

网络技术全息丛书

Dreamweaver UltraDev 4 网站开发实务

俞俊平 余安萍 俞俊军 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本书以清晰的思路、简练的文笔、丰富的示例、直观的图形和详尽的说明并配以随书光盘，全面介绍了使用 Dreamweaver UltraDev 4 快速开发 Web 应用的全过程。全书共分为 4 个部分，从基础、应用、实例和等级扩展 4 个层次系统地揭示了 Dreamweaver UltraDev 4 的核心与精华，使读者能在最短的时间内掌握 Dreamweaver UltraDev 4 的特色，设计出功能强大、实用的 Web 应用程序。

本书主要面向网站建设者，开发者和爱好者，也适用于有一定基础的计算机用户参考阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，翻版必究。

图书在版编目(CIP)数据

Dreamweaver UltraDev 4 网站开发实务/俞俊平，余安萍，俞俊军编著. -北京：电子工业出版社，2001.5

(网络技术全息丛书)

ISBN 7-5053-6653-X

I.D... II.①俞...②余...③俞... III.主页制作-应用软件，Dreamweaver UltraDev 4

IV.TP 393.092

中国版本图书馆 CIP 数据核字(2001)第 25448 号

丛 书 名：网络技术全息丛书

书 名：Dreamweaver UltraDev 4 网站开发实务

编 著 者：俞俊平 余安萍 俞俊军

责任编辑：张 毅 zhangyi@phei.com.cn

印 刷 者：

装 订 者：

出版发行：电子工业出版社 URL：<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×980 1/16 印张：27.25 字数：610 千字

版 次：2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

书 号：ISBN 7-5053-6653-X

TP·3705

印 数：6000 册 定价：39.00 元

凡购买电子工业出版社的图书如有缺页、倒页、脱页者，请向购买书店调换；若书店售缺，请与本社发行部联系调换。电话：68279077。

前 言

随着 Web 技术的广泛应用，掌握开发 Web 应用的技能也就成了网站开发中的一种必然趋势。以前，开发 Web 应用是非常复杂的，我们不得不借助于一些 Web 服务器上所带的可编程 API，并通过 C 或 Java 等语言将这些 API 增加到 Web 站点中。而现在，随着 ASP、JSP 以及 CFML 等技术的出现，开发 Web 应用的过程简单了许多，并且功能也越来越复杂、越来越强大。同时，广大 HTML 开发者也面临着如何更快、更方便地开发出 Web 应用的瓶颈问题。

Dreamweaver UltraDev(以下全书简称为 DW UltraDev)是第一款能够让开发人员、编程人员和设计人员在多种服务器平台上，可视化地创建和编辑数据驱动 Web 应用的软件产品。这意味着开发者只要学习一种设计环境即可，而不用再去考虑客户端的输出需求。这就使得它成为了能够满足当今各种应用开发需要的理想工具。

本书以简单实用、易学习、易掌握为宗旨；以清晰的思路、简练的文笔、丰富的示例、直观的图形和详尽的文档并配以随书光盘全面讲述了使用 DW UltraDev 4 快速开发 Web 应用的全过程；循序渐进地引导读者轻松地学习创建网站和开发 Web 应用的技能；让读者在最短的时间内掌握 DW UltraDev 4 的特色，设计出功能强大、实用的 Web 应用程序。

全书分为四大部分，第一部分“认识 DW UltraDev 4”，包括第 1 章到第 3 章，这是本书的引子，也是本书的基础，主要介绍了 DW UltraDev 4 的特点、应用与安装，以及 DW UltraDev 4 的界面基础和 DW UltraDev 4 的简单使用。第二部分“使用 DW UltraDev 4 开发 Web 应用”，包括第 4 章到第 10 章，这是本书的核心，主要讲述了 DW UltraDev 4 的文档、站点和页面处理基础，并按照 Web 应用的开发流程依次介绍了 DW UltraDev 4 的功能。第三部分“DW UltraDev 4 应用实例”，包括第 11 章到第 13 章，主要介绍了三个不同的综合示例与经典应用，目的是提高读者使用 DW UltraDev 4 的实际动手能力。第四部分“DW UltraDev 4 高级应用与扩展”，包括第 14 章到第 17 章，这是本书的关键和升华，主要讲述了插件的安装与使用，以及定制 DW UltraDev 4 的界面和扩展 DW UltraDev 4 功能的方法。

本书附有一张配套光盘，其中包含了书中所有示例的全部源代码和素材，所有代码都经过严格调试，完全能正常运行。除此之外，为了便于读者开发网站，光盘中还包含了一些 DW UltraDev 的常用插件，以及 DW UltraDev 4 的 30 天试用版软件和其他相关共享软件。

本书由飞雪漫工作室的俞俊平、余安萍和俞俊军主笔，另外余显明、王正梅、余小多、华军、赵静、高环等在本书的编写过程中也给予了大力帮助。

尽管在编写过程中，我们进行了认真细致的审改工作，但由于时间仓促，加上作者水平有限，书中难免存在不足之处，恳请读者批评指正。

作 者

目 录

第一部分 认识 DW UltraDev 4

第 1 章	DW UltraDev 4 概述	3
1.1	DW UltraDev 4 火爆登场	3
1.1.1	Web 应用的发展	3
1.1.2	DW UltraDev 的出现	4
1.2	DW UltraDev 4 的特点	5
1.2.1	DW UltraDev 4 的强大功能	5
1.2.2	DW UltraDev 的主要应用	14
1.2.3	DW UltraDev 和其他同类产品的比较	15
1.3	安装 DW UltraDev 4	17
第 2 章	DW UltraDev 4 界面基础	22
2.1	启动 DW UltraDev 4	22
2.2	认识 DW UltraDev 4 的窗口结构	25
2.3	DW UltraDev 4 的主菜单	28
2.3.1	File 菜单	28
2.3.2	Edit 菜单	29
2.3.3	View 菜单	31
2.3.4	Insert 菜单	33
2.3.5	Modify 菜单	35
2.3.6	Text 菜单	38
2.3.7	Commands 菜单	39
2.3.8	Site 菜单	40
2.3.9	Window 菜单	41
2.3.10	Help 菜单	42
2.4	基本面板功能	43
2.4.1	启动面板	44
2.4.2	对象面板	46
2.4.3	属性面板	51

2.5	控制浮动面板	52
2.6	使用其他界面元素和工具	55
2.6.1	使用状态栏	55
2.6.2	使用工具栏	57
2.6.3	使用快捷菜单	58
2.6.4	使用网格	59
2.6.5	使用标尺	60
2.7	设置 DW UltraDev 4 的界面属性	61
2.7.1	打开属性设置窗口	61
2.7.2	设置 General 属性	62
2.7.3	设置 Panels 属性	63
2.7.4	设置 Fonts / Encoding 属性	65
2.7.5	设置 Status Bar 属性	66
第 3 章	体验 DW UltraDev 4	68
3.1	定义本地站点	68
3.1.1	定义本地站点文件夹	68
3.1.2	确定服务器端技术	69
3.2	定义远程站点	70
3.2.1	创建发布目录	70
3.2.2	定义远端站点目录	70
3.2.3	配置站点的 URL 前缀	73
3.3	创建数据库链接	74
3.3.1	在已设置 DSN 的 Web 服务器上创建数据库链接	76
3.3.2	在未设置 DSN 的 Web 服务器上创建数据库链接	78
3.4	创建 ASP 示例页面	79
3.4.1	创建简单的查询/结果页面	79
3.4.2	创建细节页面	87
3.4.3	加入记录浏览功能	91

第二部分 使用 DW UltraDev 4 开发 Web 应用

第 4 章	站点和文档初步	95
4.1	站点和文档概述	95
4.1.1	本地计算机和 Internet 服务器	96

4.1.2	本地站点和远程站点	96
4.1.3	上传和下载	97
4.2	站点规划	98
4.2.1	网页的设计流程	98
4.2.2	规划站点结构	98
4.2.3	规划站点的浏览体系	100
4.2.4	规划模板和库的使用	101
4.3	创建本地站点	101
4.4	站点操作	104
4.4.1	打开站点	105
4.4.2	编辑站点	105
4.4.3	复制站点	106
4.4.4	删除站点	106
4.5	操作站点文件	107
4.5.1	创建文件和文件夹	107
4.5.2	移动和复制文件或文件夹	108
4.5.3	编辑和删除文件	109
4.5.4	设置站点主页	110
4.6	使用站点窗口	110
4.6.1	查看本地站点	110
4.6.2	改变站点显示	113
4.7	创建和编辑文档	115
4.7.1	创建空白文档	116
4.7.2	打开现有文档	116
4.7.3	基于模板创建文档	116
4.7.4	存储文档	117
4.7.5	关闭文档	117
4.8	添加文本和插入对象	118
4.8.1	添加文本和常用对象	118
4.8.2	插入特殊字符	118
4.8.3	插入水平线	120
4.9	在文档窗口中选择元素	121
4.9.1	选择元素的方法	121
4.9.2	选择不可见元素	122
4.10	设置文档属性	124

4.11	使用历史面板	125
4.11.1	关于历史面板	125
4.11.2	撤销操作	127
4.11.3	自动化任务	127
4.12	查看和编辑网页头部内容	132
4.12.1	显示头部元素	132
4.12.2	编辑网页的头部内容	132
4.12.3	认识头部元素	134
第 5 章	编辑 HTML 页面	137
5.1	Code Inspector	137
5.2	编辑 HTML 标记	141
5.2.1	插入 HTML 模式	145
5.2.2	编辑 Tag 模式	145
5.2.3	封装 Tag 模式	146
5.2.4	设置 Quick Tag 编辑器的属性	147
5.3	设置 HTML 格式特性	147
5.3.1	HTML 重写属性	148
5.3.2	HTML 格式化属性	149
5.3.3	HTML 颜色属性	152
5.4	净化 HTML 源代码	153
5.4.1	净化 HTML 源代码	153
5.4.2	净化 Microsoft word HTML 源代码	155
5.5	外部 HTML 编辑器	157
5.6	其他编辑功能	158
5.6.1	纠正无效标记	158
5.6.2	插入脚本	158
5.6.3	插入注释	159
第 6 章	UltraDev 开发 Web 应用基础	161
6.1	Web 应用开发流程	161
6.2	开发前的准备工作	162
6.2.1	定义站点	162
6.2.2	配置 DW UltraDev	163
6.2.3	创建数据库链接	165
第 7 章	创建记录集	172
7.1	记录集概述	172

7.2	定义记录集	173
7.2.1	不使用 SQL 定义记录集	173
7.2.2	使用 SQL 定义记录集	176
7.2.3	为结果页面定义记录集	178
7.3	操作记录集	180
7.3.1	复制记录集	180
7.3.2	调用存储过程	181
7.3.3	创建存储过程对象	181
7.3.4	编辑和删除记录集	182
7.4	编写 SQL 查询语句	183
7.4.1	编写最简单的 SQL 语句	183
7.4.2	包含整个数据库表	183
7.4.3	限制字段的数目	184
7.4.4	限制记录的数目	184
7.4.5	对记录进行排序	185
第 8 章	创建动态网页	186
8.1	在网页中加入动态内容	186
8.1.1	加入动态内容的步骤	186
8.1.2	定义数据源	187
8.1.3	使文本动态化	189
8.1.4	使图片动态化	191
8.1.5	使表单动态化	192
8.1.6	使 HTML 属性动态化	195
8.1.7	使其他对象动态化	197
8.2	创建不同类型的动态网页	198
8.2.1	创建搜索网页	198
8.2.2	创建结果网页	198
8.2.3	创建细节网页	199
8.2.4	创建记录编辑网页	200
第 9 章	激活动态网页	201
9.1	概述	201
9.2	显示多条记录	202
9.2.1	创建 Repeat Region 服务器行为	202
9.2.2	创建记录计数	203
9.3	在记录间移动	204

9.3.1	创建记录浏览链接	205
9.3.2	移动到特定记录	206
9.3.3	增强浏览功能	207
9.4	在页面之间传递信息	209
9.4.1	获取 HTML 表单中的数据	209
9.4.2	定位到细节页面	210
9.4.3	定位到相关页面	211
9.5	编辑记录	211
9.5.1	在数据库中插入记录	211
9.5.2	在数据库中更新记录	212
9.5.3	在数据库中删除记录	214
第 10 章	编辑和调试动态网页	216
10.1	动态网页的基本编辑操作	216
10.1.1	改变记录集的内容	216
10.1.2	替换和删除动态内容	217
10.1.3	修改服务器行为	218
10.2	Live Data 环境下的编辑操作	219
10.2.1	在 Live Data 窗口中编辑页面	219
10.2.2	使用 Live Data Settings 对话框	222
10.3	测试 Web 应用的链接	223
10.3.1	配置 DW UltraDev	223
10.3.2	在浏览器中进行预览	224
10.4	编辑源代码	225
10.5	编辑和创建数据格式	225
10.5.1	编辑数据格式	226
10.5.2	创建新的数据格式	227

第三部分 DW UltraDev 4 应用实例

第 11 章	Web 应用综合示例	231
11.1	开发 Web 应用前的准备工作	231
11.1.1	创建 DSN	232
11.1.2	定义站点	234
11.1.3	创建数据库连接	235

11.2	在页面中增加动态内容	238
11.2.1	为页面增加一个记录集	238
11.2.2	在页面中增加动态数据	240
11.2.3	在 Live Data 模式下编辑页面	242
11.2.4	在页面中显示多个记录	243
11.2.5	浏览记录	244
11.3	进一步增加动态页面的功能	246
11.3.1	增强浏览功能	246
11.3.2	处理细节页面	247
11.3.3	定义一个过滤的记录集	248
11.3.4	在细节页面中增加动态内容	250
11.3.5	往数据库中增加新记录	252
第 12 章	创建留言簿	255
12.1	开发留言簿前的准备工作	255
12.1.1	设计基本网页	256
12.1.2	创建数据库	258
12.1.3	创建 DSN	261
12.1.4	定义站点	261
12.1.5	创建数据库链接	263
12.2	完成填写留言页面	265
12.3	完成浏览留言页面	266
12.3.1	定义留言记录集	266
12.3.2	显示浏览留言页面	267
12.3.3	显示多条留言记录	270
12.3.4	增加浏览链接	271
12.3.5	增强浏览功能	271
第 13 章	创建查询系统	272
13.1	设计准备工作	272
13.1.1	新建站点	273
13.1.2	创建初始网页及数据库	275
13.2	定义 DSN 和数据库连接	279
13.2.1	定义 DSN	279
13.2.2	创建数据库连接	280
13.3	添加动态内容	281
13.3.1	创建判断页	281

13.3.2	创建结果页	284
13.3.3	查询系统测试	286

第四部分 DW UltraDev 4 高级应用与扩展

第 14 章	插件的安装与使用	291
14.1	插件概述	291
14.2	安装插件	292
14.2.1	安装插件	292
14.2.2	测试插件	295
14.3	管理插件	297
14.3.1	插件管理器的主要功能	297
14.3.2	对插件进行排序	297
14.3.3	激活或禁止插件	298
14.3.4	删除插件	299
14.4	一个声音插件的使用	300
14.4.1	安装声音插件	300
14.4.2	使用声音插件	301
第 15 章	定制 DW UltraDev 4 界面	302
15.1	定制 DW UltraDev 4 界面概述	302
15.2	定制对象面板	303
15.3	创建一个简单对象	305
15.4	改变缺省文件类型	306
15.5	定制 DW UltraDev 菜单	307
15.5.1	编辑 Commands 菜单	307
15.5.2	菜单语法	308
15.5.3	菜单重排	308
15.5.4	菜单项语法	309
15.5.5	键盘快捷键	310
15.5.6	改变菜单项名字	311
15.6	定制对话框界面	311
15.7	编辑 HTML 源码格式化配置文件	312
15.8	浏览器配置文件	313
15.9	扩展 DW UltraDev	316

15.10	第三方标记的定制	317
15.10.1	第三方标记的定制	317
15.10.2	文档窗口显示定制标记的方法	319
15.10.3	避免改写第三方标记	320
第 16 章	DW UltraDev 4 的 DOM 模型	323
16.1	DOM 概念	324
16.2	DW UltraDev 4 的 DOM	324
16.2.1	DW UltraDev 对象及其特性	326
16.2.2	DOM	327
16.2.3	文档对象的特性和方法	327
16.2.4	HTML 标记对象的特性和方法	328
16.2.5	文本对象的特性和方法	329
16.2.6	注释对象的特性和方法	330
16.3	树控制	330
16.4	JavaScript 在扩展中的工作方式	332
16.4.1	DW UltraDev 4 处理扩展的方式	332
16.4.2	在启动和关闭时运行 scripts	333
第 17 章	DW UltraDev 4 API 及其使用	335
17.1	DW UltraDev 4 的 JavaScript API	335
17.1.1	API 函数概述	336
17.1.2	最重要的函数	337
17.1.3	Behavior 函数	337
17.1.4	剪贴板函数	344
17.1.5	命令函数	346
17.1.6	转换函数	347
17.1.7	CSS 风格函数	347
17.1.8	外部应用函数	350
17.1.9	文件处理函数	354
17.1.10	Find/Replace 函数	360
17.1.11	Frame 和 frameset 函数	364
17.1.12	通用编辑函数	364
17.1.13	全局应用函数	373
17.1.14	全局文档函数	375
17.1.15	History 函数	376
17.1.16	HTML 风格函数	381

17.1.17	层和影像地图函数	383
17.1.18	窗口函数	385
17.1.19	Enabler 函数	388
17.2	文件 I/O API	402
17.3	Design Notes API	404
17.3.1	Design Notes JavaScript API	405
17.3.2	Design Notes C API	406
17.4	Database API	406
17.5	Commands API	409
17.6	Data Sources API	410
17.7	Server Behaviors API	411
17.8	其他 API	412
17.8.1	Fireworks Integration API	412
17.8.2	HTTP API	412
17.8.3	C-Level 扩展 API	413
17.8.4	Objects API	415
17.8.5	Menu Commands API	415
17.8.6	Property Inspectors API	416
17.8.7	Floating Palettes API	416
17.8.8	Behaviors API	416
17.8.9	Data Translators API	417
17.8.10	Server Models API	417

第 一 部 分

认识 DW UltraDev 4

- ◆ 第 1 章 DW UltraDev 4 概述
 - ◆ 第 2 章 DW UltraDev 4 界面基础
 - ◆ 第 3 章 体验 DW UltraDev 4
-

第 1 章 DW UltraDev 4 概述

本章概要：

- ☒ DW UltraDev 4 火爆登场
- ☒ DW UltraDev 4 的特点
- ☒ 安装 DW UltraDev 4

Macromedia 公司的产品可谓独具匠心，风靡全球的网页制作三剑客(Dreamweaver、Flash 和 Fireworks) 才上市不久，一个全新概念的新产品 DW UltraDev 又上市了。其强大的功能震撼了业界人士，同时也深信它很快就会成为替代 Dreamweaver 的新一代网页制作领域的佼佼者。本章我们就来看一看它的‘庐山真面目’。

1.1 DW UltraDev 4 火爆登场

1.1.1 Web 应用的发展

在介绍 DW UltraDev 4 之前，我们必须首先了解 Web 应用的概念。目前，“Web 应用”这个术语被越来越多的媒体和网站所引用，那么，Web 应用究竟包含了什么样的内容呢？

简单地说，一个 Web 应用其实就是一些网页的集合。只不过这些网页不同于一般的静态网页，它们包含了服务器逻辑并在服务器端运行，从而动态地产生结果返回到客户的浏览器端。Web 应用通常可以用来取代传统的客户机/服务器软件。典型的 Web 应用包括聊天室、BBS、在线商店等。

随着 Web 技术的广泛应用，掌握开发 Web 应用的技能也就成了一种必然趋势。以前，开发 Web 应用是非常复杂的，我们不得不借助于一些 Web 服务器(如 Microsoft Internet Information Server、Netscape Enterprise Server 等)上所带的可编程 API，并通过 C 或 Java 等语言将这些 API 增加到 Web 站点中，因此对 Web 应用开发者的要求较高。

现在,随着 Microsoft 公司的 Active Server Page(ASP)、Sun 公司的 JavaServer Page(JSP) 以及 Allaire 公司的 ColdFusion Markup Language(CFML) 等技术的出现,开发 Web 应用的过程简单了许多。这些新技术都是基于传统的 HTML 发展起来的,Web 开发者只需借助一些基于服务器端的脚本语句就可完成 Web 应用的设计。正因为如此,现在的各种 Web 应用如雨后春笋般地涌现出来,并且功能也越来越复杂、越来越强大。但是广大 HTML 开发者也同时面临着一个越来越紧迫的问题,即如何尽可能快、尽可能方便地开发出 Web 应用。

1.1.2 DW UltraDev 的出现

幸运的是,Macromedia 公司发布了它的快速 Web 应用开发可视化解决方案——DW UltraDev 1.0,并在几个月之后很快又推出了 DW UltraDev 4,其版本号一下子从 1.0 直接跨越到 4.0,由此也可以看到 DW UltraDev 的神速发展。

DW UltraDev 是第一款能够让开发人员、编程人员和设计人员在多种服务器平台上,可视化地创建和编辑数据驱动 Web 应用的软件产品。这意味着开发者只要学习一种设计环境即可,而不用再去考虑客户端的输出需求。

Macromedia 公司首席技术官 Kevin Lynch 说:“在设计时,可视化动态服务器端内容将会给人们开发 Web 应用的方法带来革命性的变化。”DW UltraDev 将会在创建跨数据库、应用服务器和操作系统的 Web 应用时,给开发人员带来更高的工作效率。

DW UltraDev 是一款基于 Dreamweaver 构架的新一代产品。如果读者使用过 Dreamweaver,就会觉得它的界面和工作环境是非常熟悉的。如果以前读者是用手工或者另外的方案进行 Web 应用开发的,那么,在使用了 DW UltraDev 后,就会觉得开发 Web 应用非常容易而且效率很高。

DW UltraDev 1.0 除了自身特有的功能以外,还提供了 Dreamweaver 3.0 的全部功能。而 DW UltraDev 4 则包含了 Dreamweaver 4.0 的全部功能。DW UltraDev 采用了 Dreamweaver 中的往返 HTML(Roundtrip HTML)、往返 XML(Roundtrip XML) 和往返服务器标记 Roundtrip Server Markup),以保持应用程序源代码的完整性。DW UltraDev 1.0 的操作界面与 Dreamweaver 3.0 极为相似,而 DW UltraDev 4 的操作界面与 Dreamweaver 4.0 也极为相似(这大概也是 Macromedia 公司急于把 DW UltraDev 从 1.0 直接升级到 4.0 的原因之一,即,将 DW UltraDev 和 Dreamweaver 的版本号统一起来)。开发人员可以利用它与 Macromedia Fireworks、Flash 和 Shockwave 紧密集成的诸多优点,也可以使用 Dreamweaver 的可扩展架构来扩展和定制 Web 应用。

DW UltraDev 能使开发者在主流服务器平台上创建 Web 应用,这些平台包括用于创建 ASP 应用的 IIS(Microsoft Internet Information Server)、创建 CFML 应用的 Allaire ColdFusion,以及创建 JSP 应用的 IBM WebSphere 和 iPlanet Web Server、Enterprise Edition。

DW UltraDev 也可以与 ODBC、ADO 或 JDBC 数据库源链接，或通过 COM 对象和 JavaBeans 与传统数据库系统链接。

严格地说，DW UltraDev 是 Macromedia Dreamweaver 和 Macromedia Drumbeat 两者综合起来的完全换代产品(Drumbeat 也是由 Macromedia 公司出品的另外一个知名的开发程序，用来开发 ASP 及 JSP)。

1.2 DW UltraDev 4 的特点

1.2.1 DW UltraDev 4 的强大功能

1. 支持 ASP、JSP 和 CFML

DW UltraDev 是一个高度集成的可视化设计工具，主要应用于驱动数据库的 Web 应用程序，如电子商务、个性化处理、动态创建网页等。它可大大加速网络时代电子交易应用中的项目交付以及从网页和表单的动态生成到企业级的解决方案，如电子商店、库存管理系统和企业内部局域网的数据库应用。DW UltraDev 可以生成采用 ASP、JSP 和 CFML 的应用程序，能满足当今网站建设中的应用开发的各种需要并可在一个单一的开发环境中，建立 ASP、JSP 和 Cold Fusion 网站，如图 1.1 所示。

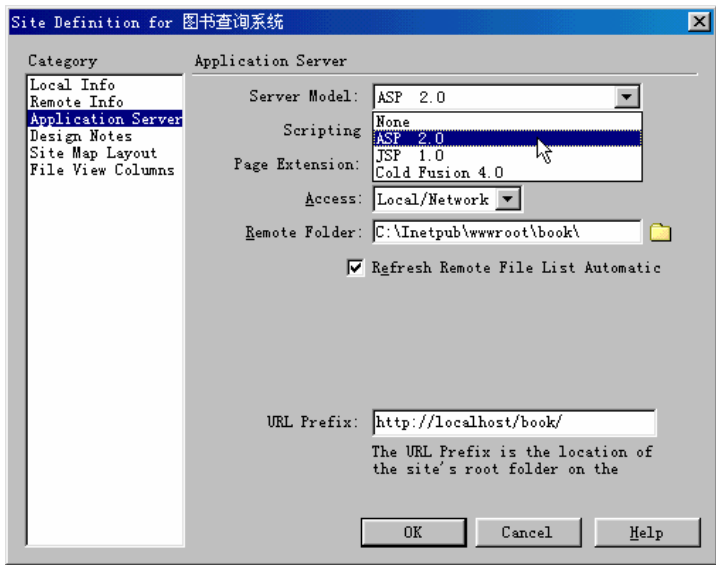
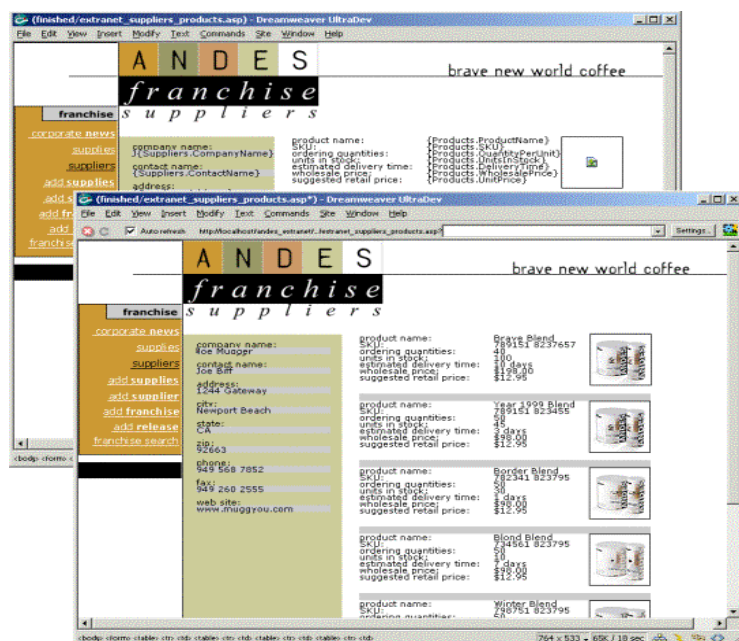


图 1-1
支持 ASP、
JSP 和 CFML

2. 实时数据预览

DW UltraDev 4 的实时数据流编辑功能使用户能够在编程环境中实时预览可编辑数据的 Web 应用情况。应用开发人员可以检查数据库、生成查询和创建应用逻辑,然后使用实时数据预览(Live Data Preview)查看动态内容。这个独特功能减少了在非可视化开发环境与 Web 浏览器之间来回切换以查看设计或测试功能所需花费的时间。因此,在处理服务器数据的同时也可以继续进行页面设计。由于它所具有的服务器行为库,使它具有最高的效率,很容易地生成这些能显示、浏览和更新数据库的内容的网页,如图 1-2 所示。

图 1-2
实时数据
预览功能



3. 服务器行为

使用服务器行为库,可以帮助用户轻松创建用于显示、浏览和更新数据库内容的页面,极大地提高工作效率。图 1-3 所示的就是一个服务器行为窗口。

4. 数据绑定检查器

有了数据绑定检查器,通过拖放操作即可把页面中的对象绑定到动态内容中。无需编写代码就可以插入动态文本、图像或设置列表、复选框、输入域和其他 HTML 标记的属性值。在可视化视图中工作时,用户可以使用任何方式格式化数据,如图 1-4 所示。

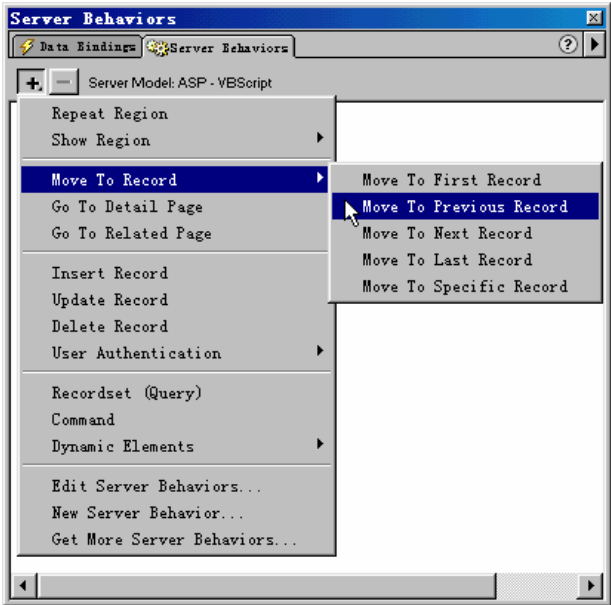


图 1-3
服务器行为窗口

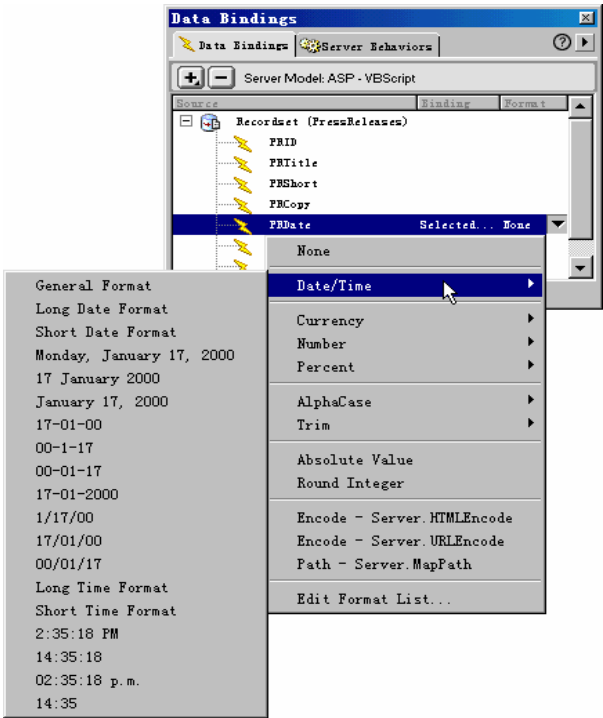
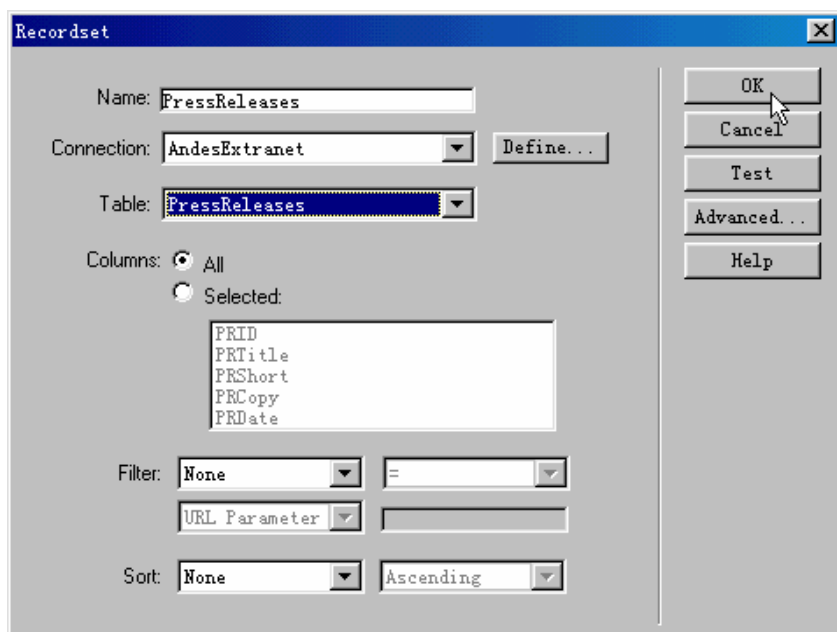


图 1-4
数据绑定检查器

5. 数据库对象

使用数据库对象，无需了解或编写 SQL(结构化查询语言)，就可以即时添加链接数据库所需的所有代码，如图 1-5 所示。

图 1-5
数据库对象



6. 查询编辑器

使用查询编辑器可以方便地浏览数据库中的表、视图和存储过程，快速测试 SQL，确保得到预期结果。在查询编辑器中，也可以编辑自己的 SQL，或从自己喜爱的 SQL 编辑环境中导入，如图 1-6 所示。

7. 往返 HTML、快速标记编辑器和往返服务器标记

采用往返 HTML(Roundtrip HTML)、快速标记编辑器(Quick Tag Editor) 和往返服务器标记(Roundtrip Server Markup) 技术，可以在不牺牲对代码的控制的前提下提高应用开发速度。在可视环境中工作的同时，也能保持对代码的完全访问。在一个视图中所作的修改，可以立即在另一个视图中反映出来，如图 1-7 所示。

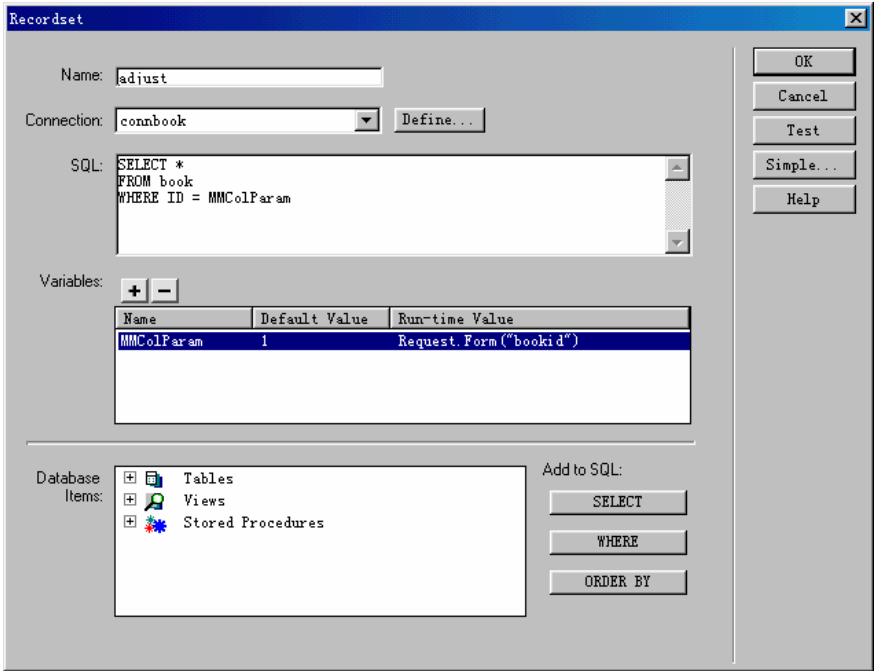


图 1-6
查询编辑器

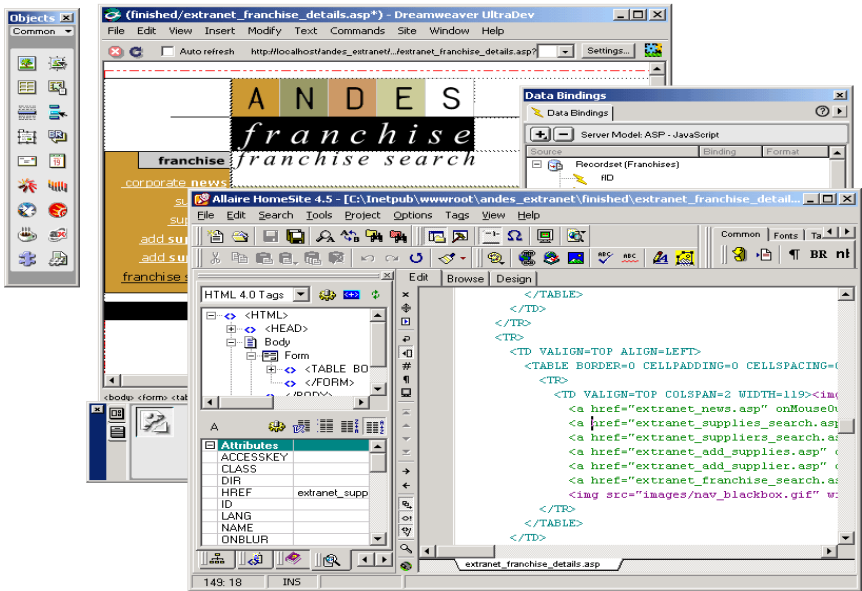


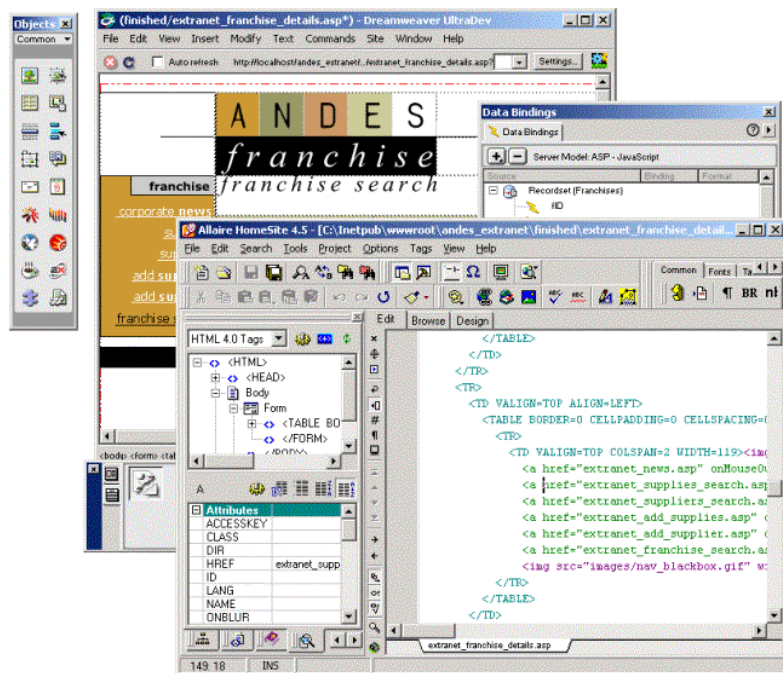
图 1-7
往返HTML、
快速标记编
辑器和往返
服务器标记

8. 集成文本编辑器

UltraDev 的商务完全版包含了紧密集成的 HomeSite (Windows) 和 BBEdit 试用版 (Macintosh), 使用户能够完全控制和访问代码, 从而在 Web 网站中轻松享受桌面发布级别的控制。集成文本编辑器还提供字符和段落样式与带有 HTML 新样式着色文字 (即, 使用各种不同的颜色表示具有不同功能的文字) 的快速链接。

另外, CSS 支持让用户可以很轻松地为一个网站定制字符和段落的格式, 这样就给网页设计带来了无可比拟的灵活性和可控性。HTML 可编辑表单提供了解决表格中各种元素格式的方案。图 1-8 中所示的就是集成在 UltraDev 中的 HomeSite 编辑器。

图 1-8
集成在
UltraDev
中的
HomeSite
编辑器



9. 干净的代码

UltraDev 生成的代码如同手工编写的代码一样干净, 没有垃圾! 与其他同类产品不同, UltraDev 提供的服务器对象和脚本不使用、不请求上传额外的文件到服务器。当把 UltraDev 的对象和脚本添加到页面时, 可以在源码视图中看到所有的代码。这样, 就不必担心遗漏额外的嵌入文件, 如图 1-9 所示。

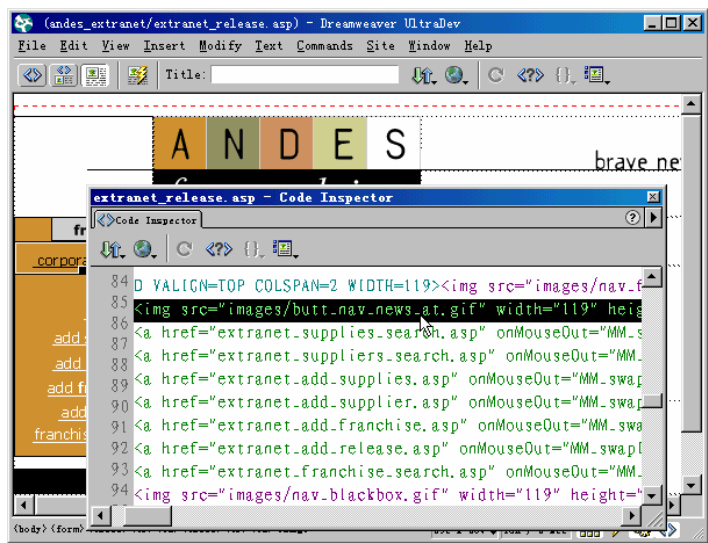


图 1-9
DW Ultra-
Dev 生成
的是干净
的代码

10. HTML 样式、CSS 和 HTML 表格编辑

在 HTML 样式、CSS 和 HTML 表格编辑方面，对于网站文本的控制，用户完全可以享受到桌面出版的水平。使用新的 HTML 样式面板，可以对文本快速应用字符和段落的样式组合。对层叠样式表（CSS）的支持，能够轻易地为一个站点配置字符和段落级样式，而且，整个开发团队均可共享这些层叠样式表。在 HTML 样式和 CSS 样式之间的选择，具有相当大的灵活性和可控性。复杂的 HTML 表格编辑功能，能够快速设置单元格、行和列及其离散选择的格式，能够快速地对表格中的相关元素，就像使用 Microsoft Word 或 Excel 所能感觉到的一样，节约了检索和格式化表格的时间。预定义和可编辑的表格格式可以让用户迅速构建非常独特的表格样式。图 1-10 中所示的就是各种不同的 HTML 样式。

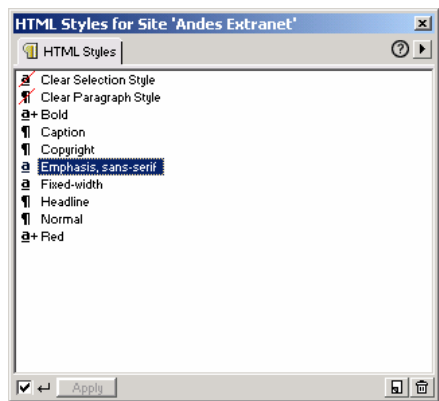
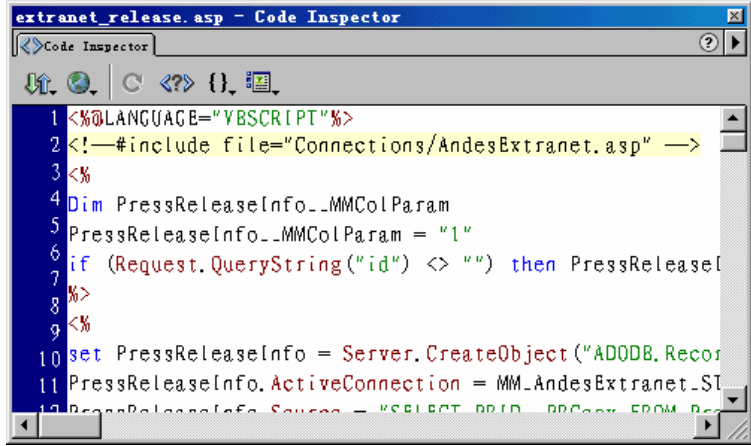


图 1-10
各种不同
的 HTML
样式

11. 可定制的工作环境

使用 JavaScript 和 XML 可以定制 UltraDev 的工作环境，如修改菜单、命令、对象和 HTML 着色等，大大提高了工作效率，如图 1-11 所示。

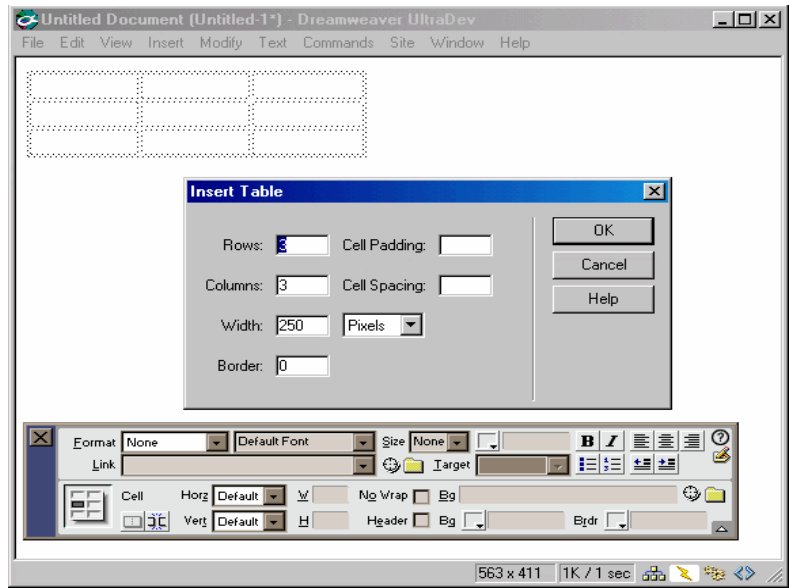
图 1-11
可定制的
工作环境



12. 动态数据格式化

无需手工编写代码，即可快速地把货币和日期格式、URL 编码等应用到数据库内容和服务器表达式中，如图 1-12 所示。

图 1-12
动态数据
格式化



13. 支持工业标准的应用服务器

支持使用 Microsoft IIS、Allaire ColdFusion、IBM WebSphere、BEA WebLogic 和 Netscape Enterprise Server 等工业标准服务器创建站点，如图 1-13 所示。并可轻易地扩展 UltraDev 来支持其他类型的服务器，如 PHP 和 Tango。



图 1-13
支持工业
标准的应
用服务器

14. COM 和 JavaBean 扩展能力

支持使用标准应用服务器、COM 对象或 JavaBeans，创建可以与后端数据库系统链接的站点。通过 Dreamweaver 提供的功能强大的扩展 API，可以方便地把自己公司的对象添加到 UltraDev 中。

15. 广泛的数据库支持

使用 ODBC、JDBC、ADO 或本地驱动程序，可以准确地指定站点怎样与数据库链接，以获得最佳性能。可以配置标准数据库与 Oracle、SQL Server、Informix 和 Sybase 等标准工业服务器的链接。DW UltraDev 4 还引入了远程数据库链接特性，使得开发者在设计阶段花费很少或者不花费任何代码就可以链接到数据库。开发者不再需要定义一个设计时刻的数据库链接，UltraDev 会自动在设计阶段使用运行时刻的数据库链接。

16. 可编程的 JavaScript API

使用 DW UltraDev 功能强大的 API，可以添加自己定制的对象、行为、服务器脚本和命令等，根据自己的需要定制 UltraDev 工作环境，从而提高了工作效率。

17. 与主流开发工具集成

可以与 Adobe Photoshop、Macromedia Fireworks 和 Microsoft Office 等主流软件结合，进行流水作业。利用清理 Word HTML（Clean Up Word HTML）和集成表格化数据控件等功能，使用户能够轻易地从 Microsoft Word 以及 Microsoft Excel 和其他电子表格中获取内容。

18. 活动对象支持

在 DW UltraDev 4 中增加了对活动对象的支持,从而可以加速 Web 应用的开发进程。使用活动对象可以使得用户只用一个操作步骤就可以创建一些高级的页面组件。例如,可以使用活动对象来创建主页面和细节页面记录集,或者创建一个记录插入或更新表单,或者创建记录浏览条以及记录计数器等。

19. 站点访问服务器行为

DW UltraDev 4 给开发者提供了可以控制用户访问站点的工具。例如,我们可以创建这样的一些网页,使得用户在第一次进入站点和网页时要进行注册,以便该用户将来可以登录到站点;或者给不同的用户授予不同的访问权限,使得只有那些拥有访问权限的用户可以访问站点。

20. 增强的代码处理功能

DW UltraDev 4 新增的代码查看和语法着色功能使得处理 ASP、JSP 或 CFML 代码比以往更加容易。在设计阶段,DW UltraDev 还可以识别出应用文件 `global.asa`、`application.cfm` 和服务器端嵌入(SSI)文件。这些变化使得实时数据窗口的功能比以往更加强大。

21. 简化的插件模板

DW UltraDev 4 简化了其体系结构,使得创建插件更加容易。同时,DW UltraDev 4 也提供了一个新的服务器行为创建工具,使得用户可以轻松地通过已有的服务器行为创建新的服务器行为,并定制插入页面中的代码。

1.2.2 DW UltraDev 4 的主要应用

DW UltraDev 4 在实际中的应用很多,以下是一些典型应用。

1. 网站个性化

根据参观网站的客户的兴趣、爱好和他们自己的标准,可以很个性化地、动态地建造自己的网站。

2. 销售和库存管理

开发以浏览器为基础的销售和库存管理系统,这个系统能够让在外出差的和未在单位的市场人员实时地对相关数据进行浏览、编辑和更新。

3. 与客户的交互性

Web 应用的开发允许客户浏览信息数据库, 在这个数据库中允许他们进行选择、购买和在线传递订货请求。

4. 电子商务

可以创建电子商务网站来拓展传统销售渠道之外的产品销售方式。

5. 新闻和信息

开发一个信息量极为丰富的用户易于使用和浏览的网站。

6. 任务自动化的管理

可替代传统的效率极低的纸张处理模式。开发一种自动的客户机/服务器信息管理系统, 并可通过浏览器来进行操作。使用这个管理系统能根据一定的信息原则自动地分派和跟踪指定的任务。

7. 网上学习

通过网上自己定制学习计划来开展社区学习。

8. 社区建设

建立一个访问者能互相交流经验的论坛。

9. 人员通信录

人力资源部可根据人员发展的情况, 在 Web 上随时添加或删除人员。

10. 项目管理

在网上跟踪项目进展情况。针对不同的参加项目人员制订不同等级的管理权限。给管理人员分派监督项目情况的权限。

1.2.3 DW UltraDev 4 和其他同类产品的比较

目前, 开发 Web 应用的工具除了 DW UltraDev 4 之外, 还有以下三种非常有名的产品: Microsoft Visual Interdev 6.0、ColdFusion Studio 4.0 和 Adobe Golive With Dynamic Link。在表 1.1 中详细列出了它们之间的比较结果。从中显然可以看出, DW UltraDev 4 在各个方面都是一枝独秀。DW UltraDev 4 目前是第一个也是惟一个允许 Web 开发者

能够跨平台地在一个单一的可视化开发环境中快速开发 ASP、JSP 和 ColdFusion 的 Web 应用的软件。

表 1-1 DWUltraDev 4 和其他同类产品的比较

	DW UltraDev 4	Microsoft Visual Interdev 6.0	ColdFusion Studio 4.0	Adobe Golive With Dynamic Link
平台				
Macintosh	√	-	-	√
Windows	√	√	√	√
Web 应用开发环境				
生成 ASP 代码	√	√	-	√
生成 CFML	√	-	√	-
生成 JSP	√	-	-	-
往返服务器标记 (Roundtrip Server Markup)	√	-	-	-
干净代码 (没有多余的文件留在服务器上)	√	-	-	-
实时数据预览 (Live Data Preview)	√	-	-	-
服务器脚本库 (Server Script Library)	√	-	√	-
很容易把属性和文本绑定到数据库和服务器脚本	√	-	-	-
应用向导	-	-	√	-
实时数据个性化 (Live Data Personalization)	√	-	-	-
服务器脚本调试程序 (Server Script Debugger)	-	√	-	-
脚本代码完成 (Script Code Completion)	-	√	-	-
服务器组件集成 (COM、JavaBeans)	√	√	-	-
动态数据格式化 (Dynamic Data Formatting)	√	-	-	-
使用实时或测试服务器工作	√	√	-	-
数据库工具				
查询生成器 (Query Builder)	√	√	√	-
用 ODBC 链接数据库	√	√	-	√
用 OLE DB 链接数据库	√	-	-	-

(续表)

	DW UltraDev 4	Microsoft Visual Interdev 6.0	ColdFusion Studio 4.0	Adobe Golive With Dynamic Link
用 JDBC 链接数据库	√	-	-	-
连接数据库的对象集	√	√	√	-
根据测试和配置需要采用不同的数据库设置	√	-	-	-
设计和修改数据库计划	-	√	-	-
加入和测试数据库命令	√	√	-	-
快速创建数据库驱动的表单和报表	√	√	-	√
团队开发支持	√	√	-	-
扩展性				
可扩展的对象面板	√	-	-	√
XML 标记支持	√	-	√	√
创建自定义的宏	√	-	√	-
可定制的工作空间	√	√	√	-
可定制的菜单	√	-	√	-
可定制的键盘快捷键	√	-	√	-
可编程的 JavaScript API	√	-	-	-
团队开发				
开发备忘录	√	-	-	-
登记/检验	√	√	√	-
可视化的站点管理	√	√	-	√
与主流的Web 应用的可操作性	√	√	-	-

1.3 安装 DW UltraDev 4

DW UltraDev 4 支持 Windows 平台和 Macintosh 平台。本书主要以 Windows 操作系统为例来讲述 DW UltraDev 4 的使用。

DW UltraDev 4 支持的 Windows 平台包括：Windows 9x、Windows Me、Windows NT 4.0、Windows 2000，可以说基本上支持所有的 Windows 平台。系统要求是：48MB 以上

内存,30MB 以上硬盘。除此之外,系统中还需要安装有一个 ASP 2.0、JSP 1.0 或 ColdFusion Web 4.0 服务器。本书主要介绍 ASP, 因此读者必须在系统中安装 Microsoft Personal Web Server 4.0 或更高版本(针对 Windows 9x/Me/NT Workstation), 或者 Microsoft IIS 4.0 或更高版本(针对 Windows 2000/NT Server)。

另外, 读者还需要事先在系统中安装 Microsoft MDAC(Microsoft Data Access Components) 2.1 或更高版本, 它包含在 Microsoft Office 97/2000 中, 建议读者先安装 Microsoft Office。当然, 也可以到 <http://www.microsoft.com/data/download.htm> 站点上去下载 MDAC。

一切准备好以后, 就可以开始安装 DW UltraDev 4 了。和安装大部分软件一样, 运行安装程序, 首先会进行解压缩, 如图 1-14 所示。当把压缩包中的所有文件都释放到硬盘中以后, 就会出现如图 1-15 所示的欢迎界面。

图 1-14
安装程序
进行解压
缩

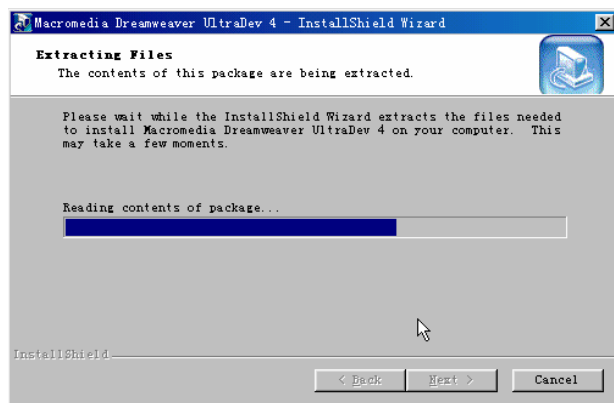
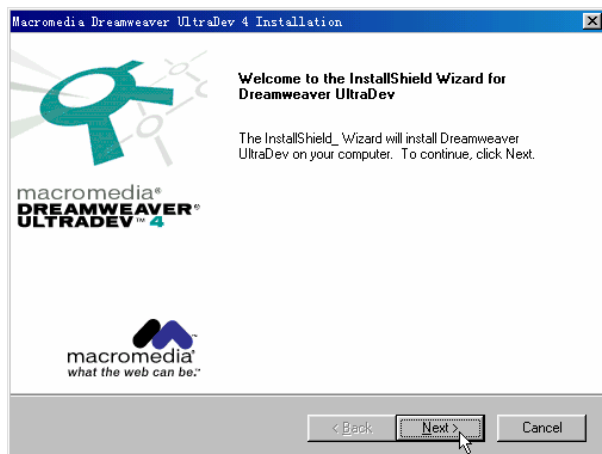


图 1-15
安装欢迎
界面



单击 Next 按钮, 将出现如图 1-16 所示的软件许可协议, 单击 Yes 按钮表示接受,

于是出现如图 1-17 所示的“输入产品序列号”窗口，其中需要输入用户名、机构和产品序列号。输入正确信息后，单击 Next 按钮继续安装。

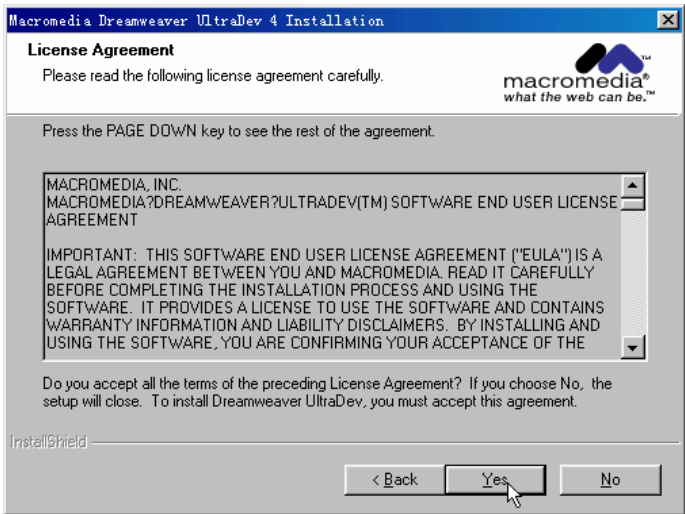


图 1-16
软件许可
协议

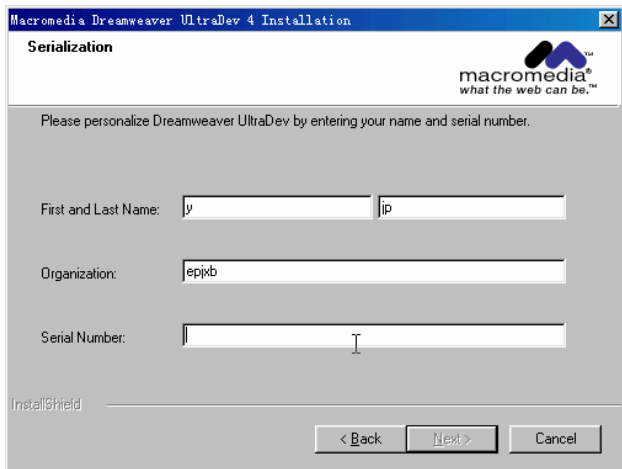


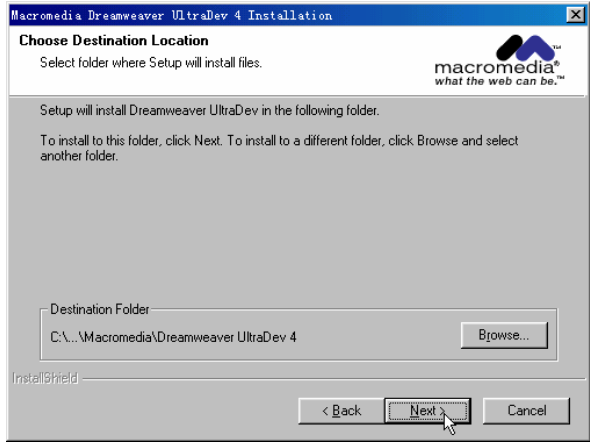
图 1-17
输入产品
序列号

接下来会出现如图 1-18 所示的警告窗口，提示用户在使用 DW UltraDev 4 之前必须首先安装 Microsoft MDAC 2.1 或更高版本。请单击“确定”按钮，于是会出现如图 1-19 所示的窗口。需要确定 DW UltraDev 4 的安装路径，用户可以接受缺省值，或者单击 Browse 按钮选择一个新的安装路径，然后单击 Next 按钮。

图 1-18
MDAC 警告窗口

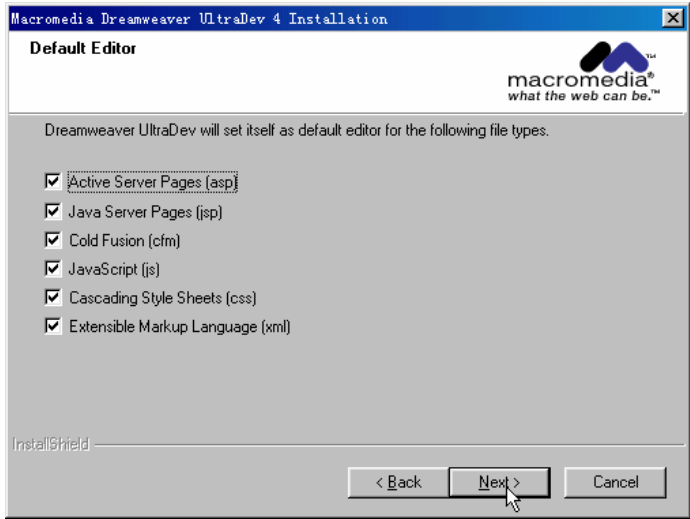


图 1-19
确定安装路径



现在需要确定各种不同的文件类型（包括.asp、.cfm、.jsp、.htm、.html、.js 和.css）的缺省编辑器，建议接受缺省值，即将所有的缺省编辑器都确定为 DW UltraDev 4，如图 1-20 所示。然后连续两次单击 Next 按钮，安装程序便开始复制文件，如图 1-21 所示。

图 1-20
选择缺省编辑器



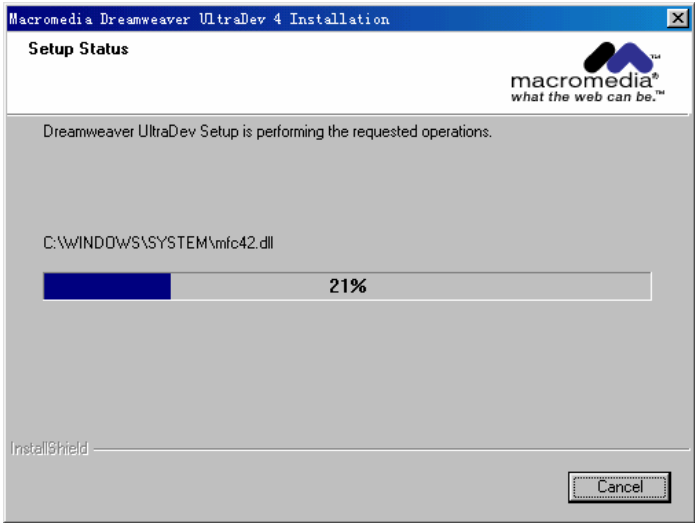


图 1-21
安装程序
复制文件

等复制过程结束，再单击 Finish 按钮，DW UltraDev 4 安装过程便结束。系统会自动在缺省浏览器（如 IE）中打开 DW UltraDev 4 的欢迎页面，如图 1-22 所示。请仔细阅读该页面。同时，在程序组中也会增加 Macromedia Dreamweaver UltraDev 4 这一项。

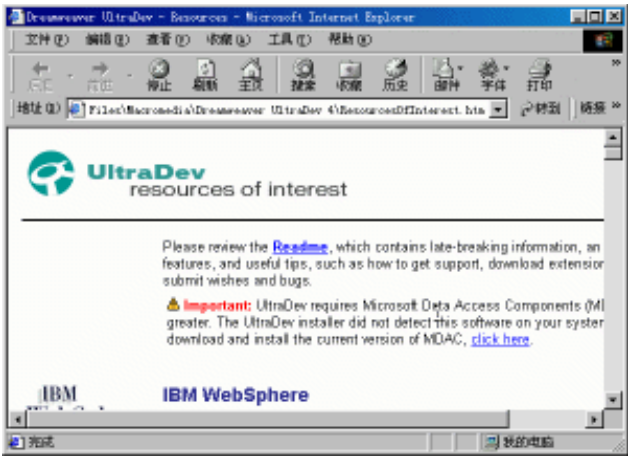


图 1-22
安装后的
欢迎页面

现在一切就绪，我们可以开始在 DW UltraDev 4 中尽情遨游了。

第 2 章 DW UltraDev 4 界面基础

本章概要：

- ☒ 启动 DW UltraDev 4
- ☒ 认识 DW UltraDev 4 的工作环境
- ☒ 熟悉 DW UltraDev 4 的主菜单
- ☒ 熟悉 DW UltraDev 4 的文档窗口、启动面板、对象面板、属性面板等各种基本面板功能
- ☒ 控制浮动面板
- ☒ 使用状态栏、快捷菜单及辅助设计工具
- ☒ 设置 DW UltraDev 4 的界面属性

前面介绍了许多 DW UltraDev 4 的新特性，或许显得过于枯燥，性急的读者可能早就按捺不住要开始摆弄这个神奇的软件了。这一章里我们就来熟悉一下 DW UltraDev 4 的界面，给大家一个稍微清晰的感性认识，为以后更好地使用 DW UltraDev 4 打下基础。

2.1 启动 DW UltraDev 4

安装了 DW UltraDev 4 之后，读者肯定要迫不及待地启动它。启动 DW UltraDev 4 的方法和启动任何其他软件一样，基本上有以下几种方式：一是标准启动方式，选择 开始 / 程序 / Macromedia Dreamweaver UltraDev 4/ Dreamweaver UltraDev 4” 即可；二是在桌面上创建一个快捷方式（很可惜 DW UltraDev 4 的安装程序没有自动完成这一点），从而节省启动 DW UltraDev 4 的步骤。创建快捷方式的方法很多，大多数都需要借助鼠标右键来完成，下面我们介绍一种最常用的方法。

将鼠标移动到“开始”按钮上，单击鼠标右键，便会出现一个快捷菜单，在其中选择“打开”菜单项，如图 2-1 所示。

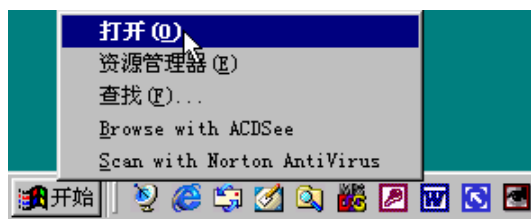


图 2-1
在“开始”
按钮上单
击鼠标右
键

系统将会打开 Start Menu 启动菜单窗口，依次双击鼠标左键进入“ Programs / Macromedia Dreamweaver UltraDev 4 ”文件夹，如图 2-2 所示。

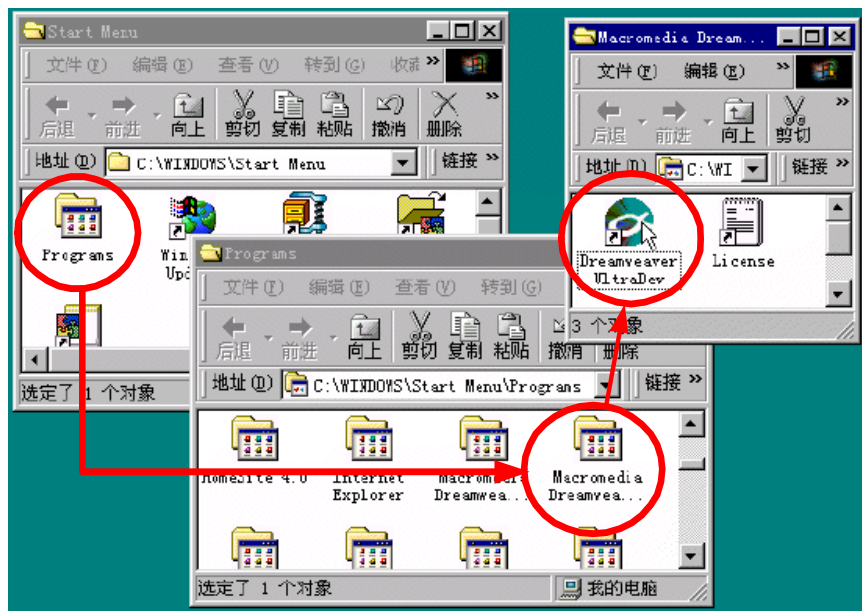


图 2-2
进入启动
菜单文件
夹

接着在 Dreamweaver UltraDev 4 文件上，按下鼠标右键不放，然后拖动到桌面上。在桌面的任何一个空白处释放鼠标右键，这时会出现一个弹出的快捷菜单，在其中选择“ 在当前位置创建快捷方式 ”菜单项，如图 2-3 所示。

于是桌面上便会会出现一个如图 2-4 所示的 DW UltraDev 4 快捷方式，用户只需双击它即可打开 DW UltraDev 4。

如果读者使用的操作系统是 Windows 98/Me 或 Windows 2000,则创建快捷方式的方法要更简单一些，只需展开到‘ 开始 / 程序 / Macromedia Dreamweaver UltraDev 4/ Dreamweaver UltraDev 4 ’，然后使用鼠标右键将 Dreamweaver UltraDev 4 拖放到桌面上，并在弹出的快捷菜单中选择‘ 在当前位置创建快捷方式 ’项即可。同样，在这两种操作系统中，也可以将快捷方式放置到桌面下方的工具栏上，从而最大程度地减少启动 DW UltraDev 4 的步骤。

图 2-3
拖动鼠标
右键来创
建快捷方
式

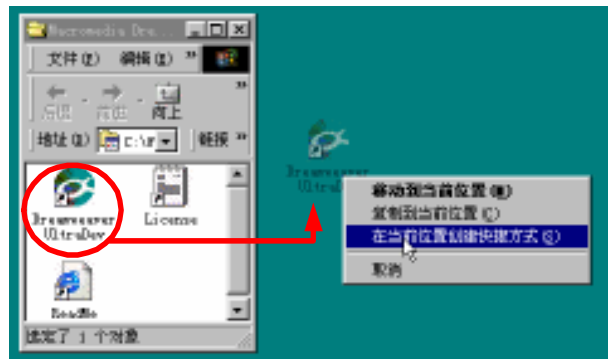
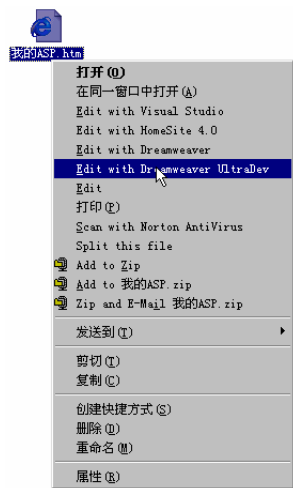


图 2-4
DW UltraDev
快捷方式



当然，以 Windows 操作系统的工作逻辑来说，我们不一定非要启动 DW UltraDev 4 之后，才能打开网页或其他文件。也就是说，这一个动作可以将它“倒过来”。由于刚才在安装过程中我们已经将 .asp、.cfm、.jsp、.htm、.html、.js 和 .css 等文件类型的默认编辑器设定为 DW UltraDev 4，因此第三种启动 DW UltraDev 4 的方法是：在资源管理器中找到任何一个具有上述后缀的文件，然后在其上单击鼠标右键，这时的快捷菜单中就应该出现“Edit with Dreamweaver UltraDev”菜单项（如图 2-5 所示），选择该项，即可启动 DW UltraDev 4。当然，使用这种方法启动 DW UltraDev 4 时，会同时打开该文件，而使用前两种方法启动 DW UltraDev 4 时，会新建一个空白的 HTML 文档。

图 2-5
使用快捷菜
单打开 DW
UltraDev



2.2 认识 DW UltraDev 4 的窗口结构

下面我们就可以开始使用 DW UltraDev 了,首先让我们熟悉一下它的窗口结构。第一次打开 DW UltraDev 4 时,会出现如图 2-6 中所示的注册窗口,提示用户通过 Web 对 DW UltraDev 进行注册。如果不想注册,请选择“Don't Remind Me Again”按钮,这个提示窗口以后启动时就不会再出现。接着就会打开如图 2-7 所示的 DW UltraDev 4 的主窗口。

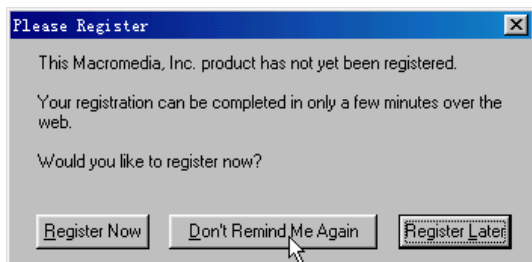


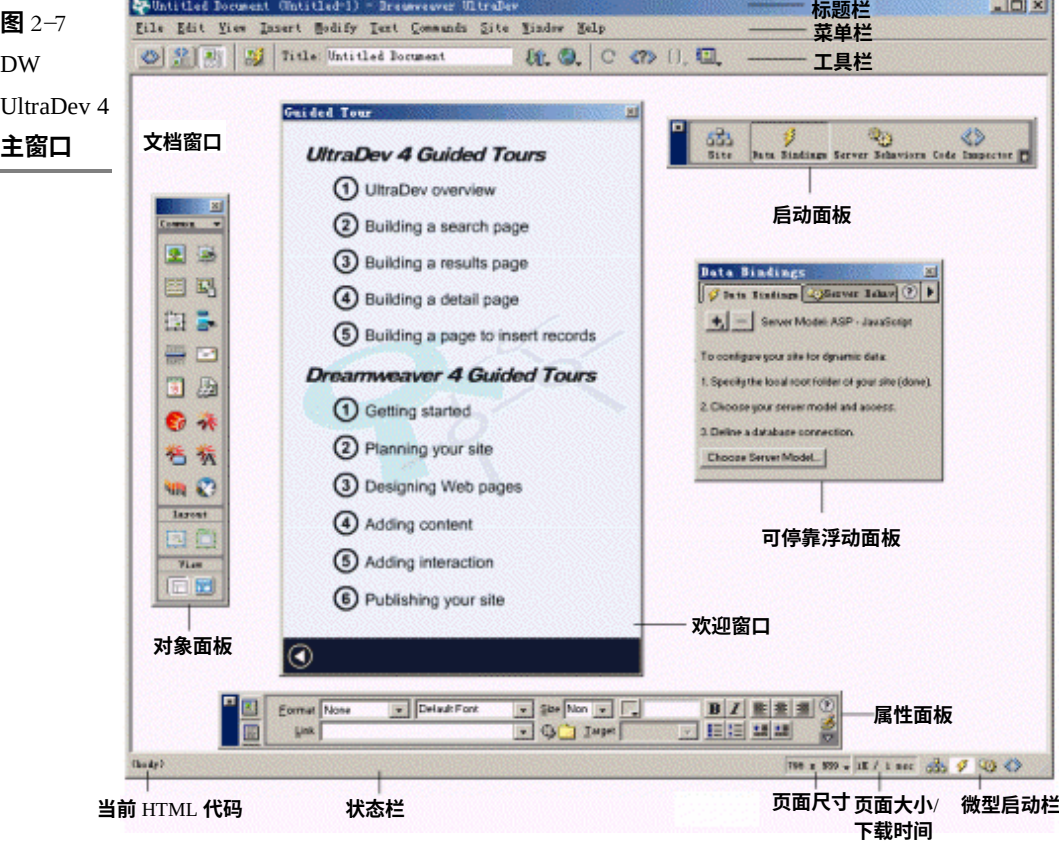
图 2-6
注册 DW
UltraDev 4

如果读者没有使用过 Dreamweaver,则第一次看到 DW UltraDev 4 的主窗口时,可能一时会感到手足无措。事实上,DW UltraDev 更像一个从 Macintosh 计算机上移植而来的产品,而不像一个纯粹的 Windows 应用程序,它的使用方法同 PageMaker 和 Photoshop 之类的软件类似,与微软建议的 Windows 应用程序的外观标准有一定的差别。事实上 DW UltraDev 4 的界面同 Dreamweaver 4.0 的界面是非常类似的,其外观同其异常灵活的功能特性分割不开,对于不同级别和经验的用户,都能够从这种应用程序外观上获得显著的工作效率。DW UltraDev 4 这种软件界面扩展了设计者视野,不再受制于屏幕显示的大小,而且 DW UltraDev 4 又提供了标尺刻度及底格功能,用户可以直接看到页面的实际显示尺寸,使得设计者在页面定位方面更加清晰明了,无须浏览器预览就清楚主页的整体页面效果,这个功能让我们感到好像我们不是在制作主页,而是在 Photoshop 中作图。

从图 2-7 中可以看出,DW UltraDev 4 的操作界面主要由主菜单、状态栏、工具栏、文档窗口和浮动面板组成。第一次启动 DW UltraDev 4 时系统还会打开一个欢迎窗口,这是引导初学者步入 DW UltraDev 4 的帮助窗口,单击其中任何一个按钮都会打开相应的帮助页面。

文档窗口中显示的是当前所创建和编辑的 HTML 文档内容。DW UltraDev 4 的所有编辑功能都是通过菜单栏和浮动面板来完成的,其中几乎所有的工作都可以通过菜单来完成,尽管利用浮动面板可以加快操作速度,但是有时候为了节省屏幕空间,会将浮动面板关闭,这时利用菜单就显得尤为重要,例如,利用 Window 菜单可以控制各种浮动

窗口和面板的开启和关闭。稍后我们会详细介绍各个菜单的功能。



对象面板、启动面板、属性面板和可停靠浮动面板通常总称作“浮动面板”，它们浮动于文档窗口之上，用户可以任意变换它们的位置，并将输入焦点在文档和面板之间来回切换。利用浮动面板来控制对页面的编写，而不是利用烦琐的对话框，这是 DW UltraDev 4 编辑网页中最令人称道的特性。在一些其他的网页编辑器，如 FrontPage 中，经常需要打开一个对话框来设置各种属性，在关闭对话框后才能看到设置结果，而在 DW UltraDev 4 中通过在浮动面板中进行设置，直接就可以在文档窗口中看到结果，避免了中间过程，从而提高了工作效率。

启动面板（Launcher）上包含了普通用户最常使用的控制按钮，通过这些按钮，可以快速启动或关闭最常用的检查器（Inspectors）和浮动面板（Palettes）。

在对象面板上包含了多种不同类型的按钮，用于在文档中创建不同类型的对象，例如图像、表格、框架等。

在属性面板中显示文档窗口中选中对象（如文本、图像、表格等）的属性，并且允

许对这些对象的属性进行修改。

可停靠浮动面板是一个工具集合的控制框架，它仿佛一个船坞，用户可以将各种浮动窗口、面板和检查器放置在其中，组合成为标签的形式，以节省屏幕空间，必要时还可以将这些浮动窗口、面板和检查器从中剥离，成为独立的浮动面板。

工具栏是 DW UltraDev 4 新增的一个界面元素(DW UltraDev 1.0 中并没有工具栏)。通过工具栏我们可以轻松地使用一些最常见的特性，如在浏览器中预览、设计日志、以不同的方式查看文档窗口等。

标题栏上显示当前正在编辑文档的标题和名称，它首先显示文档的标题，然后显示文档的文件名称。

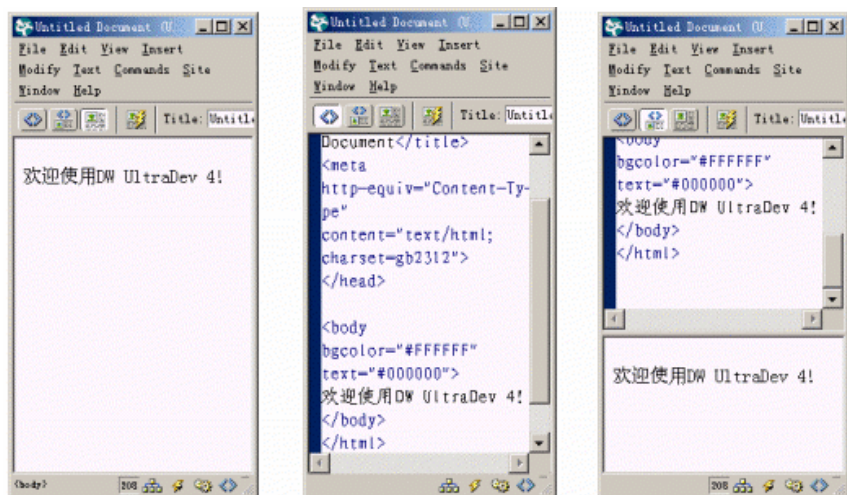


请注意文档的标题和文档的文件名称是不同的概念。文件的标题通常在文档中的<title>和</title>标记中，是在用浏览器打开文档时显示在浏览器窗口的标题栏上的名称，而文件的名称则是文档存储在磁盘上的文件名。

DW UltraDev 4 的状态行很有特色，它包括几个功能区，分别用于显示和控制文档的 HTML 源代码、显示页面尺寸、查看页面大小和传输时间，其最右方还包含了和启动栏一样的按钮组合，称作“微型启动栏”，可以完成启动栏同样的功能。

另外，DW UltraDev 4 还对文档窗口作了专门的处理，使其不仅能够完成正常的编辑工作，还可以在同一个窗口中查看当前编辑页面的 HTML 源代码。

图 2-8 中显示的就是文档窗口的三种工作方式。



(a)

(b)

(c)

图 2-8
文档窗口
的三种工
作方式

其中 (a) 图就是正常情况下的文档窗口，用户可以在此状态下完成网页的编辑工作（如键入文字等）。如果用户选择了“ View / Code ”菜单命令，或者单击了工具栏上的 Show Code View 按钮，文档窗口就会切换成 (b) 图所示的样子，其中列出了当前网页的 HTML 源代码；如果用户选择“ View / Code and Design ”菜单命令，或者单击了工具栏上的 Show Code and Design View 按钮，文档窗口就会切换成 (c) 图所示的样子，此时文档窗口被上下分割成两部分，其中上半部分显示的是页面的 HTML 源代码，而下半部分显示的是正在编辑中的页面；如果用户选择“ View / Design ”菜单命令，或者单击了工具栏上的 Show Design View 按钮，则文档窗口又会切换回 (a) 图所示的样子。

2.3 DW UltraDev 4 的主菜单

DW UltraDev 4 的主菜单共分十大类：File、Edit、View、Insert、Modify、Text、Commands、Site、Window 和 Help，作用分别为文件管理、选择区域文本编辑、观察对象、插入元素、修改页面元素、文本操作、附加命令项、站点管理、所有浮动面板切换与窗口切换和联机帮助。

2.3.1 File 菜单

File 菜单包括如图 2-9 所示的选项。

图 2-9
File 菜单

New	Ctrl+N
New from Template...	
Open...	Ctrl+O
Open in Frame...	Ctrl+Shift+O
Close	Ctrl+W
Save	Ctrl+S
Save As...	Ctrl+Shift+S
Save as Template...	
Save All Frames	
Revert	
Import	▶
Export	▶
Convert	▶
Preview in Browser	▶
Debug in Browser	▶
Check Links	Shift+F8
Check Target Browsers...	
Design Notes...	
1 ResultsAdv.asp	
2 Search.htm	
3 Insert.asp	
4 Detail.asp	
Exit	Ctrl+Q

New：新建一个文档。

New from Template：根据现有模板来新建文档。

Open：打开一个现有文档。

Open in Frame：如果目前窗口为分帧的话，此选项将在目前所处帧中打开文档。

Close：关闭目前的文档窗口（注意不是关闭 DW UltraDev）。

Save：保存文件。

Save As：另存为其他文件。

Save as Template：将文档存为模板文件。

Save All Frames：保存当前窗口的所有文档。

Revert：将文档恢复至上次存盘时的状态。

Import——

- Import XML into Template：引入 XML 到模板中。
- Import Word HTML：引入 Word 文档。
- Import Table Data：引入表格数据。

Export——

- Export Editable Regions as XML：将窗口的模块转换为 XML 文档。
- Export CSS styles：将窗口中的 CSS 样式表定义语句转换为 .css 文档。
- Export Table：输出表格。

Convert——

- 3.0 Browser compatible：将窗口文档转换为 3.0 版本浏览器兼容格式。

Preview in Browser：将窗口文档在设定的浏览器中打开（以 TMP***为文件前缀）。

Debug in Browser：在浏览器中对当前文档的脚本语句进行调试。

Check Links：检查有无出错链接。

Check Target Browsers：可以将当前文档用不同的浏览器诊断，从而可以了解 HTML 文档在不同的浏览器的兼容性和正确性。

Design Notes：打开设计日志。

Exit：关闭所有的窗口，退出 DW UltraDev 4。

此外，在 Exit 菜单项的上方还会列出最近打开的 4 个文档的文件名。

2.3.2 Edit 菜单

Edit 菜单包括如图 2-10 所示的选项。

图 2-10
Edit 菜单

Undo Typing	Ctrl+Z
Repeat Typing	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Clear	
Copy HTML	Ctrl+Shift+C
Paste HTML	Ctrl+Shift+V
Select All	Ctrl+A
Select Parent Tag	Ctrl+Shift+<
Select Child	Ctrl+Shift+>
Find and Replace...	Ctrl+F
Find Next	F3
Indent Code	Ctrl+]
Outdent Code	Ctrl+[
Balance Braces	Ctrl+'
Set Breakpoint	Ctrl+Alt+B
Remove All Breakpoints	
Edit with External Editor	Ctrl+E
Preferences...	Ctrl+U
Keyboard Shortcuts...	

