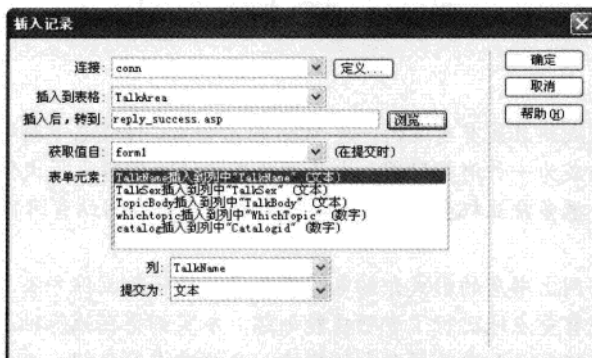


图 18.38 插入记录

18.5 实践总结

本章详细介绍了论坛的设计方法和过程，论坛系统对于技术的要求不是太高，理清分级目录的设计思路，就会发现论坛设计有很多重复的工作。但要设计一个高效的论坛系统，一方面需要提高 ASP 编程能力，另一个很重要因素就是应用 SQL 查询技术。如果了解 SQL 语法使用方法，在操作后台数据库和查询数据时就会变得轻巧高效。

实际中，论坛会按不同的主题分为许多版块，主题的设立依据是大多数用户的要求和喜好。另外，论坛也提供邮件功能，如果需要私下的交流，也可以将想说的话直接发到某人的电子信箱中。当然，建立功能强大齐全的论坛系统还需要用户不断去学习和实践。

第 19 章

Dreamweaver

网上购物（上）

电子商务是比较成熟且最有前途的一类 Web 应用形式，伴随电子商务环境和模式的日渐成熟，网上购物已经成为一种消费习惯。电子商务不但给人们带来巨大便利，而且这种灵活的商业模式也提供了很多就业机会。国内比较知名的电子商务网站有阿里巴巴、淘宝、当当、易趣、卓越等。

本实例将以一个网上书店的形式介绍电子商务网站的一般制作方法。在前面的实例中读者分不同的技术类型有重点地进行了专项技能训练，本实例将在此基础上集中展示一个完整网站的实现过程，包括展示和购物两个大的模块，分两章进行介绍。本章重点讲述网上购物系统的展示部分，包括商品分级展示、购物导航，以及具体商品页面和商品查询制作等内容。

技术要点

- 利用文本框分页导航
- 导航按钮与文本框导航同步
- 商品多级分类展示
- 商品二级页面和详细页面显示
- 栏目导航功能实现
- 站内高级查询
- 商品评级与评论功能
- 网上交互功能实现

19.1 程序设计分析

电子商务可以分为两大类：B2B、B2C。B2B 的全称是 Business to Business，即企业与企业，也可以作为大型商业买卖的交易平台，公司企业可以通过这个平台进行采购、销售、结算等，以实现降低成本，提高效率。但这种平台对性能、服务要求比较高，此类电子商务网站比较典型的有阿里巴巴。B2C 的全称是 Business to Customer，即企业与消费者，它直接面向终端的消费者，不过经营形式可以有多种，例如，一种类似大型超市，里面提供大量的货物商品，消费者可以选择商品，直接在线结账付款，如当当网上书店、卓越网上商城等；还有类似城市里面的大商场，在这个商城里面显示很多柜台或专柜，他们都属于不同的货主，消费者可以根据自己的需求直接到相应柜台购买商品，然后去商城服务台结账；在电子商城中是按类别或经营范围来划分的，例如，新浪的电子商城、263 电子商城等。

不管电子商务网站是何种形式，其基本模式是相同的，即浏览查看商品，然后下订单，

双方确认后付款交货,完成交易。当然根据规模不同,系统功能可大可小,不过一般网上商店应包含下面基本功能。

- 能够为用户提供方便快捷地浏览和查询商品信息的功能。
- 能够有效跟踪用户网上行为,实现准确交易的功能。
- 能够提供完善的后台管理,例如商品管理、用户管理、订单管理、栏目管理等。

本网上购物系统也是按着上述3个基本功能来设计,计划用2章分别进行介绍。本章重点介绍网上购物前台商品展示与查询功能实现。前台系统的最后浏览效果如图19.1所示,程序设计流程如图19.2所示。

要实现前台浏览与查询功能,读者应该学习和掌握下面技术难点。

- 商品的多级分类展示。需要用到记录集嵌套技术。
- 栏目导航功能实现。需要用到记录集嵌套技术。
- 多条件随意查询技术实现。需要熟悉SQL语句及规范。



(a)



(b)



(c)



(d)

图 19.1 浏览和查找信息浏览效果

(a) 系统首页; (b) 商品查询页面; (c) 推荐商品分类; (d) 商品展示详细页面

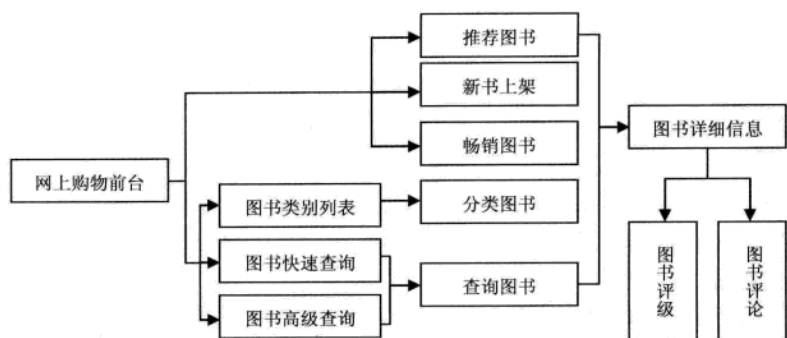


图 19.2 网上购物前台设计流程图

- 商品分类展示和多页导航。
- 用文本框进行分页导航技术。
- 对商品进行评级和评价。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“19-20”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

19.2 数据库设计

一个复杂的电子商务网站对于后台数据库的开销是非常大的。由于涉及网上交易，因此对于数据库结构的设计要求就比较严格了，同时为了数据的安全性，读者在设计数据库时应该考虑数据库的加密和数据加密处理问题。

本例数据库 db.mdb 包括 8 个表，各表存储信息的功能见表 19.1。该数据库存储在 database 文件夹中，然后在 Dreamweaver CS5 的【数据库】面板中数据库连接字符串。为了便于开发，在根目录下新建 conn.asp 文件，输入下面代码可以打开数据库连接：

```
<!--#include file="Connections/conn.asp" -->
<%
'打开数据库连接
dim conn,connstr
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open MM_conn_STRING
%>
```

表 19.1 数据库 db.mdb 各个表功能介绍

表 名	说 明	表 名	说 明
admin	管理员管理信息	product	产品记录详细信息
category	商品一级分类信息	review	用户反馈信息
delivery	网上购物交易方式	sorts	商品二级分类信息
orders	客户订货信息	user	客户资料详细信息

下面就各表结构与字段属性设计进行说明, 见表 19.2~表 19.9。

表 19.2 admin 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
admin	文本	50	否	用户名
password	文本	50	否	用户密码
rank	数字	长整型	-	用户权限

表 19.3 category 表字段列表

字 段	类 型	字段大小	必填字段	说 明
categoryid	自动编号	-	-	自动编号
category	文本	50	否	大类名称 (一级商品分类名称)
first	数字	长整型	-	是否首页显示
categoryorder	文本	50	否	分类排列顺序

表 19.4 delivery 表字段列表

字 段	类 型	字段大小	必填字段	说 明
deliveryid	自动编号	-	-	自动编号
subject	文本	50	否	送货/汇款方式名
fee	数字	单精度型	-	价钱
methord	数字	长整型	-	0 为送货方式 1 为汇款方式
deliveryidorder	数字	长整型	-	顺序

表 19.5 orders 表字段列表

字 段	类 型	字段大小	必填字段	说 明
actionid	自动编号	-	-	自动编号
username	文本	50	否	订单用户名
actiondate	日期/时间	-	-	订单日期
id	数字	长整型	-	产品编号
productnum	数字	长整型	-	产品订购数量
goods	文本	50	否	订单编号
state	数字	长整型	-	状态: 1~5 表示步骤, 6 是表示在购物车
recepit	文本	50	否	收货人
address	文本	100	否	收货地址
postcode	数字	长整型	-	邮编
comments	备注	-	-	留言
paymethord	数字	长整型	-	汇款方式编号
deliverymethord	数字	长整型	-	送货方式编号
sex	数字	长整型	-	性别
paid	数字	单精度型	-	总金额
realname	文本	50	否	姓名
useremail	文本	50	否	EMAIL
usertel	文本	50	否	电话
userid	数字	长整型	-	用户 ID

表 19.6 product 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
name	文本	50	否	书名
author	文本	50	否	作者
introduce	文本	255	否	装
productdate	日期/时间	-	-	出版日期
score	数字	单精度型	-	积分
detail	备注	-	-	详细介绍
content	备注	-	-	目录
rank	数字	长整型	-	用户评级分数
price1	数字	双精度型	-	市场价
price2	数字	双精度型	-	会员价
recommend	数字	长整型	-	推荐
solded	数字	长整型	-	订购次数
viewnum	数字	长整型	-	查看次数
discount	数字	单精度型	-	折扣
sortsid	数字	长整型	-	小类 id
categoryid	数字	长整型	-	大类 id
pic	文本	255	否	缩略图
adddate	日期/时间	-	-	添加日期
ranknum	数字	长整型	-	评级次数
vipprice	数字	双精度型	-	vip 价格
mark	文本	255	否	出版社
type	文本	255	否	ISBN
pagenum	数字	长整型	-	页数
format	文本	50	否	开本
printed	数字	长整型	-	版次
desc	文本	200	否	简介

表 19.7 review 表字段列表

字 段	类 型	字段大小	必填字段	说 明
reviewid	自动编号	-	-	自动编号
id	数字	长整型	-	对应产品编号
title	文本	50	否	标题
reviewdtm	日期/时间	-	-	日期时间
reviewcontent	备注	-	-	评论内容
audit	数字	长整型	-	是否通过审批

表 19.8 sorts 表字段列表

字 段	类 型	字段大小	必填字段	说 明
sortsid	自动编号	-	-	自动编号
sorts	文本	50	否	小类名
sortsorder	数字	长整型	-	顺序
categoryid	数字	长整型	-	对应大类编号
first	数字	长整型	-	是否在首页显示

表 19.9 user 表字段列表

字 段	类 型	字段大小	必填字段	说 明
userid	自动编号	-	-	自动编号
username	文本	50	否	用户名(登录时用)
useremail	文本	50	否	Email
password	文本	50	否	密码
identify	数字	双精度型	-	身份证号
mobile	文本	50	否	手机
userqq	文本	50	否	QQ
adddate	日期/时间	-	-	注册日期
quesion	文本	50	否	密码提示
answer	文本	50	否	密码答案
loginnum	数字	长整型	-	登录次数
lastvst	日期/时间	-	-	最后一次登录时间
recepit	文本	50	否	收货人姓名
city	文本	50	否	城市
address	文本	100	否	地址
postcode	文本	50	否	邮编
usertel	文本	50	否	电话
deliverymethord	数字	长整型	-	送货方式编号
paymethord	数字	长整型	-	汇款方式编号
sex	数字	长整型	-	性别
realname	文本	50	否	姓名
vip	是/否			是否 VIP 用户
book	文本	255	否	系统广播
score	数字	长整型	-	用户积分

19.3 页面设计

本实例网上购物前台商品浏览与查询系统共包括 14 个文件, 具体说明见表 19.10。

表 19.10 商品浏览与查询系统文件列表

文 件 名	说 明
class.asp	商品多级分类详细列表页面
commend.asp	推荐图书列表页面
comment.asp	用户对商品评论页面
Foot.htm	页面底部信息显示页面
Head.htm	页面头部标题和导航页面
hot.asp	畅销图书排行 页面
index.asp	系统首页显示页面
Left.asp	页面左侧的栏目导航和信息显示页面
new.asp	新书上架页面
rank.asp	用户评级页面
rsearch.asp	后台商品查询和显示页面
search.asp	高级查询用户输入信息页面
sub.asp	二级分类展示商品页面
vpro.asp	商品信息展示详细页面

由于前台系统页面设计相对简单,用户参照本书光盘中提供的实例模板就可以轻松理解。下面就几个主要表单页面设计进行简单介绍。

19.3.1 快速查询表单

快速查询表单位于首页上部,它提供两个查询条件,一个是用于输入书名的文本框,另一个是提供图书类别的下拉菜单,如图 19.3 所示。

在该表单中的主要技术难点就是如何实现动态下拉菜单显示,具体操作步骤如下。



图 19.3 快速查询表单效果

- 01 打开 index.asp 文件,首先定义一个 category_list 记录集,用来查询图书类别,【记录集】对话框具体设置如图 19.4 所示。
- 02 插入一个表单,在【属性】面板中设置【动作】为“rsearch.asp”,即设置表单提交数据到 rsearch.asp,设置【方法】为“GET”选项。
- 03 在表单中插入一个文本框,在【属性】面板中命名为“name”,插入一个列表/菜单,命名为“code”,设置【类型】为“菜单”。
- 04 选中 code 下拉菜单,在【属性】面板中单击【动态】按钮,打开【动态列表/菜单】对话框,在【来自记录集的选项】下拉列表中选择“category_list”记录集选项,在【值】下拉列表中选择“categoryid”字段选项,在【标签】下拉列表中选择“category”字段选项,详细设置如图 19.5 所示。

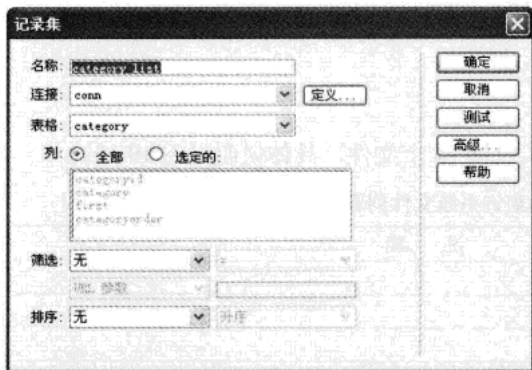


图 19.4 定义记录集



图 19.5 定义动态下拉菜单

- 05 切换到【代码】视图,在下拉列表中增加一个默认选项,代码如下:


```
<option value="" selected>所有图书</option>
```
- 06 为了避免和首页其他记录集混合在一起出错,应切换到【代码】视图,把 category_list 记录集代码段复制到运用它的表单位置前,把关闭记录集代码复制到运用它的表单位置后,删除因定义 category_list 记录集而在代码页顶部自动插入的代码行。


```
<!--#include file="Connections/conn.asp" -->
```

19.3.2 高级查询表单

高级查询相对于快速查询表单而言所能查询的条件增多而且复杂，用户可以根据不同条件在数据库中查询所需要的图书，也可以实现多种组合查询或者模糊查询，找到特定范围的图书。高级查询表单设计如图 19.6 所示。



图 19.6 高级查询表单设计效果

高级查询表单各个对象属性设置见表 19.11。

表 19.11 高级查询表单属性设置列表

对 象	名 称	类 型	设置说明
表单	frmdata	表单	【方法】设置为“GET”，【动作】设置为“rsearch.asp”
文本框	name	单行	-
文本框	author	单行	-
文本框	manufacturer	单行	-
文本框	enableddate	单行	【字符宽度】设置为 12
文本框	expiredate	单行	【字符宽度】设置为 12
文本框	smallprice	单行	【字符宽度】设置为 5
文本框	largeprice	单行	【字符宽度】设置为 5
菜单/列表	code	菜单	【初始化时选定】设置为“所有图书”，动态下拉菜单制作可参考上一小节介绍
选择框	OrderField	单选按钮	【初始状态】设置为“已勾选”，【选定值】设置为“adddate”
选择框	OrderField	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“productdate”
选择框	OrderField	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“price2”
选择框	OrderField	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“vipprice”
选择框	OrderField	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“pagenum”
选择框	OrderField	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“discount”
选择框	Order	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“ASC”
选择框	Order	单选按钮	【初始状态】设置为“未选中”，【选定值】设置为“DESC”
按钮	Submit2	按钮	【动作】设置为“提交表单”，【值】为“开始搜索”

19.3.3 评级和评论表单

评级表单页面设计了一组单选按钮组，设置选项的值分别为 1 到 10，表示 10 个等级，设计效果如图 19.7 所示。

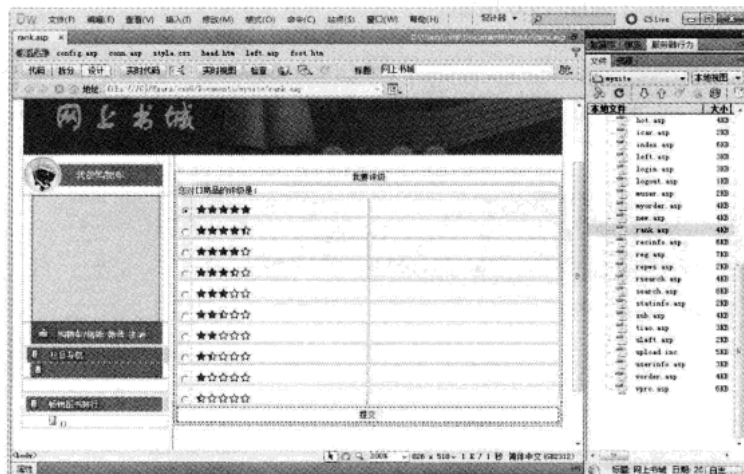


图 19.7 评级表单设计效果

然后在表单中插入一个隐藏域，属性设置如图 19.8 所示，用于传递图书的编号。

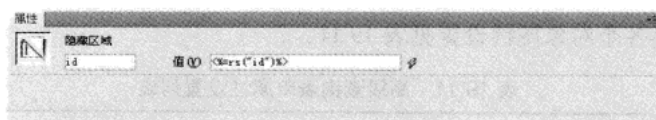


图 19.8 设置隐藏域属性

comment.asp 文件是一个用户评论页面，在该页面中提供了一个简单的表单，设计效果如图 19.9 所示。该表单域由两个文本域和两个按钮组成，具体属性设置可以参考光盘中的文件，最后在表单中插入一个隐藏域，用于传递评论图书的编号。

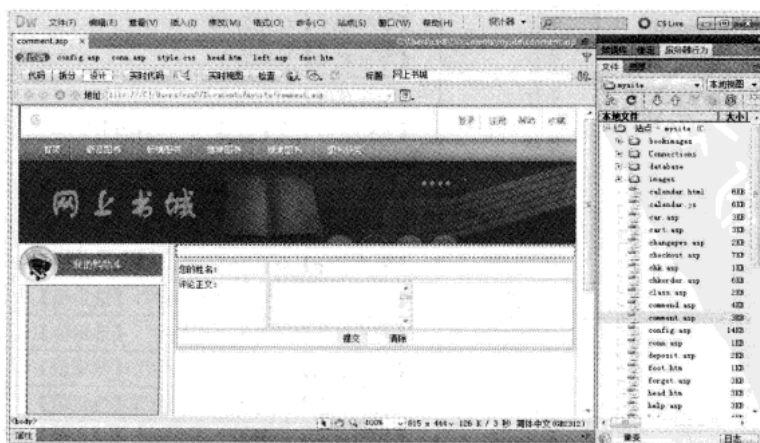


图 19.9 设置用户评论页面

19.4 商品展示

商品展示主要包括最新图书、推荐图书、畅销图书、图书分类、图书分级展示、商品导航等功能模块,下面重点介绍其中的展示和导航两个功能模块。

19.4.1 栏目导航

本系统的每页左侧都附带一个栏目导航栏,用于分类导航、显示我的购物车信息和畅销图书排行,如图 19.10 所示。




图 19.10 图书栏目导航区域

实现栏目导航功能需要解决栏目的折叠与记录集嵌套。在前面章节中曾详细介绍过这两项技术的实现方法,不过那种菜单折叠是静态的,所要折叠的单元格属性是静止的,不会随时发生变化,记录集嵌套是以纯代码的形式实现的。本节将介绍动态单元格折叠,即单元格的ID属性是在不断变化,记录集嵌套以 Dreamweaver CS5 服务器行为操作为主,适当增加一点代码,在操作上虽然简单点,但难度较大。具体操作步骤如下。

01 启动 Dreamweaver CS5,在【文件】面板中打开 left.asp 文件,在【数据库】面板中定义一个数据库连接,如图 19.11 所示。



图 19.11 【数据库】面板

02 定义记录集查询要在栏目导航中显示的图书大类。在【绑定】面板中单击面板左上角的  按钮,在弹出的下拉菜单中选择【记录集(查询)】菜单命令,打开【记录集】对话框。

【记录集】对话框具体设置如图 19.12 所示,筛选出要指定显示的大类图书。

03 以同样的方式,定义一个记录集查询所有指定的图书小类,设置如图 19.13 所示。



图 19.12 查询指定图书大类

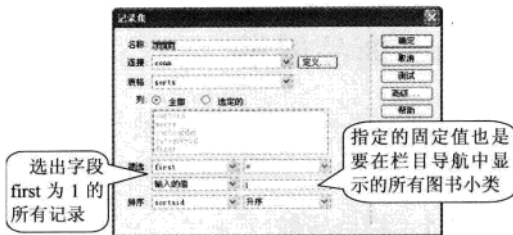



图 19.13 查询指定图书小类

04 把 `category` 记录集的 `category` 字段插入到栏目栏中，把 `sorts` 记录集的 `sorts` 字段插入栏目栏下边的行中，如图 19.14 所示。

05 定义大类重复显示。选中图书大类行和图书小类行，然后在打开【服务器行为】面板中单击【服务器行为】面板左上角的  按钮，在弹出的下拉菜单中选择【重复区域】命令，在【记录集】下拉列表中选择 `category` 记录集；在【显示】区域选择【所有记录】单选按钮，如图 19.15 所示。

06 单击【确定】按钮，即可在页面中插入重复区域服务器行为，如图 19.16 所示。

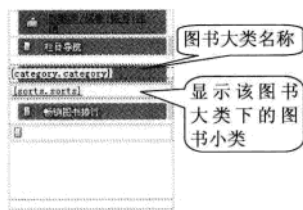


图 19.14 插入记录集字段

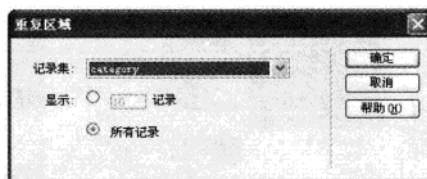


图 19.15 【重复区域】对话框



图 19.16 显示重复区域

07 切换到【代码】视图，找到 `sorts` 记录集代码段（一般定义的记录集代码段都自动放在页面代码的 `<html>` 标签的前面），将其复制到图书大类栏目内，把该记录集的开始代码（一般都自动放在页代码尾部）也复制过来。然后增加 `sorts` 记录集的筛选条件，即根据图书大类编号筛选符合该大类的图书小类。具体代码如下：

```
<% 'Dreamweaver CS5自动定义的重复区域服务器行为
While ((Repeat1_numRows < 0) AND (NOT category.EOF))
%>
<tr>
  <td height="23" background="images/lmenu 2.gif" class="dao" onClick=Menu
    (<%=m"&category("categoryid")%>)><%=category.Fields.Item("category").Value%></td>
</tr>
<tr>
  <td height="23" id=<%=m"&category("categoryid")%> style="display:none"><table
    width="100%" border="0" cellspacing="0" cellpadding="0">
    <% '从页代码顶部复制Dreamweaver CS5自动定义的sorts记录集，请注意下面SQL查询语句中又增
    加了一个筛选条件
    Dim sorts
    Dim sorts_numRows

    Set sorts = Server.CreateObject("ADODB.Recordset")
    sorts.ActiveConnection = MM_conn_STRING
    sorts.Source = "SELECT * FROM sorts WHERE categoryid=" & (category.Fields.Item("categoryid").
    Value) & " and first = " + Replace(sorts__MMColParam, "'", "'') + " ORDER BY sortsid ASC"
    sorts.CursorType = 0
    sorts.CursorLocation = 2
```

```

        sorts.LockType = 1
        sorts.Open()

        sorts_numRows = 0
    %>
        <%
        do while not sorts.eof
            %>
            <tr>
                <td class="daosub" ><a href="sub.asp?aid=<%= (category.Fields.Item
("categoryid").Value) %>&nid=<%= (sorts.Fields.Item("sortsid").Value) %>"><%= (sorts.Fields
.Item("sorts").Value) %></a></td>
            </tr>
            <%
            sorts.movenext
        loop
        %>
        <% '从页代码尾部复制sorts记录集结束代码
        sorts.Close()
        Set sorts = Nothing
        %>
        </table></td>
        </tr>
        <% '重复区域服务器行为循环语句
        Repeat1__index=Repeat1__index+1
        Repeat1__numRows=Repeat1__numRows-1
        category.MoveNext()
    Wend
    %>

```

08 在插入的图书小类名称行前后插入自定义循环语句, 以实现该图书大类下的所有图书小类重复显示。代码如下:

```

<% ' sorts记录集
Dim sorts
Dim sorts_numRows

Set sorts = Server.CreateObject("ADODB.Recordset")
sorts.ActiveConnection = MM_conn_STRING
sorts.Source = "SELECT * FROM sorts WHERE categoryid=" & (category.Fields.Item("categoryid").
Value) & " and first = " + Replace(sorts__MMColParam, "'", "'") + " ORDER BY sortsid ASC"
sorts.CursorType = 0
sorts.CursorLocation = 2
sorts.LockType = 1
sorts.Open()

sorts_numRows = 0
%>

<% '增加代码, 实现图书小类重复显示, 也可以选中该行利用重复区域服务器行为实现图书小类重复显示
do while not sorts.eof
    %>
        <tr>
            <td class="daosub"><a href="sub.asp?aid=<%= (category.Fields.Item
("categoryid").Value) %>&nid=<%= (sorts.Fields.Item("sortsid").Value) %>"><%= (sorts.Fields
.Item("sorts").Value) %></a></td>
        </tr>
        <%
        sorts.movenext
    loop
    %>
    <%
    sorts.Close()
    Set sorts = Nothing
    %>

```

- 09** 实现栏目的折叠功能。由于本栏目导航是动态栏目，实现起来比静态的折叠难度大。首先，应在头部定义一个函数，实现样式表 display 的属性自动开关切换，代码如下：

```
<script language="javascript">
function Menu(term){
    if(term.style.display=="none"){
        term.style.display="";
    }else{
        term.style.display="none";
    }
}
</script>
```

- 10** 在栏目标题栏（图书大类标题栏）插入鼠标单击动作，具体代码如下：

```
<tr>
    <td height="23" background="images/lmenu_2.gif" class="dao" onClick =Menu
    (<%= "m"&category("categoryid")%>)><div align="left"><%= (category.Fields.Item("category").
    Value)%></div></td>
</tr>
```

- 11** 在图书小类显示区域所在的单元格属性中定义动态 ID 属性，以实现对该单元格显示的动态控制。

```
<tr>
    <td height="23" id=<%= "m"&category("categoryid")%> style="display:none">
    <table width="100%" border="0" cellspacing="2" cellpadding="2">
    <% ' sorts记录集
    Dim sorts
    Dim sorts_numRows
    Set sorts = Server.CreateObject("ADODB.Recordset")
    sorts.ActiveConnection = MM_conn_STRING
    sorts.Source = "SELECT * FROM sorts WHERE categoryid=" & (category.Fields.Item("categoryid").
    Value) & " and first = " + Replace(sorts__MMColParam, "'", "'') + " ORDER BY sortsid ASC"
    sorts.CursorType = 0
    sorts.CursorLocation = 2
    sorts.LockType = 1
    sorts.Open()
    sorts_numRows = 0
    %>
```

- 12** 选中图书小类标题，如图 19.17 所示。然后在【属性】面板中的【链接】文本框中输入下面链接地址与参数。

```
sub.asp?aid=<%= (category.Fields.Item("categoryid").Value)%>&nid=<%= (sorts.Fields.Item("sortsid").Value)%>
```

- 13** 定义一个记录集 solded，用来筛选图书产品表中卖得最好的前 10 本图书，定义的记录集设置如图 19.18 所示。



图 19.17 选中图书小类标题

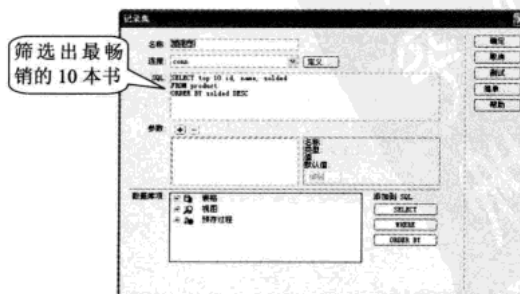


图 19.18 查询最畅销的图书

- 14** 把 `solded` 记录集中的 `name` 字段插入页面中。然后选中该字段,在【属性】面板的【链接】文本框中输入下面链接代码:

```
vpro.asp?id=<%= (solded.Fields.Item("id").Value) %>
```

- 15** 打开【服务器行为】面板,单击【服务器行为】面板左上角的 \oplus 按钮,在弹出的下拉菜单中选择【重复区域】命令,设置显示 10 条记录。单击【确定】按钮,即可在页面中插入重复区域服务器行为。

- 16** 有的图书标题可能很长,为了避免长标题对整个页面的影响,可以定义一个函数省略超宽的标题部分字符。打开 `config.asp` 配置文件,输入下面代码,定义一个自定义函数。

```
function strvalue(str,lennum)
    dim p_num,x,i
    if strlen(str)<=lennum then
        strvalue=str
    else
        p_num=0
        x=0
        do while not p_num > lennum-2
            x=x+1
            if asc(mid(str,x,1))<0 then
                p_num=int(p_num) + 2
            else
                p_num=int(p_num) + 1
            end if
            strvalue=left(trim(str),x)&"..."
        loop
    end if
end function
```

- 17** 返回 `left.asp` 文件,把 `strvalue` 函数插入代码中。

```
<td align="left"><a href="vpro.asp?id=<%= (solded.Fields.Item("id").Value) %>" target=
_blank"><%=strvalue(trim
((solded.Fields.Item("name").Value)),20)%><%= (solded.Fields.Item("solded").Value) %></
a></td>
```

- 18** 由于 `left.asp` 文件是一个被导入文件,即要被导入到其他文件中,作为页面的一部分,为了避免发生冲突和错误,最后要删除该页面顶部的 `<%@"%>` 标记代码行。

```
<%@LANGUAGE="VBSCRIPT" %>
```

这样就可以实现栏目导航功能。

19.4.2 购物首页

网上购物首页主要包括头部标题 (`head.htm`)、左侧栏目导航 (`left.asp`)、底部的版权信息 (`foot.htm`),以及首页中重点显示的“新书上架”、“推荐图书”和“快速搜索”这 3 个栏目,如图 19.19 所示。其中快速搜索功能将在后面章节中介绍,本节将重点介绍“新书上架”和“推荐图书”这两个栏目功能实现。

实现首页功能的具体步骤如下。

- 01** 在 Dreamweaver CS5 中打开 `index.asp` 文件,先定义两个记录集,分别用来查询最新上市图书和推荐图书,如图 19.20 和图 19.21 所示。

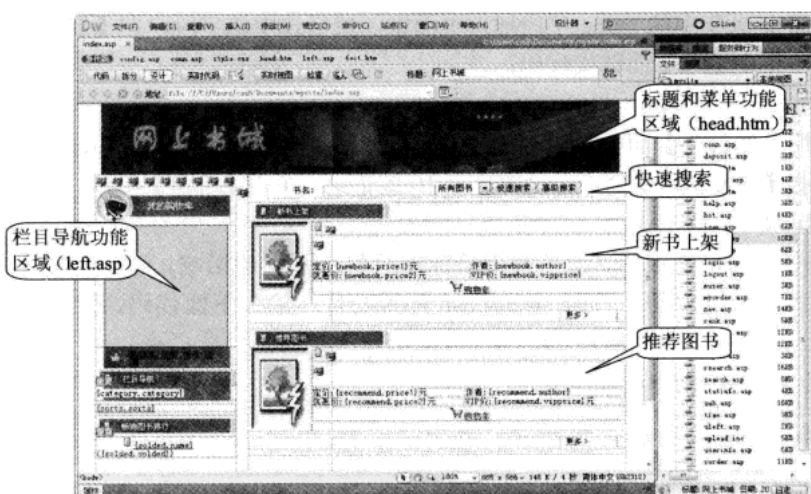


图 19.19 首页设计效果

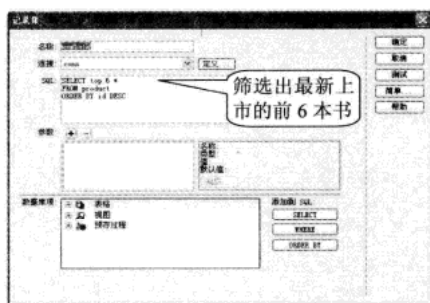


图 19.20 查询最新图书

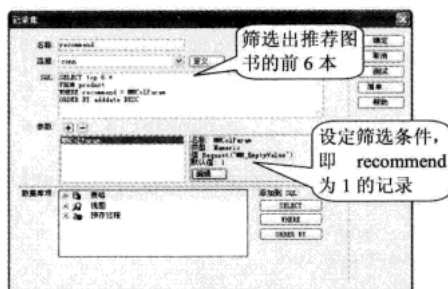


图 19.21 查询推荐图书

02 分别把 newbook 记录集和 recommend 记录集相应字段插入页面, 如图 19.22 所示。

03 在插入记录集时, 图书标题、内容介绍和封面图片还需要加入处理函数, 这里没有直接插入, 而是在【代码】视图下输入。代码如下:

```
<td width="17%" height="130" align="center"
valign="middle" class="shadow"><a href="vpro.asp?
id=<%=trim(newbook("id"))%>" target="_blank">
<imgsrc="<%=newbook.Fields.Item("pic").Value%>
" width="85" height="125" border="0"></a></td>
<td width="33%" align="left" valign="top">
<table width="100%" border="0" cellspacing="1" cellpadding="0">
<tr>
<td colspan="2"><span class=
"booktitle"><%=strvalue(trim(newbook("name")),24)%></span></td>
</tr>
<tr>
<td height="40" colspan="2" class="bookinfo"><%=strvalue(trim(newbook("desc")),
100)%><br>
<br></td>
</tr>
```

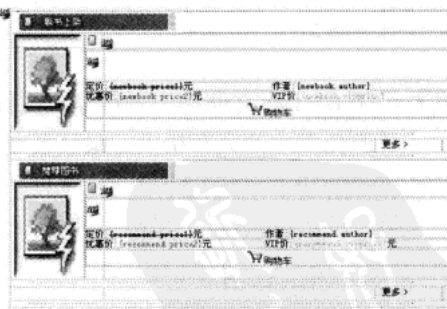


图 19.22 插入记录集字段


- 04** 上面代码中设置图片的超链接为 `vpro.asp?id=<%=trim(newbook("id"))%>`, 并传递该最新图书的编号。为了使图片看起来美观, 用 CSS 样式给图片增加阴影特效。

```
.shadow {
    filter: DropShadow(Color=#B4B4B4, OffX=3, OffY=5, Positive=1);
}
```