Undo Typing：撤消当前操作返回到上一次状态。

Repeat Typing：恢复撤消动作。

Cut：剪切选择区域。

Copy：复制选择区域。

Paste：将内存中的文档粘贴到当前窗口中。

Clear：清空当前选择区域。

Copy HTML：与 Copy 不同，它将所选择区域文档块仅作为 HTML 代码块载入内存。

Paste HTML：将文字以 HTML 格式的形式粘贴到当前窗口文档中。

Select All：将窗口中的所有内容全部选中。

Select Parent Tag：选择当前标记的上一层标记所表示的区域。

Select Child：选择当前标记的下一层标记所表示的区域。

Find and Replace：查找和替换功能几乎是编辑软件都有的功能选项，但 DW UltraDev 4 的查找和替换功能大大增强。它可以在当前窗口查找和替换，也可以在整个站点查找和替换，可以在预览窗口查找和替换，也可以在 HTML 源程序中查找和替换源代码。

Find Next：查找满足查询条件的下一项。

Indent Code：代码缩进。

Outdent Code：代码反向缩进。

Balance Braces：括号匹配。

Set Breakpoint：设置断点，这是 DW UltraDev 4 新增的功能，可以在页面中对 JavaScript 代码进行跟踪调试。

Remove All Breakpoints：删除所有断点。

Edit with External Editor：启动外部编辑器进行编辑，这是 DW UltraDev 4 的一大优点。Macromedia 公司知道 DW UltraDev 虽然很好，但它并不是万能的，因此它允许用户附上自己熟悉的 HTML 编辑器一同工作：在 PC 机上 DW UltraDev 4 和 Homesite 捆绑在一起，在 MAC 机上它与 BBEdit 捆绑在一起，如果在任一方修改并保存后，另一方编辑器会自动检测，并提醒用户是否重新载入。

Preferences：设置 DW UltraDev 4 工作参数，包括选择外部编辑器、FTP、浏览器设定、层、默认字体等众多参数。

Keyboard Shortcuts：键盘快捷方式，可以查看和修改 DW UltraDev 所有命令的快捷键。

2.3.3 View 菜单

View 菜单包括如图 2-11 所示的选项。

✓ Code	
Design	
Code and Design	
Switch Views	Ctrl+Tab
Refresh Design View	F5
Design View on Top	
Live Data	Ctrl+Shift+R
Refresh Live Data	Ctrl+R
Live Data Settings...	
Head Content	Ctrl+Shift+W
Table View	▶
Visual Aids	▶
Code View Options	▶
Rulers	▶
Grid	▶
Tracing Image	▶
Plugins	▶
Show Panels	F4
✓ Toolbar	Ctrl+Shift+T

图 2-11
View 菜单

Code：以代码视图的方式查看文档窗口。

Design：以设计视图的方式查看文档窗口。

Code and Design：同时以代码视图和设计视图的方式查看文档窗口。

Switch Views：在代码视图和设计视图之间进行切换。

Refresh Design View：刷新设计视图。

Design View on Top：当代码视图和设计视图同时显示时，使得设计视图显示在文档窗口上半部分（默认时代码视图位于上半部分）。

Live Data：使当前文档进入实时预览状态。

Refresh Live Data：更新文档的预览内容。

Live Data Settings：设置实时预览。

Head Content：显示 HTML 文档头部的设置状况，即<head>与</head>之间的部分。

Table View：确定表格的显示方式。

- Standard View：以标准方式显示表格。
- Layout View：以平面布局方式显示表格。
- Show Layout Table Tabs：当以平面布局方式显示表格时，在表格最上方显示一些用于控制的标签。

Visual Aids：控制界面中某些元素是否显示。

- Hide All：隐藏所有边界和不可见元素。
- Table Borders：显示（或隐藏）表格的边框。
- Layer Borders：显示（或隐藏）层的边框。
- Frame Borders：显示（或隐藏）帧的边框。
- Image Maps：显示（或隐藏）影像地图标记。
- Invisible Elements：显示（或隐藏）不可见元素，如锚、备注、Script、空格等。

Code View Options：控制 HTML 代码的显示样式，这也是 DW UltraDev 4 新增的一项功能，大大提高了代码的编辑功能。

- Word Wrap：代码自动换行。
- Line Numbers：显示行号。
- Highlight Invalid HTML：加亮显示无效的 HTML 代码。
- Syntax Coloring：根据语法对代码进行着色。
- Auto Indent：自动缩进。

Rulers：标尺，可以更直观、更精确地观看文档窗口。

- Show：显示（或隐藏）标尺。
- Reset Origin：恢复标尺原点。
- Pixels：标尺单位为像素。
- Inches：标尺单位为英寸。
- Centimeter：标尺单位为厘米。

Grid：网格。

- Show Grid：显示（或隐藏）网格。
- Snap To Grid：吸附网格。
- Edit Grid：设置网格的具体参数，比如点状或线状、每个单位的具体大小等。

Tracing Image：它可以将一幅图片嵌入页面的任何一个位置，而且可以改变透明度，其作用是用来控制页面的布局。这时该图片所占据的 BODY 中的背景设置将会在 DW UltraDev 4 预览界面中失效，但在任何一个浏览器中都不会显示出该图片，图片格式可以为 JPG、GIF、PNG。

- Show：显示追踪图像。
- Align With Selection：追踪图像与选择区域对齐。
- Adjust Position：调整追踪图像的位置。
- Reset Position：恢复追踪图像至起始位置。
- Load：载入追踪图像，并可以调整相应属性，如透明度、位置等。

Plugins：用来控制窗口文档中的插件的播放。

- Play：播放插件。
- Stop：停放播放插件。
- Play All：播放窗口所有的插件。
- Stop All：停放播放窗口所有的插件。

Show/Hide Panels：显示或隐藏各控制面板。

Toolbar：用来选择是否显示（或隐藏）工具栏。

2.3.4 Insert 菜单

Insert 菜单包括如图 2-12 所示的选项。

Image：插入图像。

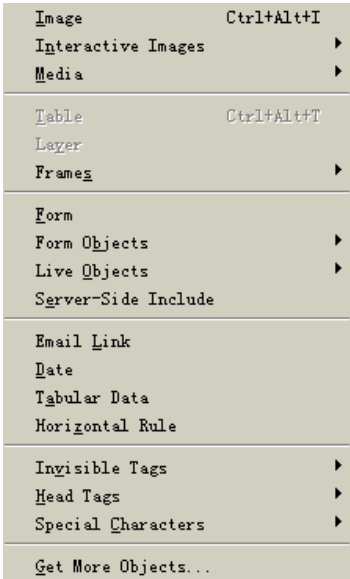
Interactive Images：插入可交互的图像。

- Rollover Image：建立一个互动图像。
- Navigation Bar：插入浏览条。
- Flash Button：插入 Flash 按钮。
- Flash Text：插入 Flash 文本。
- Fireworks HTML：插入 Fireworks 的 HTML 代码。

Media：插入各种外部媒体对象，其子菜单有 Flash、Shockwave、Generator、Applet、Plugin 和 ActiveX。

Table：插入表格。

图 2-12
Insert 菜单



Layer：插入层。

Frames：插入框架，其子菜单有 Left、Right、Top、Bottom、Left and Top、Left Top、Top Left、Split。

Form：插入一个表单。

Form Object：关于表单的各种对象。

- Text Field：插入文本框。
- Button：插入按钮。
- Check Box：插入复选框。
- Radio Button：插入单选框。
- List/Menu：插入下拉框。
- File Field：插入文件输入框。
- Image Field：插入图像框。
- Hidden Field：插入隐藏框。
- Jump Menu：插入跳转下拉菜单。

Live Objects：插入活动对象。

- Master Detail Page Set：主页面和细节页面集。
- Recordset Navigation Bar：记录集浏览条。
- Recordset Navigation Status：记录集浏览状态。
- Record Insertion Form：插入记录表单。
- Record Update Form：记录更新表单。

Server-Side Include：编辑一段服务端引用脚本。

E-mail Link：插入 E-mail 链接。

Date：插入日期和时间。

Tabular Data：插入表格数据。

Horizontal Rule：插入一条直线。

Invisible Tags：插入不可见标记。

- Named Anchor：定义一个锚链接。
- Script：加上一段脚本(JavaScript、VBScript)。
- Comment：加上一段备注。

Head Tags：在 HEAD 头部加上描述语句或关键字等，其子菜单有 Meta、Keywords、Description、Refresh、Base 和 Link。

Special Characters：插入各种特殊字符，其子菜单有 Line Break(换行符
)、Non-Breaking Space(空格，不过在 DW UltraDev 4 预览窗口看不出来)、Copyright(版权符?)、Registered(注册商标?)、Trademark(商标?)、Pound(英镑£)、Yen(人民币¥)、Euro(欧元€)、Em-Dash(分隔符—)、Left Quote(左双引号‘)、Right Quote(右双引号”)和 Other(其他符号)

Get More Objects：到<http://www.macromedia.com/exchange/ultradev/>网站上去获取更多的对象。

2.3.5 Modify 菜单

Modify 菜单包括如图 2-13 所示的选项。

Page Properties：修改当前窗口文档的属性，包括背景、背景色、链接色彩、字符串编码、文档标题等。

Selection Properties：弹出属性浮动面板，并显示当前所选元素的相应参数，比如当前所选为字体，则弹出的属性浮动面板显示的将是字体的所有相关属性。

Quick Tag Editor：打开快速标记编辑器。

Make Link：制作链接(改变链接)。

Remove Link：取消链接。

Open Linked Page：打开所链接的页面(mailto、http 之类的远端页面是打不开的)。

Link Target：选择打开链接的目标位置，其子菜单有 Default Target、_blank、_parent、_self、_top、mainFrame、Set，如果当前文档中包含了框架，则子菜单中还会包括各框架的名字(如 leftFrame 等)。

Table：这部分的子菜单均是针对于表格进行操作的，功能比较强。

图 2-13
Modify 菜单

Page Properties...	Ctrl+J
✓ Selection Properties	Ctrl+Shift+J
Quick Tag Editor	Ctrl+T
Make Link	Ctrl+L
Remove Link	Ctrl+Shift+L
Open Linked Page...	
Link Target	▶
Table	▶
Frameset	▶
Navigation Bar	
Arrange	▶
Align	▶
Convert	▶
Library	▶
Templates	▶
Timeline	▶
Connections...	

- Select Table：选择表格。
- Merge Cell：合并单元格。
- Split Cell：分割单元格。
- Insert Row：插入行。
- Insert Columns：插入列。
- Insert Rows or Columns：插入行或列。
- Delete Row：删除行。
- Delete Columns：删除列。
- Increase Rows Span：合并当前光标所在表格单元的上下格。
- Increase Columns Span：合并当前光标所在表格单元的左右格。
- Decrease Rows Span：是 Increase Rows Span 的反操作。
- Decrease Columns Span：是 Increase Columns Span 的反操作。
- Clear Cell Heights：清空定义的单元格的高度。
- Clear Cell Widths：清空定义的单元格的宽度。
- Convert Widths to Pixels：将宽度转化为像素。
- Convert Widths to Percent：将宽度转化为百分比。

Frameset：是用于框架的，其子菜单有 Edit NoFrames Content(编辑不支持框架时的显示内容)、Split Frame Left(分割左框架)、Split Frame Right(分割右框架)、Split Frame Up(分割上框架) 和 Split Frame Down(分割下框架)。

Navigation Bar：修改浏览条。

Arrange：对页面中的对象进行编排。

- Bring To Front：将所选对象放置到所有对象的最上面。
- Send To Back：将所选对象放置到所有对象的最下面。
- Prevent Layer Overlaps：防止层与层交叠，选上此项后，当用户再画层时，若靠近原来的层，会出现一个 STOP 的标志⊙，并且无法将一个层移动到另一个层的上方。

Align：对页面中的对象进行对齐控制，其子菜单有 Left(左对齐)、Right(右对齐)、Top(上对齐)、Bottom(下对齐)、Make Same Width(使对象宽度相同)和 Make Same Height(使对象高度相同)。

Convert：格式转换。

- Tables to Layers：将表格转换为层结构。
- Layers to Table：将层转换成表格结构。

Library：这部分的子菜单是用来对库进行操作的。

- Add Object To Library：将对象增加到库中。
- Update Current Page：更新当前页。
- Update Pages：更新多页。

Templates：模板，功能与 Library 的功能很相似，都是建立一个固定模型，然后在其他页面中可以反复调用，以提高效率。其选项中的下拉菜单均是对模块的控制。

- Apply Template to Page：将模板应用到页面中。
- Detach from Template：从模板中分离。
- Open Attached Template：打开附着的模板。
- Update Current Page：更新当前页。
- Update Pages：更新多页。
- New Editable Region：创建新的可编辑区域。
- Mark Selection as Editable：将选择部分标记为可编辑。
- Unmark Editable Region：取消可编辑区域的标记。
- No Editable Regions：取消所有可编辑区域。

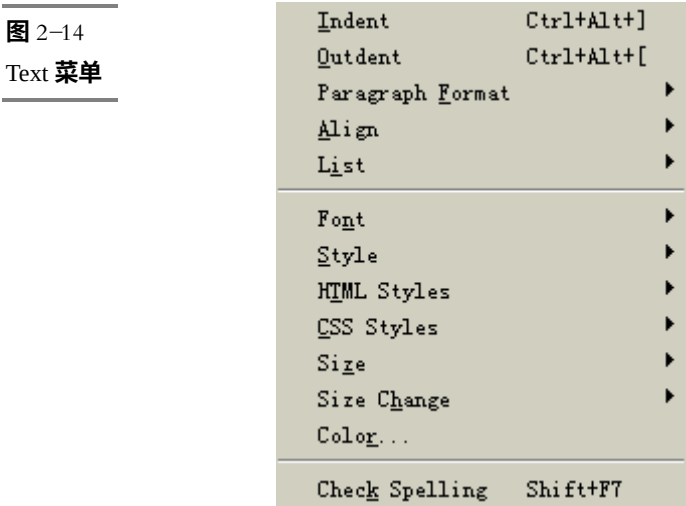
Timeline：此项目是对时间链的修改设置。

- Add Object to Timeline：将选取的对象(层或图像)加入到时间链中。
- Add Behavior to Timeline：在时间链中加入行为，以指定任意一帧触发事件。
- Record Path of Layer：利用这个选项可以任意调整一层中的元素的路径，这样可以用它来实现页面元素的曲线运动。
- Add Keyframe：增加关键帧。
- Remove Keyframe：删除关键帧。
- Change Object：改变对象。
- Remove Object：删除对象。

- Remove Behavior：删除行为。
 - Add Frame：增加帧。
 - Remove Frame：删除帧。
 - Add Timeline：增加时间链。
 - Remove Timeline：删除时间链。
 - Rename Timeline：重命名时间链。
- Connections：增加、修改或删除连接。

2.3.6 Text 菜单

Text 菜单包括如图 2-14 所示选项。



- Indent：页面左右同时缩进，可多次使用。
- Outdent：是 Indent 的反操作，页面左右同时向外扩张。
- Paragraph Format：段落格式，包括段落符、HTML 中的 H1～H6，以及预格式化的文本。
- Align：选择对齐方式，左对齐、居中或右对齐。
- List：包含 Ul、Ol 和 Dl，分别对应有序列表、无序列表和定义列表，并可以设置这些列表的属性。
- Font：选择字形，最下面的一项可以编辑字体列表。
- Style：字体风格选项，比如字体加粗、斜体、加下划线、字体变形等。

- HTML Styles：使用或定义 HTML 样式。
- CSS Styles：使用或定义 CSS 样式表。
- Size：设置字体大小，从 1~7 号。
- Size Change：从相对的角度将字体增大或减少若干号。
- Color：定义字体颜色。
- Check Spelling：检查单词拼写是否正确，适用于英文。

2.3.7 Commands 菜单

Commands 菜单包括如图 2-15 所示的选项。

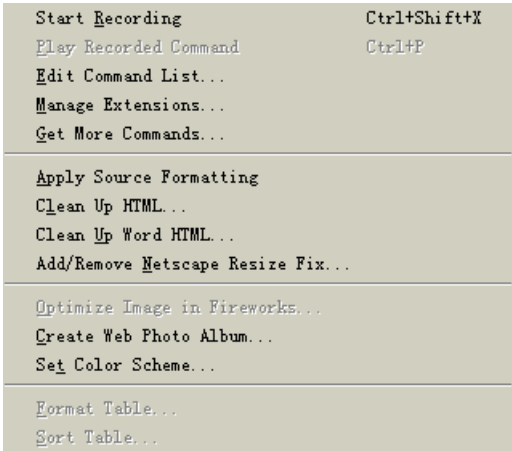


图 2-15
Commands
菜单

- Start Recording：开始记录用户操作过程。
- Play Recorded Command：重新播放所记录的用户操作。
- Edit Command List：编辑命令列表。
- Manage Extensions：打开 Extensions 插件管理器。
- Get More Commands：到 <http://www.macromedia.com/exchange/ultradev/> 网站上去获取更多的命令。
- Apply Source Formatting：应用源格式。
- Clean Up HTML：这就是除去 HTML 垃圾代码的工具。其中的对话框中有几组复选框可供选择，如空标签、多余的无用的嵌套等，还可以自己设定要清除的特定标签。
- Clean Up Word HTML：清除使用 Word 产生的 HTML 中的垃圾代码。
- Add/Remove Netscape Resize Fix：由于一些版本的 Netscape Navigator 包含了这样的 Bug，当用户改变浏览器窗口的大小时当前页面无法正常显示，此命令可以通过加

入 JavaScript 代码来解决这个问题。

Optimize Image in Fireworks：对使用 Fireworks 产生的图像进行优化。

Create Web Photo Album：创建一个 Web 相册，需要 Fireworks 支持。

Set Color Scheme：色彩配置向导，这里有许多已经匹配的背景色、文本色、超链接色，用户只需选择就可以直接运用到文本中去。

Format Table：提供了许多已有的表格格式，只要选择就可直接使用。

Sort Table：表格排序。

2.3.8 Site 菜单

Site 菜单包括如图 2-16 所示的选项。

图 2-16

Site 菜单

Site Files	F8
Site Map	Alt+F8
New Site...	
Open Site	
Define Sites...	
Get	Ctrl+Shift+D
Check Out	Ctrl+Alt+Shift+D
Put	Ctrl+Shift+U
Check In	Ctrl+Alt+Shift+U
Undo Check Out	
Reports...	
Check Links Sitewide	Ctrl+F8
Remove Connection Scripts	
Locate in Local Site	
Locate in Remote Site	

Site Files：在 Site 窗口中列出站点中的文件。

Site Map：在 Site 窗口中显示站点地图。

New Site：创建一个新站点。

Open Site：打开现有站点。

Define Sites：定义站点。

Get：从远端下载站点。

Check Out：可以翻译为“检验”，主要用于团队开发的情况。此时“Check out a file”意味着“我正在编辑该文件，请其他人不要修改它”。如果一个文件被设置为“检验”状态，则系统首先会把该文件以 Get 方式下载到本地站点，在站点窗口的列表中紧挨着该文件图标旁边会出现一个“检验”标记（通常是一个“√”符号）。如果该标记为绿色，则表明是当前用户锁定了该文件；如果该标记是红色，则表明是其他用户锁定了该文件。同时，在文档窗口的列表中还会显示出锁定该文件的用户名。

- Put：上传站点。
- Check In：可以翻译为“ 登记 ”，与“ Check Out ”作用相反。将某一个被“ 检验 ”的文件“ 登记 ”以后，系统首先会把该文件以 Put 方式上传到服务器，并允许其他用户“ 检验 ”并修改该文件。同时本地的该文件会被设置为“ 只读 ”属性，以防止其他用户修改该文件时当前用户也同时进行编辑操作。
- Undo Check Out：取消 Check Out 操作。
- Reports：生成测试报告。这是 DW UltraDev 4 新增的功能，可以非常方便地帮助我们找到页面中的问题。
- Check Links Sitewide：检查站点中的所有超连接。
- Remove Connection Scripts：删除用于链接数据库的脚本。
- Locate in Local Site：定位到本地站点。
- Locate in Remote Site：定位到远端站点。

2.3.9 Window 菜单

Window 菜单包括如图 2-17 所示的选项。

✓ Objects	Ctrl+F2
✓ Properties	Ctrl+F3
✓ Launcher	
Data Bindings	Ctrl+F10
Server Behaviors	Ctrl+F9
✓ Site Files	F8
Site Map	Alt+F8
Assets	F11
Behaviors	Shift+F3
Code Inspector	F10
CSS Styles	Shift+F11
Frames	Shift+F2
History	Shift+F10
HTML Styles	Ctrl+F11
Layers	F2
Library	
Reference	Ctrl+Shift+F1
Templates	
Timelines	Shift+F9
Arrange Panels	
Hide Panels	F4
Minimize All	Shift+F4
Restore All	Alt+Shift+F4
Untitled-1	

图 2-17
Window
菜单

Objects：切换对象浮动面板。

Properties：切换属性浮动面板。

Launcher：切换快速启动浮动面板。

Data Bindings：切换数据绑定面板。

Server Behaviors：切换服务器行为面板。

Site Files：切换站点文件管理器。

Site Map：切换站点地图。

Assets：切换 Assets 面板。Assets 面板是 DW UltraDev 4 新增加的面板，其中放置了本站点所使用的所有资源，包括图片、颜色、外部 URL、脚本语句以及 Flash、Shockwave、QuickTime、模板和库等项。这使得站点维护者可以轻易地对本站所使用的资源一目了然，非常方便地在本站范围内重用这些资源。

Behaviors：切换行为编辑器。

Code Inspector：切换 HTML 源码检查器面板。

CSS Styles：切换 CSS 样式表编辑器。

Frames：切换帧浮动面板。

History：切换历史记录面板。

HTML Styles：切换 HTML 样式表编辑器。

Layers：切换层浮动面板。

Library：切换库面板。

Reference：切换 Reference 面板。这也是 DW UltraDev 4 所新增加的面板。它是一个让使用者能够快速找到有关 HTML、JavaScript 和 CSS 的参考资料的工具。

Templetes：切换模板编辑器。

Timelines：切换时间链。

Arrange Panels：排列所有的浮动面板。

Hide Panels：隐藏所有的浮动面板。

Minimize All：将 DW UltraDev 4 中所有窗口（包括浮动面板、文档窗口和站点窗口等全部窗口）进行最小化。

Restore All：和 Minimize All 作用相反，将 DW UltraDev 4 中所有窗口恢复到原来的状态。

另外，Window 菜单条的最下方还会列出所有已打开的文档名称。

2.3.10 Help 菜单

Help 菜单包括如图 2-18 所示的选项。

Welcome	
Using UltraDev	F1
Using Dreamweaver	Ctrl+F1
Reference	Shift+F1
What's New	
Guided Tour	
Lessons	
UltraDev Exchange	
Manage Extensions...	
UltraDev Support Center	
Macromedia Online Forums	
Extending Dreamweaver & UltraDev	
Creating and Submitting Extensions	
Register UltraDev	
About UltraDev	

图 2-18
Help 菜单

- Welcome：打开欢迎窗口。
- Using UltraDev：打开 UltraDev 的帮助页面。
- Using Dreamweaver：打开 Dreamweaver 帮助页面。
- Reference：和“Window / Reference”菜单命令是重复的，作用是打开 Reference 面板。
- What's New：打开 UltraDev 和 Dreamweaver 的新增功能页面。
- Guided Tour：打开 UltraDev 和 Dreamweaver 指南页面。
- Lessons：打开 UltraDev 和 Dreamweaver 教学页面。
- UltraDev Exchange：打开 Macromedia 公司的 Exchange 扩展网站，网址为 <http://www.macromedia.com/exchange/ultradev/>。
- Manage Extensions：和“Command / Manage Extensions”菜单命令的作用是相同的，打开 Extensions 插件管理器。
- UltraDev Support Center：打开 Macromedia 公司的 UltraDev Support Center 支持中心网站，网址为<http://www.macromedia.com/support/ultradev/>。
- Macromedia Online Forums：打开 Macromedia 公司的在线论坛，网址为 <http://www.macromedia.com/support/forums/>。
- Extending Dreamweaver & UltraDev：打开扩展 Dreamweaver 和 UltraDev 帮助页面。
- Creating and Submitting Extensions：打开、创建并提交插件的帮助页面。
- Register UltraDev：注册 UltraDev。
- About UltraDev：打开 UltraDev 的介绍窗口。

2.4 基本面板功能

大大小小的面板是 DW UltraDev 4 最富特色的工具。没有使用过 Dreamweaver 的读

者可能会觉得 DW UltraDev 4 的界面非常复杂。其实，这些面板的使用是非常简单方便的，下面我们就介绍三种最基本的面板。

2.4.1 启动面板

启动面板主要用于快速启动相应的任务，或显示/隐藏相应的浮动面板。在状态栏的右方有一个微型启动栏，它同启动面板上的按钮是一样的，如图 2-19 所示。

图 2-19
启动面板
和微型启动
栏的对应
关系



启动面板上包含了 4 个按钮，分别是：Site(站点) 按钮、Data Bindings(数据绑定) 按钮、Server Behaviors(服务器行为) 按钮和 Code Inspector(HTML 源代码检查器) 按钮。启动面板上的按钮都是按下/弹起类型的按钮，也被称作切换按钮。换句话说，这些按钮可以保持被按下的状态。单击启动面板上的按钮，使之被按下，会激活相应的浮动面板窗口，再次单击其上的按钮，可以使之弹起，这会将相应的浮动面板窗口隐藏。例如，在默认状态下，HTML 源代码检查器是一个单独形式的窗口，单击启动面板上的“ Code Inspector ”(HTML 源代码) 按钮，可以激活 HTML 源代码检查器，显示当前编辑文档的源代码，再次单击该按钮，则可以重新隐藏 HTML 源代码检查器。



1. Site 窗口

单击 Site 按钮即可打开 Site 窗口，如图 2-20 所示。

如果读者使用过 LeapFTP 或 CuteFTP 等之类的文件上传软件，那么就会对 Site 窗口有点似曾相识的感觉。一方面，Site 窗口可以利用树状结构或文件列表的形式让开发者一睹网站中的所有文件；另一方面，Site 窗口又能作为一个 FTP 传输界面，实现站点的上传和下载功能。



2. Data Bindings 面板

单击 Data Bindings 按钮即可打开 Data Bindings 面板，如图 2-21 所示。

Data Bindings 面板的主要作用是用来完成数据库操作。

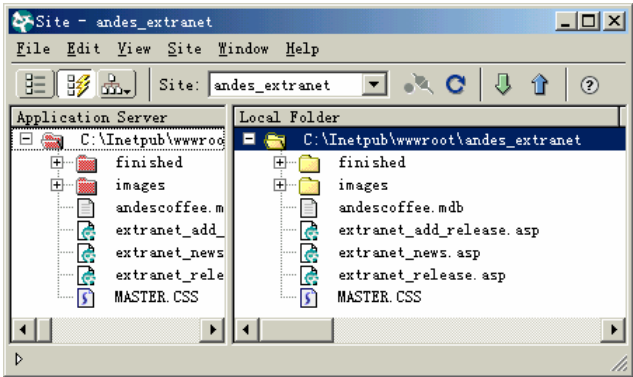


图 2-20
Site 窗口

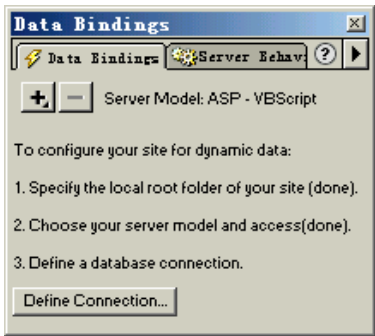


图 2-21
Data Bindings 面板

3. Server Behaviors 面板

单击 Server Behaviors 按钮可以打开 Server Behaviors 面板，如图 2-22 所示。

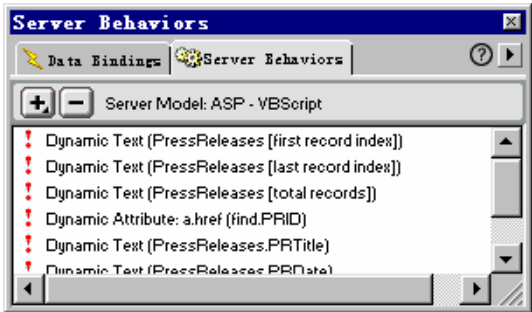


图 2-22
Server Behaviors 面板

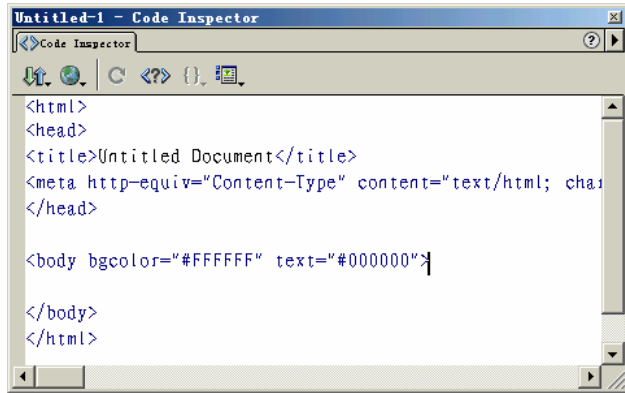
Server Behaviors 面板的主要作用是用来增加一些服务器端行为。

4. Code Inspector 面板

单击 Code Inspector 按钮可以打开 Code Inspector 面板(又称为 HTML 源代码检查器)，如图 2-23 所示。

图 2-23

Code
Inspector
面板



在文档窗口中进行所见即所得的编辑时，在 Code Inspector 面板中可以立即显示编辑中的网页的 HTML 源代码，可以通过 Code Inspector 面板随时查看相应的 HTML 代码，也可以直接在 Code Inspector 面板中修改代码，文档窗口中也会随之做相应的改变，方便熟悉 HTML 或其他脚本语言的使用者来编写网页。

5. 改变启动面板的方向

默认状态下，启动面板是水平放置的，单击面板右下角“垂直放置”标记，可以将启动面板变为垂直放置，同时该“垂直放置”标记变为“水平放置”标记；在垂直状态下，单击“水平放置”标记，又可以恢复启动面板的水平状态，如图 2-24 所示。

图 2-24

改变启动
面板的水
平和垂直
状态



2.4.2 对象面板

对象面板上包含一些按钮，在网页编辑过程中，单击这些按钮，可以往网页中添加相应的对象，如图像、表格、框架和各种其他媒体等。

对象面板的空间是有限的，为了能够充分利用有限的面板空间，针对不同类型的对象，对象面板采用了多层结构，将不同类型的对象按钮放置于不同的浮动面板层中。

单击对象面板上的向下箭头 (如图 2-25 所示)，会打开一个下拉菜单，显示对象面板上所包含对象的类型，对象面板默认包含 8 个嵌板：Character(字符对象)、Common (常用对象)、Forms(表单对象)、Frames(框架对象)、Head(头部元素)、Invisibles (不可见元素)、Live(活动对象) 和 Special(特殊对象)。默认状态下显示的是 Common (常用) 类型，如果从该下拉菜单中选择其他类型，则在对象面板上会显示相应类型的按钮。用户还可以修改面板中的任何对象，或创建自己的对象。

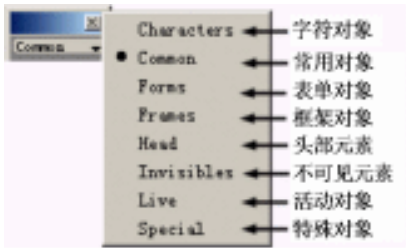


图 2-25
展开对象
面板的嵌
板

另外，和 DW UltraDev 1.0 相比，DW UltraDev 4 在对象面板的下方还增加了两个控制项，如图 2-26 所示。其中 Layout 用来控制页面的布局，它包括了“ 画布局单元 ”和“ 画布局表 ”这两个控制；View 用来控制页面中的视图显示方式，“ 标准视图 ”表示的就是普通的方式，而“ 布局视图 ”表示的是以一种更形象、更易于用户控制的方式，只有选择了此项，上面介绍的 Layout 中的控制项还是可用的，否则它们是变灰状态，不可用。

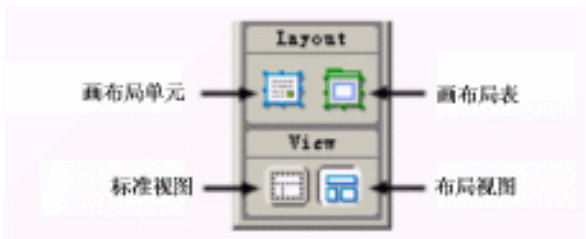


图 2-26
DW
UltraDev 4
新增的控
制项

1. 字符对象

字符对象面板中包含了一些特殊字符，如版权符号、弯曲的双引号和商标符号等，如图 2-27 所示。

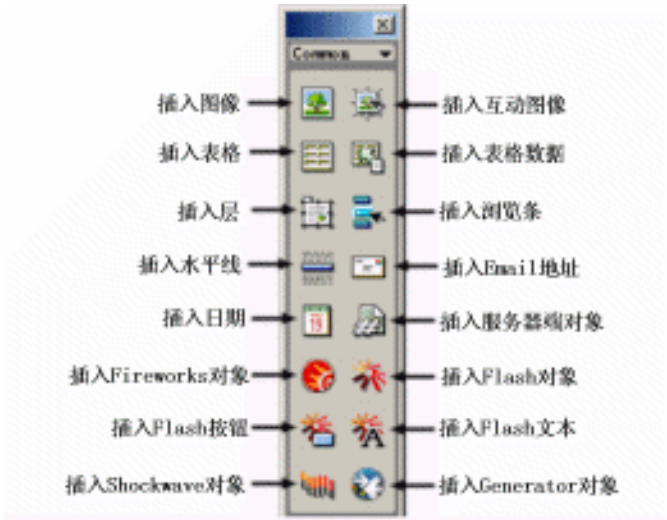
图 2-27
字符对象
面板



2. 常用对象

常用对象面板中包含了最常用的对象，如图像、表格和层等，如图 2-28 所示。

图 2-28
常用对象
面板



3. 表单对象

表单对象面板中包含了创建表单和表单元素的按钮，如图 2-29 所示。

4. 框架对象

框架对象面板中包含了常用框架结构，如图 2-30 所示。



图 2-29
表单对象
面板

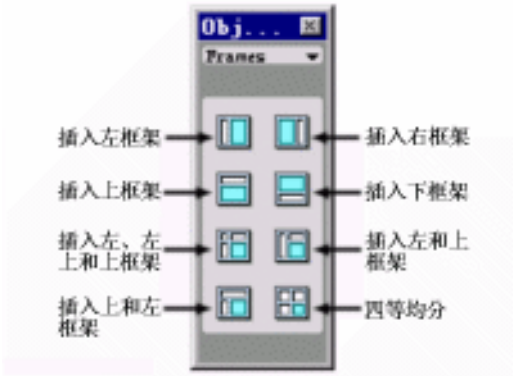


图 2-30
框架对象
面板

5. 头部元素对象

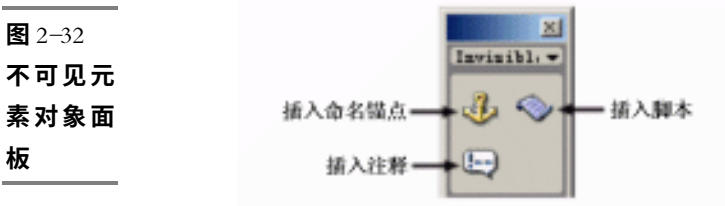
头部元素对象面板中包含了添加不同头部元素的按钮，如 Meta、Keywords 和 Base 标记等，如图 2-31 所示。



图 2-31
头部元素
对象面板

6. 不可见元素对象

不可见元素对象面板中包含了创建文档窗口中不可见对象（如命名锚点）的按钮，如图 2-32 所示。选择“View / Invisible Elements”菜单命令可显示标记这些对象位置的图标。单击文档窗口中不可见元素的图标可选择该对象并改变其属性。



7. 活动对象

在 DW UltraDev 4 中增加了对活动对象的支持,从而可以加速 Web 应用的开发进程。使用活动对象可以使得用户只用一个操作步骤就可以创建一些高级的页面组件。图 2-33 所示的就是活动对象面板，其中包括了 5 个相关的活动对象。



8. 特殊对象

特殊对象面板如图 2-34 所示，其中包含了 3 个特殊对象：Java Applet 对象、Plugin 插件对象和 ActiveX 对象。



2.4.3 属性面板

属性面板主要用于显示在文档窗口中所选中元素（如页面上的文本、图像、表格、层等等）的属性，并允许用户在属性面板中对元素属性直接进行修改。属性面板几乎是最常用的一个面板，因为无论要编辑哪个对象的属性，都要用到它。而且，它的内容也随着选择的不同对象而改变，因此操作起来相当方便。例如，当我们在页面中选定文本时，属性面板的显示如图 2-35 所示，其中可以看到文本的格式、字体、大小、颜色等项；而当我们在页面中选定一幅图像时，属性检查器的显示如图 2-36 所示，其中显示图像的缩图、大小、宽度和高度、源文件名、对齐方式等项。

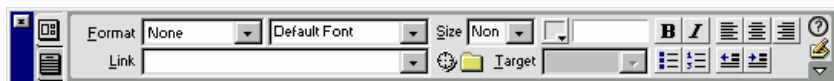


图 2-35
选中文本
时的属性
面板

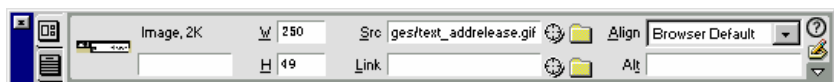


图 2-36
选中图像
时的属性
面板

在属性面板的右下角，通常都有一个下三角形标记，单击该标记，可以展开属性面板，显示更多的属性设置内容。当属性面板被展开时，右下角的标记变为上三角形标记，单击该标记，又可以重新折叠属性面板，恢复原先的外观，如图 2-37 所示。

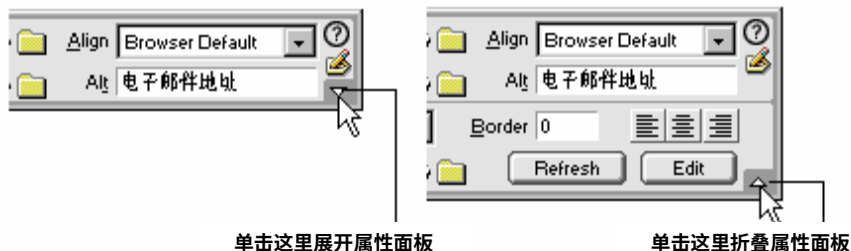


图 2-37
属性面板
的展开和
折叠标记

要了解属性检查器中当前显示的各项属性的详细信息，单击右上角的帮助图标(?)。

2.5 控制浮动面板

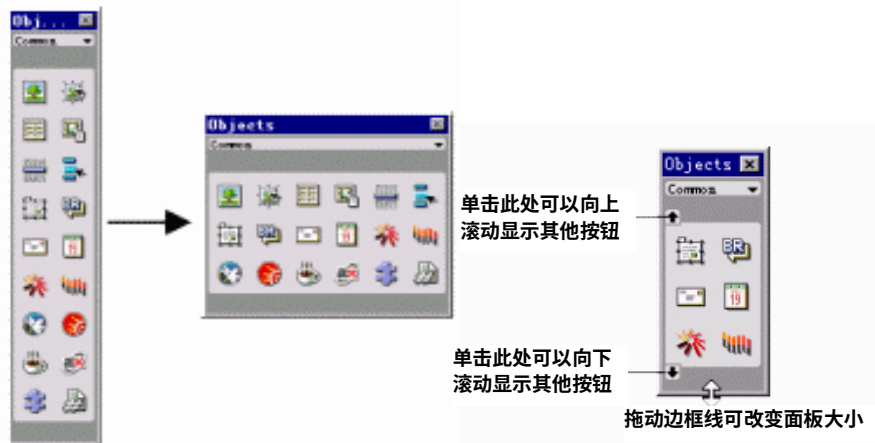
1. 浮动面板的移动和重设大小

针对不同的控制对象，还有其他一些浮动面板：样式面板、库面板、行为面板、历史面板、时间链面板、模板面板、框架面板、图层面板等，这几个面板针对不同的对象起作用。在后面的章节中我们将分别介绍。

和普通的 Windows 应用程序没有什么不同，通过拖动浮动面板的标题栏，就可以将浮动面板在桌面上移动，浮动面板的位置并不局限于 Dreamweaver 的文档窗口内，实际上，它可以移动到 Windows 桌面上的任何位置。可以将浮动面板完全放置在文档窗口的视野之外，这样既不影响文档的编辑，也不影响浮动面板的使用。

除了启动面板和属性面板，其他大多数的浮动面板，包括对象面板和可停靠浮动面板，其大小都可以任意调整，通过拖动浮动面板的边框，就可以改变其大小。在改变对象面板大小时，当对象面板的大小不足以容纳其上的元素时，会在左下角或右上角出现箭头标记，单击该标记即可显示其余元素，如图 2-38 所示。

图 2-38
改变对象
面板的大
小



如果屏幕上的浮动面板的位置变得过于混乱，甚至超出了桌面范围，不便操作，这时可以打开 Window(窗口)菜单，选择‘ Arrange Panels ’(重新排列浮动面板)命令，DW UltraDev 4 会将所有显示的浮动面板在桌面上重新排列，往桌面的左、上、下方向靠齐，便于用户调整和使用。

2. 拆分和组合浮动面板

在 DW UltraDev 4 对浮动面板的管理中,最具特色的是采用可停靠浮动面板的方式。利用可停靠浮动面板,可以将众多浮动面板窗口综合放置在一个窗口中,组合成为标签的形式,不仅有效地节省屏幕空间,而且也更便于使用。图 2-39 中显示了一个带有 4 个标签的可停靠浮动面板,其中 4 个标签实际上都是单独的浮动面板。单击其中的任何一个标签,即可显示对应的面板。

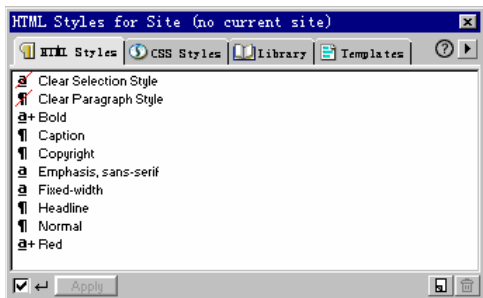


图 2-39
带有多个
标签的可
停靠浮动
面板

可停靠浮动面板是可以进行人工组合和拆分的,要将某个以标签形式出现的浮动面板从组合中拆分出来,只要拖动其标签,将之移动到可停靠浮动面板之外就可以了,如图 2-40 所示,这时可停靠浮动面板中的一个浮动面板就被拆分出来,如图 2-41 所示。

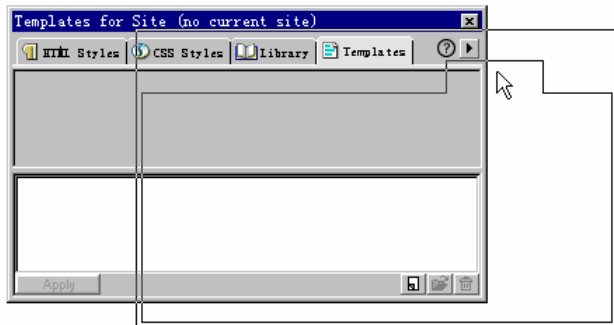
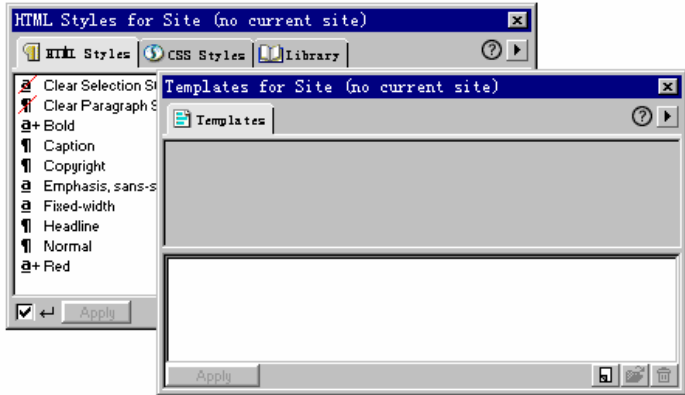


图 2-40
拆分可停
靠浮动面
板



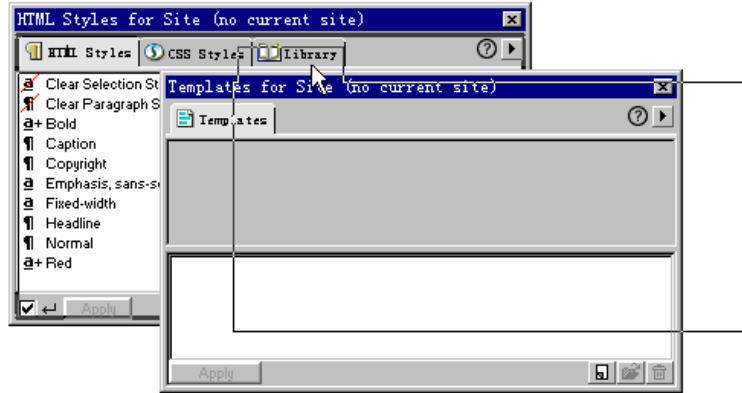
被拆分出的浮动面板仍然是可停靠浮动面板,换句话说,可以以标签形式出现的所有浮动面板,其本身都是可停靠浮动面板。在图 2-41 中,被拆分出的 Templates 浮动面板虽然是单独的窗口,但它带有一个标签(就是它本身),这表明它仍然是可停靠的。换句话说,它也可以接纳其他的浮动面板,组合多个标签的形式。例如,我们可以从原先的可停靠浮动面板中再次拆分出 CSS Styles 面板,然后将它同 Templates 面板组合成为一个带有两个标签的可停靠浮动面板。

图 2-41
拆分出的
浮动面板



如果希望将某个浮动面板同其他的浮动面板组合成多个标签的形式，则可以拖动该浮动面板的标签(可停靠浮动面板至少都会带有一个标签，也即它本身)，移动到目标浮动面板的窗口范围内，如图 2-42 所示，当目标窗口显示粗边框时释放鼠标，单独的浮动面板就被组合成标签形式。

图 2-42
合并可停
靠面板



如果窗口中同时显示了两个可停靠浮动面板，而希望将某个面板中的标签移动到另一个面板中，可以将该标签直接从源面板拖动到目标面板上。因为对象面板没有标签，所以不能拖曳对象面板到其他面板上，但可以拖曳其他面板的标签到对象面板上来组合它们。

3. 显示和隐藏浮动面板

如果不希望显示某个浮动面板，可以通过单击面板窗口上的“关闭”按钮来关闭它，如图 2-43 所示。需要注意的是，不同的浮动面板，关闭按钮的位置或许也不同，通常它

出现在浮动面板标题栏的右端或上端。

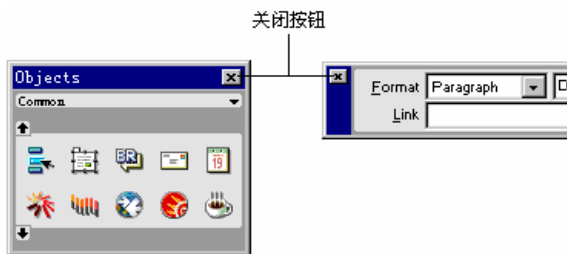


图 2-43
浮动面板
上的“关
闭”按钮

另外，通过 Window（窗口）菜单同样可以进行浮动面板的隐藏和显示操作，打开菜单，选中相应的菜单项，即可显示相应的浮动面板，被显示的浮动面板左端带有“√”的标记。再次选择带有“√”的菜单项，则会将相应的面板重新隐藏。

用户可以从 Window（窗口）菜单的编排上看出众多浮动面板的区别，启动面板、对象面板和属性面板不是可停靠浮动面板，它们处于菜单的第一条分隔线上，而 Data Bindings 面板和 Server Behaviors 面板是 DW UltraDev 4 新增加的（相对于 Dreamweaver 来说），它们处于菜单的第二条分隔线上，其他的可停靠浮动面板都位于菜单的中间区域。

在实际应用过程中，通常会在屏幕上显示很多的浮动面板，要将这众多的浮动面板一一关闭是很麻烦的事情。事实上，用户之所以要关闭浮动面板，主要是为了节省文档窗口的空间，在输入完毕前，很可能还要继续使用面板，这时可以利用 DW UltraDev 的暂时隐藏浮动面板特性，打开 Window（窗口）菜单，选择“Hide Panels”（隐藏浮动面板）命令，即可将所有显示的浮动面板暂时隐藏，屏幕上只留下文档窗口，便于文档的编辑，如果希望重新显示原先屏幕上显示的浮动面板，可以再次打开 Window（窗口）菜单，选择“Show Panels”（显示浮动面板）命令。

通常用户可能更喜欢使用热键，而不是操作菜单，隐藏浮动面板命令的热键是 F4。在操作过程中，按下 F4 键，就可以将所有屏幕上原先显示的浮动面板隐藏起来，再次按下 F4 键，又可以重新显示所有原先显示的浮动面板。

2.6 使用其他界面元素和工具

2.6.1 使用状态栏

可不要小看 DW UltraDev 4 文档窗口下方的状态栏，正确地使用状态栏同样也可以大大方便我们的操作。状态栏的结构如图 2-44 所示。

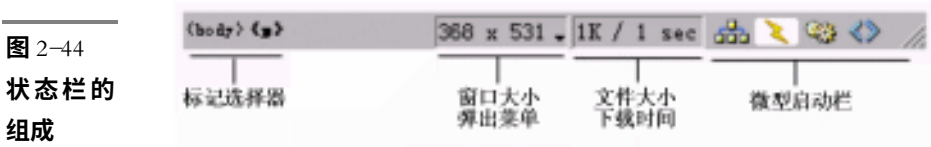


图 2-44
状态栏的
组成

1. 标记选择器

标记选择器主要有两个用途：显示当前插入点位置的 HTML 源代码标记和选中标记在文档中对应的内容。每当用户在文档窗口中对文档内容进行了格式化时，标记选择器中都会显示相应的标记。例如，如果在文档窗口中将一段文字加粗，这实际上是将该段文字的两端分别放上和标记，当用户将插入点放置到该段文字中时，标记选择器上就会显示标记。

如果希望选中某个标记在文档中对应的内容，可以直接在标记选择器上单击该标记按钮。例如，文档主体通常包含在<body>和</body>标记中，单击标记选择器上的<body>按钮，即可选中文档所有位于<body>和</body>标记中的内容。

在标记选择器上单击鼠标右键，会出现如图 2-45 所示的菜单，允许用户对标记进一步处理，具体方法将在后面章节中介绍。



图 2-45
“ 标记选
择器”上的
右键快捷
菜单

2. 窗口大小弹出菜单

在窗口大小弹出菜单区域中会显示当前文档窗口的尺寸大小，以像素作为单位。当拖动文档窗口边框改变窗口大小时，会看到状态行上窗口大小弹出菜单区域中显示的数值也会相应发生变化。

窗口大小弹出菜单主要用于设置文档窗口和显示器屏幕之间的对应关系，单击该菜单区域任意位置，即可打开该菜单，如图 2-46 所示。该菜单项左边显示的是文档窗口的大小，右边显示的屏幕的大小，根据显示器屏幕大小选择相应的菜单项，就会发现文档窗口大小相应发生变化。

例如，如果从窗口大小弹出菜单中选择“ 760×420 (800×600, Maximized) ”，则表明在分辨率为 800×600 像素的显示器上，将浏览器窗口最大化即可完全观看当前文档窗口大小范围内所制作的网页。在这里 DW UltraDev 4 计算了浏览器边框的尺寸，并将它

从显示器尺寸中剔除，所以文档窗口的大小总是比设置的屏幕分辨率要小一些。

如果用户希望定制自己需要的文档窗口大小，可以从菜单中选中“ Edit Sizes”(编辑大小) 项，打开一个对话框，自行编辑需要的尺寸，该对话框实际上就是 DW UltraDev 4 的参数设置对话框，请参见 2.7 节。

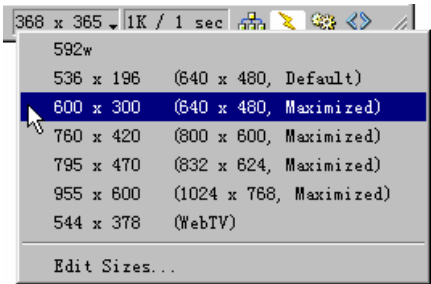


图 2-46
窗口大小
弹出菜单

3. 文件大小和下载时间

在状态行上“ 文件大小和下载时间 ”区域显示当前编辑文档的大小，和该文档在 Internet 上被完全下载的时间，针对不同的下载速率，下载时间当然也不相同，默认状态下，系统假设用户使用 56Kb/s 的调制解调器下载该网页。

4. 微型启动栏

微型启动栏就是启动面板的一个缩小板，在启动面板被关闭的情况下，也能帮助用户快速使用其上的功能。如果用户始终将启动面板显示在桌面上，则状态行上的微型启动栏显得很多余，则也可以将之从状态行上移去，请参见 2.7 节。

2.6.2 使用工具栏

DW UltraDev 4 的工具栏及其相关说明如图 2-47 所示。

要想显示或者隐藏工具栏，可以通过选择“ View / Toolbar ”菜单命令来实现。DW UltraDev 4 的工具栏上包含了三个用来在不同的视图(代码视图、设计视图以及代码和设计视图) 之间进行切换的按钮，这一点在前面我们已经介绍过。

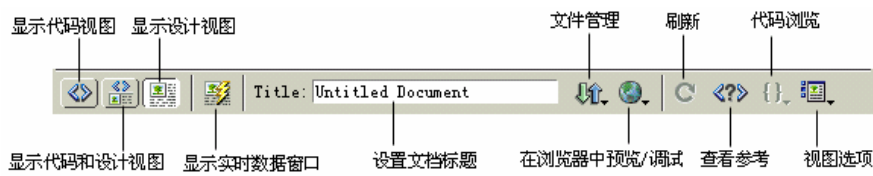


图 2-47
工具栏及
相关说明

除此之外，工具栏上还包含了许多其他常用的按钮。下面分别进行简单地介绍。

- 如果要切换到实时数据窗口，请单击 Show Live Data 按钮，再次单击该按钮，又可切换回原来的文档窗口。
- 如果要为页面设置标题，请在 Title 文本框中直接输入即可。在以前的版本中，要想设置或者修改页面的标题，要经过许多步骤，DW UltraDev 4 意识到了这一缺点，将这一功能集成到了工具栏上，大大方便了用户。
- 单击 File Management 按钮可以打开文件管理下拉菜单。某一个按钮是否包含有下拉菜单，可以从其右下角是否有一个向下的三角形标志中看出来。
- 单击 Preview/Debug in Browser 按钮也可以打开一个下拉菜单，其中包含了在浏览器中进行预览、在浏览器中调试等菜单命令。
- 单击 Refresh Design View 按钮可以刷新设计视图。此按钮仅仅当打开了代码视图或者同时打开了代码视图和设计视图，并且在代码视图中作了修改时才有效。
- 如果在编辑页面的过程中想查看相关的参考信息，请单击 Reference 按钮，就会打开 Reference 面板，其中包含了与 HTML、CSS 和 JavaScript 相关的参考信息。
- 单击 Code Navigation 按钮可以打开一个下拉菜单，其中包含了一些与设置断点相关的菜单命令。
- 单击 View Option 按钮可以打开一个下拉菜单。对于三个不同的视图来说，这个下拉菜单中包含的菜单选项也是不同的。由此可见 DW UltraDev 4 的智能化。借助这个下拉菜单可以极大地方便用户的操作。

2.6.3 使用快捷菜单

所谓快捷菜单，就是指当用户在应用程序界面上单击鼠标右键时出现的弹出菜单。目前，越来越多的高级设计软件都具有辅助设计用的快捷菜单，DW UltraDev 4 当然也不例外。快捷菜单的最大功能，就是可以简化用户的操作步骤，同时也减少了用户频繁移动鼠标时所产生的疲倦感。例如，网页上的一行文字，如果要做复制与粘贴的动作，最基本的操作步骤是首先选取欲处理的文字，然后选择“Edit / Copy”菜单命令将其放置到剪贴板中，并使用“Edit / Paste”命令粘贴到新的地方。同样的动作，也可以使用快捷菜单来完成。首先也是选取文字，然后将鼠标移到文字上，单击鼠标右键，即可在弹出的快捷菜单中找到相同的命令。

更重要的是，DW UltraDev 4 中的快捷菜单是上下文相关的，也就是说，它能根据当前操作对象的不同，弹出不同的快捷菜单，非常具有“智能化”。

2.6.4 使用网格

除了上面介绍的各种面板、状态栏、菜单之外，其他辅助设计工具还有“网格”和“标尺”。

一个完全空白的网页底稿，对于网页初学者来说是一项重大的挑战，而纵横交错的网格，除了有助于初学者的学习使用之外，对老手而言，也是对物体精确位的重要工具之一。

缺省情况下网格并没有显示在文档窗口中，启动网格的方法是：选取菜单命令 View / Grid / Show Grid”即可，如图 2-48 所示。这时 Show Grid 菜单项前面会出现一个“√”标记。如果要关闭网格，只需再一次选择 ShowGrid 菜单项，则其前面的“√”标记消失，同时网格也消失。

网格除了可以帮助用户在创建对象时辅助定位以外，还有一个重要的功能就是“自动吸附”(snap)。如果打开了“自动吸附”功能，则系统会在文档窗口中均匀放置许多吸附点，当用户创建、移动对象或者改变对象的大小时，系统都会自动地把该对象移动到最近的吸附点上，就像有磁石把对象吸过去一样。

打开自动吸附功能的方法是，选取 View / Grid / Snap To Grid”菜单命令，于是 Snap To Grid 菜单项前面会出现一个“√”标记，表明自动吸附功能已经激活。如果要关闭自动吸附功能，只需再一次选择 Snap To Grid 菜单项，则其前面的“√”标记消失，表明自动吸附功能已经禁止。



自动吸附功能与网格是否在文档窗口中可见是无关的。也就是说，即使网格没有显示在文档窗口中，只要打开了自动吸附功能，该功能就有效。

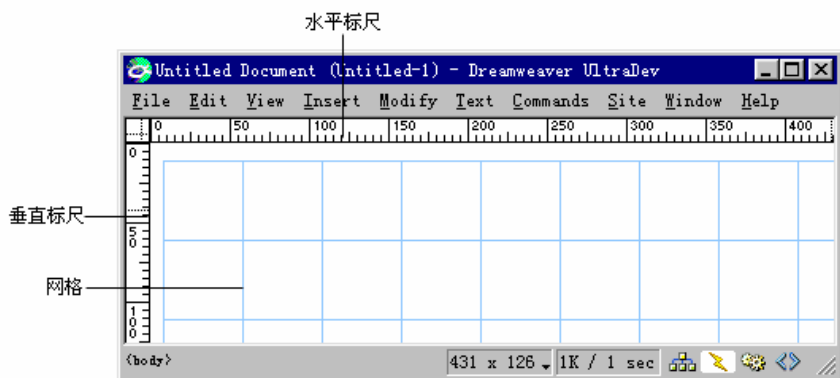
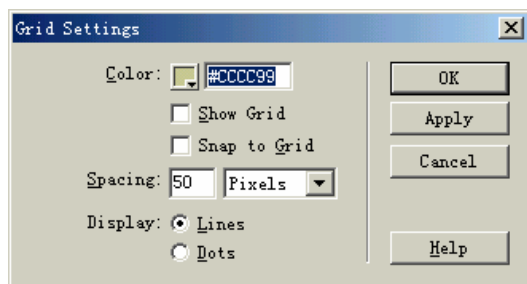


图 2-48
一个带有
标尺和网
格的文档
窗口

系统预设为每间隔 50 像素出现一条网格线，颜色为蓝色，并且每 5 个像素放置一个吸附点。如果需要修改，请选择 View / Grid / Edit Grid”菜单命令，则会出现如图 2-49 所示的 Grid Settings 窗口。

图 2-49

Grid
Settings 窗
口



Color 域用来确定网格线的显示颜色，缺省颜色是亮蓝色。如果需要修改，可以单击  并选取合适的颜色，或者在文本框中直接输入颜色的 16 进制值。

Show Grid 复选框的作用是控制网格是否可见，它的功能等同于菜单项 View / Grid / Show Grid”。Snap to Grid 复选框的作用是控制是否打开自动吸附功能，其功能等同于菜单项 View / Grid / Snap To Grid”。

Spacing 域用来设定网格线之间的距离，用户可以在文本框中输入固定的值，然后在下拉列表中选择度量单位。度量单位有三种，分别是：pixels(像素)、inches(英寸)和 centimeters(厘米)。注意，网格线之间的最小距离是 25 pixels。

Display 域用来确定网格是以直线形式(Lines) 显示还是以点线形式(Dots) 显示。

2.6.5 使用标尺

标尺与网格通常都是交互配合使用的。标尺分为水平标尺和垂直标尺，如图 2.48 所示。标尺工具会以系统所设定的屏幕分辨率(如 800×600 像素、1024×768 像素等) 作为 DW UltraDev 4 文档窗口最大化显示时的最大刻度，每 50 像素为一大刻度，每 5 像素为一小刻度。

缺省情况下标尺是隐藏起来的。若要显示标尺，只需选择“ View / Rulers / Show ”菜单命令即可。预设的标尺单位为 Pixels(像素)，在“ View / Rulers ”子菜单中，用户还可以选择 Inches(英寸) 和 Centimeters(厘米) 作为标尺单位。

缺省的标尺原点为窗口的左上角。当然，用户也可以自己设定原点的位置。读者应该注意到，在文档窗口左上角、水平标尺和垂直标尺的交错处有一个  方框。我们只需从该方框中拖曳一个十字线到文档窗口中的指定位置，即可改变标尺的原点，如图 2-50 所示。

如果想还原标尺原点位置，有以下三种方法：

- ※ 双击鼠标左键：只需在方框内双击鼠标左键，即可还原标尺原点位置。
- ※ 使用快捷菜单：在方框内单击鼠标右键，弹出一个快捷菜单，如图 2-51 所示。然后在快捷菜单中选择 Reset Origin 命令，即可还原原点位置。
- ※ 使用菜单命令：选择菜单命令“View / Rulers / Reset Origin”也可以还原标尺原点位置。

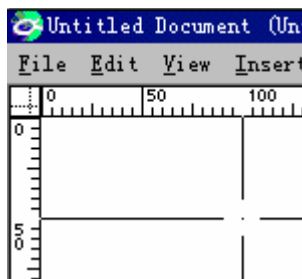


图 2-50
改变标尺
原点位置

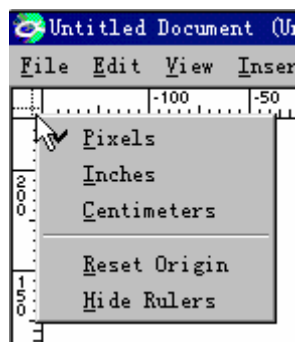


图 2-51
使用快捷
菜单还原
标尺原点
位置

2.7 设置 DW UltraDev 4 的界面属性

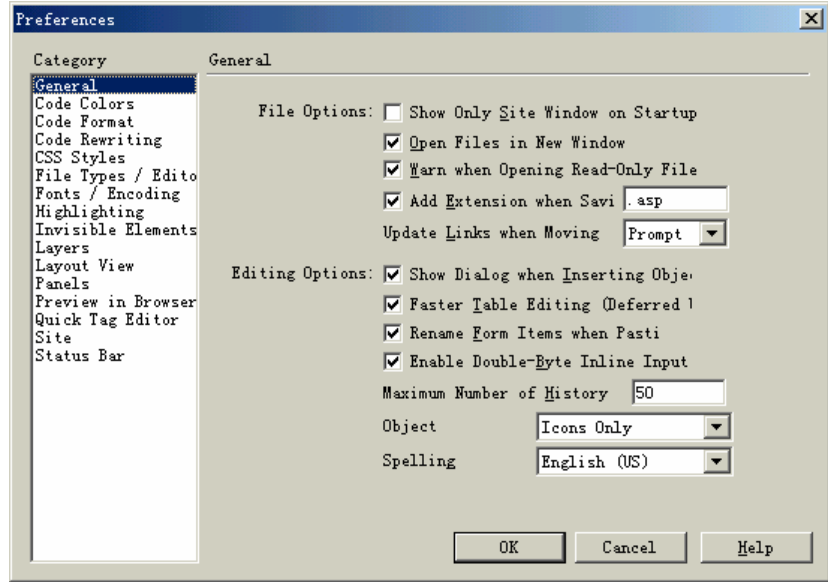
2.7.1 打开属性设置窗口

在文档窗口中选择“Edit / Preferences”菜单，即可打开 Preferences 属性设置窗口，如图 2-52 所示。

Preferences 窗口的作用是设置 DW UltraDev 4 的界面外观(如面板的颜色、大小等)和它的一些默认值(如<P>、<PRE>标记所包含文本的默认的字大小等)。从图 2-52 中可以看到 Preferences 窗口分为左右两部分，左边为一个列表，包括状态栏(Status Bar)、

字体样式(Fonts/Encoding) 等项, 右边是左边当前选取的列表项的具体选项, 它根据左边选定的列表条目切换。例如在图 2-52 中, 左边选定 General 项, 则右边显示对于 General 这个条目所能作的选择, 如面板颜色(Color Scheme) 等。因为这个列表条目及它们的所有选项内容众多, 在本节仅说明与 DW UltraDev 4 总体功能有关的列表项, 其他与具体功能有关的项将在介绍具体功能时说明, 在这里读者知道 Preferences 面板的用途和使用方法就可以了。

图 2-52
Preferences
属性设置
窗口



2.7.2 设置 General 属性

General 属性包括对文档窗口、对象面板、属性面板、站点管理方式等的设置, 因为这些内容是 DW UltraDev 4 的主要内容, 图 2.52 显示了 General 属性的有关选项, 以下简要介绍部分选项。

- ✱ File Options 域中的 4 个复选框用来控制与文件相关的操作。其中, 如果选择了 Show Only Site Window on Startup 复选框, 则当用户启动 DW UltraDev 4 时, 系统不再打开文档窗口, 而只打开站点窗口, 同时对应的对象面板和属性面板都变成灰色禁止的状态; 如果选择 Open Files in New Window 复选框, 则 DW UltraDev 4 会在一个新窗口中打开新的文档, 而不会在当前文档窗口中打开新文档并替换现有文档; 如果选择了 Warn when Opening Read-Only File 复选框, 则当打开一个具有“只读”属性的文档时, 系统会弹出一个警告窗口, 用户可以选择取消“只读”属性; 如果选择了 Add Extension when Saving 复选框, 则保存网页文件时自动增加文件扩展名,

如“ .asp ”等，具体用哪一个扩展名可在此选项末尾的输入框内指定。

- ※ Update Links when Moving 决定当站点中的文件被移动、删除、重命名时，站点中文件的链接关系如何改变。打开该选项的下拉列表，将列出三个选择：Always、Never、Prompt。Always 意为自动修改链接，Never 为不修改，Prompt 为显示修改提示，提醒网页制作者去修改。默认值为 Prompt。
- ※ Editing Options 域中的 4 个复选框用来控制与编辑相关的操作。如果选择了 Show Dialog when Inserting Object 复选框，则当用户往文档中插入对象时，系统会弹出一个相应的对话框以使用户能够设置相应的属性，否则插入对象时不会弹出任何对话框，用户只能借助属性面板来进行设置；如果选择了 Faster Table Editing (Deferred Update) 复选框，则使得在表格中进行编辑的操作速度加快，这是通过延迟表格宽度和高度的更新来实现的，也就是说，当用户在表格中进行编辑操作时，表格并不会马上更新，除非用户在表格外单击了一下鼠标，或者在编辑过程中按下了 Ctrl+Space 组合键，表格才会更新；如果选择了 Rename Form Items When Pasting 复选框，则当用户从其他文档中复制了一个表单项并粘贴到当前文档中时，系统会自动修改该表单项的名字；如果选择了 Enable Double-Byte Inline Input 复选框，则 DW UltraDev 4 就可以支持在文档窗口中直接输入双字节的字符（如中文或日文等）。
- ※ Maximum Number of History 文本框中的数字决定了历史面板中可以保留并显示的历史记录数。通常情况下，默认的 50 已经能够满足需要了。
- ※ Object Panel 下拉列表：在此可以定义对象面板的外观，此选项有三种选择：Icons and Text、Icons Only、Text Only，分别表示对象面板使用图标和文本、仅用图标、仅用文本，图 2-53 所示的左、中、右三图就分别表示了上述三种情况下的对象面板。
- ※ Spelling Dictionary 下拉列表可以选择用于拼写检查的字典。DW UltraDev 4 中只有英文字典，而没有中文字典，也就是说，只能检查英文拼写情况，而无法对中文进行检查。

2.7.3 设置 Panels 属性

通过 Panels（浮动面板）设置项可以指定哪个面板和检查器总是显示在文档窗口前面，哪个可以被文档窗口遮挡，也可以设置哪个面板和检查器显示在启动面板中。显示在启动面板的所有项目也同时显示在状态栏的微型启动面板中。

浮动面板的设置步骤如下。

- ④ 在 Preferences 窗口中，从 Category（类型）列表中选择 Panels（浮动面板）项，则设置窗口变成如图 2-54 所示的样子。

图 2-53
对象面板
的三种显
示风格

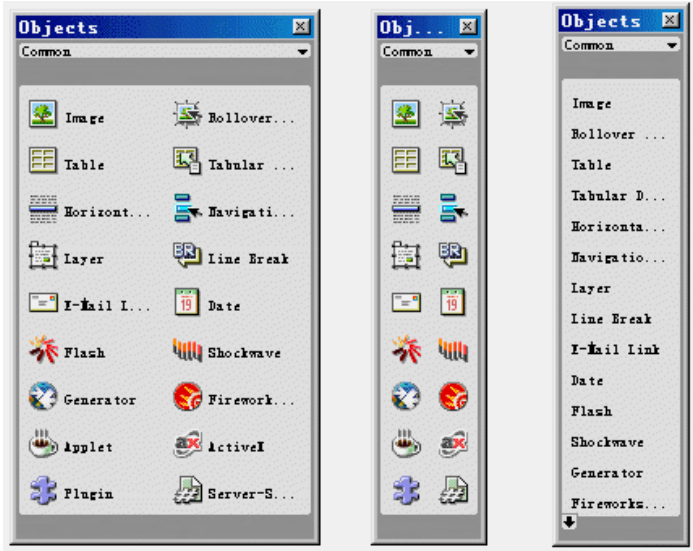
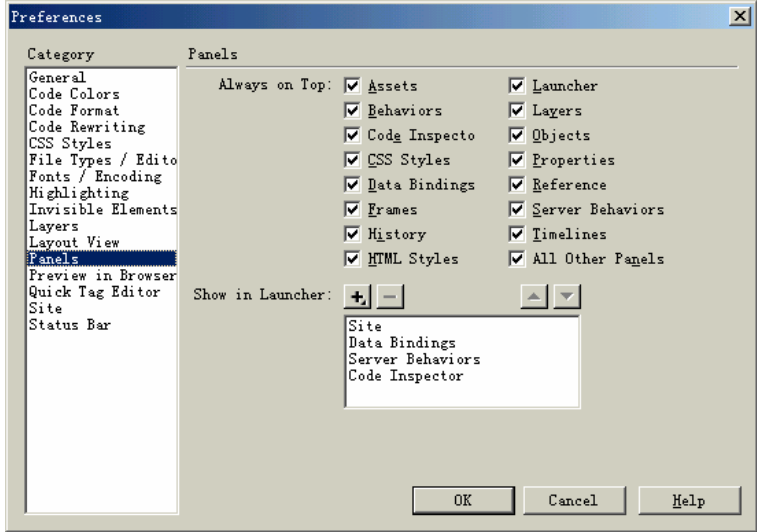


图 2-54
设置
Panels



- 2 在 Always on Top(总是在前面) 栏中, 把你希望显示在文档窗口后面的面板的复选框取消。例如, 如果你想使文档窗口遮挡 HTML 源码检查器, 请取消 Code Inspector 复选框的选择。这样, 只有当 HTML 源码检查器处于活动状态时, 才会显示在文档窗口的前面。在默认状态下, 所有的窗口、面板、检查器被选定显示在文档窗口的前面。
- 3 在 Show in Launcher(在启动面板中显示) 栏中, 单击加号按钮 $+$, 将出现如图 2-55

所示的菜单，其中变灰的菜单项就是已经显示在启动面板中的按钮，只需单击其他菜单项就可以增加要在启动面板中显示的面板或检查器；如果要删除在启动面板中显示的面板或检查器，只需在列表中选择该项，然后单击减号按钮即可；如果要调整面板或检查器在启动面板中的排列顺序，只需选择该项，然后单击上三角按钮和下三角按钮即可。



图 2-55
在启动面
板中增加
项

 设置完成后，单击 OK 按钮即可生效。

2.7.4 设置 Fonts / Encoding 属性

在 Preferences 窗口中，从 Category（类型）列表中选择 Fonts/Encoding 项，则设置窗口变成如图 2-56 所示的样子。在此可以改变 DW UltraDev 4 默认的字体和字大小等。

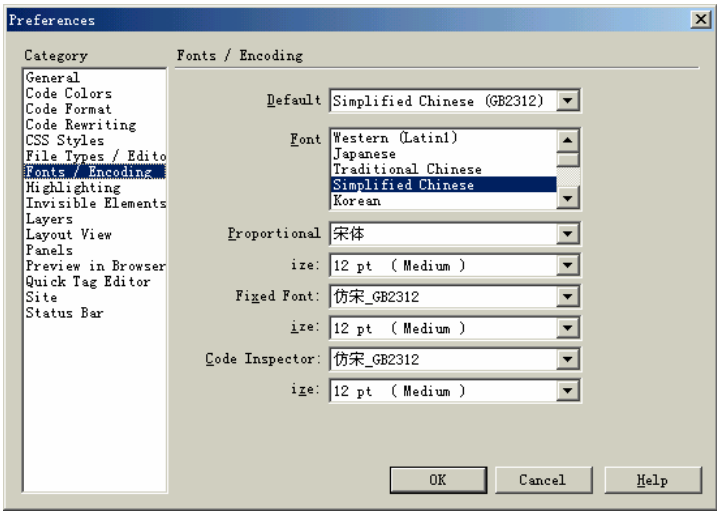


图 2-56
Preferences
面板的
Font/Enco-
ding 的初
始页

- ※ Default 和 Font 域：主要作用是设置中文环境。在网页中，不同的语言文字，应该有不同的文字编码方式，特别是对于采用双字节文本的国家，不同的文字内码，显示出来的内容也绝不相同。例如，中国大陆使用的是简体汉字，采用 GB2312 内码（通称国标码），对于港、澳、台地区使用的是繁体汉字，则采用 BIG5 内码（通称大五码）。如果网页的编码方式没有设置正确，则在浏览器中不能正确显示这些文字。虽然 DW UltraDev 4 没有中文版本，但是它可以在中文的操作系统下正确使用，也完全支持中文双字节的环境，DW UltraDev 4 甚至会自动检测用户操作系统的语言，自动设置正确的文字编码方式，在默认情况下，无需任何设置，即可进行正确的中文网页创作。也可以指定其他的编码方式，只需在 Default 下拉列表中选择其他值即可。对使用简体中文版本 Windows 操作系统的读者来说，请确保 Default 下拉列表中选中的是“Simplified Chinese(gb2312)”（简体中文 gb2312），并在 Font 列表中选择“Simplified Chinese”项，则创建的网页就可以在中文简体环境下的浏览器中正确浏览。
- ※ Proportional：此选项定义 HTML 文件中包含在<P>...</P>标记内的文本的默认字体和字大小，DW UltraDev 4 默认字体是宋体，字大小是 12pt，如果需要将 DW UltraDev 4 的初始默认字体改为“隶书”，大小改为 18pt，只需打开 Proportional 的下拉列表，选择“隶书”项，然后再打开 Proportional 选项下的 Size 下拉列表，选择字大小为 18pt，按 OK 键即可。
- ※ Fixed Font：此选项定义 HTML 文件中包含在<PRE>...</PRE>标记内的文本的默认字体和字大小，用法类似 Proportional。
- ※ Code Inspector：此选项定义 HTML 源代码检查器中的文本的字体和字大小，用法类似 Proportional。

2.7.5 设置 Status Bar 属性

在 Preferences 窗口中还可以修改状态栏的属性，从 Category（类型）列表中选择 Status Bar 项，则设置窗口变成如图 2-57 所示的样子。下面介绍各个属性的设置方法。

- ※ Window Sizes：在 Window Sizes 旁是各种文档窗口大小表，一行为一种大小，每行的数值和描述可用鼠标点击后随意修改，还可增添行，输入新的窗口大小。在文档窗口的状态栏按文档窗口大小选择钮，弹出的就是这张表。用鼠标在“Window Sizes”（窗口大小）区域相应位置单击，激活编辑状态，然后输入需要的数字和描述信息，即可定制窗口大小弹出菜单。

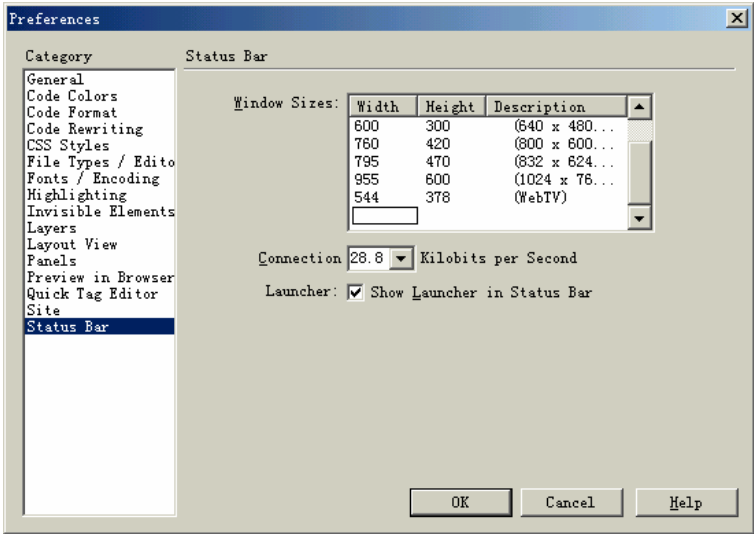


图 2-57
Preferences
面板的
Status Bar
页

- ✧ Connection：在下拉列表中选择相应的选项，即可设置下载速度，DW UltraDev 4 会根据这里假设的下载速度重新计算当前文档的下载时间。
- ✧ Launcher：如果选中“ Launcher ”（启动）后面的复选框，即可在状态行上显示微型启动栏；清除该复选框，则不在状态行上显示微型启动栏。

第 3 章 体验 DW UltraDev 4

本章概要：

- ☒ 定义一个本地站点，并确定该站点的服务器技术
- ☒ 定义一个远程站点，并配置 Live Data(实时数据) 窗口
- ☒ 创建数据库链接
- ☒ 创建动态 ASP 页面

本章我们将通过创建一个简单的 Web 应用，让读者体验一下 DW UltraDev 4 的强大功能。

3.1 定义本地站点

在使用 DW UltraDev 创建 Web 应用之前，首先要定义一个本地站点。本地站点的作用就是将一个特定 Web 站点的所有文档或文件都保存在本地磁盘中。定义一个本地站点需要完成以下两项工作：

- ※ 定义一个本地站点文件夹
- ※ 确定服务器端采用的技术

3.1.1 定义本地站点文件夹

在安装完 DW UltraDev 4 以后，UltraDev 会自动为用户创建五个本地站点：一个用于 ASP，一个用于 JSP，一个用于 ColdFusion，还有两个分别用于 Dreamweaver 的示例(Tutorial) 和教学(Lessons)。为简单起见，我们只介绍使用最为广泛的 ASP 来创建 Web 应用，并直接使用 UltraDev 自动创建的 ASP 站点作为本章的示例。

为了选择一个本地站点，请启动 DW UltraDev，然后打开 Site 窗口(读者可以用以下三种方法中的任何一种来打开 Site 窗口：一是选择“ Site / Site Files ”菜单命令；二是

使用快捷键 F5；三是使用启动面板)，在工具条上的下拉菜单“UD4 Tutorial - ASP”中选择“UD4 Tutorial - ASP”，如图 3-1 所示。

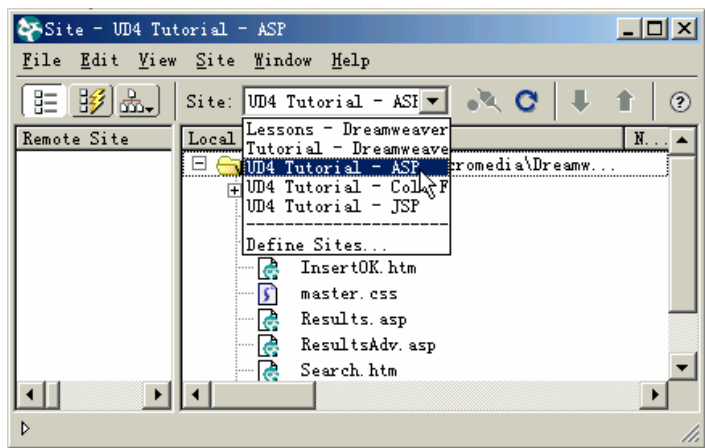


图 3-1
在 Site 窗
口中选择
ASP 示例

UltraDev 在安装过程中将 ASP 示例文件放在了 Program Files\Macromedia\Dreamweaver UltraDev 4\Tutorial\UltraDev ASP tutorial 目录下。该目录下有两个子目录：一个是 Compass Intranet，它就是 ASP 示例站点所在的目录；另一个是 Compass Intranet Backup，顾名思义，它是一个备份目录。这两个目录中的文件及子目录是完全一样的，之所以要使用 Compass Intranet Backup 这个备份目录，是因为我们在学习示例的过程中要对原来的示例文件作一些修改，如果用户要返回到示例的原始状态，只需将 Compass Intranet Backup 目录下的文件复制到 Compass Intranet 目录中，覆盖原来的文件即可。

3.1.2 确定服务器端技术

服务器端的技术实际上就是指 ASP、JSP 或 ColdFusion，它们使得 Web 服务器具有了在运行时刻生成动态页面的能力。只有确定了服务器端技术，UltraDev 才能知道在动态页面中插入何种形式的服务器端脚本（即服务器行为）。一旦选择了一种服务器技术，它就将对整个站点起作用，而不是对单独的页面起作用，这样就可以保证 Web 应用中的所有页面是相互兼容的。

由于我们已经选择了 ASP 示例站点，因此就无需再确定服务器端技术了，UltraDev 在安装过程中已经自动设置好了。如果是定义一个新站点，则必须确定服务器端技术。

3.2 定义远程站点

除了本地站点以外，我们还需要定义一个远端站点，这样 UltraDev 就会知道本地站点上的所有文件将存放到 Web 服务器上的何处。同时，定义了远端站点后，我们也就可以使用 Live Data(实时数据)窗口来预览动态页面中的内容了。

为了定义一个远端站点，必须完成下面的工作：

- ※ 在 Web 服务器上为示例创建一个发布目录。
- ※ 在 UltraDev 中定义 Web 服务器上的发布目录，然后将所有示例文件上传到该目录中。
- ※ 配置 UltraDev，以便能够使用 Live Data(实时数据)窗口来预览动态页面中的内容。

3.2.1 创建发布目录

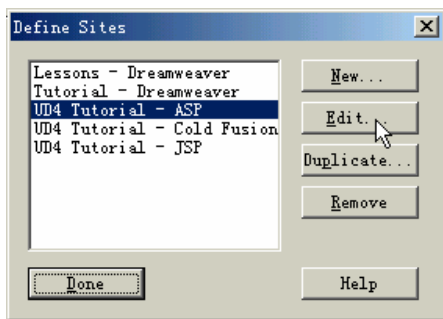
创建发布目录非常简单，只需在 Web 服务器(如 Windows 9x/Me/NT Workstation 上的 Microsoft Personal Web Server，或者 Windows 2000/NT Server 上的 Microsoft IIS) 的主目录 Inetpub\wwwroot 下新建一个名为 test 的目录即可。