- 05** 插入记录集字段时, Dreamweaver CS5 自动插入如下代码。

```
<%= (recommend.Fields.Item("author").Value) %>
```

实际上上面代码可以简化为 `<%= (recommend ("author")) %>` 形式, 这样看起来更容易明白。

- 06** 选中“新书上架”栏目中的插入图书信息所在行和下边的虚线行, 如图 19.23 所示。然后打开【服务器行为】面板, 单击【服务器行为】面板左上角的  按钮, 在弹出的下拉菜单中选择【重复区域】命令, 在【记录集】下拉列表中选择 newbook 记录集, 在【显示】区域设置每页只显示 6 条记录, 如图 19.24 所示。

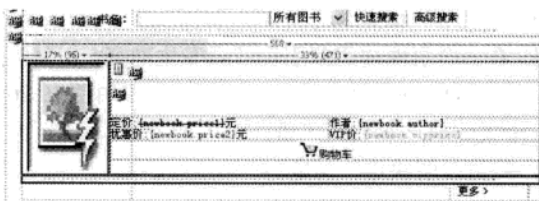


图 19.23 选中要定义重复区域的行

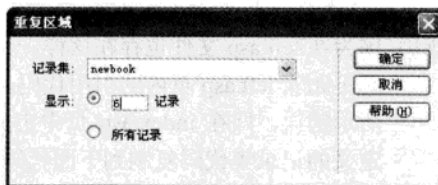


图 19.24 定义重复区域

- 07** 切换到【代码】视图, 在显示图书信息单元格上下加入循环语句, 实现在横向重复显示信息。代码如下:

```
<% 'Dreamweaver CS5自动插入的重复区域代码
While ((Repeat1__numRows1 <> 0) AND (NOT newbook.EOF))
%>
<tr>
<% '定义一行重复显示2个记录
for num=1 To 2
if newbook.eof then exit for '判断如果没有记录则退出循环
%>
    <td width="17%" height="130" align="center" valign="middle" class="shadow"><a href
="vpro.asp?id=<%=trim(newbook("id"))%>" target="_blank">" width="85" height="125" border="0"></a></td>
    <td width="33%" align="left" valign="top">
        ..... '图书详细信息
        </td>
        <% '下移一个记录
        newbook.MoveNext()
        next
        newbook.MovePrevious() '当跳出循环时, 要把记录指针返回
        %>
    </td>
</tr>
<tr><td colspan="4" align="center"></td></tr>
<% 'Dreamweaver CS5自动插入的重复区域循环语句
Repeat1__index1=Repeat1__index1+1
Repeat1__numRows1=Repeat1__numRows1-1
newbook.MoveNext()
Wend
%>
```

08 以同样的方式实现“推荐图书”栏目内的图书记录在行内 2 次重复显示。

09 在“最新图书”栏目中选中“购物车”，在属性面板中的【链接】文本框中输入下面代码：

```
<%= (newbook.Fields.Item("id").Value) %>
```

10 选中“更多”文本，在属性面板中的【链接】文本框中输入“new.asp”。

11 同样在“推荐图书”栏目中选中“购物车”，在属性面板中的【链接】文本框中输入下面代码。

```
<%= (recommend.Fields.Item("id").Value) %>
```

12 选中“更多”，在属性面板中的【链接】文本框中输入“commend.asp”。

13 按 F12 在浏览器中运行该页面，结果出现下面错误，如图 19.25 所示。名称重定义，也就是重复定义下面代码：

```
Dim MM_conn_STRING
```

14 在 left.asp 页面中切换到【代码】视图，删除顶部的导入文件，代码如下：

```
<!--#include file="Connections/conn.asp" -->
```

15 因为在 left.asp 文件也存在这样一行代码。在一个页面内重复引用就会发生冲突。如果此时不删除 left.asp 页内的引用代码，而是删除 index.asp 页内的代码，则会出现如图 19.26 所示错误，即在 index.asp 页中定义的记录集因缺乏参数而出错。可以把 index.asp 页内定义的记录集代码复制到下面代码后面，就可以解决这个问题。

```
<!--#include file="left.asp"-->
```



图 19.25 错误信息（一）



图 19.26 错误信息（二）



提示

在 Dreamweaver CS5 中定义记录集都是放在页代码顶部，按先后顺序往上堆放，最后定义的记录集位于最上面。

同时，浏览器在执行页代码时，按照从上到下的顺序解释程序代码，而且 Dreamweaver CS5 在插入记录集等服务器行为时，都是按照一个模式来定义。因此，当在一个页面内定义多个记录集，而且又相互引用字段变量，或者在页面内插入其他页面文件，就会出现很多错误，最主要的就是重复定义问题，这也是 Dreamweaver CS5 目前不能克服的一个弱点，同时服务器行为代码臃肿，不易查看，也导致很多用户甘愿自己动手手写代码，手写代码简洁灵活而又健壮。

16 再次在浏览器中运行，结果还是存在错误，错误页面如图 19.27 所示。解决这种问题有

如下三种方法。

(1) 在 index.asp 文件中用替换的方法把所有 Repeat1_numRows 变量替换为其他名字, 避免和 left.asp 文件内的 Repeat1_numRows 变量重复。

(2) 在 index.asp 文件中找到定义 Repeat1_numRows 变量的代码, 删除该定义变量代码。

(3) 在 index.asp 文件中把所有定义的记录集、重复区域服务器行为所产生的代码复制到 <!--#include file="left.asp"-->行后面, 最好放在运用该记录集单元格或行上面。

在上面代码中, 最 方法是第 3 种, 使用前两种方法运行页面没有问题, 但是单击页面内的链接就会出现如图 19.28 所示的错误。



图 19.27 错误信息(三)



图 19.28 错误信息(四)

总之, 在使用 Dreamweaver CS5 定义服务器行为制作复杂查询或操作的页面时, 会出现很多意想不到的错误, 读者有时会感觉防不 防。解决的最 方法当然是自己手写代码, 另一个办法就是在页面内移动服务器行为定义的代码固定位置, 一般的原则就是在页面哪个区域(表格或单元格)使用服务器行为, 就把代码复制到其周围, 避免和页面其他区域的服务器行为代码 杂在一起。例如, 在本例 index.asp 文件中, 就把对应的代码都复制到相对应的单元格周围。在复制时用户必须清楚各个代码段的用途。整个页面调整代码如下:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="manage/inc/config.asp"-->
<!--#include file="inc/conn.asp"-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>网上书城</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<!--#include file="head.htm"-->
<table width="780" border="0" align="center" cellpadding="0" cellspacing="0">
  <!--#include file="left.asp"-->
  ..... '搜索区域
  <% '把顶部的定义的新book记录集复制到这
  Dim newbook
  Dim newbook_numRows

  Set newbook = Server.CreateObject("ADODB.Recordset")
  newbook.ActiveConnection = MM_conn_STRING
```

```

newbook.Source = "SELECT top 6 * FROM product ORDER BY id DESC"
newbook.CursorType = 2
newbook.CursorLocation = 2
newbook.LockType = 1
newbook.Open()

newbook_numRows = 0
%>
<% '把定义的重复区域复制到这
Dim Repeat1_numRows1
Dim Repeat1_index1

Repeat1_numRows1 = -1
Repeat1_index1 = 0
newbook_numRows = newbook_numRows + Repeat1_numRows1
%>
.....新书上架表格区域
<% '把newbook记录集结束语句复制到这
newbook.Close()
Set newbook = Nothing
%>
.....
<% '把顶部的recommend记录集复制到这
Dim recommend
Dim recommend_numRows

Set recommend = Server.CreateObject("ADODB.Recordset")
recommend.ActiveConnection = MM_conn_STRING
recommend.Source = "SELECT top 6 * FROM product WHERE recommend = " + Replace(recommend
MMColParam, "'", "'') + " ORDER BY adddate DESC"
recommend.CursorType = 2
recommend.CursorLocation = 2
recommend.LockType = 1
recommend.Open()

recommend_numRows = 0
%>
<% '把定义的重复区域复制到这
Dim Repeat2_numRows1
Dim Repeat2_index1

Repeat2_numRows1 = 6
Repeat2_index1 = 0
recommend_numRows = recommend_numRows + Repeat2_numRows1
%>
.....推荐图书表格区域
<%
Repeat2_index1=Repeat2_index1+1
Repeat2_numRows1=Repeat2_numRows1-1
recommend.MoveNext()
Wend
%>
<% '把recommend记录集结束语句复制到这
recommend.Close()
Set recommend = Nothing
%>
.....
<!--#include file="foot.htm"-->
</body>
</html>

```

19.4.3 商品分类

商品分类页面是网上购物不可缺少的一个功能页面，用户可以在商品分类页面中按类分

级列表快速找到自己需要的商品,如图 19.29 所示。在首页中曾设计了一个栏目导航区也具有分类查询功能,但由于首页版面珍贵,仅提供了主要类别查询。



图 19.29 图书分类页面效果

本系统分类查询页主要包括两个级别,即包括图书大类和图书小类。下面详细介绍商品分类查询页的实现步骤。

- 01** 在 Dreamweaver CS5 中打开 class.asp 文件,先定义 2 个记录集,用来查询数据库中图书大类和图书小类记录。

其中查询图书大类的【记录集】对话框设置如图 19.30 所示。在查询小类图书时,需要参照大类图书编号筛选该大类下的所有小类图书,因此在设置【记录集】时,先设置筛选条件为 sorts 表中的 categoryid 字段为一个常值,然后在【代码】视图中进行修改。具体设置如图 19.31 所示。

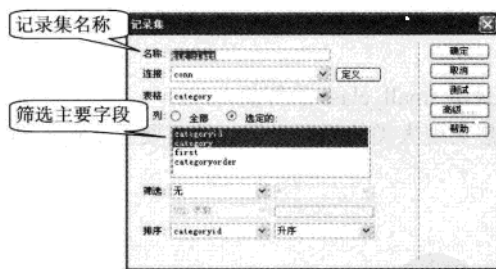


图 19.30 图书大类查询设置



图 19.31 图书小类查询设置

- 02** 把 big_class 记录集中的 category 字段插入表格的第 1 行单元格,把 small_class 记录集中的 sorts 字段插入到表格的第 2 行,如图 19.32 所示。

- 03** 选中插入的 sorts 字段,然后在属性面板的【链接】文本框中输入链接地址和传递参数。具体代码如下:

```
sub.asp?aid=<%= (big_class.Fields.Item("categoryid").Value) %>&nid=<%= (small_class.Fields.Item("sortsid").Value) %>
```

- 04** 定义图书小类重复显示。选中插入的 sorts 字段,注意不要选择整个单元格或行,打开

【服务器行为】面板，单击【服务器行为】面板左上角的 \oplus 按钮，在弹出的下拉菜单中选择【重复区域】命令，打开【重复区域】对话框，设置如图 19.33 所示。



图 19.32 插入记录集字段

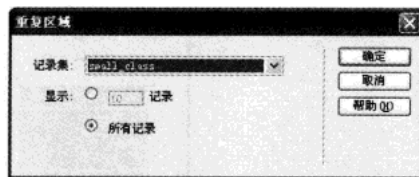


图 19.33 设置【重复区域】对话框

- 05** 切换到【代码】视图，找到 small_class 记录集代码段，重新设置 small_class 记录集的筛选条件。代码如下：

```
<% 定义的 small_class 记录集代码段
Dim small_class
Dim small_class_numRows

Set small_class = Server.CreateObject("ADODB.Recordset")
small_class.ActiveConnection = MM_conn_STRING
small_class.Source = "SELECT sortsid, sorts FROM sorts WHERE categoryid = "&(big_class.
Fields.Item("categoryid").Value)&" ORDER BY sortorder ASC"
small_class.CursorType = 0
small_class.CursorLocation = 2
small_class.LockType = 1
small_class.Open()
small_class_numRows = 0
%>
```

- 06** 把 small_class 记录集代码段复制到应用该记录集 small_class 的行前。

- 07** 由于【重复区域】服务器行为不能嵌套，要想实现大类的循环显示，就需要手工输入循环语句。代码如下：

```
<% 定义大类循环语句
do while not big_class.eof
%>
    <tr><td>
        <table border="0" cellspacing="0" cellpadding="2">
            <tr><td></td>
                <td><span class="title"><%= (big_class.Fields.Item("category").Value) %>
</span></td>
            </tr>
        </table></td>
    </tr>
    <tr>
        <td style="padding-left: 30px;">
<% 定义并移动过来的 small_class 记录集
Dim small_class
Dim small_class_numRows
Set small_class = Server.CreateObject("ADODB.Recordset")
```

```

small_class.ActiveConnection = MM_conn_STRING
small_class.Source = "SELECT sortsid, sorts FROM sorts WHERE categoryid = "&(big_class.
Fields.Item("categoryid").Value)&" ORDER BY sortorder ASC"
small_class.CursorType = 0
small_class.CursorLocation = 2
small_class.LockType = 1
small_class.Open()
small_class_numRows = 0
%>
<% ' 从代码页顶部移过来的重复区域定义变量
Dim Repeat3__numRows
Dim Repeat3__index
Repeat3__numRows = -1
Repeat3__index = 0
small_class_numRows = small_class_numRows + Repeat3__numRows
%>
<% ' 定义的重复区域代码
While ((Repeat3__numRows <> 0) AND (NOT small_class.EOF))
%>
    <a href=sub.asp?aid=<%= (big_class.Fields.Item("categoryid").Value) %>&nid=
<%= (small_class.Fields.Item("sortsid").Value) %>><%=trim((small_class.Fields.Item("sorts
").Value)) %></a>
    <%
        Repeat3__index=Repeat3__index+1
        Repeat3__numRows=Repeat3__numRows-1
        small_class.MoveNext()
    Wend
%>
<% ' 从代码页底部移过来的small_class记录集结束语句
small_class.Close()
Set small_class = Nothing
%>
</td></tr>
<% ' 结束图书大类的循环
big_class.movenext
loop
%>

```

19.4.4 二级商品显示页面

二级商品显示页面是针对首页和商品分类页来说的。当用户在商品分类页内选择某个商品小类, 或者在首页栏目导航内选择某个小类, 系统会显示该小类下所有商品, 如图 19.34 所示。如果在系统菜单栏内选择某个分类菜单, 也可以显示该类下所有商品, 如图 19.35 所示。



图 19.34 普通类图书二级页面



图 19.35 特定类图书二级页面

实现商品二级页面功能的操作步骤如下。

- 01** 打开 sub.asp 页面。根据传递参数 aid 和 nid 的值定义 3 个记录集，用于查询所属商品大类的名称和编号，指定商品小类的名称和编号，以及指定商品小类的所有商品记录。定义的记录集如图 19.36 所示。

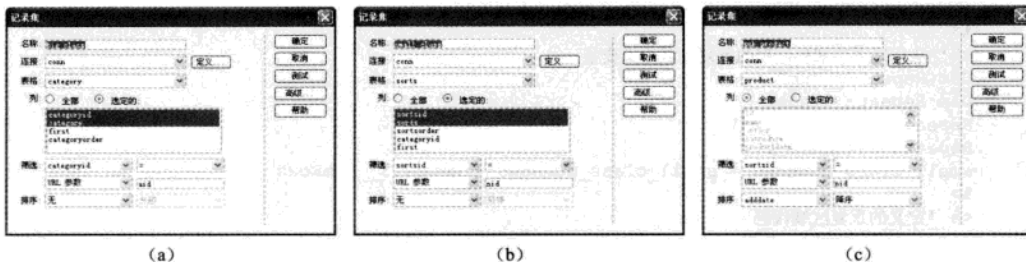


图 19.36 定义记录集

(a) big_class 记录集; (b) small_class 记录集; (c) sub_product 记录集

- 02** 分别把记录集中的字段插入到页面中，具体插入位置如图 19.37 所示。选中其中的动态图像，在属性面板的【链接】文本框中设置其链接文本如下：

```
vpro.asp?id=<%=trim(sub_product("id"))%>
```

选择“购物车”图像，在属性面板的【链接】文本框中设置其链接文本如下：

```
icar.asp?id=<%= (sub_product.Fields.Item("id").Value) %> &action=add
```



图 19.37 插入记录集字段

- 03** 选中插入商品字段所在行以及下面虚线行，如图 19.38 所示。

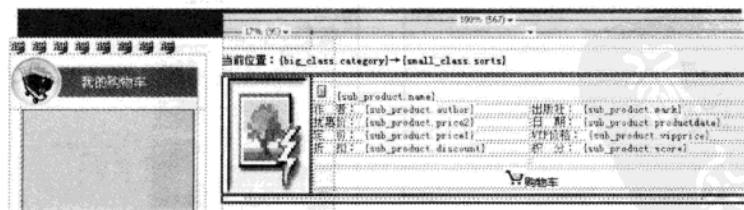


图 19.38 选中要重复显示的行

- 04** 选择【窗口】|【服务器行为】命令，打开【服务器行为】面板。单击【服务器行为】面板左上角的 按钮，在弹出的下拉菜单中选择【重复区域】命令。在打开的【重复区域】对话框中设置重复显示功能，如图 19.39 所示。

- 05 把光标置于重复区域下面一行, 选择【插入记录】|【数据对象】|【记录集分页】|【记录集导航条】命令, 打开【记录集导航条】对话框设置分页导航, 如图 19.40 所示。

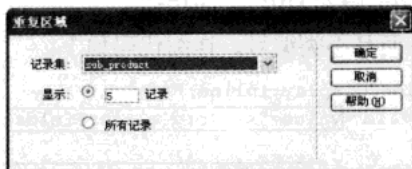


图 19.39 设置【重复区域】对话框

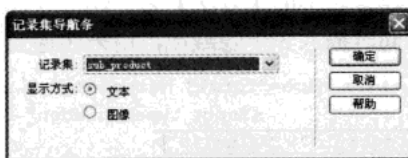


图 19.40 设置【记录集导航条】对话框

- 06 切换到【代码】视图, 找到插入的记录集导航条文本按钮代码段, 删除嵌入在 ASP 代码两边的表格标签, 实现导航文本排列紧。并在代码段后面输入下面代码, 实现获取分页信息, 并显示该信息。

```
<tr align="center"><td colspan="2"><br>
    <% If MM_offset <> 0 Then %>
        <a href="<%=MM_moveFirst%>">第一页</a>
    <% End If ' end MM_offset <> 0 %>
    <% If MM_offset <> 0 Then %>
        <a href="<%=MM_movePrev%>">前一页</a>
    <% End If ' end MM_offset <> 0 %>
    <% If Not MM_atTotal Then %>
        <a href="<%=MM_moveNext%>">下一页</a>
    <% End If ' end Not MM_atTotal %>
    <% If Not MM_atTotal Then %>
        <a href="<%=MM_moveLast%>">最后一页</a>
    <% End If ' end Not MM_atTotal %>
```

<'''定义pagenum变量获取记录集页数, 定义pagenow变量获取记录集当前页, 其中num变量是在上面代码中定义并存储用户设置的每页显示记录数, sub_product_first变量是系统在插入记录集导航条时自动生成变量, 用来存储每页的第1条记录数

```
dim pagenum
dim pagenow
pagenum = int(sub_product.recordcount/num)
if pagenum < sub_product.recordcount/num then pagenum = pagenum + 1
pagenow =int(sub_product_first/num) + 1
%>
```

```
        页次: <%=pagenow%>/<%=pagenum%>页, 共有<b>&nbsp;&nbsp;&nbsp;<%=sub_product.
recordcount%>&nbsp;&nbsp;&nbsp;</b>种商品, <b>&nbsp;&nbsp;&nbsp;<%=num%>&nbsp;&nbsp;&nbsp;</b>种商品/页
</td></tr>
```



提示

当用 recordcount 系统变量来获取记录集中记录数时, 要注意在【服务器行为】面板中选中 sub_product 记录集, 在【面板】中不要设置【光标类型】为“仅向前”选项, 否则变量无效, 如图 19.41 所示。该方法可以针对 Dreamweaver 8 定义的记录集, 使用 Dreamweaver CS5 版本定义记录集时, 系统改用 Command 对象来定义记录集, 而不是使用 Recordset 对象。由于特殊的需要, 本例部分记录集代码使用了 Recordset 对象来定义的记录集。

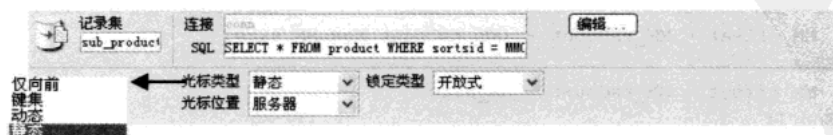


图 19.41 设置记录集光标类型

- 07** 通过输入页数进行导航。在上面代码段后面插入一个文本框对象，为了实现文本框与其他文本信息同行显示，设置文本框 CSS 样式为 `style="display:inline;"`。同时在属性面板中设置表单的【方法】为“GET”，即使表单域内的值附在 URL 后面传递。
- 08** 同时插入 3 个隐藏域，用来传递当前商品大类和商品小类编号，详细代码如下：

```
<form name="form1" method="get" action="" style="display:inline;">
    <input type="hidden" id="aid" name="aid" value="<%= (big class.Fields.Item
("categoryid").Value) %>">
    <input type="hidden" id="nid" name="nid" value="<%= (small class.Fields.
Item("sortsid").Value) %>">
    <input type="hidden" id="action" name="action" value="go">
    转到: <input type='text' name='offset' size=2 maxlength=10 class=smallInput
value="<%=pagenow %>">
    &nbsp;&nbsp;&nbsp;<input type='submit' value='GO' name='cndok'></form>
</form> <br><br></td>
```

其中文本框的名称必须命名为 `offset`，该名称是系统插入记录集导航条默认的传递记录参数。

- 09** 为了实现文本输入页数和记录集导航条导航同步，还需要增加代码，使用户输入的页数和系统默认的 `offset` 变量值等价。在【代码】视图中找到记录集导航条定义的下面代码段。

由于记录集导航条代码很多，初学者可能会很迷，不过查找灰色注释行“*** Move To Record: handle 'index' or 'offset' parameter”能快速找到目标段。

```
<%
' *** Move To Record: handle 'index' or 'offset' parameter

if (Not MM_paramIsDefined And MM_rsCount <> 0) then

    ' use index parameter if defined, otherwise use offset parameter
    MM_param = Request.QueryString("index")
    If (MM_param = "") Then
        MM_param = Request.QueryString("offset")
    End If
    '''插入下面一段代码，判断offset传递变量的值，是系统传递的值，还是用户在文本框中输入值，如果是用户
    输入的值，还需要转化为系统offset指定的等价值，这样就可以实现文本框和文本导航按钮同步
    if MM_param mod num <> 0 and MM_param>=0 then
        MM_param = (MM_param-1) * num
    else
        MM_param = MM_param
    end if
    If (MM_param <> "") Then
        MM_offset = Int(MM_param)
    End If

    ' if we have a record count, check if we are past the end of the recordset
    If (MM_rsCount <> -1) Then
        If (MM_offset >= MM_rsCount Or MM_offset = -1) Then ' past end or move last
            If ((MM_rsCount Mod MM_size) > 0) Then ' last page not a full repeat region
                MM_offset = MM_rsCount - (MM_rsCount Mod MM_size)
            Else
                MM_offset = MM_rsCount - MM_size
            End If
        End If
    End If
End If
```

```

' move the cursor to the selected record
MM_index = 0
While ((Not MM_rs.EOF) And (MM_index < MM_offset Or MM_offset = -1))
    MM_rs.MoveNext
    MM_index = MM_index + 1
Wend
If (MM_rs.EOF) Then
    MM_offset = MM_index ' set MM_offset to the last possible record
End If

End If
%>

```

- 10** 在【代码】视图中找到 sub_product 记录集代码段,在其后面输入下面代码,来检测用户选择的商品分类是否存在记录。

```

<%'定义的sub_product记录集代码段
Dim sub_product
Dim sub_product_numRows

Set sub_product = Server.CreateObject("ADODB.Recordset")
sub_product.ActiveConnection = MM_conn_STRING
sub_product.Source = "SELECT * FROM product WHERE sortsid = " + Replace(sub_product__
MMColParam, "'", "'") + " ORDER BY adddate DESC"
sub_product.CursorType = 3
sub_product.CursorLocation = 2
sub_product.LockType = 3
sub_product.Open()

sub_product_numRows = 0
%>
<%'检测用户选择的商品类别是否存在记录
if err.number<>0 then
    call MsgBox("数据库中无数据","Back","None")
    response.End
end if
if sub_product.eof And sub_product.bof then
    call MsgBox("对不起,目前没有更新商品!", "Back", "None")
    response.End
end if
%>

```

- 11** 把代码页中的下面行代码删除,避免和导入的 left.asp 文件内代码重定义。同时把前面操作中定义的服务器行为代码全部复制到插入 left.asp 文件单元格后面。

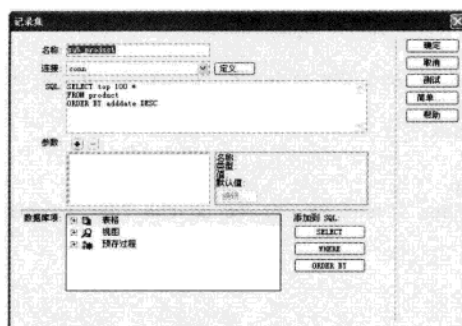
```
<!--#include file="Connections/conn.asp" -->
```

有关新书上架页面(new.asp)、畅销图书页面(hot.asp)和推荐图书页面(commend.asp)3个页面的功能实现方法与 sub.asp 页面的实现方法基本相同,如图 19.42 所示,具体操作可参照上面介绍步骤来实现。

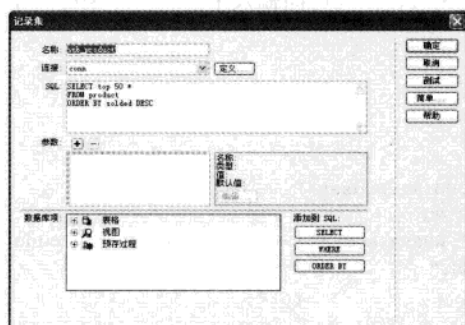


提示

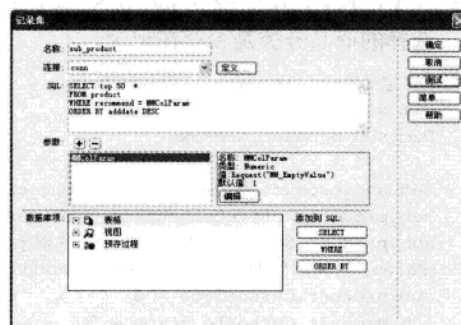
用户可以把 sub.asp 文件另存为 new.asp、hot.asp 和 commend.asp 文件,然后在【绑定】面板中删除其中的 big_class 和 small_class 记录集,在【代码】视图中删除有关应用 big_class 和 small_class 记录集的代码行,同时稍稍改动 sub_product 记录集查询条件就可以实现不同的页面显示功能,最后把页标题图像换一下图像即可。



(a)



(b)



(c)

图 19.42 设置记录集查询条件

(a) 新书查询设置; (b) 畅销图书查询设置; (c) 推荐图书查询设置

19.4.5 商品详细页面

商品详细页面主要是展示指定商品的详细信息,也可以包括一些互动功能,例如,对商品的评论、用户留言、使用反馈、给商品打分等功能。本系统的商品详细页面浏览效果如图 19.43 所示。



(a)



(b)

图 19.43 商品详细页面浏览效果

(a) 商品详细信息; (b) 商品互动功能

实现本页功能的具体操作步骤如下。

- 01** 打开 vpro.asp 文件, 首先定义一个记录集, 根据传递的商品 id 编号查询该商品的详细信息。设置的【记录集】对话框如图 19.44 所示。
- 02** 把 book 记录集中的所有字段插入到页面相应位置, 如图 19.45 所示。

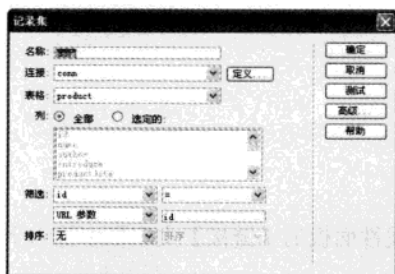


图 19.44 定义记录集

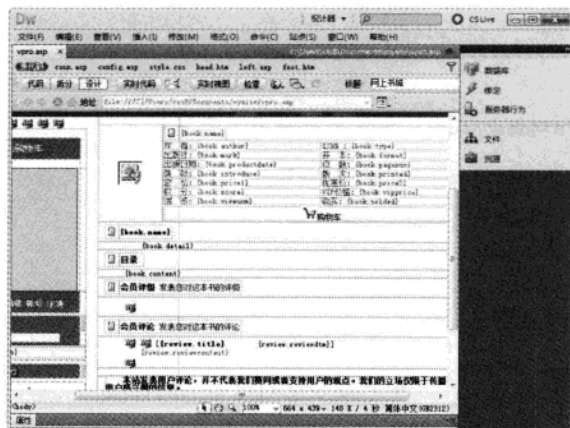


图 19.45 商品详细页面浏览效果

- 03** 打开【服务器行为】面板, 选中“记录集 (book)”, 然后在【属性】面板中设置【锁定类型】为“开放式”选项, 如图 19.46 所示。这样就可以实现通过该记录集对原数据库的更新与修改 (注意, 用 Dreamweaver CS5 定义记录集没有该功能, 需要读者手动输入记录集定义代码, 具体说明可以参阅实例效果源代码)。



图 19.46 设置记录集的锁定类型

- 04** 切换到【代码】视图, 把 book 记录集代码段移动到导入 left.asp 文件的单元格之后。同时, 在定义记录集代码段后面加入下面代码。

```
<% 实现对原数据库中viewnum字段的密切
book("viewnum")=book("viewnum")+1
book.update
%>
```

选中购物车, 在属性面板中设置【链接】文本框的值, 代码如下:

```
<a href="icart.asp?id=<%= (book.Fields.Item("id").
Value) %> &action=add" target="pcart" ><img src=
"images/car.gif" width="23", height="20" border="0">
购物车</a></td>
```

- 05** 再定义一个记录集, 用来查询用户给商品打的分, 具体设置如图 19.47 所示。
- 06** 在商品详细页面的底部输入下面代码, 实现互动功能。

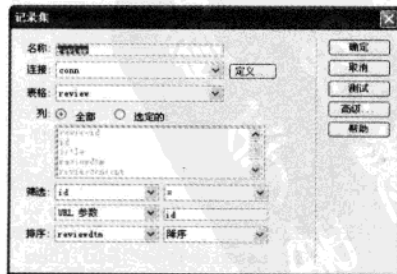


图 19.47 定义记录集


```

rs("rank")=rs("rank")+1      '增加评级人数
rs("ranknum")=rs("ranknum")+request("radiobutton")  '累积增加评级分数
rs.update
rs.close
Call MsgBox("您对本商品的评论星级已成功提交!", "Close", "None")
rs.close
set rs=nothing
session("id")=id             '把商品编号保存到session变量
session.timeout=1
response.End
end if
%>

```

- 02** 打开 comment.asp 文件, 在【代码】视图的顶部输入下面代码, 实现向数据库中写入用户对指定商品评价。

```

<%
dim id
id=request.QueryString("id")      '获取指定商品编号
if NOT isempty(request("CommentSubmit")) then  '判断用户是否向数据库中增加评论
    set rs=server.CreateObject("adodb.recordset") '定义记录集
    rs.open "select * from review",conn,1,3
    rs.addnew      '增加记录
    rs("id")=id
    rs("title")=HTMLEncode2(trim(request("title")))
    rs("reviewcontent")=HTMLEncode2(trim(request("reviewcontent")))
    rs("reviewdtm")=now()
    rs("audit")=0
    rs.update
    rs.close
    set rs=nothing
    call MsgBox("您的评论已成功提交, 经我们审核通过后方可发布!", "Close", "None")
    response.End
end if
%>

```

- 03** 其中的 HTMLEncode2 是一个自定义函数, 其功能就是检测用户是否输入标题和发表评论。代码如下:

```

<script language="javascript">
<!--
function checkspace(checkstr) {
    var str = '';
    for(i = 0; i < checkstr.length; i++) {
        str = str + ' ';
    }
    return (str == checkstr);
}
function check()
{
    if(checkspace(document.reviewform.title.value)) {
        document.reviewform.title.focus();
        alert("请填写您的姓名!");
        return false;
    }
    if(checkspace(document.reviewform.reviewcontent.value)) {
        document.reviewform.reviewcontent.focus();
        alert("请填写评论正文!");
        return false;
    }
}
//-->
</script>

```

19.5 实践总结

本章详细介绍了网上商城前台展示与查询系统的制作方法，包括一级分类、二级分类、推荐图书、新书图书、畅销图书、图书详细页面、查询和用户交互几个功能模块。商品展示和查询是网上购物的最重要一环，是用户选择购物的基础，因此在设计时如何实现商品浏览的方便快捷是衡量该系统功能的标准。在展示商品时应力求展示商品所有信息，方便用户查看，这是设计该系统应首先考虑的问题，否则展示的效果就会弱。本章难点是商品查询设计，这需要用户多实践，逐步掌握。



上一章介绍了网上购物系统的商品前台展示部分，本章将重点介绍网上购物系统中的购物车功能的实现。购物车是用户临时存放所购物品的地方，如同超市中的购物车。用户在购物车里可以随意对选择的商品进行添加、修改和取消操作；也可以选择一种商品付款后再返回继续购买，或者直至选购好所有的商品再去收银台。购物车是建立购物网站的重要一步，下面就来介绍购物车功能的实现方法。

技术要点

- 订单设计
- 跨表、分支和计算查询
- 多级购物管理
- 多记录集混合使用
- 复杂表单实现
- 用户注册、登录和密码服务

20.1 程序设计分析

购物系统是购物网站最主要的部分，通过不同的方式浏览、查询各种商品，将所要购买的商品放入购物车。在购物车中会列出所购商品的各项信息，例如，商品的编号、商品名称、商品单价、商品数量和价格总计等。在购物车中可以修改购买数量或取消商品的购买，甚至用户还可以通过“继续购物”超链接返回继续购物，或者通过“结账”按钮完成购物流程。

本章将介绍用户账户管理和购物流程功能实现，购物系统的一般流程如图 20.1 所示。

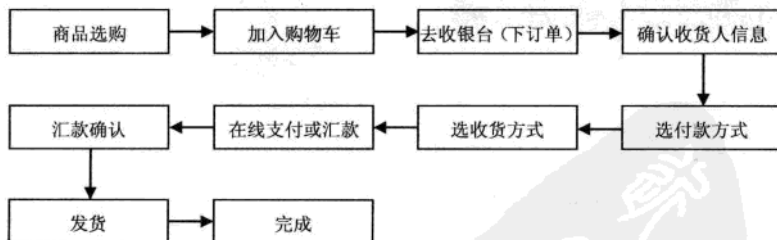
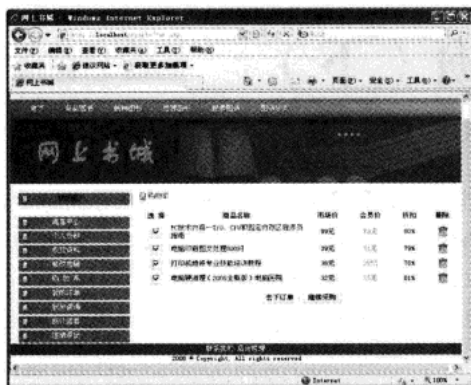


图 20.1 网上购物的一般流程

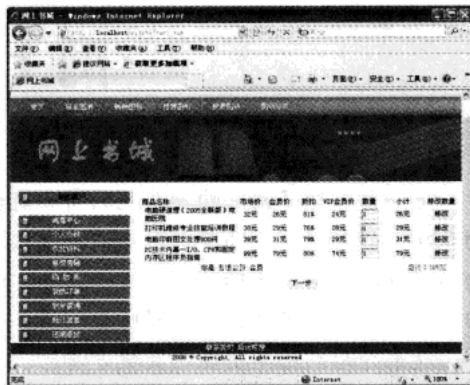
本章系统的浏览效果如图 20.2 所示，程序设计流程如图 20.3 所示。

实现用户账户和购物车功能，读者应该需要学习和解决下面技术难题。

- 用户注册、登录与账户管理。
- 购物车技术实现，要涉及表间查询和条件查询。
- 订单操作技术，包括订单生成、订单确定、订单列表和订单明细。



(a)



(b)



(c)



(d)



(e)



(f)

图 20.2 购物系统浏览效果

(a) 购物车页面；(b) 确定订单页面；(c) 确定收货信息页面；
(d) 提交订单信息页面；(e) 订单列表页面；(f) 订单明细页面

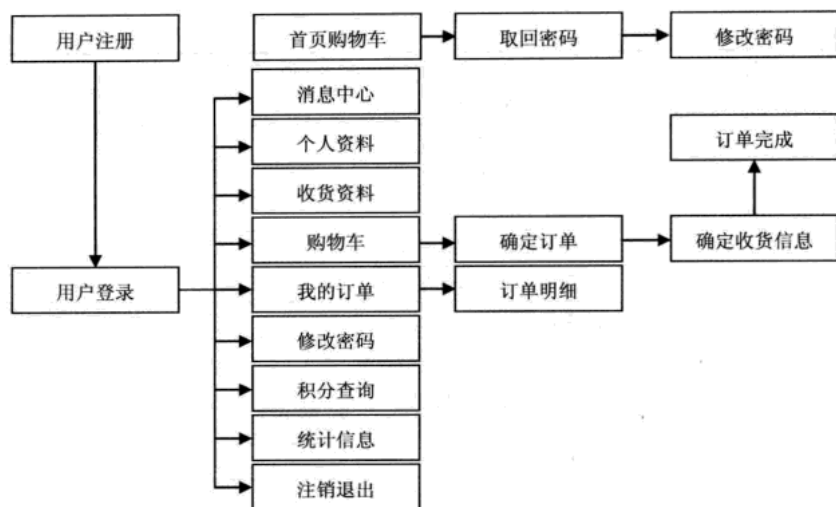


图 20.3 网上购物前台设计流程图



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“19-20”/“电子商城前台功能实现”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。

20.2 数据库设计

数据库设计可以参照第 19 章介绍，本章页面主要是查询数据库中的 orders 数据表，也需要参考查询 product 和 user 表中数据，在涉及收货方式和支付方式时还需要查询 delivery 表。

20.3 页面设计

网上购物用户账户管理和购物系统共有 21 个文件，具体说明见表 20.1。

表 20.1 用户账户管理和购物系统文件列表

文 件 名	说 明
car.asp	购物车页面，显示用户存放在购物车中的商品，并进行操作
cart.asp	订单确定页面，在该页面中可以确定购买哪些商品以及商品数量
changepps.asp	修改用户密码页面
checkout.asp	结账页面，在该页面中确定收货方式和支付方式
chkorder.asp	订单明细页面，在该页面中显示某笔订单的详细信息
deposit.asp	用户积分查询页面
forget.asp	取回密码页面
help.asp	用户帮助页面

续表

文 件 名	说 明
icar.asp	首页购物车页面
login.asp	用户登录页面
logout.asp	注销用户文件
muser.asp	消息中心页面
myorder.asp	详细显示我的订单列表页面
recinfo.asp	收货资料修改与保存页面
Reg.asp	用户注册页面
Repws.asp	取回密码之后修改密码页面
Statinfo.asp	统计信息显示页面
Tiao.asp	用户注册新用户之前要确定的同意条款
Uleft.asp	用户账户菜单页面
Userinfo.asp	用户个人资料显示与修改页面
Vorder.asp	订单完成之后显示的订单信息页面

由于前台系统页面设计相对简单,用户可以参照本书光盘中提供的实例模板就可以轻松理解,因此就不再赘述。

20.4 购物系统基本功能

购物系统基本功能主要包括用户信息管理,以及购物基本功能。下面就来介绍如何实现这些功能。


20.4.1 用户注册

用户要想实现网上购物,必须先在系统中注册一个账户,并登录之后才可以拥有购买商品的权利。在前面章节中曾详细介绍过用户注册实例,不过本系统比较复杂,如图 20.4 所示。

本用户注册页面注册信息比较多,且对信息的检测与验证比较周密完善,符合实际应用需要。实现本功能的具体操作步骤如下。

01 打开 reg.asp 文件,为了保证插入用户记录时使数据库中的字段信息完整,应先插入 7 个隐藏域,具体操作就不再介绍。代码如下:

```
<input type="hidden" id="deliverymethord" name="deliverymethord" value=0>
<input type="hidden" id="paymethord" name="paymethord" value=0>
<input type="hidden" id="score" name="score" value=0>
<input type="hidden" id="postcode" name="postcode" value=0>
<input type="hidden" id="loginnum" name="loginnum" value=0>
<input type="hidden" id="lastvst" name="lastvst" value="<%=now()%">
<input type="hidden" id="adddate" name="adddate" value="<%=now()%">
```

02 定义“插入记录”服务器行为。打开【服务器行为】面板,单击面板左上角的按钮,在弹出的下拉菜单中选择【插入记录】命令,打开【插入记录】对话框,具体设置如图 20.5 所示。

在【连接】下拉列表中选择前面定义好的数据源 conn;在【插入到表格】下拉列表中选择要插入的数据表 user;在【插入后,转到】文本框中设置插入记录成功后要跳转到 muser.asp。在【表元素】列表框中分别设置每个表单对象中的数据要插入数据库中 user 表中哪个字段。



用户在命名表单对象时，最好与数据表中的字段保持一致，这样会避免出错，而且系统会自动设置表单对象要插入的数据库字段。

设置完毕，单击【确定】按钮即可实现插入记录功能。



图 20.4 用户注册页面



图 20.5 设置【插入记录】对话框

- 03** 切换到【代码】视图，找到插入记录代码段，一般位于代码页顶部（<html>标签前面），为了避免和其他代码混合，建议用户把该代码段复制到表单代码区域前。
- 04** 在插入记录代码段前面输入下面代码。

```
<%  
if NOT isempty(request("RegSubmit")) then '判断用户是否是提交注册信息  
    if session("regtimes")=1 then '如果已经注册，则提示不能再注册  
        call MsgBox("对不起，您刚注册过用户!", "Back", "None")  
        response.end  
    end if  
    '定义一个记录集，验证数据库中是否已经存在注册用户  
    set rs=server.CreateObject("adodb.recordset")  
    rs.open "select username,useremail from [user] where username='"&trim(request  
("username"))&"' or useremail='"&trim(request("useremail"))&"'",conn,1,1  
    if not rs.eof and not rs.bof then  
        call MsgBox("您输入的用户名或Email地址已存在，请返回重新输入!", "Back", "None")  
    else '如果没有该用户，则可以允许把注册信息插入数据库中
```

- 05** 在插入记录代码区域中找到下面代码段，然后在其中插入 4 句代码，用来保存用户信息到 Cookies(cookieName)数组变量中，同时设置用户已登录变量 session("regtimes")和保存用户名到 Session("MM Username")中。

```
response.Cookies(cookieName) ("username")=trim(request("username"))
response.Cookies(cookieName).expires=date+1
Session("MM_Username") = trim(request("username"))
session("regtimes")=1
session.Timeout=1
```

```
Response.Redirect(MM_editRedirectUrl)
End If
End If
%>
```

06 在插入记录代码段后面再输入下面代码:

```
<%
'结束条件, 并关闭记录集rs
end if
rs.close
set rs=nothing
end if
%>
```

20.4.2 用户登录

当用户注册完毕, 系统会自动登录; 如果退出登录, 再次购买商品就需要重新登录了, 登录页面设计如图 20.6 所示。

实现登录功能的具体操作步骤如下。

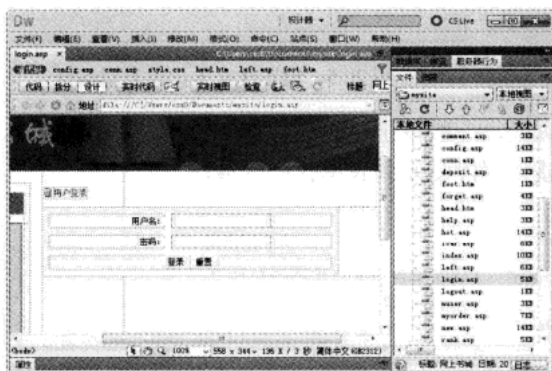
01 打开 login.asp 文件, 在【服务器行为】面板中单击面板左上角的  按钮, 在弹出的下拉菜单中选择【用户身份验证】|【登录用户】命令, 打开【登录用户】对话框, 设置如图 20.7 所示。

图 20.6 用户登录页面设计效果

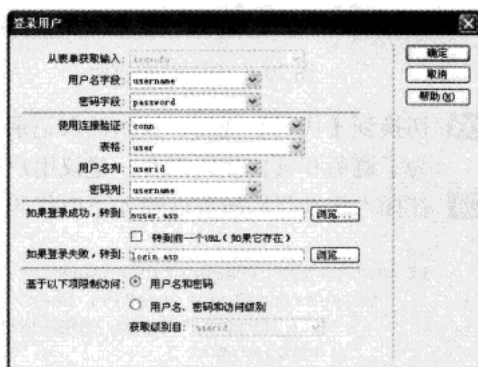


图 20.7 设置【登录用户】对话框

02 切换到【代码】视图, 找到用户登录代码段, 选中并移至登录表单代码区域前, 删除页面顶部代码所自动插入的下面代码行, 避免与 <!--#include file="conn.asp"--> 文件中的变量定义发生冲突。在 conn.asp 文件中包含下面的代码行。

```
<!--#include file="Connections/conn.asp" -->
```

03 在用户登录代码段中增加下面代码, 用来保存用户信息到 Cookies(cookieName) 数组变量中, 同时设置用户已登录变量 session("regtimes") 和保存用户名到 Session("MM_Username") 中。

```

'-----
response.Cookies(cookieName)("username")=MM_valUsername
response.Cookies(cookieName)("vip")=MM_rsUser("vip")
sql="Update [user] set lastvst=#"&now()&"#,loginnum="#&MM_rsUser("loginnum")+1&"
where userid="#&MM_rsUser("userid")
conn.execute sql
'-----

If (MM_fldUserAuthorization <> "") Then
    Session("MM_UserAuthorization") = CStr(MM_rsUser.Fields.Item(MM_fldUserAutho-
rization).Value)
Else
    Session("MM_UserAuthorization") = ""

```

```

End If
if CStr(Request.QueryString("accessdenied")) <> "" And false Then
    MM_redirectLoginSuccess = Request.QueryString("accessdenied")
End If
MM_rsUser.Close
Response.Redirect (MM_redirectLoginSuccess)
End If
MM_rsUser.Close
Response.Redirect (MM_redirectLoginFailed)
End If
%>

```

- 04** 为了避免用户未登录就进入购物,需要在该购物系统中每个文件中先检验用户是否登录。读者可以新建 `chk.asp` 文件,专门用来验证用户是否登录。`chk.asp` 文件中的代码如下:

```

<%
if Session("MM_Username")="" then
    call MsgBox("请先登录!", "GoUrl", "login.asp")
    response.End
end if
%>

```

- 05** 在其他页面的代码顶部用下面语句行包含该文件。

```

<!--#include file="chk.asp"-->

```

20.4.3 注销用户

登录成功之后,如果用户想退出登录,可以注销用户信息,避免别人利用用户信息进行操作。实现的方法比较简单,打开 `logout.asp` 文件,输入下面代码即可:

```

<!--#include file="config.asp"-->
<%
response.Cookies(cookieName).Expires = NOW() -1
response.Cookies(cookieName)("username")=""
response.Cookies(cookieName)("vip")=""
response.Write "<script language=javascript>alert('您已成功注销!');";
response.redirect "index.asp"
%>

```

20.4.4 消息中心

消息中心页面显示当用户登录成功之后显示的信息,如图 20.8 所示。具体操作步骤如下。

- 01** 打开 `muser.asp` 文件,首先定义一个记录集,查询用户消息,【记录集】对话框具体设置如图 20.9 所示。

- 02** 切换到【代码】视图,修改其中的查询条件,具体代码如下:

```

<%
Dim book_MMColParam
book_MMColParam = "1"
If (Request.Cookies("username") <> "") Then
    book_MMColParam = Request.Cookies(cookieName)("username")
End If
%>

```

- 03** 在记录集代码段后面输入下面代码。

```

<%
if book.eof and book.bof then
    call MsgBox("错误: 此用户记录不存在!", "Back", "None")

```

```
end if
%>
```



图 20.8 消息中心页面

选定其中的 user_id、username、useremail、password 和 book5 个字段

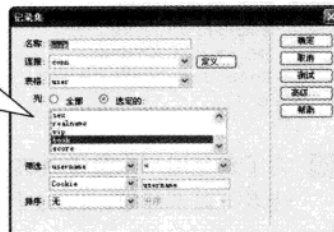


图 20.9 定义记录集 book

04 把记录集 book 中的 book 字段插入页面中，如图 20.10 所示。

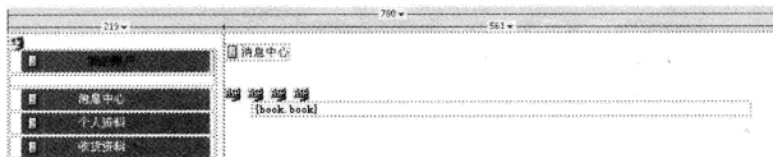


图 20.10 插入记录集字段

20.4.5 个人资料

当用户登录之后，可以在用户系统中浏览和修改个人资料，个人资料页面设计如图 20.11 所示。

该页面主要运用了更新记录服务器行为和绑定记录技术，具体操作步骤如下。

01 打开 userinfo.asp，先定义一个记录集，命名为 user，用来查询指定用户名的资料记录，设置的【记录集】对话框如图 20.12 所示。



图 20.11 个人资料页面

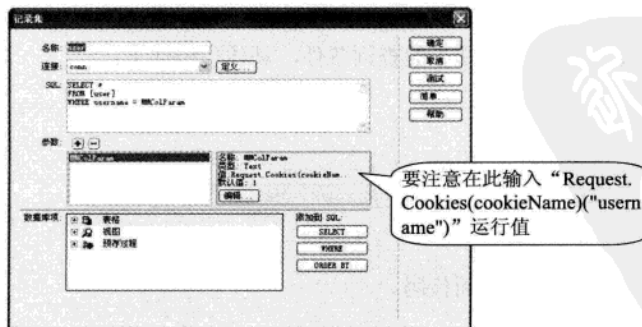


图 20.12 定义记录集



用户可以在【绑定】面板中，单击鼠标右键在弹出的快捷菜单中选择【拷贝】命令，通过粘贴可以快速实现定义，然后根据需要适当修改其设置的参数。

02 把记录集中的各个字段绑定到表单对象中，如图 20.16 所示。

03 选中“性别”下拉菜单，然后在属性面板中单击【动态】按钮，在打开的【动态列表/菜单】对话框中，设置下拉菜单的选取值，设置如图 20.17 所示。

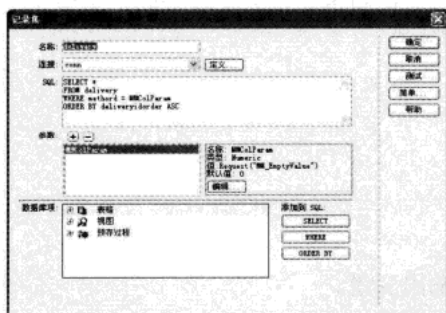


图 20.16 绑定记录集字段

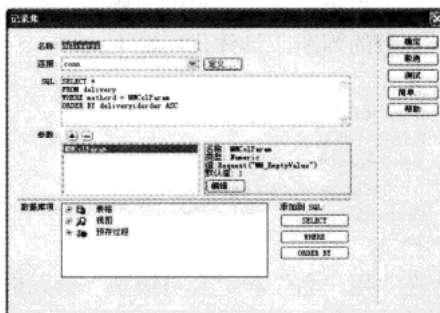


图 20.17 设置性别下拉菜单的选取值

04 设置送货方式和支付方式的动态列表值。先定义两个记录集以获取数据库中的送货方式和支付方式，如图 20.18 所示。



(a)

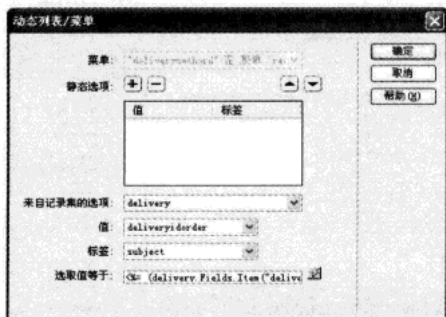


(b)

图 20.18 定义送货方式和支付方式记录集

(a) 送货方式；(b) 支付方式

05 分别把记录集绑定到列表框属性值上，方法是选中列表框，单击属性面板中【动态】按钮，在打开的【动态列表/菜单】对话框中设置属性的动态值，如图 20.19 所示。



(a)



(b)

图 20.19 设置送货方式和支付方式动态列表框

(a) 送货方式；(b) 支付方式

- 06 在【服务器行为】面板中定义一个更新记录服务器行为,【更新记录】对话框设置如图 20.20 所示。



图 20.20 设置【更新记录】对话框

20.4.7 修改密码

用户除了可以更改个人资料和收货资料外,也可以修改个人密码,页面设计效果如图 20.21 所示。

完成本页功能需要先定义一个记录集来获取用户密码,然后比较表单中的输入密码,如果输入正确则允许用户修改密码,否则要求重新输入,实现的操作步骤如下。

- 01 打开 changepws.asp 文件,先定义一个记录集 passwordold,【记录集】对话框设置如图 20.22 所示。

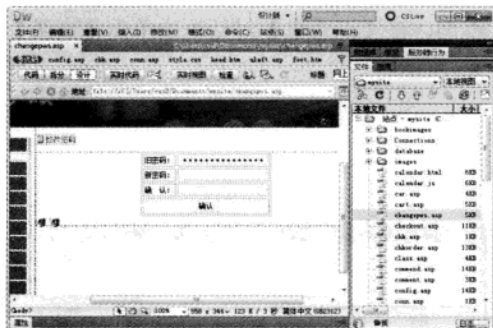


图 20.21 修改个人密码页面

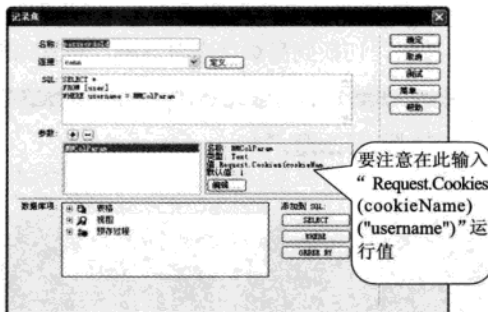


图 20.22 设置【记录集】对话框

- 02 切换到【代码】视图,找到记录集代码段,在其后面输入下面代码,用来判断用户输入密码是否正确,如果错误则返回重新输入,如果正确则允许用户修改密码。

```
<%
if NOT isempty(request("ChangePwsSubmit")) then
    if (trim(request("password"))) <> trim(passwordold("password")) then
        call MsgBox("对不起,您输入的原密码错误!", "none", "none")
    else

```

- 03 在【服务器行为】面板中定义一个更新记录服务器行为,【更新记录】对话框设置如图 20.23 所示。

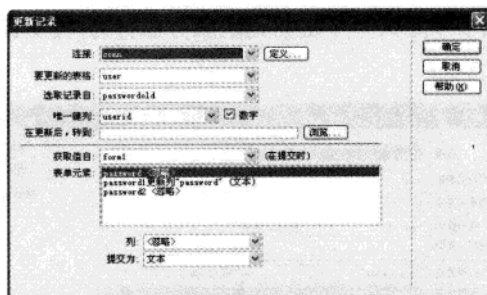


图 20.23 设置【更新记录】对话框

- 04 再次切换到【代码】视图，找到更新记录代码段，在其后面输入下面代码，结束条件结构，并提示密码更改成功。

```
<%
    call MsgBox("密码更改成功!", "none", "none")
end if
end if
%>
```

20.4.8 购物列表

购物列表页面主要用来显示用户选购的商品，浏览效果如图 20.24 所示。

在购物车页面内除了列表显示选购的商品，也可以删除不想要的商品，或者选取其中部分或全部商品去下订单，也可以继续选购商品。实现该页面功能的操作步骤如下。

- 01 打开 car.asp 文件，定义一个记录集 car，获取选购商品信息，由于该记录集查询涉及产品表 product 和订单表 orders，因此需要交叉查询，在【记录集】对话框中单击【高级】按钮，切换到高级状态对话框，设置如图 20.25 所示。

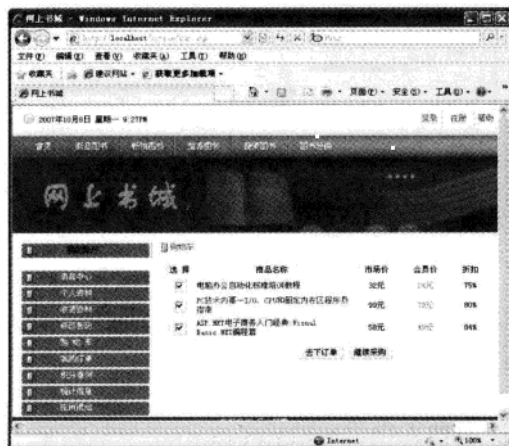


图 20.24 购物车页面浏览效果

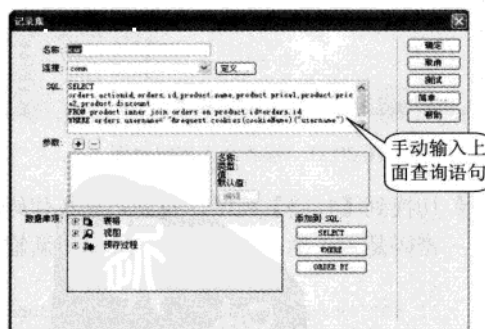


图 20.25 设置【记录集】对话框

在【SQL】文本框中输入下面查询语句。

```
SELECT orders.actionid,orders.id,product.name,product.price1,product.price2, product.
discount
FROM product inner join orders on product.id=orders.id
```

```
WHERE orders.username='&request.cookies(cookieName)("username")&' and orders.state=6
```

- 02** 把记录集中的字段绑定到页面单元格中,并定义该行为重复区域,【重复区域】对话框设置如图 20.26 所示。定义后的页面如图 20.27 所示。
- 03** 选中该行中的复选框,在属性面板中设置【选定值】文本框为“<%= (car.Fields.Item("id").Value)%>”。
- 04** 选中“删除”图标,在属性面板的【链接】文本框中输入下面代码,实现对该条商品的删除操作。

```
car.asp?action=del&actionid=<%= (car.Fields.Item("actionid").Value)%>
```

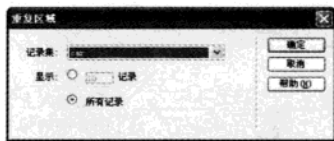


图 20.26 设置【重复区域】对话框

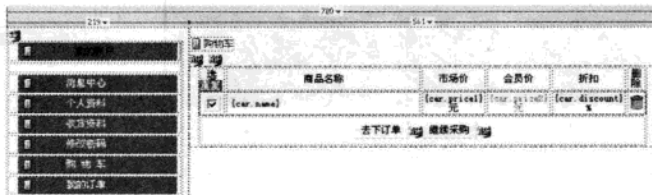


图 20.27 插入记录集的页面

- 05** 选中【商品名称】列中的插入记录集字段 {car.name},然后在属性面板的【链接】文本框中输入下面代码实现浏览该条商品的详细信息。

```
vpro.asp?id=<%= (car.Fields.Item("id").Value)%>
```

- 06** 切换到【代码】视图,把定义的记录集 car 转移到插入行前面。在代码页顶部输入下面代码,用来获取用户操作信息,当用户单击“删除”图标后,会删除该条商品记录。

```
<%
dim id,username,action
action=request.QueryString("action")
username=trim(request.cookies(cookieName)("username"))
id=request.QueryString("id")
set rs=server.CreateObject("adodb.recordset")
if action="del" then
    conn.execute "delete from orders where actionid='&request.QueryString("actionid")'
    response.redirect "car.asp"
end if
%>
```

- 07** 在代码页底部,分别在“去下订单”和“继续采购”按钮的属性中增加动作,代码如下:

```
<input type=submit name=Submit value=去下订单 onclick="location='cart.asp'">
<%
if action<>"addtocart" then
%>
<input type=button name=Submit2 value=继续采购 onclick=javascript:window.close()>
<%
end if
%>
```

20.4.9 下订单

挑选到购物车中的商品需要用户确定是否要买,并设置要购买的数量。这些功能都需要在下订单页面中完成,下订单页面浏览效果如图 20.28 所示。

实现该页面功能的具体操作步骤如下。

- 01** 打开 cart.asp 文件, 先定义一个记录集 cart。在打开的【记录集】对话框中单击【高级】按钮, 切换到高级状态, 在【SQL】文本框中输入下面查询语句, 设置的【记录集】对话框如图 20.29 所示。

```
SELECT id,name,price1,price2,vipprice,discount
FROM product
WHERE id in ("&id&")
ORDER BY id
```

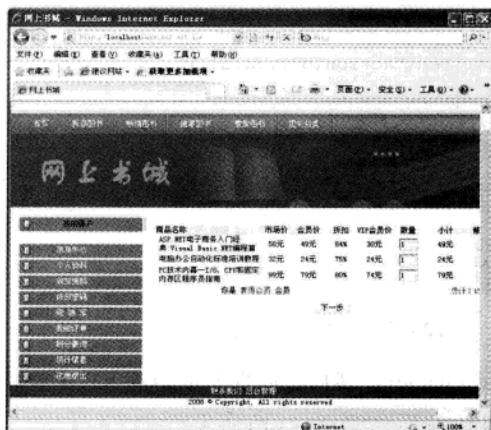


图 20.28 下订单页面浏览效果

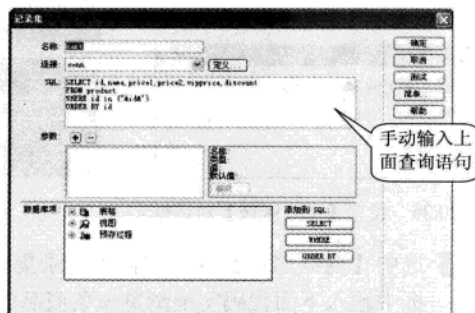


图 20.29 设置【记录集】对话框

- 02** 把记录集中的字段绑定到页面单元格中, 并定义该行为重复区域, 【重复区域】对话框设置如图 20.30 所示。定义后的页面如图 20.31 所示。

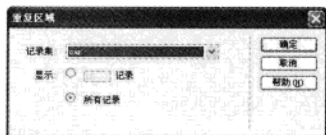


图 20.30 设置【重复区域】对话框



图 20.31 插入记录集的页面

- 03** 切换到【代码】视图, 把定义的 cart 记录集转移到插入记录集字段行前, 然后在代码页顶部输入下面代码。

```
<% 判断购物车内是否选购了商品
dim id,i,iii ,books,bookscount,product
id=request("id")
if id="" then
    call MsgBox("你的购物车内没有商品!", "Back", "None")
    response.end
end if
%>
```

- 04** 在 cart 记录集代码段后面及重复区域代码段前面输入下面代码, 获取用户在购物车页面中选中多少条记录。

```
<%
    iii=0
    bookscount=request.QueryString("id").count
%>
```

05 找到重复区域代码段, 在其中输入下面代码。

```
<%
    dim quantity
    Quantity = CInt( Request( "shop"&cart("id")) )
    If Quantity <=0 Then Quantity = 1
%>
    <tr align="center"><td style='PADDING-LEFT: 5px' align="left"><%= (cart.Fields.Item
("name").Value)%>
        <input type=hidden name=name value=<%= (cart.Fields.Item("name").Value)%>>    </td><td
><%= (cart.Fields.Item("price1").Value)%>元</td>
        <input type=hidden name=price2 value=<%= (cart.Fields.Item("price2").Value)%>>
        <td><%= (cart.Fields.Item("price2").Value)%>元</td>
        <td><%= (cart.Fields.Item("discount").Value)*100%>%</td>
        <td><%= (cart.Fields.Item("vipprice").Value)%>元</td>
        <td><input name="shop"&cart("id")%>" type="text" size="3" value="<%=Quantity%"
onKeyPress= "return regInput(this, /^[0-9]*$/,"
        String.fromCharCode(event.keyCode))"onpaste = "return regInput(this, /^[0-9]*$/,"
window.clipboardData.getData('Text'))"ondrop= "return regInput(this, /^[0-9]*$/,"event.
dataTransfer.getData('Text'))"> </td><td>
        <% '定义记录集获取用户当前级别和身份
        Dim rsvip, strvip, strdeposit, txtvip
        set rsvip=server.CreateObject("adodb.recordset")
        rsvip.open "select vip from [user] where username='"&request.cookies(cookieName)
("username")&"', conn, 1, 1
        strvip = rsvip("vip")
        if strvip = true then
            txtvip = "VIP会员"
        if Quantity<=1 then
            '计算不同用户应付价钱
            response.write cart("vipprice")*1&"元"
        else
            response.write cart("vipprice")*Quantity&"元"
        end if
        iii=cart("vipprice")*Quantity+iii
        else
            txtvip = "普通会员"
        if Quantity<=1 then
            response.write cart("price2")*1&"元"
        else
            response.write cart("price2")*Quantity&"元"
        end if
        iii=cart("price2")*Quantity+iii
        end if
        %>
    </td>
```

06 在【代码】视图中, 设置“修改”按钮的修改动作, 代码如下:

```
<input type="submit" name="Submit" value="修改" onClick="this.form.action='cart.
asp?action=modify';this.form.submit()">
```

07 在【代码】视图中, 设置“下一步”按钮的动作, 代码如下:

```
<input type="submit" name="Submit2" value="下一步" onClick="this.form.action='checkout.
asp';this.form.submit()">
```

20.4.10 结账

完成订单的确定之后就可以到收银台结账了。进入结账页面, 在该页面中将要求用户确定收货资料, 并获取所有订单记录编号, 结账页面设计如图 20.32 所示。

实现该页面功能的操作步骤如下。



图 20.32 结账页面浏览效果

- 01 打开 checkout.asp 文件，先定义记录集 user 以获取用户信息，设置【记录集】对话框如图 20.33 所示。
- 02 把记录集字段绑定到页面单元格中，再定义两个记录集分别用来获取送货方式和支付方式，有关动态表单属性及记录集定义可以参见第 20.4.6 小节中的详细介绍，该页面与收货资料页面基本相同。
- 03 切换到【代码】视图，定义一个记录集 product，用来获取选购商品编号，并把它们绑定到隐藏域的值上，实现选购信息的传递。定义的记录集 product 对话框如图 20.34 所示。

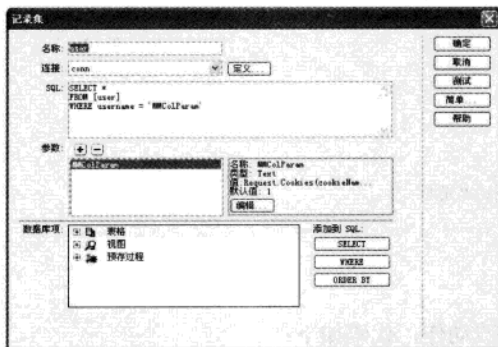


图 20.33 设置【记录集】对话框

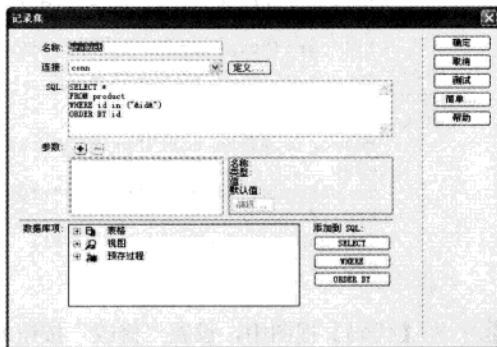


图 20.34 设置【记录集】对话框

- 04 在表单中插入一个隐藏域，在属性面板中设置动态名称和值，如图 20.35 所示。隐藏域的完整代码如下：

```
<input name="<%= "shop"&product("id") %>" type="hidden" value="<%= cint(request("shop"&product("id"))) %>">
```

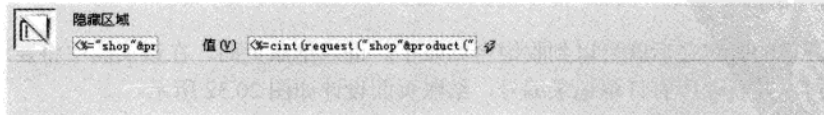


图 20.35 设置隐藏域属性面板

05 选中该隐藏域, 定义重复区域服务器行为。【重复区域】对话框设置如图 20.36 所示。

06 在定义重复区域时, 可能循环代码会包括其他代码, 因此需要把循环语句分别转移到隐藏域前后, 也同时把记录集代码段也移至隐藏域前后。调整后的完整代码如下:

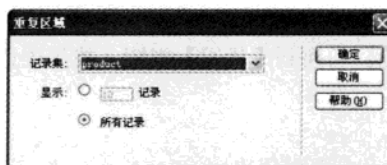


图 20.36 设置【重复区域】对话框