在创建完发布目录后，访问远端站点的 URL 就变成：http://web_server_name/test/。

3.2.2 定义远端站点目录

在 Site 窗口中选择 Site / Define Sites 菜单命令，将出现如图 3-2 所示的 Define Sites 窗口。

图 3-2
Define
Sites 窗口



在窗口的列表中选择“UD4 Tutorial - ASP”，然后再单击 Edit 按钮，将出现如图 3-3 所示的 Site Definition 窗口。

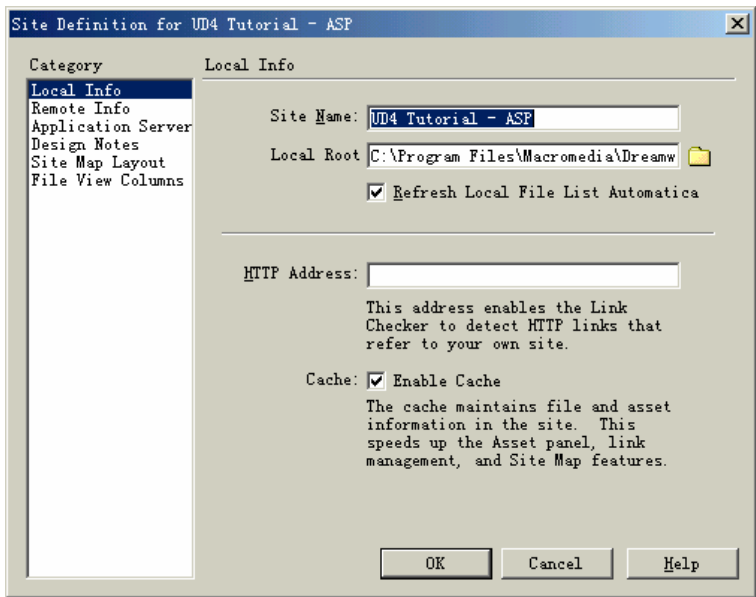


图 3-3
Site
Definition
窗口

在窗口左边的Category列表中，选择Remote Info项，则Site Definition窗口变成如图3-4所示。

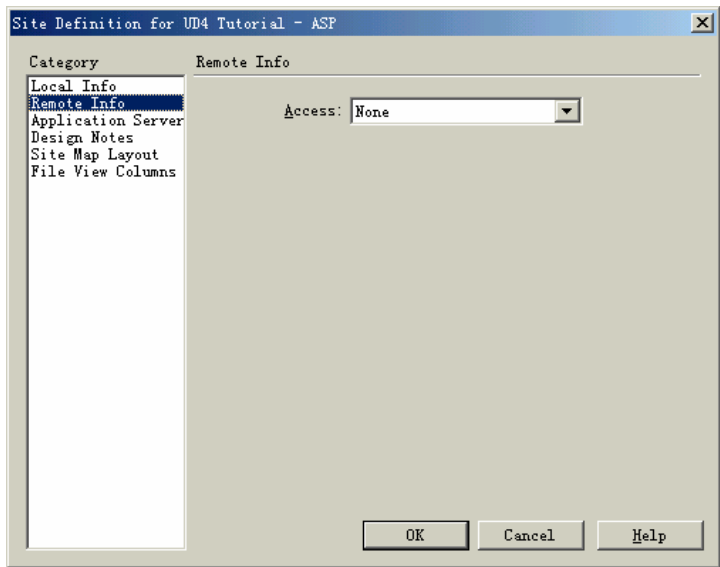
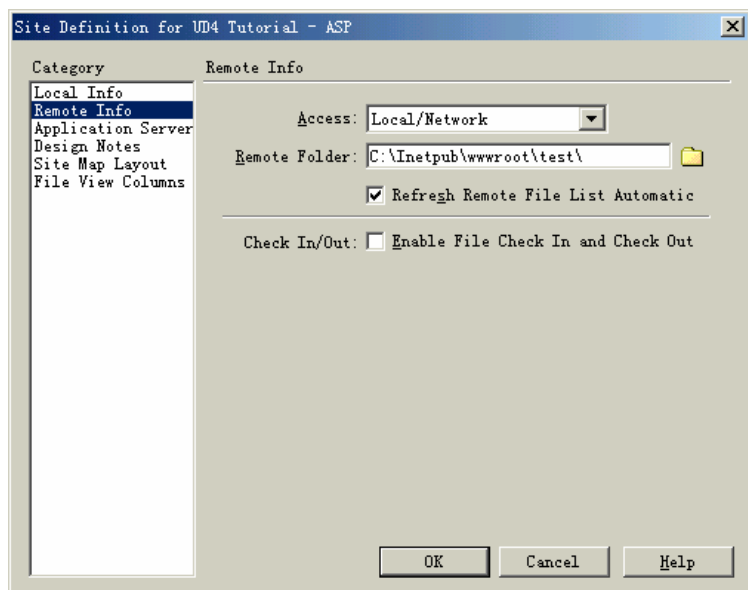


图 3-4
设置 Web
Server 选
项

在 Server Access 下拉列表中有五个选项：None、FTP、Local/Network、SourceSafe Database 和 WebDAV。为简单起见，我们假定是在一个局域网环境中，这也是最常见的

一种情况。因此在这里需要选择 Local/Network，然后在 Remote Folder 文本框中输入刚才创建的发布目录的绝对路径。如果 DW UltraDev 4 和 Web 服务器安装在了同一台计算机上，则可以直接在 Remote Folder 文本框中输入路径，如 C:\Inetpub\wwwroot\test\（具体位置取决于 Web 服务器所安装的位置），也可以单击  按钮定位到该目录，如图 3-5 所示。如果 DW UltraDev 4 和 Web 服务器没有安装在同一台计算机上，则需要先将 Web 服务器上的发布目录共享一下，并授予足够的权限，然后单击  按钮通过网上邻居定位到该目录。

图 3-5
设置访问
Web 服务
器的方式



如果有多人合作来开发一个站点，可以在“Check In/Out”域中选中 Enable File Check In and Check Out 复选框。这样，当一个作者对一个文件进行“检验”（Check Out）时，其他作者将无法修改该文件，直到该作者完成了修改，并“登记”（Check In）到站点上时，其他作者才能获权修改该文件，当然首先也要进行“检验”操作。为简单起见，在此处我们先不选择该复选框。

单击 OK 按钮，并单击 Done 按钮返回到 Site 窗口。这里的 Site 窗口如图 3-6 所示。其中左边的分栏中列出了远端站点的发布目录，同时工具栏上的 Get（下载）按钮  和 Put（上传）按钮  都变成了允许状态。

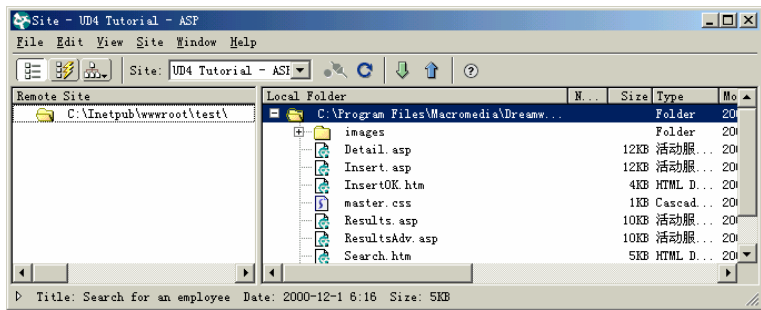


图 3-6
远端站点
目录设置
成功

在右边分栏中选择所有的文件和目录，然后单击  按钮，将所有文件上传到Web服务器的发布目录中。首先在弹出窗口中单击“确定”按钮，表示要将整个站点上传到服务器。然后如果出现如图3-7所示的窗口，则请单击Yes按钮，把所有独立的文件都包含进来。一段时间以后，文件上传便会成功，如图3-8所示。

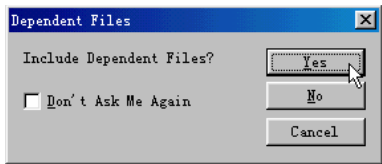


图 3-7
Dependent
Files 窗口

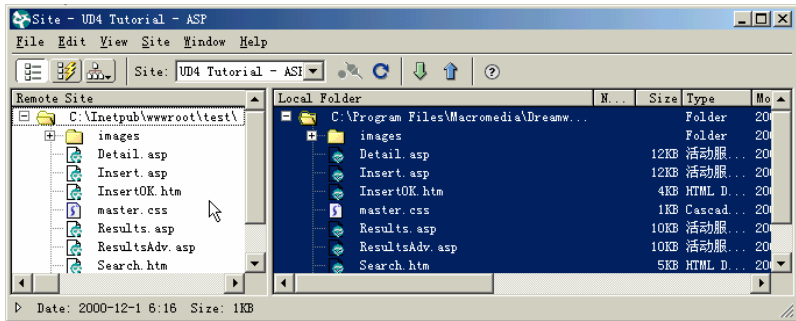


图 3-8
文件上传
成功

3.2.3 配置站点的 URL 前缀

DW UltraDev 4 中功能强大的 Live Data（实时数据）窗口可以让用户在编辑网页的同时预览其中的动态内容。Live Data 窗口需要远端站点中的 Web 服务器的支持。可以这样理解，Live Data 窗口把 Web 服务器“借”过来运行本地的网页，并在本地的窗口中将结果显示出来。

使用过 DW UltraDev 1.0 的用户可能会知道，在使用 Live Data 窗口之前，首先必须进行配置，以便 DW UltraDev 4 能够知道使用哪一个 Web 服务器来运行 Live Data 窗口

中的动态网页。而在 DW UltraDev 4 中可以省略这一步了。DW UltraDev 4 引入了远程数据库连接特性，使得开发者在设计阶段花费很少的或者不花费任何代码就可以链接到数据库。开发者不再需要定义一个设计时刻的数据库连接，UltraDev 会自动在设计阶段使用运行时刻的数据库连接。这一特性大大增强了实时数据窗口与数据库之间进行链接的功能。

当然，我们还是需要确定一下站点的 URL 前缀，以便 UltraDev 能够在设计阶段正确地找到 Web 服务器。

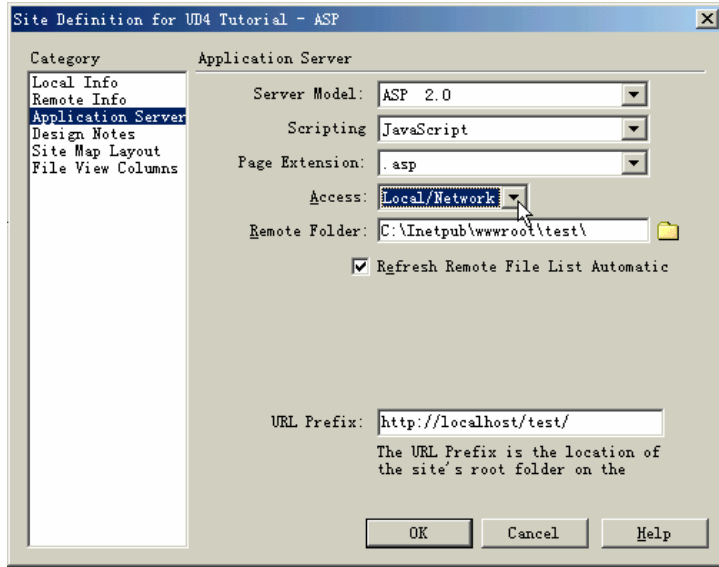
在 Site 窗口中打开如图 3-3 所示的 Site Definition 窗口，然后在窗口左边的 Category 列表中，选择 Application Server 项，并把 Access 下拉列表中的值由 None 改为 Local/Network，如图 3-9 所示。

图 3-9

设置

Application

Server 属性



在 URL Prefix 文本框中输入远端站点的 URL 前缀。在本例中，如果 DW UltraDev 4 和 Web 服务器安装在了同一台计算机上，则应该输入 http://localhost/test/；如果 DW UltraDev 4 和 Web 服务器没有安装在同一台计算机上，则应该输入 http://web_server_name/test/（请用实际的 Web 服务器域名替换这里的 web_server_name）。最后依次单击 OK 和 Done 按钮完成设置。

3.3 创建数据库链接

一个数据库链接是指用来建立一个到数据库的链接所需要定义的一些参数。没有数

数据库链接，我们的 Web 应用就无法知道如何同数据库打交道。本示例所使用的数据库名为 `compasstravel.mdb`，它位于 `Program Files\Macromedia\Dreamweaver UltraDev 4\Tutorial\UltraDev tutorial database` 目录下，我们将创建一个到该数据库的运行时刻的链接。不同的服务器技术创建的运行时刻链接是不一样的，对于 ASP 来说，我们需要创建一个 ADO（Microsoft ActiveX Data Objects）运行时刻链接。

本示例所使用的数据库文件是用 Microsoft Access 创建的。虽然并不需要用户的计算机上安装 Microsoft Access 来运行本示例，但是必须有相应的 Microsoft Access Driver 驱动程序来读取这个数据库文件。对于 Windows 用户来说，Microsoft Access Driver 驱动程序的版本要在 3.5 以上，如果已经安装了 Microsoft Office 2000，或者 Web 服务器使用的是 Windows NT Server 或 Windows 2000 Server，则该驱动程序已经安装了。

为了确定是否安装了 Microsoft Access Driver 驱动程序，请打开系统的“控制面板”，然后打开“ODBC 数据源管理器”窗口（通常双击“ODBC 数据源”图标即可），然后切换到驱动程序面板，如图 3-10 所示。



图 3-10
确定是否
安装
Microsoft
Access
Driver 驱
动程序

如果在“当前系统所安装的 ODBC 驱动程序”列表中找到了“Microsoft Access Driver (*.mdb)”项，则表明系统已经安装了 Microsoft Access Driver 驱动程序；否则，就需要通过安装 Microsoft Office 2000，或者从 <http://www.microsoft.com/data/download.htm> 网站上下载并安装 Microsoft Data Access Components（MDAC）2.1 来安装 Microsoft Access Driver 驱动程序。

如果 Web 服务器和 DW UltraDev 运行在同一台计算机上，则我们可以使用一个数据源名（DSN，Data Source Name）来创建与数据库的链接。通常把 DSN 看作是与数据库进行链接的一种快捷方法。幸运的是，DW UltraDev 在安装过程中已经为示例数据库

compasstravel.mdb 设置好了 DNS。读者可以打开“ODBC 数据源管理器”窗口，然后切换到“系统 DSN”面板，就可以在“系统数据源”列表中找到名为 compasstravel 的 DSN，如图 3-11 所示。

如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上，则我们既可以在 Web 服务器上创建一个 DSN 来实现与数据库的连接，也可以使用一个连接字符串来创建与示例数据库的连接。

下面我们对这两种情况分别进行介绍。

图 3-11
系统已经
安装的
DSN

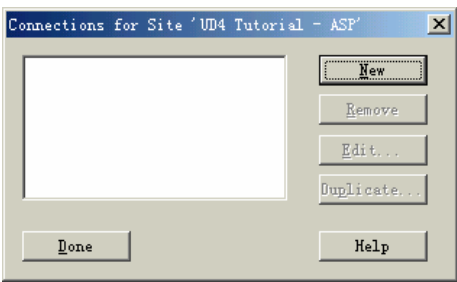


3.3.1 在已设置 DSN 的 Web 服务器上创建数据库连接

如果 Web 服务器上已经设置好了 DSN，则我们可以按照下面的方法来创建一个数据库连接。

首先确保已经在站点窗口中打开了“UD4 Tutorial - ASP”站点，然后在 DW UltraDev 的文档窗口中，选择“Modify / Connections”菜单命令，则出现如图 3-12 所示的 Connections 窗口。可以看出，此时窗口中没有任何数据库连接。

图 3-12
Connections
窗口



然后单击 **New** 按钮，这时在 **New** 按钮上会出现一个下拉菜单，在其中选择“Data Source Name(DSN)”菜单命令则出现如图 3-13 所示的 Data Source Name(DSN) 窗口。

在窗口的 **Connection Name** 文本框中输入数据库的连接名为 **connCompass**（通常在一个表示数据库链接的名字前面加上前缀 **conn**，以便于记忆）。如果 Web 服务器和 DW UltraDev 运行在同一台计算机上，则请确保在 **UltraDev Should Connect** 域中已经选择了 **Using Local DSN** 单选钮，然后在 **Data Source Name(DSN)** 下拉列表中选择已经创建好的名为 **compasstravel** 的 DSN，如图 3-13 所示；如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上，则请确保在 **UltraDev Should Connect** 域中已经选择了 **Using DSN On The Application Server** 单选钮，然后在 **Data Source Name(DSN)** 文本框中输入 **compasstravel**，如图 3-14 所示。

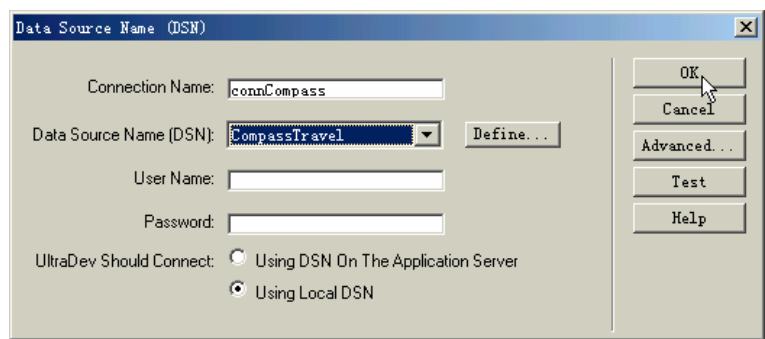


图 3-13
Data Source
Name
(DSN) 窗
口

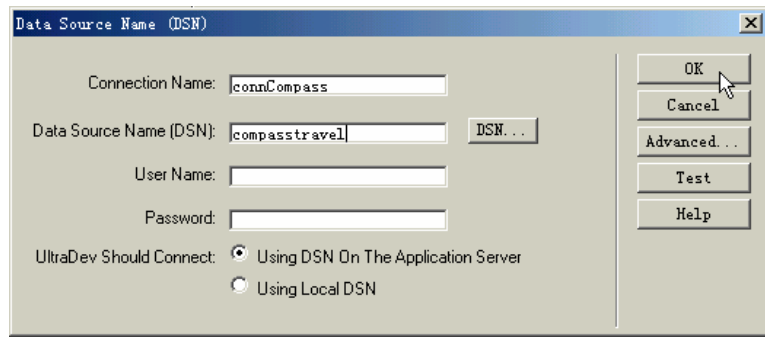
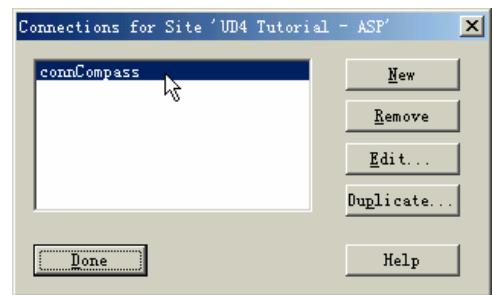


图 3-14
使用远端
Web 服务
器上的
DSN

设置完成后，单击 **OK** 按钮关闭 Data Source Nam(DSN) 窗口并返回到 **Connections** 窗口，这时在 **Connections** 窗口中就会出现一个新的名为 **connCompass** 的数据库链接，如图 3-15 所示。

单击 **Done** 按钮关闭 **Connections** 窗口。于是 **connCompass** 数据库链接创建完成。

图 3-15
数据库链接
connCompass
创建成功

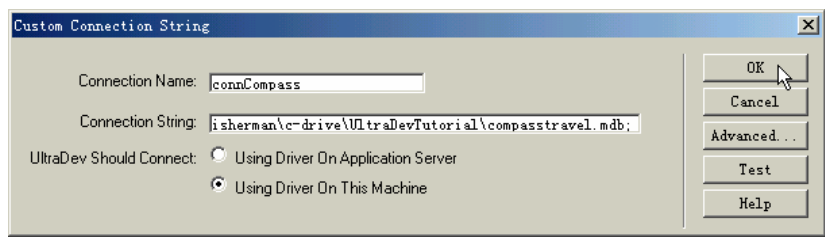


3.3.2 在未设置 DSN 的 Web 服务器上创建数据库链接

如果 Web 服务器上没有设置好 DSN，则我们可以按照下面的方法使用连接字符串来创建一个数据库链接：

首先在 DW UltraDev 的文档窗口中，选择“ Modify / Connections ”菜单命令，则出现如图 3-12 所示的 Connections 窗口。然后单击 New 按钮，在弹出的快捷菜单上选择“ Custom Connection String ”菜单命令，则出现如图 3-16 所示的 Custom Connection String 窗口。

图 3-16
Custom
Connection
String 窗口



在 Connection Name 文本框中输入数据库连接名为 connCompass，然后在 Connection String 文本框中输入一个连接到位于 Web 服务器上的数据库文件 compasstravel.mdb 的连接字符串，如图 3.16 所示。例如，如果该数据库文件位于一个名为 fisherman 的 Web 服务器的 c:\UltraDevTutorial\目录下，则连接字符串应该如下所示：

Provider=MSDASQL;Driver={Microsoft Access Driver (*.mdb) }; DBQ=\\fisherman\c-drive\UltraDevTutorial\compasstravel.mdb;

如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上，则请确保在 UltraDev Should Connect 域中已经选择了 Using DSN On Application Server 单选钮；如果 Web 服务器和 DW UltraDev 运行在同一台计算机上，则请确保在 UltraDev Should Connect 域中已经选择了 Using Driver On This Machine 单选钮。

设置完成后，单击 OK 按钮关闭 Custom Connection String 窗口并返回到 Connections

窗口,这时在 Connections 窗口中就会出现一个新的名为 connCompass 的数据库链接。单击 Done 按钮关闭 Connections 窗口。于是数据库链接创建完成。

3.4 创建 ASP 示例页面

下面我们可以开始为 Web 应用创建动态的 ASP 页面了。在下面的示例中,我们首先要为站点 Compass 创建一套简单的查询/结果页面,以便公司的员工能够用它们查询数据库的相关信息。为简单起见,DW UltraDev 已经为我们创建好了相关的查询页面和结果页面的 HTML 结构,我们要做的只是将其变成动态的页面,协调地工作在一起。然后,我们还要创建一个细节页面,以便能够显示出每个员工的详细信息(甚至包括员工的照片),并允许用户进行浏览。

3.4.1 创建简单的查询/结果页面

1. 准备查询页面

首先我们需要一个查询页面以便能够获取用户的查询参数及条件。一个简单的查询页面通常使用一个 HTML 表单来得到一个单一的参数并将其提交给服务器上的结果页面。结果页面用来根据用户的查询条件生成并显示查询结果。

由于本示例中的查询页面和结果都已经准备好了,我们要做的工作就是在查询页面的 HTML 表单中确定提交表单后用来处理查询参数的结果页面。具体步骤如下:

 确保站点窗口处于打开状态,并打开了我们的示例站点“UD4 Tutorial - ASP”,然后在 Local Folder 列下找到文件 Search.htm 并双击它,于是在文档窗口中便会打开这个查询页面,该页面允许用户通过部门(Department)这个条件来查询员工信息,如图 3-17 所示。

 在文档窗口中单击任何一个表单对象,如“DEPARTMENT”文字、下拉列表或者“Search”按钮,则状态栏上的 HTML 标记区域会变成如图 3-18 所示的样子,在其中找到并单击“<form>”,于是页面中的表单被选中。

如果属性面板没有打开,请选择“Window / Properties”菜单命令打开它,如图 3-19 所示。在 Action 域中,单击位于文本框右边的图标,于是会出现一个如图 3-20 所示的文件选择窗口,在其中选择名为 Results.asp 的结果页面,然后单击 OK 按钮。

图 3-17
查询页面

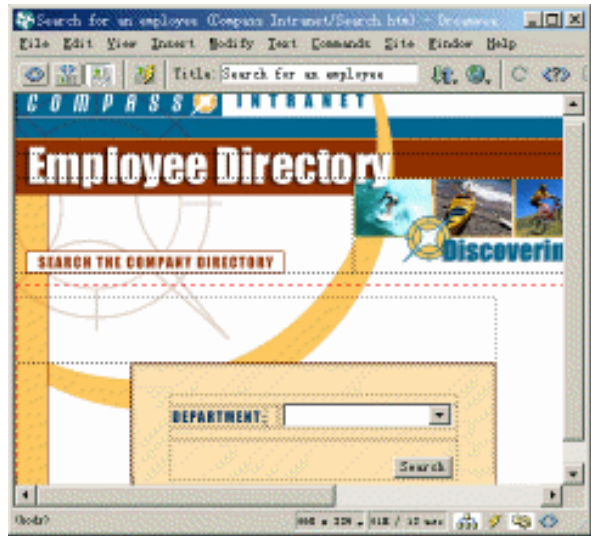


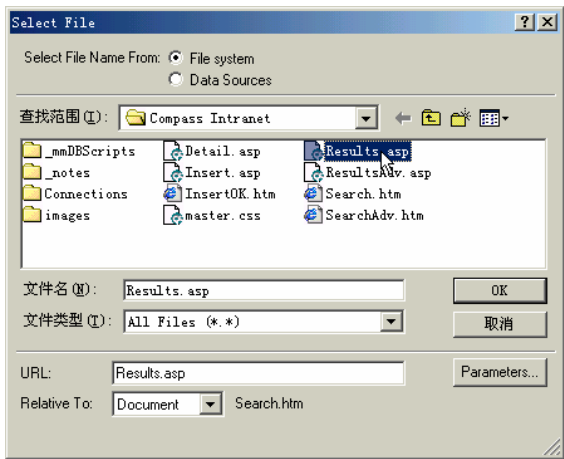
图 3-18
通过状态
栏选中表
单

```
<body> <table> <tr> <td> <form> </td> </tr> </table> </body>
```

图 3-19
表单的属
性面板



图 3-20
选择结果
页面



- 3 在图 3-19 所示的属性面板中的 Method 下拉列表中选择 GET 提交方法，这样就可以使得查询参数能够和 URL 字符串一起传递给服务器，并打开结果页面。
- 4 单击表单中的下拉列表对象，并在属性面板的 List/Menu 文本框中输入“ mnuDept ”替换掉到原来的命名，如图 3-21 所示。

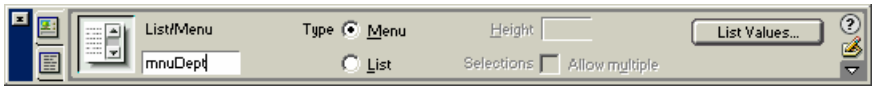


图 3-21
给下拉列表表单对象重新命名

请保存所做的工作。于是查询页面就准备好了，下面要做的工作就是对结果页面进行处理。

2. 为结果页面定义一个记录集

一个记录集是指从数据库中的一个或多个表中提取出来的数据的子集。记录集将作为动态页面的数据源。在定义记录集的时候，为了提高效率，应该只包含 Web 应用所需要的字段和数据，这通常要对数据库中的记录进行过滤。为结果页面定义一个记录集的方法如下：

打开 Site 窗口，并确信“ UD4 Tutorial - ASP ”站点出现在窗口中，然后在 Site 窗口右边的本地分栏中双击 Results.asp 文件，在文档窗口中打开该页面。

在 Data Bindings(数据绑定) 面板中，单击  按钮，在下拉菜单中选择 Recordset (Query) 项，如图 3-22 所示。

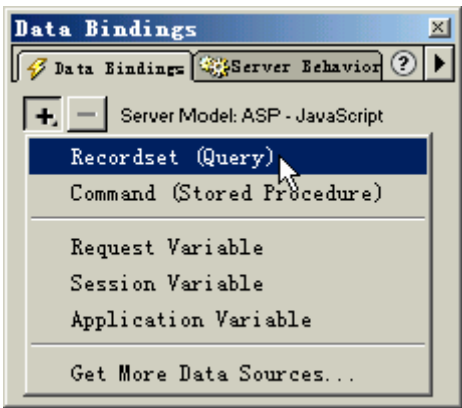
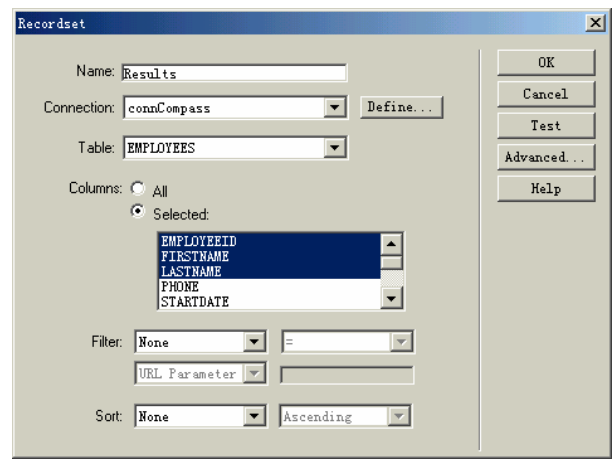


图 3-22
增加记录集

于是会出现如图 3-23 所示的简单版本的 Recordset 窗口。在其中的 Name 文本框中，输入 Results，请注意不要在记录集名中使用空格或其他的特殊字符。然后在 Connection

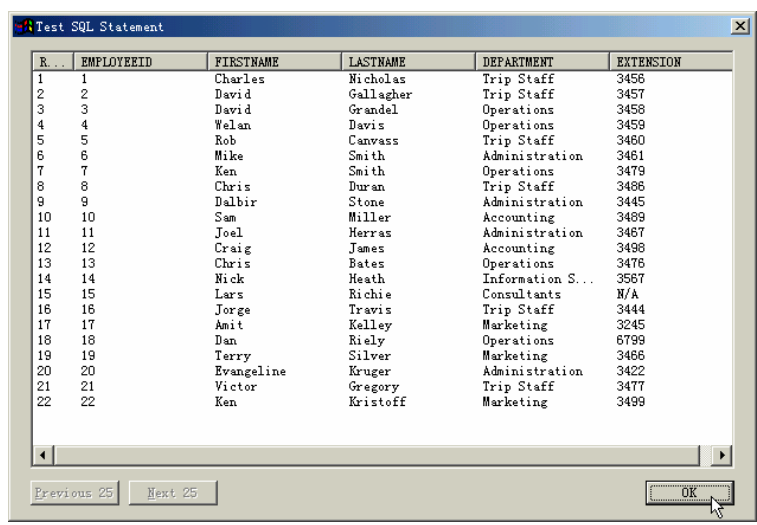
下拉列表中选择 connCompass 项，接下来 Recordset 窗口中各控制域的值会发生相应变化。在 Table 下拉列表(其中显示的是数据库中的所有表名)中，选择 EMPLOYEES 项，并在 Columns 域中，选择 Selected 单选钮，然后选择几个特定的字段：EMPLOYEEID，用来惟一的标识记录；FIRSTNAME，用来表示员工的姓；LASTNAME，用来表示员工的名字；DEPARTMENT，用来表示员工的部门；EXTENSION，用来表示员工的分机。在选择这几个字段的时候，请使用 Ctrl 键来辅助选择。

图 3-23
Recordset
窗口



现在单击 test 按钮对记录集进行测试，则会出现如图 3-24 所示的 Test SQL Statement 窗口，其中显示了记录集中的各个记录。连接两次单击 OK 按钮，并选择“File / Save”菜单命令保存所做的工作。

图 3-24
测试记录
集



这里，UltraDev 会将记录集作为数据源列表放置在 Data Bindings 面板中，如图 3-25 所示。如果要查看记录集中的各个字段，只需单击“+”号展开记录集即可。单击“-”号可以收缩记录集。

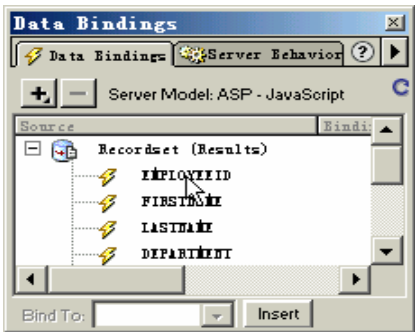


图 3-25
Data Bindings 面板中的数据源

3. 在结果页面中增加动态内容

在定义记录集以后，我们就可以将记录集中的各个字段作为对象放置到页面中了。在 Results.asp 文档窗口中选取中间表格中的“LAST”文字，然后在 Data Bindings 面板中选择 LASTNAME 项，并单击 Insert 按钮，或者将 LASTNAME 项拖放到文档窗口中选中的“LAST”文字上，如图 3-26 所示。

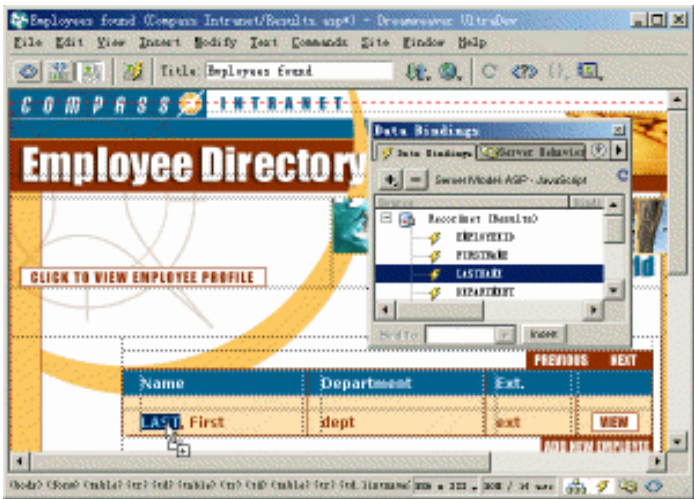


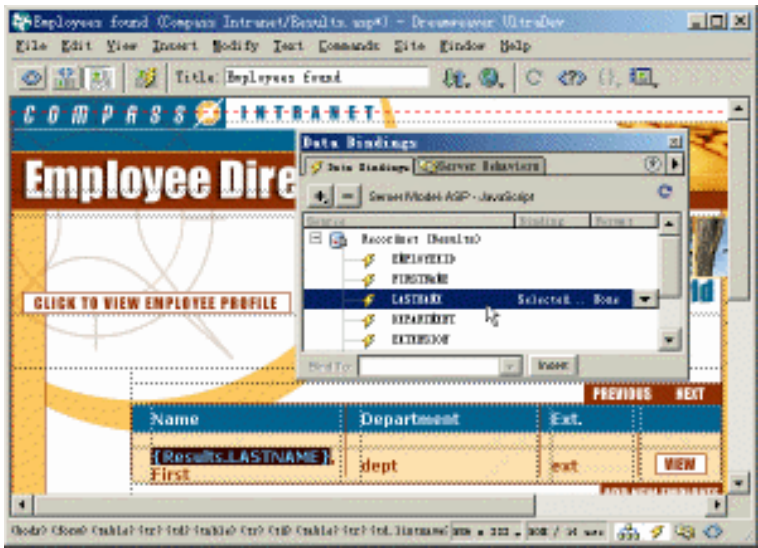
图 3-26
将字段拖放到页面文字上

这时“LAST”文字将被占位符（此时为“{Results.LASTNAME}”）所代替，如图 3-27 所示。同时在 HTML 源文件中也增加了相应的服务器端脚本。当服务器运行该页面时，占位符将被记录集中的实际数据所取代。

接下来，我们将采取同样的步骤把员工的姓和部门增加到页面中。在页面中间的表

格中，双击文字“ First ”以选取它，然后在 Data Bindings 面板中选择 FIRSTNAME 项，并单击 **Insert** 按钮，或者将 FIRSTNAME 项拖放到文档窗口中选中的“ First ”文字上，则“ First ”也将被占位符所代替；同样，双击页面中的“ dept ”文字，选取它，然后在 Data Bindings 面板中选择 DEPARTMENT 项，并单击 **Insert** 按钮，或者将 DEPARTMENT 项拖放到文档窗口中选中的“ dept ”文字上，则“ dept ”也将被占位符所代替；双击页面中的“ ext ”文字，选取它，然后在 Data Bindings 面板中选择 EXTENSION 项，并单击 **Insert** 按钮，或者将 EXTENSION 项拖放到文档窗口中选中的“ ext ”文字上，则“ ext ”也将被占位符所代替。

图 3-27
文档中的
文字被占
位符代替



保存所做的工作。接下来就可以使用 Live Data 窗口来预览动态页面中的实时数据了。选择“ View / Live Data ”菜单命令，或者单击工具栏上的  按钮，则 Live Data 窗口将出现在文档窗口中，并且页面中的三个占位符都被记录集中的实际数据所代替，如图 3-28 所示。此时，我们还可以在 Live Data 窗口中调整页面布局，增加或删除动态内容，甚至改变页面的服务器行为。

选择“ View / Visual Aids / Invisible Elements ”菜单命令将显示在动态内容上的高亮显示去掉，以使用户能够看到页面的实际显示效果。最后别忘记保存所做的工作。

如果用户在上述操作过程中出现了错误，则可以打开 Server Behaviors(服务器行为) 面板，选择其中的某项动态内容，然后单击  按钮删除它。如图 3-29 所示。

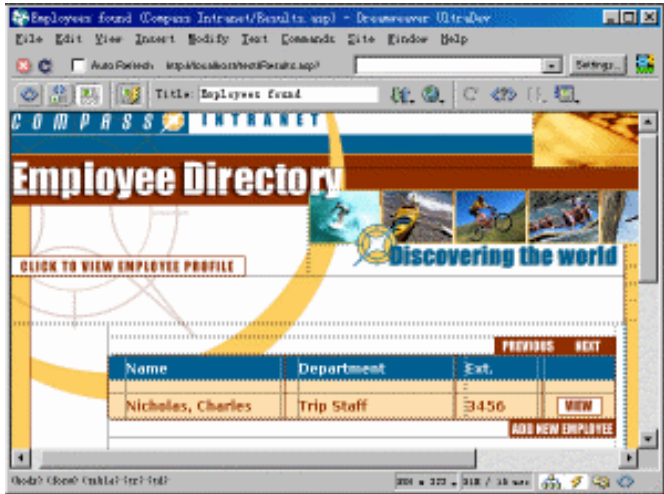


图 3-28
在 Live
Data 窗口
中预览动
态内容

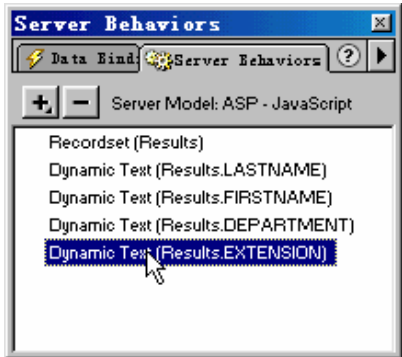


图 3-29
删除错误
动态内容

4. 在结果页面中增加重复域

在图 3-28 所示的 Live Data 预览窗口中，可以看到，缺省情况下 UltraDev 只在页面中显示一条记录，这显然不符合我们的要求。通常我们希望能够在页面中显示尽可能多的记录。

下面我们需要增加一些服务器行为，以便动态页面能够更好的工作。例如，用户应该能够在一个单独的页面上看到所有员工的概况信息，并且能够点击查看任意一个员工的详细信息。

为了实现这一点，我们需要增加两个服务器行为：一个用来在同一页面中显示多个记录；另一个用来打开某一记录的细节页面。

为了实现在同一页面中显示多个记录，需要在该页面中增加一个重复域，这得使用 Repeat Region 这个服务器行为，它能够重复存放有数据的页面元素。在本例中，这个页

面元素就是表格中的行。当服务器运行该页面时，Repeat Region 服务器行为将重复表格的行元素，直到 Results 记录集中的所有记录都显示出来，每个表格行将显示其中的一条记录内容。

首先请选择 View / Live Data ”菜单命令将 Live Data 窗口切换到文档窗口，然后单击中间表格中 4 个占位符中的任何一个以选择它，然后在状态行上找到最右边的一个“ <tr> ”标记并单击它，于是就会选取动态内容所在的表格行。

接下来打开或者切换到 Server Behaviors 面板，单击其中的  按钮，并在出现的菜单中选择 Repeat Region 菜单项，于是出现如图 3-30 所示的 Repeat Region 窗口。



确保在 Recordset 域中选择了 Results 记录集，然后选取 Records 单选钮，在文本框中输入数字 5 代替原来的 10，并单击 OK 按钮。

现在再次将文档窗口切换到 Live Data 窗口，读者可以看到页面中把前 5 条记录都显示出来了，而不只是显示第一条记录，如图 3-31 所示。

最后别忘记保存所做的工作。



5. 在结果页面中增加到细节页面的链接

为了能够在结果页面中打开与每条记录相关的细节页面，我们必须在结果页面中增加一些到细节页面的链接。当然，仅仅增加超链接是不够的，为了从记录集中取得正确的数据，细节页面还需要知道用户在结果页面上选择了哪一个记录。我们需要按照下面的步骤来实现上述要求。

首先将 Live Data 窗口切换到文档窗口，然后在结果页面中单击 **VIEW** 图片以选取它。确保 Server Behaviors 面板处于打开状态，单击其中的 **+** 按钮，然后在出现的菜单中选择 Go to Detail Page 菜单项，于是出现如图 3-32 所示的 Go to Detail Page 窗口。

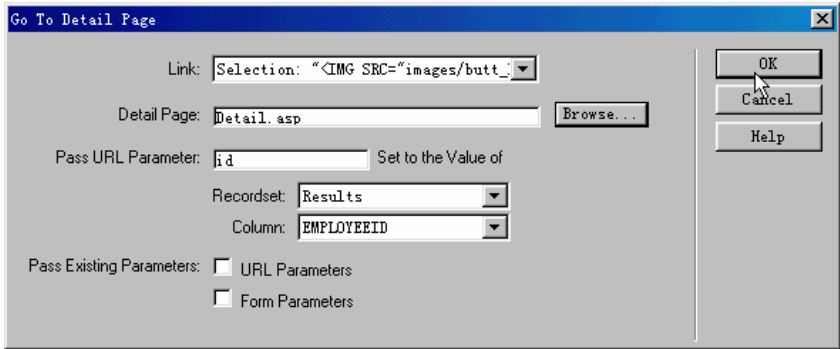


图 3-32
Go to Detail
Page 窗口

在 Detail Page 文本框中，单击右边的 Browse 按钮，然后选择预先设计好的细节页面 Detail.asp，在 Pass URL Parameter 文本框中，输入“ id”。这样做的目的是将一个名为“ id ”的参数传递给细节页面，以便确定用户所选择的链接字段。服务器将根据 Recordset 和 Column 这两个下拉列表中的值来设定参数“ id ”的值。本例中，id 的值为 Results 记录集中的 EMPLOYEEID 字段的值。

单击 OK 按钮并保存所做的工作。现在，当用户单击结果页面上的链接字段时，不仅会打开细节页面，同时用户所选择的记录信息也会传递给细节页面，从而显示出正确的信息。



在 Live Data 窗口中链接无法工作。为了查看链接效果，必须将该页面上传到 Web 服务器上，然后在浏览器中打开服务器上的该页面。

3.4.2 创建细节页面

创建细节页面的过程同创建结果页面的过程是一样的，也需要先定义一个记录集，

然后在页面中增加动态内容，最后增加服务器行为使页面工作。在细节页面中，我们要增加三个服务器行为：一个允许页面查找并显示特定的记录；一个允许用户在记录间移动；另一个实现在不需要的时候隐藏超链接的功能。

1. 为细节页面定义一个记录集

选择“Window / Site Files”菜单命令切换到 Site 窗口，然后在 Site 窗口右边的本地分栏中双击 Detail.asp 文件，在文档窗口中打开该页面。

打开 Data Bindings (数据绑定) 面板，单击  按钮，在下拉菜单中选择 Recordset (Query) 项，于是出现 Recordset 窗口。

在 Name 文本框中输入 Details 作为记录集名，注意不要在记录集名中使用空格或其他特殊字符。在 Connection 下拉列表中选择 connCompass 选项，于是 Recordset 窗口中各个域的值会相应发生变化。在 Table 下拉列表中选择 EMPLOYEES 选项。在 Columns 域中，选取 All 单选钮，以便选取数据库表中的所有字段，如图 3-33 所示。然后单击 Test 按钮对记录集进行测试，测试成功后，单击 OK 按钮关闭 Recordset 窗口，并保存所做的工作。

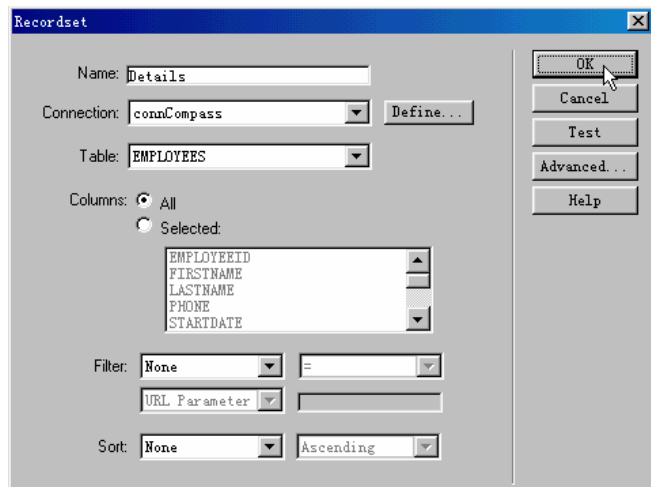


图 3-33
为细节页
面定义记
录集

2. 在细节页面中增加动态内容

在细节页面中定义了记录集以后，我们就可以把该记录集作为细节页面中动态内容的数据源。

确保 Data Bindings 面板已经打开，并展开 Details 记录集。在文档窗口中双击位于表格中蓝色行的“LastName”字符串，然后在 Data Bindings 面板中选取其中的 LASTNAME 字段，并将其拖放到所选取的“LastName”字符串上去，这时在该单元格

中会出现{Details.LastName}占位符。

重复上面的步骤（即在细节页面中选择一段文本，然后把相应的数据源拖放到其上），完成下面的操作：

- ※ 在文档窗口中选择字符串“ FirstName ”，将 Data Bindings 面板中的 FIRSTNAME 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ number ”，将 Data Bindings 面板中的 PHONE 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ date ”，将 Data Bindings 面板中的 STARTDATE 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ dept ”，将 Data Bindings 面板中的 DEPARTMENT 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ ext ”，将 Data Bindings 面板中的 EXTENSION 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ email ”，将 Data Bindings 面板中的 EMAIL 字段拖放到其上。
- ※ 在文档窗口中选择字符串“ notes ”，将 Data Bindings 面板中的 NOTES 字段拖放到其上。

接下来，我们将对页面中的照片的源属性进行绑定，以便能够显示出相应员工的正确照片。该源属性由一个保存在数据库中的字符串（如“ jones_lyn.jpg ”）组成。

确保属性面板是打开的，单击细节页面中的人物照片，然后再单击属性面板中 Src 文本框右边的图标，于是会出现一个选择图像源的对话框。在 Select File Name From 域中选择 Data Sources 单选钮，于是会出现一系列数据源，如图 3-34 所示。

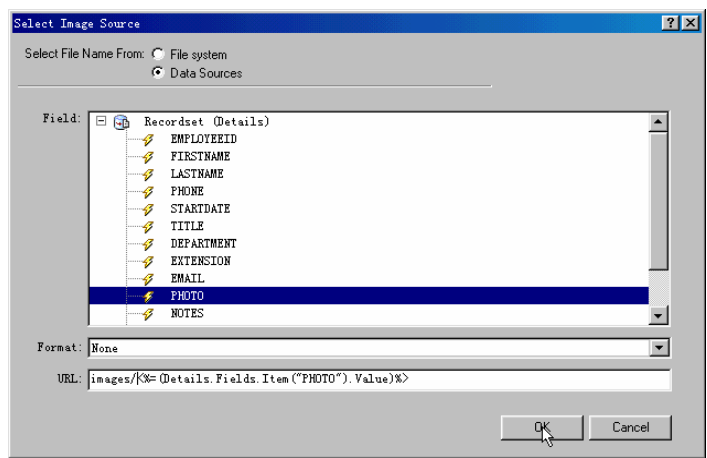


图 3-34
选择照片
的数据源

在 Field 列表中选择 PHOTO 字段，并在 URL 文本框中现有字符串的最前面输入文字‘ images/’(不包括双引号)。由于数据库中包含的并不是真正的照片，只是照片的文件名，而不是照片所在的文件夹完整路径，因此这一步的作用是让 UltraDev 能够根据数据库中的文件名组合成正确的路径，以便能够显示出正确的照片。

现在请把文档窗口切换到 Live Data 窗口，对刚才添加的动态内容进行预览。读者可以发现在 Live Data 窗口的细节页面中显示出了第一条记录的详细信息，如图 3-35 所示。最后请保存所做的工作。

图 3-35
Live Data
窗口中显
示出第一
条记录的
详细信息

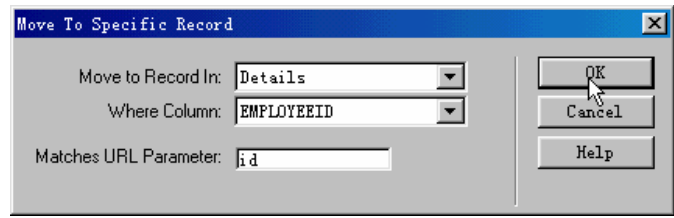


3. 使页面能够查找并显示某一记录

下面我们需要往页面中添加一些必须的服务器行为，以便页面能够正常工作。前面我们提到过，当在浏览器打开细节页面时，其中显示的内容应该是用户在结果页面中所选择的某个特定记录的详细信息。为了实现这一点，我们需要使用 Move to Specific Record 这个服务器行为。

确保 Server Behavior 面板已经打开，然后单击  按钮，并在出现的菜单中选择 Move to Record / Move to Specific Record” 菜单命令，于是出现 Move to Specific Record 窗口，如图 3-36 所示。

图 3-36
Move to
Specific
Record 窗
口



确保 Move to Record In 字段中的值为 Details，Where Column 字段中的值为 EMPLOYEEID，并且 Matches URL Parameter 文本框中的值为 id。这样，细节页面就能

够根据 URL 中的 id 参数的值，在 EMPLOYEEID 字段中进行查询，并显示满足条件的记录。最后单击 OK 按钮。

保存所做的工作，并把修改过的文件上传到 Web 服务器上，然后通过浏览器打开 Web 服务器上的 Results.asp 结果页面。现在当用户点击不同的字段链接时，浏览器能够自动打开正确的细节页面了。

3.4.3 加入记录浏览功能

1. 激活浏览链接

在前面介绍结果页面时读者会注意到，我们只显示出了前 5 条记录，而记录集中实际共有 22 条记录。为了让用户查看能够其他各记录的信息，结果页面还应该具有允许用户在记录间进行移动浏览的功能，也就是说，结果页面应该允许用户向前或向后浏览不同的记录。在结果页面中，我们已经看到了 **PREVIOUS** 和 **NEXT** 这两个用于浏览的图片，下面的工作就是激活这两个浏览链接。

首先在文档窗口中打开 Results.asp 页面，选择页面中的 **PREVIOUS** 图片，然后在 Server Behaviors 面板中，单击  按钮，并在出现的菜单中选择“Move to Record / Move to Previous Record”菜单命令，于是出现如图 3-37 所示的 Move to Previous Record 窗口。

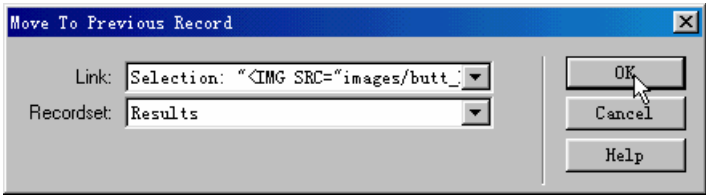


图 3-37
Move to
Previous
Record 窗
口

确保 Recordset 字段的值为 Results，然后单击 OK 按钮。
重复上述步骤，为 **NEXT** 图片添加服务器行为，只不过这次需要选择 Move toRecord / Move to Next Record”菜单命令。最后保存所做的工作。

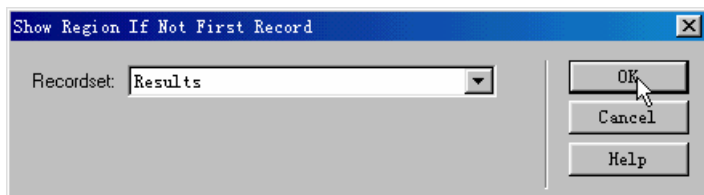
2. 隐藏浏览链接

当用户把上述结果页面上传到 Web 服务器上，并在浏览器中进行浏览时，会发现结果页面还有一个问题：当浏览到第一条记录或者最后一条记录时，相关浏览链接仍然显示出来，但此时它已经没有了相应的浏览功能。例如，目前已经浏览到第一条记录，此时再单击 **PREVIOUS** 图片这个浏览链接已经没有意义了，系统并不会定位到上一条记录，因为此时已经没有上一条记录了。

为了解决这个问题，我们需要往结果页面中添加一点智能化功能：当浏览到第一条记录或者最后一条记录时，让相关浏览链接自动隐藏起来。这可以通过 Show Region 服务器行为来实现。

首先，请在结果页面中选择 **PREVIOUS** 图片，然后在 Server Behaviors 面板中单击  按钮，并在出现的菜单中选择“ Show Region / Show Region If Not First Record ”菜单命令，于是出现如图 3-38 所示的 Show Region If Not First Record 窗口。

图 3-38
Show
Region If
Not First
Record 窗
口



在 Show Region 子菜单中共有 6 个菜单命令，分别针对了记录集是否为空、非空、第一条记录、不是第一条记录、最后一条记录以及不是最后一条记录这 6 种情况下显示浏览链接的控制。在此处之所以选择“ Show Region If Not First Record ”，作用就是当没有浏览到第一条记录时才显示该链接。单击 OK 按钮关闭图 3.37 所示的对话框。

重复上述步骤，为 **NEXT** 图片增加 Show Region 服务器行为，只不过此时要选取“ Show Region / Show Region If Not Last Record ”菜单命令。

最后保存所做的工作。

到此，我们的示例就设计完成了。请将修改过的页面上传到 Web 服务器上，然后在浏览器中打开服务器上的 Results.asp 结果页面，测试一下是否工作正常。正确的结果应该是当用户浏览到第一页时，只出现 **NEXT** 图片；当用户浏览到最后一页时，只出现 **PREVIOUS** 图片；而在其他各页时，这两个图片都出现。

第 二 部 分

使用 DW UltraDev 4 开发 Web 应用

- ◆ 第 4 章 站点和文档初步
 - ◆ 第 5 章 编辑 HTML 页面
 - ◆ 第 6 章 UltraDev 开发 Web 应用基础
 - ◆ 第 7 章 创建记录集
 - ◆ 第 8 章 创建动态网页
 - ◆ 第 9 章 激活动态网页
 - ◆ 第 10 章 编辑和调试动态网页
-

第 4 章 站点和文档初步

本章概要：

- ☒ 站点和文档概述
- ☒ 站点规划
- ☒ 创建本地站点
- ☒ 站点操作
- ☒ 使用站点窗口
- ☒ 操作站点文件
- ☒ 创建和编辑文档
- ☒ 添加文本和插入对象
- ☒ 设置文档属性
- ☒ 使用历史面板
- ☒ 查看和编辑网页头部内容

从本章开始，我们进入本书的第二部分，即使用 DW UltraDev 4 开发 Web 应用。由于篇幅所限，我们不打算详细介绍 DW UltraDev 4 的网页编辑和网站管理功能，因为 DW UltraDev 4 的这一部分功能就是 Dreamweaver 4 的全部功能，请不熟悉的读者参考有关 Dreamweaver 4 的书籍。我们只在第 4、5 章简单介绍一下站点、文档、HTML 编辑的有关功能，重点放在介绍 Web 应用的开发上。

4.1 站点和文档概述

所谓站点，可以看作是一系列文档的组合。这些文档之间通过各种链接关联起来，可能拥有相似的属性，例如描述相关的主体、采用相似的设计，或实现相同的目的等，也可能只是毫无意义的链接。利用浏览器，就可以从一个文档跳转到另一个文档，实现对整个网站的浏览。

我们可以使用 DW UltraDev 来创建单个的网页,但大多数情况下,我们更希望能够将这些单独的网页组合起来,成为一个站点,拥有自己的网站,这是每个网页创作者的梦想。DW UltraDev 不仅提供了网页编辑功能,而且带有强大的站点管理功能。利用 DW UltraDev,我们可以首先在本地的计算机的磁盘上创建本地站点,从全局上控制站点结构,管理站点中的各种文档,以完成对文档的编辑。在完成站点文档的编辑后,可以利用 DW UltraDev 将本地站点发送到远端站点上的服务器中,创建真正的站点。

在开始介绍 DW UltraDev 的站点管理和网页编辑功能之前,我们必须首先了解几个基本概念。

4.1.1 本地计算机和 Internet 服务器

当我们在 Internet 上冲浪时,我们所浏览的网页都是存储在 Internet 服务器上的。所谓 Internet 服务器,就是用于提供 Internet 服务(包括 WWW、FTP、E-mail 等)的计算机。对于 WWW 浏览服务来说,Internet 服务器主要用于存储用户所浏览的 Web 站点和页面。对于大多数用户来说,Internet 服务器只是一个逻辑上的名称,而不是真正的可知实体。例如,我们无法也无需知道该服务器到底有多少台、性能如何、配置如何、放置在什么地方,等等。用户所访问的网站,可能存储在大洋彼岸的某个角落的计算机上,也可能就存储在隔壁的电脑上,这些信息都是无关紧要的。因为我们在浏览网页时,并不需要了解它们的实际位置,只需要在地址栏输入网址,按下回车,就可以轻松实现对网页的浏览。

对于浏览网页的客户来说,他们所使用的计算机被称作本地计算机。因为用户直接在计算机上操作,启动浏览器,打开网页,本地计算机对于用户来说是真正的实体。本地计算机和 Internet 服务器之间,通过各种线路(如电话线、ISDN 或其他缆线等)进行连接,以实现相互的通信。当然,在连接线路中,还要通过各种各样的中间环节,这里就不多做介绍了。

4.1.2 本地站点和远程站点

在理解了 Internet 服务器和本地计算机的概念后,了解远程站点(或称作远端站点)和本地站点就很容易了。图 4-1 显示了本地站点和远端站点的关系。

严格地说,站点也同样是由文档和文档所在的文件夹组成。设计良好的网站通常具有科学的结构。利用不同的文件夹,将不同的网页内容分门别类的保存,这是设计网站的必要前提。结构良好的网站,不仅便于管理,也便于更新。



图 4-1
本地站点
和远端站
点

我们在 Internet 上所浏览的各种网站，归根到底其实就是用浏览器打开存储于 Internet 服务器上的 HTML 文档及其他相关资源。我们通常将存储于 Internet 服务器上的站点和相关文档称作远端站点。

利用 DW UltraDev 可以对位于 Internet 服务器上的站点文档直接进行编辑和管理，但这在很多时候非常不便。诸多因素，例如网络速度和网络的不稳定性等，都会对管理和编辑操作带来影响。另一个更重要的原因是，直接对位于 Internet 服务器上的文档和站点进行操作，必须始终保持同 Internet 的连接，这意味着会花费不必要的上网费用，目前在国内，很多人会觉得这是一个沉重的负担。

既然位于 Internet 服务器上的站点仍然是以文件和文件夹作为基本要素的磁盘组织形式，在这种情况下，我们会考虑，能不能首先在本地计算机的磁盘上构建出整个网站的框架，编辑相应的文档，然后再将之放置到 Internet 服务器上呢？答案是肯定的，这就是本地站点的概念。

利用 DW UltraDev，我们可以在本地计算机上，创建出站点的框架，从整体上对站点全局进行把握。由于这时候没有同 Internet 连接，因此有充裕的时间完成站点的设计，进行完善的测试。在站点设计完毕，可以利用 DW UltraDev 本身所带的上传工具或其他工具，例如 FTP 程序，将本地站点上传到 Internet 服务器上，形成远端站点。

4.1.3 上传和下载

一般来说，资源从 Internet 服务器传递到本地计算机的过程，称作下载；而资源从本地计算机传递到 Internet 服务器上的过程，称作上传。

实际上，在正常的浏览过程中，经常会进行上传和下载操作。例如，我们在本地计算机上浏览网页，实际上就是将 Internet 服务器上的网页下载到本地计算机上，然后进行浏览；很多网站（如电子商务网站或免费电子邮件网站），都会要求用户输入用户名称和密码，这实际上就是将用户的信息上传到 Internet 服务器上。

另外,利用其他的一些工具,例如 FTP 程序,可以直接将 Internet 服务器上的站点结构及其中的文档下载到本地计算机,经过修改,再利用相应的工具将修改后的网页上传到 Internet 服务器上,以实现对该站点的更新。

DW UltraDev 中内置了强大的 FTP 功能,可以帮助我们实现对站点文档的上传和下载。

4.2 站点规划

4.2.1 网页的设计流程

为了更好地进行站点管理和网页创作,我们还需要了解在 DW UltraDev 中的网页设计和出版流程。

一般来说,创建站点的第一步是对站点进行规划。我们需要了解创建站点的目的,确定它要提供什么服务,网页中应该出现什么内容,等等。在这一步里,利用一张纸和一支笔就能很好地解决问题。有时候,一个好的构思比实际的技术显得更为重要,因为它直接决定了站点质量和将来的访问流量。

第二步是创建站点的基本结构。利用 DW UltraDev,可以在本地计算机上构建出整个站点的框架,并在各个文件夹中合理的安置文档。如果已经构建了自己的站点,也可以利用 DW UltraDev 来编辑和更新现有的站点。DW UltraDev 可以在站点窗口中以两种方式显示站点结构,一种是目录结构,另一种是站点地图。我们可以使用站点地图方式快速构建和查看站点原型。

第三步是具体的网页创作过程。一旦创建了本地站点,就可以在其中组织文档和数据。一般来说,文档就是在访问站点时可以浏览的网页。文档中可能包含其他类型的数据,例如文本、图像、声音、动画和超级链接等。可以利用 DW UltraDev 创建空白的文档,可以利用模板来批量生成具有统一风格的文档,也可以打开和编辑由其他应用程序产生的文档。

最后一步,在站点编辑完成后,需要将本地的站点同位于 Internet 服务器上的远端站点关联起来,然后定期更新。

4.2.2 规划站点结构

在 DW UltraDev 中,“站点”这个术语,既可以用于表示位于 Internet 服务器上的远端站点,也可以用于表示位于本地计算机上的本地站点。一般来说,应该首先在本地计

算机上构建本地站点，创建合理的站点结构，使用合理组织形式来管理站点中的文档，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 Internet 服务器上，以便他人的浏览。

合理的站点结构，能够加快对站点的设计，提高工作效率，节省时间。如果将一切网页都存储在一个目录下，当站点的规模越来越大时，管理起来就会变得很不容易。因此，一般来说，应该利用文件夹来管理文档。在规划站点结构时，一般应该遵循如下一些规则：

1. 用文件夹来保存文档

一般来说，应该用文件夹来合理构建文档的结构。首先为站点创建一个根文件夹（根目录），然后在其中创建多个子文件夹，再将文档分门别类存储到相应的文件夹下，必要时，可以创建多级子文件夹。例如，可以在 About 文件夹中放置用于说明公司介绍的网页，可以在 Product 文件夹中放置关于公司产品方面的网页。

2. 使用合理的文件名称

使用合理的文件名非常重要，特别是在网站的规模变得很大时。文件名应该容易理解，让人看了就能够知道网页表述的内容。

要注意的是，尽管中文文件名对于中国人来说，更清晰易懂，但是应该避免使用中文文件名。因为很多 Internet 服务器使用的是英文操作系统，不能对中文文件名提供很好的支持，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。如果读者对英文不熟悉，可以用汉语拼音作为文件的名称。

另外，很多 Internet 服务器采用的是 UNIX 操作系统，它是区分文件的大小写的。例如 Index.html 和 index.html 是完全不同的两个文件，而且可以同时出现在一个文件夹中。因此，建议在构建的站点中，全部使用小写的文件名称。

3. 合理分配文档中的资源

文档中不仅仅是文字，它还可以包含其他任何类型的对象，例如图像、声音、动画等，这些文档资源通常不能直接存储在 HTML 文档中，因此需要特别考虑它们的存放位置。一般来说，可以在站点中创建一个 Resource（资源）文件夹，然后将相应的资源保存在该文件夹中。通常有两种方式存储资源，一种是整个站点共用一个 Resource 文件夹，所有的文档资源都保存在其中，当然，在 Resource 文件夹中可以再有子文件夹，按照不同的资源类型，分门别类对资源进行存储。另一种存储资源的方式是在每个存储不同类型文档的文件夹中都创建一个 Resource 文件夹，然后在其中按类型分门别类地存储资源。这两种存储方式各有其便利之处，使用得比较多的是前一种方式，因为它可以从整体上对整个文档的资源进行控制。

4. 将本地站点和远端站点设置为同样的结构

这一点非常重要。为了便于维护和管理，应该将远端站点的结构设计成与本地站点相同。这样在本地站点上相应文件夹和文件上的操作，都可以同远端站点上的文件夹和文件一一对应。当操作完本地站点后，利用 DW UltraDev 将本地站点上传到 Internet 服务器上，可以保证远端站点是本地站点的完整拷贝，以避免发生错误。

4.2.3 规划站点的浏览体系

通常站点都会包含多个网页，如何让用户知道这些网页并访问它们，这是网站创建者必须考虑到的事情。如果用户不知道如何访问需要的网页，也就无法得到他们要获得的信息，网站的目的也就没有达到。一般来说，应该在网站创建时期，规划站点的浏览机制，目的是提供清晰易懂的浏览方法，采用标准统一的网页组织形式，引导用户轻松自如地访问每个他们要访问的网页。在规划站点的浏览机制时，一般可以考虑如下的方法：

1. 创建返回主页链接

应该在站点的每个页面上，都放置返回主页的链接。我们可能都遇到过这样的事情，在浏览了多个页面之后，迷失了自己的方向，不知道如何返回到最初的地方，很多浏览者因此失去耐心，转而开始浏览其他网站。如果在网页中包含返回主页的链接，就可以确保用户在不知道自己目前位置的情况下，快速返回到一个熟悉的环境中，继续浏览站点中其他内容。返回主页的链接，能起到很强的挽留用户的作用。

2. 显示网站专题目录

应该在主页或任何一个页面上，提供站点的简明目录结构，引导用户从一个页面快速进入到其他的页面上。很多网站使用框架技术，在页面的顶端或是左端显示当前网站的专题目录，单击相应的链接，就可以从一个专题页面中，快速跳转到另一个专题页面上。例如，DW UltraDev 的帮助系统实际上就是采用了框架技术，它在页面左方显示专题目录，用户只需单击相应的目录项，即可快速跳转到需要的网页上。

3. 显示当前位置

无论在任何网页上，都要在显要位置标出当前网页在站点中的位置，或是显示当前网页说明的主题，以帮助用户了解他们到底在访问什么地方。如果页面嵌套过多，则可以通过创建“前进”和“后退”之类的链接，帮助用户进行浏览。

4. 搜索和索引

对于一些数据型的网站,应该给用户提供搜索的功能,或是给用户提供索引检索的权力,使用户快速查找到自己需要的信息。DW UltraDev 的帮助系统就采用了这种机制,它利用框架技术,在页面顶端建立了目录、搜索和索引等链接,以使用户快速找到他们需要的信息。

5. 反馈

网页在出版后,或多或少会存在一些问题,从用户那里及时获取他们对网站的意见和建议是非常重要的。为了及时从用户那里了解到相关信息,应该在网页上提供用户同网页创作者或网站管理员的联系途径。常用的方法是将网页创作者或网站管理员公布在网页上,或是创建一个 E-mail 超级链接,帮助用户将信息回馈给网站。

4.2.4 规划模板和库的使用

通常,站点中的网页风格应该具有统一性,这样才能够突出站点要表述的主题,也同时能够帮助用户快速了解站点的结构和浏览机制。

在 DW UltraDev 中,可以利用模板和库来快速批量创建具有相同或相似风格的网页,然后再在这个基础上对网页进行必要修改,以实现网页的风格统一。文档风格统一化的特征之一就是在多个网页上重复出现某些对象,如文本、图像或声音等。例如,可以在每个网页的左上角放置公司的徽标,在右下角放置创作者的联系地址。实际创作时,维护这种风格的操作可能并不简单。例如,一个公司的徽标可能由几幅更小的图像和文本组合而成,要放置公司的徽标,不仅需要页面中插入图像和文本,还需要精确调节它们之间的相对位置,以最后形成徽标。如果要往多个网页上放置徽标,对每个网页都需要进行上述烦琐的操作,而且还可能由于位置的摆放不齐等原因,造成网页间徽标形象的不统一,从而影响网站的整体质量。

为了解决这种问题,在 DW UltraDev 中引入了模板和库的概念。在创作网页时,可以将这些被重复使用的东西或组合制作成模板或库并保存起来,当下次要往网页上放置相同对象或组合时,只需要简单地从库中调用就可以了,这不仅简化了操作,而且可以确保页面间对象或组合的绝对一致。

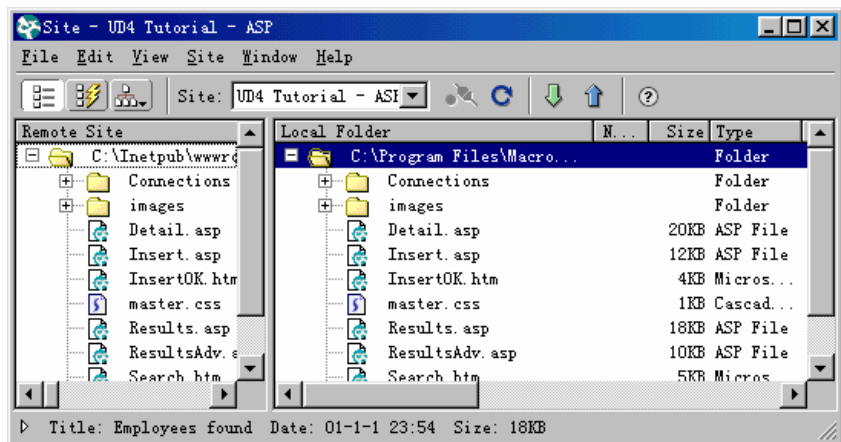
4.3 创建本地站点

使用 DW UltraDev 的第一步就是创建一个本地站点,下面介绍其步骤。

创建和管理本地站点，主要是在站点（Site）窗口中进行的，因此首先要打开站点窗口。有多种方法可以打开站点窗口：首先启动 DW UltraDev，默认情况下，它会进入文档窗口，然后选择“Window / Site Files”菜单命令或者“Site / Site Files”菜单命令，即可打开站点窗口；或者单击启动面板上的“Site”（站点）按钮，或者按下 F5 快捷键，也可打开站点窗口。

一个站点窗口如图 4-2 所示。它左边的窗格中称作远端站点文件列表窗格，其中显示的就是保存在远端服务器上的站点的文档；右边的窗格称作本地站点文件列表窗格，其中显示的就是本地站点内容。

图 4-2
站点窗口



要构建本地站点，可以按照如下方法进行操作。

首先要确定本地站点的根目录所在位置。如果希望创建一个全空的本地站点，可以利用 Windows 的资源管理器，在合适的位置上（通常是某个驱动器的根部）创建一个文件夹，用于作为本地站点的根目录。例如，假设我们想创建一个介绍飞雪漫工作室的站点，可以在 C 驱动器根部上创建 fisherman 目录。

然后从文档窗口或站点窗口中，选择“Site / New Site”菜单命令，即可打开如图 4-3 所示的 Site Definition 对话框。

在对话框的“Site Name”（站点名称）文本框中，可以输入该站点的名称，如“fisherman”。在“Local Root Folder”（本地根目录）文本框中，输入该本地站点根目录的位置。用户既可以直接在该文本框中输入目录名，也可以单击其后的文件夹按钮，打开一个如图 4-4 所示的 Choose Local Folder 对话框，然后从磁盘上定位根目录；如果刚才没有创建站点的根目录，也可以单击对话框中工具栏上的按钮新建一个目录，并进行命名，如图 4-4 所示。选择了根目录名后，单击“打开”按钮打开它，然后再单击“保存”按钮关闭 Choose Local Folder 对话框。

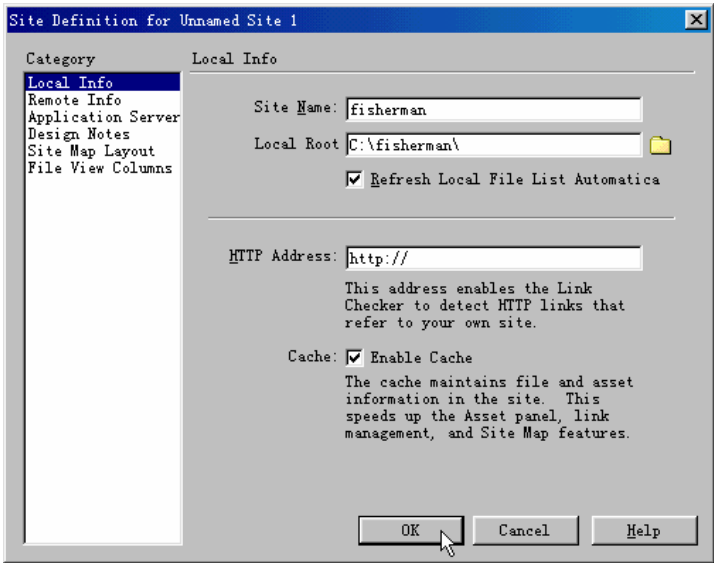


图 4-3
Site
Definition
对话框

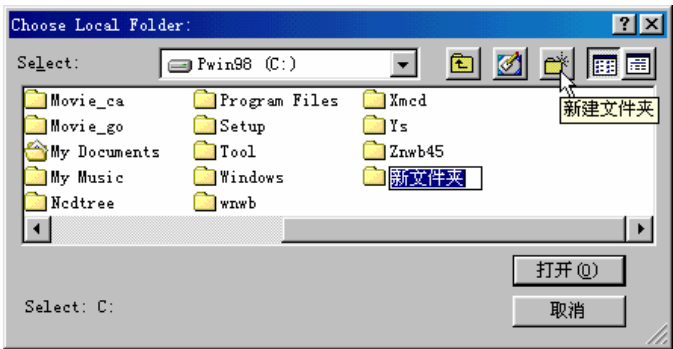


图 4-4
Choose
Local
Folder 对
话框

如果选中了 Site Definition 对话框中的“ Refresh Local File List Automatically ”（自动刷新本地文件列表）复选框，则每当我们往本地站点中拷贝文件，或是从本地站点中删除文件时，都会自动刷新站点窗口中的站点文件列表。

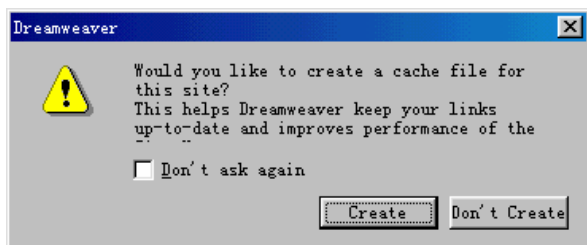
下面还有两项可以不填：“ HTTP Address ”（本地网站的网页文档上传后被浏览时需要输入的 HTTP 地址）文本框和“ Cache ”（缓存）复选框。

“ HTTP Address ”文本框的作用是使 DW UltraDev 的链接检查器可以正确地检查 HTML 代码中的绝对地址，例如假设在此处将 HTTP 地址设为 http://www.yoursite.com ”，这样如果在 HTML 代码中使用了相应的绝对地址（如 http://www.yoursite.com/page01.htm），链接检查器也可以正确识别它。

“ Cache ”复选框的作用是加速链接更新时的速度，同时在站点地图模式中，清晰地反映当前站点的结构。如果硬盘空间不是很小的话，建议选择此项。

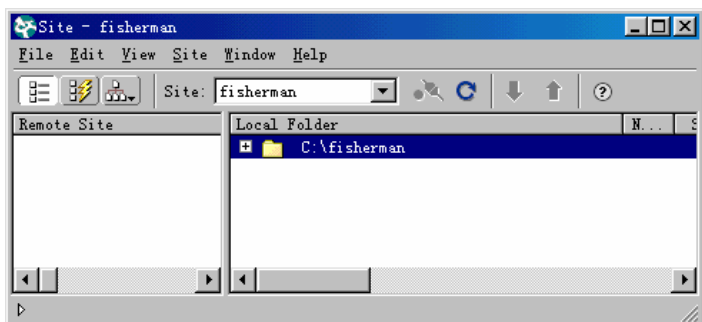
设置结束后,请单击 Site Definition 对话框中的 OK 按钮,确定操作。如果刚才没有选中“Cache”复选框,还会出现一个提示,允许我们再次选择是否创建站点缓存,如图 4-5 所示。

图 4-5
是否创建
缓存



如果单击 Create 按钮,则可以创建缓存。如果单击 Don't Create 按钮,则不创建缓存。无论是否创建缓存,都会创建了一个新的本地站点,如图 4-6 所示。在窗口标题栏上可以看到刚才创建的网站的名称,窗口左侧的远端站点文件列表窗格中目前还是空的,而右侧的本地文件列表窗格中,已经显示了刚才设定的本地文件夹。

图 4-6
新创建的
本地站点



要创建其他的本地站点,可以重复上面的操作。实际上也可以将磁盘上现有的文档组织当作本地站点来打开,只需要在图 4-3 所示的 Local Root 文本框中,填入相应的根目录信息即可。利用该特性,我们可以对现有的本地站点进行管理。从这里我们也可以看出站点的概念同文档不同,文档可以是已经存在的,但是站点则是新创建的,换句话说,站点只是文档的组织形式。

4.4 站点操作

除了创建站点以外,在 DW UltraDev 中,我们还可以对站点进行许多其他的操作,

如打开站点、编辑站点、删除站点和复制站点等。

4.4.1 打开站点

打开 DW UltraDev 已有的站点有两种方法：一种方法是在文档窗口或站点窗口中选择“ Site / Open Site ”菜单项，打开 Open Site 子菜单，其中会列出目前已经创建的所有本地站点名，如图 4-7 中的左图所示。从该菜单中选择某一个站点名即可打开它；另一种方法是在站点窗口中的工具栏上单击下拉列表fisherman右边的按钮，于是会出现一个下拉菜单，其中也列出了目前已经创建的所有本地站点名，如图 4-7 中的右图所示。同样在其中选择某一项即可打开相应的站点。

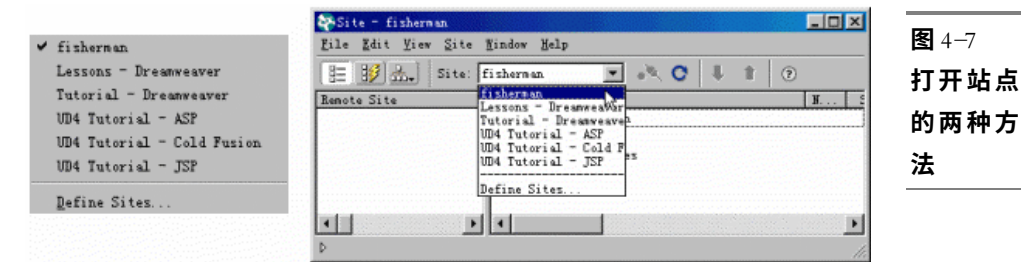


图 4-7
打开站点
的两种方
法

4.4.2 编辑站点

编辑站点需要借助 Define Sites 窗口。打开 Define Sites 窗口也有多种方法：可以在文档窗口或站点窗口选择“ Site / Define Sites ”菜单命令或“ Site / Open Site / Define Sites ”菜单命令打开它；也可以在站点窗口中的工具栏上单击下拉列表fisherman右边的按钮，在下拉菜单中选择“ Define Sites ”菜单命令打开它。图 4-8 所示的就是打开后的 Define Sites 窗口。

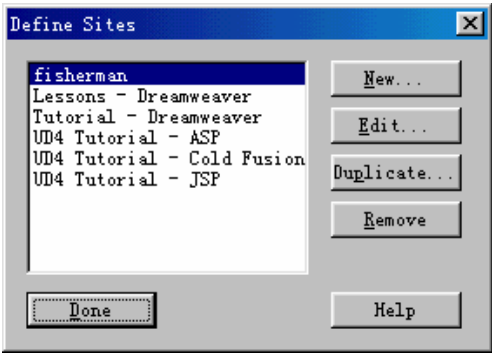


图 4-8
Define
Sites 窗口

Define Sites 窗口的列表中列出了目前已经创建的所有本地站点名。如果要对某一个站点进行编辑,可以单击窗口中的 Edit 按钮,于是就会打开类似于图 4-3 所示的 Site Definition 对话框,其中显示的是该站点的所有属性设置,用户可以对其进行编辑,然后单击 OK 按钮即可完成编辑操作。

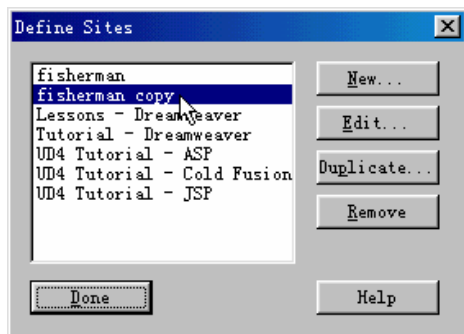
4.4.3 复制站点

有时候希望创建多个结构相同或类似的站点,这可以利用站点的复制特性。首先从一个基准站点上复制出多个站点,然后再根据需要,分别对各站点进行编辑,这能够极大提高工作效率。

如果希望复制站点,可以按照如下步骤进行操作。

首先打开如图 4.8 所示的 Define Sites 窗口,然后从列表中选择要复制的站点名称。单击 Duplicate”(复制)按钮,即可将该站点复制,新复制出的站点名称会出现在 Define Sites 窗口的站点列表中。该名称采用原站点名称后添加“ copy”字样的形式,例如在图 4-9 中我们将站点 fisherman”进行了复制,于是在列表中多了一个名为 fisherman copy”的新站点。接下来可以选中新复制出的站点,然后单击“ Edit”(编辑)按钮,对之进行编辑,例如修改名称等。在返回到图 4-9 所示的对话框后,单击“ Done”(完成)按钮,可以关闭 Define Sites 窗口,完成操作。

图 4-9
复制站点



4.4.4 删除站点

如果不再需要利用 DW UltraDev 对某个本地站点进行操作,则可以将之从站点列表中删除。

首先打开 Define Sites 窗口,然后从列表中选择要删除的站点名称。单击“ Remove”(删除)按钮,这时会出现一个提示对话框,询问是否要删除本地站点(一旦站点被删除,将无法恢复),如图 4-10 所示。选择“是”按钮,即可删除该站点。

删除站点后，会返回到 Define Sites 窗口中。单击“ Done ”按钮，完成操作。

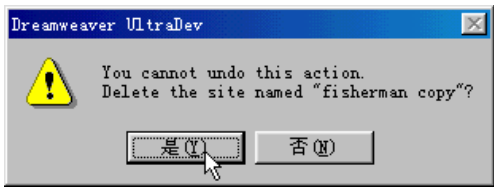


图 4-10
是否删除
站点



上述删除站点的操作实际上只是删除了 DW UltraDev 同该本地站点之间的关系，但是实际的本地站点内容，包括文件夹和文档等，都仍然保存在磁盘相应的位置上，读者可以重新创建一个指向其位置的新站点，重新对其进行管理。

4.5 操作站点文件

无论我们是创建空白的文档，还是利用已有的文档构建站点，都需要对站点中的文件夹或文件进行操作。利用站点窗口，可以对本地站点中的文件夹和文件进行创建、删除、移动和复制等操作。

4.5.1 创建文件和文件夹

在本地站点中创建文件夹是非常简单的。其具体步骤如下。

从站点窗口的本地站点文件列表窗格中，选中要新建文件夹的父级文件夹。然后选择“ File / New Folder ”菜单命令，或者在父级文件夹上单击鼠标右键，在弹出的快捷菜单中选择“ New Folder ”菜单命令，即可创建一个新的文件夹。文件夹被刚创建时，其名称区域处于编辑状态，如图 4-11 所示。读者可以随即输入其名称。输入名称后，单击输入区外任意位置，即可完成文件夹的命名。

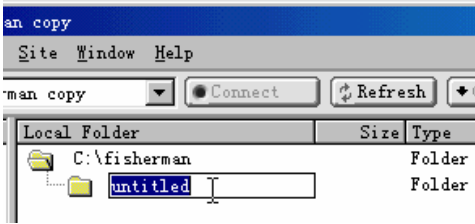
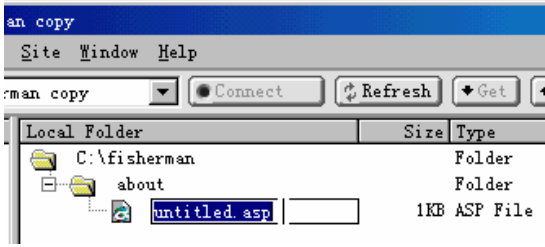


图 4-11
创建文件
夹

创建文件和创建文件夹的步骤类似：

从站点窗口的本地站点文件列表窗格中，选中要保存新建文件的文件夹。然后选择“File / New File”菜单命令，或者在该文件夹上单击鼠标右键，在弹出的快捷菜单中选择“New File”菜单命令，即可创建一个新的文件。文件被刚创建时，其名称区域处于编辑状态，如图 4-12 所示。读者可以随即输入其名称。输入名称后，单击输入区外任意位置，即可完成对文件的命名。

图 4-12
创建文件



如果希望修改文件夹或文件的名称，可以单击文件夹或文件的名称，或按下 F2 键，激活其文字编辑状态，然后输入正确的名称。

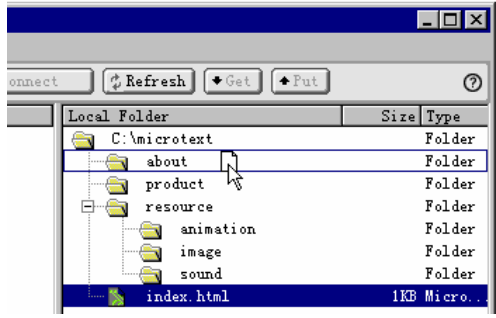
4.5.2 移动和复制文件或文件夹

如同大多数的文件管理器一样，在站点窗口中也可以利用剪切、复制和粘贴操作来实现文件或文件夹的移动和复制。方法如下。

从站点窗口的本地站点文件列表窗格中，选中要移动或复制的文件(或文件夹)。如果要进行移动操作，可以选择“Edit / Cut”菜单命令；如果要进行复制操作，可以选择“Edit / Copy”菜单命令。然后选中目的文件夹，选择“Edit / Paste”菜单命令，文件或文件夹就被移动或复制到相应的文件夹中。

使用鼠标拖动的方法，也可以实现文件或文件夹的移动操作，方法是从站点窗口的本地站点文件列表窗格中，选中要移动或复制的文件(或文件夹)。拖动选中的文件或文件夹，然后移动到目标文件夹中，再释放鼠标，如图 4-13 所示。

图 4-13
拖动文件



另外,使用几个 Windows 通常快捷键 Ctrl+C(复制)、Ctrl+X(剪切)和 Ctrl+V(粘贴)以及鼠标右键的快捷菜单也可以实现移动和复制文件或文件夹的目的。