```
<% '转移后的记录集代码段
Dim product
Dim product_numRows

Set product = Server.CreateObject("ADODB.Recordset")
product.ActiveConnection = MM_conn_STRING
product.Source = "SELECT * FROM product WHERE id in ("&id&") ORDER BY id"
product.CursorType = 0
product.CursorLocation = 2
product.LockType = 1
product.Open()

product_numRows = 0
%>
<% '转移后的重复区域定义变量代码
Dim Repeat1__numRows
Dim Repeat1__index

Repeat1__numRows = -1
Repeat1__index = 0
product_numRows = product_numRows + Repeat1__numRows
%>
<% '转移后的重复区域循环语句
While ((Repeat1__numRows <> 0) AND (NOT product.EOF))
%>

<% '结束重复区域
Repeat1__index=Repeat1__index+1
Repeat1__numRows=Repeat1__numRows-1
product.MoveNext()
Wend
%>
<% '结束记录集
product.Close()
Set product = Nothing
%>
```

20.4.11 完成订单

当用户在结账页面单击【提交订单】按钮后, 即完成订单操作, 这时页面将会显示当前用户所提交订单的全部信息, 如图 20.37 所示。

实现该页面功能的操作步骤如下。

01 打开 vorder.asp 文件, 定义一个记录集 rsvip 以查询用户的级别和身份, 设置的【记录集】对话框如图



图 20.37 完成订单后生成的详细信息页面

20.38 所示。

- 02** 切换到【代码】视图，找到该记录集代码段，在其后面输入下面代码获取用户级别和身份。

```
<%
Dim strvip
strvip = rsvip("vip")
%>
```

- 03** 在代码页底部把记录集 rsvip 结束语句转移到上面代码的下面，实现关闭该记录集，避免以后相互混合出错。



本页面需要定义 7 个记录集，当记录集很多时就容易出错。为了避免出错，用户应定义完一个记录集就立即把该记录集代码段转移到应用该记录集的单元格左右，不需要时立即关闭。

本例步骤是按照在代码页中从上到下的顺序进行介绍，用户应注意代码前后顺序。

- 04** 在下面输入存储商品编号、用户编号和当前时间信息的代码。

```
<%
dim shijian,goods
dim userid,id,score
id=request("id")
userid=request("userid")
shijian=now()
goods=year(shijian)&month(shijian)&day(shijian)&hour(shijian)&minute(shijian)&second(shijian)
%>
```

- 05** 在其下面定义一个记录集 rs2，设置的【记录集】对话框如图 20.39 所示。

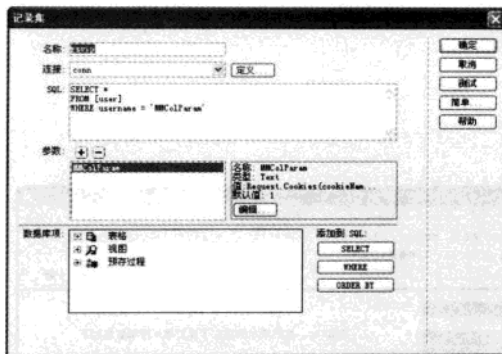


图 20.38 设置【记录集】对话框

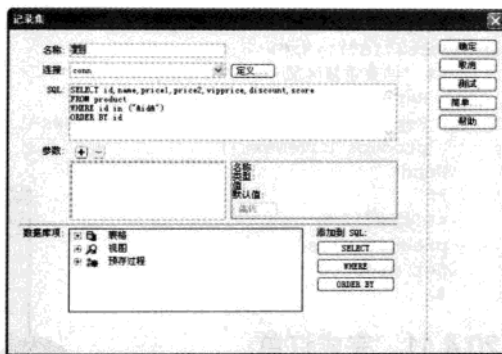


图 20.39 设置【记录集】对话框

- 06** 输入下面代码，实现循环保存订单信息到数据库中。

```
<%
goods=year(shijian)&month(shijian)&day(shijian)&hour(shijian)&minute(shijian)&second(shijian)

do while not rs2.eof
%>
<%
set rs=server.CreateObject("adodb.recordset")
rs.open "select * from orders",conn,1,3
```

```

rs.addnew
score=score+rs2("score")
rs("username")=trim(request.cookies(cookieName)("username"))
rs("id")=rs2("id")
rs("actiondate")=shijian
rs("productnum")=Cint(Request("shop"&rs2("id")))
rs("state")=1
rs("goods")=goods
rs("postcode")=int(request("postcode"))
rs("recepit")=trim(request("recepit"))
rs("address")=trim(request("address"))
rs("paymethord")=int(request("paymethord"))
rs("deliverymethord")=int(request("deliverymethord"))
rs("sex")=int(request("sex"))
rs("comments")=(trim(request("comments")))

if strvip = true then
    rs("paid")=rs2("vipprice")*Cint(Request("shop"&rs2("id")))
else
    rs("paid")=rs2("price2")*Cint(Request("shop"&rs2("id")))
end if
rs("realname")=trim(request("realname"))
rs("useremail")=trim(request("useremail"))
rs("usertel")=trim(request("usertel"))
rs("userid")=userid
rs.update
rs.close
conn.execute "delete from orders where username='"&request.cookies(cookieName)
("username")&"' and id in ("&id&") and state=6"
%>
<%
rs2.movenext
loop
rs2.close

rs2.open "select score from [user] where userid="&userid,conn,1,3
rs2("score")=rs2("score")+int(score)
rs2.Update
%>
<%
rs2.Close()
Set rs2 = Nothing
%>

```

07 在下面定义一个记录集，以交叉查询商品和订单信息，在【记录集】高级对话框的【SQL】文本框中输入下面 SQL 语句，【记录集】对话框设置如图 20.40 所示。然后切换到【代码】视图，在记录集代码段中删除 SQL 字符串最后的引号。

```

SELECT product.id,product.name,product.
price1,vipprice,product.price2,orders.sex,
orders.realname,orders.recepit,orders.goods,
orders.postcode,orders.comments,orders.
paymethord,orders.deliverymethord,orders.
paid,orders.productnum
FROM product inner join orders on
product.id=orders.id
WHERE orders.username='"&request.cookies(cookieName)("username")&"' and state=1 and
goods='"&goods&"'

```

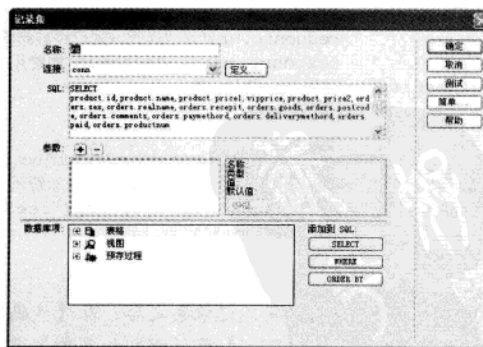


图 20.40 设置【记录集】对话框

- 08** 把 rs 记录集中的字段插入到页面单元格中，然后设置该行为重复区域，【重复区域】对话框设置如图 20.41 所示。插入记录集字段的页面如图 20.42 所示。



图 20.42 插入记录集的页面

- 09** 切换到【代码】视图，在上面定义的重复区域内再定义一个记录集 `rsvip1`，把该记录集代码段转移到重复区域代码段内。

- 10** 在记录集 `rsvip1` 下面输入代码。

```
<%
if rsvip1("vip") = true then
%>
<td align="center"><%=rs("vipprice") * rs("productnum")%>元</td>
</tr>
<%
    iii=rs("vipprice") * rs("productnum")+iiii
else
%>
    <td align="center"><%=rs("price2") * rs("productnum")%>元</td></tr>
<%
    iii=rs("price2") * rs("productnum")+iiii
end if
%>
```

- 11** 定义一个记录集 rs3，设置【记录集】对话框如图 20.43 所示。然后切换到【代码】视图，在记录集代码段中删除 SQL 字符串最后的引号。

- 12** 在记录集 rs3 代码下面输入代码。

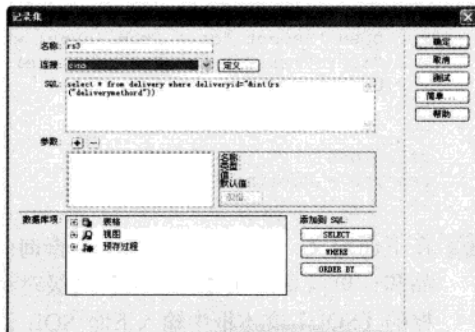
[illegible]

图 20.43 定义 rs3 记录集



当用户定义记录集之后，在【代码】视图中删除了记录集的SQL查询语句，则该记录集将不被【绑定】面板识别和显示，当加上引号之后又可以被【绑定】面板识别和显示。

- 13** 在页面收货人后面输入下面代码，用来判断收货人性别并设置称呼。

```
<%response.Write trim(request("recepit"))
```

```
if request("sex")=0 then  
response.Write "&nbsp;  (先生)"  
else  
response.Write "&nbsp;  (女士)"  
end if%>
```

- 14** 定义记录集 rs4,【记录集】对话框设置如图 20.44 所示。然后切换到【代码】视图,在记录集代码段中删除 SQL 字符串最后的引号。

- 15** 在记录集 rs4 下面输入代码实现动态显示送货方式。

```
<%if rs4.eof and rs4.bof then
    response.write "方式已经被删除"
else
    response.Write trim(rs4("subject"))
end if %>
```

- 16** 定义记录集 rs5,【记录集】对话框设置如图 20.45 所示。然后切换到【代码】视图,在记录集代码段中删除 SQL 字符串最后的引号。

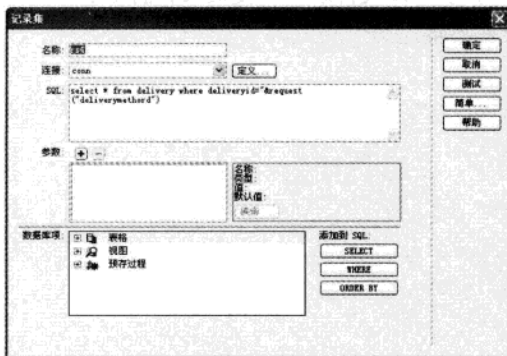


图 20.44 定义 rs4 记录集

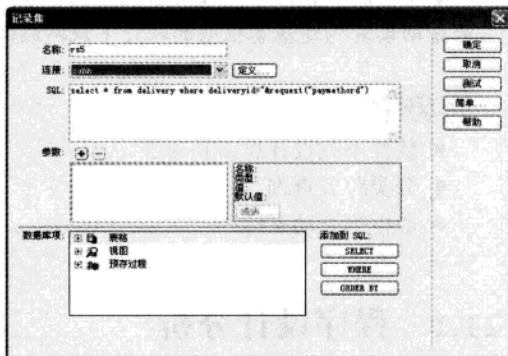


图 20.45 定义 rs5 记录集

- 17** 在记录集 rs5 下面输入代码实现动态显示支付方式:

```
<%if rs5.eof and rs5.bof then
    response.write "方式已经被删除"
else
    response.Write trim(rs5("subject"))
end if %>
```

20.5 实践总结

本章主要讲述了购物系统的设计和 basic 功能实现，用户应着重掌握 Cookies 数据集合的使用。当浏览某网站时，Cookies 可以记录用户 ID、密码、浏览过的网页及停留的时间等信息。当再次进入该网站时，系统能够通过读取 Cookies 数据集合中的数据，获取相关信息，并执行相应的动作，有关 Cookie 对象更详细的介绍可以参阅第 3.9 节介绍。

第 21 章

Dreamweaver

文件下载演示模块

国内各大门户网站及 IT 资讯网站都有下载频道。另外还有一些专业下载网站，例如，华军软件园、天空软件站和太平洋下载频道等。用户除了可以直接从互联网上下载软件外，还能在互联网上寻找到各种免费的专业下载工具，以帮助用户提高下载资料的速度和丰富下载功能。

本章将演示软件下载应用系统的基本模式和功能，不深入讲解下载技术本身的实现方法，用来帮助读者构建基本框架的软件下载网站的开发思路。

技术要点

- 软件下载设计思路和方法
- 分类跨表查询
- 数据列表动态排序

21.1 程序设计分析

读者经常上网下载软件、音乐、动画等资料，虽然不同网站下载速度有快有慢，但下载方法都是相同的，即提供一个不被浏览器解析的文件格式的超链接，当用户单击这个超链接时会自动弹出下载对话框，允许把该文件从服务器上保存到本地计算机中。

本章将介绍一个简单的软件下载站点的设计方法和技巧，同时还介绍如何实现下载软件的分类和排序。整个系统可以分为 3 级链接关系，即在主页选择一种软件分类方式，然后在分类页面中寻找需要下载的软件，单击该文件的链接，进入该软件的具体介绍页面，在该页面中单击“下载”链接文本，就可以下载该软件。整个系统的浏览效果如图 21.1 所示。程序设计流程如图 21.2 所示。

请读者注意，本系统仅是一个演示实例，不提供真正的软件下载，只提供 1 个模拟压缩文件演示软件下载过程。页面用到的各个软件的相关信息只是模仿相关下载软件站点所提供的宣传信息，提供这些信息的目的是为读者提供一个模拟的学习环境。同时，读者在互联网上真正下载软件时，一定要先用杀毒软件对下载的软件进行杀毒，否则容易中毒。下载软件要上权威下载网站，尽量不要上小网站进行下载。

要实现本例功能，读者需要学习和解决下面技术问题。

- 编写 SQL 查询语句，实现多种分类查询。
- 利用 SQL 语句实现下载列表动态排序。
- 具体软件的下载方法和数据结构设计。

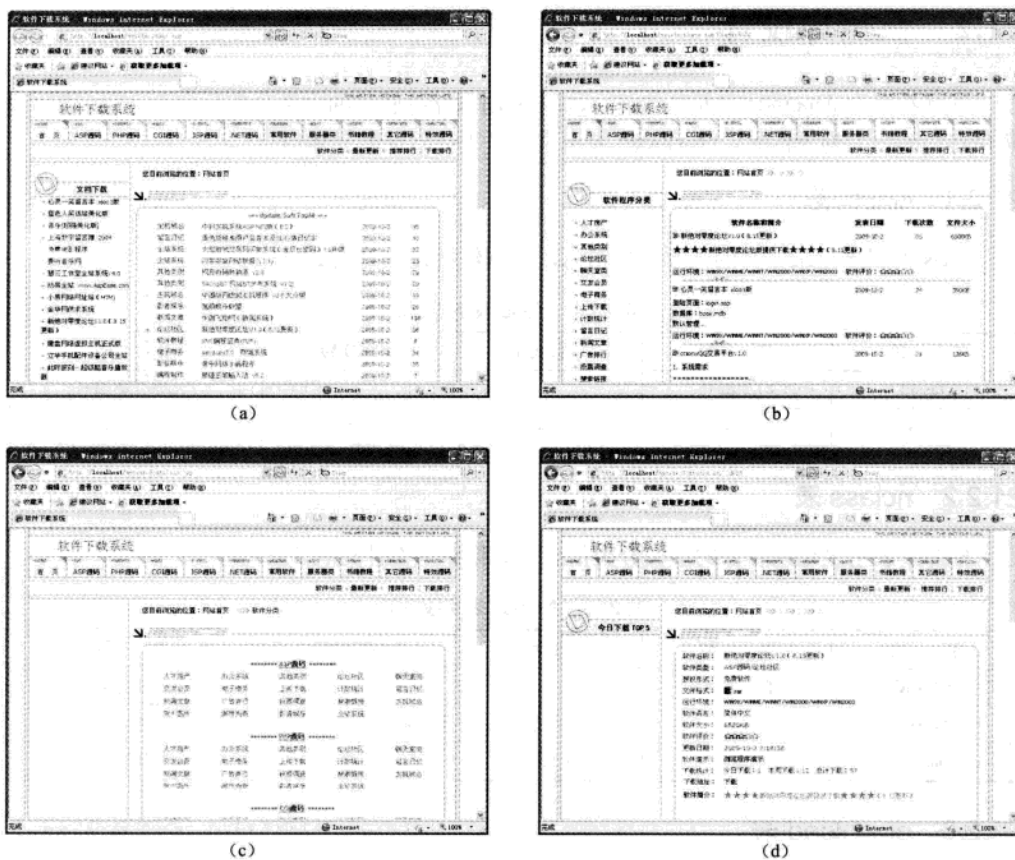


图 21.1 软件下载浏览效果

(a) 软件下载主页; (b) 软件分类列表页面; (c) 软件分类列表页面; (d) 具体下载页面

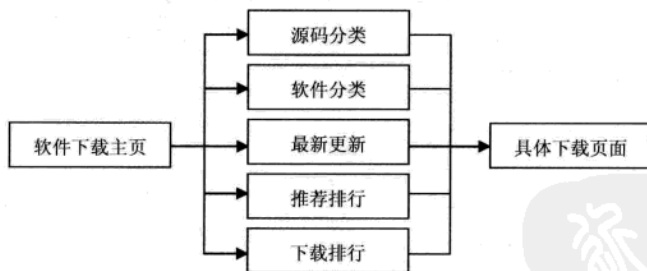


图 21.2 软件下载设计流程图



提示

学习本实例之前,读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录,启动 Dreamweaver CS5 创建一个动态站点,远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”,【访问】应为“本地/网络”。

把本书光盘中的“21”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构,并帮助读者建立了数据库连接。

21.2 数据库设计

本例数据库 data.mdb 包括 3 个表，class 表用来储存软件的大类列表信息，nclass 表用来储存软件的小类列表信息，download 表用来储存下载软件的具体信息。

21.2.1 class 表

class 表用来记录软件的大类分类信息，数据结构设计见表 21.1。

表 21.1 class 表字段列表

字 段	类 型	字段大小	必填字段	说 明
classid	自动编号	-	-	大类分类编号
class	文本	255	否	大类分类名称

21.2.2 nclass 表

Nclass 表用来记录软件的小类分类信息，数据结构设计见表 21.2。

表 21.2 Nclass 表字段列表

字 段	类 型	字段大小	必填字段	说 明
nclassid	自动编号	-	-	小类分类编号
nclass	文本	255	否	小类分类名称
classid	数字	长整型	否	大类分类编号

21.2.3 download 表

download 表用来记录具体软件的详细信息，数据结构设计见表 21.3。

表 21.3 download 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	软件编号
filename	文本	255	否	软件下载地址
filename1	文本	255	否	软件下载地址 1
filename2	文本	255	否	软件下载地址 2
filename3	文本	255	否	软件下载地址 3
filename4	文本	255	否	软件下载地址 4
showname	文本	255	否	软件的显示名称
bb	数字	长整型	否	文件的格式
note	备注	-	-	软件的说明
classid	数字	长整型	否	软件所属大类类别
nclassid	数字	长整型	否	软件所属小类类别
fromurl	文本	255	否	软件演示地址
xyyf	文本	255	否	软件语言版本

续表

字 段	类 型	字段大小	必填字段	说 明
orders	文本	255	否	软件发行类型
size	数字	长整型	否	软件大小
system	文本	255	否	软件支持系统
hot	数字	长整型	否	软件评价等级
dayhits	数字	长整型	否	今天下载数
weekhits	数字	长整型	否	本周下载数
lasthits	日期/时间	-	-	最后下载时间
hits	数字	长整型	否	总下载数
stop	数字	长整型	否	是否精品软件
hots	数字	长整型	否	是否推荐软件
dateandtime	日期/时间	-	-	软件上传时间

21.3 页面设计

本实例共有 16 个文件，各个文件的具体功能说明见表 21.4。

表 21.4 软件下载系统文件列表

文 件 名	说 明
class.asp	源码分类列表页面
conn.asp	数据库连接文件
footer.asp	页脚页面
index.asp	系统主页面，即首页页面
down.asp	下载分类列表页面
down_all.asp	下载总量排行页面
down_command.asp	推荐下载排行页面
down_today.asp	今日下载排行页面
down_week.asp	本周下载排行页面
download.asp	软件下载处理文件
recommend.asp	推荐分类下载列表页面
softclass.asp	软件分类列表页面
software.asp	软件详细下载页面
header.asp	系统标题和导航页面
update.asp	更新分类下载列表页面
down_class.asp	软件分类下载排行页面

本系统页面设计比较简单，具体设计就不再进行详细介绍，用户可参见光盘实例模板，下面重点介绍各个页面功能实现。

21.4 软件下载功能实现

本系统的各页面功能有很多相似之处，基本思路都是如何定义记录集、查询数据库中分类记录，然后把它们重复显示在页面中。因此，本节将重点介绍 SQL 字符串的编写技巧。

21.4.1 下载排行页面

在本系统中存在多种形式的下载排行，例如，今日下载排行、本周下载排行、下载总量排行、推荐排行等。这些功能模块的设计方法基本相同，不同点就是 SQL 查询语句的区别。为便于在不同页面中引用这些功能模块，分别设计成独立的文件，方便其他列表页面的引用。

下载排行页面的查询设计操作步骤如下。

- 01 打开 down_today.asp 文件，该页面是一个独立的表格，设计如图 21.3 所示。注意不要包含<html>、<head>和<body>其他 HTML 文件结构标记。
- 02 在“记录集”面板中定义一个 rs 记录集，用来查询今天下载排行的前几名，“记录集”对话框设置如图 21.4 所示。



图 21.3 down_today.asp 文件设计效果

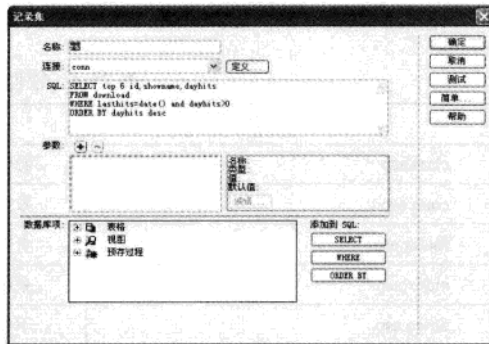


图 21.4 定义今天排行记录集

注意，该查询条件只能在【记录集】对话框高级状态下手动输入，查询条件是从 download 表中筛选出 lasthits 字段为今天，同时被下载的次数不能少于一次。具体 SQL 字符串如下：

```
SELECT top 6 id, showname, dayhits
FROM download
WHERE lasthits=date() and dayhits>0
ORDER BY dayhits desc
```

- 03 在【设计】视图中绑定 showname 字段。然后选中绑定字段所在的行（要准确选取，建议在【代码】视图下操作），在【服务器行为】面板中定义重复区域，设置如图 21.5 所示。
- 04 切换到【代码】视图中，删除如下两行代码，避免该文件被其他页面引用时与其他定义记录集的文件代码发生定义重复的冲突。

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="Connections/conn.asp" -->
```

同时，由于 Dreamweaver CS5 在定义记录集和重复区域时，在不同的文件中都使用同名

的变量, 这样就为定义有 Dreamweaver CS5 服务器行为或记录集的文件相互引用带来很多潜在的冲突。为避免这类问题的发生, 在需要使用这些方法时, 可以在【代码】视图中用查找和替换功能, 把其中容易冲突的变量替换为其他名称, 如 rs_numRows、Repeat_numRows、Repeat1_index 等。

05 以相同的方法定义其他页面的 SQL 查询语句, 相同的操作就不再重复。打开 down_week.asp 文件, 定义 rs 记录集的【记录集】对话框, 如图 21.6 所示。

其中 NewWeek 和 OldWeek 是两个自定义变量, 在【代码】视图中的记录集代码段前面输入下面代码来计算本周和上周的时间。

```
<%
OldWeek = WeekDay(Date())-1
If OldWeek = 0 Then OldWeek = 7
OldWeek = Date()-OldWeek
NewWeek = Date()+(9-WeekDay(Date()))
%>
```

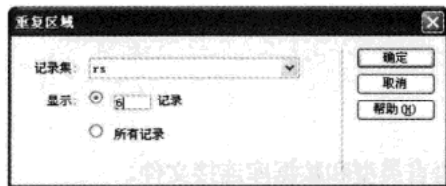


图 21.5 定义重复区域

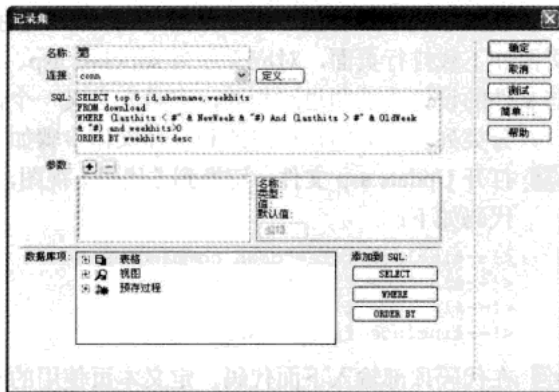


图 21.6 定义本周排行记录集

06 打开 down_command.asp 文件, 定义推荐下载排行 rs 记录集的【记录集】对话框, 如图 21.7 所示。

要注意 WHERE 查询语句中的跨表查询条件 download.Nclassid=Nclass.Nclassid。

07 打开 down_class.asp 文件, 定义软件分类下载排行 rs 记录集的【记录集】对话框, 如图 21.8 所示。

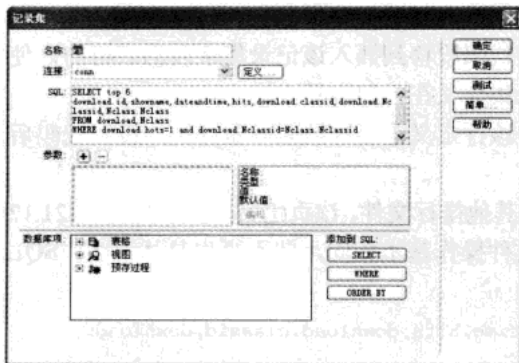
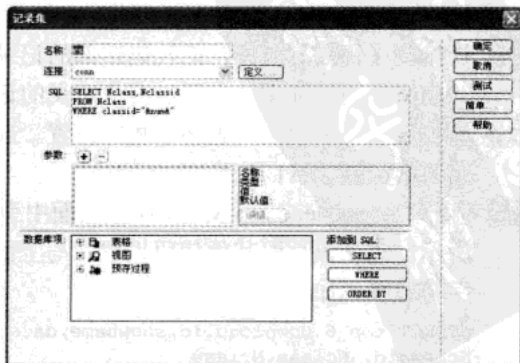


图 21.7 定义推荐下载排行记录集



要注意切换到【代码】视图，在记录集代码段前边输入下面代码，用来获取软件分类编号，并定义一个条件，当分类编号为空时，将不执行该查询。

```
<%
num=request("classid")
if request("classid")<>" then
%>
```

- 08** 打开 down_all.asp 文件，定义下载总排行 rs 记录集的【记录集】对话框，如图 21.9 所示。

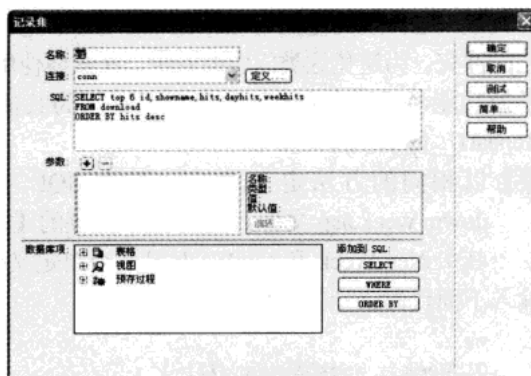


图 21.9 定义下载总排行记录集

21.4.2 分类列表页面

在本系统中共设置了 4 个分类列表页面，包括软件分类页面、最新更新页面、推荐排行页面和下载排行页面，对应文件为 softclass.asp、update.asp、recommend.asp 和 down.asp。这些文件的设计方法和思路相同，也是先定义一个记录集，然后插入页面中重复显示。

分类列表页面的设计思路和具体操作步骤如下。

- 01** 打开 Update.asp 文件，切换到【代码】视图，在页面右侧表格中引入各种下载排行文件，代码如下：

```
<!--#include file="down_command.asp"-->
<!--#include file="down_today.asp"-->
<!--#include file="down_week.asp"-->
<!--#include file="down_all.asp"-->
```

- 02** 在代码顶部输入下面代码，定义本页使用的脚本语言类型和数据库连接文件。

```
<%@LANGUAGE="VBSCRIPT"%>
<!--#include file="conn.asp" -->
```

- 03** 在【记录集】对话框中定义一个记录集 rs，用来查询最新软件更新。关于更新查询需要跨表进行，设置筛选条件为 download.Nclassid=Nclass.Nclassid，【记录集】对话框设置如图 21.10 所示。SQL 字符串代码如下：

```
SELECT top 6 download.id,showname,dateandtime,hits,download.classid,download.
Nclassid, Nclass.Nclass
FROM download,Nclass
WHERE download.hots=1 and download.Nclassid=Nclass.Nclassid
```

- 04** 切换到【代码】视图，把代码顶部的记录集代码段移到插入该记录集字段的表格前，使其位于引用外部文件之后，避免定义的记录集交叉在一起容易出错。

- 05** 将该记录集字段插入表格中相应位置。选中该行定义重复区域服务器行为，设置代码后的页面如图 21.11 所示。

- 06** 打开 Recommend.asp 文件，在该页面中引入其他排行文件，该页定义的记录集如图 21.12 所示。关于该页面其他代码设置，与上面文件操作基本相同，就不再作具体说明。SQL 字符串代码如下：

```
SELECT top 6 download.id,showname,dateandtime,hits,download.classid,download.
Nclassid, Nclass.Nclass
FROM download,Nclass
WHERE download.hots=1 and download.Nclassid=Nclass.Nclassid
```



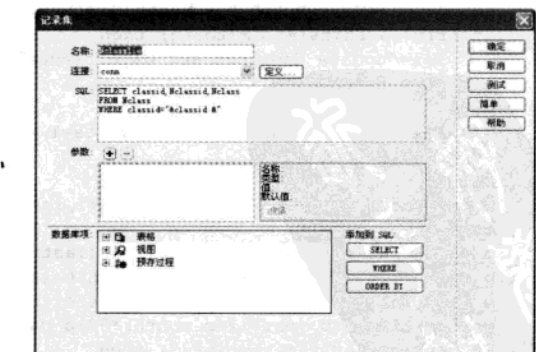
图 21.11 Update.asp 设置效果

07 打开 `SoftClass.asp` 文件，在该页面中引入其他排行文件。由于该页面有两级分类列表，即先根据大类排列，再根据各大类的编号，排列该大类下的所有小类，页面设计效果如图 21.13 所示。



图 21.13 软件分类列表页面设计效果

08 因此，先要定义两个记录集，分别查询软件分类中的大类和小类，设置如图 21.14 所示。



(b)

图 21.14 定义记录集

(a) 查询大类记录集; (b) 查询小类记录集

- 09** 把大类记录集 classlie 中的分类名称字段插入页面，并设置链接，代码如下：

```
<a href=class.asp?classid=<%=classid%>><%=classlie("class")%></a>
```

在插入小类的类别名称时，要注意先把该记录集 Nclasslie 的代码移到该表格前边大类重复区域代码内，否则会出错。

21.4.3 具体下载页面

打开 Software.asp 文件，在该页面内将要完成两件任务：一是要即时更新数据库，如下载次数、最后点击时间、下载总数等；二是要显示下载软件的详细信息，根据条件提供下载链接。实现该功能的具体操作步骤如下。

- 01** 打开 Software.asp 文件，定义一个记录集查询下载软件相关信息，【记录集】对话框设置如图 21.15 所示。

切换到【代码】视图，在该记录集代码段前加入下面一句代码，用来获取传递的下载软件的编号。

```
<%  
softid=request("id")  
%>
```

- 02** 在 rs 记录集代码段后面输入下面代码，用来判断并更新数据库中的下载信息。代码如下：

```
<%  
if not rs.eof then  
    showname=rs("showname")  
    bb=rs("bb")  
    classid=rs("classid")  
    Nclassid=rs("Nclassid")  
    classname=rs("class")  
    Nclassname=rs("Nclass")  
    lasthits=rs("lasthits")  
end if  
rs.close  
tdate=year(Now()) & "-" & month(Now()) & "-" & day(Now())  
if trim(lasthits)=trim(tdate) then  
    sql="update download set dayhits=dayhits+1 where id=" & request("id")  
    conn.Execute(sql)  
else  
    sql="update download set dayhits=1 where id=" & request("id")  
    conn.Execute(sql)  
end if  
sql="update download set hits=hits+1,lasthits=" & tdate & " where ID=" & request("id")  
conn.Execute(sql)  
p_year=CInt(year(Now()))-CInt(year(lasthits))  
p_month=CInt(month(Now()))-CInt(month(lasthits))  
p_day=CInt(day(Now()))-CInt(day(lasthits))  
period_time=((p_year*12+p_month)*30+p_day)  
if cint(period_time)=<cint(7) then  
    sql="update download set weekhits=weekhits+1 where id=" & request("id")  
    conn.Execute(sql)  
else
```

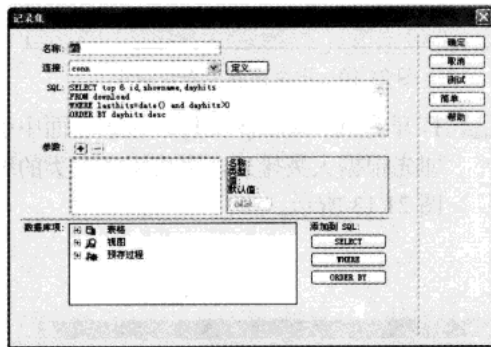


图 21.15 定义记录集

```

sql="update download set weekhits=1 where id=" &request("id")
conn.Execute(sql)
end if
%>

```

03 定义一个记录集，查询指定软件的全部信息。代码如下：

```

<%
sql="select * from download where id=" &request("id")
rs.open sql,conn,1,1
%>

```

04 把该记录集的相关字段信息插入各个单元格中，其中要注意各种条件设置，例如，根据是否提供软件演示地址，有条件显示该单元格的功能。主要代码如下：

```

<TD>软件演示: </TD><TD>
<%if rs("fromurl")<>"" then%>
<a href="<%=rs("fromurl")%>" target="_blank">浏览程序演示</a>
<%end if%>
</TD>

```

同样的条件设置还有：

```

<%if rs("filename")="" and rs("filename1")="" and rs("filename2")="" then%>
暂时没有下载
<%else%>
<a href="download.asp?downid=1&id=<%=rs("id")%>" target="_blank">下载</a></TD></TR>
<%end if%>

```

其他细节就不再赘述，读者可参考光盘实例。

21.5 根据字段进行排序

在本软件下载系统中，提供了软件下载列表动态排序功能，如图 21.16 所示，当单击不同的标题时会按该标题所在字段进行降序排列列表。



(a)

(b)

图 21.16 动态排序列表浏览效果

(a) 按下载次数排序；(b) 按下载软件大小排序

动态列表排序在网站中经常见到，其技术核心只是设置不同的 SQL 查询字符串，即设置 order 关键词为一个动态变量，然后根据访问者单击的标题来动态排列列表。

打开 Class.asp 文件，其中的动态列表排序功能的核心代码如下：

```

<%
dim totalPut,CurrentPage,TotalPages,i,j,keyword,sql,rs,updown,order_name
order_name=Request("Order")
if not isempty(request("page")) then
    currentPage=cint(request("page"))
else
    currentPage=1
end if
if request("updown")<>" " then
    updown="desc"
else
    updown=""
end if
select case order_name
    case "showname"
        order_name="showname"
    case "hot"
        order_name="hot"
    case "dateandtime"
        order_name="dateandtime"
    case "hits"
        order_name="hits"
    case "orders"
        order_name="orders"
    case "size"
        order_name="[size]"
    case else
        order_name="dateandtime"
        updown="desc"
end select
set rs=server.createobject("adodb.recordset")
dim classid,Nclassid
dim classname,Nclassname
if request("classid")="" then
    classid=""
    classname="全部分类软件"
    classid="classid=1 and "
    sql="select class from class where classid=1"
    rs.open sql,conn,1,1
    if rs.bof and rs.eof then
        response.write ""
        response.end
    else
        classname=rs("class")
    end if
    rs.close
else
    classid="classid=" & cstr(request("classid")) & " and "
    sql="select class from class where classid=" & cstr(request("classid"))
    rs.open sql,conn,1,1
    classname=rs("class")
    rs.close
end if
if request("Nclassid")="" then
    Nclassid=""
    Nclassname="全部分类软件"
else
    Nclassid=" Nclassid=" & cstr(request("Nclassid")) & " and "
    sql="select Nclass.Nclass,class.class from Nclass,class where Nclass.classid=
class.classid and Nclass.Nclassid=" & cstr(request("Nclassid"))
    rs.open sql,conn,1,1
    classname=rs("class")

```

```

Nclassname=rs("Nclass")
rs.close
end if
%>

```

上面代码就是根据不同的传递参数设置 `order` 关键词的变量值来实现动态排序功能的。

21.6 实践总结

本章重点介绍了软件下载系统的基本设计思路和方法，并介绍了与下载相关的一些技术实现。制作软件下载系统时，要注意设置好数据库中下载文件的详细链接地址。同时，下载文件最好以压缩的形式提供，避免被浏览器打开或解析。

当然读者也可以进一步开发出在线视频和在线音频浏览功能，它们实现的方法是相同的，例如，加上下面一行代码可以以下载的形式在线播放。

```

<A href="#" onClick="windowOpen('<%=play%>?id=<%=rs("id")%>&amp;downid=1')"> <%=rs
("name")%></A>

```

也可以插入视频插件，然后把记录集绑定到播放插件的文件属性值上。代码如下：

```

<object id="mplayer" width="400" height="80" classid="CLSID:22d6f312-b0f6-11d0-94ab-
0080c74c7e95"
align="baseline" border="0" standby="Loading Microsoft Windows Media Player
components..."
type="application/x-oleobject">
<param name="FileName" value="<%=Rs2.Fields.Item("MusicUrl").Value%>">
...
</object>

```

第 22 章

Dreamweaver

电子相册

电子相册就是可以在 Web 上欣赏的区别于 CD/VCD 的静止图片的应用系统，其内容不局限于摄影照片，也可以包括各种艺术创作图片。电子相册具有传统相册无法比拟的优越性：图、文、声、像并茂的表现手法，随意修改编辑的功能，快速的检索方式，永不褪色的恒久保存特性，以及廉价复制分发的优越手段。

本例将详细介绍一个简单的电子相册的制作过程，主要包括图片的上传和管理。图片上传是 ASP 开发中一项复杂的技术，需要用到外部组件，然后通过调用外部组件并进行代码编写来实现。

技术要点

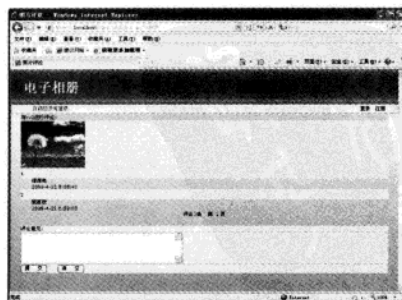
- 自定义组件
- 图片上传技术
- 下载并使用无组件上传程序包
- 嵌套记录集
- 文件系统组件运用
- 复杂条件页面设计

22.1 程序设计分析

电子相册是比较时尚的 Web 应用程序，其核心功能就是利用网页来管理图片。本例介绍了一个多用户环境下的网页图片管理系统，它可以实现图片上传、网上管理、网上浏览和交流的功能（见图 22.1），程序设计流程如图 22.2 所示。



(a)



(b)

图 22.1 电子相册浏览效果
(a) 浏览图片；(b) 评论图片

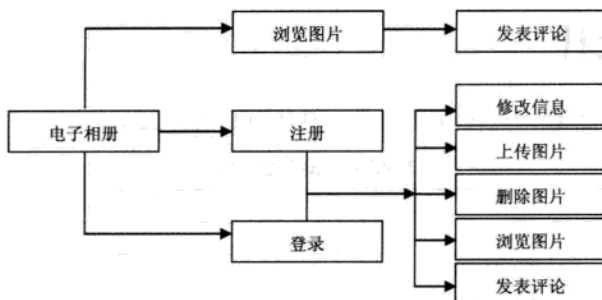


图 22.2 电子相册设计流程图

22.2 数据库设计

本实例数据库 db.mdb 包括 admin、zp 和 pl 三个表。admin 表用来存储注册用户信息，数据结构设计见表 22.1。zp 表用来存储上传图片信息，数据结构设计见表 22.2。pl 表用来存储对上传图片评论的信息，数据结构设计见表 22.3。

表 22.1 admin 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
yhm	文本	30	否	用户名
pass	文本	30	否	用户密码
xb	文本	4	否	性别
lx	文本	8	否	浏览范围
jb	文本	8	否	权限
rq	日期/时间	-	-	注册时间

表 22.2 zp 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
zp	文本	50	否	图片名称和存储地址
rq	日期/时间	-	-	上传时间
wjdx	数字	长整型	-	图片文件大小
lx	文本	6	否	浏览范围
zt	文本	50	否	图片显示名称
yhm	文本	30	否	上传用户名
pls	数字	长整型	-	评论次数

表 22.3 pl 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	自动编号
yhm	文本	30	否	用户名，设置【默认值】为“无名氏”
pl	备注	-	-	相关图片评论
zp	文本	10	否	图片编号
rq	日期/时间	-	-	评论时间

22.3 页面设计

本电子相册应用系统共包含 10 个文件，具体说明见表 22.4。

表 22.4 系统文件列表

文件名	说 明
index.asp	图片浏览主页面
add_pic.asp	增加图片页面
del_pic.asp	数据库连接与自定义函数
search_result.asp	删除图片文件
head.asp	头部结构页面
pic_dl.asp	用户登录页面
pic_zc.asp	用户注册页面
pl_pic.asp	图片评论页面
scfile.asp	上传图片文件到服务器端文件夹，并把相应信息增加到数据库
upload_class.inc	自定义上传图片类

下面就几个主要页面设计进行简单介绍。

22.3.1 上传表单页面

add_pic.asp 页面主要由一个表单构成，用来选择要传递的图片，以及图片名称和是否公开，页面设计效果如图 22.3 所示。

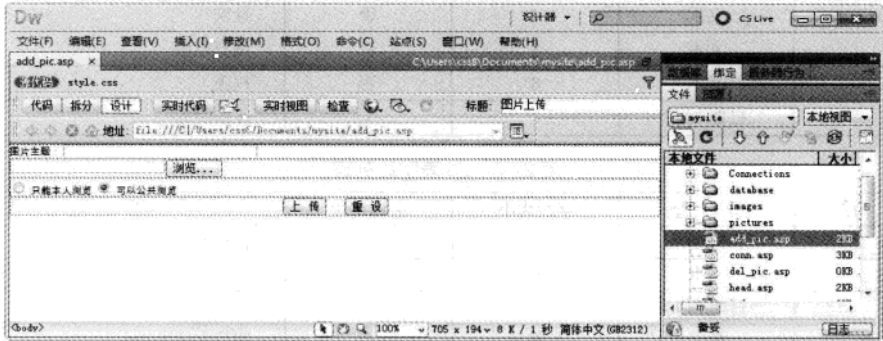


图 22.3 add_pic.asp 页面设计效果

在该页面中，其中表单属性设置见表 22.5。

表 22.5 上传表单属性设置列表

对 象	名 称	类 型	设置说明
表单	dd	表单	【动作】设置为“scfile.asp”；【方法】设置为“POST”，【MIME 类型】为“multipart/form-data”
文本框	zt	单行	【字符宽度】设置为 30，【最多字符数】为 100
文件域	file1	文件域	-
选择框	qx	单选按钮	【选定值】为 1、0；当【选定值】为 1 时，设置【初始状态】为“已勾选”
按钮	Submit	提交	【值】为“上传”，【动作】为“提交表单”
按钮	Submit2	重设	【值】为“重设”，【动作】为“重设表单”

22.3.2 头部页面

head.asp 页面是一个简单的表格, 包含系统标题栏和导航信息栏, 作为其他文件的嵌入文件, 起方便管理的作用, 页面设计效果如图 22.4 所示。

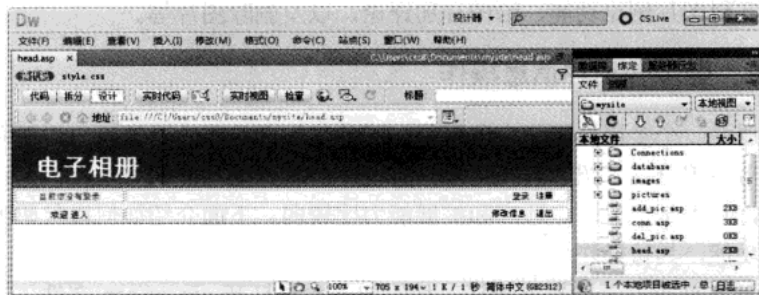
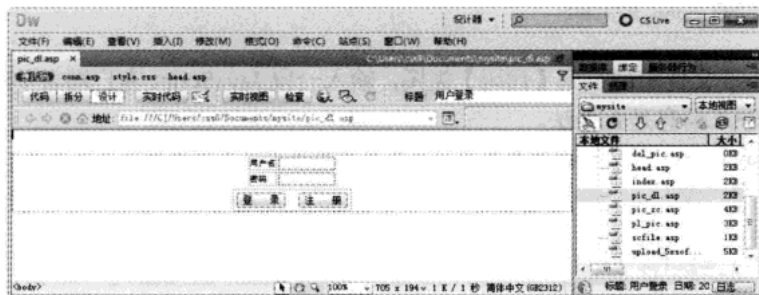


图 22.4 头部页面设计效果

该页面嵌入一个简单的条件语句, 根据用户是否登录显示不同的导航栏信息。

22.3.3 其他表单页面

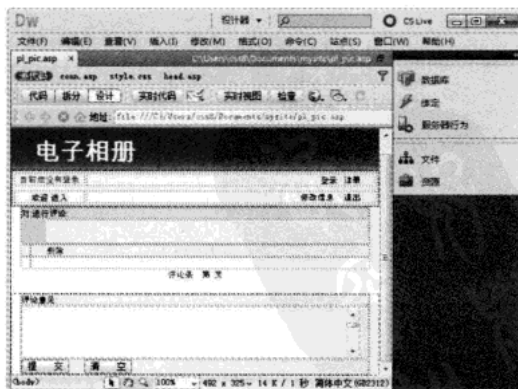
有关登录、注册和发表评论页面都是简单的表单页面, 设计效果如图 22.5 所示。



(a)



(b)



(c)

图 22.5 其他表单页面设计效果

(a) pic_dl.asp 登录表单页面; (b) pic_zc.asp 注册表单页面; (c) pl_pic.asp 评论表单页面

其中 pic_zc.asp 注册表单页面由两个表单组成, 一个是注册表单, 另一个是修改用户信息表单, 根据不同的条件实现显示不同的页面功能。

22.4 电子相册浏览与管理

下面详细介绍本系统基本功能的实现过程和方法,这主要包括用户登录和网上图片管理操作,例如,实现图片浏览功能、给图片加评论,以及删除图片等。

22.4.1 注册、登录和用户信息操作

本电子相册是一个多用户操作应用系统,因此工作之前要先设计用户注册和登录功能模块,在前面章节中曾分别介绍了可视化操作实现注册和登录功能的方法;但是可视化实现该功能代码臃肿,且不是很灵活,适应和扩展能力比较弱。下面介绍如何以代码方法实现高效的登录和注册功能,具体操作步骤如下。

01 在 conn.asp 文件中打开数据库连接,代码如下。再打开 pic_zc.asp 文件,导入数据库连接文件 conn.asp。

```
<!--#include file="Connections/conn.asp"-->
<% 如果出错,则跳出下一步
on error resume next
'定义一个数据库连接对象的具体实例
set conn=server.createobject("adodb.connection")
'打开该数据库连接
conn.open MM_conn_STRING
%>
```

02 在 pic_zc.asp 文件中切换到【代码】视图,输入一段 JavaScript 客户端表单验证代码,以验证用户输入信息是否合法。

```
<script language='javascript'>
function xgjc(mm)
{
if (mm=="")
{alert('密码不能为空!');
document.de.mm.focus();
return false;}
if (mm.indexOf("'")>=0 || mm.indexOf("<")>=0 || mm.indexOf(">")>=0 || mm.indexOf(".")>=0)
{
alert('不能有乱码!');
document.de.mm.focus();
return false;
}}

function zcjc(yhm,mm)
{
if (yhm=="")
{alert('用户名不能为空!');
document.zc.yhm.focus();
return false;}
var aa = "";
aa = yhm.substring(0,1);
if ((aa<='z'&&aa>='a') || (aa<='Z'&&aa>='A'))
{}
else
{alert('首位不是字母!');
document.zc.yhm.focus();
return false;}
if (yhm.indexOf("'")>=0 || yhm.indexOf("<")>=0 || yhm.indexOf(">")>=0 || yhm.indexOf(".")>=0)
{alert('用户名不能有乱码!');
}
```

```

document.zc.yhm.focus();
return false;}

if(mm=="")
{alert('密码不能为空!');
document.zc.mm.focus();
return false;}
if(mm.indexOf("'")>=0 || mm.indexOf("<")>=0 || mm.indexOf(">")>=0 || mm.indexOf(".")>=0)
{alert('不能有乱码!');
document.zc.mm.focus();
return false;}
}

```

- 03** 利用 Request.QueryString 数据集合获取查询字符串传递的参数值, 用 Request.Form 数据集合获取提交表单传递的登录。代码如下:

```

<% 获取查询字符串和表单信息
cz=request.querystring("cz")
mm=request.form("mm")
yhm=request.form("yhm")
qx=request.form("qx")
xb=request.form("xb")
%>

```

- 04** 输入下面代码实现对用户注册信息进行修改, 并把它存储到 Session 变量中。

```

<% 判断, 如果是修改表单传递过来的值, 同时密码和用户名不为空, 修改用户名和密码
if cz="xgxg" and mm<>" and yhm<>" then
    exec="update admin set pass='"+setpwd(mm)+"',lx='"+qx+"',xb='"+xb+"' where yhm='
"+yhm+"'"
    conn.execute exec '更新数据表
    conn.close
    '存储用户名、密码、权限和性别到session变量中
    session("picpass") = mm
    session("picqx") = qx
    session("picqx")= xb
    '提示修改成功
    Response.Write "<script>alert('恭喜,修改成功!');window.location='index.asp';</script>"
end if
%>

```

- 05** 输入下面代码段, 实现用户注册功能, 并把注册信息存储到 Session 变量中。

```

<% 判断, 从注册表单传递过来的参数, 同时密码和用户名不为空
if cz="zccg" and mm<>" and yhm<>" then
    '判断是否重复
    set rs=server.createobject("adodb.recordset") '定义一个记录集, 查询条件为表单中输入的用户名
    exec="select * from admin where yhm='"+yhm+"'"
    rs.open exec,conn,1,1
    if rs.eof then '判断, 如果没有记录, 则把注册表单中的信息插入数据库中
        yhm=xrz(h(yhm))
        exec="insert into admin (xb,yhm,pass,jb,rq,lx) values ('"+xb+"','"+yhm+"',
 '"+setpwd (mm)+"','1','"+cstr(date())+"','"+qx+"')
        conn.execute exec
        conn.close
        '存储用户名、密码、权限和性别到session变量中
        session("picyhm") = yhm
        session("picxb") = xb
        session("picjb") = "1"
        session("picpass") = mm
        session("picqx") = qx
        Response.Write "<script>alert('恭喜,申请成功!');window.location='index.asp';
</script>"
    else '否则, 提示错误信息

```

```

        Response.Write "<script>alert('抱歉,此用户名已经有人用了,请换名后重新申请!');";
        window.location='pic_zc.asp?cz=zc';</script>"
    end if
end if
%>

```



提示

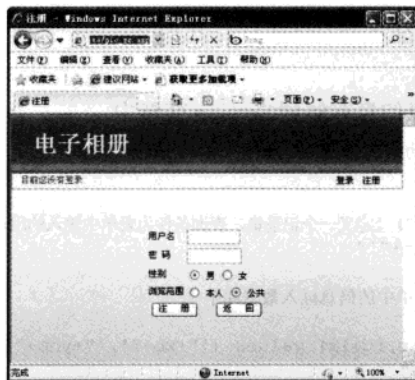
xrzh()是一个自定义函数,功能是查找并替换字符的非法引号,Setpwd()是一个自定义函数,用来实现对密码进行加密。这些函数都在 conn.asp 文件中定义,代码如下:

```

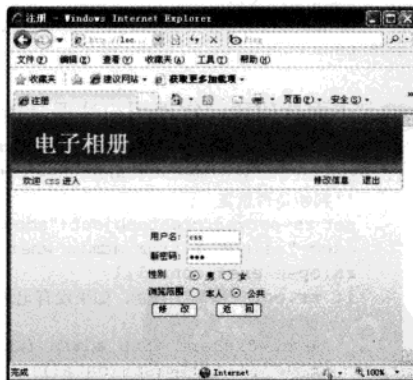
Function Xrzh(zh)
if zh<>"" then
    zh=replace(zh,"'", "&euro;")
    Xrzh = zh
end if
End Function
Function SetPwd(sPwd)
    sSql = ""
    For i = 1 To Len(sPwd)
        If bType = True Then
            sSql = sSql & Chr(-Asc(Mid(sPwd, i, 1)) - 10000)
        Else
            sSql = sSql & Chr(-Asc(Mid(sPwd, i, 1)) - 10000)
        End If
    Next
    SetPwd = sSql
End Function

```

- 06** 在两个表单前加入一个条件,实现根据不同的动作显示注册表单或者修改表单。在注册表单 zc 前输入代码<%if cz="" then%>,即当传递的参数变量 cz 为空时显示注册表单,允许用户进行注册。在修改表单 de 前输入代码<%else%>,即当传递的参数变量 cz 不为 xgxg、zccg 或者为空时,显示修改表单,允许修改用户信息。然后在代码的底部输入代码<%end if%>,结束页面条件结构。该页面运行效果如图 22.6 所示。



(a)



(b)

图 22.6 注册和修改页面运行效果

(a) 注册用户; (b) 修改用户

- 07** 登录页面主要功能是对登录用户输入信息进行检查,如果登录正确则存储该用户信息到 Session 变量中。打开 pic_dl.asp 文件,切换到【代码】视图,在代码顶部输入下面代码:

```

<!--#include file="conn.asp"-->
<% '获取表单信息
yhm=request.form("yhm")

```

则重新登录

[illegible]

``

22

只能浏览和评论图片。

该页面中主要技术难点是记录集的嵌套，例如，定义一个记录集查询并显示注册用户信息，然后又根据每个用户名定义一个记录集用来查询该用户上传的所有图片数；根据用户名定义一个记录集，用来显示该用户上传的图片；并根据每个图片的编号，定义一个记录集获取该图片的所有评论。

实现上述功能的具体操作步骤如下。

- 01** 在 Dreamweaver CS5 中，打开 conn.asp 文件，定义一个数据库连接。

```
<!--#include file="Connections/conn.asp"-->
<% 如果出错，则跳出下一步
on error resume next
'定义一个数据库连接对象的具体实例
set conn=server.createobject("adodb.connection")
'打开该数据库连接
conn.open MM_conn_STRING
%>
```

- 02** 打开 index.asp 文件，切换到【代码】视图，在第 1 行导入 conn.asp 文件。

```
<!--#include file="conn.asp"-->
```

- 03** 设计根据本页单击“删除”超链接文件而传递过来的用户名或评论用户名，删除表 admin 或表 pl 中的相应记录。

```
<% 删除指定用户信息和评论内容
zt=request.querystring("zt")
id = request.QueryString("id")
if id <> "" then
    exec = "delete from admin where yhm = '" + id + "'"
    conn.execute exec
    exec = "delete from pl where yhm = '" + id + "'"
    conn.execute exec
    conn.close
    response.Redirect("index.asp")
end if
%>
```

- 04** 定义一个记录集查询所有注册用户记录。按每页 8 条记录的设置循环显示所有注册用户信息。然后在该记录集内嵌套一个记录集，如下面方框代码所示，根据用户名查询表 zp 中该用户上传的图片数。

```
<table border="0" cellpadding="1" cellspacing="1" width="100%" background="images/
bg.gif" align="center">
<% 定义记录集获取注册用户信息
set rs=server.createobject("adodb.recordset")
exec="select * from admin order by id desc"
rs.open exec,conn,1,1
'获取传递的页码参数
page1=int(Request.QueryString("page1"))
'指定每页显示的记录数
rs.PageSize =8
'判断页码变量值超出指定范围的处理
If page1 < 1 Then page1 = 1
If page1 > rs.PageCount Then page1 = rs.PageCount
'指定当前要显示的页数
if not rs.eof then rs.AbsolutePage =page1
'循环显示注册用户记录
for i = 1 To rs.PageSize
'判断如果记录集没有则跳出循环
```

```

If rs.EOF Then Exit For
'获取记录集中的用户名
yhm = rs("yhm")
if rs.eof then exit for
'新定义一个记录集, 查询符合该用户名的上传图片信息
set rsl=server.createobject("adodb.recordset")
exec="select * from zp where yhm='" + yhm + "'"
rsl.open exec,conn,1,1
'获取记录数
gs = rsl.recordcount
'获取用户权限
qx=session("picqx")%>
<tr bgcolor="#E3E9F1" height="25"><td align="center" valign="top">
<a href=<%
'判断, 如果是公开、管理员或者自己登录, 则可以显示该用户名的所有图片
if rs("lx")="1" or session("picjb")="admin" or session("picyhm")=rs("yhm") then
    response.write "index.asp?zt="+rs("yhm")
else
'否则, 将显示空链接
    response.write "<%"
end if %>>
<%=rs("yhm")%></a></td><td align="center"><%=rs("xb")%></td>
<td align="center"><font color="#FF0000"><%=gs%></font>张照片
<%'判断, 如果不是admin管理员, 同时没有上传图片, 而且所操作的记录不是admin, 则显示删除链接
if session("picjb")="admin" and gs = 0 and rs("jb")<>"admin" then%>
<a href="index.asp?id=<%=yhm%>" onClick="javascript:confirm('是否删除!')">删除</a>
<%end if%>
</td></tr>
<%rs.movenext
next%>
<tr><td colspan="3" align="center"><a href="index.asp?zt=qb" class="white">查看
全部照片</a></td>
</tr>
</table>

```

05 在上述代码后接着输入下面代码, 实现注册用户记录分页显示。用循环结构分别显示所有记录集页码, 如果不是当前页则为页码加一对中括号并链接本页, 同时传递页码参数, 实现在不同页码之间跳转。效果如图 22.7 所示。

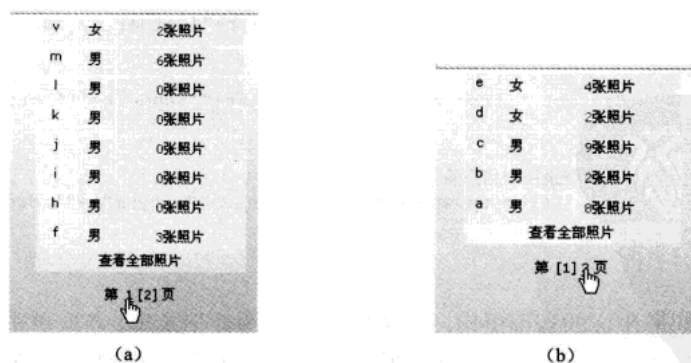


图 22.7 用户列表栏
(a) 跳转到第1页; (b) 跳转到第2页

```

<br>
<center>
第
<%'循环显示页数链接, 如果是当前页则不显示中括号标记
for j=1 to rs.PageCount%>
<a href="index.asp?page1=<%=j%>">

```

```

<%if j=page1 then
    response.write j
else
    response.write "["+cstr(j)+"]"
end if%></a>
<%next
if rs.recordcount=0 then response.Write("0")
%>
页
</center>

```

- 06** 完成用户记录的定义与显示后，接下来设计图片查询与显示，具体代码如下。在下面代码中涉及对多种条件的复杂判断，并根据不同的条件定义不同的记录集查询条件。

```

<% '定义记录集，获取上传图片信息
set rs=server.createobject("adodb.recordset")
'如果退出，则清空登录用户变量
cz=request.querystring("cz")
if cz="exit" then
    session("picyhm")=""
    session("picjb")=""
    session("picpass")=""
    response.redirect "index.asp"
end if
'判断主题是否为空，为空则显示所有图片
if zt="" then zt="qb"
'判断是否显示全部图片
if zt="qb" then
'判断如果是管理员，则显示所有图片
    if session("picjb")="admin" then
        exec="select * from zp order by id desc"
'否则只显示被公开浏览的图片或者登录用户自己上传图片
    else
        exec="select * from zp where lx='1' or yhm='"+session("picyhm")+"' order by id desc"
    end if
'当主题不为空时
else
    if zt<>"" then
'判断如果是管理员或者登录用户，则查询该主题下的所有图片
        if session("picjb")="admin" or session("picyhm")=zt then
            exec="select * from zp where yhm='"+zt+"' order by id desc"
            '否则，就只能显示主题下的所有公开图片
        else
            exec="select * from zp where yhm='"+zt+"' and lx='1' order by id desc"
        end if
    else
'如果主题为空，则按登录用户信息查询
        exec="select * from zp where yhm='"+session("picyhm")+"' order by id desc"
    end if
end if
%>

```

- 07** 利用上面代码段生成的查询语句，获取表 **zp** 中的图片记录集。获取单击页码时传递过来的页码数，接着设定每页显示的图片数，然后把传递的页码参数指定为当前页码显示，实现在各页之间切换，最后利用循环语句显示定义记录集中所有图片。

```

<%
rs.open exec,conn,1,1
'获取页码数
page =int(Request.QueryString("page"))
'指定每页显示页数
rs.PageSize =16

```



```

</center>
<%rs.close
conn.close
set conn=nothing%>
</td></tr></table>
<table width="100%" border="0" cellspacing="1" cellpadding="0" align="center">
  <tr bgcolor="#95ADD2" >
    <td height="25" align="center">&nbsp;&nbsp;&nbsp;</td>
  </tr>
</table>
</body>
</html>

```

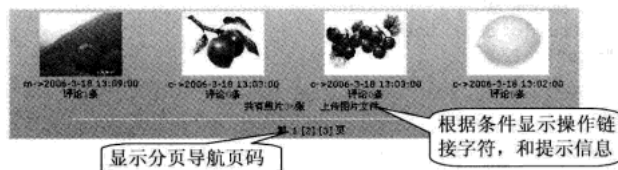


图 22.8 图片分页显示效果

22.4.3 图片评论

图片评论页面 (pl_pic.asp) 具有显示和发表评论两项主要功能。利用 index.asp 文件传递过来的 id 参数, 分别执行对 zp 和 pl 数据表的读写操作, 页面设计效果如图 22.9 所示。

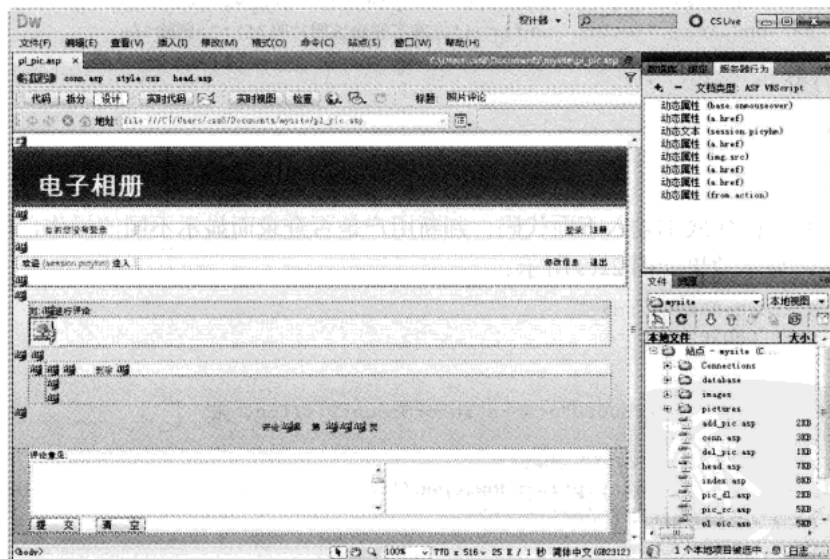


图 22.9 评论页面设计效果

详细代码如下:

```

' 导入数据库连接文件
<!--#include file="conn.asp"-->
<html><head></head>
<body>
<!--#include file="head.asp"-->
<% ' 获取字符串传递变量值, 以及表单输入值
id=request.querystring("id")

```

22

[illegible]

22.4.4 删除服务器上的图片文件

del_pic.asp 文件利用 ASP 内置组件 File Access 的 FileSystemObject 对象实现对上传图片进行删除。

FileSystemObject 对象使服务器端的应用程序能创建、改变、移动和删除文件夹，或探测文件或文件夹是否存在，找出有关文件夹的信息，例如名称、被创建或最后一次修改的日期等。FileSystemObject 对象包含在 Scripting 类型库中，该库由 Sscrn.dll 文件提供，用户可以在 C:\windows\system 目录中找到它。

创建 FileSystemObject 对象的语法如下:

```
Set fs=Server.Createobject("Scripting.FileSystemObject")
```

FileSystemObject 对象共有 24 种方法和一种属性，见表 22.6。语法如下：

```
FileSystemObject.{Property|method}
```

其中，Property 表示属性，method 表示方法，两个参数只能选择其中的一个。

表 22.6 FileSystemObject 对象的方法

语 法	说 明
Object. BuildPath(path,name)	在由 path 指定的目录后面加上由 name 指定的文件或文件夹
Object.CopyFile Source, Dest[, [,overwrite]]	将一个或多个文件从某位置复制到另一位置, 在源文件 Source 可以使用通配符。参数 overwrite 表示当目标文件存在时是否覆盖, 有两个取值: True 或 False
Object.CopyFolder Source, Dest [,overwrite]	将一个文件夹从某位置复制到另一位置。参数 overwrite 表示当日标文件存在时是否覆盖, 有两个取值: True 或 False
Object.CreateFolder(foldername)	创建一个名为 foldername 的文件夹。如果以该名字命名的文件夹已经存在, 则会产生一个错误
Object.CreateTextFile filename [,overwrite[,Unicode]]	创建一个名为 filename 的文件并返回 TextStream 对象, 参数 overwrite 是可选的, 表明当已存在一个同名文件时, 是否将其覆盖, 有两个取值: True 或 False。参数 Unicode 是可选的, 默认为 False, 指示该文件是否用 ASCII 码形式进行创建和保存。如果选择 True, 将以 Unicode 方式创建
Object.DeleteFile filespec[,force]	删除一个指定的文件, 参数 force 设置只读文件是否可被删除, 有两个取值: True 或 False
Object.DeleteFolder folderspec [,force]	删除一个指定的文件夹和其中的内容, 参数 force 的含意同上
Object.DriveExists(drivespec)	如果指定的驱动器存在, 则返回 True, 否则返回 False
Object.FileExists(filespec)	如果指定的文件存在, 则返回 True, 否则返回 False
Object.FolderExists(filespec)	如果指定的文件夹存在, 返回 True, 否则返回 False
Object.GetAbsolutePathName (pathspec)	从 pathspec 指定的路径中返回相对应的绝对路径名
Object.GetBaseName(path)	返回一个字符串, 该字符串包含路径最后一个组成部分的名字
Object.GetDrivespec	返回与指定路径中驱动器相对应的 Drive 对象
Object.GetDriveName(path)	返回一个包含指定路径中驱动器名的字符串
Object.GetExtensionName(path)	返回一个字符串, 该字符串包含路径最后一个组成部分的扩展名
Object.GetFile(filespec)	返回一个与指定路径中某文件相应的 File 对象
Object.GetFileName(pathspec)	返回指定路径 (不是指定驱动器路径部分) 的最后一个组成部分
Object.GetFolder(folderspec)	返回一个与指定路径中某文件夹相应的 Folder 对象
Object.GetParentFolderName (path)	返回一个字符串, 该字符串包含指定路径中最后一个组成部分的父文件夹
Object.GetSpecialFolder (folderspec)	返回一个指定的特殊文件夹
Object.GetTempName	返回随机生成的临时文件或文件夹的名称, 用于执行要求临时文件或文件夹的操作
Object.MoveFilesource, Dest	将一个或多个文件从某位置移动到另一位置, 在源文件 Source 中可以使用通配符
Object.MoveFolder source, Dest	将一个或多个文件夹从某位置移动到另一位置, 在源文件 Source 中可以使用通配符
Object.OpenTextFile (filename[,iomode[,create[,folrmat]]])	打开指定的文件并返回一个 TextStream 对象, 可以读取此对象或将其追加到文件。参数 iomode 的值为 1 时表示文件以只读方式打开, iomode 的值为 8 时表示文件以追加方式打开; 参数 create 为 True 时, 文件以创建方式打开; 参数 format 指定文件的格式: -2, -1, 0 分别对应系统默认、unicode 和 ASCII