值得注意的是,如果移动或复制的是文件,由于文件的位置发生了变化,其中的链接信息(特别是相对链接)可能也应该相应发生变化。如果发生了变化,则 DW UltraDev 会打开如图 4-14 所示的对话框,提示是否要更新被移动或复制文件中的链接信息。从列表选中要更新的文件,单击“ Update”(更新)按钮,则更新文件中的链接信息;单击“ Don't Update”(不更新)按钮,则不对被文件中的链接进行更新。

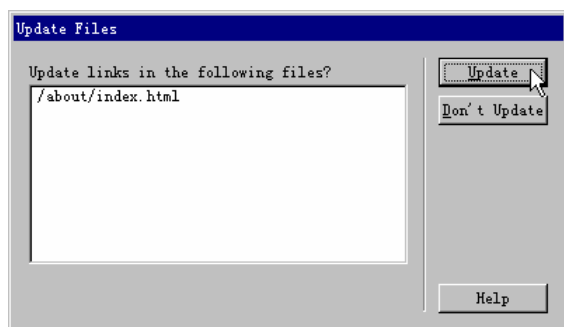


图 4-14
移动文件
时提示更
新链接

4.5.3 编辑和删除文件

要编辑站点文件,可以在本地站点文件列表中,双击相应的文件图标,即可启动 DW UltraDev 的文档窗口,并打开相应的文件等候编辑。文件编辑完毕,从文档窗口中保存文档,即可对本地站点中的文件进行更新。

一般来说,可以首先构建整个站点,同时在各个文件夹中创建好需要编辑的文件。然后再在文档窗口中分别对这些文件进行编辑,最终构建完整的网站内容。

要从本地文件列表窗格中删除文件,可以选中要删除的文件或文件夹,然后选择“ File / Delete ”菜单命令,或是直接按下 Del 键。这时系统会出现一个提示对话框,询问是否要真正删除文件或文件夹,确定操作后,即可将文件或文件夹从本地站点中删除。

值得注意的是,同站点的删除操作不同,这种对文件或文件夹的删除操作,会从磁盘上真正删除相应的文件或文件夹。

在对站点中的文件夹或文件进行了修改之后,需要对本地站点文件列表进行刷新,才可以看到修改后的结果。如果在定义站点时,用户在图 4-3 所示的对话框上选中了“ Refresh Local File List Automatically ”复选框,则文件列表的刷新操作会自动完成。如果没有选中该复选框,则需要手工刷新文件列表,方法是:选择站点窗口中的“ View / Refresh Local ”菜单命令,可以对本地站点的文件列表进行刷新;选择“ View / Refresh Remote ”菜单命令,可以对远端站点文件列表窗格中的内容进行刷新;而单击工具栏上

的  按钮，或者选择“View / Refresh”菜单命令，则可以同时刷新本地站点文件列表和远端站点文件列表。

4.5.4 设置站点主页

在本地站点文件列表窗格中，可以创建作为主页的文档，也将某个现有的文档指定为主页，站点中所有文档的链接关系都可以通过主页来创建。

要创建主页，请在站点窗口中，选择“Site / New Home Page”菜单命令，这时本地站点文件列表窗格中，就会新建一个文件，该文件就是主页。请将该文件重命名为希望的名字。

如果要将现有的文件设置为主页，请在本地站点文件列表窗格中，选中要设置为主页的文件，然后选择“Site / Set as Home Page”菜单命令即可。

上述操作同样也可以借助鼠标右键的快捷菜单来实现。

4.6 使用站点窗口

除了上面介绍的对站点的一些操作以外，我们还可以对站点窗口本身进行一些操作。

4.6.1 查看本地站点

站点窗口中可以同时显示本地站点和远程站点的内容。本地站点可以以文件列表的形式显示，也可以以站点地图的形式显示，或者两者同时显示；而远程站点只能以文件列表的形式显示。在查看一个远程站点之前，必须首先创建一个，这一点我们将在后面的章节中介绍。现在我们先来看一下如何查看本地站点。

在站点窗口工具栏的最左边有三个按钮，其中左边一个是站点文件按钮 ，中间一个是 Web 应用服务器按钮 ，右边一个是站点地图按钮 。在缺省情况下，当读者打开站点窗口时，其中就是文件列表的形式显示站点结构的。左边的窗格显示的是远程站点的文件列表，而右边的窗格显示的是本地站点的文件列表。

如果要显示本地站点的站点地图，则请单击工具栏上的站点地图按钮 ，或者选择“Window / Site Map”菜单命令，即可在站点窗口的左边窗格中显示站点地图，而右边的窗格仍然显示本地站点的文件列表。如图 4-15 所示。

利用站点地图，可以以图形化的方式查看站点结构。在站点地图中，通常可以显示当前站点中文档之间的链接关系，也可以构建文档之间的链接，我们会在后面有关章节

介绍如何利用站点地图构建网页之间的链接。

如果希望重新显示远程站点文件列表窗格，则请单击工具栏上的站点文件按钮，或者选择“Window / Site File”菜单命令，于是站点窗口就会恢复到缺省时的情况，即左边的窗格显示的是远程站点的文件列表，而右边的窗格显示的是本地站点的文件列表。

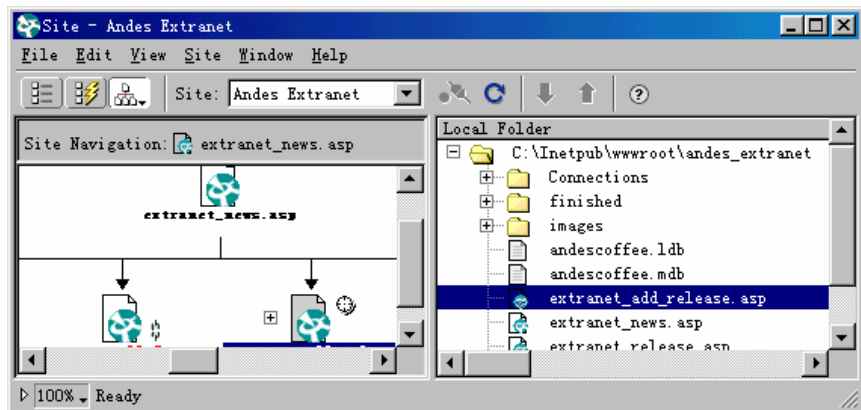


图 4-15
显示站点
地图



如果打开的是磁盘上现有的文档组合来构建本地站点，必须激活站点的 Cache，即缓存，才可以在站点地图中显示文档页面间的链接关系。DW UltraDev 会自动对文档中的链接进行分析，并将之显示。要激活站点的 Cache，可以从图 4-3 所示的对话框上选中“Cache”复选框。

如果用户没有给本地站点设置主页，则也无法显示站点地图，此时系统会出现一个如图 4-16 所示的提示框。读者可以利用 4.5.4 一节介绍的方法，首先设置文档主页。也可以单击提示框中的 Define Sites 按钮，然后单击 Edit 按钮打开相应站点的 Site Definition 窗口，在 Site Map Layout 面板中设置站点的主页。

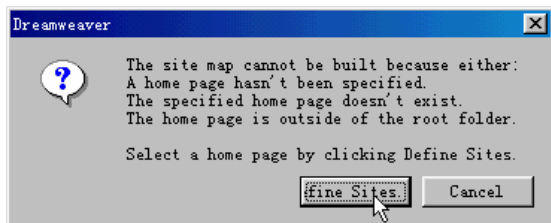
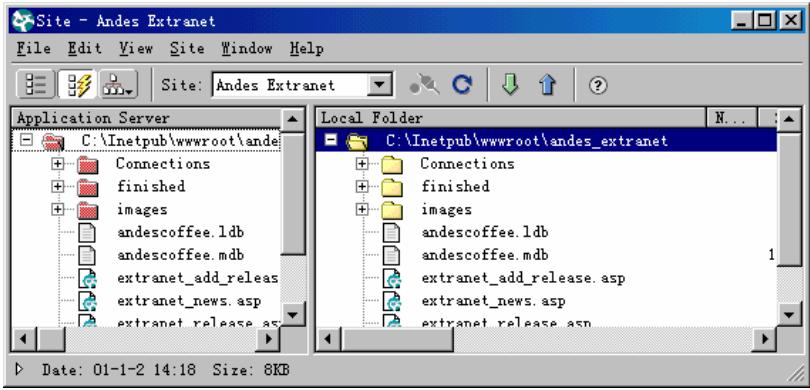


图 4-16
没有为站
点设置主
页

如果要在站点窗口中查看 Web 应用服务器的情况，请单击工具栏上的 Web 应用服务器按钮，于是站点窗口就会变成图 4-17 所示的样子。其中左边的窗格中显示的是 Web 应用服务器的文件列表，而右边显示的仍然是本地站点的文件列表。

图 4-17
查看 Web
应用服务
器



如果希望在整个站点窗口中只显示站点地图，可以在站点地图按钮上按下鼠标左键(不要释放)并停留片刻，这时会出现如图 4-18 所示的下拉菜单，选择“Map Only”菜单命令，则可以隐藏站点窗口右边的本地站点文件列表窗格，而以整个站点窗口的空间显示站点地图，如图 4-19 所示。如果在图 4-18 所示的下拉菜单中选择“Map and File”菜单命令，则站点窗口又会以图 4-15 中所示的形式同时显示本地文件列表和站点地图。

图 4-18
显示下拉
菜单

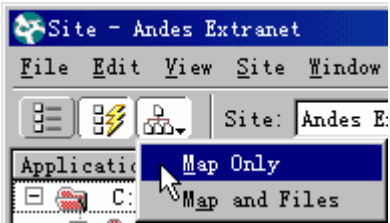
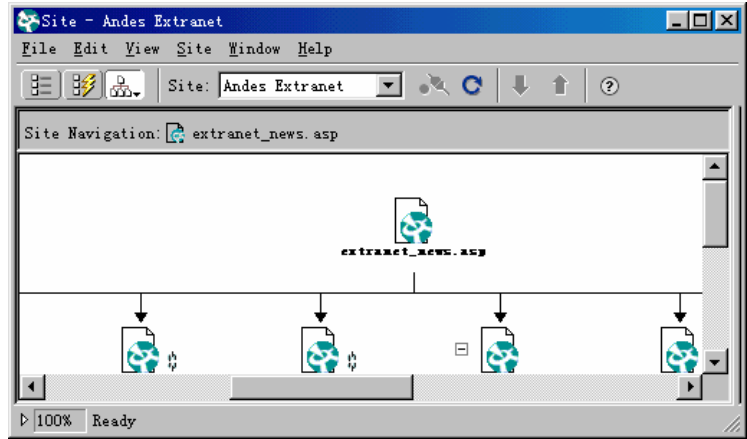


图 4-19
只显示站
点地图



4.6.2 改变站点显示

在缺省情况下，站点窗口中的远程站点文件列表（或本地站点地图）显示在左边的窗格中，而本地站点文件列表显示在右边的窗格中。其实，我们也可以根据自己的喜好改变这个布局。

在站点窗口中选择 **Edit / Preferences** 菜单命令，然后在 Preferences 窗口的 Category 列表中选择“**Site**”项，于是打开如图 4-20 所示的 Site 面板。

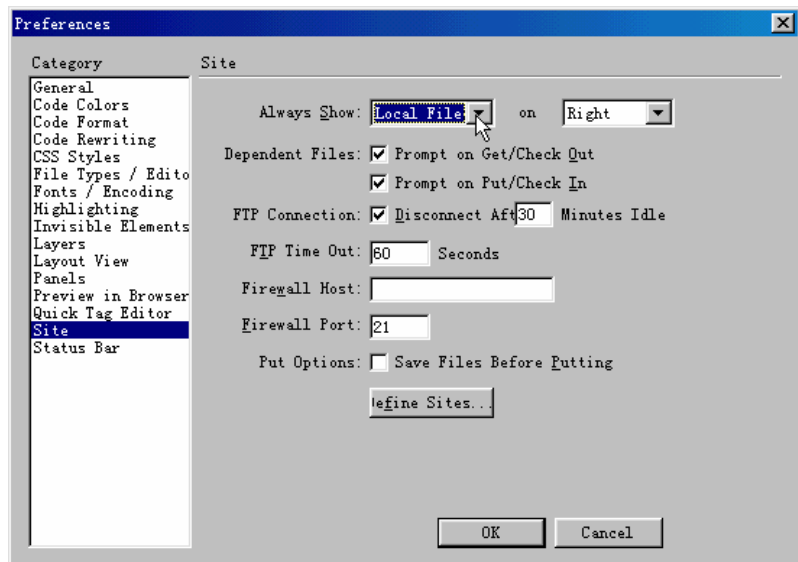


图 4-20
Site 面板

在 **Always Show** 域的第一个下拉菜单中选择“**Local Files**”项，然后在第二个下拉菜单中选择“**Left**”或“**Right**”，这里所选择值决定了站点窗口将在哪个窗格中显示本地站点文件列表。自然，远程站点文件列表（或本地站点地图）就会显示在剩下的那个窗格中。例如，如果我们选择了在左边的窗格中显示本地站点文件列表，则远程站点文件列表（或本地站点地图）就会显示在右边的窗格中。

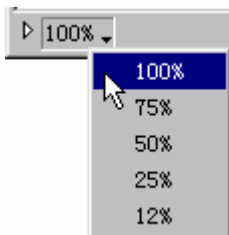
最后单击 **OK** 按钮即可完成设置。

在站点地图中，默认状态下文档图标下显示的是文件名称，这通常不利于识别文件内容，也不便于管理文档。我们也可以将文档图标下的文字显示为文档标题（也即位于 `<title>` 和 `</title>` 标记之间的文字），方法是选择 **View / Show Page Title** 菜单命令即可。

我们还可以改变站点地图的显示比例。在显示了站点地图以后，站点窗口左下角的状态栏中就会出现一个百分比显示框 **100%**。单击该百分比区域，就会打开一个如图 4-21

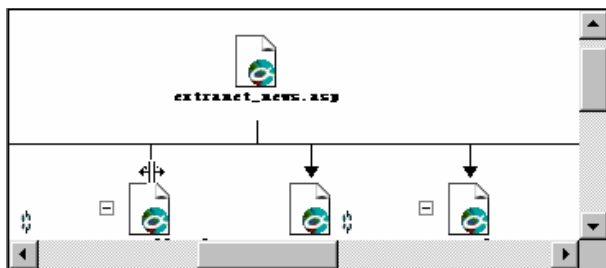
所示的比例菜单，然后选择需要的比例。于是站点地图就会以设定的比例进行缩放显示。

图 4-21
改变站点
地图的显
示比例



除了改变站点地图的显示比例以外，我们还可以改变站点地图中每个文件图标之间的距离。读者在站点地图中的每个分支文件图标的上方都会发现一个向下的箭头↓，将鼠标放置到该箭头上，其开头就会变成+|↓，左右拖动鼠标，就可以改变文件图标之间的距离，如图 4-22 所示。

图 4-22
改变文件
图标之间
的距离



在 DW UltraDev 中，还允许用户设置站点地图的相关选项，方法如下。

选择“ Site / Define Sites ”菜单命令，打开 Define Sites 对话框。在列表中选择要设置的站点名称，然后单击“ Edit ”按钮，即可打开 Site Definition 窗口。在窗口左边的 Category 列表中选择“ Site Map Layout ”项，于是打开如图 4-23 所示的 Site Map Layout 面板。

用户在“ Home Page ”文本框也可以确定站点的主页，既可以直接输入主页的路径和名称，也可以单击右方的  按钮，从磁盘上选择主页文件。在“ Number of Columns ”（列数）文本框中，输入站点地图的列数。在“ Column Width ”（列宽）文本框中，输入站点地图的列宽。在“ Icon Labels ”域，可以设置站点地图中文件图标下方显示的文字类型，如果选择了“ File Names ”单选钮，则在图标下显示文件名称；如果选择了“ Page Titles ”单选钮，则在图标下显示网页的标题。在“ Options ”（选项）区域，可以设置站点地图的其他选项，如果选中“ Display Files Marked as Hidden ”（显示标记为隐藏的文件）复选框，则在站点地图中不仅显示正常的文件，还显示隐藏文件；如果选中“ Display Dependent Files ”（显示依赖文件）复选框，则在站点地图中，显示各个文件之间的链接关系。

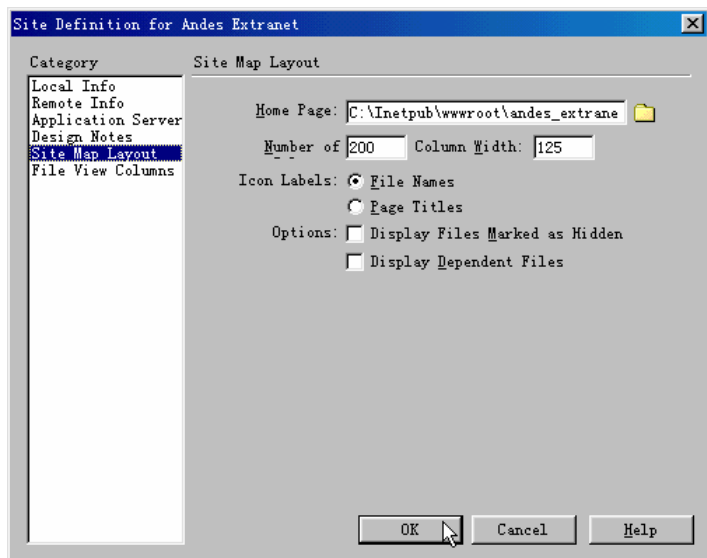


图 4-23
Site Map
Layout 面
板

设置完成之后，单击“OK”按钮，返回 Define Sites 对话框，单击“Done”按钮，即可完成操作。

如果站点中文件列表过长，有限的站点窗口空间就不能很好地显示这些文件列表。这时可以利用 DW UltraDev 的隐藏窗格功能，将远端站点窗格暂时隐藏起来，以充分利用站点窗口空间。

单击站点窗口左下角状态栏上的指向右端的三角形按钮 ，即可隐藏远端站点窗格，在整个站点窗口中只显示本地站点文件列表，同时该三角形按钮  变为指向左端的形状 。再次单击指向右端的三角形按钮 ，又可以恢复成两个窗格的站点窗口状态。

4.7 创建和编辑文档

文档操作可以看作是设计网页的基本操作，它包括打开和编辑文档、控制文档属性、定义文档标题等多个方面，在设计网页时都是必须考虑到的。从一个良好的设计基础入手创作网页，不仅能起到事半功倍的效果，而且也能充分体现创作者的专业能力。

4.7.1 创建空白文档

当我们启动 DW UltraDev 时,系统会直接创建一个空白文档,等候编辑。如果我们要另外创建一个空白文档,可以选择“File / New”菜单命令,此时会另外启动一个 DW UltraDev 文档窗口,使我们在新窗口中编辑新文档,原先的窗口中仍然是以前文档的内容。可以通过在 Windows 的任务栏上单击相应的窗口状态条在两个 DW UltraDev 文档窗口中切换。这种特性和 Word 2000 非常类似。

如果读者打开空白文档的 HTML 源代码检查器,就会发现空白文档其实并不是“空白”的,其中已经包含了 HTML、HEAD、BODY 等标记。

4.7.2 打开现有文档

在 DW UltraDev 已经启动的情况下,选择“File / Open”菜单命令,这时候会出现 Windows 标准的文件打开对话框,选择需要的文件,按下“打开”按钮,即可打开该文档。如果要编辑的文档是由 Microsoft Word 创建的,则可以选择“File / Import / Import Word HTML”菜单命令来打开该文档, DW UltraDev 在打开该文档的同时会自动清除由 Word 所产生的一些无用代码。

同创建空白文档一样,在打开文档时也会启动新的 DW UltraDev 文档窗口,载入被打开的文档。如果该文档已被打开,则会自动切换到载入该文档的 DW UltraDev 文档窗口。

4.7.3 基于模板创建文档

利用模板,可以批量创建具有相同格式的文档。要想创建基于模板的文档,可以选择“File / New from Template”菜单命令。这时会出现如图 4-24 所示的 Select Template 对话框,其中的 Templates 列表中列有所选站点中可使用的模板。选择需要使用的模板,然后单击 Select 按钮,即可基于模板创建新文档。如果选择了 Update Page when Template Change 复选框,则当模板发生变化时,使用该模板的页面自动会进行更新。



要使用模板来创建文档,必须首先创建本地的站点。通常模板会存储在站点的根目录中。当然,用户也必须先在站点中创建模板,具体的创建方法在此不作介绍,请参考相关书籍。

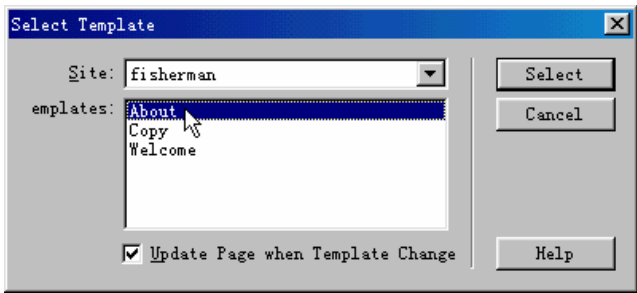


图 4-24
Select
Template
对话框

4.7.4 存储文档

如果同时打开了多个 DW UltraDev 窗口，则切换到要保存的文档所在的窗口。然后选择“File / Save”菜单命令，或是按下 Ctrl+S 组合键。如果文档尚未被保存，则会出现 Windows 标准的文件存储对话框。选择路径并输入文件名，按下“保存”按钮，即可存储文档。如果文档已经被保存，则会直接存储文档，不会出现 Windows 的文件操作对话框。

如果希望将文档以另外的名称保存，可以选择“File / Save As”菜单命令。这时会出现 Windows 标准的文件存储对话框。选择路径并输入新的文件名，按下“保存”按钮，即可将文档以另外的名称保存。

4.7.5 关闭文档

如果要关闭正在编辑的文档，请切换到要关闭文档的窗口，然后选择“File / Close”菜单命令，或者直接单击窗口右上角的  按钮。

如果文档尚未保存，则会出现提示对话框，提示读者保存文档。单击 Yes 按钮则保存文档，单击 No 按钮则不保存文档，单击 Cancel 按钮则放弃关闭操作。

如果系统中只打开了一个 DW UltraDev 文档窗口，则在关闭文档后，会出现一个如图 4-25 所示的对话框，提示是否要同时退出 DW UltraDev。单击 Yes 按钮则同时退出 DW UltraDev；单击 No 按钮则不退出 DW UltraDev，同时又会创建一个空白文档等候编辑。如果不希望再次看到该对话框，可以选中其上的复选框。

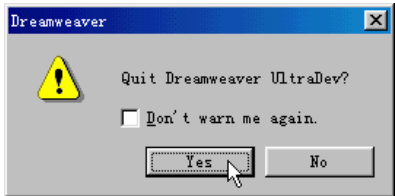


图 4-25
关闭 DW
UltraDev

4.8 添加文本和插入对象

我们可以在文档窗口中输入或粘贴文本，也可以往文档中插入图像、表格、层等常用对象，还能往文档中轻松地添加特殊字符等特殊对象。

4.8.1 添加文本和常用对象

要在文档中添加文本，可以直接在文档窗口中键入文本。或者在其他的应用程序或窗口中复制需要的文本，然后切换回 DW UltraDev 文档窗口，将插入点设置到要放置文本的地方，选择“Edit / Paste”菜单命令即可将剪贴板中的文本粘贴到文档中。

要往文档中插入图像、表格或其他常用对象，可以使用“Insert”菜单，选择要插入对象的菜单项命令，即可将对象插入到当前文档窗口插入点所在的位置。或者也可以打开对象面板，找到要插入对象对应的按钮，单击相应的按钮，或者将该按钮拖动到文档中要放置对象的位置，即可在相应位置插入对象。我们会在相应的章节中介绍各种对象的插入方法。

一般来说，当插入对象时，都会出现一个对话框，允许用户进一步指定设置。如果不希望看到这些对话框，可以选择“Edit / Preferences”菜单命令，打开 Preferences 对话框，从左边的 Category 列表中选择“General”项，然后在右边取消对“Show Dialog When Inserting Objects”（当插入对象时显示对话框）复选框的选择。

4.8.2 插入特殊字符

在 HTML 中，插入特殊字符是一件很麻烦的事，因为特殊字符同普通的字符有所区别，我们无法从键盘上直接输入，而且很多字符或许并不包含在当前文档的字符编码中，要在页面中显示它们，必须使用其他的方法。

一般来说，在 HTML 中，一个特殊字符有两种表达方式，一种称作数字参考，一种称作实体参考。所谓数字参考，就是用数字来表示文档中的特殊字符，通常由前缀“&#”，加上数值，再加上后缀“;”而组成，例如“©”对应于特殊字符“©”，而“®”对应于特殊字符“®”。所谓实体参考，实际上就是用有意义的名称来表示特殊字符，通常由前缀“&”，加上字符对应的名称，再加上后缀“;”而组成，例如，可以使用“©”来表示版权符号“©”，用“®”来表示注册商标符号“®”，很显然，这比数字要容易记忆得多。

遗憾的是，不是所有的浏览器都能够正确认出采用实体参考方式的特殊字符，但是

它们都能够识别出采用数字参考方式的特殊字符，因此，对于一些特别不常见的字符，应该使用数字参考方式。当然，对于那些常见的特殊字符，使用其实体参考方式是安全的，我们在实际应用中，只要记住这些常用特殊字符字符的实体参考就足够使用了。

表 4-1 列出了常用的一些字符实体参考和数字参考。

表 4-1 常见的字符及其参考		
字符实体参考	字符数字参考	显 示
#nbsp;	 	
©	©	?
®	®	?
™	™	?
£	£	£
€	€	€
¥	¥	¥
¢	¢	¢
§	§	§
<	<	<
>	>	>
&	&	&
"	"	"
×	×	×
±	±	±
·	¸	?

尽管记忆字符的参考非常不易，但是在 DW UltraDev 中，插入特殊字符却变得非常简单。单击对象面板上的向下箭头，在打开的下拉菜单中选择“ Characters ”，这时对象面板上就会显示常用特殊字符所对应的按钮，如图 4-26 所示。

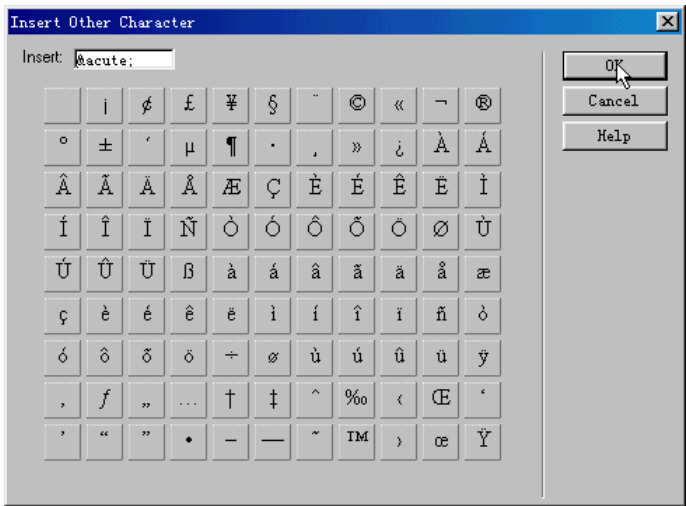
单击相应的按钮，即可插入相应的特殊字符，有些特殊字符插入后使用实体参考方式，而有些字符插入后使用数字参考方式。

如果对象面板上没有需要的字符，可以单击面板上右下角的“ Insert Other Character ”按钮，这时会打开一个如图 4.27 所示的对话框，给我们更多选择。

图 4-26
字符对象
面板



图 4-27
插入其他
字符



在对话框中单击需要的字符，该字符的参考就会出现在“ Insert ”文本框中，单击 OK 按钮，即可将字符插入到文档中。

4.8.3 插入水平线

水平线在 HTML 文档中经常被用到，它主要用于分隔文档内容，使文档结构清晰明白，有时候，合理使用水平线可以获得非常好的效果。一篇内容繁杂的文档，如果合理放置几条水平线，很可能就变得层次分明，便于阅读。

要往文档中插入水平线，请在文档中将插入点放置到要插入水平线的地方。然后单

击常用对象面板 (Common) 上的“ Insert Horizontal Rule”(插入水平线) 按钮, 这时文档中就插入了一条水平线, 选中该水平线, 可以在属性面板上编辑其属性, 如图 4-28 所示。

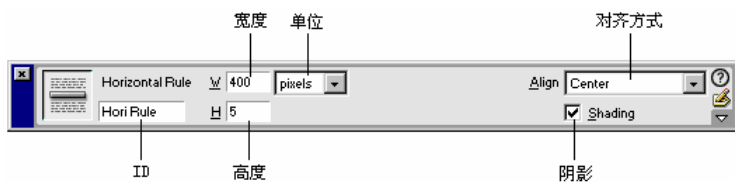


图 4-28
水平线的
属性面板

在“ W ”文本框中, 可以输入水平线的宽度, 这个值可以是像素值, 也可以是针对页面水平宽度的百分比, 这取决于它右方单位列表框中的选择。在“ H ”文本框中可以输入水平线的高度, 它只能以像素作为单位。在“ W ”文本框右边的下拉列表中, 可以选择单位。您可以选择“ pixels”(像素), 或是“ %”(百分比)。在“ Align”(对齐方式) 下拉列表中, 可以水平线在水平位置上的对齐方式。可以是“ Left”(左对齐)、“ Center”(居中对齐)或“ Right”(右对齐), 或者可以选择“ Default”(默认), 使用默认的对齐方式。如果选中“ Shading”(阴影) 复选框, 则水平线带有阴影效果。在属性面板左下方的文本框中, 可以为该水平线指定 ID, 在 HTML 4.0 规范中, 几乎所有的标记都可以带有 ID 属性, 用于标识该对象, 但是在一般情况下都不会使用到它们。

4.9 在文档窗口中选择元素

4.9.1 选择元素的方法

通常, 要在文档窗口中选择某个元素, 只需要单击它, 或是拖动鼠标选中它就可以了。但在实际应用中, 有些元素在屏幕上是不可见 (例如源代码中的注释信息) 的, 而另外一些元素则很难完全选中 (例如一对完整的标记, 它在屏幕上也是看不见的), 这时就需要使用一些其他的手段。一般来说, 在文档窗口中选择元素时, 可以参考如下的方法:

- ✧ 如果要选中的是普通对象或元素, 只需要单击它, 或是拖动鼠标选中它。
- ✧ 如果要选中的是看不见的元素, 可以通过 DW UltraDev 的参数设置对话框, 设置相应的选项, 将这些不可见元素显示, 然后再选中它们。
- ✧ 值得注意的是, 某些对象在页面上出现的位置和它们相应的代码在 HTML 源文件中

插入的位置是不一样的。例如，一个层可以出现在页面上的任何位置，但是定义层的代码只能位于 HTML 源文件中的一个固定位置上。

- ✱ 如果要选中的是标记（包括标记所含的内容），可以利用文档窗口下方的标记选择器来选中它们，如图 4-29 所示。标记选择器上总是显示那些包围当前选择区域或者插入点处的所有标记。

图 4-29
选择标记



例如，假设我们定义了一个图像链接，则其对应的 HTML 代码应该是：

```
<a href="http://www.macromedia.com"></a>
```

如果单击文档窗口中的图像，则标记选择器上的“”标记会自动被选中，同时，在 HTML 源代码检查器中的对应代码“”也会被选中（反相显示）。如果想选择图像链接，则请先单击文档窗口中的图像，然后单击标记选择器上的“”标记左边出现的“<a>”标记即可。

另外，利用“Edit”菜单中的相关菜单项，还可以选中当前标记的父标记或第一个子标记。

4.9.2 选择不可见元素

文档窗口毕竟是可视化操作的窗口，很多实际上在源代码中出现的内容，并不会在文档窗口中直接显示。如果希望对这些不可见元素进行编辑，就不得不转到代码级别的编辑器，如 HTML 源代码检查器中进行。

为了解决这种问题，DW UltraDev 可以在文档窗口中将每种不同类型的不可见元素以一个小图标的形式显示出来，以便于用户在文档窗口中对之定位。实际上，在默认设置下，大多数的不可见元素都可以以图标的方式在文档窗口中显示。有时候需要定制这些显示设置，例如，对于某些始终不会被编辑的不可见元素，可以设置它不在文档窗口中显示，以免影响视觉，而另外一些在编辑过程中要用到的不可见元素，则应该设置它在文档窗口中可显示。设置方法如下：

选择“Edit / Preferences”菜单命令，打开 Preferences 对话框。在左方的 Category 列表中单击“ Invisible Elements ”（不可见元素）项，于是出现如图 4-30 所示的 Invisible Elements 面板。

选中 Show 域中的相应复选框，即可以以相应图标显示该不可见元素；清除相应复选框，则不显示该元素（但是，可以在 HTML 源代码检视器中进行编辑）。表 4-2 中列

出了各个不可见元素对应的图标及其含义。

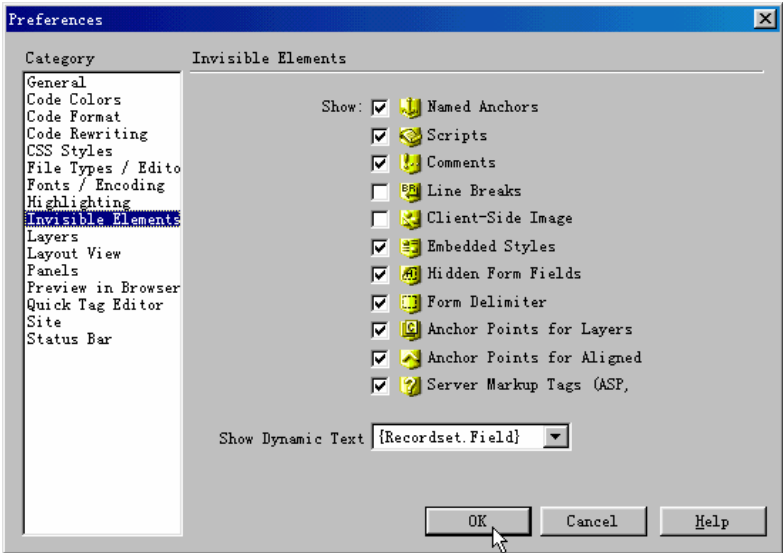


图 4-30 Invisible Elements 面板

表 4-2 各个不可见元素对应的图标及其含义

图 标	含 义
	命名锚
	脚本
	注释
	换行
	客户端映像
	嵌入样式
	隐藏的表单域
	表单定界符
	指向层的锚点
	指向对齐元素的锚点
	服务器端脚本标记（如 ASP、CFML 等）

除此之外，要在文档窗口中显示不可见元素，还需要激活 DW UltraDev 显示不可见

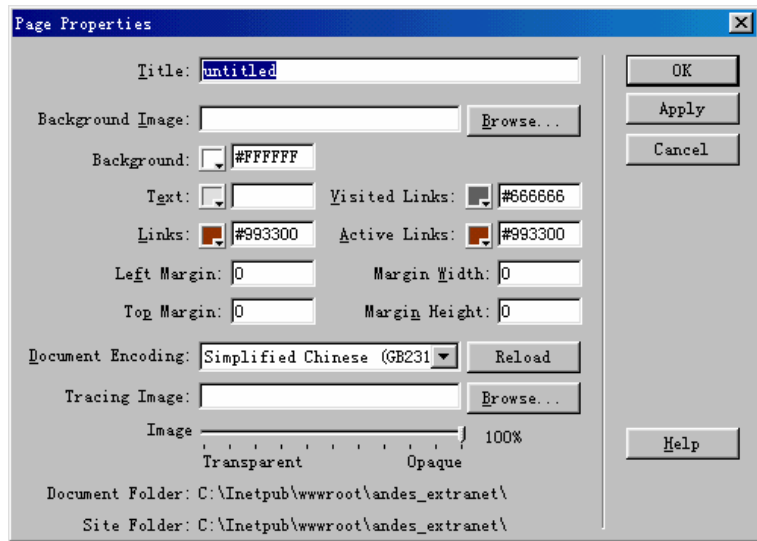
元素的特性，方法是选择“View / Visual Aids / Invisible Elements”菜单命令。当它被选中时，左方会出现选中标记，这时即激活了 DW UltraDev 的显示不可见元素特性，不可见元素可以出现在文档窗口中。再次选择“View / Visual Aids / Invisible Elements”菜单命令，清除对该菜单项的选择，则所有的不可见元素都不会显示在文档窗口中。

4.10 设置文档属性

文档的页面属性包括文档的标题、文档的文字编码方式、文档正文中各个元素的颜色等，正确设置文档属性，是成功编写网页的必要前提。

要设置文档的页面属性，请选择“Modify / Page Properties”菜单命令，或是在文档窗口中单击鼠标右键，在快捷菜单中选择“Page Properties”菜单命令，这时会打开如图 4-31 所示的 Page Properties 对话框，允许用户设置文档页面的整体属性。

图 4-31
Page
Properties
对话框



下面对 Page Properties 对话框中各个选项作一说明。

- ✧ Title(标题)：在文本框中，可以输入文档的页面标题，它实际上是输入位于 HTML 头部的<title>和</title>标记之间的内容，在浏览页面时，会显示在浏览器的标题栏上。
- ✧ Background Image(背景图像)：在文本框中，可以输入希望用作 HTML 文档的背景图像的路径和文件名称，单击 Browse 按钮，可以从磁盘上选择图像文件。这里不仅可以输入本地图像文件的路径和文件名称，也可以用 URL 的形式输入其他位置的图像名称。实际上，这里设置的是<body>标记的 background 属性。

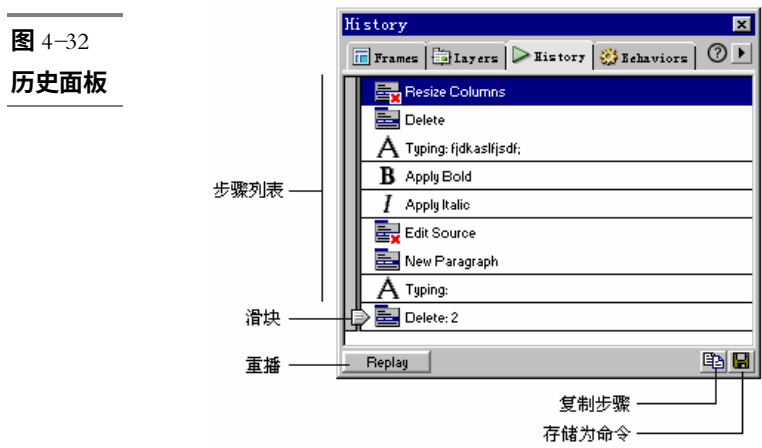
- ※ Background(背景) :在这里可以为文档页面指定背景颜色。实际上设置的是<body>标记的 bgcolor 属性。
- ※ Text(文本) :这里可以设置文档页面中文字的前景颜色,实际上设置的是<body>标记的 text 属性。
- ※ Links(链接) :这里可以设置文档页面中尚未访问过的超级链接的文字颜色,实际上设置的是<body>标记的 link 属性。
- ※ Visited Links(访问过的链接) :这里可以设置文档页面中已经访问过的超级链接的文字颜色,实际上设置的是<body>标记的 vlink 属性。
- ※ Active Links(正在访问的链接) :这里可以设置文档页面中正在访问的超级链接的文字颜色,实际上设置的是<body>标记的 alink 属性。
- ※ Left Margin(左边距)、Top Margin(上边距)、Margin Width(页边宽度) 和 Margin Height(页边高度) :在这些文本框中,可以设置页面文字同页面边缘的间距。实际上分别设置的是<body>标记的 leftmargin、topmargin、marginwidth 和 marginheight 属性。
- ※ Document Encoding(文档编码方式) :在该列表框中,可以选择当前文档页面的文字编码方式,对于大陆地区使用的中文网页,应该选择“ Simplified Chinese (GB2312) ”。如果这里没有需要的编码方式,可以从列表中选择“ Other ”,这种情况下,页面会使用浏览器运行环境的编码方式。
- ※ Tracing Image(草稿图) :在该文本框中,可以指定一幅图像作为网页创作时的草稿图,它显示在文档的背景上,便于在网页创作时进行定位和安放对象。在实际生成网页时,它并不显示在网页中,我们会在后面的章节中介绍其使用方法。
- ※ Image(图像) :拖动该滑块,可以改变草稿图的透明程度,使用适当的透明程度可以避免干扰网页的编辑。
设置完成后,单击 OK 按钮即可。

4.11 使用历史面板

4.11.1 关于历史面板

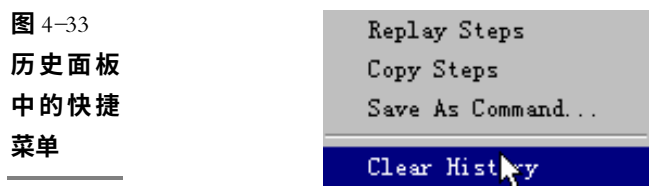
DW UltraDev 的历史面板记录了我们在文档窗口中操作的历史。使用历史面板,可以帮助我们撤销一个或多个步骤,或是重复进行一个或多个步骤的操作。这一点和 Photoshop 中的历史面板非常相似。在网页的创作过程中,历史面板的作用举足轻重。

要显示历史面板，可以选择“ Window / History ”菜单命令，或者按下 F9 快捷键。在历史面板中，会显示当前文档中进行过的操作列表，以及操作的动作内容。一个典型的历史面板结构如图 4-32 所示。



在默认情况下，在历史面板中最多可记录前面 50 步内的操作，当然我们也可以对这个数字进行任意修改。选择“ Edit / Preferences ”菜单命令，打开 Preferences 对话框，从左方目录列表中选择“ General ”项，然后在右方的“ Maximum Number of History Steps ”（历史记录的最大步骤数目）文本框中输入需要的数目值。如前面的图 4-25 所示。

如果确保历史面板中的记录不再需要，也可以将历史面板中的记录清空，以避免可能的误操作。其方法是在历史面板中单击鼠标右键，打开快捷菜单，如图 4-33 所示。也可以单击面板右上角三角图案的按钮，打开相应的菜单。



选择“ Clear History ”菜单命令，这时会出现一个如图 4-34 所示的提示对话框，告诉用户如果清空历史面板，则不能撤销操作。单击“ 是 ”按钮，即可清空历史面板中的所有操作记录。



4.11.2 撤销操作

如果在文档中进行了误操作，可以利用 DW UltraDev 的撤销操作功能，撤销这些操作。

如果仅仅希望撤销前一步操作，可以直接选择“Edit / Undo”菜单命令，或是按下 Ctrl+Z 组合键。连续进行该操作，可以一步一步撤销上次的改变。

利用历史面板同样可以实现对误操作的撤销，并且利用历史面板最大的优势在于，它可以一次撤销多步误操作。其操作方法如下。

在历史面板上，找到希望撤销的最后一步。拖动滑块到该步骤所在的位置，也可以直接单击该位置列表左端位置，这时滑块会自动跳到该位置上，如图 4-35 所示。这时文档窗口中自动恢复到撤销操作后的状态，同时历史面板中被撤销的操作列表会变为灰色。如果希望恢复这些操作，可以重新将滑块拖动到该步骤上，这时被撤销的操作又会被恢复。



图 4-35
利用历史
面板撤销
多个步骤



一旦撤销了多步操作，然后又进行了新操作，则原先被撤销的操作将不可再恢复，历史面板中被撤销的操作列表会消失，而代之以新的操作步骤。

4.11.3 自动化任务

1. 重复操作步骤

在历史面板中，不仅可以撤销已经进行过的操作，而且还可以重复已经进行过的操作步骤，这就是历史面板的自动完成任务特性。利用历史面板，不仅可以在当前的文档

中重复步骤，还可以记录这些步骤，以在其他的文档窗口中重复进行相同的操作。

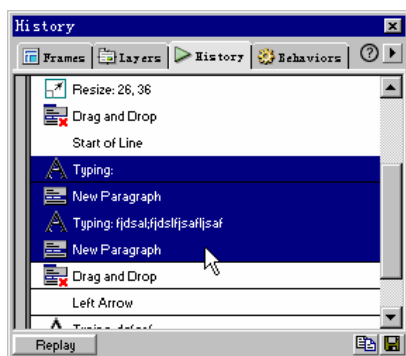
如果仅仅希望重复前面进行的最后一步操作，只需选择“Edit / Repeat”菜单命令即可。这个菜单命令是智能的，它能够反应出用户最后所进行的操作。例如，用户如果输入了“飞雪漫工作室”这6个字，该命令将变成“Repeat Typing”，执行该命令可以重复输入多个“飞雪漫工作室”。但是，如果用户刚刚执行了 Undo 或 Redo 操作，则无法使用 Repeat 命令。

Repeat 命令只能重复最后一次操作，如果想重复前面的任何一次操作，或者想同时重复前面的若干步操作，则需要使用历史面板。

如果想重复某一步操作，请在历史面板的列表中选择某一步操作，然后单击 **Replay** 按钮，即可重复该操作，同时在历史面板列表的最下方也会记录下这一重复操作本身。

如果想重复最近的一系列连续步骤，则可以在列表中的第一个操作上按下鼠标左键不放，然后拖到最后一个操作上放下鼠标左键，或者在第一个操作上单击鼠标左键，然后按下 Shift 键不放，在最后一个操作上单击鼠标左键，并释放 Shift 键，于是这些连续的步骤就被选中，如图 4-36 所示。单击 **Replay** 按钮，即可重复这些连续操作，同时在历史面板列表的最下方也会记录下这一重复操作本身。

图 4-36
选中要重
复的连续
操作



如果想重复前面执行过的一些不连续的操作，则请在列表中单击鼠标左键选中某一个操作，然后按下 Ctrl 键不放，依次单击鼠标左键选择其他各个操作，最后释放 Ctrl 键，于是就可以选中一些不连续的操作。单击 **Replay** 按钮，即可重复这些操作，同时在历史面板列表的最下方也会记录下这一重复操作本身。



在文档窗口中，一些鼠标的移动操作，例如单击或拖动选中的对象的操作，不能被重复进行，也不能被存储为命令。要避免出现这种情况，可以使用箭头键来替代鼠标完成在文档中设置插入点的操作。另外，某些其他类型的步骤也不能被重复，例如在页面中将某个页面元素拖动到另外的位置的操作。当进行这类操作时，在历

史面板中会出现一个带有红色小叉的菜单命令图标,如 Drag and Drop。如果重复执行这样的操作,则会出现如图 4-37 所示的提示窗口,表明该操作无法重复执行,将被忽略。

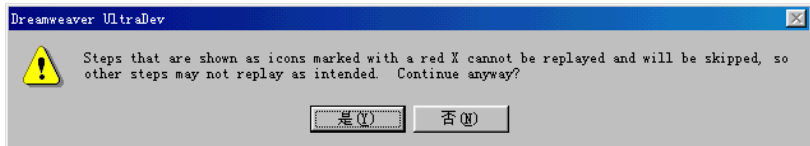


图 4-37
无法重复执行该操作

2. 在其他对象或文档上应用操作步骤

使用历史面板除了可以重复执行以前的一些操作之外,还可以在其他对象上应用操作步骤。方法很简单,只需选中某对象,然后在历史面板中选中要重复进行的操作。单击“Replay”按钮即可。

值得注意的是,如果选择了多个对象,然后利用历史面板来重复操作步骤,则这些对象被作为单个的选中区域来对待,而不会分别对这些对象应用操作。例如,读者无法选中多幅图像,然后利用重复操作的方式重新设置每个图像的大小。

另外,我们不仅可以在当前文档中重复进行某个操作序列,也可以在其他文档中应用当前文档中的操作序列。方法是:在历史面板上,选中希望组合成操作序列的操作项,然后单击历史面板上的“复制步骤”按钮,这时选中的操作就被复制到剪贴板上。打开或者切换到要应用操作序列的文档,将插入点设置到需要的位置,或是选中需要应用操作的对象。选择“Edit / Paste”菜单命令,即可在目标文档中应用相应的操作序列。当然,这时在目标文档的历史面板中只会记录一个粘贴的操作步骤。



如果操作序列中包含复制操作或粘贴操作时,不能在文档间应用这些操作序列。否则会导致不可预料的结果。另外,如果读者将位于剪贴板中的操作序列粘贴到 HTML 源代码检视器中,它会以 JavaScript 代码的形式出现,利用该特性,可以学习这些操作到底是如何用 JavaScript 代码来表示的。

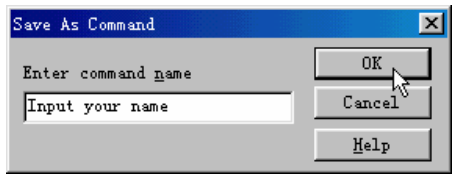
3. 创建并编辑命令

历史面板中的操作列表是很不安全的,在关闭文档后,历史面板中的内容就会消失。如果希望在将来仍然可以重复进行某个操作序列,可以将该操作序列制作成新的命令,该命令会出现在 DW UltraDev 的“Commands”菜单中。

要创建新命令,可以按照如下步骤进行操作。

从历史面板中，选中要组合成步骤序列的操作项。单击“存储为命令”按钮，这时会出现如图 4-38 所示的对话框，提示输入命令的名称。输入适当的名称，然后单击 OK 按钮即可确定操作。

图 4-38
创建新命令



经过上述的操作，新的命令菜单项就会出现在 DW UltraDev 的 Commands 菜单中。下次需要应用该步骤序列时，打开“Commands”菜单，选择该命令项即可。

该操作实际上是将步骤序列对应的 JavaScript 代码存储到 DW UltraDev 安装目录中的 Configuration/Commands 文件夹中。

如果希望为该命令指定快捷键，可以在快捷键字母前添加字符“&”。例如，输入“T&ype My Name”，则菜单上会显示为“Type My Name”，这样，通过按下 Ctrl+C 打开“Commands”菜单，再按下 y 键，即可启动该命令。

如果希望改变命令的名称，或删除命令，可以按照如下方法进行。

选择“Commands / Edit Command List”菜单命令，这时会出现如图 4-39 所示的 Edit Command List 对话框，上面显示所有自定义的命令。要修改某个命令的名称，可以单击该命令项，这时该命令项变为可编辑状态。输入需要的名称，单击该命令项外任意位置，即可实现对命令的名称的修改。如果希望删除某条命令，可以选中该命令，然后单击“Delete”按钮即可。单击“关闭”按钮，可以关闭该对话框。

图 4-39
编辑命令列表



4. 录制命令

在 DW UltraDev 中, 还允许我们录制一组操作过程, 将之作为一个命令来保存。这有些像 Word 中的录制宏特性。激活录制特性后, 在文档中进行的操作都会被记录下来, 直到关闭录制特性为止。录制命令操作同我们利用历史面板进行的重复步骤操作不同, 具体表现在下面几个方面:

- ※ 录制命令是在文档中正在进行的, 而在历史面板中选择步骤序列, 实际上选中的是已经操作过的步骤。
- ※ 录制命令时, DW UltraDev 会自动防止记录那些不能录制的命令, 例如鼠标的移动或对象的选择。
- ※ 在录制命令时, 如果切换到其他的文档窗口, DW UltraDev 不会录制在其他文档窗口的操作。
- ※ DW UltraDev 一次只能录制一个命令, 一旦开始录制新的命令, 则原先的命令就会丢失, 因此 DW UltraDev 的录制命令特性适用于那些需要暂时使用的操作过程。所以, 如果希望将命令永久保存, 可以在录制命令之后, 通过历史面板选择这些步骤序列, 并将之存储为命令。

要录制命令以供临时使用, 可以按照如下步骤进行操作。

选择打开“ Commands / Start Recording ”(开始录制) 菜单命令, 或是按下 Ctrl+Shift+X 组合键, 这时鼠标指针变为一个“录音带”的形状 。然后进行需要的文档操作, 这些操作会被录制下来。如果希望停止录制, 可以选择“ Commands / Stop Recording ”(停止录制) 菜单命令, 或是再次按下 Ctrl+Shift+X 组合键。

命令被录制好后, 可以按照如下的方法在文档中使用:

选择“ Commands / Play Recorded Command ”(播放录制的命令) 菜单命令, 即可使用被录制的命令。也可以通过按下 Ctrl+P 组合键来应用被录制的命令。

被录制的命令只适合暂时使用, 如果再次录制新命令, 原先的命令就会被覆盖。如果希望永久保存录制的命令, 可以按照如下方法进行操作:

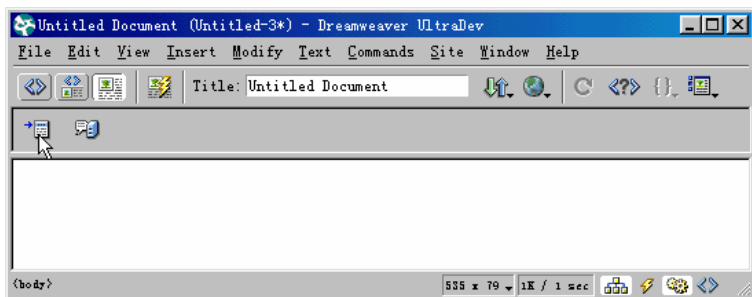
在文档中使用一次录制的命令, 这时历史面板上会出现一个操作项, 标以“ Run Command ”  Run Command。在历史面板中, 选中该“ Run Command ”操作项。单击面板右下角磁盘形象的按钮 , 然后在出现的对话框中输入命令名称。按下“ OK ”按钮, 即可将该录制的命令保存。这时该命令就会出现在“ Commands ”(命令) 菜单中。

4.12 查看和编辑网页头部内容

4.12.1 显示头部元素

在文档窗口中，主要显示的是正文部分的内容，也即位于<body>和</body>中的内容（我们暂时不考虑框架的情况）。实际上很多不可见元素是位于 HTML 的头部的，也即位于<head>和</head>标记之间，如果希望看到这些头部内容，则请选择“View / Head Content”（头部内容）菜单命令，当该菜单项被选中时，会在左方出现选中标记。这时在 DW UltraDev 的文档窗口上方中会多出一栏，其中显示头部内容，如图 4-40 所示。

图 4-40
显示头部
内容



再次打开“View”（查看）菜单，选择“Head Content”（头部内容）命令，清除对该项的选中，则又可以从文档窗口中隐藏头部信息。

在默认状态下，在文档的头部窗格中我们可以看到两个不可见元素标记：一个是文档的标题，也就是位于<title>和</title>之间的文本；另一个是 META 标记，定义当前文档的文字编码方式。

4.12.2 编辑网页的头部内容

我们已经知道，一个 HTML 文件，通常由包括在<head>和</head>标记间的头部，和包括在<body>和</body>标记的正文两个部分组成。文档的标题信息就存储在 HTML 的头部位置，在浏览页面时，它会显示在浏览器的标题栏，当将页面放入浏览器的“收藏夹”时，文档的标题又会作为收藏夹中项目的名称。

除了标题之外，头部还可以包括很多非常重要的信息，例如文档的类型、语言的编码方式、JavaScript 和 VBScript 的函数和变量，以及针对每个引擎的关键字和内容指示

符等。一般来说，不需要输入全部的头内容。但是最好为文档设置标题和编码方式信息，养成良好的创作习惯。

要往文档的头添加信息，可以先在文档窗口中显示文档的头内容。然后在对象面板上，打开其上的下拉菜单，选择“Head”菜单命令，打开如图 4-41 所示的 Head 对象面板。

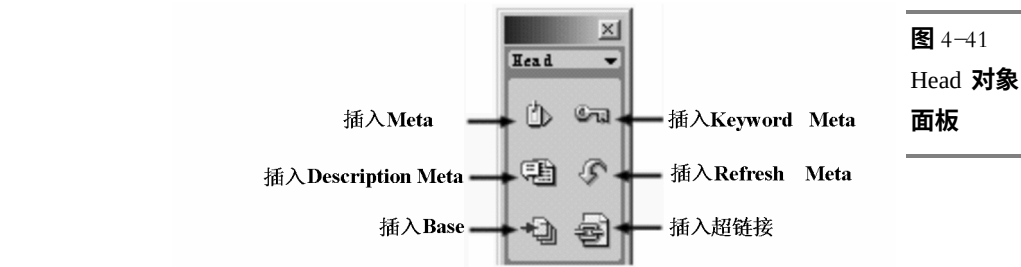


图 4-41
Head 对象
面板

要插入某种元素，可以单击相应的按钮，这时会出现一个对话框。在其中输入需要的信息，确定操作，即可往文档的头添加数据。如果希望编辑头部信息，可以在文档窗口的头部窗格单击选中相应的标记，然后在属性面板上修改。

除了使用对象面板之外，也可以通过“Insert / Head Tags”菜单命令打开“Head Tags”子菜单，再选择相应的命令来实现头部元素的插入。

下面以在页面中插入自动刷新时间为例，介绍插入并编辑网页头部内容的方法。我们可以为网页设置自动刷新特性，让它在被浏览器显示时，隔一段指定的时间，然后跳转到某个页面或是重新刷新自身。方法是单击 Head 对象面板上的“Insert Refresh”按钮, 这时会出现如图 4-42 所示的对话框。

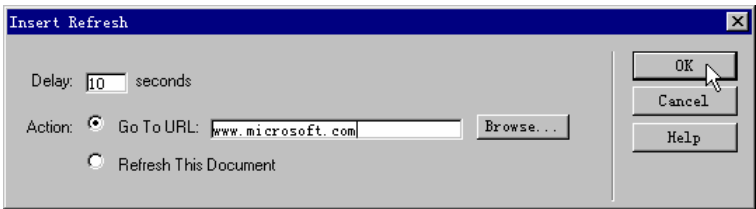


图 4-42
定义刷新
特性

在“Delay...seconds”文本框中，输入网页刷新的间隔时间。在“Action”域可以设置刷新的动作，读者可以设定在指定的时间间隔到达后跳转到某个页面上，或是仅仅重新在浏览器中刷新自身。设置完毕，按下 OK 按钮确定。

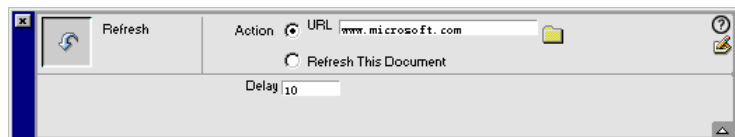
自动刷新特性目前已经被越来越多的网页所使用。例如，可以首先在一个页面上显示欢迎信息，经过一段时间，自动跳转到指定的网页上。另外一种使用自动刷新特性的原因是网站地址的迁移，这时可以首先在原先网址的主页上显示新的网址信息，以通知访问旧网址的用户，然后经过一段时间，自动跳转到新地址上。

图 4-42 的设置实际上添加了如下的语句，我们可以看到，它仍然是元数据的一种表现形式。

```
<meta http-equiv="refresh" content="10;URL=www.microsoft.com">
```

要编辑自动刷新设置，可以从文档的头部窗格中，选中刷新标记，然后在属性面板上修改，如图 4-43 所示。值得注意的是，读者需要展开属性面板才能看到定义延迟时间的文本框。

图 4-43
设置自动
刷新时的
属性面板



4.12.3 认识头部元素

元数据(meta data) 是 HTML 头部的主要组成部分，主要用于表示一个文档的页面信息，这些信息可以被计算机识别，用于某些特定的场合。在 HTML 中，元数据的表现方式是多种多样的。除了元数据之外，在 HTML 头部还可以包含其他类型的标记，例如 <base>标记和<link>标记等。

1. <meta>标记

<meta>标记是实现元数据的主要标记，它能够提供文档的关键字、作者、描述等多种信息，在 HTML 的头部可以包括任意数量的<meta>标记。<meta>标记是非封闭类型的标记，它包括如下几种属性：

- ✧ name = string
name 属性用于定义一个元数据属性的名称，其中 string 是用于表示名称的字符串。
- ✧ content = cdata
content 属性用于定义元数据的属性值，其中 cdata 表示相关数据。
- ✧ scheme = cdata
scheme 属性说明用于解释元数据属性值的机制，其中 cdata 表示相关数据。
- ✧ http-equiv = string
http-equiv 属性可以用于替代 name 属性，HTTP 服务器可以使用该属性来从 HTTP 响应头部收集信息，其中 string 是用于表示名称的字符串。
- ✧ charset = encoding
charset 属性用于定义文档的字符编码方式，其中 encoding 表示编码方式。

在 HTML 规范中，并没有硬性指定的属性标准，因此用户可以自行定义元数据。例如，可以采用如下语句定义作者信息。

```
<meta name = "author" content = "俞俊平">
```

有些搜索引擎使用关键字(keywords) 和描述(description) 来进行搜索。这时，在文档中可以提供如下语句：

```
<meta name = "keywords" content = "俞俊平, 飞雪漫工作室,Fisherman">
<meta name = "description" content = "这是飞雪漫工作室的主页">
```

2. <base>标记

<base>标记定义了文档的基准 URL 地址，文档中所有的相对地址形式的 URL 都是相对于这里定义的 URL 而言的。一个文档中的<base>标记不能多于一个，必须放于头部，并且应该在任何包含 URL 地址的语句之前。<base>包含如下的属性：

- ※ href = url
href 属性指定了文档的基准 URL 地址。该属性在<base>标记中是必须存在的。
例如，如果希望将文档的基准 URL 定义为“ http://www.microsoft.com ”，则可以使用如下语句：

```
<base href = "http://www.microsoft.com">
```

当定义了基准 URL 地址之后，文档中所有引用的 URL 地址都从该基准 URL 地址开始。例如，对于上面的语句，如果文档中一个超级链接指向 china/document.html，则它实际上指向的是如下 URL 地址：

```
http://www.microsoft.com/china/document.html
```

- ※ target = frameset
target 属性同框架一起使用，它定义了当文档中的链接被点击后，在哪一个框架窗口中展开页面。如果文档中超级链接没有明确指定展开页面的目标框架窗口，则就使用这里定义的地址代替。其中 frameset 可以被设置如下的值，如表 4-3 所示。

表 4-3 target 的属性值

属 性 值	含 义
_blank	表明在新窗口中展开链接指向的页面
_self	在当前文档的框架窗口中展开页面
_parent	在当前文档的父级框架窗口中展开页面
_top	在链接所在的完整窗口中展开页面

3. <link>标记

<link>标记定义了文档之间的包含。在 HTML 的头部可以包含任意数量的<link>标记。由于有些浏览器并不能很好处理 link 属性，因此不建议使用它。<link>标记带有很多属性，下面是一些常用的属性：

※ href = url

href 属性指向链接资源所在的 URL，其中 url 是链接的地址。

※ title = string

title 属性用于描述该链接关系。其中 string 是描述关系的字符串。

※ rel = linktype

rel 属性定义了文档和所链接资源的链接关系。其中，linktype 表明链接类型，可能的值包括 Alternate、Stylesheet、Start、Next、Prev、Contents、Index、Glossary、Copyright、Chapter、Section、Subsection、Appendix、Help 和 Bookmark 等。如果希望指定不止一个链接关系，可以在这些值之间用空格隔开。

※ rev = linktype

rev 属性定义了文档和所链接资源之间的反向关系。其中 linktype 表明链接类型，其可能的取值同 rel 属性相同。

第 5 章 编辑 HTML 页面

本章概要：

- ☒ Code Inspector
- ☒ 编辑 HTML 标记
- ☒ 设置 HTML 格式特性
- ☒ 净化 HTML 源代码
- ☒ 外部 HTML 编辑器
- ☒ 其他编辑功能

DW UltraDev 4 为我们创建和编辑 HTML 文件提供了图形界面工具，这使我们的工作更加方便快捷，但是有时我们也需要直接访问底层的 HTML 代码，利用 DW UltraDev 4 的 Code Inspector 和 Quick Tag 编辑器可以满足我们的要求。

另外，DW UltraDev 4 也为那些不是 DW UltraDev 4 创建的 HTML 文件的处理提供了有效的方法。利用 Roundtrip HTML，我们可以确保 DW UltraDev 不会改变手工 HTML 代码，除非你希望它这样做；我们可以设置 DW UltraDev 4 重写内容以及不重写内容的各种选项。相反，利用 Clean Up HTML 和 Clean Up Word HTML，DW UltraDev 4 可以清除性能中混乱的和不必要的 HTML。

5.1 Code Inspector

HTML 源码检查器(Code inspector) 为我们显示当前文档的 HTML 源代码，并按照 HTML 颜色设置配置，以彩色显示各种标记。当我们增加或修改 Document 窗口的内容时，DW UltraDev 4 会立即在 HTML 源码检查器中显示当前所做的修改。因此，如果你还不熟悉 HTML 语言，那么通过 HTML 源码检查器来学习 HTML 是个很好的途径；如果你对 HTML 比较熟悉，那么 HTML 源码编辑器也会为你提供许多方便，它可以提醒你指定标记或属性的正确语法。同时，如果你对自己有足够的信心的话，你也可以打开

HTML 源码检查器对 HTML 源码进行直接修改。

在文档窗口所做的修改会立即反映到 HTML 源码检查器中。在 HTML 源码检查器中所做的修改，只有当你单击 HTML 检查编辑器之外的地方时，你在 HTML 检查编辑器中所做的变化才会在 Document 窗口中反映出来。

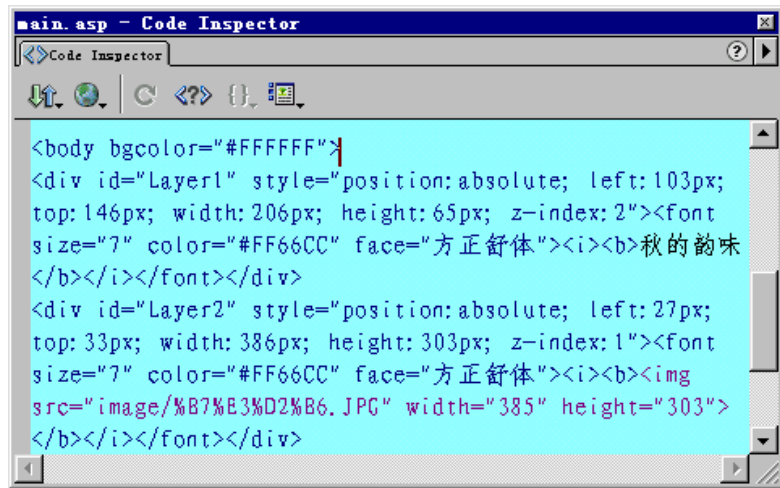
DW UltraDev 4 不会重写你在 HTML 源码检查器中直接键入的 HTML 代码。如果你在 HTML 源码检查器中键入了无效的 HTML 代码，那么当你在 Document 窗口单击时，DW UltraDev 将以高亮状态突出显示无效的标记。

打开 HTML 源码检查器有以下几种操作方法：

- ※ 选择“Window / Code Inspector”菜单；
- ※ 在启动面板或小型启动面板（在 Document 窗口的右下角）中单击 Code Inspector 图标。
- ※ 按 F10 键。

打开的 HTML 源码检查器如图 5-1 所示：

图 5-1
Code
Inspector



HTML 源码检查器面板为我们提供了以下一些操作按钮：

1. 文件管理按钮 (File Management)

此按钮包括以下一些子菜单：

- Turn off Read Only
- Get
- Check out
- Put
- Check in

- Undo Check out
- Design Notes
- Locate in sites

2. 预览/调试按钮 (Preview/Debug in Browser🌐)

此按钮包括以下一些子菜单：

- Preview in explore
- Debug in explore
- Edit Browser List

3. 刷新设计视图按钮 (Refresh design view🔄)

单击该按钮，当前所作的代码修改，会立即反映到文档界面上。

4. 引用按钮 (Reference🔍)

单击此按钮，将弹出 Reference 对话框，列出当前光标所在文档代码引用的标记，并给予解释，如图 5-2 所示：

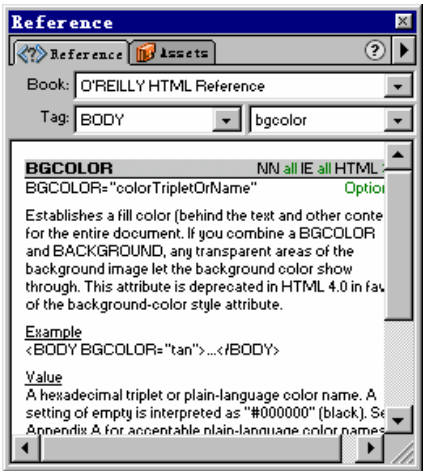


图 5-2
Reference
对话框

由图 5-2 可以看到此对话框还有一个 Assets 面板，该面板显示了当前站点所引用的所有资源，如图 5-3 所示。

5. 代码导航按钮 (Code Navigation🔍)

此按钮包括以下一些子菜单：

- Set breakpoint

- Remove All Breakpoints
- no functions defined

图 5-3
Assets 面
板



6. 显示控制选项按钮 (View Options)

此按钮包括以下一些显示控制选项：

- Word Wrap
- Line numbers
- Highlight Invalid HTML
- Syntax Coloring
- Auto Indent

我们可以通过选择 Wrap(绕排) 选项或取消 Wrap 选项的选择来打开或关闭行绕排设置。打开行绕排不会在 HTML 代码中插入断行符，它只是简单地使行绕排以便于查看而已。也可以通过选择 Line Numbers(行号) 选项或取消 Line Numbers 选项的选择来确定是否显示行号。



同时按下 Ctrl+Tab 键，可以实现在 Document 窗口和 HTML 源码检查器窗口之间切换控制焦点。

要编辑少量的 HTML 标记，可以使用 Quick Tag(快速标记) 编辑器，而无需显示 HTML 源码检查器。为了显示快速标记编辑器，可按 Control+T 键，或选择 Modify / Quick Tag Editor ” 菜单。

5.2 编辑 HTML 标记

本节介绍一下在 Document 窗口中如何编辑一个 HTML 标记。

利用 Quick Tag(快速标记)编辑器,我们可以在 Document 窗口内快速检查和编辑一个标记的 HTML 源代码,而不需要首先切换到 HTML 源码检查器,修改之后,再返回到 Document 窗口。打开 Quick Tag 编辑器最简单的方法是同时按下 Control+T(Windows 系统)键或 Command+T(Macintosh 系统)。

Quick Tag 编辑器有以下三种模式:

- 插入 HTML,用于插入新的 HTML 代码。
- 编辑 Tag,用于编辑一个已存在的标记。
- 封装 Tag,用于为当前选定内容封装一个新的标记。

Document 窗口内的当前选定内容决定 Quick Tag 编辑器打开的模式。



当 Quick Tag 编辑器处于打开状态时,我们可以利用打开 Quick Tag 编辑器的同一键盘快捷键在各种模式之间来回切换。

在这三种模式下,Quick Tag 编辑器的基本操作是相同的:打开编辑器、输入或编辑标记和属性,然后关闭编辑器。这三种模式之间的差别在于当我们编辑标记时,DW UltraDev 对标记处理的内容。



无论 Quick Tag 编辑器以何种模式显示,我们都可以拖放编辑器的句柄以移动编辑器。

虽然 Quick Tag 编辑器可以为我们提供方便快捷的编辑方式,但如果你希望实现比较复杂的 HTML 编辑功能,还是请使用 HTML 源码检查器。

1. 在 Quick Tag 编辑器中输入或编辑标记和属性

其步骤如下:



下面两种方式任选一种:



选定一个对象、一段文本或一个标记。



不选定任何内容,单击 Document 窗口以放置一个插入点。

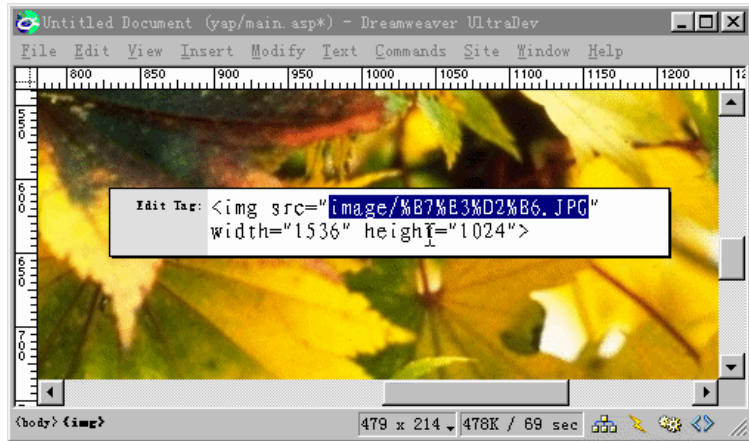


按下 Control+T(Windows 系统)或 Command+T(Macintosh 系统),或者单击 Property

检查器中的 Quick Tag Editor 按钮.

此时 Quick Tag Editor 就会根据当前选定内容,以正确的模式打开。图 5-4 展示的是以编辑标记(Edit Tag)模式打开的 Quick Tag Editor。

图 5-4
Quick Tag
Editor



此时,在插入 HTML 模式或封装标记模式下,我们就可以输入 HTML 标记了。在编辑标记模式下,我们可以输入新的属性、编辑已有属性或编辑标记的名称。

-  如果要编辑已有属性值或名称(或编辑标记名称),按 Tab 键在各元素之间移动,直到选定你希望编辑的目标属性或标记。按下 Shift+Tab 键时移动方向相反,从后向前移动。
-  如果在编辑一个属性名称时你停留了片刻,那么将会出现一个下拉式提示菜单(hint menu),它为你列出当前正编辑的标记拥有的所有有效属性。如果 DW UltraDev 4 不能识别正在编辑的标记,此时提示菜单将列出 DW UltraDev 4 能识别的所有标记的所有属性。



Hints drop-down menus 显示 TagAttributeList.txt 文件中包含的所有标记和属性,该文件位于 DW UltraDev 安装路径下的 Configuration 文件夹中。读者可以从此文件中增加或移走标记,以便更好的辅助你编辑 HTML 源码。

为了利用 DW UltraDev 4 为我们提供的这一方便功能—提示菜单,我们必须选中 Preferences 对话框中的 Quick Tag Editor 面板上的 Enable Tag Hints 选项。如图 5-5 所示。

控制提示菜单有以下几种方法:

- ✧ 当你开始键入属性名称时,提示菜单将滚动到以你键入的字母开始的属性名称处,并高亮状态显示此属性名称。
- ✧ 为了在提示菜单中移动高亮状态显示的位置,可以利用向上箭头键(Up)和向下箭

头键(Down)，也可以利用提示菜单本身的滚动条。

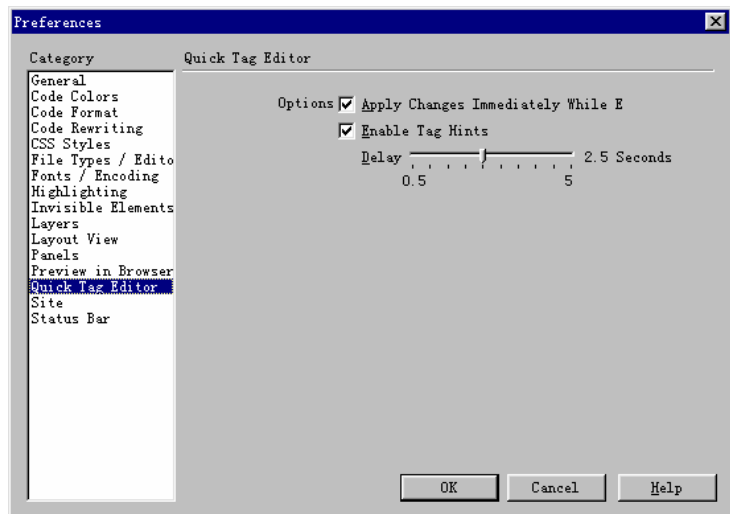


图 5-5
设置提示
菜单的显
示状态

※ 如果提示菜单中当前高亮显示的属性名称就是你所需要的属性名称，直接按下 Enter 键(Windows 平台)或 Return 键(Macintosh 平台)，此时，高亮显示的属性名称就自动输入 Quick Tag Editor。另一种选择方法，就是双击提示菜单中你所需要的属性名称，也能达到同样的效果。

※ 如果不需要选择任何属性名称，并希望取消提示菜单，我们可以按下 Esc 键，或者更简单的方法就是继续输入你想输入的内容。此时提示菜单自动消失。

如果我們是在输入或编辑标记名称时稍作暂停，那么将显示的是一个类似的下拉提示菜单，只是该提示菜单中显示的内容不同而已，此时它为我们列出所有可用的标记名称，而不是属性名称。

5 在编辑标记模式下，如果你对 HTML 源码进行了一个修改，然后按下 Tab 键或 Shift+Tab 键，那么你所作的修改会立刻生效。如果我们不希望发生这种情况，也就是说，我们希望退出 Quick Tag Editor 时，所作的修改才应该生效，那么我们应该去掉 Preferences 对话框中 Quick Tag Editor 面板上的 Apply Changes Immediately While Editing 选项，如图 5-5 所示。



如果没有选中 Apply Changes Immediately While Editing 选项，那么你所作的多步修改在 History 面板中只当作一步来保存，并且 Undo 操作也是将其当作一步来处理的。

6 为了给标记增加新属性，利用 Tab 键、箭头键以及空格键移动插入点，将它放置到

你希望增加属性的位置。接着为标记键入一个有效的属性。当你键入时稍作暂停，就会像编辑属性时一样，将出现一个下拉提示菜单。



继续编辑和增加需要的属性名称和值。



当完成标记编辑时，有以下几种方式退出 Quick Tag Editor：

- ※ 如果希望退出 Quick Tag Editor，并且抛弃所有还未应用的修改，那么请按 Esc 键。（如果当前使用的编辑标记模式，并选中 Apply Changes Immediately While Editing 选项，那么所有已经应用的修改都会生效。）
- ※ 如果希望退出 Quick Tag Editor，并且应用所作的修改，那么请按 Enter 键（Windows 平台）或 Return 键（Macintosh 平台），或者单击 Quick Tag Editor 以外的任何地方。
- ※ 如果你在 Quick Tag Editor 中使用了任何无效的 HTML 代码，在关闭 Quick Tag Editor 时，DW UltraDev 将显示一个警告信息对话框，如图 5-6 所示，并自动去掉无效代码。

图 5-6

无效 HTML
代码警告对
话框



2. 选择当前标记的父标记

以下两种方法任选其一：

- ※ 同时按下 Control+Shift+<键（Windows 平台）或 Command+Shift+<键（Macintosh 平台）。
- ※ 选择“Edit / Select Parent Tag”菜单项。

3. 选择当前标记包含的第一个子标记

以下两种方法任选其一：

- ※ 同时按下 Control+Shift+>键（Windows 平台）或 Command+Shift+>键（Macintosh 平台）。
- ※ 选择“Edit / Select Child”菜单项。



如果当前标记不包含任何其他标记，那么 Select Child 将当前标记的内容。

4. 切换 Quick Tag Editor 的三种模式

按下 Control+T 键(Windows 平台)或 Command+T 键(Macintosh 平台),将会启动 Quick Tag Editor。之后,每按一次刚才使用的启动快捷键,就会改变 Quick Tag Editor 的模式。即,利用键盘快捷键可以使 Quick Tag Editor 在插入 HTML 模式、编辑标记模式和封装标记模式之间来回切换。

5.2.1 插入 HTML 模式

当你在没有选中任何对象时打开 Quick Tag Editor,此时它是处于插入 HTML 模式(Insert HTML mode),它仅显示一对尖括号(“<”和“>”),并且在这对尖括号内部有一个插入点,如图所示 5-7。在此模式下,我们可以输入任意的 HTML 字符串,包括多个标记和文本,或者其他位于标记之间的内容。当关闭此编辑器之后,已经输入的 HTML 源码将插入到插入点处。如果由于疏忽大意,我们忘记输入结束标记,Quick Tag Editor 会自动为我们加上相应的结束标记。

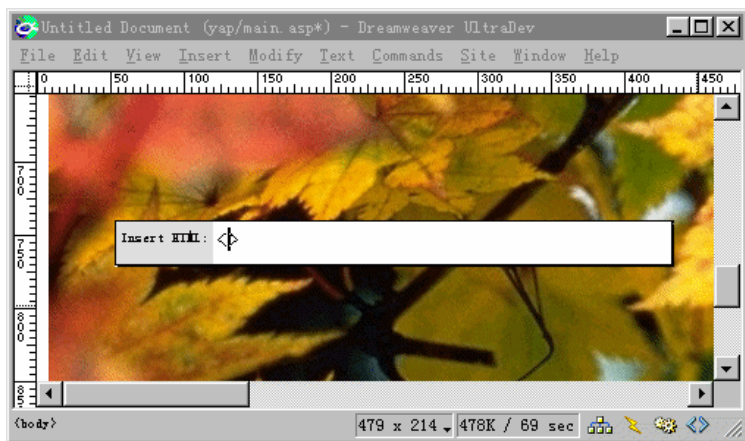


图 5-7
插入
HTML 模
式

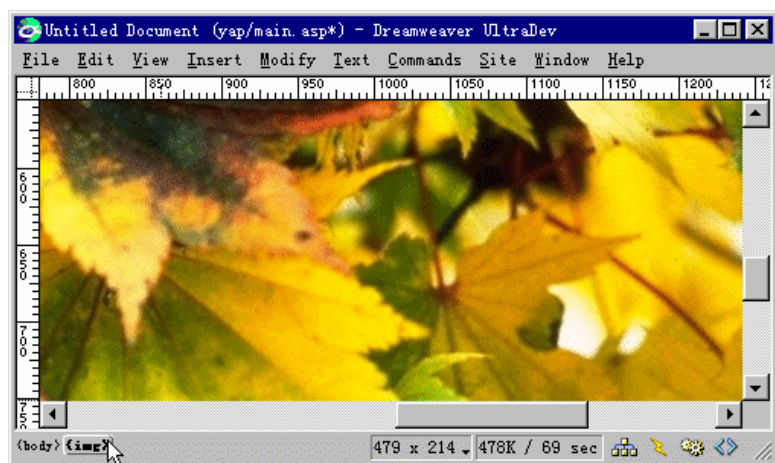
5.2.2 编辑 Tag 模式

如果当前选定包含一个完整标记,那么此时打开 Quick Tag Editor,它会以编辑标记模式(Edit Tag mode)出现,如图 5.4 所示。例如,如果选定的是一个图片,或者如果选定的是一个 FONT 标记和相应的 /FONT 标记,以及两个标记之间的所有文本,那么 Quick Tag Editor 以 Edit Tag 模式启动。确保当前选定开始并结束于相匹配标记的最简单方法是:选定 Document 窗口左下方列出的标记选定符中的需要标记。图 5-8 所示的

Document 窗口左下方只列出了两个标记：<body>和，这样，如果我们要选定图片，就可以直接单击标记，这样我们就已经选定了包含完整标记的图片对象。

图 5-8

Document
窗口左下
方的标记
列



当当前选定包含一个未匹配的开始或关闭标记时，Quick Tag Editor 也以 Edit Tag 模式打开。在这种情况下，当前选定已经扩展到包含它的父标记以及它自身的内容。例如，如果当前段落由 HTML 源码：

```
<p>秋天的<b>枫叶</b>展示秋的韵味。</p>
```

组成，并且当前选定为：

```
的<b>枫叶
```

没有包括结束标记：，那么 Quick Tag 编辑器将以 Edit Tag 模式启动，并将整个段落当作当前选定内容。

在 Edit Tag 模式中，我们仅能编辑一个开始标记。如果我们将插入点移到与开始标记对应的结束标记之后的范围，并输入另一个标记，那么关闭 Quick Tag 编辑器时，DW UltraDev 将弹出一个错误信息对话框，如图 5-6 所示，提示输入的是无效 HTML 代码，并且 DW UltraDev 忽略这些键入的无效 HTML 代码。

5.2.3 封装 Tag 模式

当当前选定不仅仅是由一个完整标记组成时，Quick Tag Editor 通常以封装标记模式（Wrap Tag mode）启动。例如，如果当前选定是没有特殊格式的文本，那么 Quick Tag 编辑器就以 Wrap Tag 模式启动。

但是存在一个例外情况，当当前选定包含一个不匹配的开始或结束标记时，Quick Tag 编辑器将以 Edit Tag 模式启动，而不是 Wrap Tag 模式，并且当前选定已经扩展到包

含它的父标记以及它自身的内容。

当关闭 Quick Tag 编辑器时,已经输入的标记将插入到当前选定的开始处,并且 DW UltraDev 会自动为你在当前选定内容之后添加一个匹配的结束标记。

下面介绍一个小技巧,目的是从文档中移走一个不需要的标记,并且如果该标记包含有内容的话,在原位置保留其内容。

实现方法有以下两种:

- ✱ 选定 Document 窗口左下方列出的标记选定符中的标记,然后从上下文菜单中选择 Remove Tag 菜单项。
 - ✱ 选定 Document 窗口中任意的内容,或单击 Document 窗口以便将插入点放置在此窗口中,然后从上下文菜单中选择 Remove Tag 菜单项。
- 要注意的是,并不是每一种标记都可以被移走。

5.2.4 设置 Quick Tag 编辑器的属性

Quick Tag Editor 属性配置为我们程序员提供了许多方便的性能,我们可以配置在 Quick Tag Editor 窗口中所作的修改是否自动刷新相应的 Document 窗口里的内容。另外,我们也可以配置下拉提示菜单的显示与否。

打开 Quick Tag Editor 属性面板的方法是:选择“ Edit / Preferences ”菜单,在类别中选择 Quick Tag Editor 属性即可。图 5-5 就是 Quick Tag Editor 属性面板。

Apply Changes Immediately While Editing

此选项用于控制在 Edit Tag 模式下,当我们在各属性之间来回移动时,在 Quick Tag Editor 面板中所作的改变是否自动更新 Document 窗口相应内容。如果关闭此选项,我们必须按 Enter 键以便对 Document 应用所作的修改。

此选项在 Insert HTML 模式和 Wrap Tag 下是被忽略不计的,在这两种模式下,我们必须按 Enter 键以便对 Document 应用所作的修改。