在 del_pic.asp 文件中输入下面代码。

```
<!-- #include file="conn.asp" -->
<% '定义记录集, 删除表zp中的指定图片记录
str_id=request("sid")
strPictrue=Request("Content")
page=request.querystring("page")
strSQL=" Delete * FROM zp where id='"& str_id
Conn.Execute(strSQL)
'定义一个FileSystemObject对象, 用来删除指定的图片
```

```

Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
if instr(strpictrue,"http://")=0 and instr(strpictrue,":\")=0 then
    strPictrue=Server.mappath(strpictrue)
    if fso.fileExists(strPictrue) then fso.DeleteFile strPictrue
end if
'跳转到index.asp文件
Response.Redirect("index.asp?page="+page)
%>

```

22.5 图片上传

许多基于 Web 的应用都涉及文件上传操作。常见的文件上传技术有：基于 HTTP 协议的、基于 VB（或 DELPHI 等编程语言）开发的文件上传组件的、基于数据库技术的等。这些方法一般都需要编程者能同时掌握 Web 技术、数据库技术或 CGI 技术或组件技术，对编程者的要求较高。

- 基于 HTTP 协议。该方法需要编程者利用第三方软件，如 DELPHI、VB 等，在应用程序中先进行 HTTP 协议编程，然后将待上传文件内容按 HTTP 协议的格式打包，最后向 Web 服务器发送上传的请求报文，从而实现文件的上传。因为 DELPHI 和 VB 不能编写完整的 Web 网络程序，只能编写 Web 小应用程序，因此，该方法只用于功能受限的网络应用。
- 基于 VB（或 DELPHI 等）开发的文件上传组件。该方法利用 VB（或 DELPHI 等编程语言）开发 ASP 服务器组件，实现特定的文件上传服务。它首先利用 ASP 表单功能将文件（二进制格式）从用户端上传到服务器端，然后使用 VB 开发的组件，对二进制文件进行处理，成为可以正常读写的文件。该方法要求编程者不仅掌握 ASP 语言，而且还能利用 VB 等第三方语言进行组件编程，增加了开发的难度。
- 基于数据库技术。该方法和上个方法有类似之处，不同的地方在于对上传的二进制文件的处理上，它使用数据库来保存二进制文件。无论是小型数据库还是大型数据库都提供了存储二进制数据的数据类型，只要以 Append Chunk 方式将数据存入相应的字段就可以了。该方法虽然简单可行，但是因为每次上传的文件大小都是不一样的，因此会对数据库的空间造成很大的浪费，降低了数据的访问速度；并且使得文件只能在数据库环境下进行访问，造成了很大的不便。

下面介绍利用 ASP 技术直接实现文件上传功能。

22.5.1 无组件上传程序包

本节将介绍利用无组件上传程序包来实现图片文件的上传，这将使程序变得更加简单，此程序包可以到网站下载。

实际上该程序包就是一个 ASP 文件 upload_5xsoft.inc，该文件定义了 2 个自定义类。用户可以自己模仿建立自己的类，详细代码可参见本书光盘实例。



扩展名为 inc 表示其是包含 (include) 文件。upload_5xsoft.inc 文件可以免费使用，但是没有作者的许可不得用于商业目的。

upload_5xsoft.inc 中包含 2 个类，即 upload_5xsoft 和 FileInfo。upload_5xsoft 类的主要功

能是从客户端上传文件到服务器指定的目录中，属性见表 22.7。

表 22.7 upload_5xsoft 类的属性列表

属 性	说 明
objForm	表单的 Dictionary 对象
objFile	文件的 Dictionary 对象
Version	类的版本号

upload_5xsoft 类的方法见表 22.8。

表 22.8 upload_5xsoft 类的方法列表

方 法	说 明
Form(strForm)	取得表单名
File(strFile)	取得文件名
Class_Initialize	类初始化
Class_Terminate	释放类资源并终止类
GetFilePath(FullPath)	取得文件路径，参数 FullPath 为文件路径
GetFileName(FullPath)	取得文件名称，参数 FullPath 为文件路径

在类 upload_5xsoft 初始化时，程序将从上传表单中读取文件信息，这是类 upload_5xsoft 中最重要的代码，定义为 Class_Initialize 方法。因为代码中涉及 Steam 和 Dictionary 对象的使用，请读者参照相关书籍介绍理解。当然，也可以不考虑 Class_Initialize 的实现方法，直接在上传文件时使用下面的代码，即可自动执行 Class_Initialize 方法将上传文件信息读取到类 upload_5xsoft 中。

```
Set upload=new upload_5xsoft
```

类 FileInfo 的主要功能是取得文件信息，它的属性见表 22.9。

表 22.9 FileInfo 类的属性列表

属 性	说 明
FormName	上传的表单名
FileName	上传的文件名
FilePath	上传文件路径
FileSize	上传文件大小
FileType	上传文件类型
FileStart	上传文件的开始位置

FileInfo 类的方法见表 22.10。

表 22.10 FileInfo 类的方法列表

方 法	说 明
Class_Initialize	类初始化
SaveAs(FullPath)	保存上传的文件。参数 FullPath 为文件路径

下面小节将介绍 upload_5xsoft.inc 组件的具体使用，会对读者理解类 upload_5xsoft 和类 FileInfo 有很大的帮助，如果读者不解这两个类的实现方法，也可以直接使用它们上传文件，不需要阅读。

22.5.2 上传图片

利用 upload_5xsoft.inc 组件实现上传图片的方法如下。

- 01** 打开 scfile.asp 文件，输入下面代码，设置服务器端脚本执行的时间，并导入数据库连接文件和上传组件文件。

```
<%Server.ScriptTimeout=5000%>
<!--#include file="conn.asp"-->
<!--#include file="upload_5xsoft.inc"-->
```

- 02** 输入下面代码，利用 upload_5xsoft.inc 组件实现图片上传。

```
<body>
<%dim upload,file,formName,formPath,iCount,exec,wjm
set upload=new upload_5xsoft
formPath="pictures/"
wjm=upload.form("file")
wjdx=0
for each formName in upload.objFile
    if upload.form("zclj")="" then
        set file=upload.file(formName)
        wjdx = file.filesize
        kzm = right(file.filename,4)
    else
        kzm = right(wjm,4)
    end if
    if lcase(kzm) <> ".gif" and lcase(kzm) <> ".bmp" and lcase(kzm) <> ".jpg" then response.
Redirect("add_pic.asp")
    if upload.form("zclj") <> "" or wjdx < 819200 then
        set rs=server.CreateObject("adodb.recordset")
        name=formPath + replace(cstr(date()),"-","") + replace(cstr(time()),":","") + kzm
        sql="select * from zp"
        rs.open sql,conn,1,3
        rs.addnew
        rs("rq")=right(cstr(date()),len(cstr(date()))-2)+" "+left(cstr(time()),instrrev
(cstr(time()),":")-1)
        if upload.form("zclj")="" then
            rs("zp")=name
            dx=cstr(file.filesize/1024)
            if len(dx) > 6 then dx=left(dx,6)
            rs("wjdx") = dx + "K"
        else
            rs("zp")=wjm
            rs("wjdx") = "0K"
        end if
        rs("lx")=upload.form("qx")
        zt=upload.form("zt")
        zt=xrz(zt)
        rs("zt")=zt
        yhm=session("picyhm")
        rs("yhm")=yhm
        if wjdx>0 and wjdx<819200 and upload.form("zclj")="" then
            file.SaveAs Server.mappath(name)
        end if
        iCount=iCount+1
        rs.update
        rs.close
        cg="1"
    end if
    set file=nothing
next
```

```
set upload=nothing ''删除此对象
sub HtmEnd(Msg)
    set upload=nothing
    response.end
end sub
if cg="1" then response.redirect "add_pic.asp">
不能上传大于800K的文件(0.8M)!
</body>
```

22.6 实践总结

本章详细介绍了电子相册的制作方法。在本系统中涉及一个非常关键的技术,即如何远程控制和操作文件和文件夹。在前面介绍的实例中都是通过数据库来实现对远程数据的读写,其技术核心是 ADO 数据库组件,而要想完全管理和操作服务器端文件系统,需要借用 File Access 组件。在本章介绍了 FileSystemObject 对象的基本使用,实际上 File Access 组件还包括 TextStream 对象、File 对象、Folder 对象和 Drives 对象,这些对象组合使用可以完全控制服务器端文件系统的操作。

第 23 章

同 学 录

同学录是一个为同学间交流、感情维系提供服务的网上平台。

本实例包括了同学录的基本功能，适合读者初步学习同学录这种时尚的 Web 应用形式。本系统共分前台和后台两个部分，限于篇幅本章将重点介绍前台页面的技术实现，例如，同学名片、留言、小纸条、图片分享等。有关后台技术请读者参考本书光盘实例。

技术要点

- 同学录设计思路和方法
- 分类相册技术
- 名片设计技术
- 留言条技术
- 后台管理技术

23.1 程序设计分析

本实例同学录的前台功能主要包括班级主页、班级介绍、班级留言、班级相册、留言条、注册用户、修改用户信息、找回密码等功能。整个前台浏览效果如图 23.1 所示。前台程序设计流程如图 23.2 所示。

要实现本例功能，读者应该学习和掌握下面技术难点。

- 用户分级登录与管理，用户在线跟踪技术。
- 图片分类上传与显示。
- 私下留言条技术。
- 名片设计与实现。
- 找回密码技术。
- 注册、登录与修改资料。
- 后台分类管理实现。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

把本书光盘中的“23”/“实例模板”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的基本页面结构、数据库结构，并帮助读者建立了数据库连接。



(a)



(b)



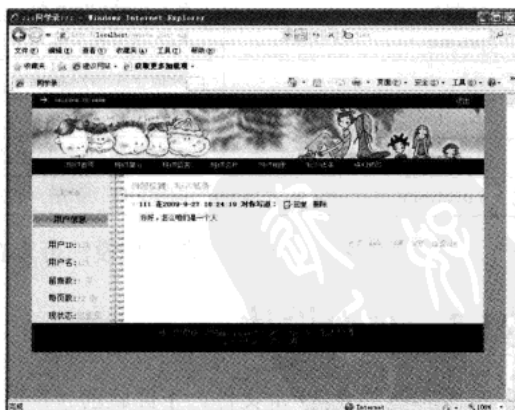
(c)



(d)



(e)



(f)

图 23.1 同学录前台浏览效果

(a) 同学录主页; (b) 修改资料; (c) 班级留言; (d) 班级名片; (e) 班级相册; (f) 留言板

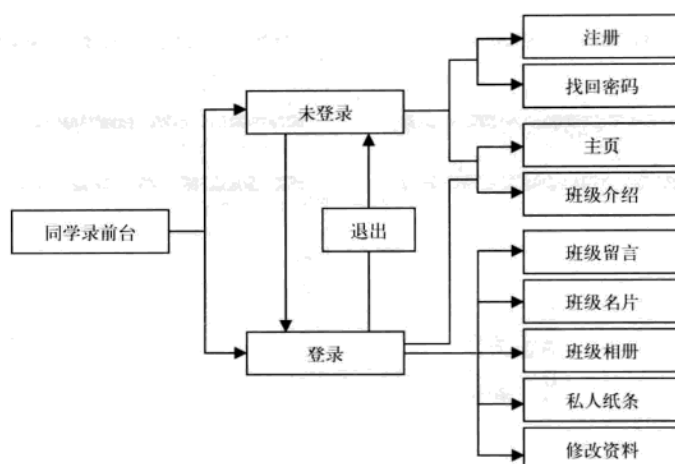


图 23.2 同学录前台设计流程图

23.2 数据库设计

本例数据库 txl.mdb 包括 10 个表，各表的功能见表 23.1。

表 23.1 数据库 txl.mdb 各个表功能介绍

表 名	说 明
classnote	存储班级留言相关信息
classnotice	存储班级公告相关信息
classpic	存储班级相册的相关信息
notdown	存储二进制数字
Online	用户在线活动记录
private	私人纸条相关信息
resume	班级学生信息记录
syssetup	系统信息记录
Txluser	存储注册用户相关信息
Loveword	同学之间互赠赠言的相关信息

关于各个表格结构与字段属性设计可查看光盘实例。

23.3 页面设计

本实例共有 39 个文件，下面就前台部分所涉及的 29 个文件进行具体说明，见表 23.2。有关后台管理模块文件可查看光盘实例。下面就几个主要页面设计进行介绍。

表 23.2 前台系统文件列表

文 件 名	说 明	文 件 名	说 明
addpic.asp	发送图片页面	look.asp	放大查看图片页面
card.asp	班级相册主页面	modify.asp	个人资料修改页面

续表

文 件 名	说 明	文 件 名	说 明
char.asp	正确与错误反馈函数文件	note.asp	班级留言主页面
footer.asp	版权信息页面	notecon.asp	班级留言处理文件
get_pwd.asp	找回密码第一步页面	online.asp	用户在线检测与处理文件
get_pwd2.asp	找回密码第二步页面	picsave.asp	上传图片保存文件
get_pwd3.asp	找回密码第三步页面	picupload.asp	上传图片页面
go.asp	发布成功提示信息页面	quit.asp	退出处理文件
guest.asp	私人纸条主页面	reg.asp	新用户注册页面
header.asp	标题栏与导航页面	regpost.asp	注册用户成功提示页面
index.asp	同学录前台主页面	sender.asp	发送纸条页面
int.asp	班级介绍主页面	shownote.asp	显示留言页面
left.asp	主页面的左侧信息栏页面	showpic.asp	班级相册主页面
login.asp	登录处理文件	syssetup.asp	系统信息文件
login2.asp	登录处理文件		

23.3.1 标题导航栏页面

header.asp 文件是一个标题栏和栏目导航页面，页面设计效果如图 23.3 所示。



图 23.3 header.asp 页面设计效果

在该页面中，主要利用 CSS 样式表技术实现动态导航栏效果，运行如图 23.4 所示。

在 header.asp 文件的【代码】视图中，先插入一个 3 行 1 列的表格，前两行主要是导入图片，在第 3 行单元格中输入下面代码。

```
<div class="test">
<ul>
  <li><a href="index.asp" target="_parent">班级首页</a></li>
  <li><a href="int.asp" target="_parent">班级简介</a></li>
  <li><a href="note.asp" target="_parent">班级留言</a></li>
  <li><a href="card.asp" target="_parent">班级名片</a></li>
  <li><a href="showpic.asp" target="_parent">班级相册</a></li>
  <li><a href="guest.asp">私人纸条</a></li>
</ul>
```

```
<li><a href="modify.asp" target="_parent">资料修改</a></li>
</ul>
</div>
```

上面代码以项目列表的形式输入要设置的导航栏目, 在没有 CSS 样式表控制的情况下, 显示效果如图 23.5 所示。



图 23.4 导航栏动态运行效果



图 23.5 项目列表显示效果

打开 style.css 样式表文件, 输入下面代码, 就可以实现动态导航显示效果。

```
.test ul{ /* 清除项目列表的默认样式并定义左侧留出3px的空隙 */
    list-style:none;
    padding:0;
    margin:0;
    padding-left:3em;
}
.test li{ /* 定义项目列表向左浮动, 实现并列显示, 同时定义背景色为黑色, 左侧留出2px的空隙 */
    float:left;
    background:#000;
    margin-left:2px;
}
.test a{ /* 定义超链接元素a块状显示 */
    display:block;
    text-align:center; /* 内部文本居中显示 */
    height:25px; /* 定义元素的高度 */
    width:80px; /* 定义元素的宽度 */
    line-height:25px; /* 定义行高等于元素的高度, 实现文本垂直居中 */
    color:#FFF; /* 定义字体显示为白色 */
}
.test a:link{ /* 定义a元素超链接默认状态下的样式 */
    background:url(arrow_off.gif) no-repeat 5px 12px;
    padding-bottom:3px; /* 增加底边空隙, 防止鼠标移过时文本错位 */
}
.test a:hover{ /* 定义鼠标移过时的样式 */
    font-weight:bold; /* 加粗文本 */
    background: url(topbg.gif) repeat-x left top; /* 替换背景图像 */
    border-bottom:solid 3px #F59256; /* 增加底边边框 */
    padding-bottom:0; /* 清除底部空隙, 这样该空隙就由增加的边框占据 */
}
```

上面代码主要设置标签属性, 使其不显示项目符号, 并向左浮动, 这样就可以实现所有项目在一行内显示, 然后设置 a 元素的属性为块状显示, 同时设置在不同鼠标状态的显示效果。



提示

CSS 样式表在网页设计中作用非常强大, 主要功能是对页面进行美化与修饰。而且 CSS 技术将是未来网页发展的标准, 要取代表格在网页布局中的主导作用, 建议用户好好学习 CSS 技术。

23.3.2 主页面设计

index.asp 文件作为前台主页面, 主要由三部分文件组成, 即标题导航栏页面文件、左侧信息显示页面文件和版权提示页面, 设计如图 23.6 所示。



图 23.6 前台主页面浏览效果

要实现 index.asp 页面效果, 可以把各个组成页面区域分割为单独的文件, 单独制作完成后再把它们组装在一起, 组装方法可以使用框架集、嵌入框架和导入文件三种。最方便易行的就是使用导入文件方法, 本页导入的代码如下:

```
<body>
<!--#include file="header.asp"-->
<table width="774" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="132" align="center" valign="top" class="left_bg unnamed1">
<!--#include file="left.asp"-->
<td background="images/bg123.gif" width="26"></td>
<td valign="top" bgcolor="#FFFFFF">
.....
</td></tr></table>
<!--#include file="footer.asp"-->
</body>
```

23.3.3 表单样式设计

在本系统中, 很多页面都会使用到文本框、按钮、下拉菜单等表单对象, 为了使整体效果更和谐, 在页面设计中全部使用 CSS 来实现表单外观的修饰, 以便与主体页面色彩体系相一致, 表单设计效果如图 23.7 所示。如果不利用 CSS 样式表修饰, 则效果如图 23.8 所示。

有关 CSS 具体样式代码可以参考实例中的样式表文件。

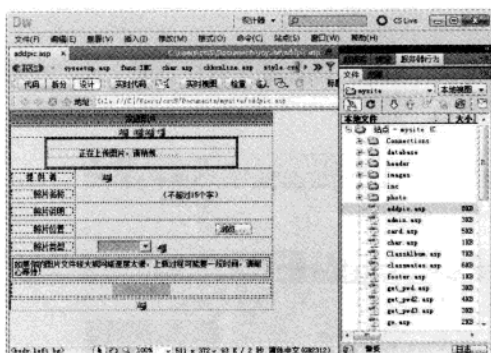


图 23.7 表单设计效果



图 23.8 表单设计效果

23.3.4 班级名片页面设计

card.asp 文件主要是显示班级名片页面, 名片一般制作都比较美观大方, 因此在本例中利用 CSS 样式表技术来制作同学名片效果, 效果如图 23.9 所示。在 Dreamweaver CS5 中利用 div 对象来控制一张名片的外观与样式, 设计效果如图 23.10 所示。



图 23.9 名片设计效果



图 23.10 设计中的名片效果

该名片的 HTML 结构代码如下:

```
<div id="card">
  <div align="center" id="div1"><%=rs("truenname")%></div>
  <div id="div2">"></div>
  <ul id="ul">
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td>QQ:</td>
        <td><%=rs("oicq")%></td>
      </tr>
      <tr>
        <td width="33%">手机:</td>
        <td><%=rs("mobile")%></td>
      </tr>
      <tr>
        <td>Tel:</td>
        <td><%=rs("telephone")%></td>
      </tr>
      <tr>
        <td>E-mail:</td>
        <td><a href="mailto:<%=rs("email")%>"><%=rs("email")%></a></td>
      </tr>
    </table>
  </ul>
</div>
```

```

        </tr>
        <tr>
            <td>工作单位:</td>
            <td><%=rs("address")%></td>
        </tr>
    </table>
</ul>
</div>

```

在 style.css 文件中输入下面代码实现对名片的控制。

```

#card { /*定义名片为绝对定位, 这样它就不受其他元素位置的影响了 */
    position:relative; /*绝对定位*/
    display:block; /*块状显示*/
    width:230px; /*定义卡片的宽度*/
    height:123px; /*定义卡片的高度*/
    background:#CCCCCC; /*定位卡片的背景色*/
    font-size:12px; /*定义字体显示大小*/
    font-family:"幼圆"; /*定义字体*/
    margin:23px; /*增加外边界*/
    padding:2px; /*增加内部边界*/
    background-image:url(images/001.jpg); /*定义背景图像*/
    float:left; /*向左浮动*/
}

#ul { /*定义列表元素的样式*/
    list-style-type:none; /*清除项目列表符号*/
    position:relative; /*定义相对定位*/
    left:5px; /*定义在原来位置的基础上向右移动5px*/
    top:10px; /*定义在原来位置的基础上向下移动10px*/
    margin:0; /*清除列表元素默认外边界*/
    padding:0; /*清除列表元素默认内边界*/
}

#card #div1 { /*定义名片的名称显示位置和样式*/
    margin-bottom:5px;
    margin-top:5px;
    font-family:"隶书";
    font-size:14px;
    color:#FF00FF;
}

#card #div2 { /*定义名片中的照片显示位置*/
    position:absolute;
    right:15px;
    top:10px;
    z-index:3;
}

```

23.3.5 注册表单设计

在本系统前台页面中提供了用户注册与修改资料功能, 这两个功能页面的设计用到了表单技术, 设计如图 23.11 所示。

由于两个页面的表单设计基本相同, 因此可以先制作其中一个表单页面, 然后复制粘贴就可以制作另一个表单页面, 具体设置可以参考光盘实例。

其中“自定义头像”栏中利用嵌入框架技术嵌入一个页面文件, 代码如下:

```

<iframe name="ad" frameborder=0 width=300 height=50 scrolling=no src=picupload.asp>
</iframe>

```

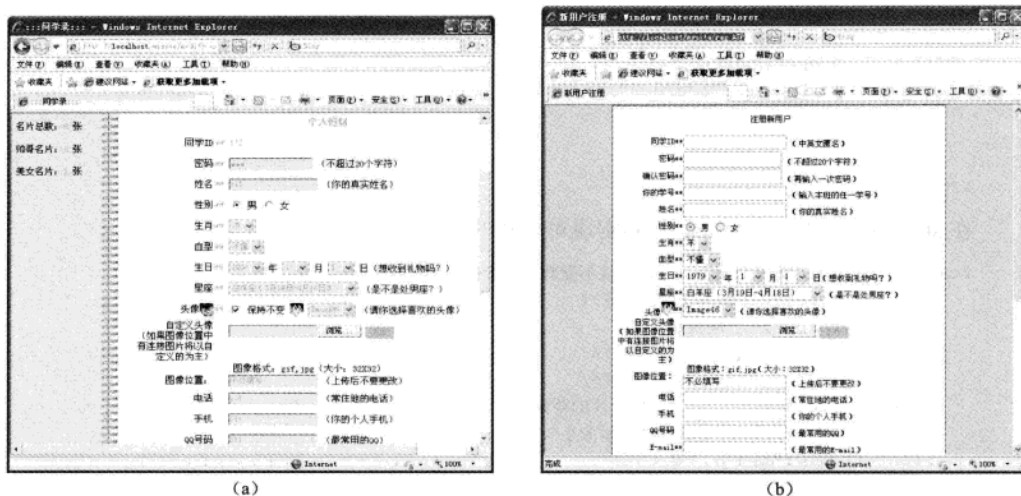


图 23.11 注册与修改资料表单设计效果

(a) 修改用户资料; (b) 注册用户资料

picupload.asp 文件是一个表单对象页面, 包含一个文件域和两个隐藏域, 如图 23.12 所示。



图 23.12 picupload.asp 文件页面效果

23.3.6 动态脚本设计

利用脚本技术可以设计一些动态网页效果, 也可以向其他页面传递参数, 而不需要设计表单域。在“上传图片”和“写纸条”文本链接中, 就是使用动态脚本来完成打开一个文件并传递参数的。

打开 showpic.asp 文件, 选中“上传文件”文本, 在属性面板中设置【链接】文本框为“#”, 建立一个空链接, 如图 23.13 所示。然后切换到【代码】视图, 输入下面代码:

```
<td height="18" valign="bottom"><div align="center"><a href="#" onClick="window.open('addpic.asp', 'ToseePlay', 'top=0, scrollbars=1, resizable=1, height=250, width=423, left='+ (screen.width-650)/2); return false;"><span class="text_red">上传照片</span></a></div></td>
```

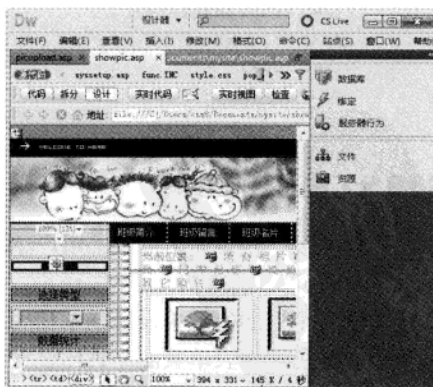


图 23.13 设置空链接

上面方框内的代码表示当用户单击鼠标时,将打开 addpic.asp 文件,并指定打开浏览器窗口大小与位置等属性。

打开 guest.asp 文件,其中的“写纸条”空链接文本的代码设置与 showpic.asp 文件中“上传文件”文本相同,代码如下:

```
<td height="24"><div align="center"><a href="#" title="发送私人留言"
onClick="window.open('sender.asp','ToseePlay','top=0,scrollbars=1,resizable=1,height=160,width=400,left='+ (screen.width-650)/2);return false;"><span class="text_red">写纸条
</span></a></div></td>
```

在上一节的注册表单设计中,关于出生年、月、日下拉菜单列表值也是利用动态脚本实现下拉列表值的自动显示的,其主要利用循环语句来实现列表项的自动设置值,具体代码如下:

```
<select name="byear">
  <%循环设置值
  dim i
  for i=1970 to 1990
  if i=1979 then '判断是否是1979,如果是则设置该数为默认值
  %>
  <option value="<%=i%>" selected><%=i%></option>
  <%else%>
  <option value="<%=i%>"><%=i%></option>
  <%end if%>
  <%next%>
</select>年<select name="bmonth">
  <%dim j
  for j=1 to 12%>
  <option value="<%=j%>"><%=j%></option>
  <%next%>
</select>月<select name="bday">
  <%dim k
  for k=1 to 31%>
  <option value="<%=k%>"><%=k%></option>
  <%next%>
</select>
```

通过动态脚本实现的自动列表值效果如图 23.14 所示,这种方式在数据非常多时就更显得简洁。



图 23.14 设置动态列表值

23.4 同学录前台主要功能实现

由于篇幅的限制,本实例功能没有逐一展开,重点介绍主要功能以及主要页面的实现过程,包括主页、班级留言页面、班级名片页面和班级相册页面。很多具体功能请用户参考光

盘实例，本节重点给出设计思路和主要代码。

23.4.1 同学录主页面

index.asp 文件作为同学录主页面，主要功能是显示各种统计信息，包括用户信息、网站信息、在线信息，以及最新留言、图片和班级公告信息显示。同时会根据用户是否登录、是否是管理员而提供不同的功能页面，如图 23.15 所示。



图 23.15 同学录主页显示分区

主页面功能实现的具体操作步骤如下。

- 01 打开 header.asp 文件，在该页面中判断用户身份而显示不同功能。先判断用户名变量 userid 是否存在，如果存在则定义一个记录集，在数据库中查询该用户的信息，再判断该用户的管理等级；如果变量 manclass 的值是 2，则显示管理功能链接，否则只显示退出功能链接；如果用户编号变量 userid 为空，表明该用户没有登录，则只显示注册和查找密码功能，如图 23.16 所示。具体代码如下：

```
<%
if userid<>"" then
    set rs=server.createobject("ADODB.Recordset")
    sql="select * from txluser where userid='"&userid&'" order by id desc"
    rs.open sql,conn,1,1
    if rs("manclass")=2 then
%>
<a href="admin.asp">管理</a>
<%end if%>
```

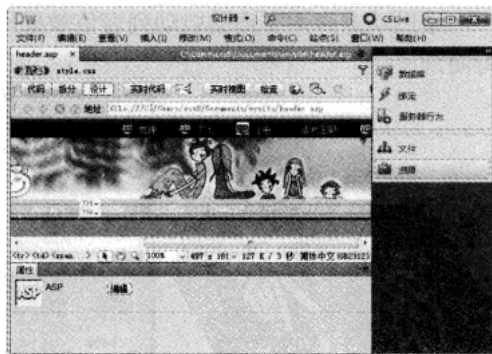


图 23.16 用户身份检测

```

<a href="quit.asp">退出</a>
<%else%>
<a href="reg.asp" target="_blank">注册 </a><a href="get_pwd.asp" target="_blank">查
找密码</a>
<%set rs=nothing
end if
%>

```

02 打开 left.asp 文件, 该页面主要提供用户登录、显示用户信息、站点统计、在线信息等内容, 如图 23.17 所示。

用户登录与检测是本页功能实现的第 1 步。如果 userid<"成立, 则定义记录集查询该用户的信息 (记录集定义可参见上步提供的代码), 并显示用户信息, 如图 23.18 所示。

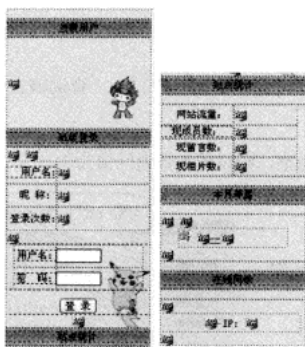


图 23.17 站点信息查询与显示 (一)

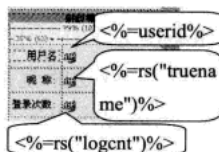


图 23.18 站点信息查询与显示 (二)

如果没有登录则显示一个表单域, 提供一个输入文本框和一个密码域文本框, 并设置表单的【动作】为“login.asp”, 即把表单输入内容提交给 login.asp 文件进行处理。

03 打开 login.asp 文件, 在【代码】视图中输入下面代码, 实现对用户输入信息的检测和处理, 如果输入信息合法并正确则写入用户信息, 并设置 cookies 和 session 变量, 保存用户信息。

```

'导入数据库连接文件
<!--#include file="inc/db.inc" -->
<% '定义函数, 替换字符串中非法字符
Function Sqlstr(data)
    sqlstr=Replace(data,"'", "'")
End Function
userid=sqlstr(request.form("userid"))      '获取用户登录用户名
password=sqlstr(request.form("password"))    '获取用户登录密码
sql= "select userid,password from txluser where userid='"&userid&"' and password=
'"&password&"'"
set rs=conn.execute(sql)                    '查询用户信息记录
if rs.eof or rs.bof then                     '检测该用户是否存在
%>
<script language="javascript">
alert("可能是你的同学ID或密码输入有误!!")
parent.location.href="index.asp"
</script>
<%else '定义用户记录集
mysql="select truename,logcnt,logtime from [txluser] where userid='"&userid&"'"
set rs=Server.CreateObject ("ADODB.RECORDSET")
rs.open mysql,conn,1,3
rs("logcnt")=rs("logcnt")+1                '增加用户登录次数
rs("logtime")=now                          '增加用户登录时间

```

```

truename=rs("truename")      '获取用户真实名称
rs.update                    '更新记录集
set rs=nothing
set conn=nothing
response.cookies("myuserid")=userid      '保存用户输入信息到cookies变量中
response.cookies("mypwd")=password
session("myuserid")=userid              '保存用户输入信息到session变量中
session("mypwd")=request.form("password")
Application("onuser")=truename          '保存用户真实姓名到应用程序全局变量中
response.redirect "index.asp"          '返回主页
end if
%>

```

因为要统计数据库中多个表中的记录数,为了简洁,可以先定义一个函数,用来统计某个表中的记录数,代码如下:

```

function allrec(tabname) '函数名及要统计的数据表
    tmpres=conn.execute("Select count(id) from ["&tabname&"]") '查询该表中编号个数
    allrec=tmpres(0)      '把统计数据反馈给函数本身
    set tmpres=nothing    '关闭记录集
    if isnull(allrec) then allrec=0 '如果反馈为空,则设置函数返回值为0
end function

```

04 在 left.asp 文件中引用该函数,如图 23.19 所示。

05 本月寿星区域功能就是查询用户生日为本月的用户信息,然后显示出来,代码如下:

```

<%
set rs=server.createobject("ADODB.Recordset")
sql="select truename,headpic,birthday from [txluser]"
where month(birthday)=month(Now())"
rs.open sql,conn,1,1
if rs.eof then
    response.write"<center>本月没有寿星</center>"
else
    set truename=rs("truename")
    set headpic=rs("headpic")
    do while not rs.eof
        mymonth=datepart("m",rs("birthday"))&"月"&datepart("d",rs("birthday"))&"日"&"%>
        <table width="75%" height="26" border="0" align="center" cellpadding="0"
        cellspacing="0" bordercolor="#111111" class="unnamed1" >
            <tr>
                <td width="100%" height="26"><img height=16 width=16 src=<%=headpic%>
                <a href=# onclick="window.open('sender.asp','ToseePlay','top=0,
scrollbars=1,resizable=1,height=160,width=400,left='+ (screen.width-650)/2);return false;"
title=给他/她写生日赠言>
                <%=truename%>--<%=mymonth%></a></td>
            </tr>
        </table>

    <%
rs.movenext
loop
set rs=nothing
set truename=nothing
set headpic=nothing
end if%>

```

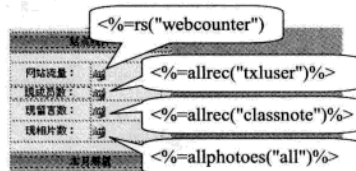


图 21.19 站点统计信息显示

06 在线同学区域功能就是查询数据表 online 中所有在线用户并显示该用户信息,代码和上面基本相同。

```

<%
sql="select truename,lasttime,userip from online"

```

```

set rs=conn.execute(sql)
if not rs.eof then
set onname=rs("truename")
set ontime=rs("lasttime")
set onip=rs("userip")
while not rs.eof%>
<table width="97%" height="25" border="0" cellpadding="0" cellspacing="0" bordercolor=
"#111111" class="unnamed1" >
<tr>
<td width="100%" align="center"><%=onname%><font color="#008000">-</font>
<span class="text_red">IP:
<%
Position = InStrRev(onip, ".")
myip=mid(onip,1,position)
response.write myip&"****"
%>
</span></td>
</tr>
</table>
<%rs.movenext
wend
set rs=nothing
set onname=nothing
set ontime=nothing
set onip=nothing
end if%>

```

07 打开 index.asp 文件，在班级公告栏中主要是查询 classnotice 数据表中的公告信息并显示出来，代码如下：

```

<%set rs=server.createobject("ADODB.Recordset")
sql="select * from classnote order by posttime desc"
rs.open sql,conn,1,1
for i=1 to 2
if rs.eof then
exit for
end if%>
.....
<%rs.movenext
next%>
<%rs.close
set rs=nothing%>

```

然后把其中的变量绑定到相应的单元格中，如图 23.20 所示。

08 最新留言和最新图片都是先定义一个记录集查询相应数据表中的记录，然后用 if 条件语句判断该记录集是否为空，不为空则用 for 循环语句循环显示指定数目的记录，主要代码可参考公告栏代码，设计效果如图 23.21 所示。