Enable Tag Hints

此选项控制当我们在 Quick Tag Editor 中输入源码时,DW UltraDev 是否为我们显示提示列表。我们可以调整提示菜单显示之前停留的时间长度。

5.3 设置 HTML 格式特性

在 DW UltraDev 中,我们可以为格式化 HTML 源代码设置以下属性。

- ※ **HTML 重写属性：**
此属性决定修改的内容，如果存在，那么当打开一个 HTML 文档时，DW UltraDev 会将所作的修改应用到 HTML 源代码。请参见 5.3.1。
- ※ **HTML 格式化属性：**
此属性决定通用 HTML 格式化选项，例如，代码行的长度和缩进量。这些属性为我们提供了修改 SourceFormat.txt 文件内容的一个用户界面。详细内容请参见 HTML 格式化属性一节。
- ※ **HTML 颜色属性：**
此属性控制在 HTML 源码检查器中 HTML 标记显示的颜色。具体内容在 HTML 颜色属性一节介绍。
- ※ **字体/编码属性：**
此属性允许我们程序员指定在 HTML 源码检查器中 HTML 源码显示的字体。
总而言之，SourceFormat.txt 文件为我们提供了精确的格式化 HTML 规范。该文件位于 DW UltraDev 4 安装目录下的 Configuration 文件夹中。对高手来说，可以在文本编辑器中编辑此文件，以便更好地控制 DW UltraDev 4 编写 HTML 源码的方式。另外，我们可以改变此文件中的某些特定标记选项，而这些选项利用 HTML 格式化属性面板是不能改变的。但是，在这里仅建议 DW UltraDev 4 高手可以对此文件进行编辑，因为此文件很重要，稍一不小心就会引起 DW UltraDev 4 程序出现严重的错误。

5.3.1 HTML 重写属性

HTML 重写(HTML Rewriting)属性决定打开 HTML 文档时 DW UltraDev 4 需要处理的事情。这些属性对你在 HTML 源码编辑器中编辑 HTML 源代码时没有任何影响。如果你关闭了这些重写选项，那么当 HTML 源码将被重写时，DW UltraDev 4 会在 Document 窗口显示无效标记项。如果你需要从 Word 文档中引入一个 HTML 文档，可以利用 Clean Up Word HTML 命令去掉任意不需要的 HTML 代码。

利用菜单“Edit / Preference”可以打开属性面板，选中 HTML Rewriting 类别后，就可以看到 HTML Rewriting 面板，如图 5-9 所示。

Fix Invalidly Nested and Unclosed Tags

此选项重写无效嵌套或重叠标记。例如，代码：`<b><i>text</b></i>`将被重写为代码：`<b><i>text</i></b>`。另外，如果缺少结束引号标记或结束括号标志时，此选项也会为我们自动插入相应的结束符号标志。

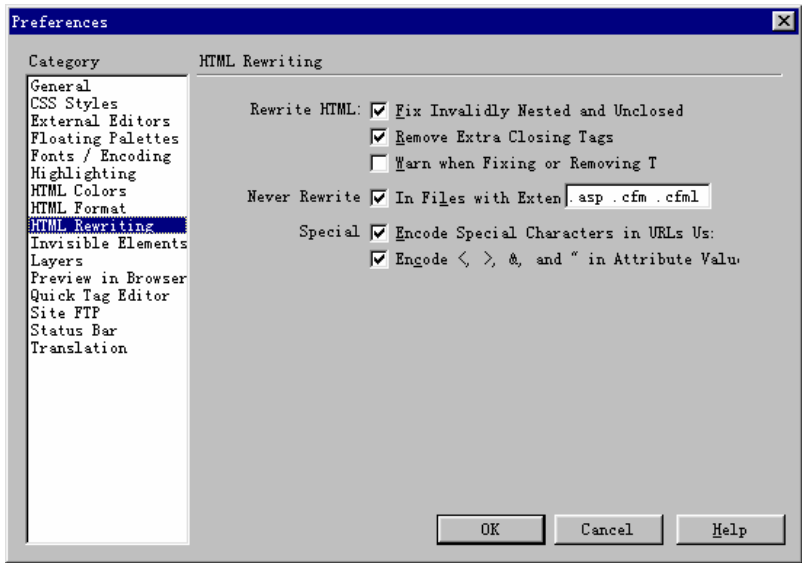


图 5-9
HTML
Rewriting
面板

Remove Extra Closing Tags

此选项删除无对应开始标记的结束标记。

Warn when Fixing or Removing Tags

此选项将显示一个 DW UltraDev 试图纠正的、技术上无效的 HTML 源码的摘要信息。此摘要信息标注了问题所在的位置(方法是列出问题所在的行号和列号), 这样我们可以找到快速更正错误。

Never Rewrite HTML:In Files with Extensions

此选项允许我们指定某些文件扩展名,禁止 DW UltraDev 对这些文件进行重写操作。此选项对包含第三方标记(例如 ASP 标记) 的文件特别有用。

Special

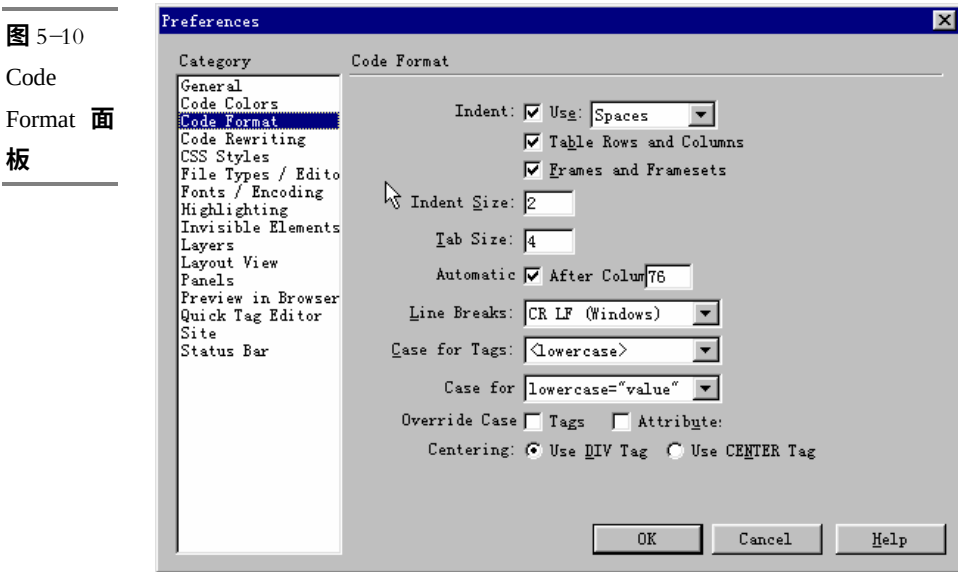
此选项包括两个特殊字符选项, 它们允许我们控制 DW UltraDev 是否在特定上下文中编码特定的字符。一般来说, 最好选中这两个选项, 除非你的文件包含特定第三方标记, 而这些标记不支持这些特定字符。

5.3.2 HTML 格式化属性

利用下列代码格式化(Code Format) 属性可以控制 HTML 源码格式, 例如, 当你

创建文档时，代码的缩进、代码行的长度，以及标记和属性名称的大小写格式。注意，以下的选项（除了 Override Case Of 选项）仅影响新文档和已有文档新增代码（在 Document 窗口内）。也就是说，当你打开一个以前创建的 HTML 文档，这些格式化选项对它们并不起作用。如果你需要重新格式化已存在的 HTML 文档，那么请利用 Apply Source Formatting 命令。

Code Format 面板如图 5-10 所示：



这些选项仅适用于在 Document 窗口中所作的修改。如果你直接在 Code Inspector 检查器中编辑 HTML 代码，那么这些选项并不发挥作用。

Indent

Indent 选项打开 SourceFormat.txt 文件中所有标注为 INDENT 的标记的缩进控制。

- Use：此选项指定是否利用 spaces 或 tabs 缩进。
- Table Rows and Columns：此选项自动缩进 TR 和 TD 标记，以便表格代码更容易阅读。必须设置了上面的 Use 选项，此选项才会处于使能状态，我们才能设置它。
- Frames and Framesets：此选项自动缩进 FRAME 和嵌套的 FRAMESET 标记，以便框架集文件更方便阅读。同上面的 Table Rows and Columns 选项一样，只有设置了 Use 选项，此选项才能设置。

Indent Size

此选项决定缩进量的大小，度量方法是：如果 Use 选项设置为 spaces，那么缩进量是指 space 数目；如果 Use 选项设置为 tabs，那么缩进量是指 tab 数目。例如，如果 Use 选项设置为 spaces，并且 Indent Size 设置为 2，那么对应的标记将缩进两个空格位。

Tab Size

此选项决定 tab 的大小，度量单位为字符空间。



如果 Indent Size 值不是 Tab Size 值的倍数，并且 Use 选项设置的是 tabs，那么标记的缩进量是 tabs 和 space 字符的组合。

Automatic After Column

当代码行一旦达到指定列宽度，那么此选项将绕排行，并给行附加一个硬回车符。与此相对应的是，在 Code Inspector 检查器中的 Wrap 检查框选项也能使超过窗口宽度的行绕排，但它只是为该行附加一个软回车符。



DW UltraDev 仅在某些特定位置插入硬回车符，这些特定位置是指在浏览器中显示时，不会改变文档外观的位置。

Line Breaks

此选项指定你远程站点所在的远程服务器的类型（Windows、Macintosh 或 UNIX）。选定正确的换行符将确保你的 HTML 源代码在远程服务器上能正确显示。



对 FTP，此选项设置仅适用于二进制传输模式；DW UltraDev 的 ASCII 传输模式忽略此选项设置。如果你利用 ASCII 模式下载文件，DW UltraDev 将基于你的计算机操作系统设置换行符；如果你利用 ASCII 模式上传文件，那么所有的换行符都设置为 CR LF。

Case for Tags 和 Case for Attributes

这两个选项控制标记和属性名称的大小写状态。但是它们仅适用于你在 Document 窗口中插入或编辑的标记和属性名称，对 Code Inspector 检查器中直接输入的标记和属性名称不起作用，也不能控制打开文档时已经存在的标记和属性的大小写状态（除非你

选定了一个或全部的 Override Case Of 选项)。

Override Case Of:Tags 和 Attributes

此选项指定是否任何时候都强迫应用指定的大小写选项，包括打开一个已经存在的 HTML 文档情况。当你选定这两个选项之一时，打开文档中的所有标记或属性都会立即转换为指定的大小写状态。你在 Code Inspector 检查器和 Quick Tag Editor 中输入的标记或属性都会转换为指定的大小写状态，就像利用对象面板插入标记和属性一样。例如，如果你希望标记名称总是以小写状态显示，那么将 Case for Tags 选项设置为小写状态，然后选定 Override Case Of:Tags 选项。这时，如果你打开一个包含大写标记名称的文档，DW UltraDev 将把这些大写标记全部转换为小写状态。

Centering

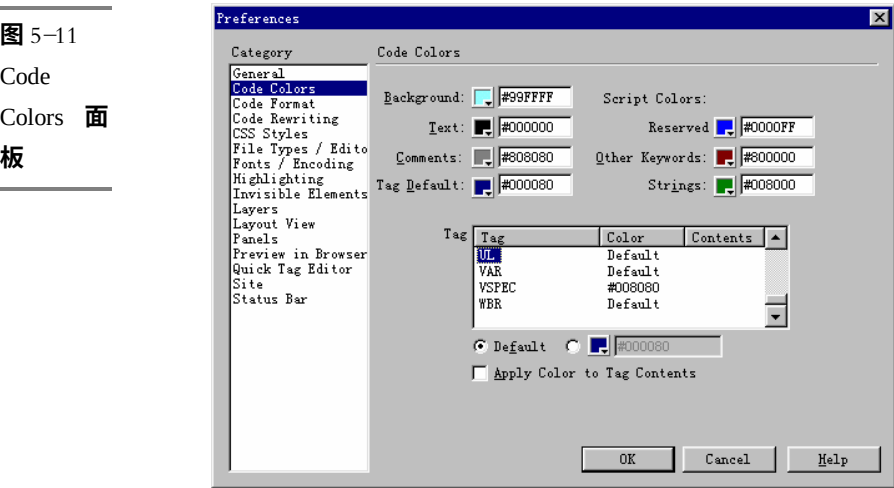
此选项指定元素是否利用 DIV ALIGN="center"或 CENTER 标记居中显示。

5.3.3 HTML 颜色属性

利用 HTML 颜色属性(Code Colors) 可以控制 Code Inspector 检查器的背景、文本和标记的颜色。Code Colors 面板如图 5-11 所示。

Background

此选项指定 Code Inspector 检查器的背景颜色。指定的颜色仅当你单击 Code Inspector 检查器或在检查器中输入代码时才会显示。当当前控制焦点不在 Code Inspector 检查器时，此检查器的背景颜色为中灰色。



Text

此属性指定出现在标记之间的文本颜色。例如，代码：`<b>插入的文本</b>`，文本“插入的文本”将以 Text 设置的顏色显示，但是此颜色属性并不适用于标记。

Comments

此选项指定注释标记（`<!-- -->`）的颜色，包括注释标记之间的内容。

Tag Default

此选项设置所有标记（注释标记除外）的颜色。我们可以利用此选项设置指定标记的颜色，从而覆盖标记的缺省颜色。

Tag Specific

此选项允许我们设置指定标记的颜色，以覆盖 Tag Default 和 Text 颜色设置。

下面介绍一下为指定标记设置颜色的方法：

（1）在 Tag Specific 列表框中选定需要设置颜色的标记。

可以利用 Shift 键同时选中多个相邻的标记，或者利用 Control 键（Windows 平台）或 Command 键（Macintosh 平台）选择多个非相邻标记。

（2）选中位于 Code Colors 面板底部的颜色样本  旁边的单选钮，然后利用以下方法之一选择一个新的颜色值：

- ※ 在紧邻颜色样本的文本域中输入十六进制的颜色值。
- ※ 单击颜色样本，然后从弹出的颜色选择面板中选择需要的颜色。

（3）为了将设置的标记颜色适用于开始标记和结束标记之间的文本，请选中 Apply Color to Tag Contents 选项。

5.4 净化 HTML 源代码

5.4.1 净化 HTML 源代码

我们可以利用 Clean Up HTML 命令移去空标记、组合嵌套的 FONT 标记，以及改善凌乱的或难以理解的 HTML 代码。

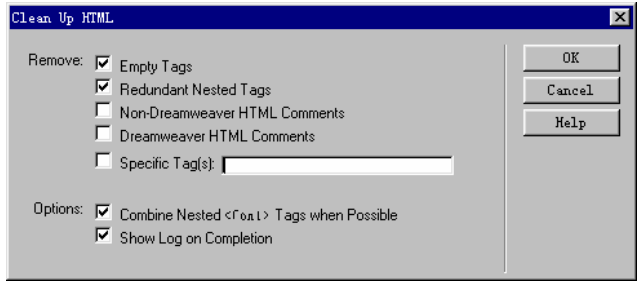
1. 净化 HTML 代码

打开一个已经存在的文档，然后选择“Commands / Clean Up HTML”菜单项，打开

Clear Up HTML 对话框，如图 5-12 所示。

图 5-12

Clean Up
HTML 对
话框



2. 根据需要选中对话框中的各种选项

Remove Empty Tags

此选项移走那些没有内容的标记。例如，代码：`<b></b>` 和 `<FONT COLOR="FF0000"></FONT>` 都被当作空标记，但是代码：`<b>`插入的文本`</b>`就不是空标记。

Remove Redundant Nested Tags

此选项移走一个标记的所有冗余实例。例如，代码：`<b>DW UltraDev 真是制作<b>网页</b>的利器</b>`，其中包围“网页”单词的 B 标记就是一个冗余标记，将被移走。

Remove Non-Dreamweaver HTML Comments

此选项将移走所有并非由 DW UltraDev 插入的注释。例如，`<!-- 文本开始 -->`代码将被移走，但是`<!-- #BeginEditable "doctitle" -->`却不会被移走，因为后者是一个 DW UltraDev 注释，标志模板中一个可编辑区域的开始处。

Remove Dreamweaver HTML Comments

此选项移走所有由 DW UltraDev 插入的注释。例如，代码`<!-- #BeginEditable "doctitle" -->`就会被移走，但是代码`<!-- 文本开始 -->`此时就不会被移走了，因为它不是由 DW UltraDev 创建的注释。移走 DW UltraDev 创建的注释将使基于模板的文档转换为普通的 HTML 文档，而库的各项转换为正常的 HTML 代码，也就是说，无论何时原始的模板或库的各项发生变化，已经移走 DW UltraDev 注释的内容都不会被刷新。

Remove Specific Tag(s)

此选项移走编辑框中指定的标记。利用此选项移走其他可视编辑器插入的定制标记，以及你不希望在你的站点上显示的那些标记（例如，BLINK）。如果需要指定多个

需要移走的标记，各标记之间用逗号分隔，例如，“ FONT,BLINK ”。

Combine Nested Tags When Possible

当两个或更多的 FONT 标记控制的是同一段文本时，利用此选项可以将它们合并为一个 FONT 标记。例如：

```
<FONT SIZE="7"><FONT COLOR="#FF0000">big red</FONT></FONT>
```

如果选中此选项，那么上面这段代码将变为：

```
<FONT SIZE="7" COLOR="#FF0000">big red</FONT>
```

Show Log on Completion

当净化操作完成时，此选项将显示一个带有文本变化详细信息的警告对话框。

3. 单击 OK 按钮

当确定你已经选择完毕正确的选项设置后，只有单击 OK 按钮才能使选中的选项发挥作用并关闭 Clear Up HTML 对话框。如果设置了 Show Log on Completion 选项，那么当净化操作完成后，DW UltraDev 将会显示一个净化信息对话框，如图 5-13 所示。

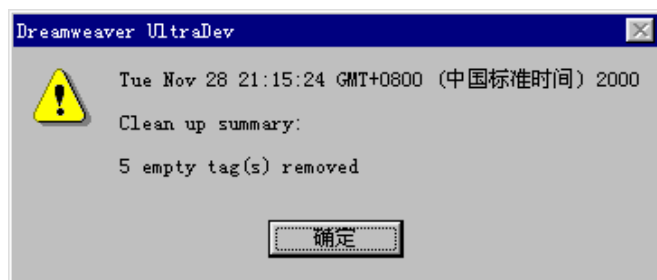


图 5-13
警告信息
对话框

5.4.2 净化 Microsoft word HTML 源代码

DW UltraDev 4 可以打开或导入由 Microsoft Word 保存的 HTML 文档文件,然后我们可以利用 Clean Up Word HTML 对话框进行设置，以移走由 Word 产生的无关的 HTML 代码。这些被移走的代码以前由 Word 用来格式化和显示文档用的，显示 HTML 文件时并不需要它们。建议读者为初始 Word 文件(.doc 文件) 做个备份,因为一旦你对 HTML 文档应用了 Clean Up Word HTML 特性，那么 Word 就有可能不能打开这个 HTML 文档了。Clean Up Word HTML 命令适用于 Word97 以及其以上版本保存的 HTML 文档。

下面介绍一下如何使用 Clean Up Word HTML 特性。

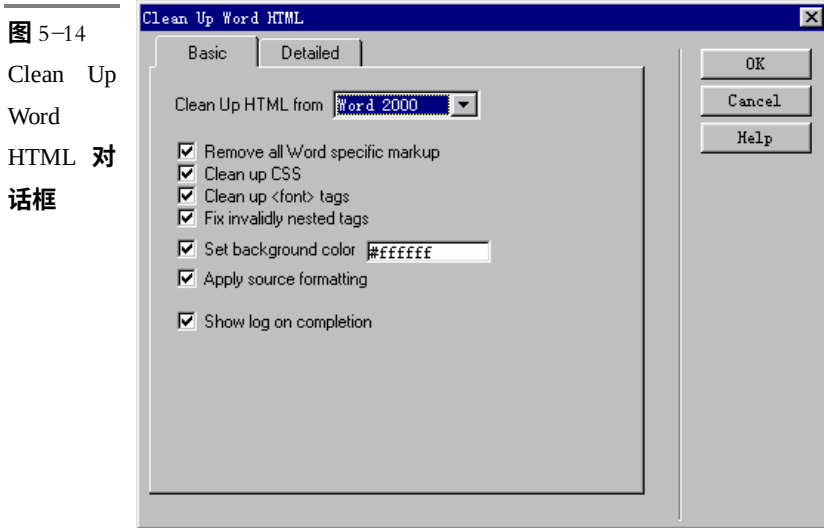
- (1) 在 Microsoft Word 中将你的文档保存为一个 HTML 文件。

(2) 利用下列方法之一在 DW UltraDev 中打开此 HTML 文档：

※ 选择 File / Import / Import Word HTML ”菜单项，并且选择需要打开的文件。此时，DW UltraDev 打开此文件，然后自动打开 Clean Up Word HTML 对话框。

※ 选择‘ File / Open ”菜单项，并且选择需要打开的文件。

(3) 利用以上这两种方法打开 Clean Up Word HTML 对话框时，我们都会发现有一个小的延时，此时 DW UltraDev 正在判断保存此 HTML 文档使用的 Word 版本号。如果 DW UltraDev 不能决定 Word 版本号，程序员可以利用弹出菜单选择正确的版本号。打开的对话框如图 5-14 所示。



(4) 根据需要设置对话框中的选项。其中 Basic 面板提供了以下几种选项：

- ※ Remove all Word Specific Markup
此选项移走所有 Word 特有的 HTML 标记，包括 XML、Word 定制的元数据和文档头部的链接标记、Word XML 标记、空段落。我们可以利用 Detailed 面板为这些选项逐个进行设置。
- ※ Clean up CSS
此选项移走所有 Word 特有的 CSS 风格，包括嵌入 CSS 风格、以‘ mso ’开头的风格属性、非 CSS 风格声明、表格的 CSS 风格属性。这些选项可以通过 Detailed 面板进一步定制。
- ※ Clean up Tags
此选项移走 HTML 标记、将缺省 Body 文本转换为字体大小为 2 的 HTML 代码。
- ※ Fix Invalidly Nested Tags
此选项从段落和头标记中移走 Word 插入的字体标记。

- ✧ Set Background Color
此选项允许我们输入十六进制表示的颜色值作为文档的背景色。如果我们没有为文档设置背景色，那么 Word HTML 文档将是一个灰色的背景。DW UltraDev 设置的缺省十六进制背景色是白色。
- ✧ Apply Source Formatting
将 HTML Format 性能面板以及 Source Format 配置文件中指定的源格式选项应用到文档上。
- ✧ Show Log on Completion
当净化操作完成时，此选项将显示一个带有文本变化详细信息的警告对话框。
(5) 单击 OK 按钮，使选中的选项发挥作用并关闭 Clean Up Word HTML 对话框。

5.5 外部 HTML 编辑器

我们可以通过外部文本编辑器利用 HTML、JavaScript 或 VBScript 进行手工编制网页代码。DW UltraDev 支持任意的外部文本编辑器，包括 Notepad (Windows 平台) 或 SimpleText (Macintosh 平台)、BBEdit、HomeSite、vi、emacs，以及 TextPad。

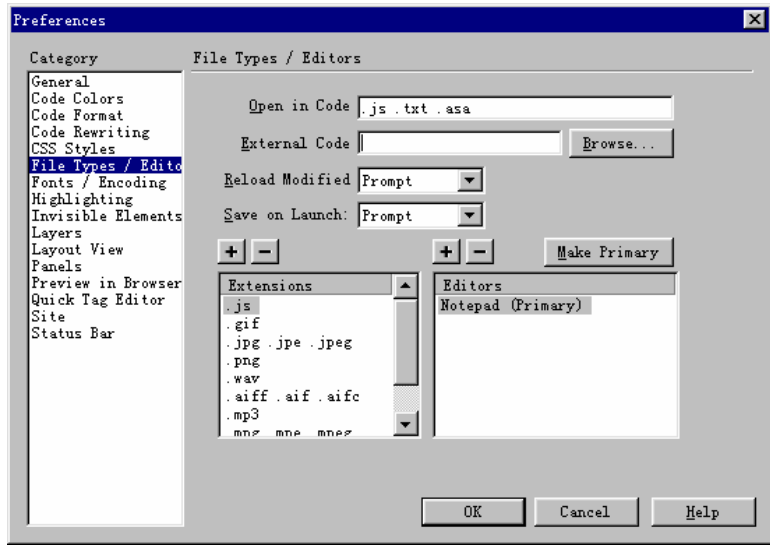
我们可以从 DW UltraDev 中启动它的第一外部文本编辑器为当前文档编辑源 HTML 代码，然后切换回 DW UltraDev 继续在图形方式下编辑。DW UltraDev 可以检测到已保存到文本中的由外部编辑器所作的任意变化，如果存在变化，它会提示你是否重新装载此文档。

选择外部编辑器

- 1 选择“Edit / Preferences”菜单项。
- 2 从 Category 列表中选择 File Types/Editors 类别，将显示 File Types/Editors 面板，如图 5-15 所示。
- 3 这一步仅适用于 Macintosh 平台) 如果希望使用 HTML 编辑器，而不是 BBEEdit，请去掉 Enable BBEEdit 集成选项；如果希望使用 BBEEdit，请选中 Enable BBEEdit 集成选项，并跳过以下步骤。(图 5-15 是 Windows 平台上的 External Editors 面板，所以没有 Enable BBEEdit 集成选项。)
- 4 单击 Browse 按钮，选择文本编辑器。
- 5 在 Reload Modified Files 选项中，指定你希望 DW UltraDev 检测到外部编辑器对文档进行了修改之后的操作。
- 6 在 Save on Launch 选项，指定 DW UltraDev 启动外部编辑器之前是否保存当前文档，选项有三个：保存、从不保存、为你显示提示信息，询问你是否保存。

图 5-15

File
Types/Edit-
ors 面板



启动外部编辑器

选择“ Edit / Edit with External Editor ”菜单项，就可以启动外部编辑器。

5.6 其他编辑功能

5.6.1 纠正无效标记

如果你在 Document 窗口或 Code Inspector 检查器中看到以亮黄色显示的 HTML 代码，这说明 DW UltraDev 发现无法显示的无效 HTML 代码。例如，当你在 Code Inspector 检查器中开始输入一个标记，然后没有完成此标记的编制，就切换到 Document 窗口，此时就会出现无效 HTML 代码。

为了纠正错误，单击黄色显示的 HTML 代码，并根据 Property 检查器中的建议修改。

5.6.2 插入脚本

我们可以不通过 Code Inspector 检查器就可以在 Document 窗口中输入 JavaScript 和 VBScript 代码。

插入一个脚本

- 1 将光标放置在需要插入的位置。
- 2 单击 Invisibles 对象面板上的 Script 按钮，或者选择“ Insert / Invisible Tags / Script ”菜单项，将打开如图 5-16 所示的 Insert Script 对话框。

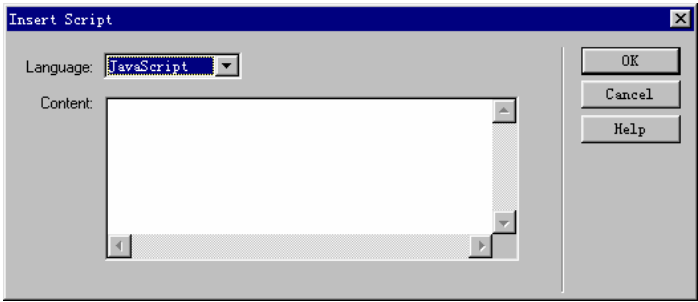


图 5-16
Insert
Script 对
话框

- 3 从对话框的弹出式菜单选择合适的脚本语言。
- 4 输入将出现在脚本标记之间的代码。
如果是为了链接到一个外部脚本文件，不需要输入任何代码，单击 OK 按钮，然后在 Property 检查器中加入链接源文件的引用，方法是在图 5-17 中的 Source 编辑框中输入链接源文件名，或利用它右边的浏览按钮，选择一个需要的链接源文件。

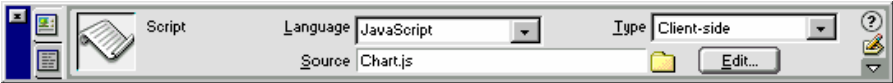


图 5-17
加入外部
脚本文件
的连接

编辑脚本

- (1) 选择 Script 图标。
- (2) 单击 Property 检查器中的 Edit 按钮，就可以开始进行对脚本的编辑。

5.6.3 插入注释

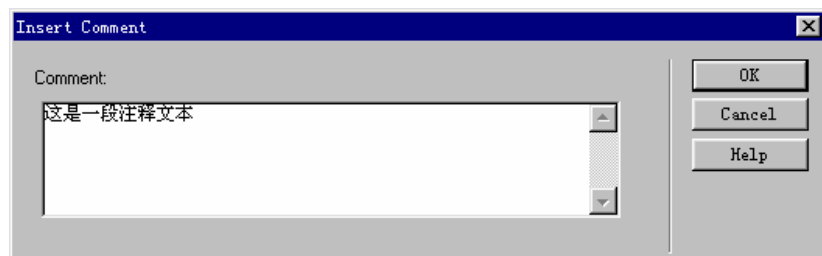
注释是插入到 HTML 代码中的一段描述性文本，用于解释代码的含义，或提供其他一些有用的信息。注释并不影响文档在浏览器中显示时的外观。如果你需要为站点中的任意文件增加更多的信息，也可以使用 Design Notes。

插入一个注释

- (1) 将插入点放置到需要插入注释的位置。
- (2) 单击对象面板中的 Invisibles 面板上的 Comment 按钮,或者选择 Insert / Invisible / Comment”菜单项,就可以输入注释文本了,如图 5-18 所示。

图 5-18

Insert
Comment
对话框



为了在 Document 窗口中显示注释标记,请选中“ View / Visual Aids / Invisible Elements”控制选项。

输入注释文本

- (1) 选择 Comments 图标。
- (2) 在 Property 检查器中输入注释文本,如图 5-19 所示。

图 5-19

Property
检查器中
输入注释
文本



创建一个注释将产生下列 HTML 代码：

```
<!--这是一段注释文本-->
```

第 6 章 UltraDev 开发 Web 应用基础

本章概要：

- ☒ Web 应用开发流程
- ☒ 开发前的准备工作，包括定义站点、配置 DW UltraDev、创建数据库链接等

从本章开始我们将进入 Web 应用开发部分，首先让我们来看一下使用 DW UltraDev 4 开发 Web 应用之前需要做哪些工作。

6.1 Web 应用开发流程

使用 DW UltraDev，Web 开发人员可以非常轻松地在站点中加入基于服务器端的功能，如数据库链接、显示服务器端内容、增加服务器端脚本以及预览动态内容等。DW UltraDev 中包含了开发 Web 应用所需要的所有特性，使用者可以很容易加入服务器端逻辑来浏览、插入、更新和删除数据库记录，显示数据库中的一条记录就像拖动鼠标一样简单，而所有的服务器代码都是自动产生的。

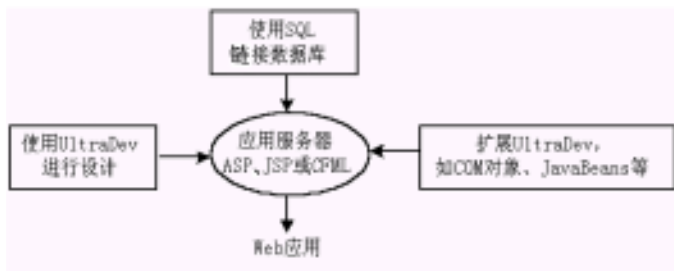
DW UltraDev 使得 Web 应用的开发过程变得流程化。一般来说，使用 DW UltraDev 开发一个 Web 应用可分为以下五个步骤：

- 1 设计用户界面，其中包含了一个或多个页面。这一步可以使用 DW UltraDev 的页面设计工具来完成。在这个阶段，该 Web 应用将不会显示任何数据，或者执行任何任务。
- 2 创建一个记录集。记录集是指来自一个数据库中的数据所组成的子集合。它通常临时存放在服务器的内存中，以便使 Web 应用能够更方便快捷地访问数据库，从而使执行速度更快一些、效率更高一些。
- 3 在页面中加入动态数据或内容。这是通过数据绑定(Data Bindings) 面板来完成。
- 4 在页面中加入服务器行为(Server Behaviors)，以便使其正常工作。服务器行为实际上是指一组指令，它们是在运行时刻由服务器来执行的。

5 对 Web 应用进行编辑和调试。例如，可以使用 Live Data（实时数据）窗口对动态页面的布局进行微调。一旦对页面的布局满意了，并且 Web 应用工作正常了，就可以将其上传到 Web 服务器上了。

使用 DW UltraDev 创建的 Web 应用工作示意图如图 6-1 所示。从图中可以看出，使用 DW UltraDev 创建 Web 应用时具有很强的灵活性。用户可以使用 JavaScript、Java 或 C 语言等来方便地扩展 DW UltraDev 以满足自己的需求，或者使用 DW UltraDev 中丰富的 API 来增加 COM 对象、JavaBeans、ColdFusion 标记库等。

图 6-1
Web 应用
工作示意
图



6.2 开发前的准备工作

在开始开发 Web 应用之前，需要先做一些必要的准备工作。例如，必须在 DW UltraDev 中定义好站点，并将文件组织起来；必须配置 DW UltraDev，以便它能够知道使用哪一种服务器技术（ASP、JSP 或者 ColdFusion），从而能够在页面中插入正确的服务器端脚本和标记。另外，还必须告诉 DW UltraDev 使用哪一个 Web 服务器来实现在 Live Data 窗口中进行实时数据预览的功能；最后，如果我们的 Web 应用中使用了数据库，还要创建一个数据库链接，以便 Web 应用能够找到该数据库并与之交互。

6.2.1 定义站点

在 DW UltraDev 中，只需点击鼠标就可以实现对文件的管理以及文件在本地磁盘和 Web 服务器之间的传输。为了运用这一功能，必须做如下一些工作：

- ✱ 在本地磁盘上创建一个文件夹，其中存放 Web 应用需要使用的所有文件或子目录。
- ✱ 定义一个本地站点。只有如此，DW UltraDev 才能正常地工作。
- ✱ 定义一个远程站点，即 Web 服务器上存放 Web 应用所使用的文件的目录。

在第 4 章中，我们已经详细介绍了定义本地站点和远程站点的过程，不熟悉的读者不妨回过头去看一下。

6.2.2 配置 DW UltraDev

在定义了站点以后，DW UltraDev 还需要知道以下两个问题的答案才能开始创建 Web 应用：

- ※ Web 应用将使用哪一种服务器技术(ASP、JSP，等等) ？
 - ※ DW UltraDev 要使用哪一个 Web 服务器来实现在 Live Data 窗口中进行实时数据预览的功能？
- 为此，我们需要对 DW UltraDev 进行一些配置。

1. 确定服务器技术

只有确定了服务器技术，DW UltraDev 才能正确地在页面中插入适当的服务器端脚本和标记。例如，如果服务器采用的是基于 ASP 的技术，则 DW UltraDev 就可以往页面中插入 VBScripts 或 JavaScripts 脚本和标记，而不会插入 Java 代码或 ColdFusion 脚本。

我们只能为一个站点定义一种服务器技术，而不能为站点中的一些页面定义一种技术，而为另一些页面定义另一种技术。这样就能保证整个 Web 应用是协调一致的。

下面介绍为一个站点确定服务器技术的方法。

首先，启动 DW UltraDev，然后选择“ Site / Define Sites ”菜单命令打开 Define Sites 窗口，在其中选择已经事先定义好的站点名，然后单击“ Edit ”按钮打开 Site Definition 窗口。单击窗口中 Category 列表中的 Application Server 项，则 Site Definition 窗口变成如图 6-2 所示的样子。

在 Server Model 下拉列表中选择正确的服务器模型，可以选择 ASP、JSP 或 ColdFusion。如果选择的是 ASP，则请在 Default Scripting 下拉列表中选择缺省的脚本语言，可以选择 VBScript 或 JavaScript；如果选择的是 JSP 或 ColdFusion，则系统已经自动设置好了缺省的脚本语言，用户没有选择的余地。

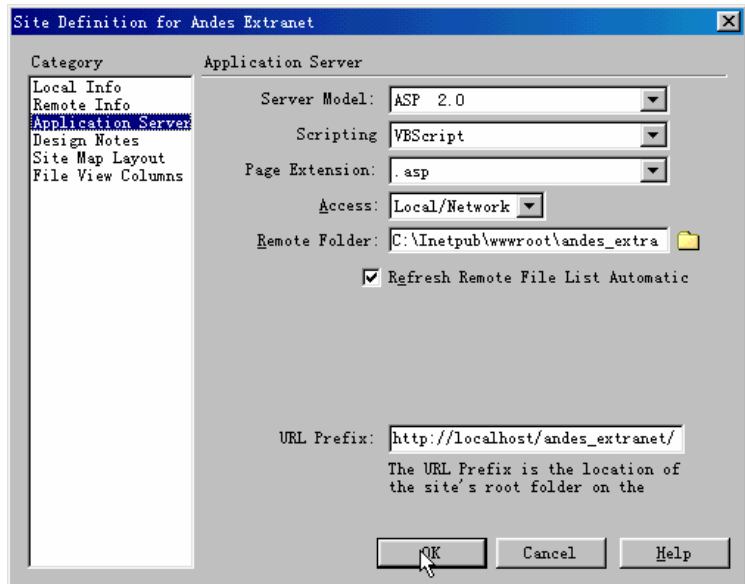
在 Default Page Extension 下拉列表中确定文档窗口中的缺省文件后缀名(如.asp)，系统将自动给所有在文档窗口中创建的文件添加这个后缀。



缺省的.asp、.jsp 和.cfm 后缀名并不会影响静态页面(如后缀名为.htm 和.html 的页面)。但是，如果把缺省的后缀名改为.htm 或.html，则将会使在页面中创建的所有动态功能失效，此时，我们必须将这些页面的后缀名重新修改为.asp、.jsp 或.cfm。

图 6-2

Site
Definition
窗口



保持 Site Definition 窗口，接下来我们继续配置实时数据预览功能。

2. 配置 Live Data 窗口

使用 Live Data 窗口，系统可以使用实际的数据库记录值来代替页面中的占位符，从而实现实时数据预览功能。当然，这一功能必须配合 Web 服务器来完成。当用户把文档窗口切换到 Live Data 窗口时，系统会把当前文档临时复制到特定的 Web 服务器上进行处理，然后 Web 服务器将处理结果返回来送到 Live Data 窗口中显示，同时将存放在 Web 服务器上的临时复制文档删除。

我们返回到刚才的 Site Definition 窗口，继续来配置 Live Data 窗口。

在 Access 下拉列表中，选择 Live Data 窗口使用 Web 服务器的方式，可以是 FTP 或 Local/Network。



请正确地理解这里的 Local(本地)和 Remote(远端)的概念。这时的本地与远端并不是以计算机来区别的，而是以站点文件所存放的位置来区别的。即使 DW UltraDev 和 Web 服务器安装在同一台计算机上，但如果 Web 服务器所在的远程站点目录与 DW UltraDev 所使用的本地站点目录位于磁盘上的不同位置，则我们也应该选择 Remote Web Server；但如果 Web 服务器所在的远程站点目录与 DW UltraDev 所使用的本地站点目录位于磁盘上的相同位置，则应该选择 Local Web Server。

在 URL Prefix 文本框中，输入远程站点的 URL 前缀，即远程站点的根所在的 URL。

最后连续单击 OK 和 Done 按钮，完成对 Live Data 窗口的配置。

6.2.3 创建数据库链接

Web 应用中经常要用到数据库。在创建这样的 Web 应用时，我们至少应该创建一个数据库链接，否则 DW UltraDev 将无法找到数据库并与之链接。

在一个数据库链接中定义了一些参数，它们用来建立与一个数据库的链接关系。如果一个页面要同数据库打交道，则 DW UltraDev 将这些参数放置在页面的服务器端脚本中。

在 DW UltraDev 中可以创建任意数目的数据库链接。系统将这些链接都存储起来，等待 Web 应用调用它们。我们也可以对现有的数据库链接进行删除或编辑操作。

通常 DW UltraDev 将这些数据库链接都保存在 UltraDev 应用程序的 Configuration\Connections 目录下的 connections.xml 文件中，我们可以将此文件复制到其他运行 DW UltraDev 的系统中，以便能够使用这些数据库链接。

1. 创建数据库链接

在使用 ASP 技术时，需要创建一个运行时刻的数据库链接，其创建步骤如下：

在文档窗口或 Live Data 窗口中，选择“Modify / Connections”菜单命令，打开 Connections 窗口，如图 6-3 所示。然后在其中单击 New 按钮，这时在 New 按钮上会出现一个下拉菜单，其中有两个菜单命令：Custom Connection String 和 Data Source Name (DSN)。

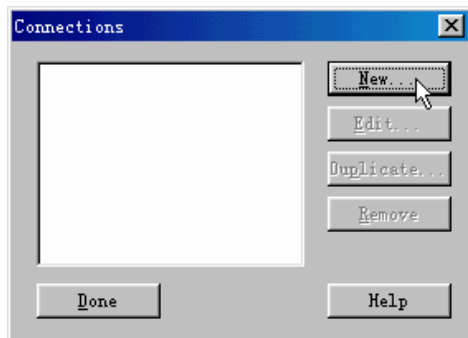
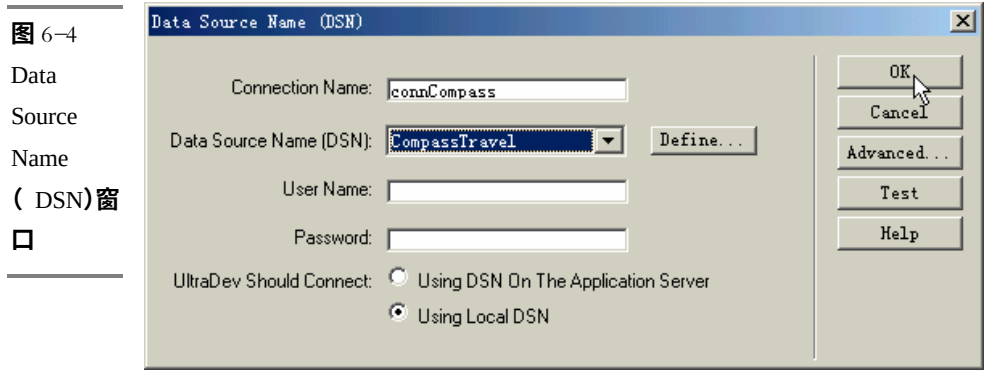


图 6-3
Connections
窗口

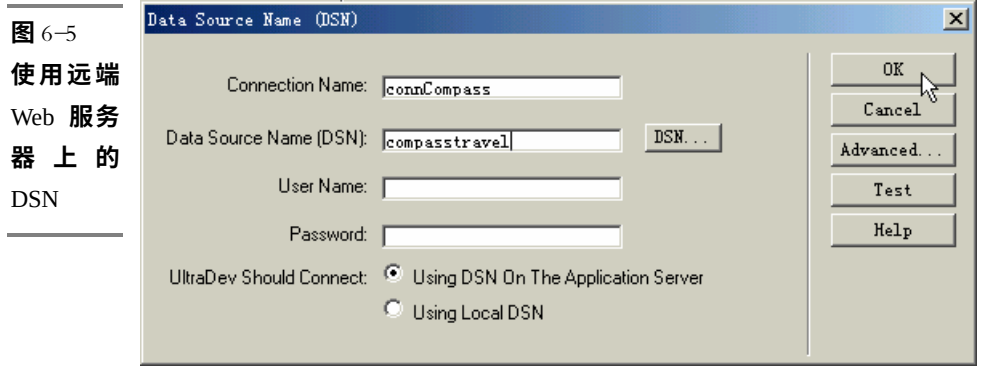
通常 UltraDev 与数据库进行链接的方式有两种，即上述两个菜单命令所决定的：自定义连接字符串 (Custom Connection String) 和数据源名 (DSN, Data Source Name)。如果 Web 服务器和 DW UltraDev 运行在同一台计算机上，则我们可以使用一个数据源名 DSN 来创建与数据库的链接。通常把 DSN 看作是与数据库进行链接的一种快捷方法；如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上，则我们既可以在 Web 服

务器上创建一个 DSN 来实现与数据库的链接,也可以使用一个连接字符串来创建与示例数据库的链接。

请用户根据实际情况选择适当的菜单命令。如果选择了 Data Source Name(DSN),则会如图 6-4 所示的 Data Source Name(DSN) 窗口。



在窗口的 Connection Name 文本框中输入数据库的链接名(通常在一个表示数据库链接的名字前面加上前缀 conn, 以便于记忆)。如果 Web 服务器和 DW UltraDev 运行在同一台计算机上, 则请确保在 UltraDev Should Connect 域中已经选择了 Using Local DSN 单选钮, 然后在 Data Source Name(DSN) 下拉列表中选择已经创建好的 DSN ; 如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上, 则请确保在 UltraDev Should Connect 域中已经选择了 Using DSN On The Application Server 单选钮, 然后在 Data Source Name(DSN) 文本框中输入已经创建好的 DSN 名, 如图 6-5 所示。

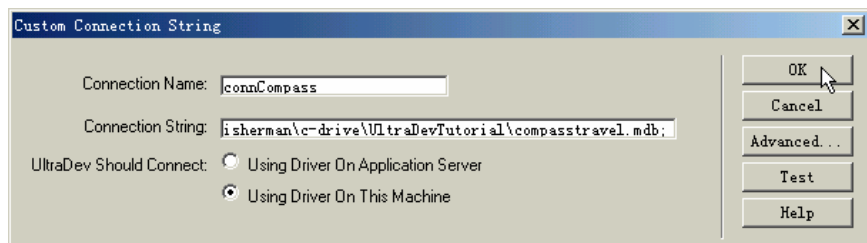


如果需要用户名和口令的话, 请在 User Name 和 Password 文本框中输入正确的用户名和口令。

如果读者在弹出的快捷菜单上选择“ Custom Connection String ”菜单命令, 则出现如图 6-6 所示的 Custom Connection String 窗口。

图 6-6

Custom
Connection
String 窗口



在 Connection Name 文本框中输入数据库链接名，然后在 Connection String 文本框中输入一个链接到位于 Web 服务器上的数据库文件的链接字符串。例如，如果该数据库文件位于一个名为 fisherman 的 Web 服务器的 c:\UltraDevTutorial\目录下，则链接字符串应该如下所示：

```
Provider=MSDASQL;Driver={Microsoft Access Driver( *.mdb) };DBQ= \\fisherman\\c-
drive\\UltraDevTutorial\\compasstravel.mdb;
```

如果 Web 服务器和 DW UltraDev 没有运行在同一台计算机上，则请确保在 UltraDev Should Connect 域中已经选择了 Using DSN On Application Server 单选钮；如果 Web 服务器和 DW UltraDev 运行在同一台计算机上，则请确保在 UltraDev Should Connect 域中已经选择了 Using Driver On This Machine 单选钮，

完成上述设置之后，可以单击 Test 按钮对新建的数据库链接进行测试。如果成功，则会弹出一个对话框，其中显示了“链接创建成功”的消息，单击“确定”按钮关闭该对话框。然后单击 OK 按钮关闭 Define Connection 窗口并返回到 Connections 窗口，此时用户应该发现在 Connections 窗口的列表中新增了一个刚才创建的数据库链接。单击 Done 按钮关闭 Connections 窗口。

2. 编辑或删除数据库链接

编辑数据库链接

仅仅修改数据库链接本身是非常简单的，只需在文档窗口或 Live Data 窗口中，选择“Modify / Connections”菜单命令，打开 Connections 窗口，然后在其中单击 Edit 按钮，打开 Define Connection 窗口，最后在该窗口中对各个域进行适当修改，单击 OK 按钮即可。但是，值得注意的是，在编辑或者删除了数据库链接以后，还需要对所有使用旧连接的页面进行更新。这样，对数据库链接所做的修改才能体现到实际应用中去。

在文档窗口中打开每一个使用了旧的数据库链接创建的记录集的页面，然后打开 Data Binding 面板，如图 6-7 所示。双击面板中的相应记录集名，便会打开 Recordset 窗口，请在该窗口的 Connection 下拉列表中重新选择此记录集所使用的数据库链接名（即修改过的数据库链接），如图 6-8 所示。然后单击 OK 按钮，于是该页面的数据库链接便

会刷新重建。按照同样的方法对所有受影响的页面进行修改。

图 6-7
Data
Bindings
面板

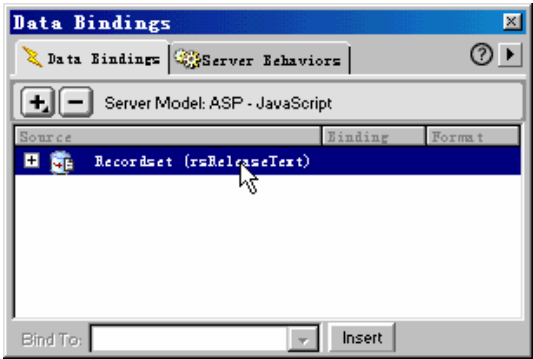
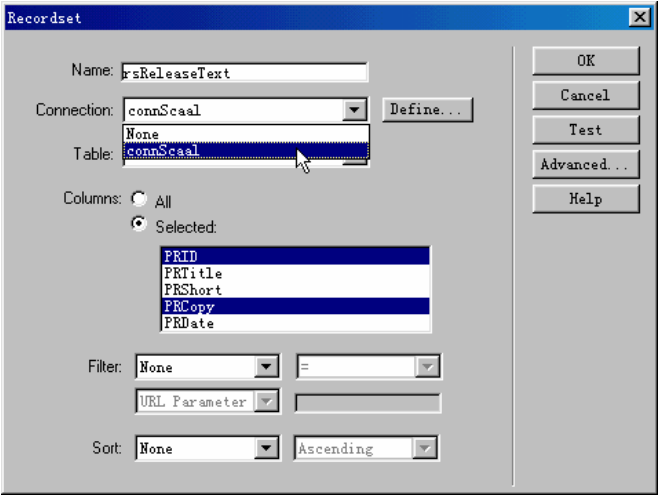


图 6-8
Recordset
窗口



删除数据库链接

删除数据库链接的过程与编辑数据库链接的过程是类似的。首先也要打开 Connections 窗口，然后在窗口的列表中选择需要删除的数据库链接名，然后单击 Remove 按钮，这时会出现一个如图 6-9 所示的警告框，表示删除数据库链接的操作是不可恢复的，单击“是”按钮即可删除它。

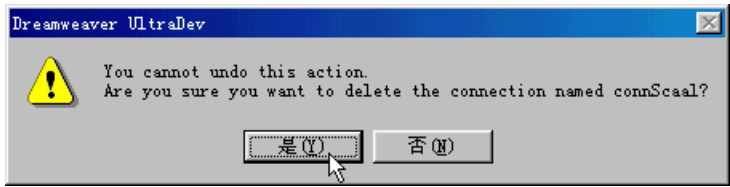


图 6-9
删除数据
库链接的
操作是不
可恢复的

接下来,依次在文档窗口中打开所有受影响的页面,在 Data Bindings 面板选择受影响的记录集名,然后单击  按钮将其删除,或者双击该记录集,打开 Recordset 窗口,在 Connection 下拉列表中重新选择此记录集所使用的数据库链接名(其他的数据库链接),最后单击 OK 按钮确定。

3. 创建 DSN

DSN 是 Data Source Names(数据源名)的缩写。DSN 中包含了创建一个到指定数据库链接所需要的所有信息,它实际上是一个数据库链接的快捷方式。如果要在 DW UltraDev 中使用 DSN,则必须在本地或远程的 Web 服务器端创建一个 DSN。

在 Windows 操作系统中,创建 DSN 需要借助“ODBC 数据源管理器”。通过操作系统的“控制面板”就可以打开“ODBC 数据源管理器”窗口(在 Windows 9x / NT 中,选择 开始 / 设置 / 控制面板”,然后双击 ODBC 数据源”图标即可;在 Windows 2000 中,选择 开始 / 程序 / 管理工具 / 数据源”),如图 6.10 所示。另外,单击 Define Connection 窗口(见图 6.4)中的 ODBC 按钮也可以打开“ODBC 数据源管理器”窗口。

我们可以为各种不同的数据库创建 DSN,如 Microsoft Access、Lotus Approach、Microsoft SQL Server、Oracle8i 等,甚至可以是 Excel Spreadsheets 和文本文件。在创建 DSN 之前,必须要确保系统中已经安装了该数据库的驱动程序,这可以从“ODBC 数据源管理器”的驱动程序面板(如图 6-10 所示)中找到。只要其中列出的 ODBC 驱动程序,我们都可以创建这种数据库的 DSN。

要创建 DSN,打开“ODBC 数据源管理器”窗口后,切换到“系统 DSN”面板,该面板中列出了当前系统中的所有 DSN 名,如图 6-11 所示。

单击“添加”按钮,出现“创建新数据源”窗口,如图 6-12 所示。在其中选择适当的驱动程序来安装数据源,如 Microsoft Access Driver(*.mdb),然后单击“完成”按钮。

图 6-10
ODBC 数
据源管理
器

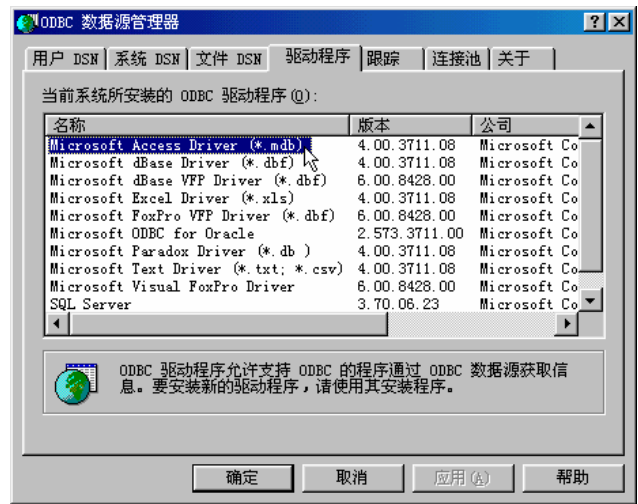


图 6-11
系统 DSN
面板



于是出现相应的 DSN 安装窗口，图 6-13 所示的是 Microsoft Access 驱动程序的安装窗口。在其中输入数据源名，并通过“选取”按钮选择使用的数据库，如果还要进行其他设置，可以单击“高级”和“选项”按钮。设置完成后，单击“确定”按钮。

随即在“系统 DSN”面板中就会列出新创建的 DSN，如图 6-14 所示。



图 6-12
选择安装
DSN 的驱
动程序

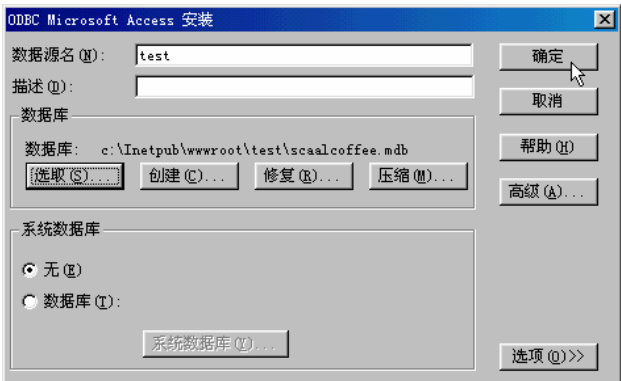


图 6-13
为 Microsoft
Access 创建
DSN 的安装
窗口

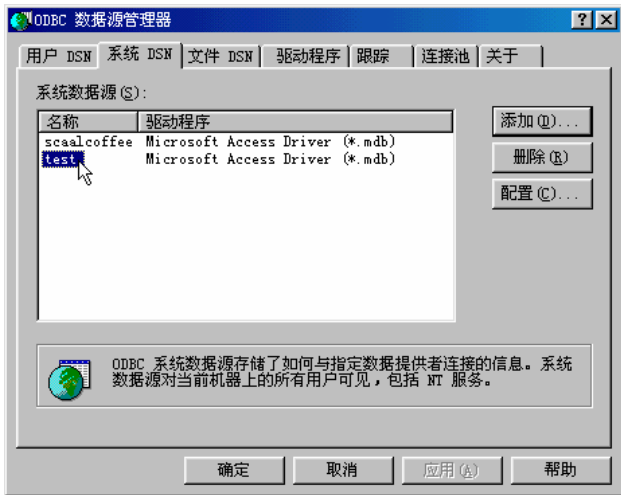


图 6-14
DSN 创建
成功

第 7 章 创建记录集

本章概要：

- ☒ 记录集概述
- ☒ 定义记录集
- ☒ 操作记录集
- ☒ 编写 SQL 查询语句

如果打算在 Web 应用中使用数据库，我们并不能直接同数据库打交道，而必须通过一个记录集来间接同数据库打交道。本章我们就介绍如何在 DW UltraDev 中同记录集打交道。

7.1 记录集概述

一个记录集是指通过一个数据库查询（SQL）语句从数据库中取出的一些特定记录的集合。因此可以把记录集看成是数据库的子集。查询语句中包含了一些特定的查询条件，它们决定了记录集中的记录内容。一个查询通常只能产生一个记录集，而且该记录集中的字段和记录都是固定的。

尽管一个记录集也可以把一个数据库表中的所有记录和字段都包含进来，但是，由于 Web 应用很少会用到一个数据库中的所有数据，因此应该尽量使记录集尽可能得小一些，基本的原则是只包含 Web 应用必须用到的记录和字段。而且，服务器在使用记录集时，通常会把记录集的内容暂存到系统的内存中，当不再需要该记录集时再将它们从内存中释放出来，因此，小的记录集所占用的内存比大的记录集要少，服务器的性能自然就会提高。

7.2 定义记录集

记录集是通过一个查询语句来定义的，在这个查询语句中包含了特定的查询条件，以便从一个数据库中查找和提取信息。DW UltraDev 使用结构化查询语言（Structured Query Language，即 SQL）来创建查询语句。虽然在 DW UltraDev 中定义记录集时并不需要读者了解 SQL，但是，如果读者了解 SQL 的话，就可以通过它来定义更满足自己要求的记录集了。



在定义一个记录集之后，记录集中的数据并不能立即在文档窗口或 Live Data 窗口中显示出来。仅仅当用户在页面中增加了记录集中的字段并切换到 Live Data 窗口后才能看到记录集中的数据。

下面我们就介绍几种定义记录集的情况。

7.2.1 不使用 SQL 定义记录集

如果读者对 SQL 不熟悉，这也不要紧。我们仍然可以通过 DW UltraDev 中的 Recordset 这一简单的对话框窗口来定义自己的记录集。使用 Recordset 窗口来定义记录集是非常简单的，就如同上一章介绍的在创建数据库链接时从下拉列表中选择链接名和数据库表一样。

下面介绍不使用 SQL 定义记录集的步骤。

首先要确保在 DW UltraDev 的文档窗口或 Live Data 窗口中打开了将要使用记录集的页面，然后在 Data Binding 面板中（可单击启动面板上的 Data Bindings 按钮或者选择“Window / Data Bindings”菜单命令打开 Data Bindings 面板）单击  按钮，在出现的菜单中选择 Recordset (Query) 菜单项，如图 7-1 所示。

于是出现如图 7-2 所示的 Recordset 窗口。在 Name 文本框中输入记录集的名称。通常，在给记录集命名时要遵循这样一种规则：在名字前面加上一个“rs”前缀，如 rsPressReleases。这样做的目的是为了将记录集同其他对象区别开来。另外，记住在记录集中不要出现空格或其他特殊字符。

然后在 Connection 下拉列表选择一个数据库链接。如果列表没有任何链接，可以单击 Define 按钮创建一个数据库链接，具体方法请参见第 11 章的内容。于是 Recordset 窗口中各域的值会相应发生变化，如 Table 下拉列表和 Columns 域中都会出现相应的缺省值，而不再为空。

图 7-1
Data
Bindings
面板

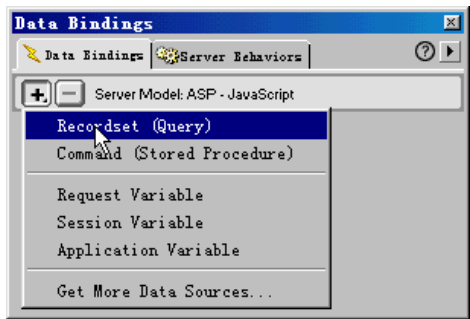
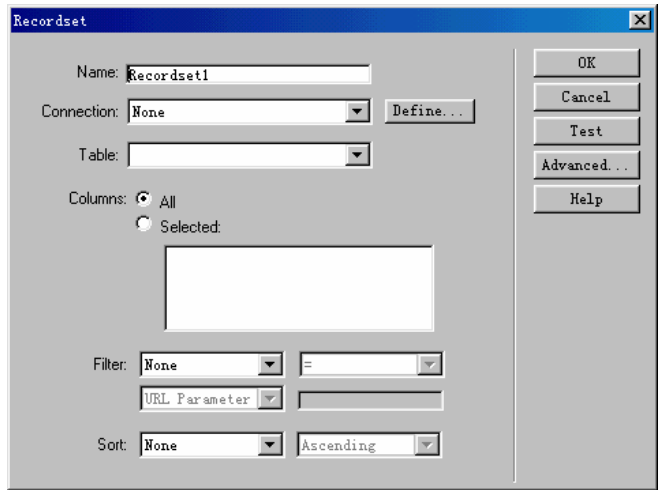


图 7-2
Recordset
窗口



在 Table 下拉列表中选择记录集将要使用的数据库表名，该列表中列出了所链接的数据库中的所有表名。

在 Columns 域的滚动列表中显示了所选择数据库表中的所有字段名。如果想在记录集中包含所选择数据库表中的所有字段，可以保持 Columns 域中的 All 单选钮的选中状态；如果只想使用表中的某些字段，则请选择 Selected，然后在下面的滚动列表中选择特定的字段，如果要选择多个字段，请借助 Ctrl 键来实现，即在用鼠标单击选择各字段的同时，按住 Ctrl 键不放。

如果只想包含数据库表中的某些特定记录，则需要对 Filter 域作一些配置。首先在第一个下拉列表中，选择一个字段，以便能够通过这个字段对记录进行过滤。在第二个下拉列表中，选择一个适当的条件运算符（如=、<、>、>=、<=、<>等），以便每一个记录中所设定字段的值能够同基准值进行比较。在第三个下拉列表中，选择 Entered Value 项，然后在第四个下拉列表中输入进行比较的基准值。这样，在创建记录集时，便会对所有的记录进行比较，只有满足这时所设定的过滤条件的记录才会被放入到记录集中。

除此之外，我们还可以对记录集中的记录进行排序，这可以通过 Sort 域来控制。在

第一个下拉列表中，选择一个字段，以便能够以这个字段为基准进行排序，然后在第二个下拉列表中选择进行排序的方法，如 Ascending(升序，如 1、2、3 或 A、B、C 等) 和 Descending(降序，如 3、2、1 或 C、B、A)。

设置完成后，就可以对记录集进行测试了，判断记录集是否创建成功。单击 Test 按钮，系统便会连接到指定的数据库并创建一个记录集的实例。例如，图 7-3 所示的就是单击 Test 按钮后所出现的 Test SQL Statement 窗口。其中每一行代替一条记录，而每一列表示记录中的每一个字段。当然，该窗口中的实际记录列表是根据读者所设定的条件来产生的，因此用户所得到的 Test SQL Statement 窗口可能和图 7-3 有所不同。

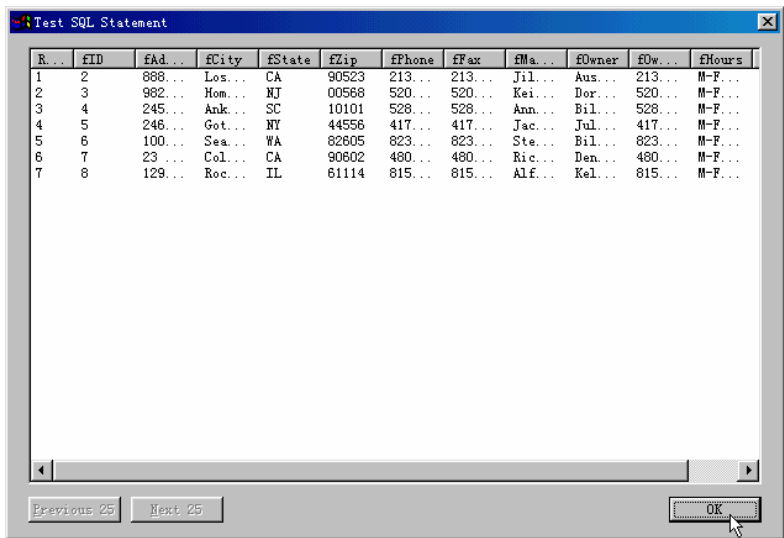


图 7-3
Test SQL
Statement
窗口

如果对测试结果满意，请连续两次单击 OK 按钮。这时在 Data Bindings 面板的列表中就会发现刚才添加的记录集，如图 7-4 所示。单击其中的‘ + ’号就可以展开记录集，查看其中的各个字段；而单击‘ - ’号可以收缩记录集。

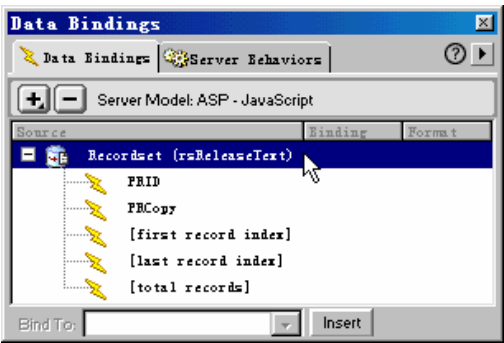


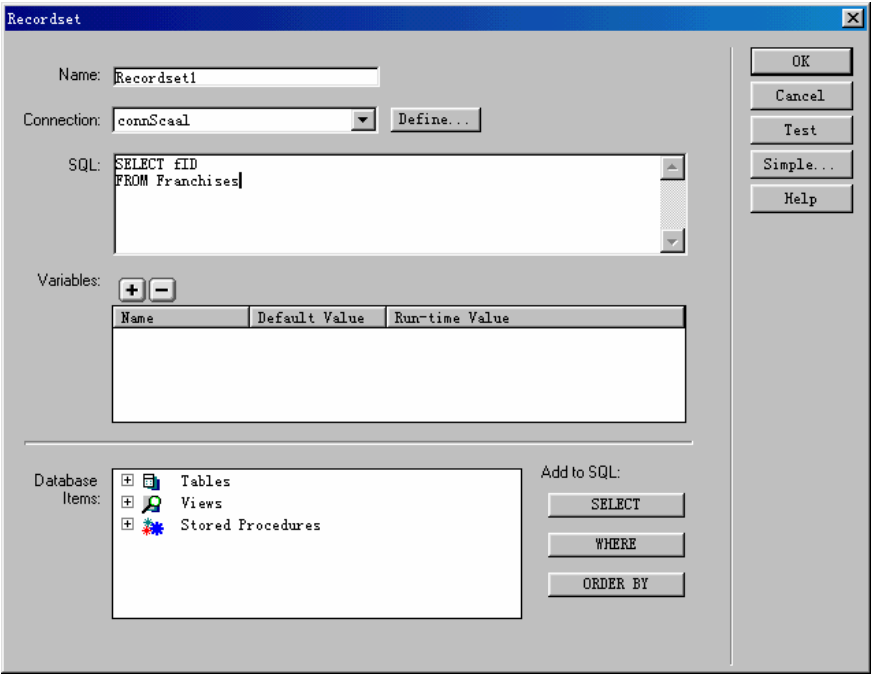
图 7-4
Data
Bindings
面板中的
记录集

7.2.2 使用 SQL 定义记录集

如果读者对 SQL 比较熟悉,那么也可以在 DW UltraDev 中使用 SQL 来定义记录集。这也是通过 Recordset 窗口来完成的,只不过此时要使用该对话框的高级部分。下面介绍其实现步骤。

用户在图 7-2 所示的 Recordset 窗口中会发现一个 Advanced 按钮 **Advanced...**,单击此按钮,便会打开如图 7-5 所示的 Recordset 窗口。这是高级版本的 Recordset 窗口,单击此窗口中的 Simple 按钮 **Simple...**,便会返回到前面介绍过的简单版本的 Recordset 窗口。

图 7-5
高级版本
的
Recordset
窗口



同样,在 Name 文本框中输入记录集的名称,并在 Connection 下拉列表中选择一个数据库链接。然后根据创建记录集的条件,在 SQL 文本框中输入正确的 SQL 语句,当然,这需要读者掌握基本的 SQL 语法。

为了减少读者的输入量,DW UltraDev 还作了一些个性化的处理。在图 7-5 所示的 Recordset 窗口的下方读者会看到 Database Items 域及其右边的三个按钮:SELECT、WHERE 和 ORDER BY。通过这些控制项,我们只需借助鼠标单击动作就能够完成大多数的 SQL 语句输入。Database Items 域的列表中包含了一些目录树,通常有表(Tables)、视图(Views)和存储过程(Stored Procedures)这三种目录树。单击其中的加号“+”可

展开目录树，而单击其中的减号“ - ”可收缩目录树，如图 7-6 所示。



图 7-6
Database
Items 字段
中的目录
树

为了借助目录树来输入 SQL 语句，首先请确保 SQL 文本框中为空，然后展开 Database Items 域的列表中的表 (Tables) 目录树，直到找到需要的数据库对象，如数据库表或表中的字段。选择该数据库对象，然后单击目录树右边的 Add to SQL 域中的任何一个按钮，即可在 SQL 文本框中加入相应的 SQL 语句。例如，如果选择了数据库表 PressReleases，然后单击了 SELECT 按钮，则在 SQL 文本框中就会出现类似“SELECT * FROM PressReleases”的 SQL 语句。同样，还可以借助 WHERE 按钮对记录进行过滤，使用 ORDER BY 按钮对记录进行排序。当然，使用这些控制项只能输入一些比较简单的 SQL 语句，一些复杂的 SQL 语句还是需要读者使用键盘在 SQL 文本框中输入，例如，使用了 WHERE 按钮后，还必须在 WHERE 语句中加入条件判断语句的基准值，甚至还要修改条件判断运算符 (将“=”号改为“<>”号等)。尽管如此，使用这些控制项还是一定程度地减少了用户的输入量。

如果要在 SQL 语句中使用变量，则可以通过 Variables 域来定义这些变量。单击 Variables 域中的  按钮便会在下面的列表中增加一行，读者可以依次在其中输入变量名、变量的缺省值以及运行时刻的值 (通常是一个服务器对象从浏览器所取得的值，如一个 ASP 的 Request 对象所返回的结果)。

例如，假设所请求的页面中有一个 HTML 表单，其中有一个名为“Name”的域，则在 ASP 中这个域的运行时刻的值应该为 Request("Name")，或 Request.Form("Name")，或 Request.QueryString("Name")，具体使用哪一种值应该取决于表单提交的方式 (如 GET 或 POST)；对于 ColdFusion 来说，其运行时刻的值应该为 #Name#；对于 JSP 来说，其运行时刻的值应该为 request.getParameter("Name")。

如果要增加多个变量，则重复上述步骤即可。如果需要删除其中的一个自定义变量，则请选择该变量，然后单击  按钮即可。图 7-7 中给出了在 Variables 域中进行相关操作的过程及结果。

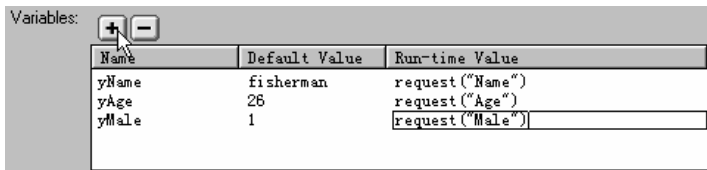


图 7-7
Variables
域

为了进一步讲述如何使用这些控制项，下面我们举一个稍微复杂一点的例子。

假设我们所操作的数据库中有一个名为 Employees 的表，其中有 emplNo、emplName 和 emplJob 这三个字段。为了生成一个 emplJob 字值为‘ varJob ’、包含字段 emplNo 和 emplName、并且根据 emplName 字段进行排序的记录集，我们要创建如下的 SQL 语句：

```
SELECT emplNo, emplName
FROM Employees
WHERE emplJob = 'varJob'
ORDER BY emplName
```

如果借助上面所讲的控制项，则可以按下面的方法方便地生成该 SQL 语句。

首先展开 Database Items 域的列表中的表 (Tables) 目录树，并进一步展开 Employees 表目录树，在展开的列表中选择 emplNo 字段，然后单击 SELECT 按钮，于是在 SQL 文本框中出现“ SELECT emplNo FROM Employees ”的 SQL 语句；同样，选择 emplName 字段，然后单击 SELECT 按钮，于是 SQL 文本框的 SQL 语句变为“ SELECT emplNo, emplName FROM Employees ”。

接下来选择 emplJob 字段，并单击 WHERE 按钮，于是 SQL 文本框的 SQL 语句变为“ SELECT emplNo, emplName FROM Employees WHERE emplJob ”。然后选择 emplName 字段，并单击 ORDER BY 按钮，于是 SQL 文本框的 SQL 语句变为“ SELECT emplNo, emplName FROM Employees WHERE emplJob ORDER BY emplName ”。

将光标输入点放置到 SQL 文本框中的 WHERE emplJob 之后，然后输入“ = 'varJob' ”。“ varJob ”是我们自己定义的一个变量，因此还需要通过 Variables 域来定义它。单击 Variables 域中的  按钮，然后在下面列表的新增行中依次输入下面的变量名、变量的缺省值以及运行时刻的值：varJob、CLERK、Request("job")。

于是 SQL 语句创建完成。

然后就可以对记录集进行测试了。在图 7-5 所示的 Recordset 窗口单击 Test 按钮，系统便会连接到指定的数据库并创建一个记录集的实例，然后将其显示在 Test SQL Statement 窗口中。如果对测试结果满意，请连续两次单击 OK 按钮。这时读者在 Data Bindings 面板的列表中就会发现刚才添加的记录集。

7.2.3 为结果页面定义记录集

所谓结果页面是指当用户在一个检索页面中提交表单之后对表单进行处理并返回处理结果的页面。通常在结果页面中包含了用来对数据库进行查询的 SQL 语句。该 SQL 语句通常要使用一个 Web 浏览器上的检索页面传递过来的查询条件，而这些查询条件又通常是保存在 SQL 语句的变量中的，最后通过这个 SQL 语句来产生结果记录集。

例如，假设在公司的网站上有一个检索页面，员工能够在该页面中查询所有收入大于某一标准的顾客名单。常见的情况是，员工在检索页面的某一个表单域中输入一个最低的收入标准值，然后单击“提交”按钮。于是该收入标准值就会被 Web 服务器传递给结果页面的 SQL 语句中，而结果页面中的 SQL 语句就根据这个收入标准值创建一个所有收入大于该标准值的记录集。

在 DW UltraDev 中为结果页面定义记录集并不需要读者了解 SQL。如同 7.2.1 节中介绍的一样，我们也可以使用简单版本的 Recordset 窗口来定义这样的记录集。



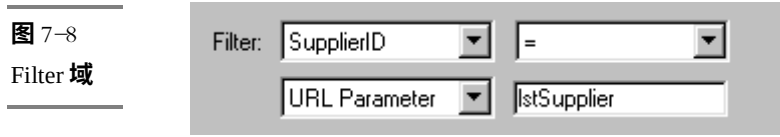
简单版本的 Recordset 窗口只能定义包含一个查询条件的记录集。如果需要使用多个查询条件，或者你愿意使用 SQL 语句为结果页面定义查询条件更加复杂的记录集，还是必须使用高级版本的 Recordset 窗口，其使用方法可参考 7.2.2 节。

下面介绍使用简单版本的 Recordset 窗口为结果页面定义记录集的步骤。

首先定义一个新的记录集或者打开一个已有的记录集。定义新记录集的方法请参见 7.2.1 节和 7.2.2 节。如果要打开一个已有的记录集，请打开 Data Binding 面板（选择“Window / Data Bindings”菜单命令或单击启动面板上的 Data Bindings 按钮）或 Server Behavior 面板（选择“Window / Server Behaviors”菜单命令或单击启动面板上的 Server Behaviors 按钮），然后在列表中的记录集名上双击鼠标左键，于是便会打开简单版本的 Recordset 窗口。如果已经打开高级版本的 Recordset 窗口，请单击其中的 Simple 按钮切换为简单版本的 Recordset 窗口。

然后对 Filter 域进行设置。在第一个下拉列表中，选择一个字段，以便能够将这个字段的值和检索页面传送过来的值进行比较，从而对记录进行过滤。在第二个下拉列表中，选择一个适当的条件运算符（如=、<、>、>=、<=、<>等），以便组成一个条件表达式。如果检索页面中的 HTML 表单使用了 GET 提交方法，则请在第三个下拉列表中选择 URL Parameter 项；如果检索页面中的 HTML 表单使用了 POST 提交方法，则请在第三个下拉列表中选择 Form Variable。在缺省情况下，HTML 表单使用的是 GET 提交方法，因此通常应该选择 URL Parameter 项。在第四个下拉列表中，输入用来给结果页面传递值的表单域的名字。

例如，假设我们想创建一个只包含某个特定供应商的产品的记录集，并且在数据库表中有一个名为 SupplierID 的字段，再假定检索页面中的 HTML 表单使用了 GET 提交方法，且表单中包含了一个名为 lstSupplier 的表单域，则我们应该按照图 7-8 所示来设置 Filter 域中的各个控制。在第一个下拉列表中选择 SupplierID 字段，在第二个下拉列表中选择等于号“=”，在第三个下拉列表中选择 URL Parameter 项，然后在最后一个文本框中输入 lstSupplier。



设置完成后，我们同样也可以对记录集进行测试。如果对测试结果满意，连续两次单击 OK 按钮，于是记录集创建完成。

7.3 操作记录集

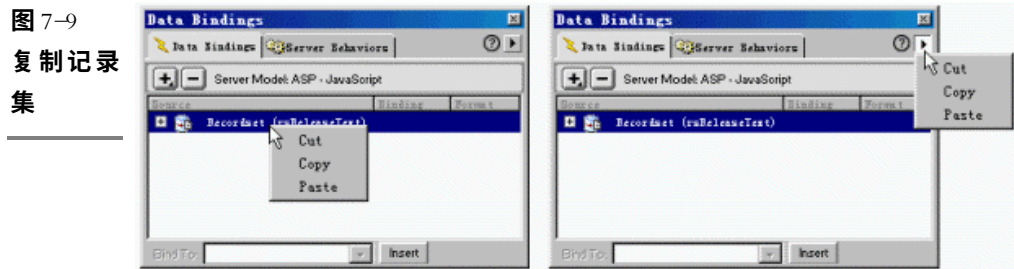
7.3.1 复制记录集

在站点中我们可以把定义好的记录集从一个页面复制到另一个页面，实现方法有两种。首先在文档窗口或 Live Data 窗口中打开已经定义好记录集的页面，然后在 Data Bindings 面板或 Server Behaviors 面板中选择定义好的记录集。

第一种实现方法是在选择的记录集上单击鼠标右键，于是会弹出一个快捷菜单，其中有三个菜单项：Cut、Copy 和 Paste。选择 Copy 即可将记录集复制到粘帖板，如图 7-9 左图所示。

第二种实现方法是用鼠标左键单击面板右上方的  按钮，于是也会出现一个快捷菜单，其中也有同样的三个菜单项，选择 Copy 即可将记录集复制到粘帖板，如图 7-9 右图所示。

将记录集复制到粘帖板中以后，在文档窗口或 Live Data 窗口中打开目标页面，然后采用上述同样的方法将记录集粘帖到目标页面中，只是此时要在快捷菜单中选择 Paste 命令。这样就完成了将记录集从一个页面复制到另一个页面的任务。



7.3.2 调用存储过程

可以将存储过程理解为一个可以被调用的函数或过程。存储过程中包含了一些存储在数据库上的一个或多个 SQL 语句,在程序中可以直接调用存储过程以实现其中所包含的 SQL 语句能实现的功能。存储过程可以返回一个或多个记录集。在 DW UltraDev 中也可以使用存储过程来创建记录集。但值得注意的是, DW UltraDev 只支持那些返回一个记录集或不返回记录集的存储过程。

下面介绍在 DW UltraDev 中使用存储过程创建记录集的步骤。

- 1 首先打开需要使用记录集的页面,然后在 Data Binding 面板中单击  按钮,在出现的菜单中选择 Recordset (Query) 菜单项,于是出现简单版本的 Recordset 窗口。单击窗口中的 Advanced 按钮切换到高级版本的 Recordset 窗口。
- 2 在 Recordset 窗口中输入记录集名,并选择包含了存储过程的数据库链接(即在使用存储过程创建记录集之前,必须在数据库中创建存储过程,然后在 DW UltraDev 创建与该数据库的链接)。接着在 Database Items 域的列表中扩展 Stored Procedures 目录树,选择你想使用的存储过程,然后单击 PROCEDURE 按钮。
- 3 如果存储过程带有参数,还需要在 Variables 域中定义这些参数变量的缺省值和运行时刻值。最后单击 OK 按钮即可完成记录集的创建。

7.3.3 创建存储过程对象

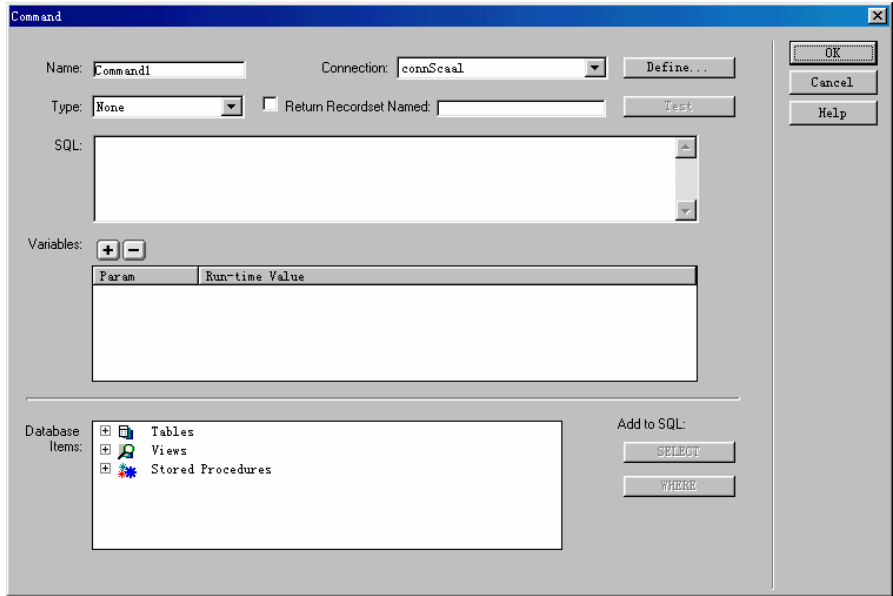
我们可以创建一个存储过程服务器对象,然后在 Web 应用中调用该对象。在 ASP 中,存储过程对象被称为 Command;在 JSP 中,存储过程对象被称为 Callable;而在 ColdFusion 中,存储过程对象被称为 Stored Procedure。

一个存储过程对象不仅可以包含一个记录集,也可以包含其他数据,如输出参数。而一个由存储过程定义的记录集对象却只能包含一个记录集。

下面介绍创建一个存储过程对象的步骤。

- 1 首先在 DW UltraDev 中打开任何一个动态页面,然后在 Data Binding 面板中单击  按钮,在出现的菜单中选择下面的一个菜单项:如果在 ASP 中,选择 Command (Stored Procedure);如果在 JSP 中,选择 Callable (Stored Procedure);如果在 ColdFusion 中,选择 select Stored Procedure。下面以 ASP 为例,会出现如图 7-10 所示的 Command 窗口。

图 7-10
Command
窗口



- 2 在 Name 文本框中输入存储过程的名字,然后在 Connection 下拉列表中选择一连接名,以便确定包含该存储过程的数据库。
- 3 在 Type 下拉列表中选择 Stored Procedure 项,然后选择 Return Recordset Named 复选框,并在右边的文本框中输入要返回的记录集名。在 Database Items 域的列表中扩展 Stored Procedures 目录树,选择你想使用的存储过程,然后单击 PROCEDURE 按钮。
- 4 最后单击 OK 按钮,于是新增的存储过程对象就会出现在 Data Bindings 面板的列表中。

7.3.4 编辑和删除记录集

对已有的记录集我们还可以进行编辑和删除操作。

如果要编辑一个记录集,请打开 Data Bindings 面板或 Server Behaviors 面板,然后在列表中的记录集名上双击鼠标左键,于是便会打开 Recordset 窗口。在该窗口中对记录集的设置进行适当修改,直到满意为止,然后单击 OK 按钮完成编辑操作。

除此之外,我们还可以使用属性面板来编辑记录集。首先打开属性面板(选择“Window / Properties”菜单项),然后在 Server Behaviors 面板中选择记录集,于是在属性面板中就会马上显示出该记录集的相关属性,如图 7-11 所示。

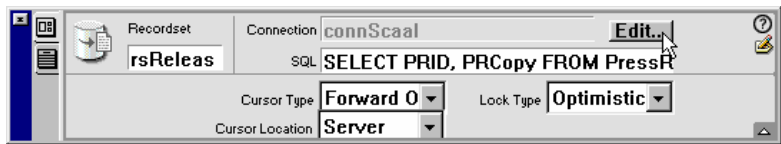


图 7-11
使用属性
面板编辑
记录集

我们可以直接在图 7-11 中的属性面板中修改记录集的有关属性，如果要修改更多的属性，可以单击其中的 Edit 按钮打开 Recordset 窗口进行更详细的编辑。