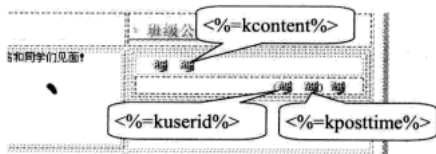


图 23.20 设置班级公告显示区域

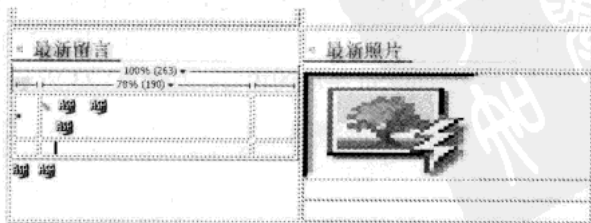


图 23.21 最新留言与图片设计效果

23.4.2 班级留言

班级留言功能模块主要包括阅读留言、写留言、编辑留言和删除留言等，设计如图 23.22 所示。实现该页功能的操作步骤如下。

- 01** 实现增加留言功能。打开 `note.asp` 文件，设置左侧的签写留言表单的【动作】为“`notecon.asp`”，然后在 `notecon.asp` 文件中输入下面代码。

```
<% '获取用户信息，判断用户是否登录
userid=session("myuserid")
password=session("mypwd")
if userid="" then
%>
<script language="Javascript">
alert("你还没有登录，无权进入!");
parent.location.href="index.asp";
</script>
<%
end if '获取表单信息，并进行简单处理
id=request("flag")
content=request.form("content")
content=replace(content,"'", "'",1)
content=replace(content,"<","<",1)
content=replace(content,">",">",1)
if id="add" then '如果隐藏域传递的变量为add，则执行插入记录动作
posttime=now() '获取当前时间
set rs=server.createobject("ADODB.Recordset") '定义记录集
sql="select truename from txluser where userid='"&userid&'"
rs.open sql,conn,1,1
truename=rs("truename") '获取用户真实的名字
rs.close
set rs=nothing '执行插入SQL语句命令
sql="insert into classnote(userid,posttime,content,truename) values ('"
sql=sql&userid&','&posttime&','&content&','&truename&')
conn.execute(sql)
response.redirect "note.asp"
else '如果隐藏域传递的变量不为add，则执行更新记录动作
conn.execute "update classnote set content='"&content&'" where id='"&id
conn.close
set conn=nothing
response.redirect "note.asp"
end if%>
```



图 23.22 班级留言设计效果

- 02** 返回到 `note.asp` 文件，选中 `content` 文本域，在【属性】面板的【初始值】文本框中输入下面代码，判断用户的操作动作。如果是修改留言，则查询要修改留言的记录，并获取该记录中 `content` 字段的内容，如图 23.23 所示。

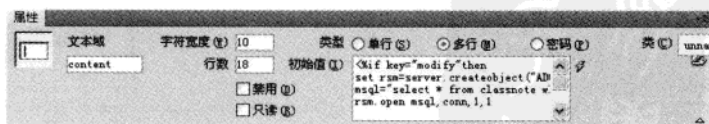


图 23.23 设置文本域初始值

```

<%if key="modify"then
set rsm=server.createobject("ADODB.Recordset")
msql="select * from classnote where id="&id
rsm.open msql,conn,1,1
response.write rsm("content")
end if%>

```

03 在页面右侧单元格中输入下面代码，实现留言显示。

```

<%set rs=server.createobject("ADODB.Recordset") '定义记录集，查询留言信息
sql="select * from classnote order by posttime desc"
rs.open sql,conn,1,1 '判断如果没有记录则给出无留言提示
if rs.eof then%>暂时没有班级留言
<%else
Rs.PageSize=5 '定义每页显示5条记录
Pageno=CLng(Request("page")) '获取当前页
If Pageno<1 Then Pageno=1 '处理当前页，避免超出页数范围
If Pageno>Rs.PageCount Then Pageno=Rs.PageCount
If pageno=""Then Pageno=1
%>
<%
Sub Showpage(Rs,pageno) '定义一个过程
Rs.AbsolutePage=pageno '设置当前要显示的页
for I=1 To Rs.PageSize '循环显示留言记录
%>


### 04 设置导航菜单。在留言显示区域的下一行中输入下面代码。



```

<%
If Pageno<>1 Then
Response.Write" 首页 ";
Response.Write" 前页 ";
Else
Response.Write" 首页 ";

```



23



507


```

```
Response.Write"&nbsp;<br>前页&nbsp;"  
End If  
If Pageno<>rs.PageCount Then  
Response.Write"&nbsp;<a href=note.asp?page="(Pageno+1)">后页</A>&nbsp;"  
Response.Write"&nbsp;<a href=note.asp?page="(Rs.PageCount)">末页</A>&nbsp;"  
Else  
Response.Write"&nbsp;<br>后页&nbsp;"  
Response.Write"&nbsp;<br>末页&nbsp;"  
End If  
%>
```

<form action='note.asp"' method="get" name="select"><div><select name=page class="menu"> <% For I=1 To Rs.PageCount Response.Write"<option>" & I & "</option>" Next %> </select> 页&nbsp; <input ><br="" style="width:26px;" type="submit" value="GO"/></form></td><td style="width: 9%; align: right;">位置<%=pageno%>/<%=rs.pagecount%> </td></td></div></form>	位置<%=pageno%>/<%=rs.pagecount%> </td>
</table> <%end if%>	

23.4.3 班级名片

班级名片主要利用样式表实现名片的制作,然后通过简单的代码来动态控制名片的显示,如图 23.24 所示。



图 23.24 名片设计效果

有关班级名片样式表在前面章节中已经介绍过，其功能设计代码如下：

```
<% '定义记录集，查询班级同学名片信息
set rs=server.createobject("ADODB.Recordset")
sql="select telephone,mobile,trueName,oicq,address,headpic,email from txluser order by
regtime desc"
rs.open sql,conn,1,1
for i=1 to rs.RecordCount      '循环显示所有名片信息
if rs.eof then
exit for
```

```

end if
%>
<div id="card">
  <div align="center" id="div1"><%=rs("truename")%></div>
  <div id="div2">"></div>
  <ul id="ul">
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td>QQ:</td>
        <td><%=rs("oicq")%></td>
      </tr>
      <tr>
        <td width="33%">手机:</td>
        <td><%=rs("mobile")%></td>
      </tr>
      <tr>
        <td>Tel:</td>
        <td><%=rs("telephone")%></td>
      </tr>
      <tr>
        <td>E-mail:</td>
        <td><a href="mailto:<%=rs("email")%>"><%=rs("email")%></a></td>
      </tr>
      <tr>
        <td>工作单位:</td>
        <td><%=rs("address")%></td>
      </tr>
    </table>
  </ul>
</div>
<%=rs.movenext
next%>
<%=rs.close
set rs=nothing
%>

```

23.4.4 班级相册

班级相册主要涉及图片上传技术，这里利用第3方提供的组件实现图片的上传。该技术在第17章中已经介绍，这里就不再重复，具体代码输入可参见光盘中的实例。相册的显示设计如图23.25所示，具体操作步骤如下。



图 23.25 班级相册设计效果

01 在显示相册时先定义一个记录集，根据用户选择类型查询数据库中相关相片信息。

```
<%dim rs,sql
set rs=server.createobject("adodb.recordset")
if keytype="all" then
sql="select classpic.*,txluser.trueusername from [classpic],[txluser] wheretxluser.userid=
classpic.name order by picid desc"
else
sql="select classpic.*,txluser.trueusername from [classpic],[txluser] where txluser.
userid=classpic.name and classpic.type='&keytype&' order by picid desc"
end if
rs.open sql,conn,1,1
if rs.eof and rs.bof then
%>
```

02 判断该记录集是否为空，分别显示不同的提示信息。

```
<td height="40"><span class="title1">当前位置: </span>
span class="title">
<%if keytype="all" then%>
所有相片 <%elseif keytype="校园风光" then%>
校园风光 <%elseif keytype="同学合影" then%>
同学合影 <%elseif keytype="个人照片" then%>
真我风采 <%else%>
其他照片 <%end if%>
</span></td>
```

03 利用循环语句重复显示所有相片。

23.5 实践总结

本章重点介绍了同学录系统的前台设计和主要页面制作方法。由于篇幅有限和系统庞杂，仅就前台几个重点页面功能进行介绍。关于更多页面设计和详细功能实现，可以参照本书光盘中的实例进行研究学习，特别是后台管理技术，用户可反复琢磨并实践。

每一个网站都可以分成两个部分：一部分是浏览者可以在网上看到的部分，即前台部分；另一部分是网站管理人员对网站进行管理的部分，即后台管理部分。还可以将这两部分按照功能模块进一步划分，实际上在前台部分划分的每一个功能模块在后台管理部分都有相应的功能模块与之对应，用户可以运行本实例研究前后台技术的运用和对应关系。

播客 (Podcast) 源自苹果电脑的 iPod 和 broadcast (广播) 的合成, 它表示在互联网上发布多媒体文件并允许用户订阅。播客在 2004 年下半年开始在互联网上流行并用来发布流媒体 (音频、视频文件)。播客和博客都是个人通过互联网发布信息的方式, 并且都需要借助于博客/播客发布程序 (通常为第三方提供的博客托管服务, 也可以是独立的个人博客/播客网站) 进行信息发布和管理。它们主要区别在于, 博客所传播的信息以文字和图片为主, 而播客传递的则是音频和视频信息为主。

本章将介绍播客应用程序的设计思路和实现技巧。流媒体有多种格式, 本章将以 Flash 流媒体格式进行介绍 (这是当前播客系统使用最广泛、最流行的流媒体格式)。由于播客应用程序系统涉及到流媒体服务器, 将以 Flash Media Server 3.5 流媒体服务器为技术框架展开介绍, 重点介绍 Flash Media Server 服务器环境的搭建和设置, 以及如何使用 Dreamweaver CS5 工具建立与 Flash Media Server 服务器通信连接和应用。

技术要点

- Flash Media Server 服务器应用
- FLV 流媒体控制和播放
- FLV 流媒体应用和管理

24.1 程序设计分析

流媒体是实现宽频应用的技术核心, 因此现在有关流媒体 (Streaming Media) 技术的研究和应用也就成为了互联网上最热门的话题。宽频时代的到来使流媒体不仅能够满足用户的各种服务需求, 而且还成为了人们重要的交流方式, 并且创造巨额的价值。很多用户希望搭建自己的网络多媒体应用系统, 但是由于技术的原因, 大多数人只能望而却步。

不过, 在开始正式的流媒体服务器架设之前, 读者应该先了解一下流媒体服务器的基础知识。目前, 在网络上传输音/视频等多媒体信息有两种解决方案: HTTP 或 FTP 下载, 以及流式传输。

HTTP 或 FTP 下载使用标准的 HTTP 和 FTP 协议, 但是由于多媒体信息都非常巨大, 下载一个多媒体文件一般需要几分钟或几小时的时间, 这就造成了看一个并不知道内容的视频, 首先需要耗费可能比整个视频都要长的时间来完成下载。这些被下载的文件还必须在下载前制作完成, 放在网络服务器上, 这样造成的直接后果就是: 网络带宽不断提高, 人们下

载的等待时间越来越少,但最终还是不能观看网上现场直播。

流式传输时,声音、影像或动画等多媒体信息由流媒体服务器向用户计算机连续、实时传送,它首先在使用者端的电脑上创建一个缓冲区,于播放前预先下载一段资料作为缓冲,用户不必等到整个文件全部下载完毕,而只需经过几秒或十数秒的启动延时即可进行观看。当多媒体信息在客户机上播放时,文件的剩余部分将在后台从服务器内继续下载。如果网络连接速度小于播放的多媒体信息需要的速度时,播放程序就会取用先前建立的一小段缓冲区内的资料,避免播放的中断,使得播放品质得以维持。

流式传输除了能够发送已经制作完成的文件外,还可以通过采集服务器实时采集现场音视频,发送到流媒体服务器端,实时提供给用户。因此,流媒体除了能够更好地承担多媒体下载一样的点播服务外,更能够应用在现场直播、电视转播、突发事件报道等多种实时性传输领域。

在流媒体传输中,标准的协议包括 RTP (Real-time Transport Protocol, 实时传输协议)、RTCP (Real-time Transport Control Protocol, 实时传输控制协议)、RTSP (Real Time Streaming Protocol, 实时流媒体协议) 和 RSVP (Resource Reserve Protocol, 资源预订协议),各种流媒体产品都是在这一些协议的基础上进行研究与开发。

流媒体文件格式一般都经过特殊编码,不仅采用较高的压缩比,还加入了许多控制信息,使其适合在网络上实时下载和播放,而不是等到下载完整的文件后才播放。常见的流媒体格式包括: asf (Advanced Streaming Format, 微软公司格式)、wmv (Windows Media Video, 微软公司格式)、rm (Real Video/Audio)、ra (Real Audio)、swf (Shock Wave Flash, Adobe 公司格式)、mov (QuickTime, Apple 公司格式)。

流媒体服务器与 Web 服务器(普通网站)相比,由于多媒体文件需要更强大的处理能力,因此它对于硬件和环境的要求都比较高。例如,流媒体服务器需要超强的 CPU 处理能力、大容量内存和硬盘等。读者可以利用自己的计算机搭建和模拟流媒体服务器环境,如果要构建商业播客系统或者对互联网开放的个人播客站点,读者应该向互联网服务商申请流媒体专业服务(一般费用都比较贵)。

本实例简单演示播客应用系统的核心功能实现,重点介绍如何搭建 FMS 服务器,实现流媒体信息通信。本实例演示效果如图 24.1 所示,程序设计流程如图 24.2 所示。



(a)



(b)

图 24.1 播客应用系统浏览效果

(a) 播客应用系统首页; (b) 播放 Flash 流媒体视频

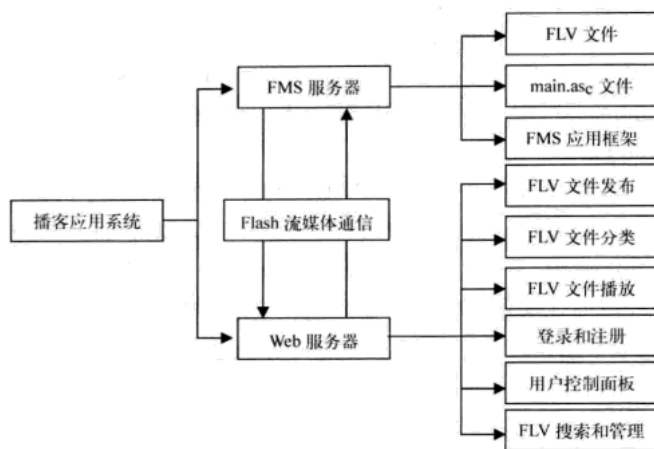


图 24.2 播客系统的设计流程图

要实现本例功能，读者需要学习和掌握下面技术难点。

- 安装和配置 Flash Media Server 服务器。
- 建立 FMS 服务器通信连接。
- 在 Web 服务器动态设置通信信息。
- FLV 流媒体播放技术。



提示

学习本实例之前，读者应该准备如下工作。

在【Internet 信息服务】窗口中建立一个虚拟目录，启动 Dreamweaver CS5 创建一个动态站点，远程文件夹应该指向虚拟目录所对应的本地目录。动态站点的【服务器类型】为“ASP VBScript”，【访问】应为“本地/网络”。

根据下面章节介绍的方法和操作步骤安装和配置 Flash Media Server 服务器。

把本书光盘中的“24”/“实例模板”/“Web 服务器端文件”文件夹中的文件全部复制到本地和远程文件夹中。实例模板提供了本实例的页面结构、数据库结构，并帮助读者建立应用程序的基本功能。

24.2 安装 FMS 服务器（Flash Media Server）

FLV 是 Flash Video 的简称，翻译为 Flash 视频，是现在存在的主流在线视频格式。该视频格式最先应用于 youtube.com 网站，由于 Google 公司以天价收购了该网站，从而引发全球对于该网站的关注，FLV 也因此红遍全球，成为最流行、最受欢迎的流媒体格式，目前国内主要视频网站都采用这种格式。

以前大家都使用 asf、wmv 和 rm 等视频格式，但是由于这些格式需要选择不同的播放器，例如，asf 和 wmv 格式使用 Media Player 进行播放，而 rm 格式用 RealPlayer 播放。这对于本地计算机没有安装相应播放器的用户来说，这些视频根本就无法收看。同时由于这些文件的容量大，下载速度比较慢，观看也不是很流畅，例如，图 24.3 所示的是中央电视台网站所使用的 wmv 格式的流媒体服务器播放的在线视频，播放速度非常慢。

而 FLV 格式播放器嵌入在浏览器中的 Flash 播放器中，所以一般用户都可以在线播放 FLV 文件，同时 FLV 文件经过全新的 Flash 算法设计，可缩小原有视频的容量，最终转换的文件

扩展名是 flv，多媒体文件大小就相对更小了。

FLV 文件播放需要流媒体服务器的支持，一般为 FMS (Flash Media Server)，最先称为 FCS (Flash Communication Server)。利用该技术可以开发 Flash 聊天室、视频会议，以及当下流行的播客等 Web 应用系统。

用户开发 FMS 应用程序，需要下载 FMS 服务程序，当前版本为 Flash Media Server 3.5.2，下载地址为 http://www.adobe.com/support/flashmediaserver/downloads_updaters.html。这是一个免费的开发者版，功能有限，支持的连接数比较少，当然如果读者仅是学习、练习使用已经够用了。但如果要开发商业应用系统，需要购买正式版（费用为几万美元，一般用户会承担不起），它可以支持无数用户连接，而且能够更大潜能地发挥服务器的工作效率。

Flash Media Server 3.5（免费版）服务器程序比较小，大小为 94.9Mb，下载到本地后显示图标如图 24.4 所示。



图 24.3 wmv 格式的流媒体服务器



FlashMediaServer3.5

图 24.4 Flash Media Server 3.5 服务器程序图标

Flash Media Server 3.5 服务器的安装很容易，安装中需要读者选择一个管理员用户名和密码，当然也可以在以后通过管理控制台来改变这个用户名和密码。具体操作步骤如下。



提示

由于 Flash Media Server 被设计成使网络连接模仿调制解调器或 DSL 线路以便节省带宽，如 NetLimiter，在安装 Flash Media Server 之前，应该移除这种软件类型。同样，不要在 Flash Media Server 被安装之后再安装这种软件类型。读者如果在客户端使用这种软件类型，那就不要在服务器所在的计算机上使用它。

- 01 下载 Flash Media Server 3.5 服务器程序之后，双击 Flash Media Server 3.5.exe 安装程序图标。安装程序将被启动，根据安装向导输入序列号，具体就不再显示。
- 02 接下来设置 Flash Media Server 安装路径，一般可以保存到系统提供的默认位置。同时接受为 Flash Media Server 程序快捷方式提供的默认位置，或是输入一个新位置用于快捷方式的安装。
- 03 在显示的窗口中输入用户名和密码作为第一个有效的服务器管理员，如图 24.5 所示。这些值

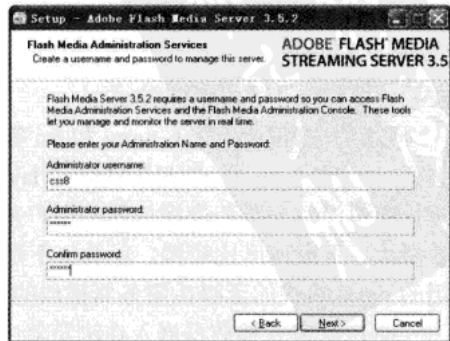


图 24.5 输入用户名和密码

将被存储在 `fms.ini` 文件中, 相对安装目录为 `\conf\fms.ini` (读者要记住该用户名和密码, 否则将无法进入管理控制台, 如果忘记可以在 `fms.ini` 文件中查找), 以后用户就可以在管理控制台中修改和添加其他管理员。

- 04** 输入 Flash Media Server 使用的服务器端口。读者可以保持默认设置, 如图 24.6 所示。默认端口是 1935, 80, 管理端口是 1111, 如果计算机安装有防火墙, 要记住把 1935, 80 和 1111 端口打开。

- 05** 在安装过程中可以复查安装选择, 统计信息如下。单击 **Back** 按钮修改前面的设置。最后单击 **【Install】** 按钮安装即可。

Summary:

```
Server Version: Adobe Flash Media Streaming Server
Destination Location:
    C:\Program Files\Adobe\Flash Media Server 3.5
Applications Directory:
    C:\Program Files\Adobe\Flash Media Server 3.5\applications
Web Server Version: Apache2.2
Web Server Location:
    C:\Program Files\Adobe\Flash Media Server 3.5\Apache2.2
Web Root Directory:
    C:\Program Files\Adobe\Flash Media Server 3.5\webroot
Server Port: 1935,80
Administration Port: 1111
Administration Name: css8
```

- 06** 最后的安装步骤用户可以查看 `readme.htm` 文件信息, 或者启动 Flash Media Server, 并选择当重新启动计算机时手动启动 Flash Media Server 的选项。选择任何自己喜欢的选项, 单击 **【Finish】** 按钮即可安装完成。如果按默认配置完成安装, 则 Flash Media Server 服务将自动启动。

- 07** 要手动启动服务器, 读者可以选择 **【开始】|【程序】|【Adobe】|【Flash Media Server 3.5】|【Start Adobe Flash Media Server 3.5.2】** 和 **【Start Flash Media Administration Server 3.5.2】** 菜单命令。

- 08** 要打开 Flash Media Server 管理控制台, 读者可以选择 **【开始】|【程序】|【Adobe】|【Flash Media Server 3.5】|【Flash Media Administration Console】** 命令即可, 打开的 Management Console 管理控制台如图 24.7 所示。读者可能会奇怪, 它并不是一个应用程序, 管理控制界面仅仅是一个 HTML 文档, 非常有趣, 因为你正在处理的是一个服务器而不是一个单机版的应用程序。

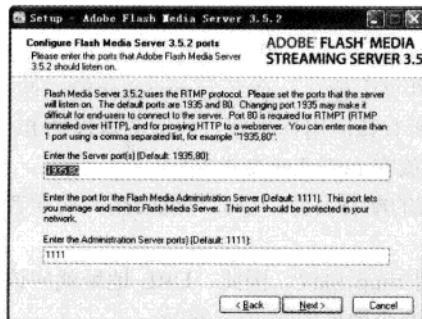


图 24.6 设置服务器端口号

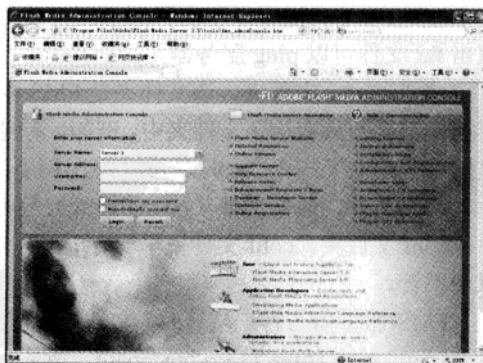


图 24.7 Management Console 管理控制台



有些杀毒软件,例如, Norton AntiVirus 和 Panda Antivirus 等,以及那些会为系统级套接字调用安装它们自己的包装的程序,可能导致 Flash Media Server 安装失败。在安装 Flash Media Server 过程中如果遇到了问题且有杀毒软件安装在计算机中,则应该先移除这些杀毒软件。

24.3 管理 FMS 服务器

安装 Flash Media Server 服务器之后, FMS 还不能够直接应用, 读者还需要简单了解和配置 FMS 服务器。

FMS 服务器默认安装之后, 所有文件都保存在 C:\Program Files\Adobe\Flash Media Server 3.5 目录下, 如图 24.8 所示。



图 24.8 Flash Media Server 服务器安装目录

当读者在本地计算机中安装 FMS 之后, 本地计算机就是 FMS 服务器, 同时也是客户端计算机, 如果安装了 IIS, 本地计算机还是 Web 服务器。

tools 文件夹。读者要启动 FMS 服务, 需要在菜单中启动【Start Flash Media Server】和【Start Flash Media Admin Server】菜单命令 (读者可以参考上节介绍方法)。如果要关闭 FMS 服务, 需要在菜单中选择【Stop Flash Media Admin Server 2】和【Stop Flash Media Server 3.5】命令。如果觉得这样操作麻烦, 可以在图 24.8 所示的目录中打开 tools 文件, 双击批处理文件 StartServerService.bat 开启 FMS 服务, 或者双击 StopServerService.bat 文件关闭 FMS 服务。双击 far 文件可以 ping 服务器是否连接上。

applications 文件夹。该文件夹是 FMS 服务器的核心文件夹。用户所建立的 FMS 服务应用程序都必须放在这个文件夹中。例如, 本例在该文件夹中建立一个名为 myboker 的文件夹, 这就等于建立了一个 Application (应用程序), 以后所有该应用程序项目的服务器端的 flv 文件、共享文件、通信文件 (如 main.asc) 等都必须存放在 myboker 文件夹中。

conf 文件夹。conf 文件夹里保存着一些服务器端的配置文件, 以后可能会用, 初学者可以先不管。

FMS 服务器与客户端 (swf 文件) 通信主要使用 rtmp 协议, 因此, URL 地址就应该是 rtmp://path 或者 rtmp://localhost/path。请读者要注意两个地址中的 “/” 符号。

管理服务器。管理 FMS 服务器可以双击 fms_adminConsole.swf (也可以按上节介绍方法

在菜单中打开), 别看它只是个 swf 文件, 所有后台管理全部都由它来实现, 输入安装时候设置的用户名和密码即可进入管理控制台, 如图 24.9 所示。

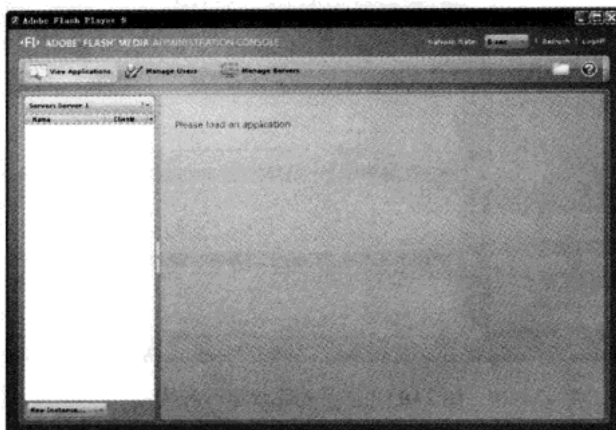


图 24.9 Flash Media Server 管理服务台

Flash Media Server 管理服务台提供了服务器上的实时性能数据。用户可以使用窗口右上角的 Refresh Rate (刷新频率) 弹出菜单来控制管理控制台中的信息刷新频率 (默认的速率是 5 秒)。单击【Refresh】按钮可以随时刷新数据。而单击【Logoff】按钮可以退出当前管理。

如果单击  按钮可以获取在线资源, 单击  按钮可以获取 FMS 服务器帮助。

FMS 管理包括 3 部分: 应用程序管理 (View Applications, 查看应用程序)、用户管理 (Manage Users, 管理用户) 和服务器管理 (Manage Server, 管理服务器)。

24.3.1 应用程序管理

单击控制台主菜单中的 View Applications (查看应用程序), 会切换到应用程序管理界面, 在这里可以显示正运行在指定的服务器或虚拟主机上的应用程序的相关信息。管理员可以实时监视应用程序的运行状态。例如, 当在本实例播客系统中同时打开 4 个播放文件页面, 则在 streams 选项卡中会显示所有正在通信的流媒体的详细信息, 如图 24.10 所示; 在 performance 选项卡中显示当前服务器运行时的系统参数, 如图 24.11 所示。

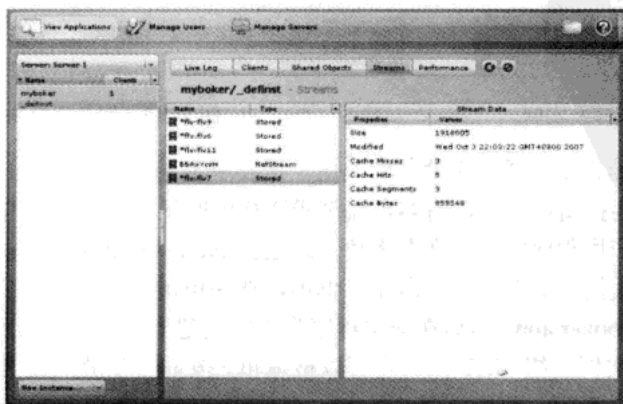


图 24.10 流媒体传输信息

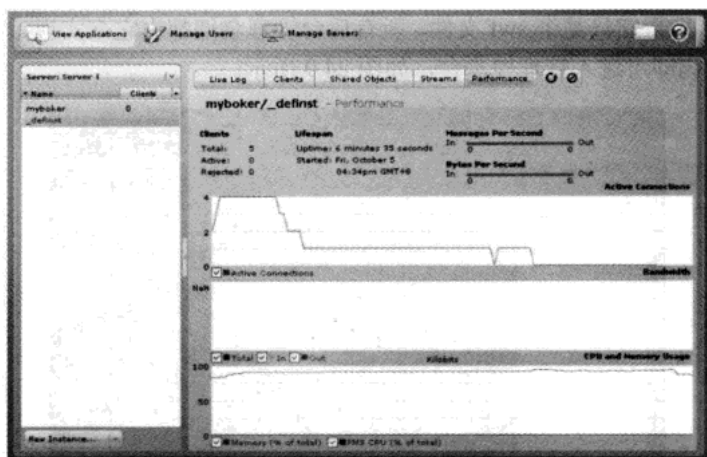




图 24.11 服务器系统运行参数

在开发 FMS 应用程序时，最常用的就是这个 View Applications，每当有客户端连接服务器时，左侧就会列出连接的是哪个应用程序，有多少连接。例如，在图 24.12 中显示为连接 applications 文件夹下的 myboker 文件夹（每个文件夹对应一个 app 应用程序）就是 myboker app，连接数为 5，_definst 是该应用程序的实例名（默认为_definst）。每个文件夹可以有若干个实例，实例之间互不影响，这个特性可以用来开发聊天室中多个独立的房间。

选中某个 app 后，或者客户端有连接，会看到当前打开的 app 的状态，单击 Live Log 选项卡可以查看服务器端的 trace。在该右侧列表框中依次显示客户端情况，例如，连接流媒体名称，如图 24.13 所示。选项卡栏后边两个小按钮  和 （reload 和 unload），调试时候比较常用。

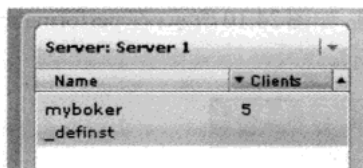


图 24.12 应用程序及实例

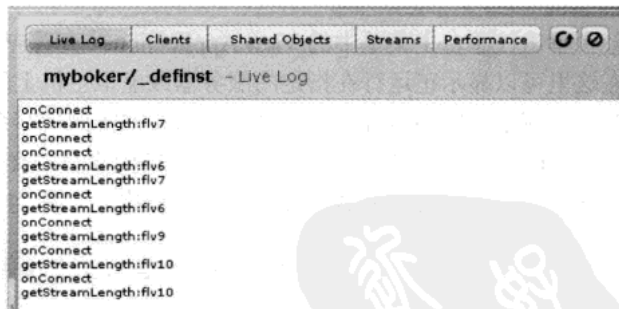


图 24.13 服务器系统运行参数

每当服务器端 main.asc 通信文件被修改后一定要单击 reload 或者 unload 按钮一下，否则不会生效，unload 会把窗口关闭，有客户端请求连接的时候还会自动打开。

创建一个新的应用程序实例。在左下角通过选择 New Instance（新实例）按钮可以创建一个新的应用程序实例，如图 24.14 所示。例如，选中单击该按钮后，在弹出的应用程序列表中选择本实例 myboker app，然后在应用程序列表中创建一个新的应用程序实例。管理控制台添加一个默认的实例前缀_definst_，它可以被编辑。按下回车键来提交名称并启动应用程序实例，如图 24.15 所示。要取消某个实例，只需要选中该实例并按【Esc+Shift】组合键即可。

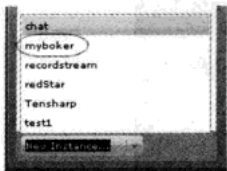


图 24.14 新建应用程序实例

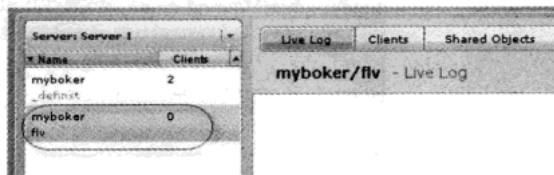


图 24.15 应用程序实例列表

24.3.2 用户管理

单击管理控制台中的 **Manage Users** 菜单，可以切换到用户管理界面，如图 24.16 所示。在这里用户可以控制具有管理权限的 Flash Media Server 用户。

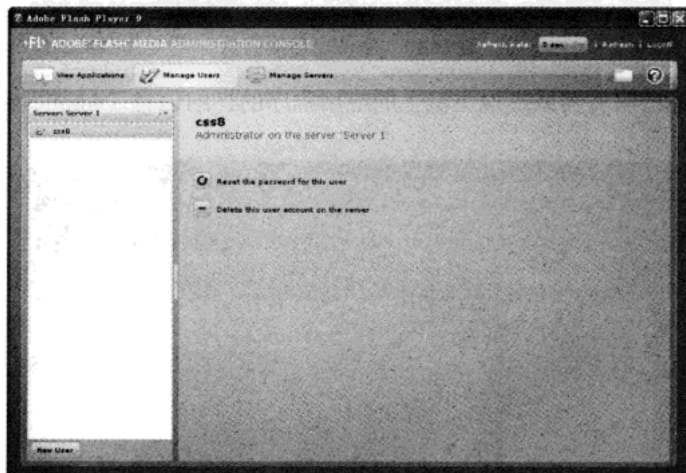


图 24.16 用户管理

在 **Manage Users**（管理用户）界面的底部单击 **New User**（新用户）按钮，可以为服务器添加一个管理员。

在窗口左侧的 **User** 面板中选择一个管理员，可以重新设置它们的密码，或者是从服务器或虚拟主机中删除它们。

当单击 **Reset the password for this user**（为该用户重新设置密码）链接时，会提示请求输入新密码。此时用户可以重新设置该用户的密码，或者单击 **Cancel**（取消）按钮取消操作。

当单击 **Delete this user account on the server**（删除服务器上的该用户账号）链接时，会提示是否确认删除操作。

24.3.3 服务器管理

单击管理控制台中的 **Manage Server** 菜单，可以切换到服务器管理界面，如图 24.9 所示。在这里用户可以控制服务器运行环境。该面板的设计可以方便管理员迅速地监视一组服务器。这些服务器被组织成一个由一台台服务器构成的树形结构。选择相应的选项卡可以完成下面操作。