删除一个记录集的操作非常简单。只需在 Data Bindings 面板或 Server Behaviors 面板中选择要删除的记录集，然后单击  按钮即可。

7.4 编写 SQL 查询语句

本节简单介绍一下如何编写 SQL 语句，以给不熟悉 SQL 的读者一个初步的认识。

7.4.1 编写最简单的 SQL 语句

创建记录集所使用的最简单的 SQL 语句是 SELECT 语句，它的作用是从一个表或多个表中取出一些特定的字段，从而组成一个记录集。下面就是一个 SELECT 语句的基本语法：

```
SELECT FieldName FROM TableName
```

这个 SQL 语句的作用是从表 TableName 中取出包含字段 FieldName 的所有记录，从而组成一个记录集。在书写 SQL 语句时，我们也可以在其中加入回车换行符、制表符或空格，以便使 SQL 语句更加易懂。例如，下面的 SQL 语句也是合法的：

```
SELECT PaidDues
FROM Members
```

7.4.2 包含整个数据库表

如果要在记录集中把数据库表中的所有内容都包含进来，则需要使用通配符“*”来表示数据库表中的所有字段。例如，假设我们有一个名为 Customers 的表，为了取出这个表中的所有字段，需要使用下面的 SQL 语句：

```
SELECT * FROM Customers
```

7.4.3 限制字段的数目

我们也可以限制记录集中所需字段的数目。方法是直接在 SQL 语句中指明需要使用的字段名即可。例如，假设我们只需要表 Customers 中的两个字段（如 YearBorn 字段和 DateLastPurchase 字段）的数据，则需要使用下面的 SQL 语句：

```
SELECT YearBorn, DateLastPurchase FROM Customers
```

7.4.4 限制记录的数目

如果要限制记录集中记录的数目，则需要使用 WHERE 语句。例如，如果我们只想在记录集中包含那些年收入超过 50000 元的客户，假设 Customers 表中有一个名为 Earnings 的字段记录了每个客户的年收入，则需要使用下面的 SQL 语句：

```
SELECT YearBorn, DateLastPurchase FROM Customers
WHERE Earnings > 50000
```

表 7-1 中给出了在由 WHERE 所组成的条件表达式中可以使用的操作符。

表 7-1 WHERE 所组成的条件表达式中可以使用的操作符

操 作 符	含 义
=	等于
LIKE	等于
<>	不等于
NOT LIKE	不等于
<	小于
>	大于
<=	小于或等于
>=	大于或等于

如果比较项是文本，则应该将其放在单引号‘ ’’中，例如：

```
... WHERE Country = 'Germany'
```

如果比较项是日期，并且使用的是 Microsoft Access 数据库，则应该将其包含在#号中，例如：

```
...WHERE DateOfBirth < #01/01/1970#
```

不同的数据库具有不同的日期转换方法，如果使用的是其他的数据库，则应该根据

该数据库的语法使用相应的转换符。

在条件表达式中还可以使用通配符。标准的通配符是百分号‘ % ’，例如：

```
...WHERE LastName LIKE 'Mc%'
```

对于 Access 数据库来说，星号‘ * ’也可以当作通配符来使用，例如：

```
...WHERE CompanyName = '*soft'
```

7.4.5 对记录进行排序

使用 ORDER BY 语句可以对记录集中的记录进行排序。排序的好处是可以使记录集中的记录更加利于用户浏览。

假设我们要对客户的年收入情况从小到大进行排序，则我们应该使用下面的 SQL 语句来排序功能：

```
SELECT LastName, FirstName, Earnings FROM Customers  
ORDER BY Earnings
```

在缺省情况下，ORDER BY 语句是按照升序（如 1, 2, 3...或 A, B, C...等）来进行排序的。如果想对记录进行降序排序，需要使用关键字 DESC，例如：

```
...ORDER BY Earnings DESC
```

以上介绍了一些最常用的 SQL 语句的使用方法，如果想进一步了解其他的 SQL 语句，请自选参考相关书籍。

第 8 章 创建动态网页

本章概要：

- ☒ 在网页中加入动态内容
- ☒ 创建各种不同的动态网页，包括搜索页面、结果页面、细节页面、插入页面、更新页面、删除页面等

Web 应用实际上就是由一系列的动态网页组成的。所谓动态网页，就是在静态网页中增加了一些动态的内容。在创建了记录集以后，我们就可以将记录集中的字段作为数据源往页面中添加动态内容，从而创建动态网页了。本章就介绍这方面的知识。

8.1 在网页中加入动态内容

8.1.1 加入动态内容的步骤

在一个页面中加入动态内容通常包含以下三个步骤：

- 1 首先，至少定义一个数据源来提供动态内容。这个数据源可以是记录集中的某一个字段，或者是一个 HTML 表单提交的数值，或者是一个服务器对象中包含的值，等等。
- 2 其次，要确定动态内容要添加的位置。在 DW UltraDev 中，我们几乎可以将动态内容添加到页面或 HTML 源代码中的任何位置：
 - ※ 可以在插入点放置要添加的动态内容。
 - ※ 可以用动态内容替代一个字符串。
 - ※ 可以将动态内容插入到一个 HTML 属性中。例如，动态内容可以定义为一幅图片的 SRC 属性值，或者定义为一个表单域的 VALUE 属性值。
- 3 最后，选择已经定义好的数据源并将其作为动态内容添加到页面中。DW UltraDev 会在页面中插入一些服务器端的脚本语句，以便指示服务器将数据源中的数据传递

到页面的 HTML 源代码中。

通常有多种方法使一个给定的页面元素动态化。例如，为了使一幅图片动态化，我们可以使用 Data Bindings 面板，也可以使用属性面板，还可以使用“Insert / Image”菜单命令。在后面一些章节中，我们将介绍一些最常用的方法来使不同的页面元素动态化。

在缺省情况下，一个 HTML 页面只能同时显示一条记录。为了能够显示出记录集中的其他记录，通常需要加入一个浏览链接。或者在页面中创建一个重复域，以便在同一个页面中能够显示多个记录。在将动态内容添加到页面中以后，还可以对这些动态内容进行编辑修改。

8.1.2 定义数据源

下面介绍定义一个数据源的步骤。

如果 Data Bindings 面板没有打开，请选择“Window / Data Bindings”菜单命令，或者单击启动面板上的 Data Bindings 按钮打开它。

单击  按钮，于是会出现一个下拉菜单，如图 8-1 所示（假设读者使用的是 ASP）。

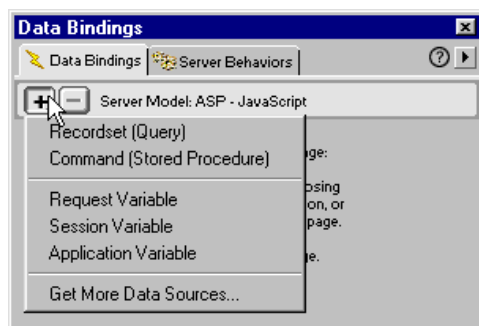
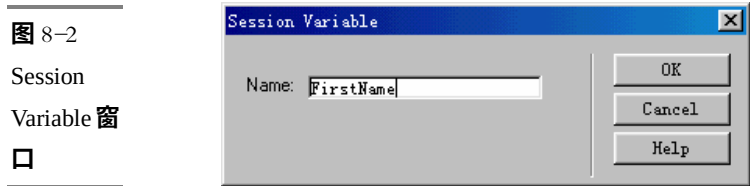


图 8-1
使用 Data
Bindings
面板定义
数据源

该下拉菜单中显示的就是读者可以选择的数据源类型。请从中选择一种来定义数据源。

- ※ 如果在菜单中选择了 Recordset 或 Stored Procedure 菜单项，则数据源的定义方法请参见第 12 章。
- ※ 如果用户选择了其他的数据源，如 Session Variable，则请在 Session Variable 窗口的 Name 域中输入其他相关的服务器对象将要使用的变量名，然后单击 OK 按钮即可。例如，假设我们想使用 Session 对象将访问者的姓保存起来，以便在后面的网页中显示出来，则请在 Name 文本框中输入变量名，如“FirstName”，然后单击 OK 按钮，如图 8-2 所示。



※ 如果选择了 Request Variable 菜单项,则需要在如图 8-3 所示的 Request Variable 窗口的 Type 下拉列表中选择 Request 变量的类型(如 QueryString、Form、Cookie 等),并在 Name 文本框中输入 Request 对象要使用的变量名,然后单击 OK 按钮。

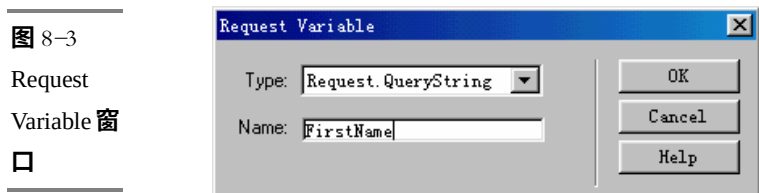
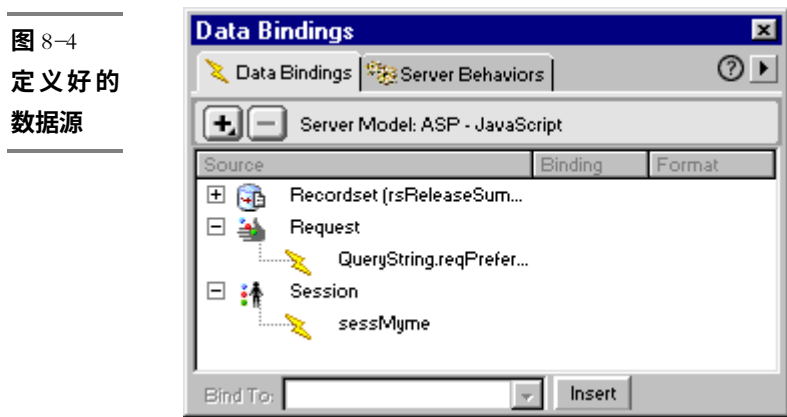


图 8-4 中所示的 Data Bindings 面板中就列出了几种已经定义的数据源,包括一个记录集、一个 Request 变量和一个 Session 变量。



如果需要删除某个定义好的数据源,则请在 Data Bindings 面板的列表中选择需要删除的数据源,然后单击  按钮即可。



在 Data Bindings 面板中删除了一个数据源之后,页面中与该数据源相关的动态内容并不会被删除。

这里还需要对服务器对象作一说明。服务器对象也可以作为 Web 应用的一种数据源。常用的服务器对象包括 Request 对象、Session 对象和 Application 对象等。

Request 对象（用于 ASP 和 JSP）用来存储浏览器提交给服务器的信息。在 ASP 中，Request 对象包含 5 个集合：Request.QueryString、Request.Form、Request.ServerVariables、Request.Cookie 和 Request.ClientCertificates。例如，Request.Form("LastName") 中包含了用户在一个使用 POST 提交方法的 HTML 表单中的 LastName 表元素中输入的值，而 Request.ServerVariables("HTTP_USER_AGENT") 中则包含了与用户的浏览器相关的信息。

Session 对象（用于 ASP、JSP 和 ColdFusion）用来存储一个用户当前的任务信息。例如，一个 Session 对象可以跟踪该用户的名字，以便我们可以做更多个性化的工作（如在后续的页面中显示该用户的名字等）。每个用户都可以创建一个不同的 Session 对象，该 Session 对象会持续一段时间，直到它被终结。

Application 对象（用于 ASP 和 ColdFusion）用来存储应用程序级的相关信息。和 Session 对象相比，它的作用范围更大，类似于全局变量。

8.1.3 使文本动态化

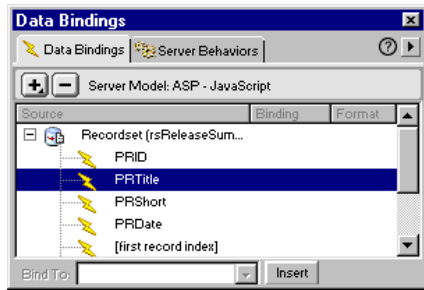
我们可以使用动态文本来代替页面中的静态文本，也可以在页面的输入点后加入动态文本。插入的动态文本会自动使用被替换的静态文本或输入点处的文本格式。例如，如果被选择的静态文本使用了一个 CSS 样式，则使用动态文本替换该静态文本时，动态文本也会自动使用该 CSS 样式。当然，我们也可以使用 DW UltraDev 的文本格式化工具来改变动态文本的格式。

使文本动态化的步骤如下：

- 1 如果 Data Bindings 面板没有打开，请选择“Window / Data Bindings”菜单命令，或者单击启动面板上的 Data Bindings 按钮打开它。
- 2 确保在 Data Bindings 面板的列表中列出了将要使用的数据源。该数据源应该包含普通的文本（即 ASCII 文本）。如果还没有定义数据源，请先按上一小节介绍的方法定义一个。
- 3 在文档窗口或 Live Data 窗口中，选择页面中需要替换的静态文本，或者将输入焦点放置在页面中需要添加动态文本的地方。然后在 Data Bindings 面板的列表中选择将要添加的数据源。如果选择的是一个记录集，请确定要使用记录集中的哪一个字段，如图 8-5 所示。
- 4 单击 Data Bindings 面板中的  按钮，或者将所选择数据源拖放到页面中，于是动态文本就出现在了页面中。如果该页面是在 Live Data 窗口中打开，则动态文本内容就会立即显示出来；如果该页面是在文档窗口中打开，则动态文本会以占位符的形式出现。对于一个记录集数据源来说，占位符的形式通常是：

{RecordsetName.ColumnName}

图 8-5
选择需要
添加的字
段



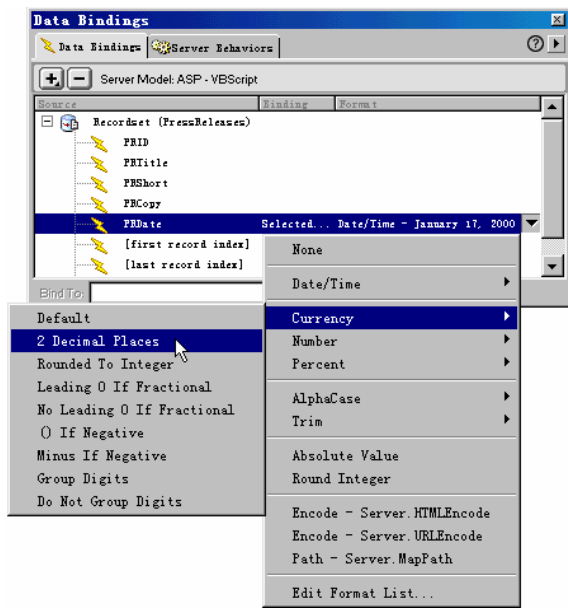
其中，RecordsetName 代表的是记录集的名字，而 ColumnName 代表的是该记录集中的字段名。

另外，我们还可以确定页面中动态文本的数据格式。例如，假设 10.989 是一个代表价钱的数据，则我们可以选择 “Currency / 2 Decimal Places” 菜单命令将其显示为 \$10.99。这种格式可以显示小数点后 2 位数字，如果多于 2 位，则会采取四舍五入的方法进行截减；如果不足 2 位，则会在其后面补 0。

给动态文本应用数据格式的常用步骤如下：

在 Live Data 窗口中选择动态文本内容，或者在文档窗口中选择表示动态文本的占位符，则 Data Bindings 面板对应的字段也被同时被选中，并且 Binding 列和 Format 列中列出了相应的值。单击 Format 列右边的 ▾ 箭头，则会弹出一个菜单，该菜单中对各种不同的数据格式进行了分类，请根据动态数据的类型选择一种合适的数据格式，如图 8-6 所示。

图 8-6
选择动态
文本内容
的数据格
式



8.1.4 使图片动态化

同样，我们也可以将页面中的图片变成动态内容。例如，假设我们想在一个页面中显示某一个班级中的所有学生档案，每个页面中都要包括学生的姓名等描述性信息以及该学生的照片。对于所有的学生来说，页面显示的布局都是一样的，但每个人的照片（和描述性文本信息）都是动态变化的。

使一幅图片动态化的步骤如下：

- 1 将输入点放置在页面中需要出现图片的地方，然后选择 **Insert / Image** 菜单命令，则会出现如图 8-7 所示的 **Select Image Source** 窗口。

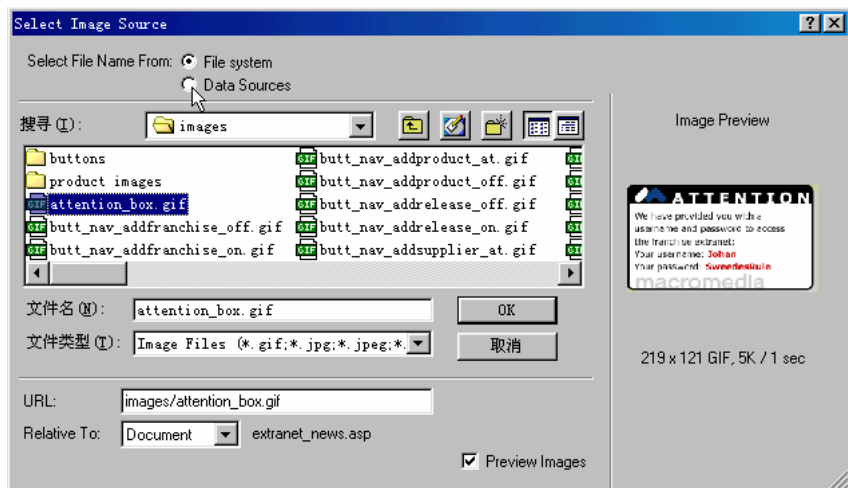


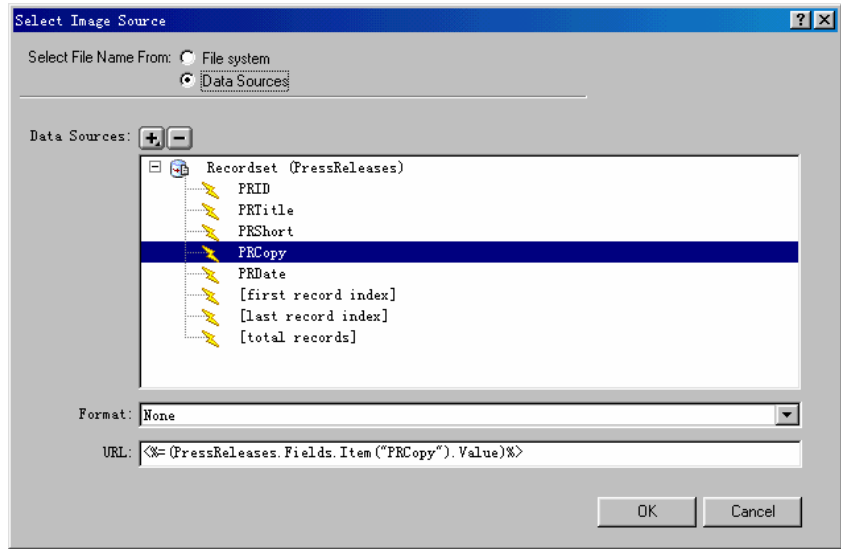
图 8-7
插入图片

- 2 选择窗口上方的 **Data Sources** 单选钮，则窗口会变成如图 8-8 所示的形状，其中列出了所有的数据源。
- 3 请在窗口的列表选择一个适当的数据源。该数据源应该是一个包含了指向图片文件所在路径的记录集。然后单击 **OK** 按钮即可在页面中插入一个动态的图片。



如果在图 8-8 的列表中没有列出记录集，则请单击 **+** 按钮定义一个。另外，DW UltraDev 不支持在数据库中直接插入二进制格式的图片文件，而只能通过指向图片文件的链接来访问它。

图 8-8
选择数据源



8.1.5 使表单动态化

在页面中显示数据库中的记录时使用得最多的就是 HTML 表单了。例如，我们可以设计一个表单来显示所有的学生信息，包括姓名、性别、年龄等。

最常用的动态表单对象包括文本框、图片框、复选框和单选按钮，我们也可以使用数据源来产生列表/菜单表单元素中的各个选项。下面我们介绍分别使这几种表单元素动态化的方法。

1. 使文本框动态化

如果 Data Bindings 面板没有打开，请选择“ Window / Data Bindings ”菜单命令，或者单击启动面板上的 Data Bindings 按钮打开它。

确保在 Data Bindings 面板的列表中列出了将要使用的数据源。该数据源应该包含文本信息。如果还没有定义数据源，请先定义一个。

在文档窗口或 Live Data 窗口中，选择 HTML 表单中的文本框，然后在 Data Bindings 面板的列表选择一个合适的数据源。请确保在 Data Bindings 面板最下方的 Bind To 下拉列表的值为 input . value ，然后单击 Bind 按钮，即可将动态内容绑定到文本框中。如图 8-9 所示。

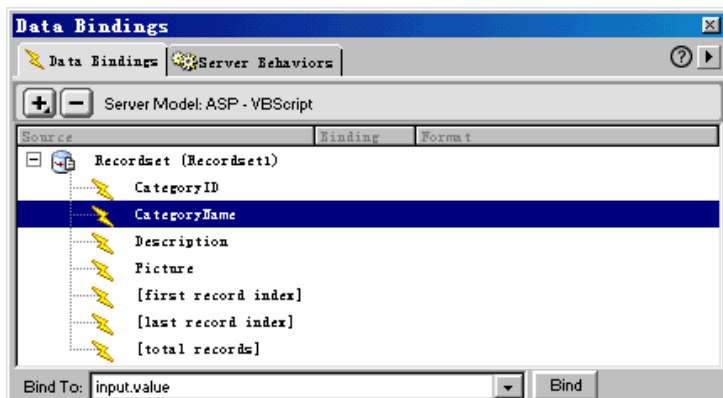


图 8-9
将动态内
容绑定到
文本框中

2. 使图片框动态化

使 HTML 表单中的图片框动态化的步骤和 8.1.4 节中介绍了使图片动态化的步骤是完全一样的，在此不再赘述。

3. 使复选框动态化

首先请选择 HTML 表单中的复选框，然后在 Server Behaviors 面板中单击  按钮，并在弹出的菜单中选择“Dynamic Elements / Dynamic Check Box”菜单项。于是出现如图 8-10 所示的 Dynamic Check Box 对话框。

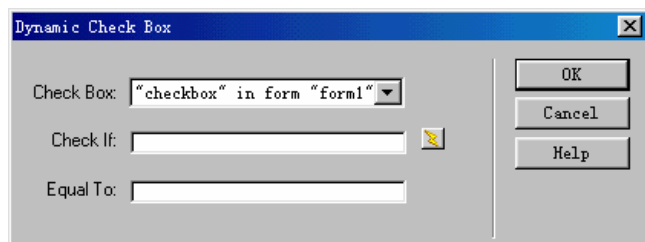


图 8-10
Dynamic
Check Box
对话框

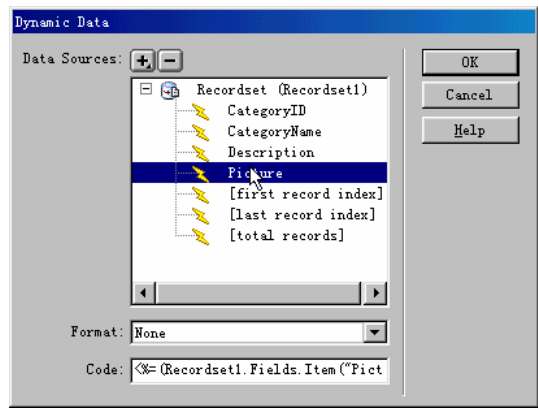
如果想实现下面的功能：当记录中的某一个字段的值等于一个固定值时系统自动选中该复选框，则可以按照下面的方法来完成。

单击图 8-10 所示对话框中 Check If 文本框旁边的  按钮，则会出现如图 8-11 所示的 Dynamic Data 对话框。在其中的列表中选择数据源中的某一个字段，通常，这个字段应该包含的是布尔类型（Boolean）的数据，如 Yes 和 No，或者 TRUE 和 FALSE 等。然后单击 OK 按钮。

在 Equal To 文本框中，输入用于比较的固定值，以决定当记录中选定字段的值等于何值时选中复选框。例如，如果想让该字段的值等于 YES 时选中复选框，请在 Equal To

文本框中输入“ YES ”。

图 8-11
Dynamic
Data 对话框

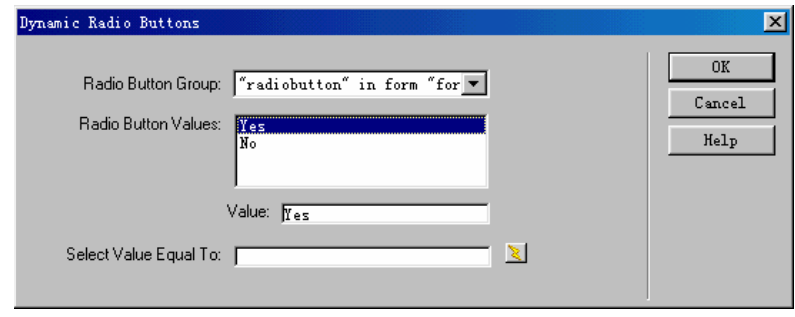


最后单击 OK 按钮。于是，该复选框被选中与否就完全依赖于相应字段的值了。

4. 使单选钮动态化

首先请确保页面中至少有一组单选钮。在 Server Behaviors 面板中单击  按钮，并在弹出的菜单中选择“ Dynamic Elements / Dynamic Radio Button ”菜单项。于是出现如图 8-12 所示的 Dynamic Radio Buttons 对话框。

图 8-12
Dynamic
Radio
Buttons 对话框



在 Radio Button Group 下拉列表中选择页面中的一组单选钮，然后我们就可以确定该组中所有单选钮的值。方法是先在 Radio Button Values 列表中选中一个单选钮，然后在 Value 文本框中输入它的值即可。

如果想实现下面的功能：当记录中的某一个字段的值等于一个单选钮的值时系统自动选中该单选钮，则可以按照下面的方法来完成。

单击图 8-12 所示对话框中 Select Value Equal To 文本框旁边的  按钮，同样也会出现 Dynamic Data 对话框。在其中的列表中选择数据源中的某一个字段(该字段的值将与单选钮的值进行比较)，然后单击 OK 按钮即可。

5. 使列表/菜单对象动态化

首先选择 HTML 表单中的列表/菜单对象，然后在 Server Behaviors 面板中单击  按钮，并在弹出的菜单中选择“Dynamic Elements / Dynamic Menu”菜单项。于是出现如图 8-13 所示的 Dynamic List/Menu 对话框。

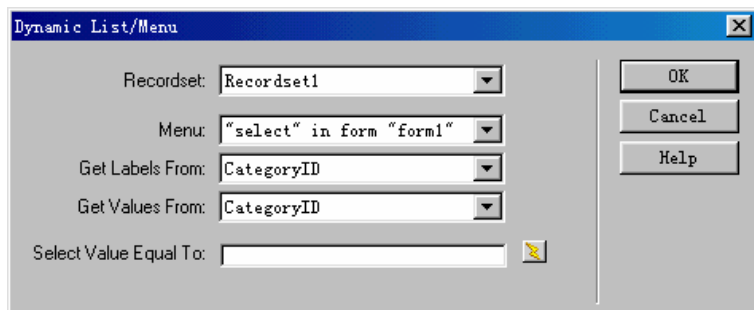


图 8-13
Dynamic
List/Menu
对话框

在 Recordset 下拉列表中选择包含菜单信息的记录集，在 Menu 下拉列表中选择页面中的某一个列表/菜单对象。接着在 Get Labels From 下拉列表中选择包含菜单项标签的字段名，在 Get Values From 下拉列表中选择包含菜单项值的字段名。

如果想实现这样的功能：当该页面在浏览器中打开的时候某一个选定的菜单项被选中，则请在 Select Value Equal To 文本框中输入与该菜单项相等的值即可。最后单击 OK 按钮。

8.1.6 使 HTML 属性动态化

通过把 HTML 属性绑定到数据中，可以让我们动态地改变页面的外观。例如，我们可以将一个表格的 BACKGROUND 属性绑定到一个记录集中的某一个字段中，然后通过改变这个字段的值来改变表格的背景颜色。

使用 Data Bindings 面板和属性面板都可以实现使 HTML 属性动态化的功能。

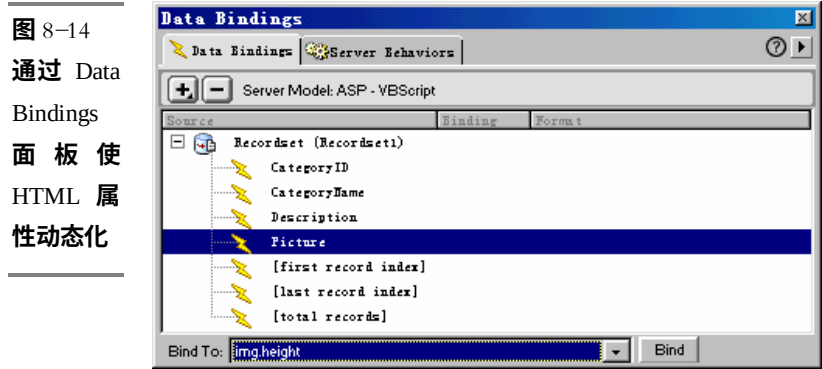
1. 使用 Data Bindings 面板使 HTML 属性动态化

如果 Data Bindings 面板没有打开，请选择“Window / Data Bindings”菜单命令，或者单击启动面板上的 Data Bindings 按钮打开它。

确保在 Data Bindings 面板的列表中列出了将要使用的数据源。该数据源应该包含了与将要绑定的 HTML 属性相关的数据。如果还没有定义数据源，请先定义一个。

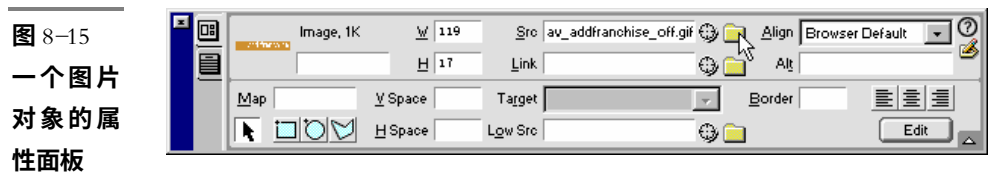
在文档窗口或 Live Data 窗口中，选择某一个 HTML 对象，然后在 Data Bindings 面板的列表选择一个数据源，然后在 Bind To 下拉列表选择一个 HTML 属性，以便绑

定到该数据源中，如图 8-14 所示。最后单击 **Bind** 按钮完成。



2. 使用属性面板使 HTML 属性动态化

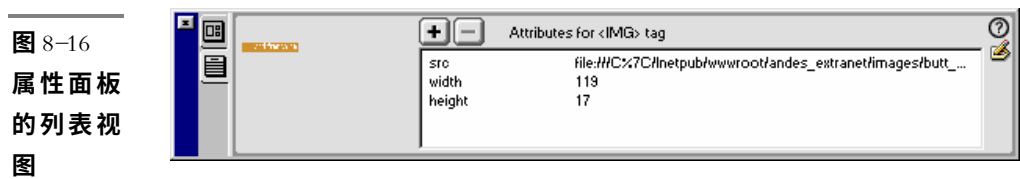
(1) 在文档窗口或 Live Data 窗口中，选择某一个 HTML 对象，然后打开属性面板（选择“Window / Properties”菜单命令），图 8-15 中所示的就是一个图片对象的属性面板。



(2) 如果在属性面板的标准视图中，所要绑定的 HTML 属性旁边有一个文件夹图标 (如图 8.15 中所示的 Src 属性)，则请单击该图标打开一个文件选择窗口，然后单击 Data Sources 单选钮显示出数据源列表。接着跳转到 (6)。

(3) 如果在属性面板的标准视图中，所要绑定的 HTML 属性旁边没有文件夹图标 (如图 8-15 中所示的宽度属性 W)，则请单击属性面板左边的 List 标签切换到列表视图，如图 8-16 所示。

(4) 如果要绑定的 HTML 属性没有出现在列表视图的列表中，则请单击其中 **+** 按钮，然后在新增项中输入属性名或者单击 **-** 按钮在弹出的菜单中选择属性名。



(5) 为了使某个属性的值动态化, 请单击该属性, 然后单击该属性值最右边的  按钮, 则会打开数据源列表窗口。

(6) 在该窗口中选择一个合适的数据源, 该数据源应该包含了与将要绑定的 HTML 属性相关的数据。如果还没有定义数据源, 请先定义一个。

(7) 最后单击 OK 按钮即可。

8.1.7 使其他对象动态化

除此之外, 我们还可以使其他的对象动态化, 包括 Java Applets、ActiveX、Flash、Shockwave、Director、Generator 等对象以及其他一些插件。当然, 在将这些对象的属性或参数绑定到记录集中的字段时, 必须要保证这些字段中有合适的值。

通常, 使一个对象动态化的步骤如下:

 在文档窗口中选择该对象, 然后打开属性面板。图 8-17 中所示的就是一个 ActiveX 对象的属性面板。

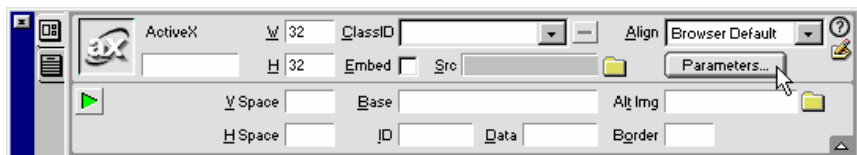


图 8-17
ActiveX
对象的属
性面板

 单击其中的 Parameters 按钮 , 打开 Parameters 窗口, 如图 8-18 所示。

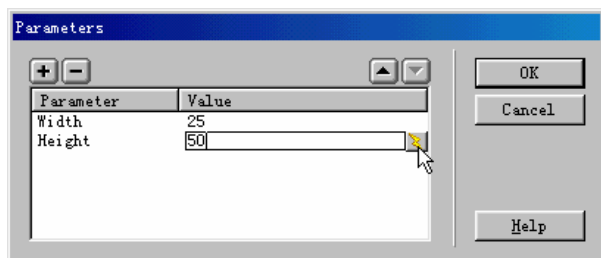


图 8-18
Parameters
窗口

 如果窗口的列表中还没有参数, 请单击  按钮进行添加。然后单击需要绑定参数的 Value 列右边的  按钮, 打开数据源列表窗口, 在其中选择一个合适的数据源。该数据源应该包含了与将要绑定的对象参数或属性相关的数据。如果还没有定义数据源, 请先定义一个。最后单击 OK 按钮即可。

8.2 创建不同类型的动态网页

一个 Web 应用中可以包含各种不同的动态网页。最常用的动态网页有搜索网页、结果网页、细节网页和记录编辑网页（即允许用户通过网页对数据库中的记录进行添加、更新和删除操作）。创建不同类型的动态网页需要不同的 HTML 代码、动态内容和服务器的行为。

8.2.1 创建搜索网页

在 Web 应用中，搜索网页是非常普遍的。用户可以在这种类型的网页中输入查询信息，并将其传送给服务器进行处理，服务器处理结束后再将结果传回到结果网页中，并显示在用户的浏览器上。

创建搜索网页是非常简单的。第一步工作就是使用 DW UltraDev 的编辑工具创建一个用户输入界面，其中至少包含一个带提交按钮的 HTML 表单。这个表单中应该根据各种查询条件来设置相应的表单元素，如文本框、复选钮、按钮等。

接着就需要设置这个 HTML 表单的 Action 属性的值，这个值就是结果网页的文件名。当用户在搜索网页中按下提交按钮之后，就把控制权交给这里所确定的结果网页。



一个页面既可以是搜索网页，也可以是结果网页。也就是说，该网页把查询条件传送到服务器上自己的一个副本中，并由该副本来充当结果网页。

8.2.2 创建结果网页

创建了搜索网页之后，就需要创建一个结果网页来查找并显示搜索结果。通常结果网页是获取搜索网页所提交的查询条件，然后形成一个查询语句在数据库中进行查询，并将查找到的记录显示出来。

同样，创建一个结果网页的第一步也是设计用户界面，不过这里的界面通常是一个 HTML 表格，用来显示多个查找到的记录。

第二步工作是以用户在搜索网页中提交的查询条件为基础来创建一个记录集。这个过程同创建普通的记录集是类似的，只不过要稍微复杂一些。它需要使用高级版本的 Recordset 窗口，并且可能需要根据实际情况自己编写 SQL 查询语句，并创建相应的变量。

创建结果网页的第三步工作是在页面中增加动态内容。其实现方法在本章前面已经介绍过。第四步工作是在页面中增加服务器行为，以便网页能够正常地工作。通常，由于结果网页中要显示多条记录，因此首先要添加一个 Repeat Region 服务器行为；结果网页另一个常见的特性是其中需要添加到细节页面的链接，以便显示某条记录更详细的信息。该链接不仅要打开细节页面，还要能告诉细节页面获取并显示哪一条记录，因此这个功能需要依靠 Go to Detail Page 服务器行为来完成；另外，如果要完善结果页面，还需要为其添加浏览链接，以使用户能够轻松地浏览其他记录，这就需要依靠 Move to Record 服务器行为来完成；为了处理浏览链接的边界效应，即当用户浏览到第一条或最后一条记录时，相应链接还存在，但却并没有相应的功能，这时应该借助 Hide Region 服务器行为将其隐藏起来。

表 8-1 中列出了创建结果网页的最低要求。

表 8-1 创建结果网页的最低要求

内 容	要 求
HTML	设计者可随意安排，但通常都是以表格的形式出现
动态内容	绑定到一个记录集的表格内容
服务器行为	Repeat Region 行为：在结果网页中显示多个记录 Move to Record 行为：用来创建浏览记录的链接 Hide Region 行为：在一些特殊情况下（如第一页或最后一页）隐藏一些浏览链接 Go to Detail Page 行为：创建到细节页面的链接

8.2.3 创建细节网页

细节网页通常是与结果网页联系在一起的。细节网页需要利用结果网页提供的信息来获取并显示某一个特定的记录。

创建细节网页的第一步也是设计用户界面。第二步是在细节网页中添加动态内容。最后一步是增加服务器行为，以便该网页能够基于结果网页的信息来找到正确的记录。这时可能需要的服务器行为包括 Move to Specific Record、Hide Region、Go to Related Page 等。

表 8-2 中列出了创建细节网页的最低要求。

表 8.2 创建细节网页的最低要求

内 容	要 求
HTML	设计者可随意安排
动态内容	绑定到一个记录集的网页内容
服务器行为	Move to Specific Record 行为：在细节网页中移动到一个特定的记录 Move to Record 行为：用来创建浏览记录的链接 Hide Region 行为：在一些特殊情况下（如第一页或最后一页）隐藏一些浏览链接 Go to Related Page 行为：创建到相关页面的链接

8.2.4 创建记录编辑网页

除此之外，我们还可以创建一些允许用户对记录进行编辑的网页，包括插入记录的网页、更新记录的网页和删除记录的网页。

同样，创建记录编辑网页的第一步是设计用户界面，其中至少需要一个带提交按钮的 HTML 表单。对于插入操作和更新操作来说，通常表单中需要把数据库中的所有字段都用表单元素表示出来（即数据库中的字段应该能够与表单中的各个表单元素一一对应起来）；而对于删除操作来说，就没有必要列出所有的字段了，只需列出几个比较独特的字段即可。

第二步工作是添加相应的服务器行为以便完成相应的记录编辑操作，如插入记录的网页中需要添加 Insert Record 服务器行为，更新记录的网页中需要添加 Update Record 服务器行为，而删除记录的网页中需要添加 Delete Record 服务器行为。这些服务器行为通常首先都要选择数据库表，然后要确定 HTML 表单中的各个表单元素与数据库表中字段的对应关系，最后还要确定一个页面的 URL 地址，以便编辑操作完成后在浏览器中显示该页面，例如插入操作完成后重新显示结果网页，以便能及时地将新插入的记录显示出来。

表 8-3、表 8-4 和表 8-5 中分别列出了创建插入记录的网页、创建更新记录的网页和创建删除记录的网页的最低要求。

表 8-3 创建插入记录的网页的最低要求

内 容	要 求
HTML	一个带提交按钮的表单
动态内容	无
服务器行为	Insert Record 行为：在数据库表中插入一个新记录

表 8-4 创建更新记录的网页的最低要求

内 容	要 求
HTML	一个带提交按钮的表单
动态内容	绑定到一个只含单一记录的记录集的表单对象
服务器行为	Update Record 行为：更新数据库表中的记录 Move to Record 行为：用来创建浏览记录的链接 Hide Region 行为：在一些特殊情况下（如第一页或最后一页）隐藏一些浏览链接

表 8-5 创建删除记录的网页的最低要求

内 容	要 求
HTML	一个带提交按钮的表单
动态内容	绑定到一个只含单一记录的记录集的表单对象
服务器行为	Delete Record 行为：删除数据库表中的记录 Move to Record 行为：用来创建浏览记录的链接 Hide Region 行为：在一些特殊情况下（如第一页或最后一页）隐藏一些浏览链接

第 9 章 激活动态网页

本章概要：

- ☒ 在同一网页中显示多条记录
- ☒ 在记录间进行移动
- ☒ 在页面之间传递信息
- ☒ 编辑记录

在上一章我们介绍了创建动态网页的基本方法，但是并没有具体介绍如何使用服务器行为来实现各种动态网页的特殊功能，如在同一页面中显示多条记录、在记录间进行移动、在页面之间传递信息以及编辑记录等。本章我们就来学习这方面的知识。

9.1 概述

创建动态网页的下一步工作就是“激活”动态网页，即给网页添加一些“智能化”的功能，以便它能够按照我们的预想进行工作。通常，这些智能化的功能都是通过在页面中增加服务器行为来实现的。一个服务器行为实际上就是一个运行在服务器端（而不是浏览器端）的 VBScript、JavaScript、Java 或 ColdFusion 脚本。

例如，在创建了一个往数据库中插入新记录的 HTML 表单之后，就必须增加一个服务器行为来完成实际的工作：将表单中的数据取出来，然后插入到数据库中。

DW UltraDev 提供了许多已经定义好的服务器行为，我们可以直接使用它们来激活动态网页。当然，我们也可以自己编写特定的服务器行为，或者在 DW UltraDev 中安装他人编写的服务器行为。

随着用户所创建的动态网页的不同，需要添加的服务器行为也会不一样。在上一章中我们已经看到了这一点。

下面，我们通过一些具体的操作来介绍如何在网页中添加服务器行为来实现各种不同的功能。

9.2 显示多条记录

我们已经知道,在同一个页面中显示多条记录需要使用 Repeat Region 这个服务器行为。同时,Data Bindings 面板也提供了一个记录集统计,我们可以使用它来为显示多个记录的页面来创建一个记录计数。

9.2.1 创建 Repeat Region 服务器行为

为了在一个页面中显示多条记录,我们必须指定一个包含了动态内容的选择区域作为重复区域(Repeated Region)。任何选择区域都可以转换成一个重复区域,最常用的是表格、表格中的某一行或多行。

例如,我们可以设计一个表来显示一个班级中所有学生的情况,表中的每一行都会显示一个学生情况,而每一列都会显示学生的不同信息(如姓名、性别、年龄等)。

要想创建这种方式的表格,只需将 Repeat Region 服务器行为应用到该表格中包含动态内容的行就可以了。虽然在设计的时候该重复区域(即被 Repeat Region 服务器行为所作用的表格行)只包含了一行,但是在实际运行时,该行会被重复多次并且每行中都会被插入不同的记录。

创建 Repeat Region 服务器行为是非常简单的,只需按下面的步骤进行操作即可。

- 1 首先请选择页面中包含了动态内容的一个区域,如表格、表格行或者一段文本。
- 2 如果 Server Behaviors 面板没有打开,请选择“Window / Server Behaviors”菜单命令,或者单击启动面板上的 Server Behaviors 按钮打开它。然后单击该面板中的  按钮,在弹出的菜单中选择“Repeat Region”菜单项,于是会出现如图 9-1 所示的 Repeat Region 对话框。

图 9-1
Repeat
Region 对
话框



- 3 请在 Recordset 下拉列表中选择一个记录集名,如果还没有定义记录集,请先定义一个。然后在 Show 域中确定每页中将要显示的记录数。如果选择了 All Records 单选钮,则会在该页中显示记录集中的所有记录;如果选择了带文本框的 Records

单选钮, 则请在文本中输入确定的数值(如 10), 则每页中都会显示该固定数目的记录。如果记录比较多, 需要分多页显示, 则必须创建一些记录浏览链接, 以使用户能够轻松地各个记录间进行浏览。

 设置完成后, 单击 OK 按钮, 即可在页面中创建一个重复区域。现在请将页面切换到 Live Data 窗口(选择“View / Live Data”菜单命令), 就会发现我们刚才创建的重复区域已经被重复多次并且被插入了不同的记录。

9.2.2 创建记录计数

在一些搜索结果页面中, 我们经常可以看到这样的提示: “显示第 1-8 条记录, 共 31 条记录”。这里的 31 就是记录计数。在结果页面中记录计数是非常重要的, 它可以让搜索者很清楚地知道一共找到了多少满足条件的记录。

借助 Data Bindings 面板我们可以创建这样的记录计数。下面我们就来创建一个典型的计数, 按照同样的方法我们也可以创建其他类型的计数。

首先请在文档窗口中输入下面的文字:

显示第-条记录, 共条记录

然后将输入焦点放置到“共”字后面。如果 Data Bindings 面板没有打开, 请选择“Window / Data Bindings”菜单命令, 或者单击启动面板上的 Data Bindings 按钮打开它。如果还没有定义记录集, 请先定义一个记录集。

在 Data Bindings 面板中展开该记录集, 找到并选择数据源列表中的 [total records] 项, 如图 9-2 所示。

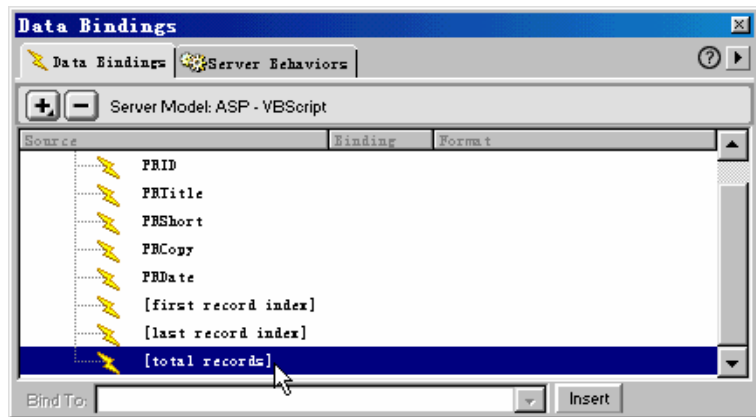


图 9-2
选择
[total
records] 数
据源

然后将 [total records] 拖放到页面中, 或者单击 Insert 按钮将其插入到页面中, 文档窗口中的文字变成下面的样子:

显示第-条记录, 共 {PressReleases_total} 条记录

显示第 {PressReleases_first} - 条记录, 共 {PressReleases_total} 条记录

表中的[first record index]项,将其拖放到页面中,或者单击  按钮将其插入到页面中,文档窗口中的文字变成下面的样子:

同样,将输入焦点放置到破折号(-)后面,然后在 Data Bindings 面板中选择数据源列表中的[last record index]项,将其拖放到页面中,或者单击  按钮将其插入到页面中,文档窗口中的文字变成下面的样子:

现在请将文档窗口切换到 Live Data 窗口,于是刚才添加的计数文字可能会变成如下

显示第{PressReleases_first}-{PressReleases_last}条记录,共{PressReleases_total}条记录

显示第 1-5 条记录,共 16 条记录

所示的样子:

用户显示的数字可能会和这里的不一样。因为这里的计数都是根据记录集中的记录数统计出来的,不同的记录集统计出来的结果自然会不一样。而且这里已经假定该页面中已经加入了 Repeat Region 服务器行为,并且设定了每页显示 5 条记录。



如果在该页面中已经加入了能够浏览到其他记录的浏览链接,如“下一页”链接,则单击该链接时页面中会显示下面的 5 条记录,同时计数文字也会自动发生变化:

显示第 6-10 条记录,共 16 条记录



由于链接在 Live Data 窗口中无法工作,因此要想对上述操作进行测试就要使用 DW UltraDev 的浏览器预览功能。首先请确保在 Preferences 窗口(使用“Edit / Preferences / Preview in Browser”菜单命令打开)中选择了“Preview Using Application Server”复选框,如图 9-3 所示。然后按下 F12 快捷键,或者选择“File / Preview in Browser”菜单命令打开浏览器进行测试即可。

9.3 在记录间移动

一个好的 Web 应用应该具有这样的功能:用户可以方便地在记录之间进行浏览。也就是说,如果我们在一个动态网页中限定了每页只显示 5 条记录,而整个记录集中共有 50 条记录可以显示,则我们应该在该动态网页中增加一些浏览链接,如“上一页”、“下一页”等。这样,用户借助这些链接就可以轻松地访问上 5 条记录和下 5 条记录。

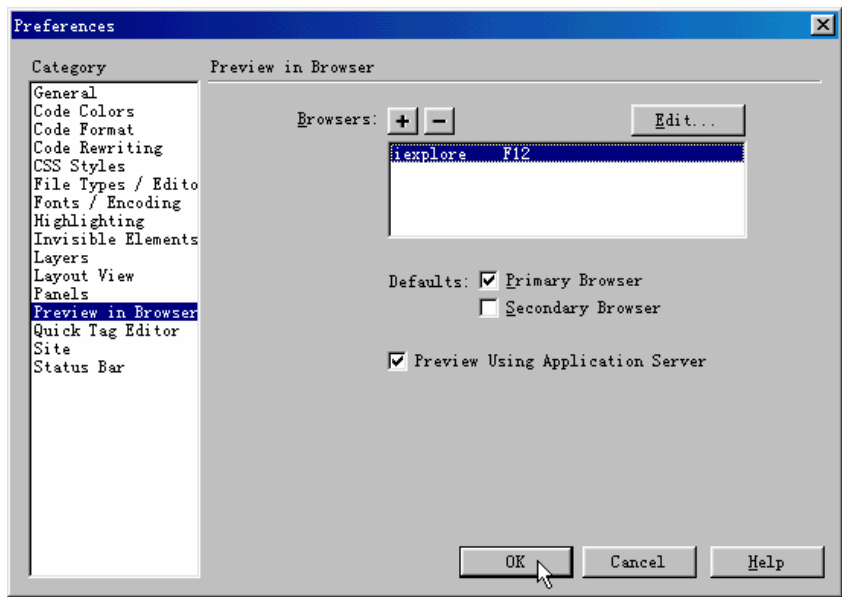


图 9-3
设置浏览
器预览选
项

在 DW UltraDev 中，这个功能实现起来也非常简单，只需在页面中添加适当的服务
器行为即可。为了方便用户在记录间进行移动，UltraDev 可以允许我们创建 4 种类型的
浏览链接：First(第一个)、Previous(上一个)、Next(下一个) 和 Last(最后一个)。

9.3.1 创建记录浏览链接

首先在页面中选择需要作为链接的字符串或图片。如果 Server Behaviors 面板没有打
开，请选择“ Window / Server Behaviors ”菜单命令，或者单击启动面板上的 Server
Behaviors 按钮打开它。然后单击该面板中的  按钮，在弹出的菜单中选择“ Move to
Record ”，于是会展开如图 9-4 所示子菜单。

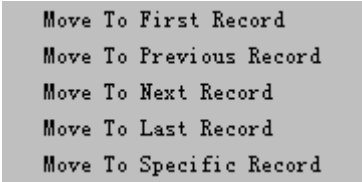
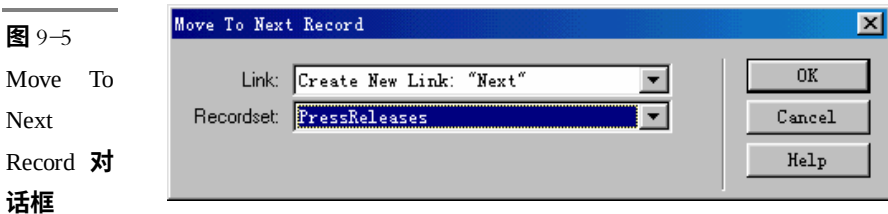


图 9-4
Move to
Record 子
菜单

该子菜单中列出了 5 种不同的服务器行为，分别是：Move To First Record(移动到
第一条记录)、Move To Previous Record(移动到上一条记录)、Move To Next Record(移
动到下一条记录)、Move To Last Record(移动到最后一条记录) 和 Move To Specific

Record(移动到特定的记录)。请根据自己的需要在子菜单中选择适当的服务器行为(创建浏览链接通常使用前 4 个服务器行为, 第 5 个服务器行为在下一节中将作介绍)。在打开的对话框中(前 4 种服务器行为打开的对话框都是类似的, 图 9-5 所示的就是 Move To Next Record 服务器行为所对应的对话框), 只需确保在 Recordset 下拉列表中选择了正确的记录集, 然后单击 OK 按钮, 即可创建相应的浏览链接。



按照同样的方法我们可以在页面中创建多个不同的浏览链接。

9.3.2 移动到特定记录

通常在一个 Web 应用中会包含一个细节页面, 用来显示一个主页面(通常是结果页面) 中列出的某一记录的详细信息。DW UltraDev 提供了 Move To Specific Record(移动到特定的记录) 这个服务器行为来实现移动到主页面所确定的特定记录的功能。具体实现步骤如下:

- 1 首先在文档窗口中打开细节页面, 请确信在细节页面中包含了一些可以显示细节信息的动态内容。
- 2 如果 Server Behaviors 面板没有打开, 请选择“ Window / Server Behaviors ”菜单命令, 或者单击启动面板上的 Server Behaviors 按钮打开它。然后单击该面板中的  按钮, 在弹出的菜单中选择“ Move to Record ”, 在展开的子菜单中选择“ Move to Specific Record ”菜单命令, 于是出现如图 9-6 所示的 Move to Specific Record 对话框。
- 3 在 Move to Record In 下拉列表中, 选择正确的记录集。接着在 Where Column 下拉列表中选择主页面将要传递给细节页面的字段名, 例如, 如果主页面要通过记录的 ID 编号来标识细节页面中所显示的记录, 则可以选择包含记录 ID 号的字段。然后在 Matches URL Parameter 文本框中输入主页面将要传递给细节页面的值所对应的变量名, 该变量名会出现在主页面中的 Go to Detail Page 服务器行为中。
- 4 最后单击 OK 按钮即可完成设置。但是, 要想在细节页面中正确显示相关信息, 还必须对主页面做相应的处理, 这一点将在 9.4.2 节中进行介绍。



图 9-6
Move to
Specific
Record 对
话框

9.3.3 增强浏览功能

在第 9.3.1 节中介绍创建记录浏览链接时，我们忽视了对一些边界问题（即当用户浏览到第一条记录或最后一条记录时情况）的处理。这往往会出现一些问题，例如，假设在页面中已经增加了“上一条”和“下一条”这两个记录浏览链接，当用户浏览到第一条记录时，此时“上一条”链接已经没有意义了，因为现在没有上一条记录了；同样，当用户浏览到最后一条记录时，“下一条”链接也没有意义了。

为了解决这个问题，DW UltraDev 提供了 Show Regions 这个服务器行为。它可以允许我们在一些特殊的情况下显示或隐藏相应的记录浏览链接。例如，当用户浏览到第一条记录时，它可以自动将“上一条”链接隐藏起来；当用户浏览到最后一条记录时，它能够自动将“下一条”链接隐藏起来；甚至当结果记录集为空时，它能够将所有相关链接都隐藏起来。

隐藏不需要的记录浏览链接的具体步骤如下。

首先请选择该记录浏览链接（如“上一条”链接）。如果 Server Behaviors 面板没有打开，请选择“Window / Server Behaviors”菜单命令，或者单击启动面板上的 Server Behaviors 按钮打开它。然后单击该面板中的  按钮，在弹出的菜单中选择“Show Region”菜单命令，于是会出现如图 9-7 所示的“Show Region”子菜单。

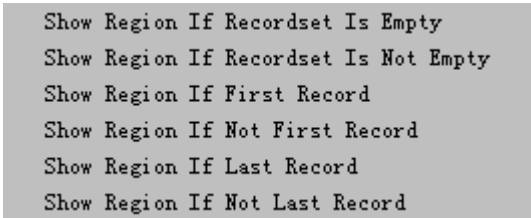


图 9-7
“ Show
Region ”子
菜单

“ Show Region ”子菜单中包含有 6 个菜单命令，它们的作用分别如下：

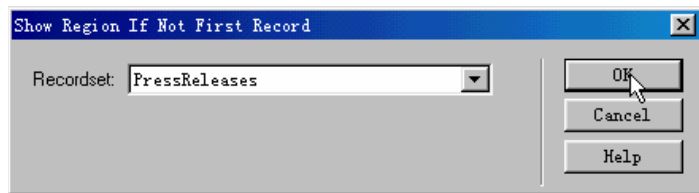
- ✧ Show Region If Recordset Is Empty：如果记录集为空，则显示浏览链接；反之，如果记录集不为空，则隐藏浏览链接。

- ※ Show Region If Recordset Is Not Empty :如果记录集不为空,则显示浏览链接;反之,如果记录集为空,则隐藏浏览链接。
- ※ Show Region If First Record :如果浏览到第一条记录,则显示浏览链接;反之,如果浏览的不是第一条记录,则隐藏浏览链接。
- ※ Show Region If Not First Record :如果浏览的不是第一条记录,则显示浏览链接;反之,如果浏览的是第一条记录,则隐藏浏览链接。
- ※ Show Region If Last Record :如果浏览到最后一条记录,则显示浏览链接;反之,如果浏览的不是最后一条记录,则隐藏浏览链接。
- ※ Show Region If Not Last Record :如果浏览的不是最后一条记录,则显示浏览链接;反之,如果浏览的是最后一条记录,则隐藏浏览链接。

这6个菜单命令包括了浏览时遇到的所有边界情况,在实际使用时我们可以根据浏览链接的功能来选择正确的菜单命令。例如,如果所选择的记录浏览链接是“上一条”链接,则应该选择“Show Region If Not First Record”菜单命令,即当浏览到第一条记录时隐藏“上一条”链接;同样,如果所选择的记录浏览链接是“下一条”链接,则应该选择“Show Region If Not Last Record”菜单命令,即当浏览到最后一条记录时隐藏“下一条”链接,依次类推。

当用户选择了某一个菜单命令后,系统通常会弹出一个对话框,图9-8所示的就是当用户选择了“Show Region If Not First Record”菜单命令后出现的对话框。用户只需在此对话框中确定要作用的记录集,即在Recordset下拉列表中选择正确的记录集即可。

图 9-8
Show
Region If
Not First
Record 对
话框



最后单击 OK 按钮即可。按照同样的步骤对其他记录浏览链接进行设置。

9.4 在页面之间传递信息

在 Web 应用中经常需要在不同的页面之间传递信息或参数。这个功能是非常重要的

的。例如，我们可以通过一个页面告诉另外一个页面显示哪条记录，或者保存某一个页面的信息，以便切换到另外一个页面时可以重用这些信息，避免用户重复输入同样的信息。

DW UltraDev 提供了下面两个服务器行为来实现信息在页面之间的传递：Go to Detail Page 和 Go to Related Page。

通常，一个 HTML 表单中的参数是用以下两种方式传递到服务器的：

- ✱ 如果表单使用 POST 提交方法将信息传送到服务器，则相应的参数是作为消息体的一部分来发送的。
- ✱ 如果表单使用 GET 提交方法，则相应的参数会附加到该表单的 Action 属性所确定的 URL 后进行发送。

如果表单使用 GET 提交方法，则在 URL 后面会增加一个问号“?”，而参数信息会附加在该问号的后面。问号的作用是将地址信息同参数信息区分开来。例如：

```
http://www.mysite.com/search_details.asp?model_number=3701
```

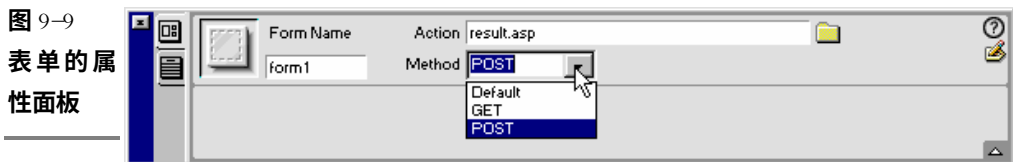
该 URL 的第一部分“http://www.mysite.com/search_details.asp”，将会打开相应的细节页面，而第二部分“?model_number=3701”，则告诉细节页面将显示哪一条记录。其中，model_number 是 HTML 表单中相应的表单域对象名，而 3701 则是用户在该域中输入的值。

9.4.1 获取 HTML 表单中的数据

在 Web 应用中要大量用到 HTML 表单来获取用户输入的数据或信息。下面我们介绍如何来配置一个 HTML 表单，以便获取用户输入的信息。

首先在页面中增加一个 HTML 表单(该表单中一定要有一个提交按钮，以便将表单数据传送到服务器)，然后在文档窗口左下角的状态栏上单击“<form>”标记选择该表单。

如果属性面板还没有打开，则请选择“Window / Properties”菜单命令打开它，其中会显示出该表单的相关属性。如图 9-9 所示。



在 Form Name 文本框中输入表单名，在 Action 文本框中输入服务器中要用来处理表单提交的数据的页面名，或者单击图标定位到该页面。在 Method 下拉列表中选择

表单的提交类型，如 GET、POST、Default。其中 Default 表示的是浏览器缺省采用的方式，通常是 GET。

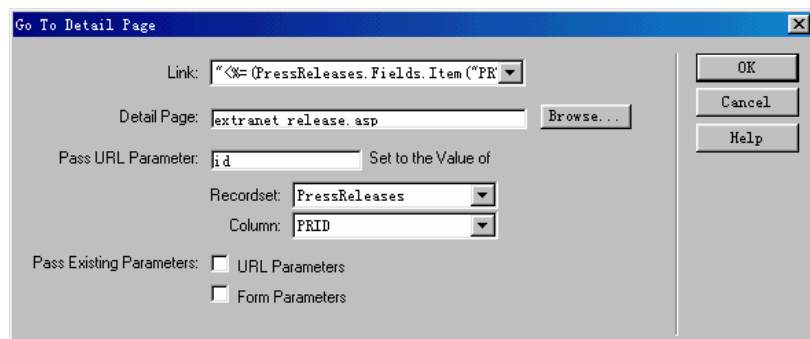
9.4.2 定位到细节页面

要想正确的定位到细节页面，还必须对主页面做相应处理。也许有的读者会想，直接在主页面中加入一些标准的 HTML 链接不就可以了吗？其实仔细一想就会知道这样是不行了，因为无法告诉细节页面显示哪一个记录。

Go to Detail Page 服务器行为可以为主页面创建一个特殊的链接，并能够正确地将信息传递到细节页面。实现定位到细节页面的功能的具体步骤如下：

- 1 在主页面中，选择需要做为链接的动态内容。该动态内容通常是一个位于重复区域中的数据占位符。
- 2 如果 Server Behaviors 面板没有打开，请选择“Window / Server Behaviors”菜单命令，或者单击启动面板上的 Server Behaviors 按钮打开它。然后单击该面板中的  按钮，在弹出的菜单中选择“Go to Detail Page”菜单命令，于是出现如图 9-10 所示的 Go to Detail Page 对话框。

图 9-10
Go to
Detail
Page 对话框



- 3 在 Detail Page 域中，单击  按钮定位到细节页面。在 Pass URL Parameter 文本框中输入用来传递表单值的变量名，而该变量名的值由 Recordset 和 Column 这两个下拉列表中的值共同来确定。Recordset 下拉列表中确定的是记录集名，Column 下拉列表中确定的是字段名。
- 4 如果包含 HTML 表单的主页面使用 POST 提交方法将数据传递给服务器，则我们也可以选择上 Form Parameters 复选框，以便将表单参数也传递给服务器。
- 5 最后单击 OK 按钮即可完成设置。

9.4.3 定位到相关页面

首先在页面中选择文本或图片对象,以便将其作为定位到相关页面的链接。在 Server Behaviors 面板中单击  按钮,在弹出的菜单中选择“Go to Related Page”,于是出现如图 9-11 所示的 Go to Related Page 对话框。

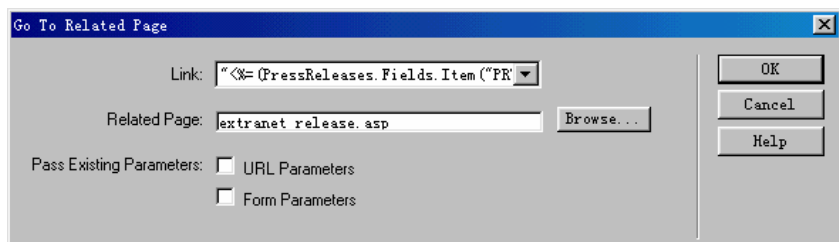


图 9-11
Go to
Related
Page 对话框

在 Related Page 文本框中,单击  按钮定位到相关的页面文件。如果包含 HTML 表单的主页面使用 POST 提交方法将数据传递给服务器,则我们也可以选择上 Form Parameters 复选框,以便将表单参数也传递给服务器。最后单击 OK 按钮即可完成设置。

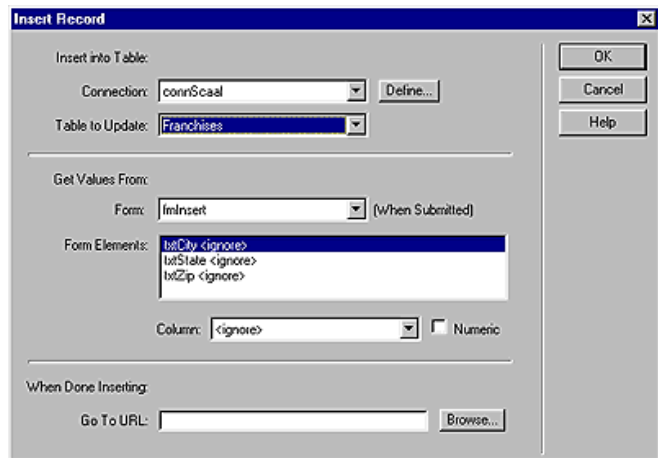
9.5 编辑记录

9.5.1 在数据库中插入记录

 首先请创建一个记录插入页面,该页面中应该包含一个带提交按钮的 HTML 表单。在文档窗口左下角的状态栏上单击“<form>”标记选择该表单。然后在 Server Behaviors 面板中单击  按钮,在弹出的菜单中选择“Insert Record”菜单命令,于是出现如图 9-12 所示的 Insert Record 对话框。

图 9-12

Insert
Record 对话框



- ② 在 Connection 和 Table to Update 这两个下拉列表中确定新记录将要插入的数据库表。在 Form 下拉列表中，确定页面中用于输入新记录数据的 HTML 表单名。然后确定每一个表单元素所对应的字段，方法是先在 Form Elements 列表中选择某一个表单元素，然后在 Column 下拉列表中选择与之对应的字段名。如果某字段只能接受数值，则请选择 Numeric 复选框。
- ③ 在 Go to URL 文本框中，输入或者借助 Browse... 按钮定位某一个页面的 URL。当新记录被插入到数据库表中以后，浏览器就会自动重新定位到这里所指定的页面中。
- ④ 最后单击 OK 按钮即可完成设置。

9.5.2 在数据库中更新记录

在 Web 应用中通常还会包含这样的功能：用户可以在页面中，对数据库中的记录进行修改和更新。这样的记录更新页面通常是一个细节页面，当然它也需要主页面的配合。主页面允许用户选择一个需要更新的记录，然后相关信息就会传递给细节页面并打开它。

要实现这样的功能，必须执行两种不同的操作：首先要在页面中显示现有的数据，以便用户可以对其进行修改；然后提交页面，用修改后的数据完成对数据库的更新。

1. 显示现有数据

首先在更新页面上创建一个 HTML 表单来显示数据。在文档窗口左下角的状态栏上单击“<form>”标记选择该表单。接下来我们要在更新页面中创建一个记录集，以便能对其中的记录进行更新。

在 Data Bindings 面板中单击  按钮，然后在出现的菜单中选择 Recordset (Query) 菜单项，于是出现简单版本的 Recordset 窗口（如果出现的是高级版本的 Recordset 窗口，

请单击其中的 Simple 按钮切换到简单版本的 Recordset 窗口)。

在 Name 文本框中给记录集进行命名, 然后在 Connection 下拉列表中选择数据库连接, 在 Table 下拉列表(其中显示的是数据库中的所有表名)中选择某一个数据库表, 并在 Columns 域中, 选择 Selected 单选钮, 以便只选择某些特定的字段。

在下面的列表选择一个关键字字段(通常是记录集 ID 编号字段), 然后根据需要选择其他需要显示在页面中的字段(可能要借助 Ctrl 键)。当然, 在表单中应该有一个与字段相对应的表单元素, 以便字段能够显示在表单中, 同时用户可以进行查看和修改。

接下来要配置 Filter 域, 以便保证关键字字段的值等于主页面通过相应的 URL 参数传递过来的值。这样就可以保证记录集中只有一条记录, 即用户在主页面中所确定的记录。例如, 假设当前数据库表中包含记录 ID 信息的字段名叫 PRID, 而且主页面是通过一个名为“id”的 URL 参数来传递相应的记录 ID 信息的, 则应该按图 9-13 所示来选择或填写相关设置。

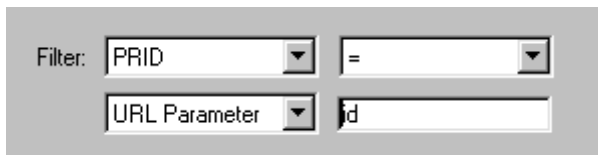


图 9-13
设置过滤
条件

最后我们还需要将每一个 HTML 表单对象绑定到记录集中的相应字段上。具体实现方法请参见 8.1.5 节的内容。

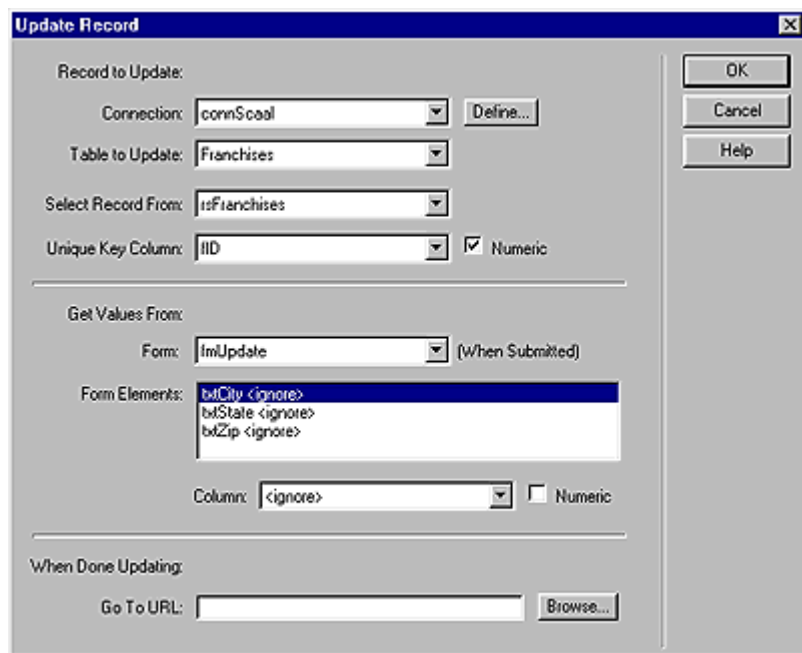
2. 更新数据库

在 Server Behaviors 面板中单击  按钮, 在弹出的菜单中选择“Update Record”菜单命令, 于是出现如图 9-14 所示的 Update Record 对话框。

在 Connection 和 Table to Update 这两个下拉列表中确定包含了需要进行更新的记录的数据库表。在 Select Record From 下拉列表中, 确定上面刚刚创建的记录集名(即包含了显示在 HTML 表单中的记录的记录集)。在 Unique Key Column 下拉列表中, 选择一个关键字字段来惟一标识数据库表中的该记录。如果其值为数值, 则请选择 Numeric 复选框。

在 Form 下拉列表中, 确定用户用来编辑修改记录数据的 HTML 表单名。然后确定每一个表单元素所对应的字段, 方法是先在 Form Elements 列表中选择某一个表单元素, 然后在 Column 下拉列表中选择与之对应的字段名。如果某字段只能接受数值, 则请选择 Numeric 复选框。

图 9-14
Update
Record 对
话框



在 Go to URL 文本框中，输入或者借助 **Browse...** 按钮定位某一个页面的 URL。当数据库表中的记录被更新以后，浏览器就会自动重新定位到这里所指定的页面中。

最后单击 OK 按钮即可完成设置。

9.5.3 在数据库中删除记录

在 Web 应用中通常还会包含这样的功能：用户可以在页面中对数据库中的记录进行修删除。这样的记录删除页面通常也是一个细节页面，当然它也需要主页面的配合。主页面允许用户选择一个需要删除的记录，然后相关信息就会传递给细节页面并打开它。

要实现这样的功能，必须执行两种不同的操作：首先要在页面中显示现有的数据，以便用户可以查看将要删除的数据；然后提交页面，将该记录从数据库中删除。

对于第一步操作来说，其实现方法同上一小节介绍的“在数据库中更新记录”的第一步操作是完全一样的，因为它们的目的基本是一样的。

完成第一步操作之后，在 Server Behaviors 面板中单击 **+** 按钮，在弹出的菜单中选择“Delete Record”菜单命令，于是出现如图 9-15 所示的 Delete Record 对话框。

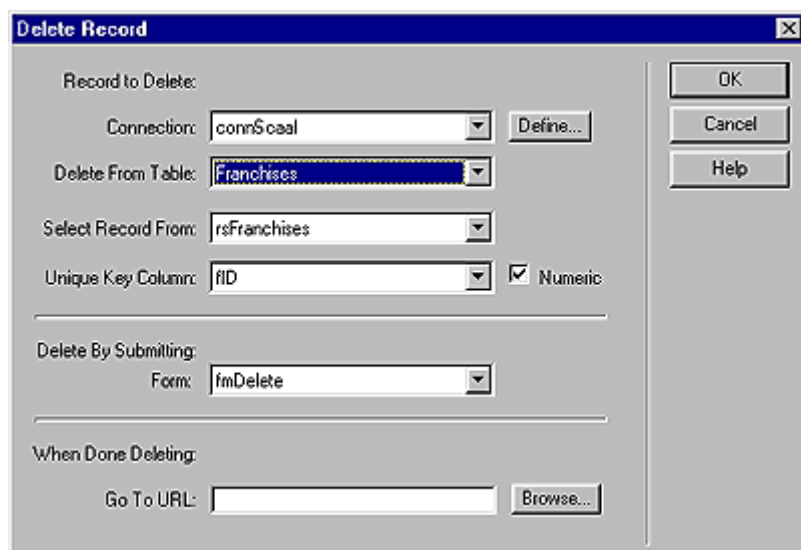


图 9-15
Delete
Record 对
话框

在 Connection 和 Delete From Table 这两个下拉列表中确定包含了将被删除的记录的数据表。在 Select Record From 下拉列表中，确定上面刚刚创建的记录集名（即包含了显示在 HTML 表单中的记录的记录集）。在 Unique Key Column 下拉列表中，选择一个关键字段来惟一标识数据库表中的该记录。如果其值为数值，则请选择 Numeric 复选框。

在 Form 下拉列表中，确定用来显示将被删除的记录数据的 HTML 表单名。在 Go to URL 文本框中，输入或者借助 Browse... 按钮定位某一个页面的 URL。当数据库表中的记录被删除以后，浏览器就会自动重新定位到这里所指定的页面中。最后单击 OK 按钮即可完成设置。

第 10 章 编辑和调试动态网页

本章概要：

- ☒ 对动态网页作基本的编辑操作
- ☒ 在 Live Data 环境下进行编辑
- ☒ 测试 Web 应用
- ☒ 编辑源代码
- ☒ 编辑和创建数据格式

在创建了静态网页，并在其中增加了动态内容和服务器行为之后，创建 Web 应用的最后一步就是对网页进行相应的编辑和调试，以便它能够正常工作。在 DW UltraDev 中有多种方法对动态网页进行编辑和调试：

- ※ 可以在文档窗口中进行编辑。
- ※ 可以在显示动态内容的 Live Data 窗口中进行编辑。
- ※ 可以在浏览器中对 Web 应用进行测试。
- ※ 可以使用 HTML 源代码检查器来编辑 HTML 代码或服务器端脚本，或者使用文档窗口及 Live Data 窗口中的快速标记编辑器来编辑各标记。
- ※ 可以编辑 DW UltraDev 中的数据格式，或者创建新的数据格式。

10.1 动态网页的基本编辑操作

我们可以在文档窗口或 Live Data 窗口中对网页进行基本的编辑操作，包括改变现有记录集的内容、替换掉为页面提供动态内容的数据源、改变页面中的服务器行为等。

10.1.1 改变记录集的内容

我们可以对现有记录集的内容进行修改以便满足 Web 应用更多的需求。例如，假设

我们要在页面中显示所有客户的电话号码，而现有的记录集中并没有相应的字段列出电话号码，于是就可以对记录集进行修改，在其中增加相应的字段。

要改变现有记录集的内容，首先在文档窗口或 Live Data 窗口中打开包含需要进行修改的记录集的页面，于是在 Data Bindings 面板或 Server Behaviors 面板的列表中就会列出该页面所使用的记录集。请在该记录集名上双击鼠标左键，于是就会打开相应的简单版本或高级版本的 Recordset 窗口。

根据自己的需要，在 Recordset 窗口作任意修改。如果满意了，可以单击其中的 Test 按钮查看更新后的记录集的内容，然后单击 OK 按钮关闭测试窗口。最后单击 OK 按钮关闭 Recordset 窗口，即可完成对现有记录集的修改。

10.1.2 替换和删除动态内容

即使已经在页面中增加了动态内容，我们还是可以对提供动态内容的数据源进行修改。例如，我们可以替换或者删除现有的动态内容。

若想替换掉现有的提供动态内容的数据源，可以这样做：

首先在文档窗口或 Live Data 窗口中打开包含需要进行编辑的动态内容的页面，这里在 Server Behaviors 面板中，通常会列出该页面中的所有服务器行为以及所有动态内容，如图 10-1 所示。

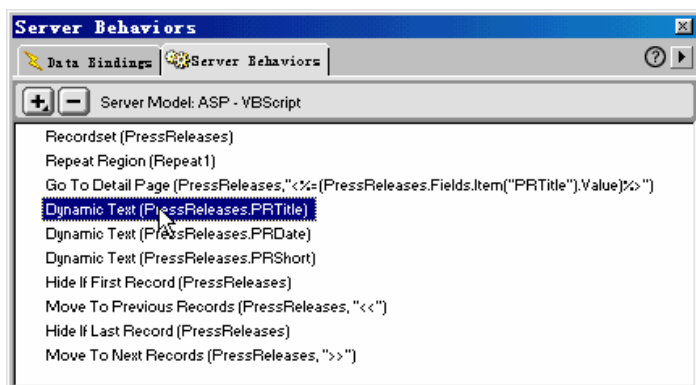


图 10-1
Server
Behaviors
面板中列
出的相关
信息

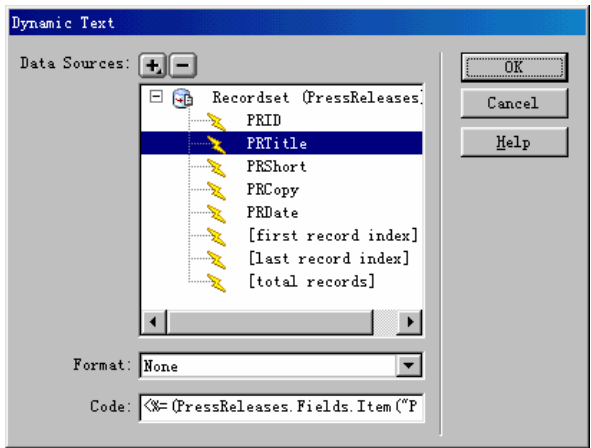
请在列表中双击需要修改的动态内容，于是就会打开相应的动态内容窗口。例如，图 10-2 所示的是双击动态文本后所出现的 Dynamic Text 窗口。用户双击不同的动态内容（如文本、图片等）时会打开不同的动态内容窗口，但是在各种窗口中都会有一个 Data Sources 域，其中会列出所有可用的数据源（当然也可以添加或者删除数据源）。请根据自己的需要选择另外一个数据源，然后单击 OK 按钮。于是就完成了修改动态内容的数据源的目标。

相对来说，删除页面中的动态内容就非常简单了。只需在页面中选择需要删除的动态内容，然后按下 Delete 键即可。

10.1.3 修改服务器行为

在往页面中添加了服务器行为之后，我们也可以删除它，或者修改它的属性。例如，如果页面中已经有一个 Repeat Region 服务器行为，它可以允许每页显示 4 条记录，现在我们想使每页可以显示更多的记录，如 10 条，则只需修改该服务器行为的属性即可。

图 10-2
Dynamic
Text 窗口



修改一个服务器行为的具体步骤如下：

首先在文档窗口或 Live Data 窗口中打开包含需要进行修改的服务器行为的页面，然后在 Server Behaviors 面板的列表中，双击需要修改的服务器行为，于是便会打开相应的服务器行为属性窗口，不同的服务器行为对应的属性窗口是不一样的。例如，图 10-3 所示的就是 Repeat Region 服务器行为的属性窗口。

图 10-3
Repeat
Region 服
务器行为
的属性窗
口



在服务器行为窗口中作出相应修改，然后单击 OK 按钮，即可使服务器行为具有了新的功能。

相对来说,删除某个页面中的服务器行为是非常简单的。只需在 Server Behaviors 面板的列表中,选择需要删除的服务器行为,然后单击按钮即可。

10.2 Live Data 环境下的编辑操作

DW UltraDev 最具特色的一项功能就是,可以允许用户在 Live Data 窗口中对实时的动态内容进行编辑,包括:

- ※ 使用 DW UltraDev 的标准设计工具调整页面布局。
- ※ 增加、编辑或删除动态内容。
- ※ 增加、编辑或删除服务器行为(当然也可以直接在 HTML 源代码检查器中编辑服务器端脚本)。

10.2.1 在 Live Data 窗口中编辑页面

读者可以使用 View / Live Data 菜单命令在文档窗口和 Live Data 窗口之间进行切换。如果一个动态页面在显示的时候还需要用户输入的数据,例如,用户在主页面中所选择的记录的 ID 号,则我们还必须使用 Live Data Settings 对话框为页面提供相应的数据。

在 Live Data 窗口中预览之前,我们必须将所有必需的文件(包括服务器端嵌入文件 SSI 和各相关文件,如图片等)都上传到服务器站点目录中。因为当 DW UltraDev 在 Live Data 窗口中打开一个动态页面时,它仅仅把该页面的一个临时拷贝上传到服务器中,而所有的相关文件并没有上传。要想上传文件,请选择 Site / Site Files 菜单命令打开 Site 窗口,选择正确的站点,然后在 Local Folder 面板中选择需要上传的文件,单击 Put 按钮即可将文件上传到远端站点上。

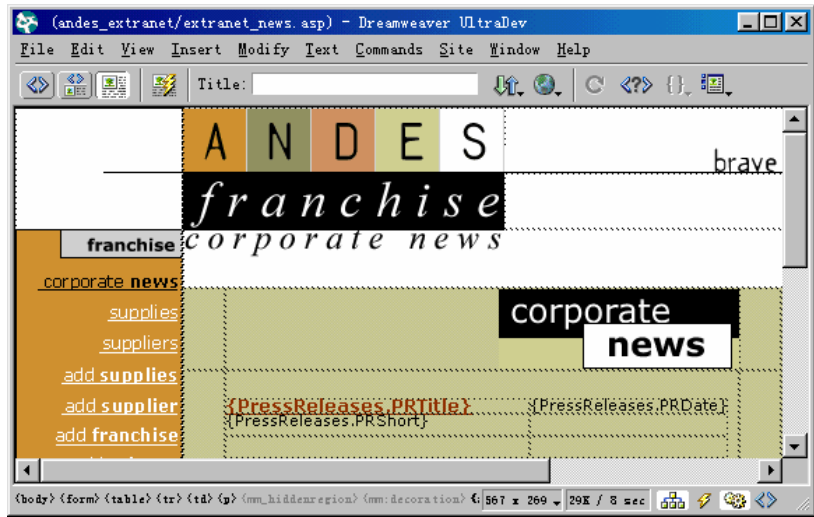
值得注意的是,Live Data 窗口也有一个缺点,就是所有的链接都无法工作。要想测试页面中的链接是否工作正常,必须在浏览器中使用 DW UltraDev 的预览功能。也许这是 DW UltraDev 一个不足的地方,希望在将来的版本中能够解决这个问题。

在 Live Data 窗口中编辑页面的具体步骤如下:

首先请在文档窗口中打开一个需要进行编辑的动态页面。在该页面中,所有的动态内容都是以占位符的形式出现。如图 10-4 所示。

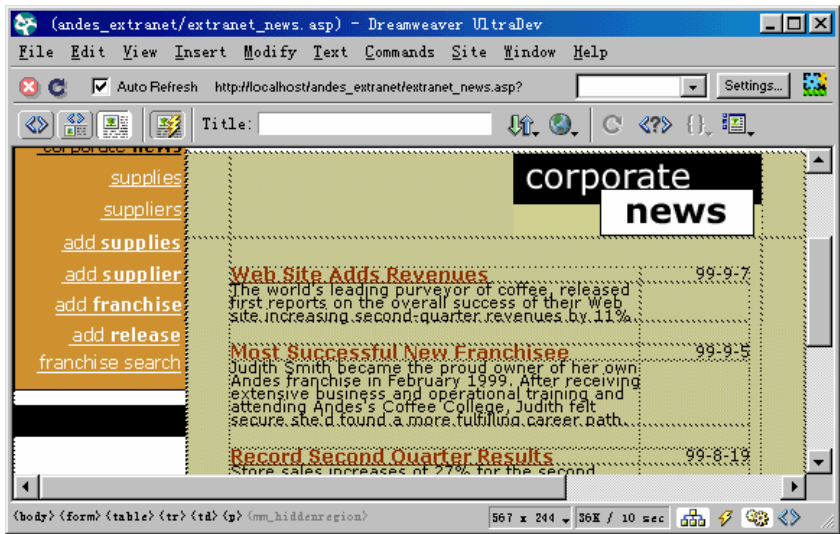
然后选择 View / Live Data 菜单命令将文档窗口切换到 Live Data 窗口。由于 DW UltraDev 在显示该页面及动态内容之前,需要运行该页面在服务器上的一个临时版本,因此,如果页面比较大的话,这个过程可能需要一段时间。如果想取消这个过程,可以单击 Live Data 窗口的工具栏最左边的停止按钮.

图 10-4
文档窗口
中的动态
页面



一段时间以后，在 Live Data 窗口中就会带有实际动态内容的页面，如图 10-5 所示。

图 10-5
Live Data
窗口中的
动态页面



如果你的动态页面的显示是如图 10-6 所示的样子，则表明系统已经自动为页面中的动态内容进行了加亮显示。

这些加亮显示往往会影响动态内容的显示，并且破坏我们对整个页面的整体认识，因此想查看页面的原始样子的话，请选择‘ View / Visual Aids / Invisible Elements ’ 菜单命令关闭该功能。

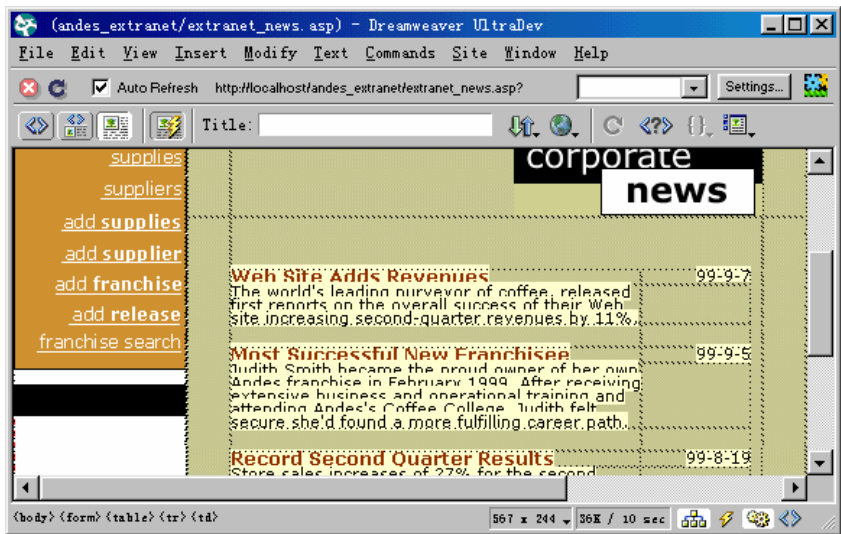


图 10-6
在 Live
Data 窗口
中加亮显
示的动态
内容

当然，加亮显示如果应用在文档窗口中的动态页面上，是有帮助的。它不仅可以帮助显示页面中的动态内容，使读者一目了然，而且也可以明确地标记出页面中使用的所有服务器行为。通常会在一个使用了服务器行为的区域的左上角显示一个灰底白字的方框，其中显示了服务器行为名，如图 10-7 所示（可以看出其中使用了三个服务器行为）。因此加亮显示功能要分场合正确地进行使用。

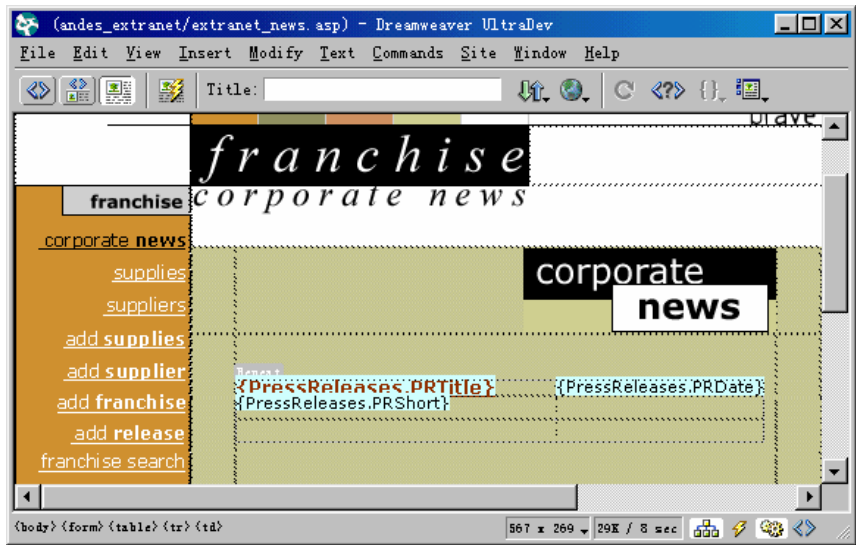


图 10-7
在文档窗
口中的加
亮显示

如果需要，请选择 Live Data 窗口中的工具栏上的 Auto refresh 复选框。这样，当我们对页面中的动态内容作任何修改时，该页面都会自动进行更新。当然，如果受各种条

件的限制和影响,数据库的链接速度比较慢,我们也可以不使用这个功能。

现在就可以对页面中的动态内容作任意的编辑操作了。例如,改变某一个动态文字的字体、颜色,移动另一个动态文字的位置,等等。这就和在普通的页面中进行编辑操作一样。

如果动态页面需要使用一个 HTML 表单通过 GET 提交方法传送过来的数据,则可以在 Live Data 窗口中的工具栏上的文本框中输入相应的数值,并单击刷新按钮,如图 10-8 所示。

图 10-8

Live Data
窗口中的
工具栏



值得注意的是,仅仅当读者在 Live Data Settings 窗口中设置了 GET 提交方法时,工具栏上才会出现文本输入框。稍后我们将介绍 Live Data Settings 窗口的设置方法。另外,在文本框中输入测试数据时要使用下面的格式:

```
name=value;
```

其中, name 是页面将要使用的变量名,而 value 是该变量中所保存的值。

除了在文本框中直接输入测试数据以外,我们也可以在 Live Data Settings 窗口中定义变量名和值,从而达到同样的目的。

最后,如果需要的话,单击刷新按钮即可看到修改后的结果。

10.2.2 使用 Live Data Settings 对话框

选择“View / Live Data Settings”菜单命令,或者在 Live Data 窗口中单击工具条上的按钮,均可打开 Live Data Settings 窗口,如图 10-9 所示。

单击 URL Request 域中的按钮,于是系统就会在列表中新建一个域,在其中的 Name 列中输入页面需要使用的变量名,使用 Tab 键切换到 Value 列,接着输入保存在该变量中的测试值。然后单击 OK 按钮。

在 Method 下拉列表中选择 HTML 表单的提交方法:POST 或 GET。在 Initialization Script 文本框中,插入相关的源代码,这些代码将会出现在页面的最前面。通常这些代码用于初始化一些 Session 变量。

如果要保存为当前页所做的设置,请选择 Save Settings for this Document 复选框。最后单击 OK 按钮即可。

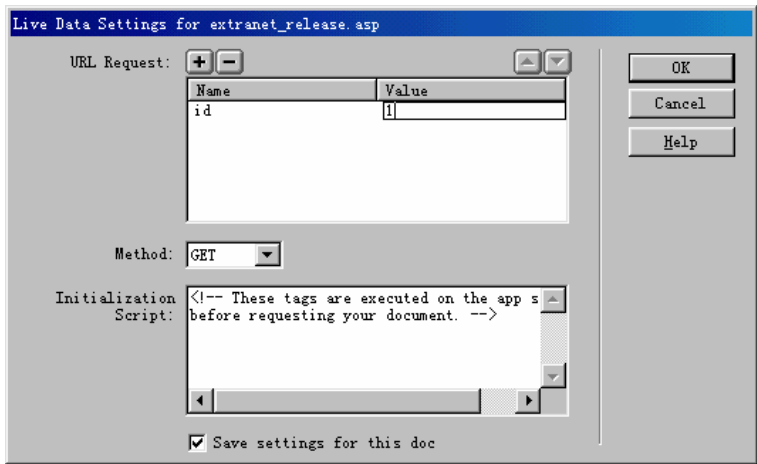


图 10-9
Live Data
Settings 窗
口

10.3 测试 Web 应用的链接

由于在 Live Data 窗口中无法测试页面中的链接是否工作正常，也就是说，我们无法在 Live Data 窗口中测试页面的交互过程。因此，必须在浏览器中使用 DW UltraDev 的 Preview in Browser 预览功能来实现对 Web 应用的链接的测试。

10.3.1 配置 DW UltraDev

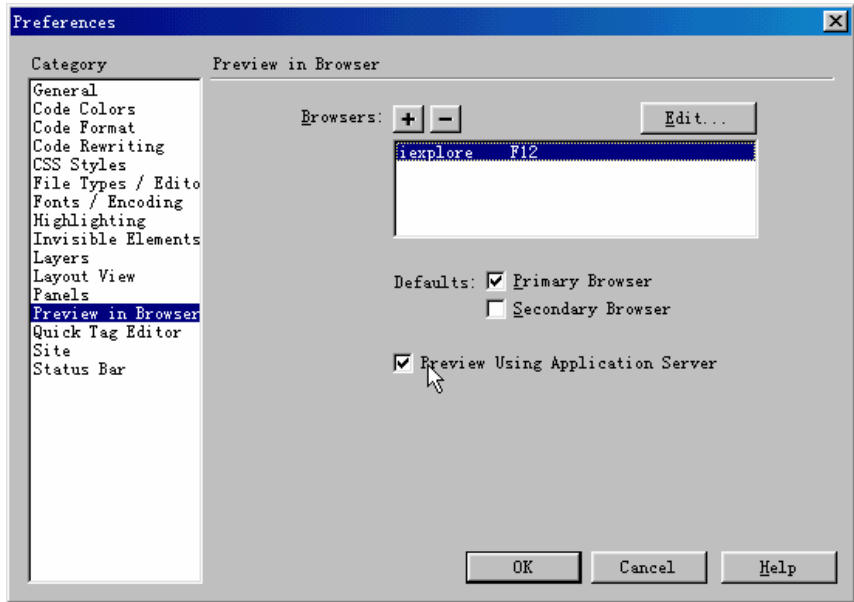
DW UltraDev 提供的在浏览器预览页面的功能是通过 Preview in Browser 这个命令来实现的。在缺省情况下，DW UltraDev 在浏览器中预览一个静态页面时会直接在本机找到该文件，然后在浏览器中创建一个临时的拷贝并显示出来。然而对于动态页面来说，所采取的方法就不一样。因为动态页面必须在服务器上运行，因此当 DW UltraDev 在浏览器中预览该页面之前必须运行位于服务器上的一个临时拷贝，当预览结束之后，DW UltraDev 会自动将该临时拷贝删除。

由于 DW UltraDev 是通过 Live Data 服务器来同真正的 Web 服务器打交道的，因此要想使用浏览器预览功能，必须对 DW UltraDev 做相应配置，以允许使用 Live Data 服务器来进行预览。

首先请选择“ Edit / Preferences ”菜单命令打开 Preferences 窗口，然后在 Category 列中选择 Preview in Browser 项，于是会出现如图 10-10 所示的属性设置窗口。请确保选择了 Preview Using Application Server 复选框。设置完成后单击 OK 按钮关闭 Preferences

窗口。

图 10-10
设置浏览器预览选项



10.3.2 在浏览器中进行预览

和在 Live Data 窗口中预览动态页面一样，在使用浏览器预览页面之前，我们也必须将所有必需的文件（包括服务器端嵌入文件 SSI 和各相关文件，如图片等）都上传到服务器站点目录中。因为 Preview in Browser 命令只会把动态页面的一个临时拷贝上传到服务器中，而所有的相关文件（如结果页面或细节页面）并没有上传。

然后在文档窗口或 Live Data 窗口中打开需要预览的页面，选择“File / Preview in Browser”菜单命令，或者按下 F12 快捷键，即可在浏览器中打开该页面，接下来就可以进行测试了。

在测试 Web 应用的时候，也要注意一定的策略。尽量按照 Web 应用的流程来进行测试。例如，如果想测试某一个页面如何处理从另一个页面的 HTML 表单中传送过来的数据，则应该先打开另一个页面，在 HTML 表单中填入适当的数据，然后提交。否则，测试页面无法接收到数据，就可能无法正常工作。

如果在测试过程中发现了问题（这也是我们进行测试的目的所在），则应该尽快在 DW UltraDev 中进行修改，然后重新在浏览器中进行测试，直到完全正确为止。

10.4 编辑源代码

在 DW UltraDev 中,可以直接使用 HTML 源代码检查器对页面的 HTML 源代码和服务器行为的服务器端脚本进行编辑。也可以不离开文档窗口或 Live Data 窗口,使用快速标记编辑器来编辑各标记。

若想使用快速标记编辑器,请先选择一个对象(如文本或标记等),然后按下 Ctrl+T 组合键,于是就会打开快速标记编辑器窗口。例如,图 10-11 所示的就是选择了页面中的某一个动态文本后所打开的快速标记编辑器窗口。

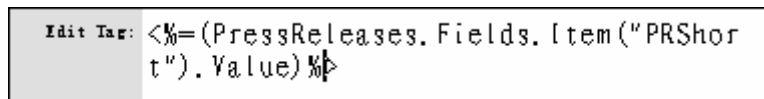


图 10-11
快速标记
编辑器窗
口

至于 HTML 源代码检查器和快速标记编辑器的使用方法,我们在第 5 章中已经作了较为详细地介绍,在此不再赘述。

值得注意的是,当用户在 HTML 源代码检查器窗口中对页面中的静态内容作了修改后,并不会立即在 Live Data 窗口中体现出来,而需要用户在 Live Data 窗口中单击一下,即激活 Live Data 窗口后才可以;同样,当用户在 HTML 源代码检查器窗口中对页面中的动态内容作了修改后,也不会立即在 Live Data 窗口中体现出来,而需要单击 Live Data 窗口中的刷新按钮  才可以。

10.5 编辑和创建数据格式

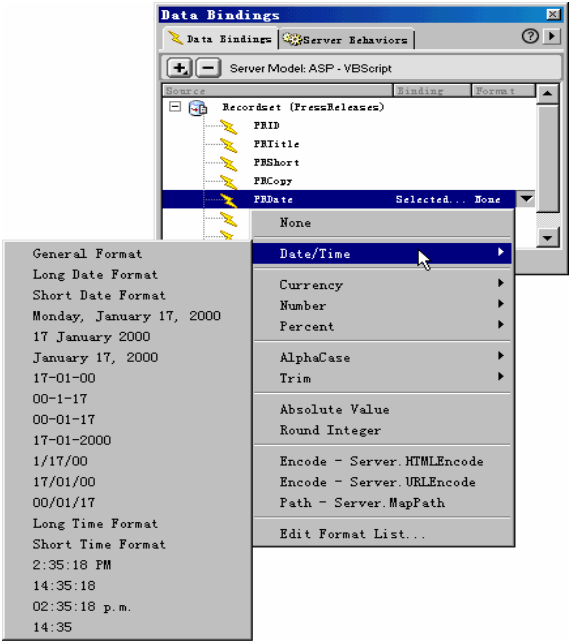
这里所说的数据格式,是指应用于页面中的动态文本上的显示方式。其作用是使数据以更友好的方式显示给用户。例如,你可以在记录集中以“ March 29, 2001 ”的方式显示日期,但对于中国人来说,“ 2001-3-29 ”的显示方式显然更为习惯。在 DW UltraDev 中已经为我们定义了许多现成的数据格式,我们即可以直接使用它,也可以根据自己的需要对现有的数据格式进行修改,或者创建新的数据格式。

10.5.1 编辑数据格式

首先请打开一个包含了动态文本的页面，然后选择需要设置数据格式的动态文本（如果是在文档窗口中，请选择该动态文本所对应的占位符 如果是在 Live Data 窗口中，请直接选择它）。

如果 Data Bindings 面板没有打开，请选择“ Window / Data Bindings ”菜单命令，或者单击启动面板上的 Data Bindings 按钮打开它。这时在列表中与动态文本相对应的字段的 Format 列中会出现相应的值。单击其中的▼按钮可以打开一个菜单，其中列出了 DW UltraDev 已经定义好的所有现成的数据格式。我们可以选择其中的数据格式直接应用到页面中，例如图 10-12 中所示的就是设置日期格式的子菜单。

图 10-12
设置日期
格式



如果对现有的数据格式不满意，也可以对其进行编辑，以满足自己的需要。在单击▼按钮弹出的菜单中选择“ Edit Format List ”菜单命令，于是便会出现如图 10-13 所示的 Edit Format List 对话框。

选择某一个需要进行编辑修改的数据格式，然后单击 Edit 按钮，便会打开相应的数据格式编辑窗口。图 10-14 所示的就是一个货币类型的数据格式编辑窗口。在该窗口中根据自己的需要做适当的修改，然后单击 OK 按钮。

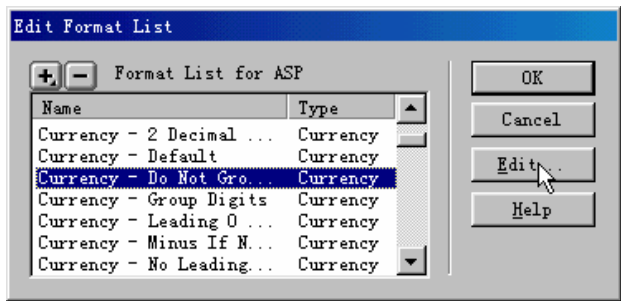


图 10-13

Edit
Format
List 对话框

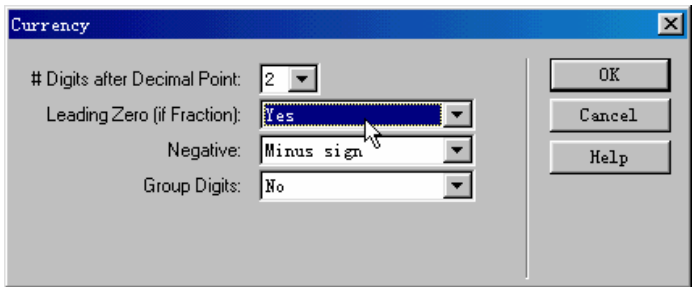


图 10-14

货币类型
的数据格
式编辑窗
口



并不是所有现有的数据格式都可以进行编辑，也有一些并不允许用户进行定制。当用户在图 10-13 所示的 Edit Format List 对话框中选择了这样的数据格式（如日期型数据格式），然后单击 Edit 按钮时，便会弹出如图 10-15 所示的警告框，表示无法对该数据格式进行编辑。



图 10-15

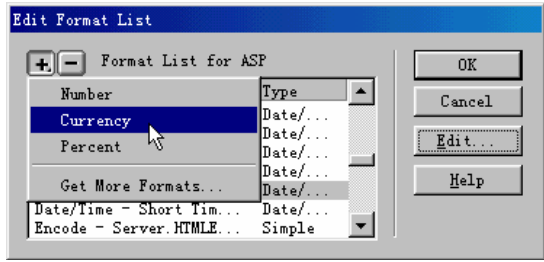
无法对所
选数据格
式进行编
辑

最后单击 OK 按钮关闭 Edit Format List 对话框，即可完成对数据格式的编辑工作。

10.5.2 创建新的数据格式

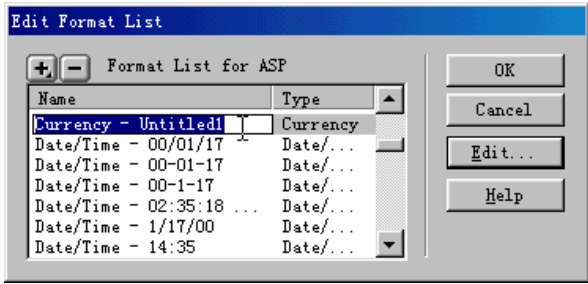
如果需要创建新的数据格式，也很简单。只需在图 10-13 所示的 Edit Format List 对话框中单击其中的  按钮，便会弹出一个下拉菜单，如图 10-16 所示。

图 10-16
创建新的
数据格式



该菜单中的上半部分列出了用户可以创建的三种数据格式：Number（数字）、Currency（货币）和 Percent（百分比）。选择其中的任何一种，都会打开一个类似于图 10-14 所示的格式编辑窗口，在其中根据自己的需要做任意修改，然后单击 OK 按钮，于是在 Edit Format List 对话框中就会新增加一项，如图 10-17 所示。在 Name 列中为新增的数据格式命名，然后单击 OK 按钮即可完成数据格式的创建过程。

图 10-17
为新增的
数据格式
命名



如果需要更多的数据格式，可以在菜单中选择“Get More Formats”菜单命令，系统就会打开浏览器窗口连接到 Macromedia 公司的网站，去获取其他的数据格式。

第 三 部 分

DW UltraDev 4 应用实例

- ◆ 第 11 章 Web 应用综合示例
 - ◆ 第 12 章 创建留言簿
 - ◆ 第 13 章 创建查询系统
-

第 11 章 Web 应用综合示例

本章概要：

- ☒ 开发 Web 应用前的准备工作，如创建 DSN、定义站点、创建数据库链接等
- ☒ 在页面中增加动态内容，包括定义记录集、增加动态内容、浏览记录等
- ☒ 增强动态页面的功能，包括增强浏览功能、定义过滤记录集、增加新记录等

在第 3 章中我们已经介绍过创建一个简单的 Web 应用的过程，现在请你回过头去看一下第 3 章的内容，一定会觉得非常简单吧！本章我们对本书第三部分的内容进行总结，介绍一个综合示例的完整开发过程，让你进一步体验一下 DW UltraDev 4 的强大功能。

11.1 开发 Web 应用前的准备工作

为简单起见，本章我们介绍 Web 应用示例时假设读者的 DW UltraDev 和 Web 服务器安装在同一台计算机上。如果读者使用的操作系统是 Windows 9x 或 Windows NT Workstation，则必须安装 Microsoft Personal Web Server 4.0；如果读者使用的操作系统是 Windows NT Server 或 Windows 2000，则必须安装 Microsoft IIS 4.0。

本章介绍的 Web 应用示例的站点文件及子目录都存放在了本书所附光盘的 andes_extranet 目录中。在开始动手学习本章示例之前，请不要忘记把整个 andes_extranet 目录都复制到 WWW 服务的根目录中。例如，如果 WWW 服务安装到了 C 盘，则请将 andes_extranet 目录复制到 C:\Inetpub\wwwroot 目录下。

在开始介绍本章的 Web 应用之前，我们先说明一下该 Web 应用的背景：本例所创建的站点是一个名为“ andes ”的咖啡公司。通过这个站点，我们可以在 Web 页面上查看该公司的产品目录、各授权代理商情况、相关产品的出版信息、各产品详细信息和产品供应商的情况，本示例的重点在于介绍如何查看相关产品的出版信息。只要读者掌握

了其方法，相信也可以依照同样的方法来查看数据库中的其他信息。

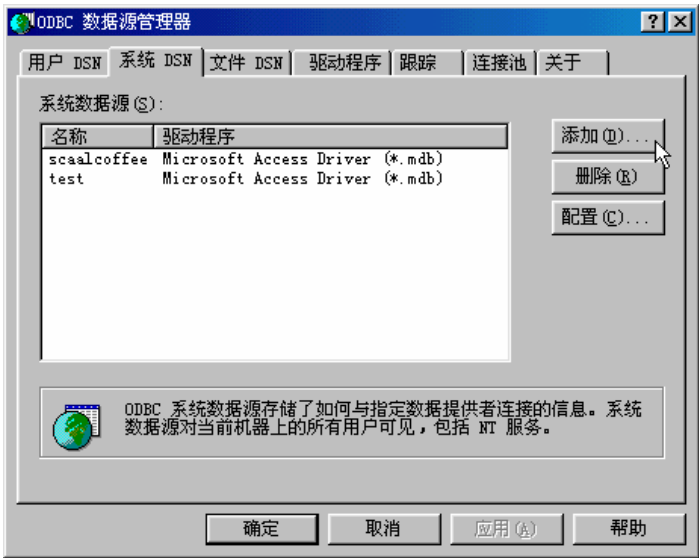
下面我们可以开始着手进行 Web 应用开发前的一些准备工作了。主要包括：创建 DSN、定义站点和创建数据库链接。

11.1.1 创建 DSN

数据库建立好之后，要设定系统的 DSN(数据源名称)，才能让网页可以知道数据库所在的位置以及数据库相关的属性。使用 DSN 的好处还有，如果移动数据库文件的位置，或是换成别种类型的数据库，只要重新设定 DSN 就可以了，不需要去修改原来使用的程序。由于本章的示例使用的是 MS Access 数据库，因此需要先创建一个使用 MS Access Driver 的 ODBC 系统 DSN。步骤如下：

④ 在 Windows 操作系统中，创建 DSN 需要借助“ODBC 数据源管理器”。通过操作系统的“控制面板”就可以打开“ODBC 数据源管理器”窗口(在 Windows 9x、NT 中，选择“开始 / 设置 / 控制面板”，然后双击“ODBC 数据源”图标即可；在 Windows 2000 中，选择“开始 / 程序 / 管理工具 / 数据源”)，如图 11-1 所示。

图 11-1
“ODBC
数据源管
理器”窗口



② 单击“系统 DSN”标签切换到“系统 DSN”面板，然后单击“添加”按钮，出现如图 11-2 所示的“创建新数据源”窗口。



图 11-2
“创建新
数据源”窗
口

3 在窗口的列表中选择“Microsoft Access Driver(*.mdb)”项，然后单击“完成”按钮，于是出现如图 11-3 所示的“ODBC Microsoft Access 安装”窗口。在“数据源名”文本框中输入“andescoffee”，使“描述”域保持为空。然后单击“选取”按钮，出现如图 11-4 所示的“选定数据库”窗口。找到 andescoffee.mdb 数据库文件所在的路径，如 C:\inetpub\wwwroot\andes_extranet\andescoffee.mdb，并选定它，然后单击“确定”按钮。

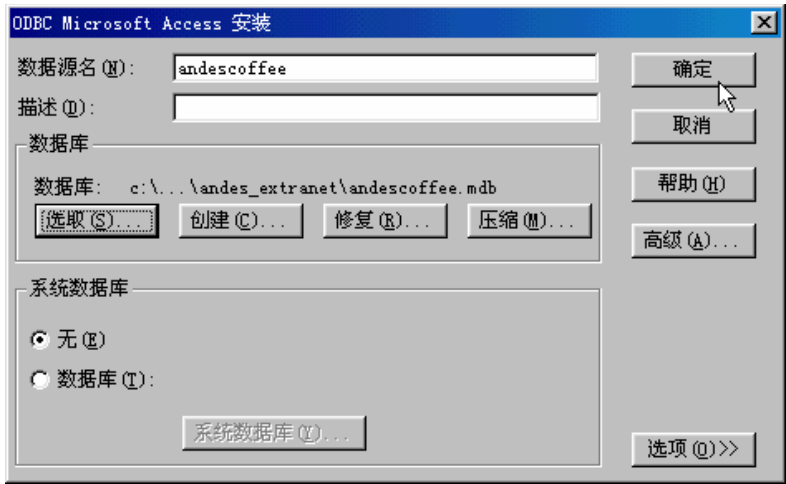
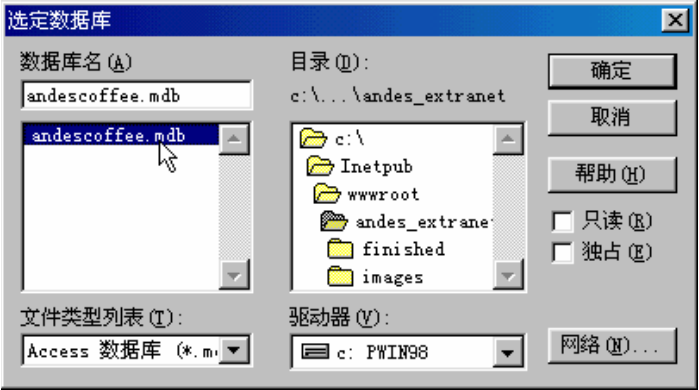


图 11-3
“ODBC
Microsoft
Access 安
装”窗口

4 再次单击“确定”按钮。这时，在“ODBC 数据源管理器”窗口的列表中就会出现刚刚创建的 DSN 的名字 andescoffee。最后单击“确定”按钮完成 DSN 的创建过程。

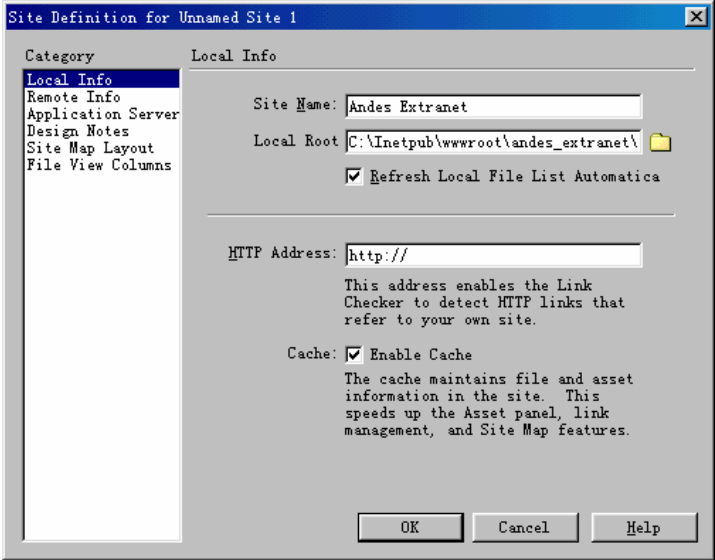
图 11-4
“选定数据库”窗口



11.1.2 定义站点

接下来启动 DW UltraDev，并打开 Site 站点管理器窗口(选择 Window / Site Files” 菜单命令，或者在启动面板中单击 Site 按钮，或者按下快捷键 F5)。然后选择 Site / New Site” 菜单命令打开 Site Definition 窗口。单击窗口左边的 Category 列表中的 Local Info 打开 Local Info 面板，如图 11-5 所示。在 Site Name 文本框中输入 Andes Extranet，在 Local Root 域中输入本地站点的根目录，C:\inetpub\wwwroot\andes_extranet，或者使用  按钮定位到该目录。

图 11-5
Site
Definition
窗口



在 Category 列表中单击 Remote Info，切换到 Remote Info 面板，如图 11-6 所示。由于本例中的本地站点和远程站点位于同一个目录下，因此不需要进行文件传输，因此请保持 Access 下拉列表中的选项为 None。

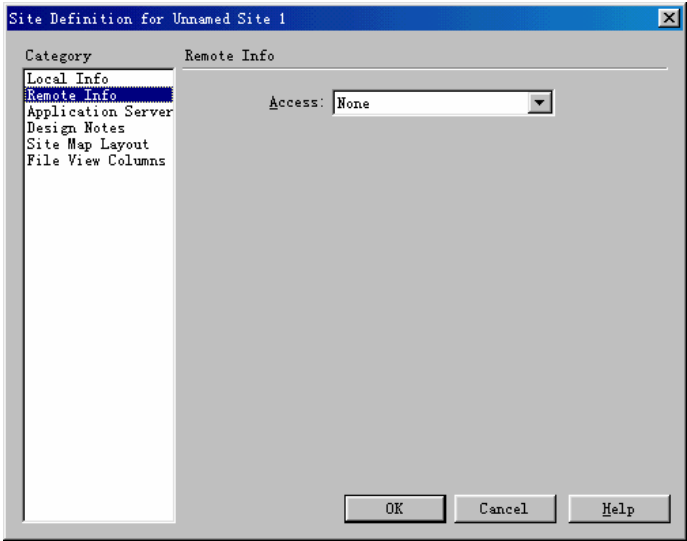


图 11-6
Remote
Info 面板

在 Category 列表中单击 Application Server 切换到 Application Server 面板，如图 11-7 所示。在 Server Model 下拉列表中选择 ASP 2.0 项，在 Default Scripting 下拉列表中选择 VBScript 或 JavaScript，设置 Default Page extension 下拉列表的值为 .asp。然后在 Access 下拉列表中选择 Local/Network，在 Remote Folder 中输入远程站点的根目录，C:\Inetpub\wwwroot\andes_extranet，或者使用  按钮定位到该目录。请注意本例中的远程站点和本地站点位于于同一个目录下。最后在 URL Prefix 文本框中输入 http://localhost/andes_extranet/。

设置完成后单击 OK 按钮关闭 Site Definition 窗口，这时可能会出现如图 11-8 所示的警告窗口，表示 DW UltraDev 将为这个新站点创建一个缓冲区以提高站点的性能，单击 OK 按钮创建它。于是站点定义完成。

11.1.3 创建数据库链接

接下来需要创建数据库链接了。

在 DW UltraDev 文档窗口中选择 Modify / Connections 菜单命令，打开 Connections 窗口，然后单击 New 按钮，在弹出的快捷菜单中选择 Data Source Name(DSN) 菜单命令，打开 Data Source Name(DSN) 窗口。设置 Connection Name 文本框的值为 AndesExtranet，然后在 Data Source Name(DSN) 下拉列表(该列表中列出了系统中的

所有 ODBC DSN) 中选择 andescoffee, 并保持 Username 和 Password 这两个文本框的值为空。如图 11-9 所示。

图 11-7
Application
Server 面板

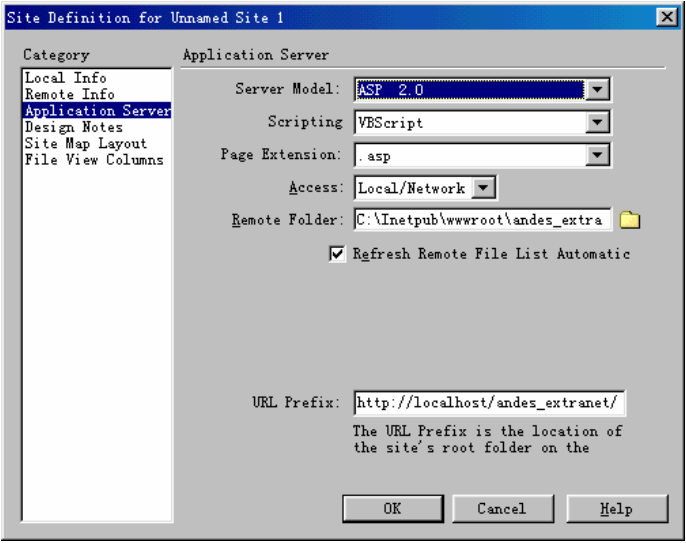


图 11-8
是否创建
缓冲区

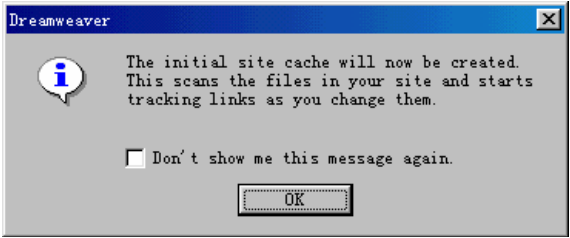
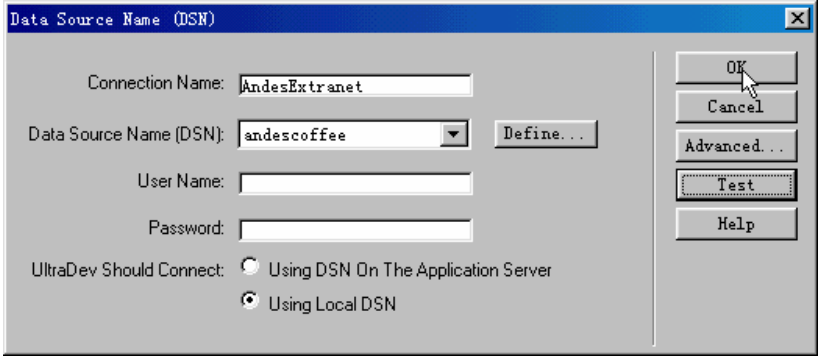


图 11-9
创建
AndesExtr-
anet 数据
库连接



单击 Test 按钮对此数据库链接进行测试。如果数据库链接创建成功,则会弹出一个对话框,其中显示“ Connection was made successfully. ”的提示信息,如图 11-10 所示。

单击“确定”按钮关闭提示窗口，然后单击 OK 按钮保存所创建的数据库链接。于是在 Connections 窗口的列表中就会显示出刚才创建的数据库链接名。

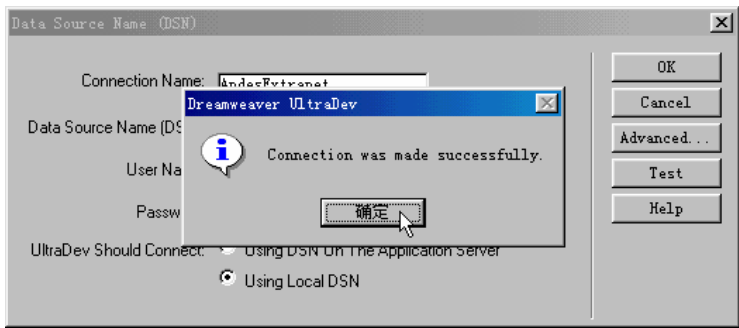


图 11-10
测试数据
库链接

由于光盘上的文件都是只读的，因此在复制完成后，不要忘了把 Inetpub\wwwroot\andes_extranet 目录及其子目录下的所有文件的只读属性去掉，而设置为可以存档。常用的修改文件属性的方法是使用系统的“资源管理器”，首先选择该目录下的所有文件，然后单击鼠标右键，在弹出的快捷菜单中选择“属性”菜单命令，于是出现如图 11-11 所示的“属性”窗口。在其中取消对“只读”复选框的选择，并选择上“存档”复选框。最后单击“确定”按钮即可取消文件的只读属性。

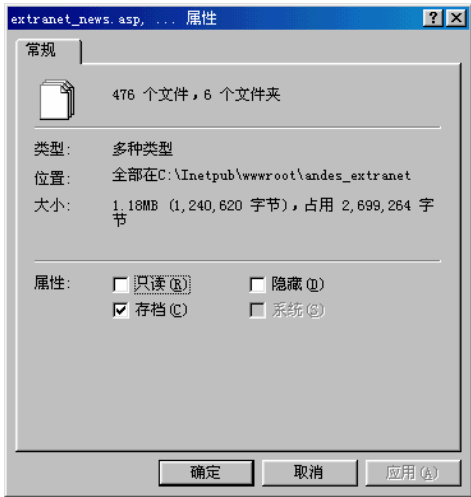


图 11-11
取消文件
的只读属
性

实际上，在 DW UltraDev 中也可以取消文件的只读属性。切换到 Site 窗口中，确保当前打开的站点是前面已经定义过的 Andes Extranet，然后将控制焦点放在 Site 窗口右边的视图中，并按下 Ctrl+A 快捷键选择所有的文件及目录，单击鼠标右键，在弹出的快捷菜单中选择 Turn Off Read Only 菜单项，于是所选文件的只读属性便取消了，现在我们就可以对这些文件进行修改并增加新的记录集了。

11.2 在页面中增加动态内容

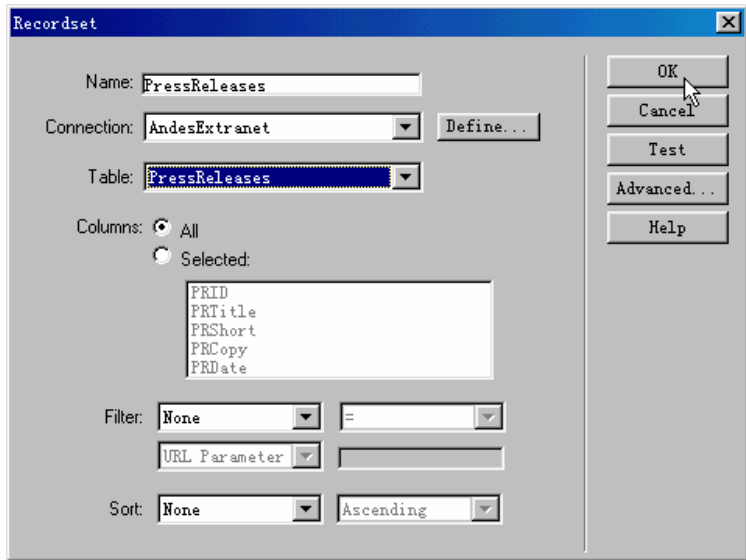
通常，程序开发者和网页设计者结合在一起进行 Web 应用的开发。当然，这两者也可以是同一个人来完成，这也是比较普遍的情况。网页设计者把网页设计好以后就交给程序开发者，后者就在前者设计的静态页面中增加一些动态内容，实现 Web 应用的开发。在本示例中，为了节省篇幅，我们在站点文件中已经提供了设计好的静态页面，因此下面的重点在于介绍如何在这些静态页面中增加动态内容。

11.2.1 为页面增加一个记录集

在 Site 窗口中，确保当前打开的站点是前面已经定义过的 Andes Extranet，然后在文件列表中双击 extranet_news.asp 文件，在文档窗口中打开它。

在 Data Bindings 面板中单击  按钮，然后在出现的菜单中选择 Recordset (Query) 菜单项，于是出现简单版本的 Recordset 窗口(如果出现的是高级版本的 Recordset 窗口，请单击 Simple 按钮切换到简单版本的 Recordset 窗口)。在 Name 文本框中输入 PressReleases，然后在 Connection 下拉列表中选择 AndesExtranet，在 Table 下拉列表(其中显示的是数据库中的所有表名)中，选择 PressReleases 项，并在 Columns 域中，选择 All 单选钮选中所有字段，如图 11-12 所示。

图 11-12
设置后的
Recordset
窗口



设置完成后，单击 Test 按钮即可对新建的记录集进行测试，这时应该出现如图 11-13 所示的测试窗口，其中显示了记录集中的所有记录。

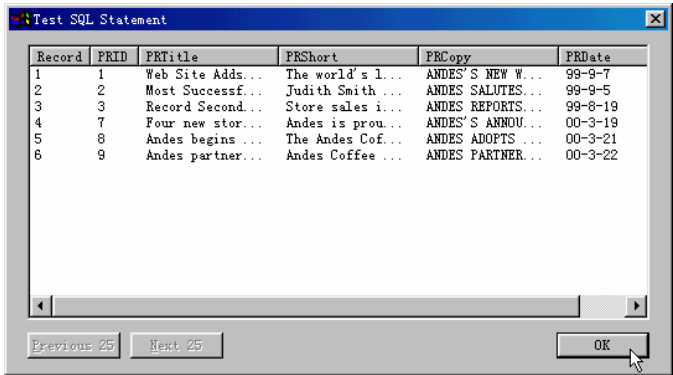


图 11-13
记录集测试结果

单击 OK 按钮关闭测试结果窗口，然后可以在 Recordset 窗口中单击 Advanced 按钮切换到高级版本的 Recordset 窗口，这时可以在窗口的 SQL 编辑域中看到系统自动为我们添加的 SQL 语句：SELECT * FROM PressReleases。如图 11-14 所示。单击 Simple 按钮返回到简单版本，然后单击 OK 按钮关闭 Recordset 窗口，从而将该记录集增加到页面中。

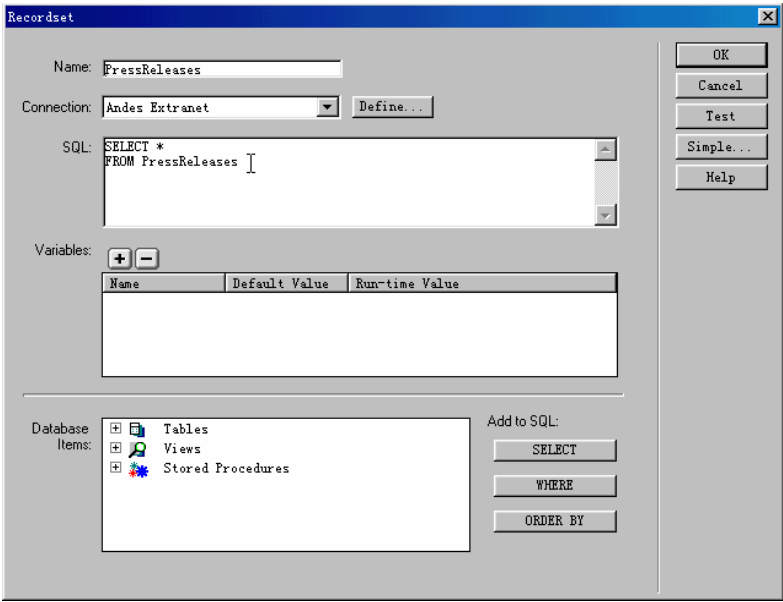


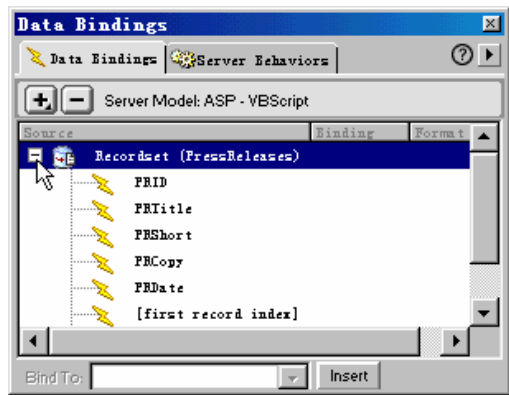
图 11-14
观看高级版本 Recordset 窗口中的 SQL 语句

11.2.2 在页面中增加动态数据

在创建了记录集以后，我们就可以使用其中的各个字段作为页面动态内容的数据源。下面我们将用数据库中的动态信息来替换页面中的静态内容。

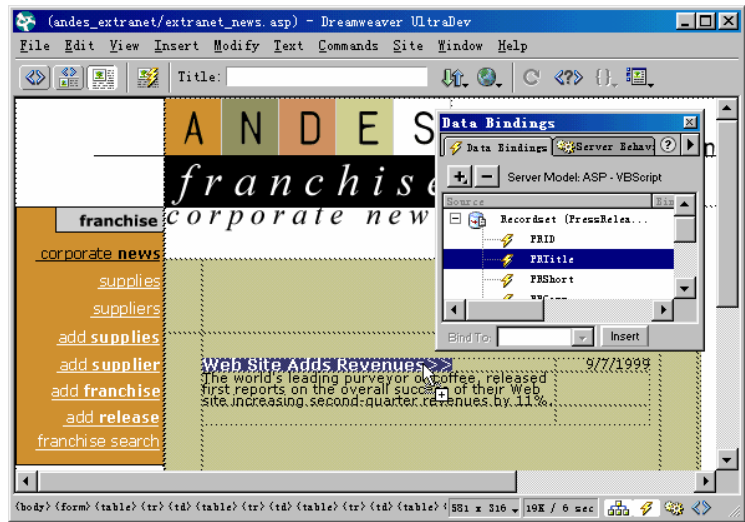
确保 Data Bindings 窗口是打开的，并且其中列出了刚才添加的记录集名，单击旁边的加号展开记录集树形结构，显示出该记录集所包含的所有字段，如图 11-15 所示。

图 11-15
Data Bindings 窗口



在 extranet_news.asp 文档窗口中选取中间表格中的“Web Site Adds Revenues >>”文字，然后在 Data Binding 面板中选择 PRTitle 项，并单击 Insert 按钮，或者将 PRTitle 项拖放到文档窗口中选中的“Web Site Adds Revenues >>”文字上，如图 11-16 所示。

图 11-16
将字段拖放到页面文字上



这时“ Release title ”文字将被占位符(此时为“ {PressReleases.PRTitle} ”)所代替,如图 11-17 所示。同时在 HTML 源文件中也增加了相应的服务器端脚本。当服务器运行该页面时,占位符将被记录集中的实际数据所取代。

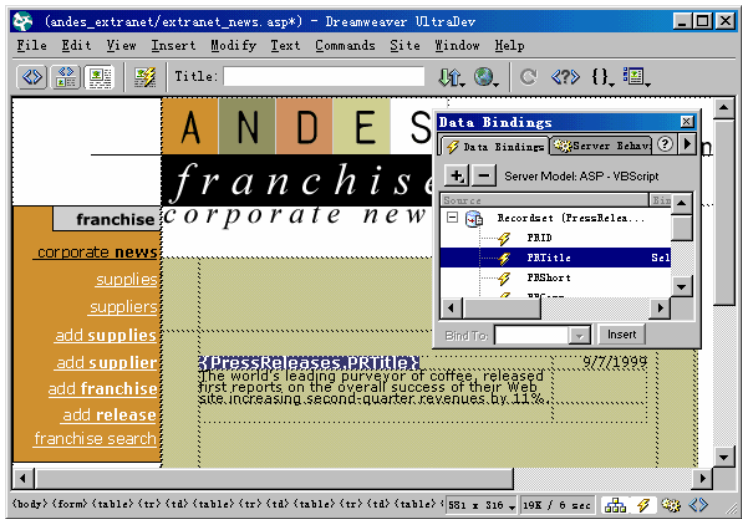


图 11-17
文档中的
文字被占
位符代替

如果要查看 DW UltraDev 自动为我们产生的代码,请按下 F10 快捷键,或者选择“ Window / Code Inspector ”菜单命令,或者在启动面板中单击 Code Inspector 按钮打开 HTML 源代码检查器,或者单击文档窗口工具栏上的 Show Code View 按钮  以及 Show Code and Design View 按钮  在文档窗口中打开代码视图,其中加粗的代码就是 DW UltraDev 自动产生的 ASP 代码,如图 11-18 所示。

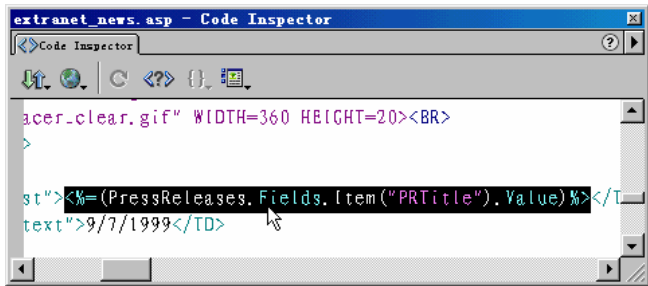


图 11-18
系统自动
添加的
ASP 代码

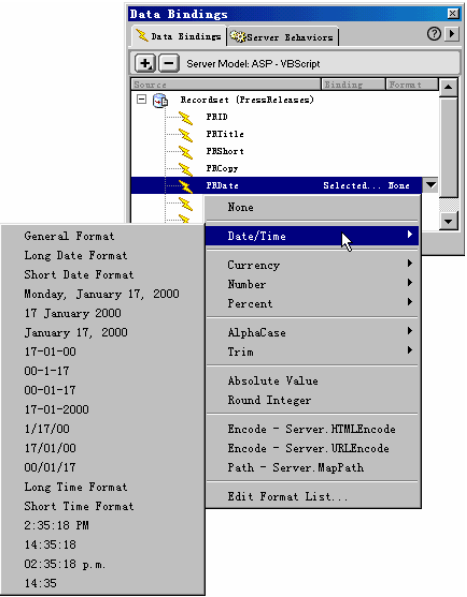
接下来,我们将采取同样的步骤把出版物的概况和出版时间增加到页面中。在页面中间的表格中,选取文字“ The world’s leading purveyor of coffee, released first reports on the overall success of their Web site increasing second-quarter revenues by 11% ”,然后在 Data Bindings 面板中选择 PRShort 项,并单击 Insert 按钮,或者将 PRShort 项拖放到文档窗口中选中的文字上,则原来的文字也被占位符所代替。同样,选取文字“ 9/7/1999 ”,然后

在 Data Binding 面板中选择 PRDate 项,并单击 **Insert** 按钮,或者将 PRDate 项拖放到文档窗口中选中的文字上,则原来的文字也被占位符所代替。

在把 PRDate 字段添加到页面中之后,在 Data Bindings 面板中的 PRDate 项旁边会显示出一些相关信息,如日期的显示格式。缺省值为 None,单击其中的  按钮可以打开一个菜单,选择‘ Date/Time / January 17,2000 ’菜单项,如图 11-19 所示。

由于这里设置的是服务器端的时间显示方式,因此不会受到客户端的影响。

图 11-19
设置日期
格式



11.2.3 在 Live Data 模式下编辑页面

接下来就可以使用 Live Data 窗口来预览动态页面中的实时数据了。选择 View / Live Data ”菜单命令,则 Live Data 窗口将出现在文档窗口中,并且页面中的三个占位符都被记录集中的实际数据所代替,如图 11-20 所示。此时,我们还可以在 Live Data 窗口中调整页面布局,增加或删除动态内容,甚至改变页面的服务器行为。

确保 Auto Refresh 复选框处于允许状态。这样,当我们对页面上的服务器脚本作了任何修改时,系统都会自动更新 Live Data 窗口中的动态内容。

Live Data 窗口不仅具有预览功能,而且还可以对页面进行编辑。下面我们对 Live Data 窗口中显示的内容作一些简单的编辑。

在网页中选择显示的动态数据‘ Web Site Adds Revenues ’,然后在属性面板中单击颜色选择器并选择另外一种颜色(如红色),如图 11-21 所示。则所选择的文字将变成红

色。

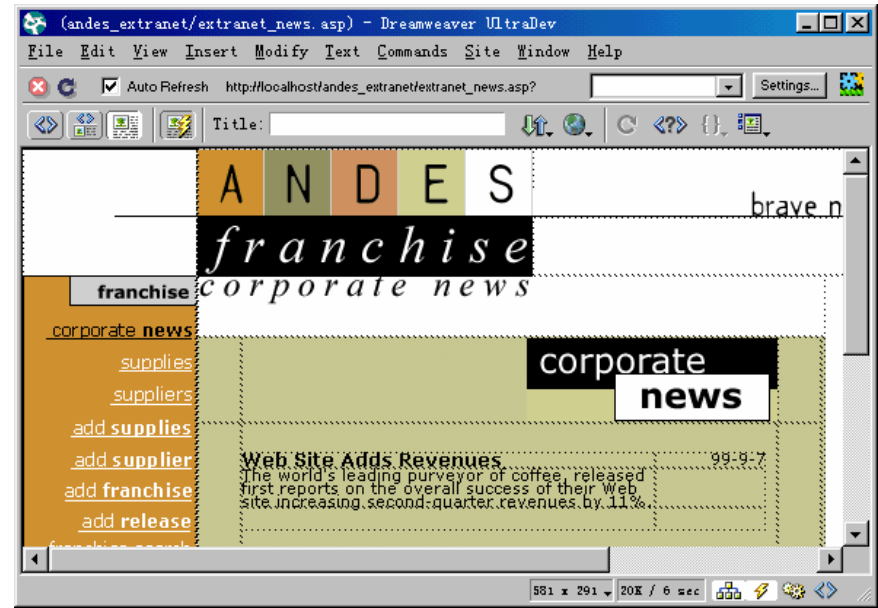


图 11-20
在 Live
Data 窗口
中预览动
态内容

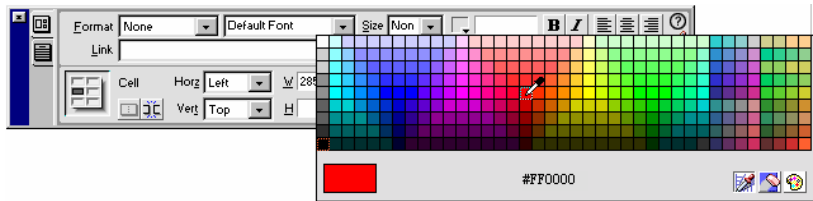


图 11-21
选择另外
一种颜色

接下来选择文字“ September 7,1999 ”，然后将其拖放到文档窗口中的其他位置，则不仅该动态数据的显示位置会发生变化，相应的 ASP 代码位置也会发生变化。最后将该文字移回到原来的位置。

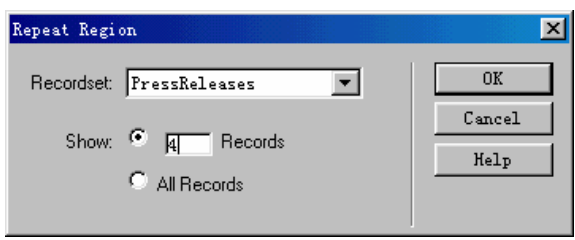
11.2.4 在页面中显示多个记录

使用 Repeat Region 服务器行为我们可以在 DW UltraDev 中实现在同一页面中显示多个记录的功能。下面我们就将使用它在同一个页面中同时显示 4 条出版记录。当我们增加 Repeat Region 服务器行为的时候，所有的服务器端脚本都是自动产生的。

首先请选择 View / Live Data ”菜单命令将 Live Data 窗口切换到文档窗口，然后单击中间表格中三个占位符中的任何一个，选择‘ Modify / Table / Select Table ”菜单命令选取整个表格。

接下来打开或者切换到 Server Behaviors 面板，单击其中的  按钮，并在出现的菜单中选择 Repeat Region 菜单项，于是出现如图 11-22 所示的 Repeat Region 窗口。

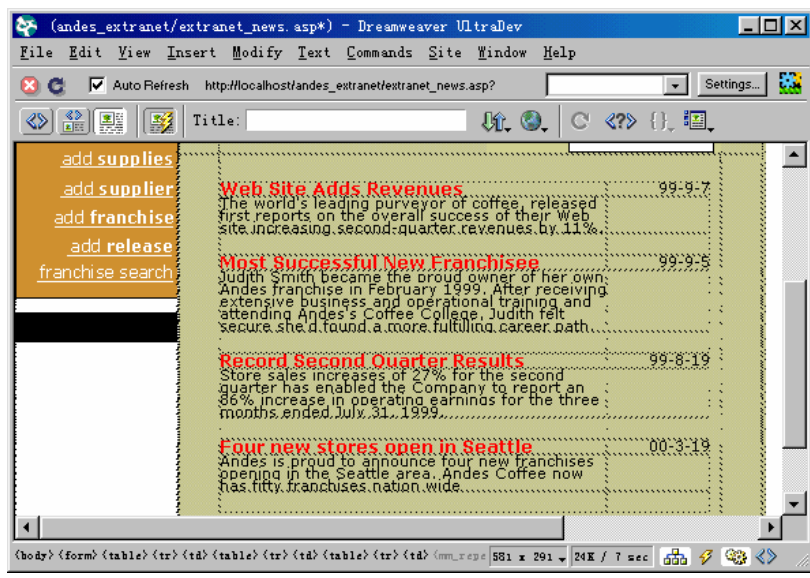
图 11-22
Repeat
Region 窗
口



确保在 Recordset 域中选择了 PressReleases 记录集，然后选取带编辑框的单选钮，在其中输入数字 4，并单击 OK 按钮。

现在再次将文档窗口切换到 Live Data 窗口，读者可以看到页面中把前 4 条记录都显示出来了，如图 11-23 所示。

图 11-23
在同一页
面中显示
多条记录



11.2.5 浏览记录

如果数据库中返回的记录有多个，则实现在这些记录间进行浏览的功能是非常重要的。用户应该能够很轻松地所有记录之间来回移动进行浏览。由于在上面的示例中我们每次只显示 4 条记录，因此需要增加一些基本的浏览链接，以使用户能够浏览前 4 条记录或后 4 条记录。

将输入光标放置在 extranet_news.asp 页面底部的版权信息上方，然后增加下面的字符串“ << >> ”，如图 11-24 所示。

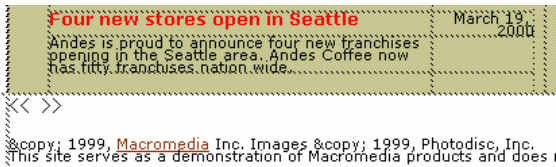


图 11-24
增加浏览
链接

双击字符“ << ”以选中它，打开或者切换到 Server Behaviors 面板，单击其中的  按钮，并在出现的菜单中选择“ Move To Record / Move To Previous Record ”菜单项，于是出现如图 11-25 所示的窗口，请确定 Link 域中所选择的文本是“ << ”，Recordset 域中选择的是 PressReleases 项，然后单击 OK 按钮。

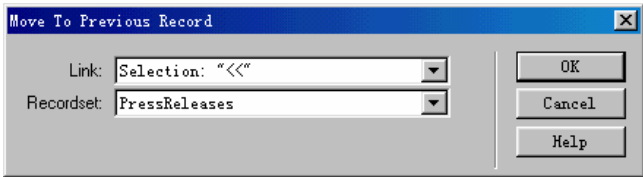


图 11-25
Move To
Previous
Record 窗
口

按照同样的方法为文本“ >> ”增加浏览链接，只是此时要选择“ Move To Record / Move To Next Record ”菜单项。

保存所做的修改，并对站点进行测试。这里的测试在 Live Data 窗口中体现不出来，必须要使用浏览器才能进行测试。因此请选择“ File / Preview In Browser ”菜单命令或者按下快捷键 F12 在浏览器中打开当前文档。

如果出现图 11-26 所示的提示窗口，表明我们虽然已经为站点 Andes Extranet 定义了应用服务器，但却没有允许使用该应用服务器来进行预览，因此要单击 Yes 按钮准备启动该功能。于是会出现如图 11-27 所示的 Preferences 窗口，选择其中的 Preview Using Application Server 复选框，然后单击 OK 按钮，即可打开浏览器。

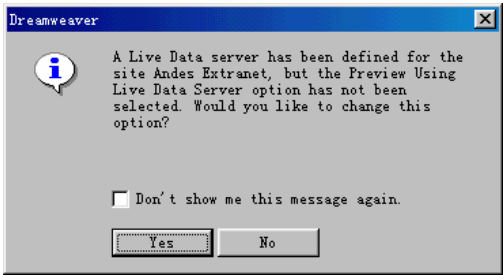
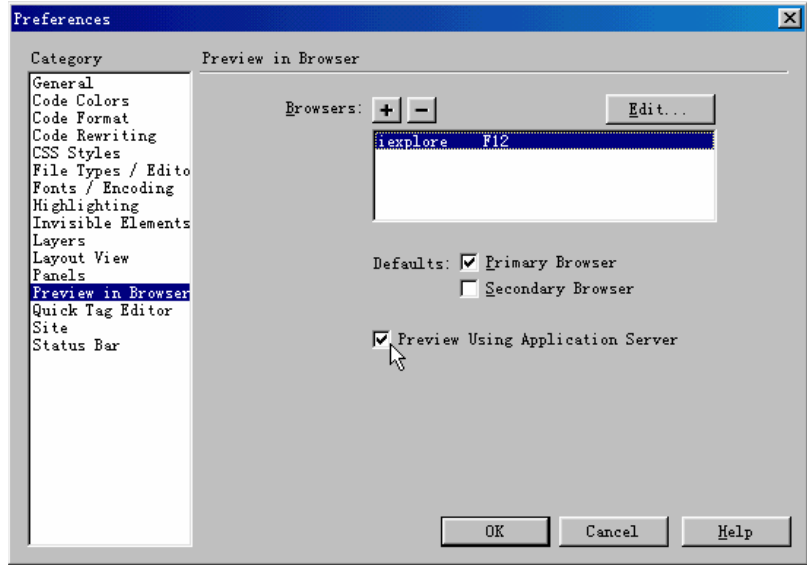


图 11-26
DW
UltraDev
提示窗口

图 11-27
Preferences
窗口



接下来就可以在浏览器中对新增的浏览链接“<<”和“>>”进行测试了。

11.3 进一步增加动态页面的功能

11.3.1 增强浏览功能

在上面的测试过程中，细心的读者会发现这样一个问题：当浏览到第一页的记录或者最后一页的记录时，相关浏览链接仍然显示出来，但此时它已经没有了相应的浏览功能。例如，目前已经浏览到第一页的 4 条记录，此时再单击“<<”浏览字符串已经没有任何意义了，系统并不会定位到上一页的 4 条记录，因为此时已经没有前 4 条记录了。同样，当用户浏览到最后一页时也存在同样的问题。

为了解决这个问题，我们需要往页面中添加一点智能化功能：当浏览到第一页记录或者最后一页记录时，让相关浏览链接自动隐藏起来。这可以通过 Show Region 服务器行为来实现。

首先请在页面中选择“<<”字符串，然后在 Server Behaviors 面板中单击⁺按钮，并在出现的菜单中选择 Show Region / Show Region If Not First Record”菜单命令，于是出现如图 11-28 所示的 Show Region If Not First Record 窗口。

在 Show Region 子菜单中共有 6 个菜单命令，分别针对了记录集是否为空、非空、第一条记录、不是第一条记录、最后一条记录以及不是最后一条记录这 6 种情况下显示

浏览链接的控制。在此处之所以选择“ Show Region If Not First Record ”，作用就是只有当没有浏览到第一条记录时才显示该链接，也就是说，当浏览到第一条记录时，服务器行为就会将“ << ”浏览字符串隐藏起来。单击 OK 按钮。单击 OK 按钮关闭图 11-28 所示的对话框。

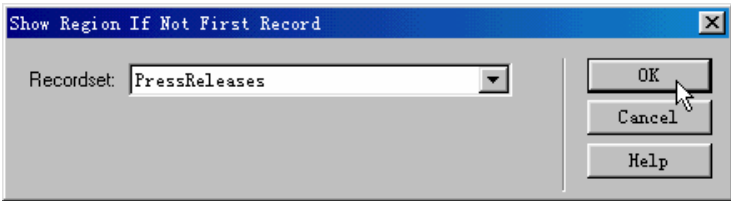


图 11-28
Hide
Region 窗
口

重复上述步骤，为“ >> ”浏览字符串增加 Show Region 服务器行为，只不过此时要选取“ Show Region / Show Region If Not Last Record ”菜单命令。

最后保存所做的工作。并在浏览器中进行测试。可以看出，由于记录集中只有 6 条记录，当显示第一页的 4 条记录时，“ << ”字符串隐藏起来了，而单击“ >> ”链接进入第二页的 2 条记录时，“ >> ”却又消失，而“ << ”又显示出来了。

此时打开 HTML 源代码检查器就可以看到系统自动为我们添加的服务器端脚本，如图 11-29 所示。

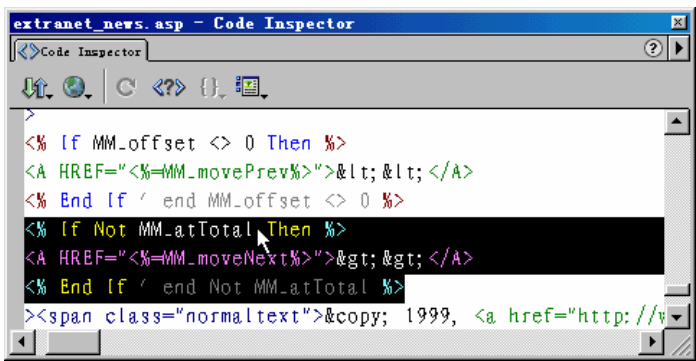


图 11-29
系统自动
添加服务
器端脚本

11.3.2 处理细节页面

在上面的 extranet_news.asp 页面中，只显示了数据库中的各出版信息的一部分内容，为了显示各出版信息的详细内容，我们还需要为其中的每一项创建一个链接，使其链接到一个细节页面 extranet_release.asp，其中显示每一个出版信息的全文细节信息。

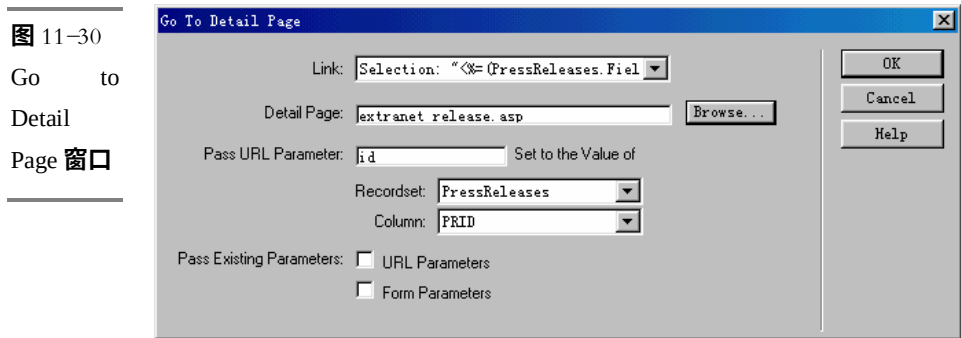
当然，这里要创建的链接并不像为静态页面创建链接那样简单了。当在主页面中单击某个链接打开细节页面时，其中显示的内容应该是用户在主页面中所选择的某个特定

记录的详细信息，也就是说，细节页面必须知道用户在主页面上单击了哪一个记录，这需要主页面给细节页面传递信息。

下面我们就介绍如何正确地在 `extranet_news.asp` 页面中添加到 `extranet_release.asp` 页面的链接，以便在 `extranet_release.asp` 页面中能够正确地显示相应的信息。

如果 `extranet_news.asp` 页面在文档窗口中显示，则请单击占位符“`{PressReleases.PRTitle}`”以选择它；如果 `extranet_news.asp` 页面在 Live Data 窗口中显示，则请单击第一条记录的标题“Web Site Adds Revenues”以选择它。

确保 Server Behavior 面板处于打开状态，单击其中的  按钮，然后在出现的菜单中选择 Go to Detail Page 菜单项，于是出现如图 11-30 所示的 Go to Detail Page 窗口。



在 Detail Page 文本框中，单击右边的 Browse 按钮，然后选择预先设计好的细节页面 `extranet_release.asp`，在 Pass URL Parameter 文本框中，输入“`id`”。这样做的目的是将一个名为“`id`”的参数传递给细节页面，以便确定用户所选择的链接字段。服务器将根据 Recordset 和 Column 这两个下拉列表中的值来设定参数“`id`”的值。本例中，`id` 的值为 `PressReleases` 记录集中的 `PRID` 字段的值。

单击 OK 按钮并保存所做的工作。按下 F12 快捷键进行测试。读者会发现主页面上每条记录的标题上都增加了一个链接，当用户单击该链接时，就会打开细节页面。只不过此时细节页面中还没有动态内容，下面我们将给细节页面创建记录集，并添加动态内容。

11.3.3 定义一个过滤的记录集

在细节页面中，我们只需要数据库中的两个字段：`PRID` 和 `PRCopy`。为了确保在细节页面中能显示正确的信息，我们要以 Go to Detail Page 服务器行为所创建的 URL 参数为基准对 `PRID` 字段进行过滤，从而得到正确的记录。

在 Site 窗口中双击 `extranet_release.asp` 文档，在文档窗口中打开它。接着在 Data

Bindings 面板中单击  按钮，然后在出现的菜单中选择 Recordset(Query) 菜单项，于是出现简单版本的 Recordset 窗口(如果出现的是高级版本的 Recordset 窗口，请单击其中的 Simple 按钮切换到简单版本的 Recordset 窗口)。在 Name 文本框中输入 PressReleaseInfo，然后在 Connection 下拉列表中选择 AndesExtranet，在 Table 下拉列表(其中显示的是数据库中的所有表名) 中，选择 PressReleases 项，并在 Columns 域中，选择 Selected 单选钮，以便只选择某些特定的字段。

借助 Ctrl 键在列表中选择 PRID 和 PRCopy 这两个字段，并将 Filter 域中第一个下拉列表的值由 None 改为 PRID，最后一个文本框的值由“ PRID ”改为“ id ”。如图 11-31 所示。

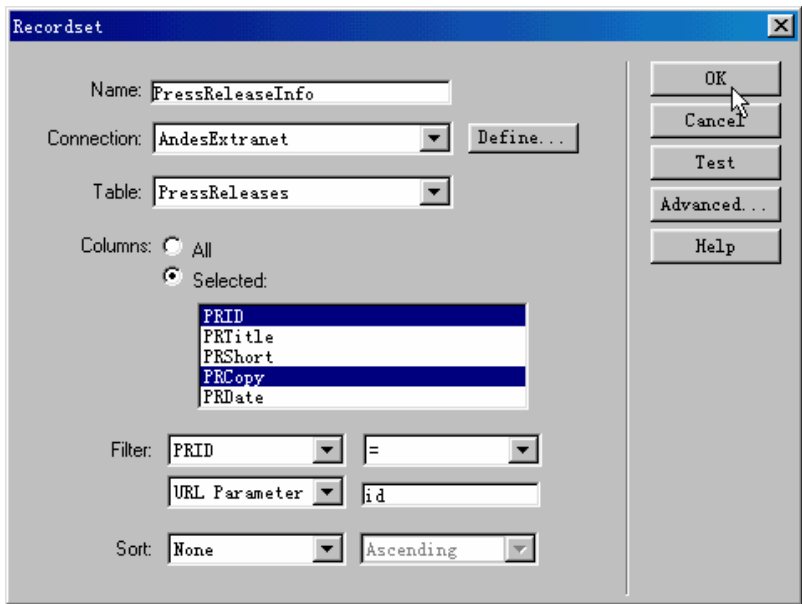


图 11-31
为细节页面创建过滤记录集

单击 Test 按钮进行测试，于是会出现如图 11-32 所示的窗口，要求输入一个 URL 参数 id 的测试值，请在文本框中输入数字 2，然后单击 OK 按钮。

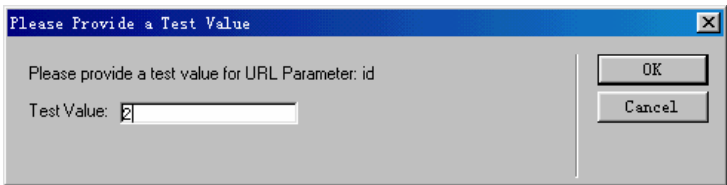
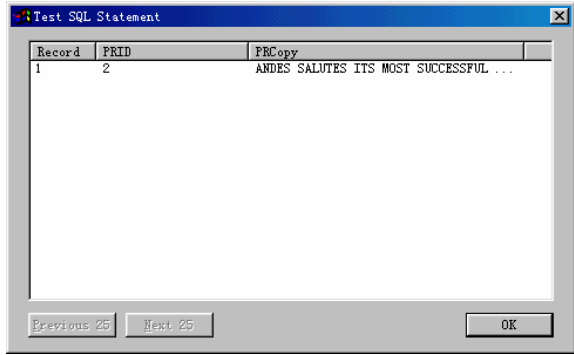


图 11-32
输入测试值

这时在 Test 窗口中我们只能看到一条记录，即 PRID 的值为 2 的那条记录，如图 11-33 所示。

图 11-33
Test 窗口
中只有一
条记录



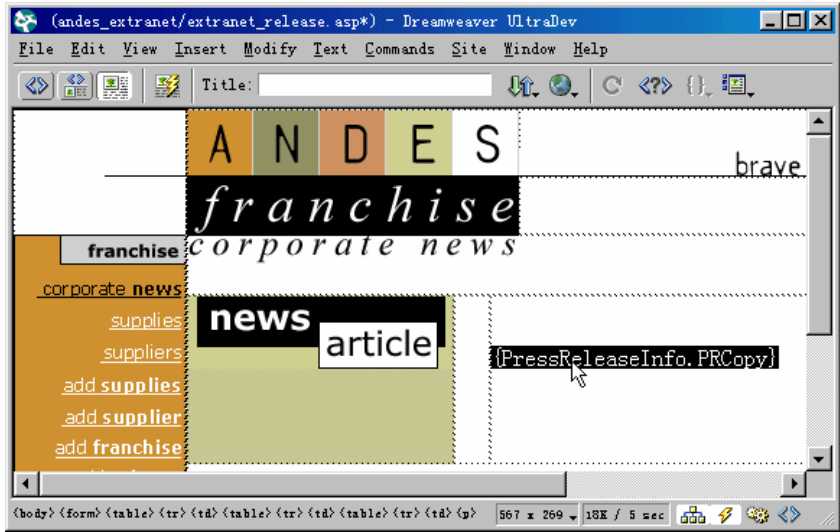
连续两次单击 OK 按钮，关闭 Test 窗口和 Recordset 窗口，将记录集增加到页面中。

11.3.4 在细节页面中增加动态内容

定义了记录集以后，我们就能够以其中的字段作为数据源添加到页面中。细节页面的动态内容非常简单，只需要 PRCopy 字段的值，即每条记录的详细信息。

确保 Data Bindings 面板已经打开，并展开 PressReleaseInfo 记录集。选取其中的 PRCopy 字段，并将其拖放到文档窗口中最大的空白表格单元中去（位于图片“news article”的右边），这时在该单元格中会出现 {PressReleaseInfo.PRCopy} 占位符，如图 11-34 所示。

图 11-34
在细节页
面中添加
动态内容



请保存所做的工作，然后把文档窗口切换到 Live Data 窗口，对刚才添加的动态内容

进行预览。读者可以发现{PressReleaseInfo.PRCopy}占位符将被记录集中的实际数据所代替。

由于细节页面需要使用 URL 参数的值来过滤记录集，而刚才我们在 Live Data 窗口中进行预览时并没有提供 URL 参数值，因此系统会自动在 Live Data 窗口中的细节页面中显示当前所在记录的值。我们也可以模拟 URL 参数值来预览显示结果，这可以通过 Live Data Settings 窗口来实现。

选择“ View / Live Data Settings ”菜单命令，或者在 Live Data 窗口中单击工具条上的 **Settings...** 按钮，均可打开 Live Data Settings 窗口，如图 11-35 所示。

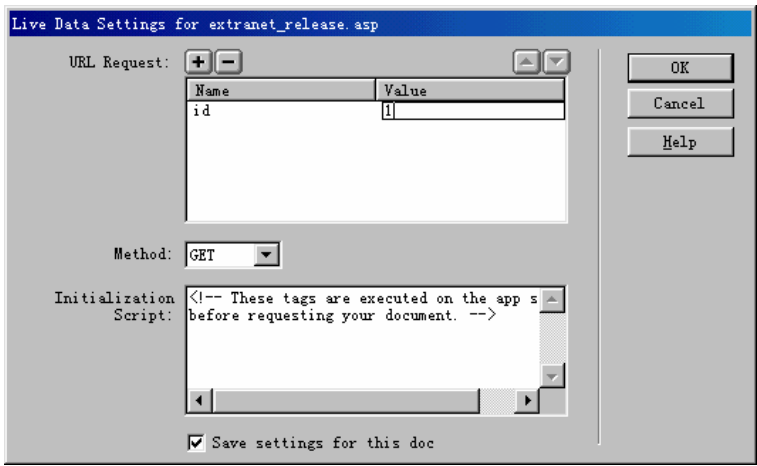


图 11-35
Live Data
Settings 窗
口

单击 URL Request 域中的 **+** 按钮，于是系统就会在列表中新建一个域，在其中的 Name 列中输入“ id ”，使用 Tab 键切换到 Value 列，接着输入数字“ 1 ”。然后单击 OK 按钮。

于是你会在 Live Data 窗口的工具条上看到如图 11-36 所示的变化，即在可输入的下拉列表中出现了“ id=1 ”的语句。这实际上就是我们增加的 URL 参数值，现在完整的 URL 应该是：

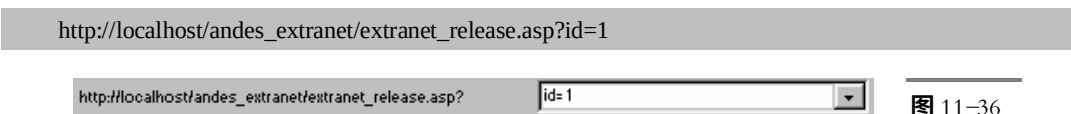


图 11-36
增加 URL
参数值

选择“ View / Refresh Live Data ”菜单命令，或者使用 Ctrl+R 快捷键，对 Live Data 窗口进行刷新，则 extranet_release.asp 页面中显示的就是 PRID 字段值为 1 的记录。

如果要在 Live Data 窗口中显示其他的记录,可以通过 Live Data Settings 窗口来修改 id 的值,或者直接在 Live Data 窗口的可输入下拉列表进行修改(如将 id=1 改为 id=2 等),然后刷新就可看到修改后的结果。

现在可以对整个站点进行测试了。在文档窗口中打开 extranet_news.asp 页面,然后按下 F12 快捷键打开浏览器进行测试。单击主页面中每条记录的标题链接,在细节页面中都会正确的显示出相应记录的详细信息。

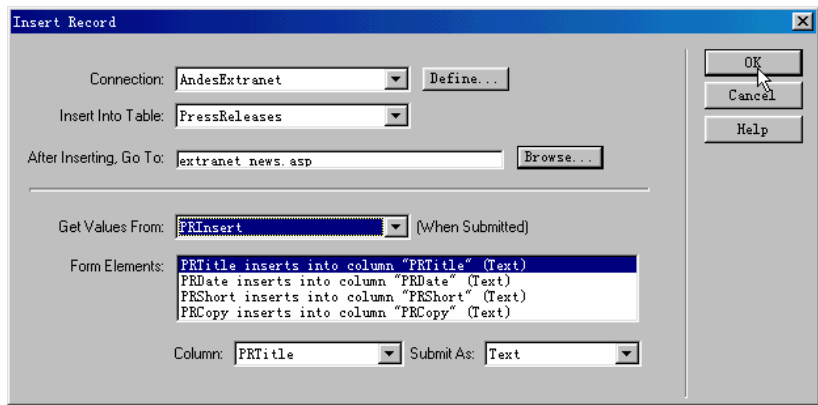
11.3.5 往数据库中增加新记录

现在我们的 Web 应用已经比较实用了。下面我们进一步增强其功能,使得用户可以通过 Web 页面往数据库中增加新记录。

切换到 Site 窗口,然后双击 Andes Extranet 站点中的 extranet_add_release.asp 文件(这个文件是事先创建好的用来增加新记录的页面,其中包含了一个表单),以便在文档窗口中打开它。

确保 Server Behavior 面板处于打开状态,单击其中的  按钮,然后在出现的菜单中选择 Insert Record 菜单项,于是出现如图 11-37 所示的 Insert Record 窗口。

图 11-37
Insert
Record 窗
口



在 Connection(数据库链接)下拉列表中选择 AndesExtranet 项,在 Insert Into Table (需要插入的数据库表)下拉列表中选择 PressReleases 项。单击  按钮来设置“After Inserting,Go To”域中的值,在打开的 Select File 窗口中,选择文件“extranet_news.asp”,然后单击 OK 按钮关闭 Select File 窗口。这样,当用户提交了表单以后,页面就会自动重新定位到 extranet_news.asp 中。

缺省情况下,窗口下半部分的 Get Values From 下拉列表中应该选择了 PRInsert 表单。PRInsert 就是 extranet_add_release.asp 文件中的表单名,该表单中包含了往数据库中增加新记录所需要的所有表单元素。

在 Form Elements (表单元素) 列表中, Insert Record 服务器行为已经自动地将表单中的每一个元素同数据库中的相应字段匹配起来了。请不要修改缺省值。

之所以 Insert Record 服务器行为能够自动完成这个匹配过程,是因为这里的表单元素名同数据库中的相应字段名是相同的。如果用户需要修改这种匹配关系,可以在列表中选择某一个表单元素,然后修改列表下方的 Column 域和 Submit As 域中的值 (即两个下拉列表中的值) 即可。

最后单击 OK 按钮,并保存所做的工作,现在可以进行测试了。
切换到 extranet_add_release.asp 文档窗口,然后按下 F12 快捷键打开浏览器窗口,在页面的表单中输入新记录的相应信息,如图 11-38 所示。

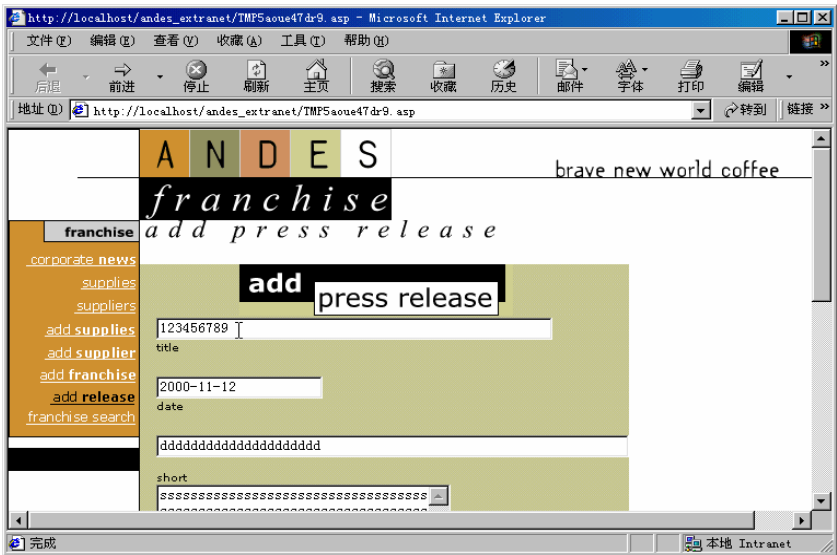
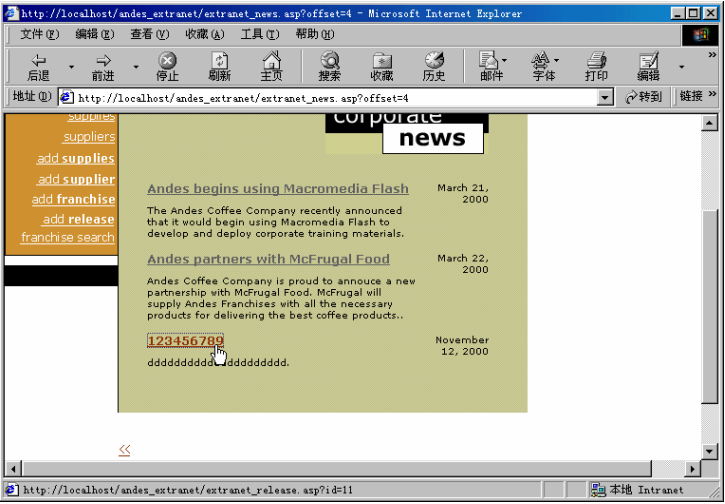


图 11-38
增加新记录

输入完成后,单击 Add 按钮提交表单,于是页面会重新定位到 extranet_news.asp 页面中,单击“ >> ”链接浏览到最后一页,就会发现我们刚才添加的记录已经出现在页面中了,如图 11-39 所示。

至此,我们的 Web 应用示例就介绍完了。当然,读者还可以继续增强本 Web 应用的功能。例如,按照同样的方法,我们还增加更新记录和删除记录的功能 (分别通过 Update Record 和 Delete Record 服务器行为来实现)。这一步工作就留给有兴趣的读者去完成吧!