- 检查应用程序正在其上运行的计算机的性能统计（**Details** 选项卡）。
- 检查服务器连接的相关信息（**Connections** 选项卡），如图 24.17 所示。

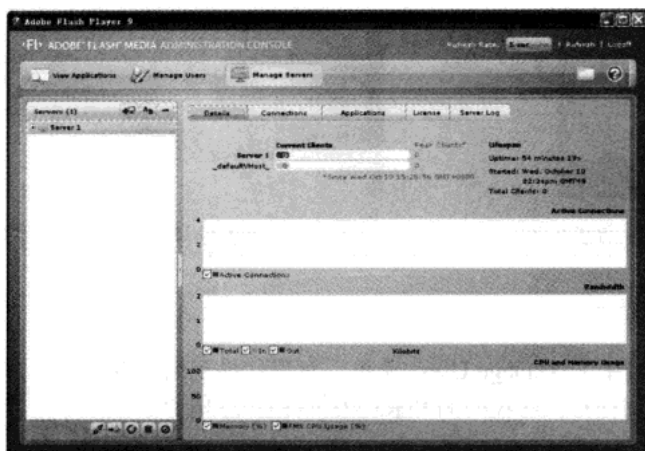


图 24.17 服务器连接信息

- 检查位于这台服务器或虚拟主机上的应用程序的详细信息 (Applications 选项卡), 如图 24.18 所示。

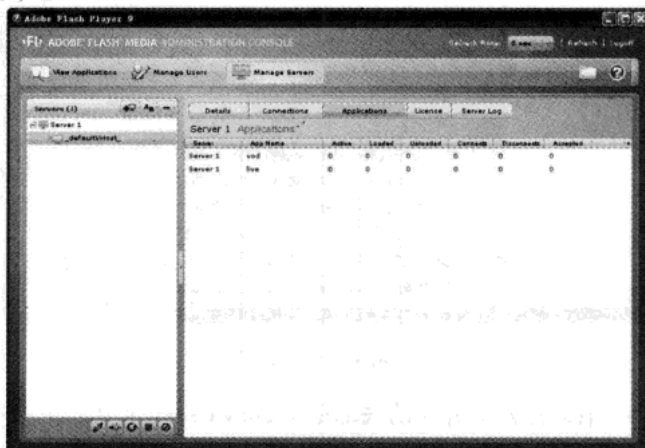


图 24.18 应用程序连接信息

- 检查服务器的许可键和文件 (License 选项卡)。
- 检查服务器的访问日志文件 (Server Log 选项卡)。

24.4 创建播客 FMS 应用程序

初步认识了 FMS 之后, 下面就来建立本播客系统的 FMS 应用程序相关工作。主要工具包括建立 FMS 应用程序、配置 FMS 资源, 操作步骤如下。

- 01** 打开 C:\Program Files\Adobe\Flash Media Server 3.5\applications 文件夹。该路径应该根据用户安装 Flash Media Server 3.5 服务器时设置的目录而确定 (具体设置可以参考第 24.2 节介绍)。
- 02** 在该文件中新建一个 myboker 文件夹, 该文件夹就等于建立一个应用程序的框架, 这个应用程序的名称为 myboker (读者也可以根据需要定义其他名称)。以后所有 FMS 服务

器端文件都必须放在该文件夹中。

- 03 在 myboker 文件夹中新建 streams 文件夹（要注意该名称是固定的），该文件夹一般用来存放流媒体文件，例如，Flash 视频文件。不同类型的 FMS 服务器文件会存储在不同的文件夹中。
- 04 在 streams 文件夹中新建_definst 文件夹（读者也可以定义为其名称），然后把所有准备播放的 FLV 文件都存储在该文件夹中。
- 05 读者应该启动 FMS 管理控制台，在 View Applications 中新建一个应用程序实例，实例名称必须与上面_definst 文件夹名称相同（这样能够保证 FMS 服务器能够找到所需要的 FLV 文件），如图 24.19 所示。
- 06 至此 FMS 服务器主框架已经建立，但还不能进行通信，下一节将利用 Dreamweaver CS5 实现与 FMS 服务器进行通信。

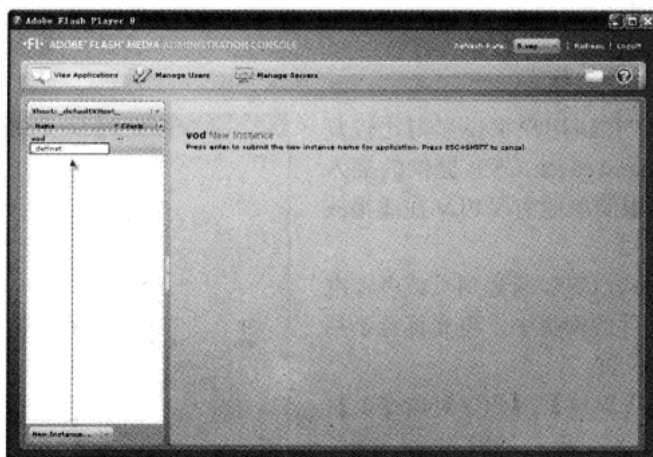


图 24.19 新建应用程序实例



提示

在 Flash Media Server 中不同类型文件应该存储在不同的文件夹中。例如，如果应用程序实例 ChatApp/MondayChat 记录了一个名为 chat 的流，则 chat.flv 文件将被保存在 /applications/ChatApp/streams/MondayChat 文件夹中；如果运行 ChatApp 的一个名为 TuesdayChat 的实例，则它的文件将被存储在 /applications/ChatApp/streams/TuesdayChat 文件夹中。总之，读者应该把不同 FLV 文件存放到 Flash Media Server 会发现它的地方，也就是说一个准确的 /streams 目录的子目录中。用户还可以为进一步组织 FLV 文件创建子目录，只要这些子目录是在正确的应用程序实例子目录中。例如，在 /streams/TuesdayChat 中，读者可以创建额外的子目录 /streams/TuesdayChat/morning 和 /streams/TuesdayChat/afternoon，并在这些子目录中存储 FLV 文件。

24.5 用 Dreamweaver CS5 实现与 FMS 服务器通信

在网页中实现 FLV 文件在线播放，可以有多种方式，常用方式如下。

24.5.1 Flash 方式

将本地 FLV 文件导入并部署到 Flash 源文件中，以库元素的形式而存在，将该元素拖入

场景，调整位置后，发布为 swf 文件即可。这种方式是目前各大视频网站常用的播放方式，它可以实现智能化控制 FLV 文件播放，以及检测 FLV 文件播放状态。当 FLV 文件还没有连接上时，可以显示各种进度条或其他动态提示信息，当 FLV 文件播放完后可以插播广告，以及相关 FLV 文件列表。这种方式体现了视频播放的人性化和易用性，但是由于涉及 Flash 编程，所以本实例不对该方式展开介绍。

注意，以上文件在本地测试应该没有问题，但上传到网络时其他浏览者是无法查看的，因为当前的播放器调用的是本地 FLV 文件，所以需要调整路径。首先需要将 FLV 文件上传到 FMS 服务器端某个网络空间，并且获得其准确的路径地址。

然后打开 Flash，选择 FLV 组件，按【Alt+F7】组合键打开【组件检查器】面板，将“ContentPath”中的路径由本地地址替换为网络绝对地址。

最后将该 Flash 发布导出 swf 文件。这时候才可以任意使用该 swf 文件，如同插入其他一般的 Flash 文件一样。

24.5.2 累进式下载

第一种方法对于初级用户来说显得比较复杂。不过使用 Dreamweaver CS5 提供的插入 Flash 视频功能可以很简单地实现 FLV 在线播放功能，操作步骤如下。

01 启动 Dreamweaver CS5，新建网页或者在准备插入 FLV 文件的网页中，将光标置于播放 FLV 文件的位置。

02 选择【插入】|【媒体】|【FLV】命令，打开【插入 FLV】对话框，如图 24.20 所示。

03 在【URL】文本框中指定 FLV 文件的相对或绝对路径。如果要指定相对路径，例如，mypath/myvideo.flv，用户可以单击【浏览】按钮，在打开的【选择文件】对话框中选择 FLV 文件。如果要指定绝对路径，可以直接输入 FLV 文件的 URL，例如，http://www.example.com/myvideo.flv。

04 在【外观】下拉列表中选择 Flash 视频组件的外观。所选外观的预览会出现在下面预览框中。

05 在【宽度】和【高度】文本框中以像素为单位指定 FLV 文件的宽度。若要让 Dreamweaver 确定 FLV 文件的准确宽度，可以单击【检测大小】按钮。如果 Dreamweaver 无法确定宽度，则必须键入宽度值。如果要保持 Flash 视频组件的宽度和高度之间的纵横比不变，还应该选中【限制】复选框。

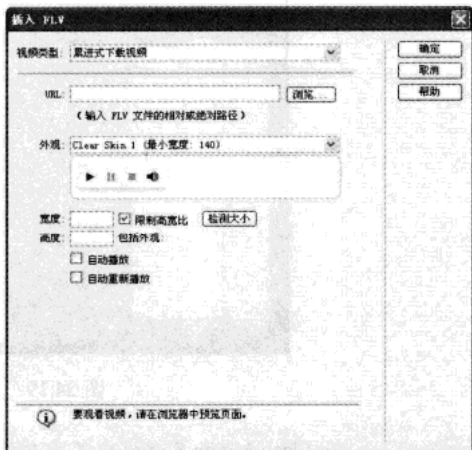


图 24.20 以累进式下载方式插入 FLV 文件



FLV 文件的宽度和高度包括外观的宽度和高度。

06 选中【自动播放】复选框可以指定网页在被打开时自动播放视频。

07 选中【自动重新播放】复选框可以指定播放控件在视频播放完之后返回起始位置重复播放。

- 08 设置完毕,单击【确定】关闭对话框,并将 Flash 视频内容添加到网页中。
- 09 插入 Flash 视频之后,系统会自动生成一个视频播放器 SWF 文件和一个外观 SWF 文件,它们用于在网页上显示 Flash 视频内容。这些文件与 Flash 视频内容所添加到的 HTML 文件存储在同一目录中。当用户上传包含 Flash 视频内容的网页时, Dreamweaver 将把相关文件的上传到服务器,如果没有上传,则读者需要手动上传。

24.5.3 流媒体

第二种方式虽然很简单,但是它极大地增加了服务器处理和带宽传输的负担。一般商业网站不会采用这种方式,它们都采用流媒体的形式实现 FLV 在线播放。流媒体除了使用 Flash 进行控制外,也可以利用 Dreamweaver 直接在网页中以流媒体的方式插入 FLV 文件。这种方式也是本播客系统采用的方式,它操作简单而且能够适应商业视频网站的开发需要。

具体操作步骤如下。

- 01 在 Dreamweaver 中打开【文件】面板,切换到本实例播客系统的“本地视图”。打开 flv_play.asp 文件,该文件专门负责与 FMS 服务器进行通信,并播放指定的 FLV 文件。
- 02 选择【插入】|【媒体】|【FLV】命令,打开【插入 FLV】对话框,然后在【视频类型】下拉菜单中选择“流视频”。详细设置如图 24.21 所示。
- 03 在【服务器 URL】文本框中输入 FMS 服务器中 FLV 文件的 URL。读者必须以 `rtmp://www.example.com/app_name/instance_name` 的形式指定服务器名称、应用程序名称和实例名称。其中 `rtmp` 表示 FMS 流媒体传输协议, `localhost` 表示本地服务器的 URL,如果申请了远程服务器,应该输入远程服务器的 URL, `myboker` 表示在第 24.4 节中建立的播客系统的 FMS 服务器应用程序名称, `_definst` 表示在 FMS 服务管理控制台中建立的实例(参见 24.4 节介绍)。
- 04 在【流名称】文本框中输入想要播放的 FLV 文件的名称, `.flv` 扩展名是可选的。在这里暂时输入“flv1”,在后面还要替换为 ASP 动态信息。
- 05 在【外观】下拉列表中指定 Flash 视频组件的外观。所选外观的预览会出现在下面预览框中。本例选择“Halo Skin 3 (最小宽度: 280)”选项。
- 06 其他选项可以参考图 24.21 所示进行设置。其中必须选中【自动播放】复选框,以实现流媒体自动播放,但是不要选中【自动重新播放】复选框,避免反复播放视频,把是否重复播放的控制权让给浏览者。
- 07 在【缓冲时间】文本框中指定在视频开始播放之前进行缓冲处理所需的时间(以秒为单位)。默认的缓冲时间设置为 0,这样在单击了【播放】按钮后视频会立即开始播放。如果选择【自动播放】,视频将在建立与服务器的连接后立即开始播放。如果用户要发送的视频的比特率高于站点访问者的连接速度,或者网络通信可能会导致带宽或连接问题,则可能需要设置缓冲时间。例如,如果要在网页页面播放视频之前将 15 秒的视频发送到

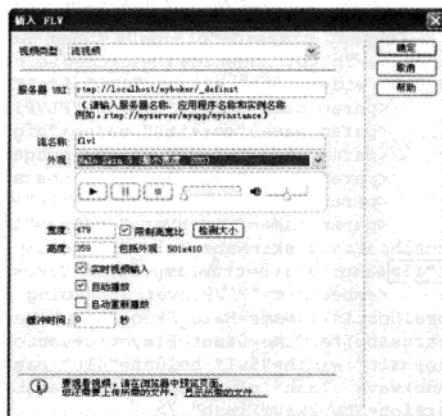


图 24.21 以流媒体方式插入 FLV 文件

网页页面，可以将缓冲时间设置为 15。

- 08** 单击【确定】按钮关闭对话框并将 Flash 视频内容添加到网页中。这时系统会自动生成一个视频播放器 SWF 文件和一个外观 SWF 文件，它们用于在网页上显示 Flash 视频。该命令还会生成一个 main.asc 文件，这是一个 FMS 服务器与客户端进行通信的文件，用户必须将该文件上传到 FMS 服务器中。在本例中读者应该把它复制到 C:\Program Files\Adobe\Flash Media Server 3.5\applications\myboker 目录下，即复制到本 FMS 应用程序目录下。

- 09** 在 Dreamweaver CS5 中切换到【代码】视图，用【查找和替换】对话框（方法是选择【编辑】|【查找和替换】菜单命令）查找“flv1”字符串，该字符串是在【插入 FLV】对话框中设置的临时 FLV 播放文件，然后把它们替换为 ASP 动态信息<%=rs("url")%>，<%=rs("url")%>表示查询数据库中指定 FLV 文件的 URL 字段的值。代码如下：

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8
,0,0,0" width="501" height="410" id="FLVPlayer">
  <param name="movie" value="FLVPlayer_Streaming.swf" />
  <param name="quality" value="high">
  <param name="wmode" value="opaque">
  <param name="scale" value="noscale">
  <param name="salign" value="lt">
  <param name="FlashVars" value="&MM_ComponentVersion=1&serverName=
localhost&skinName=Halo_Skin_3&appName=myboker/_definst&streamName=<%=rs("u
rl")%>&isLive=true&bufferTime=0&autoPlay=true&autoRewind=false" />
  <embed src="FLVPlayer_Streaming.swf" flashvars="&MM_ComponentVersion=1&serverName=
localhost&skinName=Halo_Skin_3&appName=myboker/_definst&streamName=<%=rs("url")%>&isLiv
e=true&bufferTime=0&autoPlay=true&autoRewind=false" quality="high" wmode="opaque" scale=
"noscale" width="501" height="410" name="FLVPlayer" salign="lt" type="application/x-
shockwave-flash" pluginspage="www.adobe.com/shockwave/download/download.cgi?Pl_Prod_
Version=ShockwaveFlash" />
</object>
```

- 10** 由于 flv_play.asp 文件作为一个包含文件被包含在 video_play.asp，所以有关如何定义数据库连接和记录集查询工作将在 video_play.asp 文件中去实现。在 flv_play.asp 文件中读者可以暂时不要管<%=rs("url")%>所代表的具体信息。
- 11** 至此，基本上完成了客户端网页与 FMS 服务器进行通信的工作，所插入的 FLV 视频如图 24.22 所示。这时如果读者选中插入的 FLV 视频文件，还可以在属性面板中进行修改和重新设置。

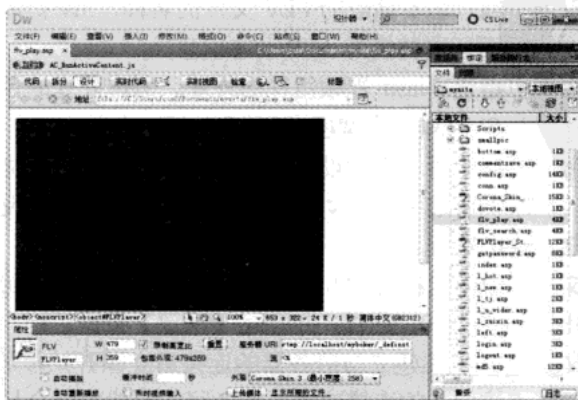


图 24.22 插入的 FLV 文件



提示

实际上读者也可以快速、轻便地利用服务器端 ActionScript 脚本语言, 进行 FMS 通信连接和实时控制 FMS 服务器端 FLV 文件的播放, 但是由于涉及一种新的语言, 所以本书不准备展开介绍, 感兴趣的读者可以找 Flash 相关书籍进行学习。

24.6 播客系统数据库设计

前面的学习过程都是围绕 FMS 服务器展开的, 实际上本播客系统的主要工作还应该在 Web 服务器端进行开发, 例如, 数据库设计与连接, 播客系统的发布、管理, 播客系统各种扩展功能等。本例数据库 flv.mdb 包括 8 个表, 详细说明如表 24.1 所示。

表 24.1 数据库 flv.mdb 各个表功能介绍

表 名	说 明
admin	管理员管理信息
author	用户管理信息
bigclass	FLV 视频文件大类信息
smallclass	FLV 视频文件小类信息
grade	用户级别信息
comment	用户评价信息
config	系统配置信息
flash	FLV 文件详细信息

下面就其中主要数据表的结构与字段属性设计进行说明, 分别参阅见表 24.2~表 24.4。

24.6.1 flash 表

flash 表用来记录 FLV 文件的详细信息, 数据结构设计见表 24.2。

表 24.2 flash 表字段列表

字 段	类 型	字段大小	必填字段	说 明
id	自动编号	-	-	分类信息编号
bigclassid	数字	长整型	否	FLV 文件的大类编号
smallclassid	数字	长整型	否	FLV 文件的小类编号
folder	文本	200	否	FLV 文件所在的文件夹
flashname	文本	255	否	FLV 文件的名称
fsize	文本	255	否	FLV 文件大小
star	文本	255	否	FLV 文件星级
url	文本	100	否	FLV 文件所在的 URL
url2	文本	100	否	FLV 文件所在的 URL
picurl	文本	255	否	FLV 文件显示的缩微图所在的 URL
remark	备注	-	-	FLV 文件说明
username	文本	50	否	FLV 文件上传者
dateandtime	日期/时间	-	-	FLV 文件上传时间
hits	数字	长整型	否	FLV 文件的点击次数
votenum	数字	长整型	否	FLV 文件的投票次数
commentnum	数字	长整型	否	FLV 文件的评论次数

续表

字 段	类 型	字段大小	必填字段	说 明
downnum	数字	长整型	否	FLV 文件的下载次数
ishot	是/否	-	-	是否是热门 FLV 文件
isgood	是/否	-	-	是否是推荐 FLV 文件
ispassed	是/否	-	-	FLV 文件是否通过审核
geshou	文本	50	否	FLV 文件歌手

24.6.2 author 表

author 表用来记录用户信息，数据结构设计见表 24.3。

表 24.3 author 表字段列表

字 段	类 型	字段大小	必填字段	说 明
userid	自动编号	-	-	新闻文章编号
username	文本	30	否	账号
password	文本	30	否	密码
sex	文本	50	否	性别
email	文本	100	否	电子邮件
homepage	文本	100	否	个人主页
tel	文本	30	否	联系电话
oicq	文本	15	否	OICQ
icq	文本	50	否	ICQ
msn	文本	50	否	MSN
zipcode	文本	10	否	邮编
address	文本	100	否	地址
remark	文本	255	否	个人说明
question	文本	50	否	问题
answer	文本	50	否	问题答案
logins	数字	长整型	否	登录次数
dateandtime	日期/时间	-	-	注册日期
flashnum	数字	长整型	否	发布的作品数
grade	数字	长整型	否	等级
islock	是/否	-	-	是否锁定用户

24.7 页面设计

本播客系统共有 41 个文件，各个文件的具体功能说明见表 24.4。

表 24.4 播客系统文件列表

文 件 名	说 明
bottom.asp	页面底部版权信息
commentsave.asp	用户评价保存文件
config.asp	系统配置函数，以及一些公用函数
conn.asp	数据库连接文件

续表

文 件 名	说 明
dovote.asp	投票处理文件
flv_play.asp	FLV 播放文件
flv_search.asp	FLV 搜索文件
getpassword.asp	取回用户密码文件
index.asp	播客系统首页文件
l_hot.asp	页面左栏显示的热点视频文件
l_new.asp	页面左栏显示的最新发布的视频文件
l_tj.asp	页面左栏显示的推荐视频文件
l_u_vider.asp	页面左栏显示的用户控制面板
l_zuixin.asp	页面左栏显示的最新视频包含文件
left.asp	页面左栏显示的用户个人信息
login.asp	用户登录处理文件
logout.asp	用户退出登录文件
md5.asp	Md5 加密文件
r_click_top.asp	页面右栏显示的点击排行列表
r_comment.asp	页面右栏显示的推荐播客列表
r_hot.asp	页面右栏显示的热点视频列表
r_login.asp	用户登录表单
r_u_video.asp	用户视频列表
r_user.asp	页面右侧显示的用户控制面板
reeg.asp	用户注册表单
reg.asp	用户注册包含文件
reman.asp	热门视频文件
sallclass.asp	视频分类列表
showcomment.asp	显示用户评论信息
showsmall.asp	显示用户的视频信息
showuser.asp	用户控制中心
top.asp	系统顶部导航条
user_index.asp	用户上传视频列表
video.asp	视频分类列表
video_inter.asp	相关视频包含文件
video_inter_list.asp	相关视频列表文件
video_intro.asp	视频文件
video_pl.asp	视频评论
video_play.asp	视频播放文件
video_tj.asp	视频推荐

有关页面的设计可查看光盘中的实例文件。其中在页面设计中,把功能相对独立的页面分为独立的文件,然后利用`<!--#include file="文件名" -->`命令导入到需要的地方,这样能节省代码,提高工作效率,方便修改和更新。例如,首页 index.asp 就是全部利用各个功能模块文件组装起来的,代码如下:

```
<body>
```

```

<!--#include file="top.asp"-->
<%if session("username") <> "" then%>
<!--#include file="r_user.asp"-->
<%else %>
<!--#include file="r_login.asp"-->
<%end if %>
<!--#include file="r_comment.asp"-->
<!--#include file="r_click_top.asp"-->
<!--#include file="l_tj.asp"-->
<!--#include file="l_new.asp"-->
<!--#include file="l_hot.asp"-->
<!--#include file="bottom.asp"-->
</body>

```

24.8 视频发布

视频文件根据不同的分类进行显示, 这样非常方便用户从不同角度查看视频文件, 例如根据发布时间、用户点击次数、管理员是否推荐等。由于分类比较详细, 限于篇幅, 仅就其中的两个页面的实现代码进行介绍, 其他页面的实现过程基本一致, 就不再重复。

24.8.1 热门分类

热门分类主要是根据数据库中 flash 表的 ishot 字段进行筛选, 然后根据不同形式显示出来, 主要包括 l_hot.asp、reman.asp 和 r_hot.asp 文件。这些文件的操作基本相同, 即定义记录集、查询指定类型的 FLV 文件, 然后把查询的记录集重复显示出来。例如, 以 reman.asp 文件为例介绍此类分类信息显示的一般方法。

- 01** 打开 reman.asp 文件, 切换到【代码】视图, 在代码顶部导入 conn.asp 数据库连接文件和 config.asp 参数和公共函数文件。

```

<!--#include file="conn.asp"-->
<!--#include file="config.asp"-->

```

- 02** 获取各种传递参数, 其中 page 表示当前页数, utype 表示视频分类类型, 代码如下:

```

currentpage = request("page")
utype = request("utype")

```

- 03** 根据 utype 参数值分别定义不同的 SQL 字符串, 实现查询不同类别的记录集, 具体代码如下:

```

select case utype
    case "推荐作品"
        sql = "select * from flash where isgood=true and ispassed=true order by hits desc"
    case "热门作品"
        sql = "select * from flash where ishot=true and ispassed=true order by dateandtime desc, id"
    case "最新作品"
        sql = "select top 12 * from flash where ispassed=true order by dateandtime desc, id"
end select

```

- 04** 定义记录集, 利用上面的 SQL 字符串打开记录集, 并判断记录集是否为空, 如果不为空则初始化各种变量, 并循环显示记录集中字段信息。

```

'初始化变量信息
maxperpage = 8
i = 1
totalpage = 1

```

```

totalrecords = 0
strfile = "reman.asp?utype=" & utype
set rs = server.createobject("adodb.recordset") '定义记录集对象实例
rs.open sql,conn,1,1 '打开记录集
if not(rs.bof and rs.eof) then '判断记录集是否不为空
    rs.pagesize = maxperpage '定义记录集每页显示的记录数
    rs.absolutepage = currentpage '定义记录集当前显示页数
    totalpage = rs.pagecount '获取记录集的页数
    totalrecords = rs.recordcount '获取记录集的总记录数
end if %>
<% do until rs.eof or i > maxperpage '循环显示记录集中的记录
    dim picbg3
    if rs("picurl") = "" then
        picbg3 = "smallpic/no.jpg"
    else
        picbg3 = rs("picurl")
    end if %>
<div class="video">
    <ul>
        <li class="v_img"><A title="点击视频欣赏" href="video_play.asp?id=<%= rs("id")
%>" target=_blank></a></li>
        <li class="v_name"><A title="点击视频欣赏" href="video_play.asp?id=<%= rs("id")
%>" target=_blank> <%= rs("flashname") %></A></li>
    </ul>
</div>
<% rs.movenext
    i = i + 1
loop
rs.close
if totalrecords > 0 then %>
<% end if %>

```

24.8.2 视频分类显示

video.asp 文件是一个分类列表文件，该文件将根据用户在导航菜单中选择不同的分类显示所在类别的所有视频文件，页面显示效果如图 24.23 所示。



图 24.23 视频分类显示效果

实现该功能的具体操作步骤如下。

- 01** 打开 video.asp 文件, 切换到【代码】视图, 首先在代码顶部输入下面代码, 实现与数据库的连接, 同时获取各种参数函数。

```
<!--#include file="conn.asp"-->
<!--#include file="config.asp"-->
```

- 02** 获取视频分类参数, 判断该参数值是否为数值, 否则调用 mb() 函数提示指定视频分类编号, 并返回。

```
<% smallclassid=request("smallclassid")
if not isinteger(smallclassid) then
    call mb("请指定视频分类ID!", "index.asp", 1)
end if%>
```

- 03** 初始化分页系列变量, 以实现记录集分页。

```
<% currentpage=request("page") '获取当前页页数
order=request("order")
if not isinteger(currentpage) then '判断当前页数是否为整型数值
    currentpage=1
else
    currentpage=clng(currentpage)
end if
maxperpage=16 '定义每页显示的记录数
i=1
totalpage=1
totalrecords=0
strfile="video.asp?smallclassid=" & smallclassid '定义分页导航中的URL参数
%>
```

- 04** 根据分类参数值定义一个记录集, 查询该分类所在的大类名称、小类名称和大类编号。由于涉及到跨表查询, 读者应该注意 SQL 字符串的编写方法。代码如下:

```
<set rs=conn.execute("select B.bigclassname,S.smallclassname,S.bigclassid from
bigclass B inner join smallclass S on B.bigclassid=S.bigclassid where S.smallclassid=
"&smallclassid)
if not (rs.bof and rs.eof) then
    bigclassname=rs(0)
    smallclassname=rs(1)
    bigclassid=rs(2)
else
    bigclassname="不详"
    smallclassname="不详"
end if
rs.close%>
```

- 05** 根据分类编号参数值定义记录集, 查询 flash 数据表中符合条件的记录, 并对该记录集进行判断, 如果为非空, 则重复显示出来, 并进行分页处理。代码如下:

```
<set rs=server.createobject("adodb.recordset") '定义记录集对象实例
if order="hits" then '判断order参数值是否为"hits", 以分别定义SQL字符串, 实现不同记录集排序
    sql="select * from flash where smallclassid=" & smallclassid & " and ispassed=true order
by hits desc"
else
    sql="select * from flash where smallclassid=" & smallclassid & " and ispassed=true order
by dateandtime desc,id"
end if
rs.open sql,conn,1,1 '打开记录集
if not(rs.bof and rs.eof) then '判断记录集是否不为空
    rs.pagesize=maxperpage '定义记录集每页显示的记录数
    rs.absolutepage=currentpage '定义记录集当前显示的页数
    totalpage=rs.pagecount '获取记录集的页数
```



```

        strTemp=strTemp & "</select>"
    end if
    strTemp=strTemp & "</td></tr></table>"
    response.write strTemp
end sub%>

```

24.9 视频播放

视频播放的设计思路其实比较简单,它主要根据其他页面传递过来的 FLV 播放文件的编号,从数据表 flash 中查询符合条件的记录,并进行判断。如果记录集为空,则提示错误;否则获取该记录的 url 字段信息,并把该记录的 FLV 文件的 URL 传递给包含的 FLV_play.asp 文件中插入的流媒体对象。由插入的<object>对象,并配合 Dreamweaver CS5 自动生成的 Scripts/AC_RunActiveContent.js 脚本文件和 FLVPlayer_Streaming.swf 文件,实现与 FMS 服务器的通信;在 FMS 服务器端 main.asc 文件的配合下,把指定服务器端具体的某个 FLV 文件以流媒体的形式向客户端发送。客户端实时接收这些信息,并不间断地利用<object>对象播放指定的 FLV 文件。

整个过程看似简单,但如果查看源代码,实际上还是比较复杂的,好在 Dreamweaver CS5 为读者提供了可视化操作方法,把所有代码编写过程全部隐藏起来。这对于初学者来说,确实是设计和开发视频网站的好途径。

具体实现方法如下。

- 01 打开 video_play.asp 文件,在该页面中设计一个 FLV 文件播放界面,设计效果如图 24.24 所示。
- 02 用语句<!--#include file="FLV_play.asp"-->把 FLV_play.asp 文件包含到界面中,如图 24.25 所示。

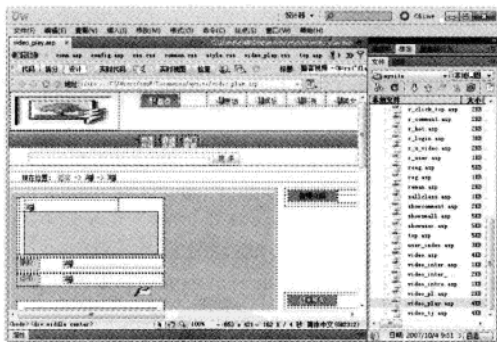


图 24.24 视频分类显示效果

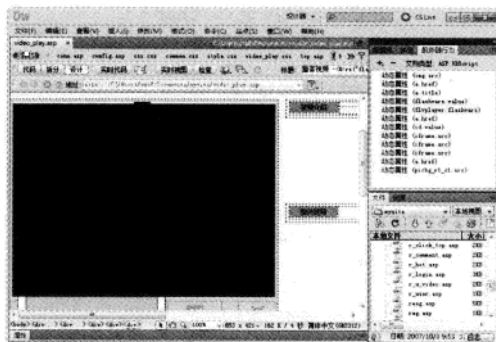


图 24.25 包含 FLV_play.asp 文件

- 03 切换到【代码】视图,在代码顶部输入下面代码。

```

<!--#include file="conn.asp"-->
<!--#include file="config.asp"-->
<%
dim id,rss,bigclassname,smallclassname,remark
id =request("id")'获取id编号参数值
if not isinteger(id) then'判断该参数值是否为整型数值
    call mb("对不起,请指定要观看的视频!", "index.asp",1)'如果不是数值,则提示错误并返回
end if

```

```

set rs=server.createobject("adodb.recordset")'定义记录集对象实例
sql="select * from flash where id=&id"定义查询指定编号的记录
rs.open sql,conn,1,3'打开记录集
if rs.bof and rs.eof then'判断该记录集是否存在记录
    call mb("对不起,没有找到相关的视频!", "index.asp", 1)'如果不存在记录,则提示错误并返回
else
    rs("hits")=rs("hits")+1'如果存在该记录,则为该记录中的hits字段累加1
    rs.update'更新记录集,实现记录集的值写入数据库
end if
set rss=conn.execute("select B.bigclassname,S.smallclassname from bigclass B inner join
smallclass S on S.bigclassid=B.bigclassid where S.smallclassid="&rs("smallclassid"))'定义
另一个记录集,查询该记录的大类和小类名称
if not(rss.bof and rss.eof) then'如果该记录集为非空,则获取大类和小类的名称
    bigclassname=rss(0)
    smallclassname=rss(1)
end if
set rss=nothing'注销记录集rss

dim url
if rs("folder")<>"" then '判断记录集中folder字段是否为空,否则把该字段值增加到URL字符串中
    url=rs("folder")&"/"&rs("url")
else
    url=rs("url")
end if
%>

```

04 把rs记录集中相关字段绑定到页面中,实现信息显示和视频播放,具体代码不再显示,读者可以参考光盘实例。视频播放效果如图 24.26 所示。

24.10 实践总结

在本章中重点介绍了 FMS 服务器的搭建和管理、Web 服务器与 FMS 服务器之间的联系和通信。读者通过本实例的学习,可以初步认识和掌握播客系统的核心功能实现过程,但要开发出功能更加强大和实用的播客系统,还需要读者结合前面介绍的基础知识以及实例模块来扩展本系统功能。

限于篇幅,本实例仅演示了视频分类和播放模块,以及用户控制模块的基本功能和框架,有关播客的前台管理和后台管理功能没有涉及。播客程序实际上也是一个非常复杂的 Web 应用系统,它涉及前台视频分类显示和视频播放、前台用户控制,以及后台系统管理等不同的大型模块。在用户控制模块中还可以包括复杂的视频上传和在线视频录制功能、用户信息管理等,而后台管理就更加复杂了,管理的内容可以包含系统管理、用户管理、视频管理和其他信息管理,以及功能管理和功能扩展等。



图 24.26 视频播放效果