图 11-39
新记录已
经增加到
数据库中



第 12 章 创建留言簿

本章概要：

- ☒ 开发留言簿前的准备工作，如设计基本网页、创建数据库、创建 DSN、定义站点、创建数据库链接等
- ☒ 完成填写留言页面
- ☒ 完成浏览留言页面

在本章中，我们来看一下如何使用 DW UltraDev 4 来创建一个非常常用的 Web 应用——留言簿。我们不用写任何一行程序就能够做出一个功能齐全、使用数据库的 ASP 留言簿了。通过本章的学习，相信读者会更深切地体会到 DW UltraDev 4 的方便、神奇之处了。

12.1 开发留言簿前的准备工作

为简单起见，本章我们介绍留言簿时假设用户的 DW UltraDev 和 Web 服务器安装在同一台计算机上。如果用户使用的操作系统是 Windows 9x 或 Windows NT Workstation，则必须安装 Microsoft Personal Web Server 4.0；如果用户使用的操作系统是 Windows NT Server 或 Windows 2000，则必须安装 Microsoft IIS 4.0。

本章介绍的留言簿示例的站点文件及子目录都存放在了本书所附光盘的 guest 目录中，供读者参考使用。在开始动手开发留言簿之前，请在 WWW 服务器的根目录中创建一个名为 guest 的子目录。例如，如果 WWW 服务安装到了 C 盘的 Inetpub\wwwroot 目录中（缺省情况），则新创建的 guest 目录的完整路径应该是 C:\Inetpub\wwwroot\guest。

下面我们可以开始着手进行 Web 应用开发前的一些准备工作了。准备工作主要包括：设计基本网页、创建数据库、创建 DSN、定义站点和创建数据库链接。

12.1.1 设计基本网页

在开始设计基本网页之前，我们先来演示一下留言簿的功能。图 12-1 中所示就是留言簿的主界面，用户可以在此处浏览其他用户的留言。每页可以显示 5 条留言，单击“上一页”和“下一页”按钮可以浏览其他的留言。每条留言都包括用户名、Email 地址、主页地址、ICQ 号码和 OICQ 号码、留言时间、留言主题和留言内容。单击其中的 Email 地址图标将会给留言者发送 Email；而单击主页地址图标将会在浏览器中打开留言者的主页。

图 12-1
留言簿的
主界面



如果用户想要留言，只需单击“我要留言”链接，就会打开如图 12-2 中所示的填写留言页面。用户只需在其中填写相应的留言信息，然后提交表单即可留言。

从上面的演示可以看出，创建这样的留言簿需要两个页面：一个用来填写留言信息，另一个用来浏览留言。使用最熟悉的网页编辑软件来创建这两个网页即可，例如使用 FrontPage、Dreamweaver 或是 DW UltraDev 来创建都可以。在此我们先将这两页的版型内容编辑出来，稍后再使用 DW UltraDev 4 将它们与数据库的内容相链接。因为这两页是标准的 HTML 网页，所以使用什么软件编辑都可以。但别忘了最后把默认扩展名由 .htm 或 .html 另存为 .asp。

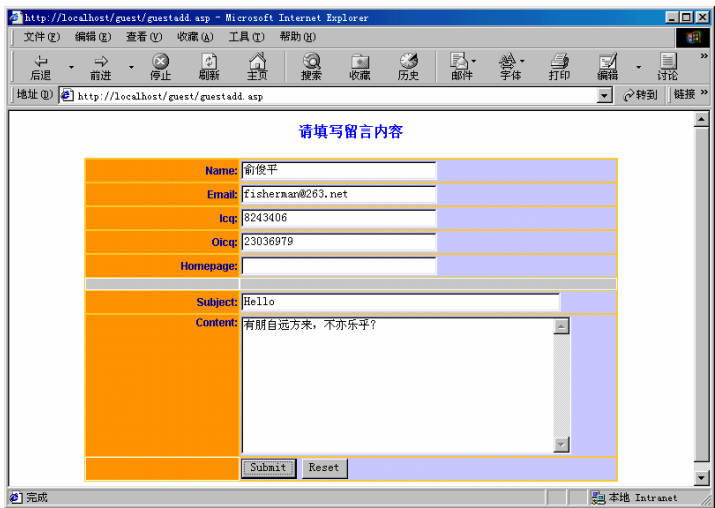


图 12-2
填写留言
页面

浏览留言的网页文件名称取为 index.asp，它是留言簿应用的首页，其内容如图 12-3 所示。填写留言的网页文件名称为 guestadd.asp，其内容如图 12-4 所示。将它们放在留言簿的网站根目录下(如 C:\InetPub\wwwroot\guest 目录)，当然还有其他很多的图形档案，可以放在网站根目录中的 img 目录下。



图 12-3
浏览留言
的网页



图 12-4
填写留言
的页面



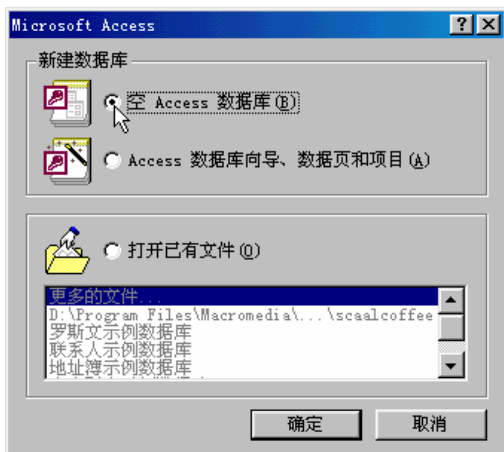
由于光盘中所带的 index.asp 文件和 guestadd.asp 文件是最终修改后的页面,因此与这里设计的基本网页是不一样的。为了使读者能够看到最开始设计的基本网页,我们特地将它们分别命名为 index'.asp 和 guestadd'.asp,请读者参考。另外,在设计网页时所需要的图片文件都可以在光盘的 guest\img 目录中找到。

12.1.2 创建数据库

网页准备好后,要准备一个储存留言内容的数据库,我们使用最简单、最容易取得的 Access 数据库。也许 Access 不适合当作大型数据库使用,但是对留言簿这样的 Web 应用来说是绰绰有余了!使用 Access 97 或 Access 2000 都可以,在这里我们以 Access 2000 做示范。本书并不打算对 Access 的功能做详细的说明,如果读者有兴趣的话,请参阅 Access 相关书籍。

首先打开 Access 2000,在出现的新建数据库窗口中选择“空 Access 数据库”单选钮,然后按下“确定”按钮,如图 12-5 所示。

图 12-5
创建空的
Access 数
据库



在“文件新建数据库”窗口中将数据库储存到留言簿的网站根目录中(如 C:\InetPub\wwwroot\guest 目录),文件名取为 guest.mdb,如图 12-6 所示。按下“创建”按钮即可存盘。

在“guest:数据库”窗口的数据库表列表中,双击“使用设计器创建表”项,如图 12-7 所示,于是会出现数据库表设计界面。在第一行的“字段名称”中输入“ID”、“数据类型”中选择“自动编号”。然后在这一行上单击鼠标右键,在弹出的菜单中选择“主键”菜单命令,于是在此行的最前面会出现一个主键标志,如图 12-8 所示。



图 12-6
保存数据
库文件

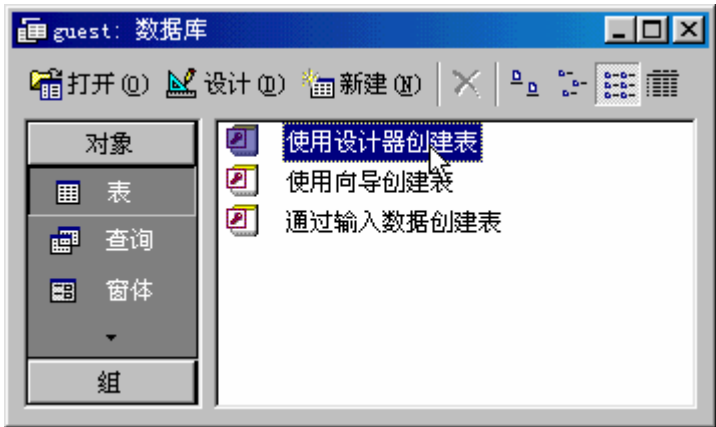


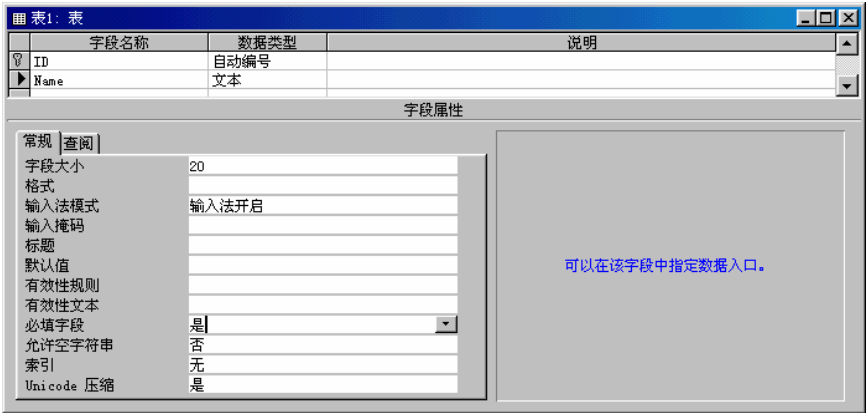
图 12-7
使用设计
器创建表

guest_message: 表		
	字段名称	数据类型
	ID	自动编号

图 12-8
输入数据
库表字段
(1)

同样，在第二行的“ 字段名称 ”中输入“ Name ”；“ 数据类型 ”中选择“ 文本 ”。在下方的 “ 常规 ” 字段属性区域中；“ 字段大小 ”中输入“ 20 ”，对于 “ 必填字段 ”选择 “ 是 ”，如图 12-9 所示。此字段用于储存留言者名字。

图 12-9
输入数据库表字段
(2)



接下来使用相同的方法，依照表 12-1 所示把其他字段也都建立起来。

表 12-1 数据库表各字段类型及含义

意 义	字 段 名 称	数 据 类 型	字 段 大 小	必 填 字 段	允 许 空 字 符 串	备 注
留言编号	ID	自动编号	-	-	-	主键
姓名	name	文本	20	是	否	-
Email	email	文本	50	否	是	-
个人主页	homepage	文本	100	否	是	-
ICQ 号码	icq	文本	15	否	是	-
OICQ 号码	oicq	文本	15	否	是	-
留言标题	subject	文本	100	是	否	-
留言内容	content	文本	255	是	否	-
留言时间	date	日期/时间	-	“默认值”输入“ Now()”	“索引”选择“是(有重复)”	-

输入完成后，把设计窗口关闭，系统会询问是否保存数据库表，选择“是”，然后将数据表的名称命名为“ guest_message ”。图 12-10 所示的就是设计完成后的数据库表“ guest_message ”。

图 12-10
设计完成
后的数据
库表
“ guest_m-
essage ”



最后关闭 Access，数据库部分已经制作完成。

12.1.3 创建 DSN

数据库建立好之后，要设定系统的 DSN(数据源名称)，才能让网页可以知道数据库所在的位置以及数据库相关的属性。使用 DSN 的好处还有，如果移动数据库文件的位置，或是换成别种类型的数据库，只要重新设定 DSN 就可以了，不需要去修改原来使用的程序。由于本章的留言簿使用的是 MS Access 数据库，因此需要先创建一个使用 MS Access Driver 的 ODBC 系统 DSN。其步骤如下所述。

由于本章的示例使用的是 MS Access 数据库，因此需要先创建一个使用 MS Access Driver 的 ODBC 系统 DSN。具体的创建步骤同 11.1.1 节中所介绍的步骤类似，只是在如图 12-11 所示的“ODBC Microsoft Access 安装”窗口中，注意在“数据源名”文本框里输入“Guestbook”，并通过“选取”按钮，确定数据库文件所在的路径，如 C:\Inetpub\wwwroot\guest\guest.mdb。

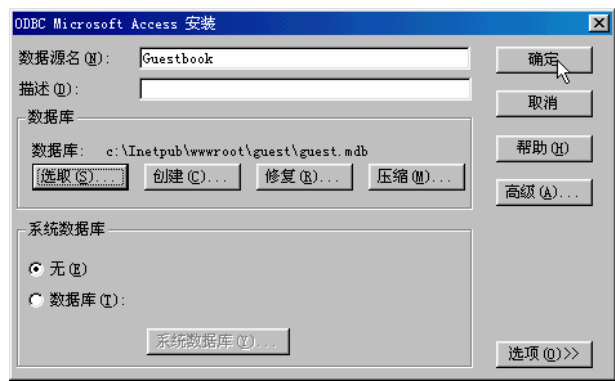


图 12-11
“ODBC
Microsoft
Access 安
装”窗口

12.1.4 定义站点

开始使用 DW UltraDev 4 制作网站应用程序的第一件事，就是定义站点。我们必须告诉 DW UltraDev 4 网站根目录在哪里，以及服务器位置、类型等等的相关信息。

定义好站点之后，也可以使用 DW UltraDev 4 的站点管理窗口来管理属于这个网站的网页和图形等文件。如果在站点管理窗口中更改文件名，移动文件位置，DW UltraDev 4 都会自动地更新网站中所有页面的链接，非常方便。不过这些功能都必须先定义好站点之后才能使用。

接下来启动 DW UltraDev 4，并打开 Site 站点管理器窗口(选择 Window / Site Files”菜单命令，或者在启动面板中单击 Site 按钮，或者按下快捷键 F5)。然后选择 Site / New

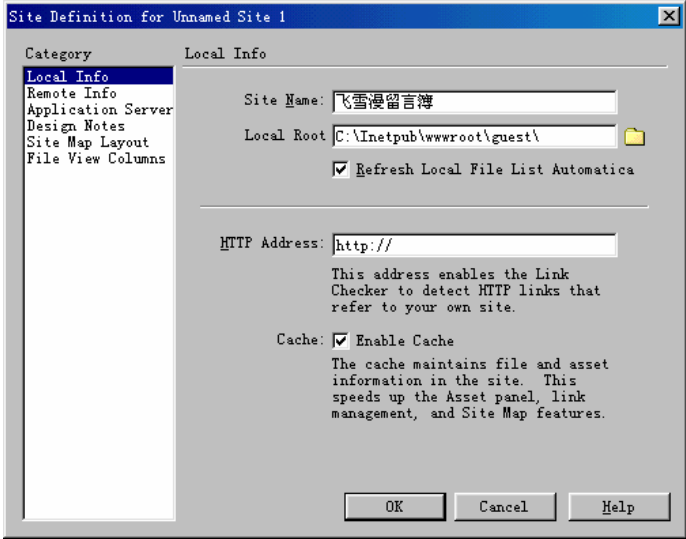
Site 菜单命令打开 Site Definition 窗口。单击此窗口左边的 Category 列表中的 Local Info，打开 Local Info 面板，如图 12-12 所示。在 Site Name 文本框中输入“ 飞雪漫留言簿 ”，在 Local Root 域中输入本地站点的根目录 C:\inetpub\wwwroot\guest\，或者使用其右侧的  按钮定位到该目录。

图 12-12

“ Site
Definition ”
窗口的

Local Info

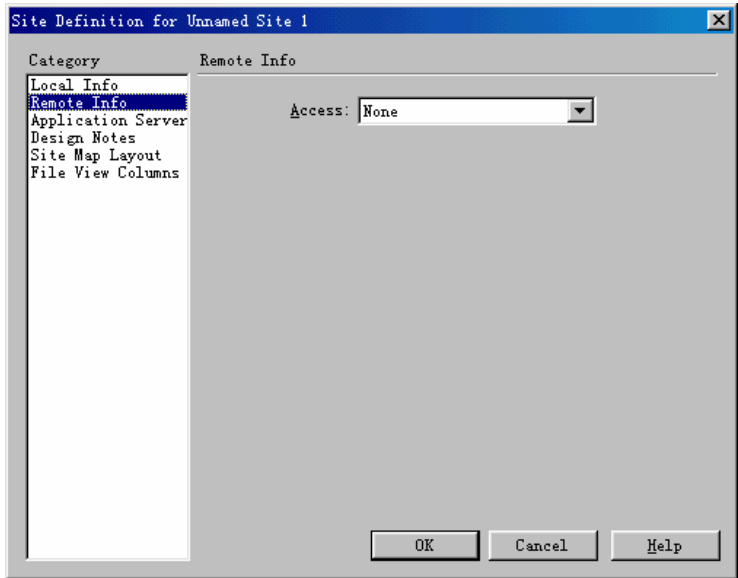
面板



在 Category 列表中单击 Remote Info 切换到 Remote Info 面板，如图 12-13 所示。由于本例中的本地站点和远程站点位于同一个目录下，因此不需要进行文件传输。请保持 Server Access 下拉列表中的选项为 None。

图 12-13

Remote
Info 面板



在 Category 列表中单击 Application Server,切换到 Application Server 面板,如图 12-14 所示。在 Server Model 下拉列表中选择 ASP 2.0 项,在 Default Scripting 下拉列表中选择 VBScript 或 JavaScript,设置 Page Extension 下拉列表的值为.asp。然后在 Access 下拉列表中选择 Local/Network,在 Remote Folder 中输入远程站点的根目录 C:\inetpub\wwwroot\guest,或者使用其右侧的按钮定位到该目录。最后在 URL Prefix 文本框中输入 http://localhost/guest/。

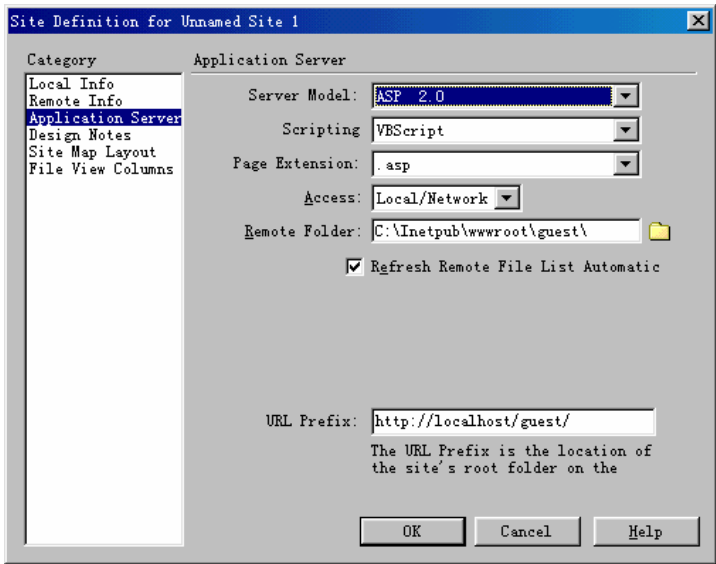


图 12-14
Application
Server 面板

设置完成后单击 OK 按钮,关闭 Site Definition 窗口,这时可能会出现如图 12-15 所示的警告窗口,表示 DW UltraDev 4 将为这个新站点创建一个缓冲区以提高站点的性能。单击 OK 按钮创建它,于是站点定义完成。

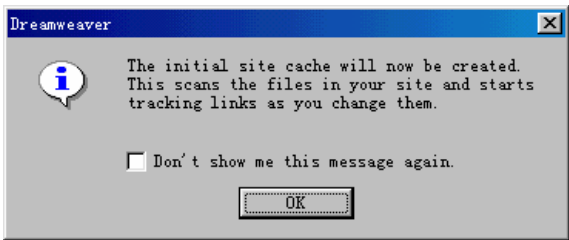


图 12-15
是否创建
缓冲区

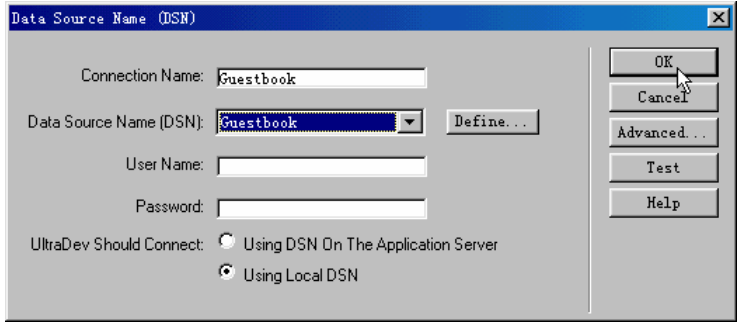
12.1.5 创建数据库链接

接下来要定义的是这个网站所要使用的数据库链接,这样一来这个网站中的网页才

能透过数据库链接来存取数据库里面的信息。之前我们已经设定好了系统 DSN，所以只要将数据库链接指定为之前所设的系统 DSN 即可。

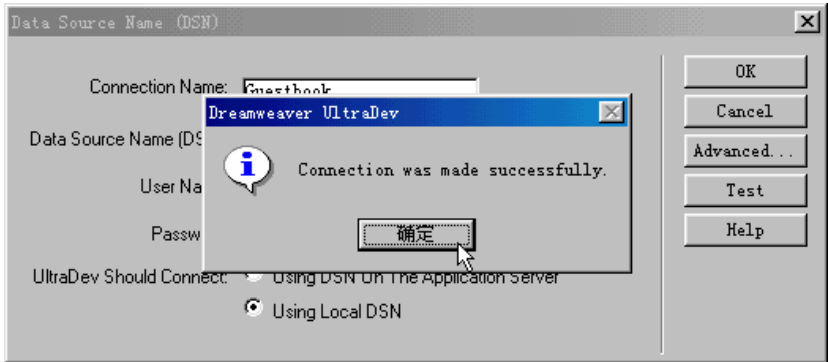
在 DW UltraDev 4 文档窗口中选择“ Modify / Connections ”菜单命令，打开 Connections 窗口。然后单击 New 按钮，在弹出的快捷菜单中选择“ Data Source Name (DSN) ”菜单命令，打开 Data Source Name(DSN) 窗口。设置 Connections Name 文本框的值为 Guestbook，然后在 Data Source Name(DSN) 下拉列表(该列表中列出了系统中的所有 ODBC DSN) 中选择 Guestbook，并保持 Username 和 Password 这两个文本框的值为空，如图 12-16 所示。

图 12-16
创 建
Guestbook
数据库链接



单击 Test 按钮对此数据库连接进行测试。如果数据库连接创建成功，则会弹出一个对话框，其中显示“ Connection was made successfully. ”的提示信息，如图 12-17 所示。单击“ 确定 ”按钮关闭提示窗口，然后单击 OK 按钮保存所创建的数据库链接。于是在 Connections 窗口的列表中就会显示出刚才创建的数据库链接名，如图 12-18 所示。最后单击 Done 按钮关闭 Connections 窗口。

图 12-17
测试数据
库链接



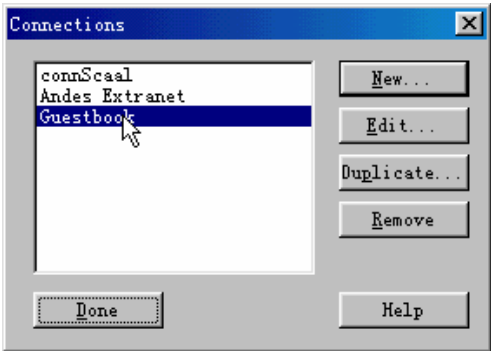


图 12-18
数据库链接创建成功

12.2 完成填写留言页面

因为现在数据库是空的，没有任何数据在里面，所以我们先来制作填写留言的页面，完成之后就可以使用这个页面将留言存进数据库中了。让我们看看如何完成一个新增数据到数据库的网页。

首先在 DW UltraDev 的文档窗口中打开 guestadd.asp 页面，然后调整一下窗体中各个表单元素的名称，最好是能够跟数据库中的相应字段名称一致，如此一来 UltraDev 可以自动帮我们配对，等等就可以看到了。

确保 Server Behaviors 面板是打开的，单击其中的  按钮，并在出现的菜单中选择“Insert Record”菜单命令，于是出现如图 12-19 所示的 Insert Record 对话框。

在该对话框的 Connection(数据库链接)下拉列表中选择 Gusetbook 项，在 Insert Into Table(需要插入的数据库表)下拉列表中选择 Guest_Message 项。

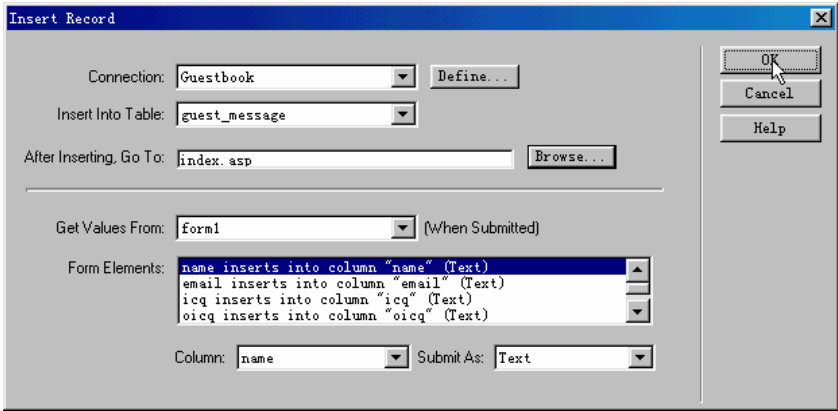


图 12-19
Insert Record 对话框

单击 **Browse...** 按钮来设置“After Inserting, Go To”域中的值，在打开的 Select File 窗口中，选择文件 index.asp，然后单击 OK 按钮关闭 Select File 窗口。这样，当用户提交了表单以后，页面就会自动重新定位到 index.asp 中。于是新增加的留言就可以马上看到。

缺省情况下，窗口下半部分的 Get Values From 下拉列表中应该选择了 from1 表单，也就是这个页面中惟一的 HTML 表单。该表单中包含了往数据库中增加新记录所需要的所有表单元素。如果各表单元素的名称跟数据库中的对应字段名称一致的话，则在 Form Elements(表单元素) 列表中，Insert Record 服务器行为已经自动地将表单中的每一个元素同数据库中的相应字段自动匹配起来了。如果没有的话，就必须自己调整一下。调整的方法为：先从 Form Element 选择一个窗体元素，再从 Column 下拉列表中选择数据表的对应字段。在表单提交时，该表单元素中所填写的数据，将会被插入到所选择的数据表字段中。因此请务必一一将它们配好对。

最后单击 OK 按钮完成插入记录的配置，并保存所做的工作。现在可以对填写留言页面进行测试了。

按下 F12 快捷键在浏览器中预览网页的样子，或是在浏览器中输入 <http://localhost/guest/index.asp> 网址直接来浏览网页。读者可以输入适当的数据，最后按下 Submit 按钮提交。如果没有什么错误讯息，并且页面自动切换到 index.asp 页面，就表示留言数据已经成功地插入到数据库中了。由于我们还没有完成浏览留言页面，因此在 index.asp 页面中看不到新插入的记录。但是该数据确实已经被保存在数据库中。如果读者不相信的话，可以使用 Access 2000 打开 guest.mdb 文件来看看。读者可以多填写几次留言，方便下一节中浏览留言页面的测试。

12.3 完成浏览留言页面

我们已经学会如何制作一个填写留言的网页了，接下来就是浏览所有留言的页面。在开始之前，得先准备好页面中要使用的记录集。这个记录集中包含了所有要显示的留言记录。

12.3.1 定义留言记录集

首先在文档窗口中打开 index.asp 页面。

在 Data Bindings 面板中单击  按钮，然后在出现的菜单中选择 Recordset(Query) 菜单项，于是出现简单版本的 Recordset 窗口(如果出现的是高级版本的 Recordset 窗口，请单击 Simple 按钮切换到简单版本的 Recordset 窗口)。在 Name 文本框中输入“ All_Message ”，然后在 Connection 下拉列表中选择“ Guestbook ”项，在 Table 下拉列

表(其中显示的是数据库中的所有表名)中,选择“ Guest_Message ”项,并在 Columns 域中,选择 All 单选钮选中所有字段,这样可以将 Guest_Message 数据库表中每条记录所有的字段都取出来。如图 12-20 所示。

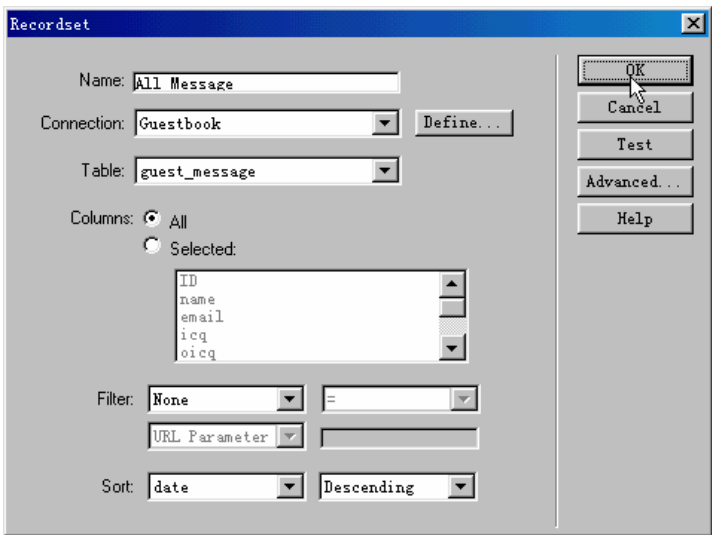


图 12-20
定义留言
记录集

通常我们还希望在显示留言时能够以留言的时间来排序,最新的留言出现在最上面。要实现这个功能需要对记录集进行排序。所以在图 12-23 中所示窗口的 Sort(排序)域的第一个下拉列表中选择“ Date ”项,这是我们数据库表中记录留言时间的字段,在第二个下拉列表中再选择“ Descending ”(递减)项,即按逆序对记录进行排序。

完成之后可以先按下 Test 按钮,看看选择出来的数据以及排序方式是不是我们所需要的。如果没错的话就连续两次单击 OK 按钮,完成记录集的设置。

12.3.2 显示浏览留言页面

接下来要做的就是将页面中预先设计好的文字,以数据库中的内容来取代。这些文字最后应该都是由 ASP 程序从数据库中取出来的动态内容。

在 index.asp 页面中的“ 俞俊平 ”这几个字要以数据库中的 Name 字段来替换掉。选取“ 俞俊平 ”,然后在 Data Binding 面板中,单击记录集 Record(All_Message) 左边的加号,将记录集展开。点选 Name 字段,再单击下面的 Insert 按钮,或者将 Name 项拖放到文档窗口中选中的“ 俞俊平 ”文字上,于是文字“ 俞俊平 ”就会被占位符“ All_Message.name ”替换掉了,如图 12-21 所示。

这里读者会发现在 Data Bindings 面板下方的 Bind To 下拉列表中的值自动变成了“ a.href ”,而我们要替换的就是标注<a>的 href 属性。选择记录集中的 Homepage 字段,然后单击 Bind 按钮将网页图标的超链接绑定到 Homepage 字段中。如图 12-24 所示。

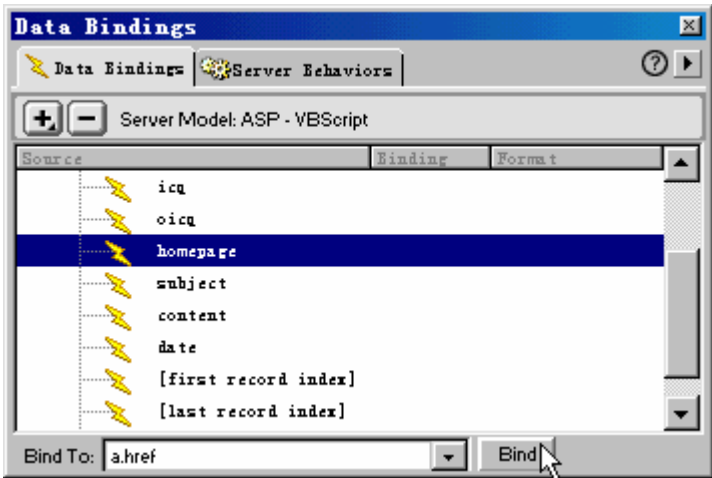


图 12-24
将网页图
标的超链
接绑定到
Homepage
字段中

使用同样的方法先选取 E-mail 图标,在属性面板中的 Link 文本框中随便输入一个网址。然后在文档窗口状态栏的地方,单击接近右方的<a>标注。再选择 Data Bindings 面板中的 E-mail 字段,按下 Bind 按钮将 E-mail 图标的超链接绑定到 E-mail 字段中。

但是,跟网页图标不太一样的是,网页网址是直接以数据库的内容来取代,但是 E-mail 的超链接必须在 E-mail 前面加上“mailto:”,而数据库中的 E-mail 字段中并没有这几个字,所以我们得帮它加上去。方法是:选取 E-mail 图标之后,在属性面板中的 Link 文本框中所有文字的最前面加上“mailto:”这几个字。如图 12-25 所示。



图 12-25
修 改
E-mail 图
标的 Link
属性

现在请将网页存盘,然后按 F12 快捷键在浏览器中进行预览。可以看到,刚才被替换的那些内容和数据都已经显示出了数据库中的实际数据。

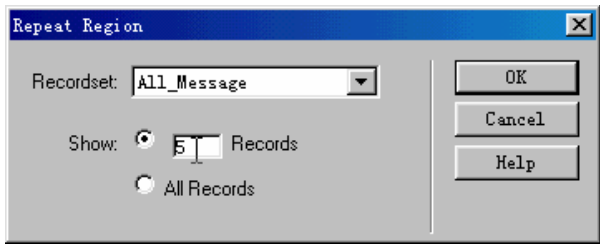
12.3.3 显示多条留言记录

刚才在测试中读者可能已经发现了，虽然已经将静态的文字替换掉了，但是为什么只能显示一条留言记录呢？别着急，下面我们就来实现多条留言的显示。我们要让放置留言记录的表格范围重复，而且里面的内容分别是每一笔留言的纪录。这当然离不开 Repeat Region 服务器行为。

首先在文档窗口中单击表格中的任何一个占位符（如留言标题占位符），然后选择“Modify / Table / Select Table”菜单命令选取整个表格。

接下来打开或者切换到 Server Behaviors 面板，单击其中的  按钮，并在出现的菜单中选择 Repeat Region 菜单项，于是出现如图 12-26 所示的 Repeat Region 窗口。

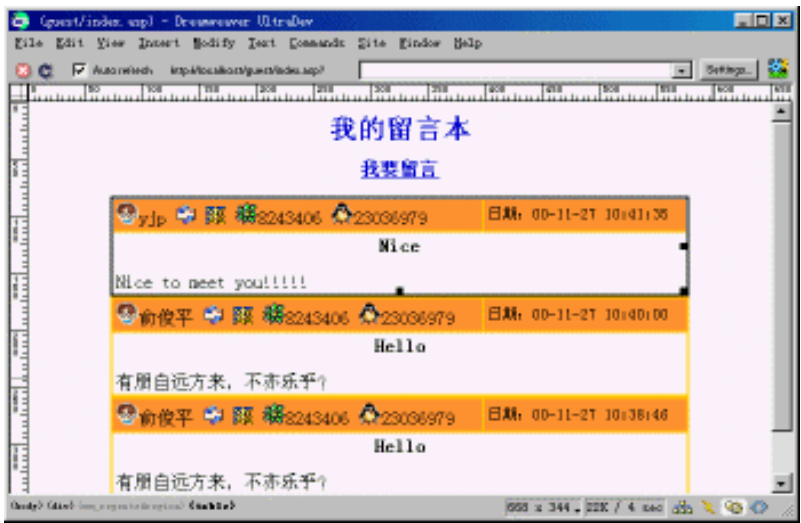
图 12-26
Repeat
Region 窗
口



确保在 Recordset 域中选择了 All_Message 记录集，然后选取带编辑框的单选按钮，在其中输入数字 5，以便每页显示 5 条记录，最后单击 OK 按钮。

现在将文档窗口切换到 Live Data 窗口，读者可以看到页面中把前 5 条记录都显示出来了，如图 12-27 所示。

图 12-27
在 Live
Data 窗口
中预览留
言



12.3.4 增加浏览链接

虽然留言都能够出现了，但一次只能显示 5 条记录(这是刚刚所设定的)，那么如何浏览数据库中的其他留言呢？为此我们需要增加一些记录浏览链接，类似于“上一页”、“下一页”的链接，让使用者可以看到所有其他的留言。

选取页面中的“上一页”文字，打开或者切换到 Server Behaviors 面板，单击其中的  按钮，并在出现的菜单中选择“Move To Record / Move To Previous Record”菜单项，于是出现如图 12-28 所示的窗口，请确定 Link 域中所选择的文本是“上一页”，Recordset 域中选择的是 All_Message 项，然后单击 OK 按钮。

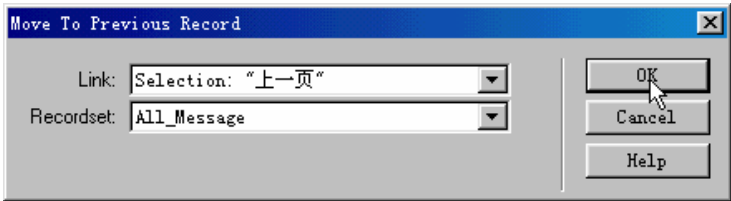


图 12-28
Move To
Previous
Record 窗
口

按照同样的方法为“下一页”文字增加浏览链接，只是此时要选择“Move To Record / Move To Next Record”菜单项。

保存所做的修改，并对站点进行测试。这里的测试在 Live Data 窗口中体现不出来，必须要使用浏览器才能进行测试。因此请选择“File / Preview In Browser”菜单命令或者按下快捷键 F12 在浏览器中打开当前文档。

12.3.5 增强浏览功能

这个问题和 11.3.1 节中介绍的是类似的，因此其解决办法也是类似的。只是此时要处理的对象是“上一页”和“下一页”文字，请读者参照 11.3.1 节的办法自行解决这个问题，在此不再赘述。

至此，我们的留言簿就设计完成了。如果读者对界面还不满意，可以任意进行修改。读到这里，相信读者一定会被 DW UltraDev 强大的功能所折服吧！

第 13 章 创建查询系统

本章概要：

- ☒ 创建查询系统的准备工作，包括新建站点、创建基本网页、创建数据库等
- ☒ 定义站点和数据库链接，包括定义站点、创建 DSN、创建数据库链接等
- ☒ 添加动态内容，包括创建判断页面和结果页面

上一章我们没有写任何一行程序就设计出了一个功能较为齐全的留言簿，这一章我们继续抛砖引玉，讲述如何创建一个简单的图书查询系统。和创建留言簿不同之处在于，本章在创建查询系统时需要读者编写几行代码。实际情况也往往如此，如果要编写一些功能强大一点的 Web 应用，不可能不写任何代码，而完全依靠 UltraDev 来实现的。通过本章的学习，希望读者能够掌握如何把 UltraDev 自动提供的代码和自己手工编写的代码结合在一起。

13.1 设计准备工作

同样为简单起见，本章我们介绍图书查询系统时假设读者的 DW UltraDev 和 Web 服务器安装在同一台计算机上。如果读者使用的操作系统是 Windows 9x 或 Windows NT Workstation，则必须安装 Microsoft Personal Web Server 4.0；如果读者使用的操作系统是 Windows NT Server 或 Windows 2000，则必须安装 Microsoft IIS 4.0。

本章介绍的图书查询系统示例的站点文件及子目录都存放在了本书所附光盘的 book 目录中，供读者参考使用。在开始动手开发查询系统之前，请 WWW 服务的根目录中创建一个名为 book 的子目录。例如，如果 WWW 服务安装到了 C 盘的 Inetpub\wwwroot 目录中（缺省情况），则新创建的 book 目录的完整路径应该是 C:\Inetpub\wwwroot\book。

其实在设计一个 Web 应用时，我们不一定要按照上一章所讲的那样，先创建好基本网页，然后再创建和定义站点，也可以反过来做。下面我们就先新建一个图书查询站点，然后再创建一些初始的网页和数据库。

另外，本章介绍的图书查询系统示例和上一章介绍的留言簿示例的另一个不同之处在于，本章的示例中本站站点和远程站点位于不同的目录中，而上一章的示例中这两者的目录是相同的，因此在实际的设计过程中也有所不同。

13.1.1 新建站点

首先启动 UltraDev，在文档窗口或站点窗口中选择“ Site / New Site ”菜单命令，打开如图 13-1 所示的 Site Definition 窗口。单击窗口左边的 Category 列表中的 Local Info 打开 Local Info 面板，在 Site Name 文本框中输入“ 图书查询系统 ”，在 Local Root 域中输入本地站点的根目录。本例中我们把本地站点的根目录放在了 C:\My Documents\book 下，当然读者也可以放在其他目录下，但不要和前面创建的 book 子目录（如 C:\Inetpub\wwwroot\book\）重合。读者可以直接在 Local Root 域中输入 C:\My Documents\book（或者读者自己指定的其他目录），或者使用  按钮定位到该目录。

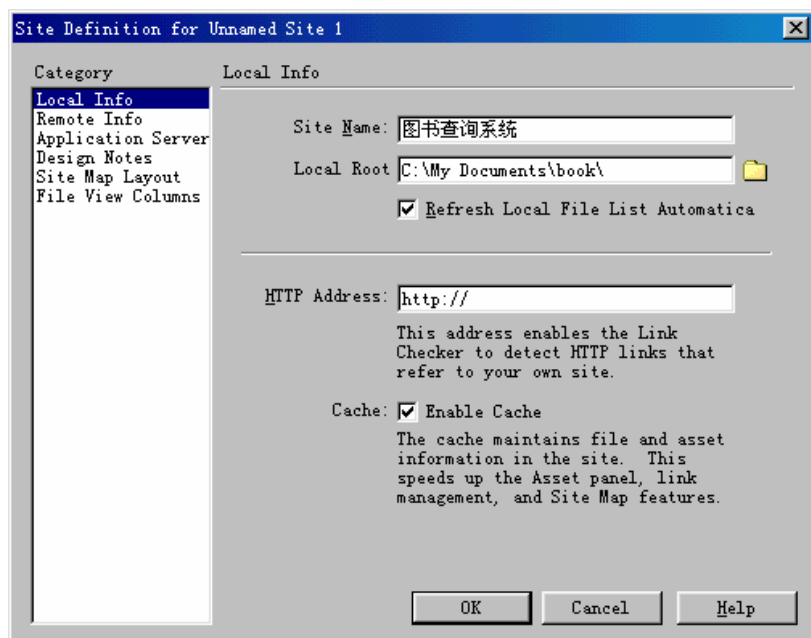
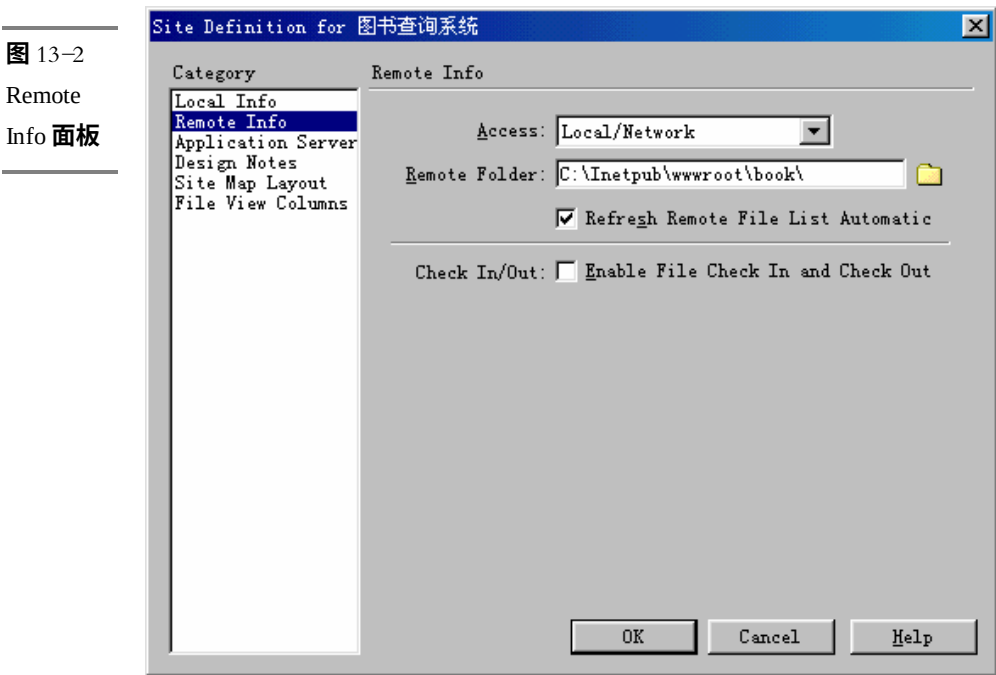


图 13-1
Site
Definition
窗口

然后在 Category 列表中单击 Remote Info 切换到 Remote Info 面板，如图 13-2 所示。由于本例中的本地站点和远程站点位于不同的目录下，因此需要进行文件传输，请在 Access 下拉列表中选择一种传输方式。如果读者的 DW UltraDev 和 Web 服务器也安装在了同一台计算机上，请选择 Local/Network 选项，否则就要选择 FTP 或其他选项了。



在 Category 列表中单击 Application Server 切换到 Application Server 面板，如图 13-3 所示。在 Server Model 下拉列表中选择 ASP 2.0 项，在 Default Scripting 下拉列表中选择 VBScript 或 JavaScript，设置 Default Page extension 下拉列表的值为 .asp。然后在 Access 下拉列表中选择 Local/Network，在 Remote Folder 中输入远程站点的根目录，C:\Inetpub\wwwroot\book，或者使用  按钮定位到该目录。最后在 URL Prefix 文本框中输入 http://localhost/book/。

设置完成后单击 OK 按钮关闭 Site Definition 窗口，这时可能会出现如图 13-4 所示的警告窗口，表示 DW UltraDev 将为这个新站点创建一个缓冲区以提高站点的性能，单击 OK 按钮创建它。

于是站点定义完成。此时站点窗口应该如图 13-5 所示。左边的窗格中显示的是远程站点的文件列表，而右边的窗格显示的是本地站点的文件列表。当然，目前这两个站点的文件都为空。下面我们就往站点中添置一些初始网页。

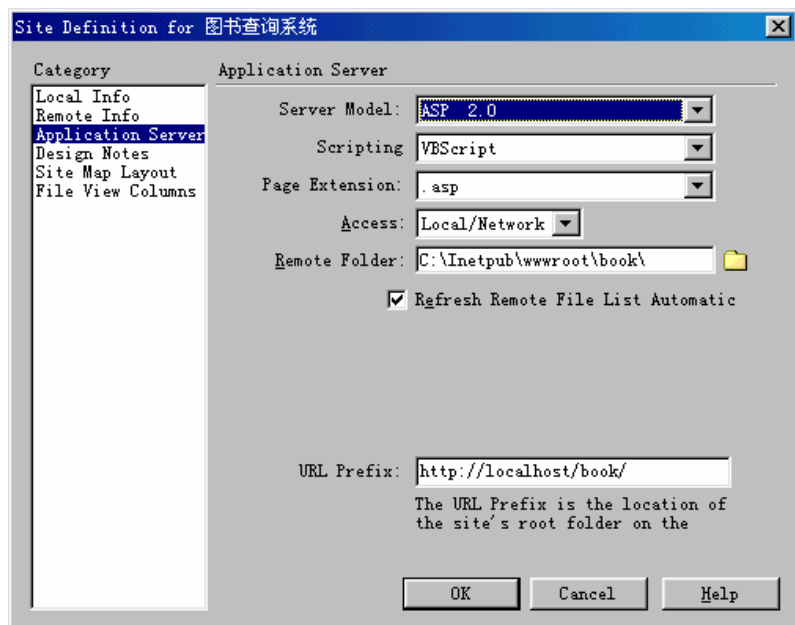


图 13-3
Application
Server 面
板

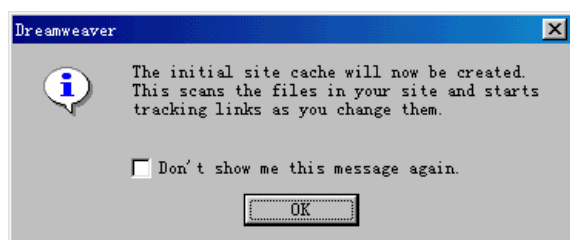


图 13-4
是否创建
缓冲区

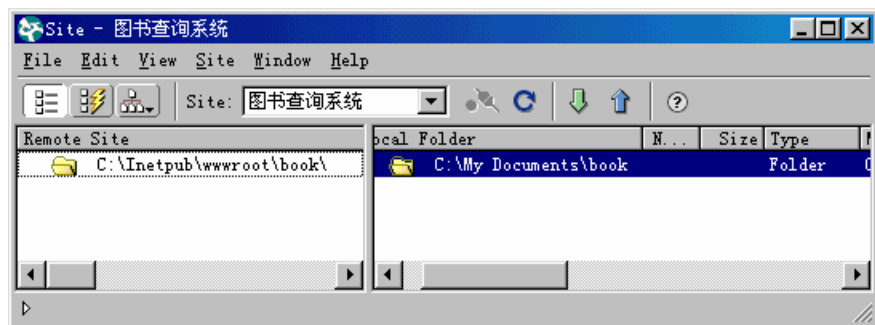
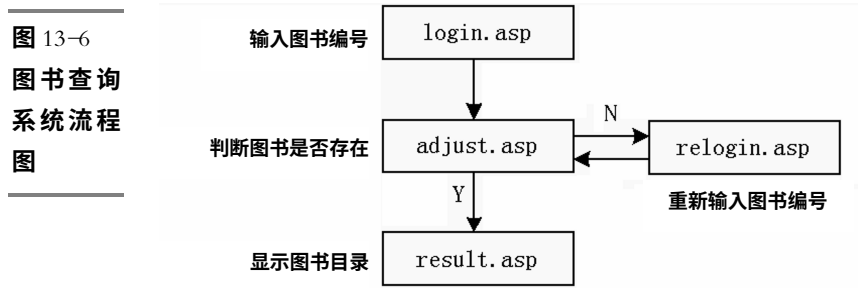


图 13-5
站点窗口

13.1.2 创建初始网页及数据库

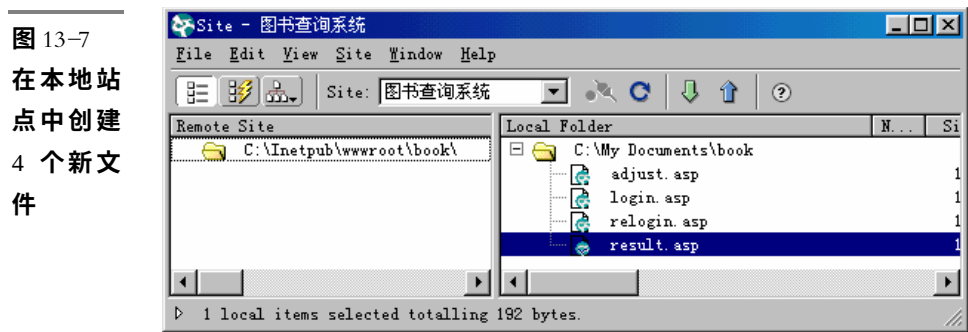
在开始创建初始网页以及数据库之前，我们首先需要了解一下图书查询系统的流

程。本章实现的图书查询系统相对比较简单，其流程图如图 13-6 所示。



login.asp 是一个查询界面，用户需要在此处输入要查询的图书编号，然后提交给服务器。服务器上的 adjust.asp 页面再判断用户所输入的图书编号是否存在，如果存在，则由 result.asp 将查询结果显示出来；如果不存在，则将控制交给 relogin.asp，让用户重新输入图书编号进行查询。

下面我们打开站点窗口，并确信打开了前面创建的“图书查询系统”站点。然后在本地站点的窗格中依次创建本系统所需要的 4 个文件：login.asp、adjust.asp、relogin.asp 和 result.asp。这可以借助“File / New File”菜单命令，或者在本地站点的窗格中单击鼠标右键，在弹出的菜单中选择“New File”菜单命令来实现。此进的站点窗口应该如图 13-7 所示。



接下来双击 login.asp 在文档窗口中打开该文件，然后按照图 13-8 所示在其中加入适当的表单、文字、文本框和按钮，将文档的标题改为“图书查询系统登录界面”。

其中，表单的属性如图 13-9 所示。注意在 Action 文本框中输入 adjust.asp，或者单击位于文本框右边的  图标，在出现的文件选择窗口中选择名为 adjust.asp 的结果页面，然后单击 OK 按钮。在 Method 下拉列表中选择 GET 提交方法，这样就可以使得查询参数能够和 URL 字符串一起传递给服务器，并打开结果页面。

表单中的文本框的属性如图 13-10 所示。注意在 TextField 文本框中把该表单元素的名字改为 bookid。



图 13-8
login.asp
页面



图 13-9
表单的属性

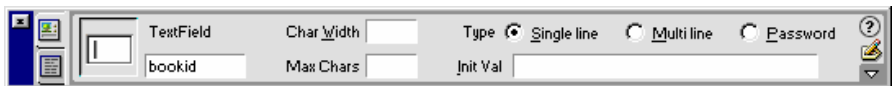


图 13-10
文本框的属性

提交按钮的属性如图 13-11 所示。

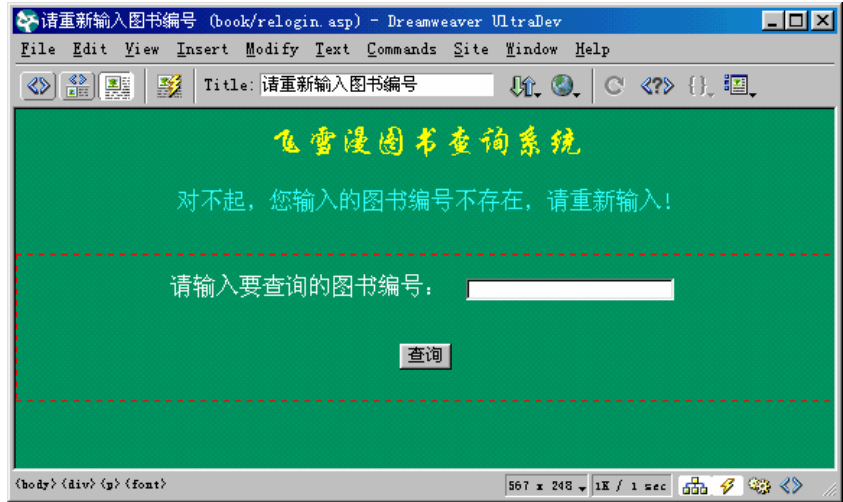


图 13-11
提交按钮
的属性

relogin.asp 页面的内容与 login.asp 的内容是基本类似的，只是 relogin.asp 页面在表单上方加入了一行提示信息，如图 13-12 所示。

因此创建 relogin.asp 的方法很简单，只需将 login.asp 页面中的内容复制下来，然后在文档窗口中打开 relogin.asp 文件，将剪贴板中的内容粘贴下来，最后在表单上方加入相应的文字即可。注意不要修改表单的属性和文本框的属性，也就是说，要使 relogin.asp 页面中的表单和文本框的属性与 login.asp 页面中的表单和文本框的属性保持一致。

图 13-12
relogin.asp
页面



result.asp 的初始页面就更加简单了, 如图 13-13 所示。其中只包含了一个 2 行 8 列的表格, 用来显示满足用户条件的图书记录。在表格的下方有一个返回到查询页面的超链接, 请注意将其 Link 属性的值设为 login.asp 即可, 如图 13-14 所示。

另外一个页面——adjust.asp 只是用来进行后台处理, 并不会在页面中显示任何东西, 因此现在可以不用管它。

图 13-13
result.asp
页面

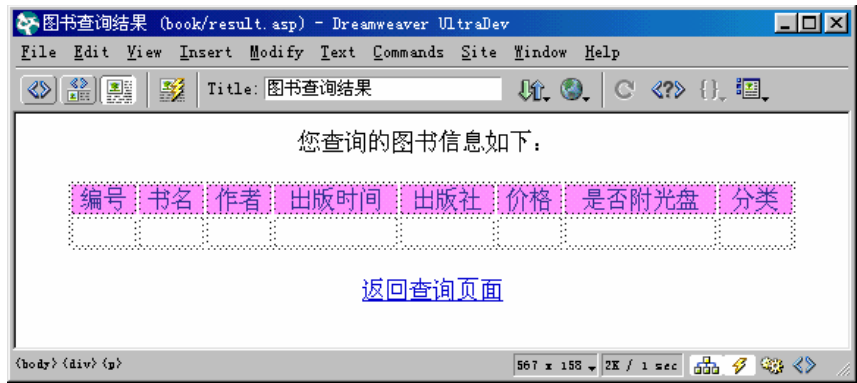


图 13-14
超链接的
属性



本示例所使用的数据库照例使用 Access 2000 来创建, 其内容如图 13-15 所示。

ID	title	author	date	press	price	disk	type
1	Visual C++ 5.0 用户指南	俞俊平, 余安萍	1998年5月1日	西安电子科技大学出版	30	☐	程序设计类
2	Visual C++ 5.0 编程指南	俞俊平, 余安萍	1998年8月1日	西安电子科技大学出版	35	☐	程序设计类
3	精通 Visual C++ 6.0	王晖, 余安萍, 王	1999年8月1日	电子工业出版社	40	☐	程序设计类
4	精通 Authorware 5.4	俞俊平, 余安萍	2000年4月1日	电子工业出版社	65	☒	多媒体类
5	Window NT/2000 建站实务	俞俊平, 余安萍	2001年2月1日	电子工业出版社	40	☒	网站设计类
6	Dreamweaver UltraDev 4 网站开发宝典	俞俊平, 余安萍	2001年5月1日	电子工业出版社	30	☐	网站设计类

图 13-15
数据库的
内容

此数据库一共包括 8 个字段, 分别是: ID(图书编号)、title(书名)、author(作者)、date(出版时间)、press(出版社)、price(价格)、disk(附光盘)、type(分类)。为了节省篇幅, 我们在此不打算详细讲述该数据库的创建过程。读者可以在光盘的 book 目录下找到这个名为 book.mdb 的数据库文件, 在学习的时候可以直接使用。

13.2 定义 DSN 和数据库链接

数据库建立好之后, 要设定系统的 DSN(数据源名称), 才能让网页可以知道数据库所在的位置以及数据库相关的属性。使用 DSN 的好处是, 如果移动数据库文件的位置, 或是换成别种类型的数据库, 只要重新设定 DSN 就可以了, 不需要去修改原来使用的程序。然后就可以利用这个 DSN 在页面中创建数据库链接。

13.2.1 定义 DSN

由于本章的示例使用的是 MS Access 数据库, 因此需要先创建一个使用 MS Access Driver 的 ODBC 系统 DSN。

具体的创建步骤同 11.1.1 节中所介绍的类似, 只是在如图 13-16 所示的“ODBC Microsoft Access 安装”窗口中, 注意在“数据源名”文本框里输入“book”, 并通过“选取”按钮, 找到 book.mdb 数据库文件所在的路径。

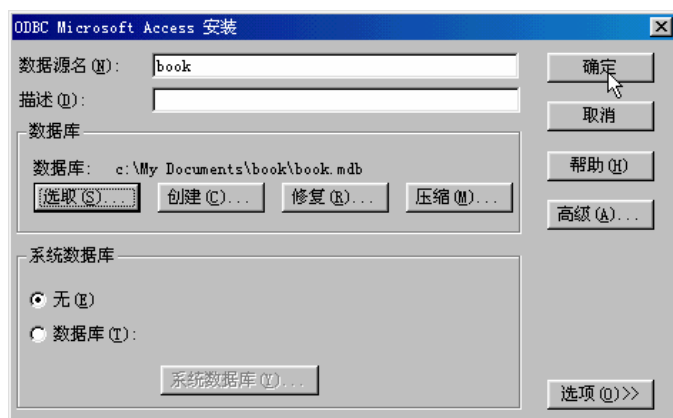


图 13-16
“ODBC
Microsoft
Access 安
装”窗口

13.2.2 创建数据库链接

接下来需要创建数据库链接了。

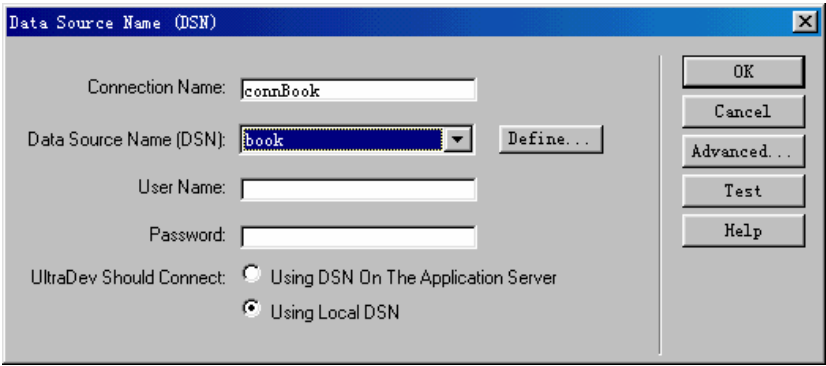
确信在站点窗口中打开了“图书查询系统”站点，然后在 DW UltraDev 文档窗口中选择“Modify / Connections”菜单命令，打开 Connections 窗口，如图 13-17 所示。此时 Connections 窗口列表中应该为空。

图 13-17
Connections
窗口



然后单击窗口中的 New 按钮，在弹出的快捷菜单中选择 Data Source Name(DSN) ”菜单命令，打开 Data Source Name(DSN) 窗口。设置 Connection Name 文本框的值为 connBook，然后在 Data Source Name(DSN) 下拉列表(该列表中列出了系统中的所有 ODBC DSN) 中选择 book，并保持 Username 和 Password 这两个文本框的值为空。如图 13-18 所示。

图 13-18
创建
AndesExtr-
anet 数据
库链接



单击 Test 按钮对此数据库链接进行测试。如果数据库链接创建成功，则会弹出一个对话框，其中显示“ Connection was made successfully. ”的提示信息，如图 13-19 所示。单击“ 确定 ”按钮关闭提示窗口，然后单击 OK 按钮保存所创建的数据库链接。于是在 Connections 窗口的列表中就会显示出刚才创建的数据库链接名，如图 13-20 所示。

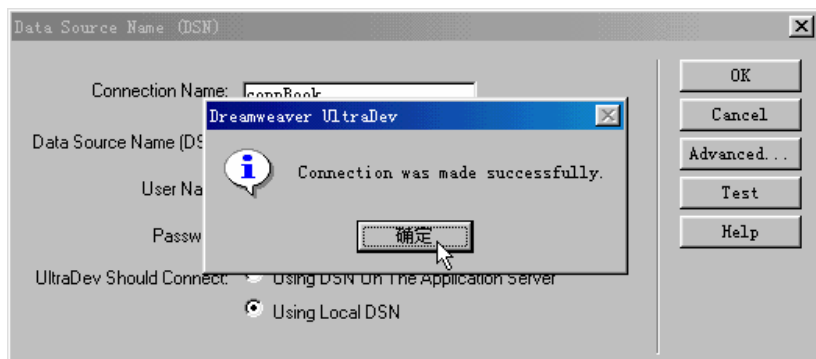


图 13-19
测试数据
库链接

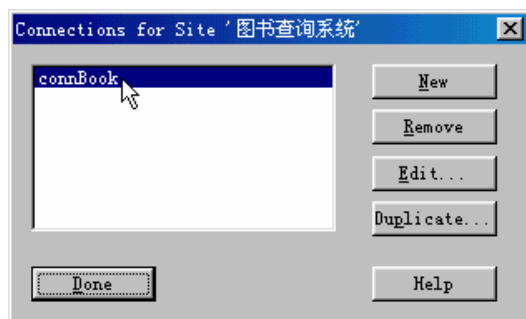


图 13-20
数据库链
接创建成
功

13.3 添加动态内容

现在一切就绪，我们可以开始在网页中添加动态内容，以实现查询系统。

13.3.1 创建判断页

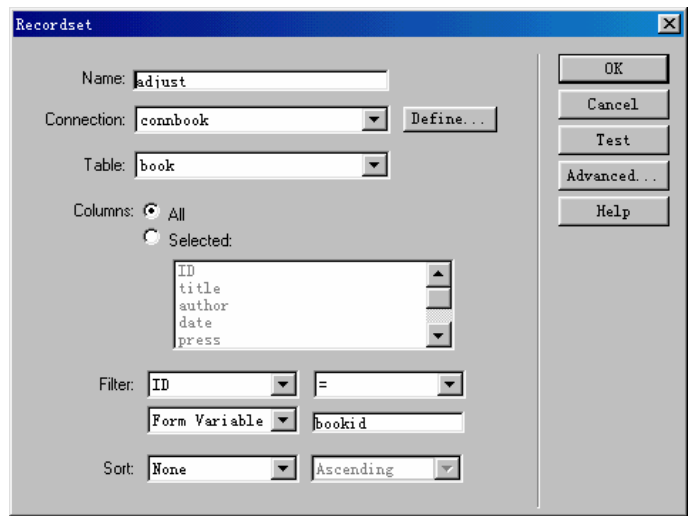
目前我们还没有对 `adjust.asp` 作任何处理，下面就来实现 `adjust.asp`。

在站点窗口的列表中双击 `adjust.asp`，在文档窗口中打开它。然后为其创建记录集。

在 Data Bindings 面板中单击 **+** 按钮，然后在出现的菜单中选择 Recordset(Query) 菜单项，于是出现简单版本的 Recordset 窗口(如果出现的是高级版本的 Recordset 窗口，请单击 Simple 按钮切换到简单版本的 Recordset 窗口)。在 Name 文本框中输入 `adjust`，然后在 Connection 下拉列表中选择 `connBook`，在 Table 下拉列表(其中显示的是数据库中的所有表名) 中，选择 `book` 项，并在 Columns 域中，选择 All 单选钮选中所有字段。在 Filter 域的第一个下拉列表中选择“ ID ”字段，确信第二个下拉列表的值为“ = ”，在

第三个下拉列表中选择“ Form Variable ”项，在最后一个文本框中输入“ bookid ”(不包括引号)，如图 13-21 所示。

图 13-21
设置后的
Recordset
窗口



设置完成后，单击 Test 按钮即可对新建的记录集进行测试，这时应该出现如图 13-22 所示的提示窗口，让读者输入进行测试的值。如果在其中输入数字“ 4 ”，然后单击 OK 按钮，则会出现如图 13-23 所示的测试窗口，其中显示了满足条件的所有记录。由于我们输入的是 4，因此测试窗口中会把记录集中的第 4 条记录显示出来。

图 13-22
输入测试
值

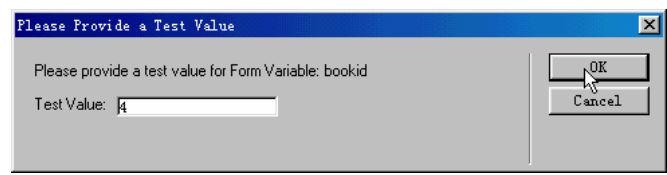
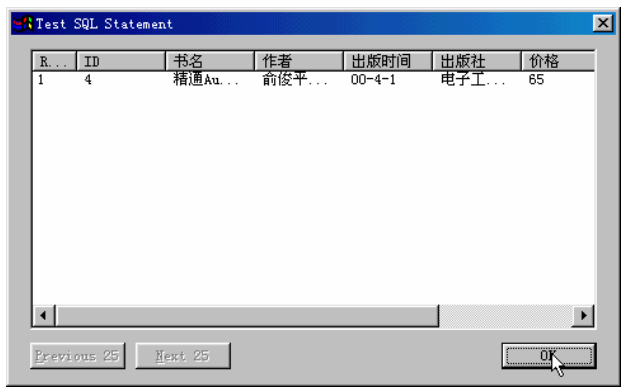


图 13-23
记录集测
试结果



单击 OK 按钮关闭测试结果窗口，然后可以在 Recordset 窗口中单击 Advanced 按钮切换到高级版本的 Recordset 窗口，这时可以在窗口的 SQL 编辑域中看到系统自动为我们添加的 SQL 语句“ SELECT * FROM book WHERE ID = MMColParam ”，同时还自动为我们添加了一个变量 MMColParam，如图 13-24 所示。单击 Simple 按钮返回到简单版本，然后单击 OK 按钮关闭 Recordset 窗口，从而将该记录集增加到页面中。

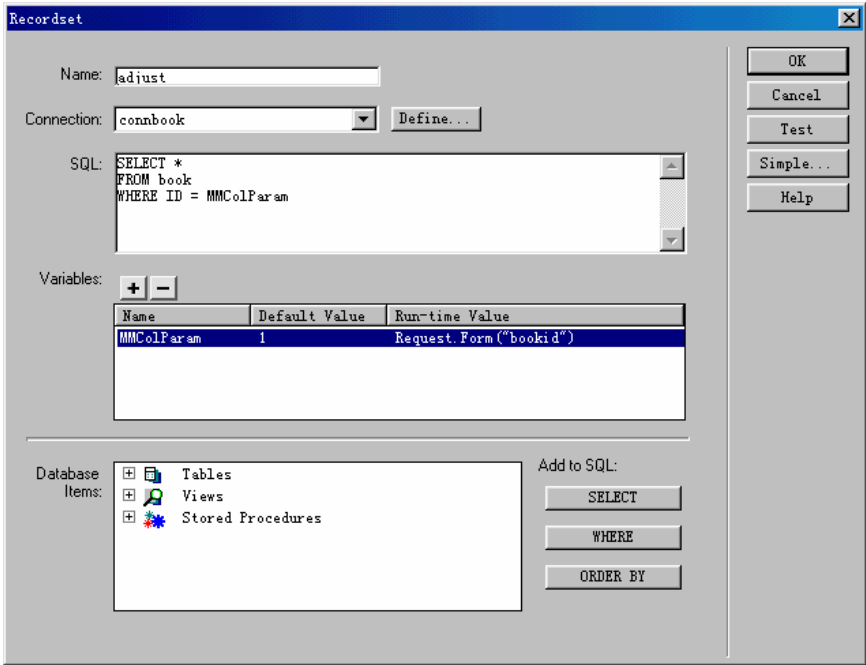


图 13-24
高级版本
的
Recordset
窗口

接下来打开 adjust.asp 的 HTML 源代码检查器窗口，或者在文档窗口中打开相应的代码视图，在“ <HTML> ”的前面加上下面所示的代码：

```
<%  
    If Not adjust.EOF Then  
        Session( "id" )=Request.Form( "bookid")  
        Response.Redirect "result.asp"  
    Else  
        Response.Redirect "relogin.asp"  
    End If  
%>
```

这段代码的作用是判断用户输入的图书编号是否存在,如果存在,则跳转到 result.asp 页面中将结果显示出来；如果不存在，则跳转到 relogin.asp 页面中重新进行输入。值

得注意的是上面代码中加粗的部分，它将用户输入的文本框中的值保存在了一个 Session 变量中。那么这个变量是作什么用的呢？等会儿创建结果页的时候读者就会知道了。

最后保存 adjust.asp 页面。



如果在上面的操作过程中出现了如图 13-25 所示的警告窗口，请不要管它。单击“确定”按钮即可。

图 13-25
警告窗口



13.3.2 创建结果页

最后需要创建 result.asp 结果页面。在站点窗口中双击 result.asp，在文档窗口中打开它。然后为其定义记录集。

在 Data Bindings 面板中单击 按钮，然后在出现的菜单中选择 Recordset(Query) 菜单项，于是出现简单版本的 Recordset 窗口(如果出现的是高级版本的 Recordset 窗口，请单击 Simple 按钮切换到简单版本的 Recordset 窗口)。在 Name 文本框中输入 result，然后在 Connection 下拉列表中选择 connBook，在 Table 下拉列表中，选择 book 项，并在 Columns 域中，选择 All 单选钮选中所有字段。在 Filter 域的第一个下拉列表中选择“ ID ”字段，确信第二个下拉列表的值为“ = ”，在第三个下拉列表中选择“ Session Variable ”项，在最后一个文本框中输入“ id ”(不包括引号)，如图 13-26 所示。

请读者注意我们在 Filter 域的第三个下拉列表中选择了“ Session Variable ”项，它的作用就是把记录集中 ID 字段的值和一个 Session 变量进行比较，而这个 Session 变量恰好就是我们在前面保存过的 Session("id")。由于 Session("id") 中保存的是用户在查询表单中输入的图书编号，因此在此处可以根据 Session("id") 的值来确定将要显示的图书信息，从而达到查询目的。

单击 OK 按钮关闭 Recordset 窗口，此时在 Data Bindings 中的记录集应该如图 13-27 所示。

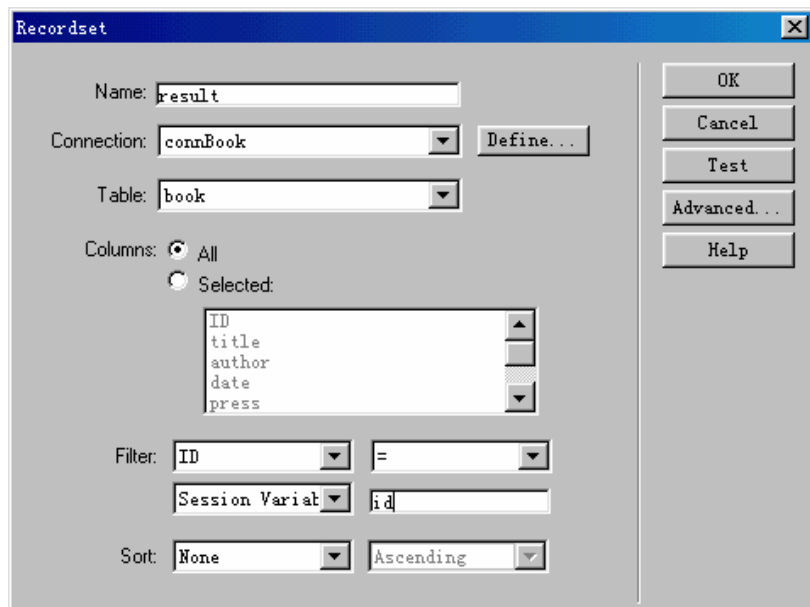


图 13-26
为结果页面
创建记录
集

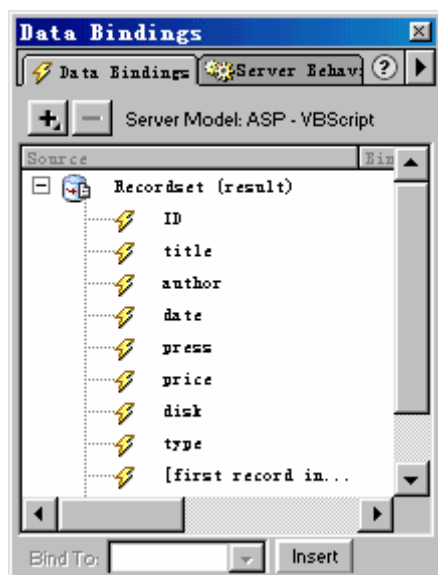
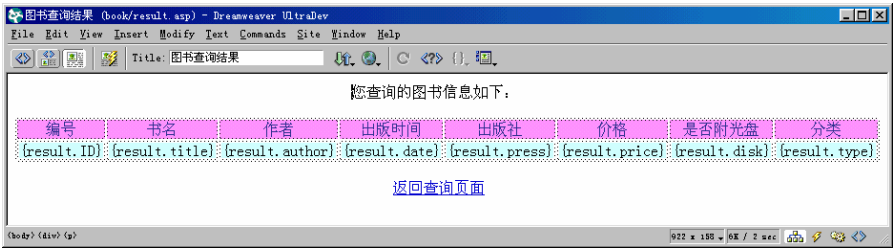


图 13-27
结果页面
的记录集

接下来——地将记录集中的各字段拖放到结果页面的相应表格单元中，例如，将 ID 字段拖放到“编号”下，将 title 字段拖放到“书名”下，依此类推。最后的结果如图 13-28 所示，表格中的第二行都被相应的占位符代替了。

最后保存所做的工作。

图 13-28
在结果页面中添加动态内容

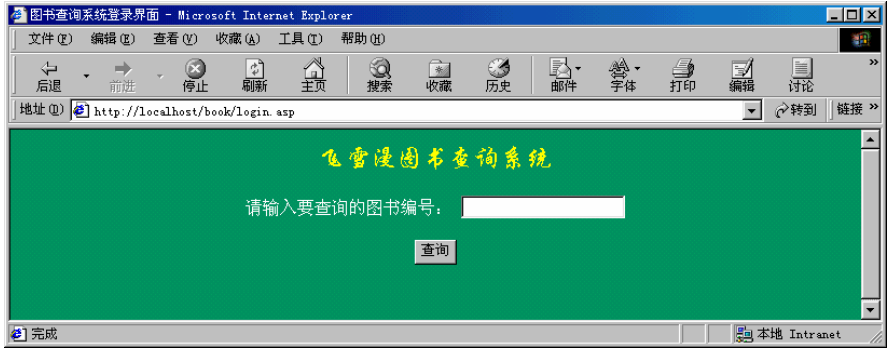


13.3.3 查询系统测试

至此，一个简单的图书查询系统就设计完成了。现在请使用站点窗口把本地站点上的所有文件都上传到远程站点中，然后打开浏览器窗口进行测试。

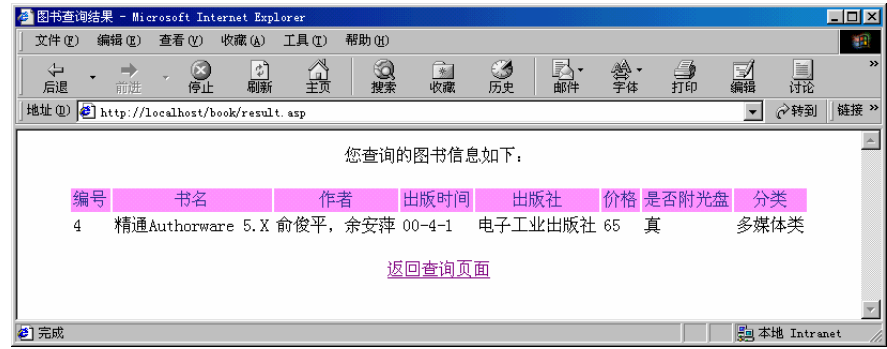
在地址栏中输入：http://localhost/book/login.asp，于是出现如图 13-29 所示的页面。

图 13-29
查询界面



请在文本框中输入要查询的图书编号，如‘ 4 ’。然后单击‘ 查询 ’按钮，于是便会出现如图 13-30 所示的结果页面。

图 13-30
查询结果



如果用户在图 13-29 的查询界面中输入的图书编号不正确，例如如果输入“ 8 ”（由

于我们的数据库中一共只有 6 条记录, 所以不存在第 8 条记录), 并单击“ 查询 ”按钮, 则会出现如图 13-31 所示的提示窗口, 让用户重新输入。

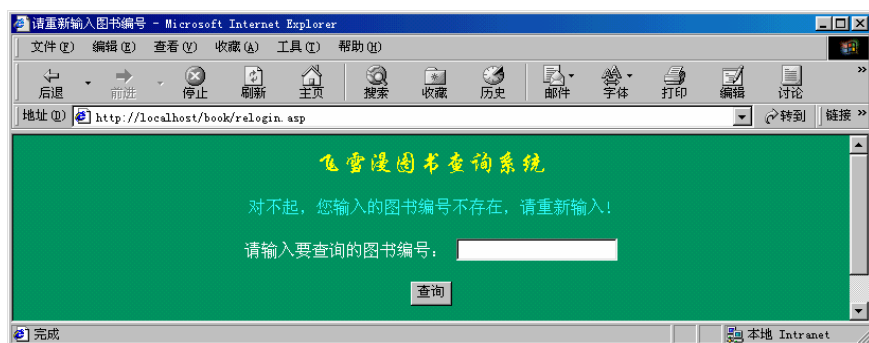


图 13-31
重新输入
图书编号

另外, 读者可能会发现, 如果在查询时没有在文本框中输入任何数值, 单击“ 查询 ”按钮时也能查出结果, 但总是显示第 1 条记录, 这是为什么呢?

原来, UltraDev 在 `adjust.asp` 页面中自动为我们新增了一个变量 `MMColParam`, 这个的作用就是存储用户在文本框中输入的值, 而其缺省值为 1。也就是说, 即使用户没有在文本框中输入任何数值, 系统也会认为用户要查询第 1 条信息。

如果要想修改这个问题, 可以这样处理: 把用户没有输入的情况看作用户输入错误处理, 即显示图 13-31 所示的窗口, 让用户重新输入图书编号。那么怎样在程序中实现呢?

其实非常简单, 只需在图 13-24 所示的高级版本的 Recordset 窗口中将变量 `MMColParam` 的缺省值改为 0 或任意一个负数即可。其中的原因就不用我们分析了吧!

第 四 部 分

DW UltraDev 4 高级应用与扩展

- ◆ 第 14 章 插件的安装与使用
- ◆ 第 15 章 定制 DW UltraDev 4 界面
- ◆ 第 16 章 DW UltraDev 4 的 DOM 模型
- ◆ 第 17 章 DW UltraDev 4 API 及其使用

.....

第 14 章 插件的安装与使用

本章概要：

- ☒ 插件概述
- ☒ 安装插件
- ☒ 管理插件

DW UltraDev 广为流传的另一个原因是它具有极强的扩展性。各种各样 DW UltraDev 所没有的功能都可以通过插件的形式添加到 DW UltraDev 中。本章我们就来看一看如何安装、使用和管理插件。

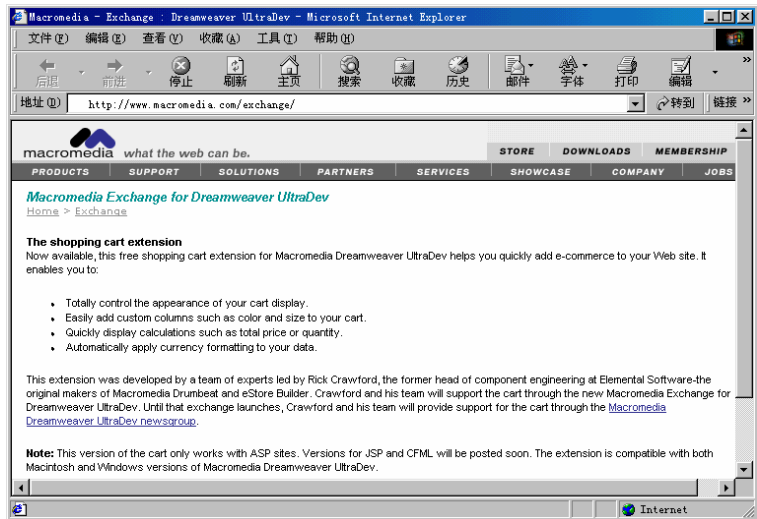
14.1 插件概述

插件实际上是一种特殊的软件，它可以增加到 Macromedia 软件中并增强该软件的功能。插件的种类比较多，例如在 Dreamweaver 和 DW UltraDev 中，插件可以是一段 HTML 代码，可以在对象对面和 Insert 菜单中增加对象；也可以是一段 JavaScript 命令，可以在 Command 菜单中增加新菜单命令，增加新的行为、属性面板和浮动面板等。

一些插件是由 Macromedia 公司创建的，而另一些插件则是由第三方开发的。为了方便广大 Macromedia 公司产品的用户，Macromedia 公司在互联网上创建一个名为 Macromedia Exchange 的站点，其 URL 网址为 <http://exchange.macromedia.com/> 或 <http://www.macromedia.com/exchange/>。该网站可以说是一个各种类型的插件大仓库。如果用户在该网站上找到了一个感兴趣的插件，都可以免费地将其下载下来，然后安装到本地的应用中。图 14-1 中所示的就是 Macromedia Exchange 站点的页面。

通常，Macromedia 的应用软件都是使用一个名为 Extension Manager 的插件管理器来对插件进行管理的。该插件管理器为广大用户安装、删除插件以及获取已经安装插件的信息等提供了一个简便快捷的方法。

图 14-1
Macromedia
Exchange 站
点的页面



14.2 安装插件

当我们在网上找到了一个合适的插件以后，就可以将插件下载下来，并安装到 DW UltraDev 中。通常把我们所下载的文件称为插件打包文件，它的后缀名为.mxp。在这个打包的文件中包含了使用该插件所需要的所有文件。

为了方便广大读者，我们在本书所附的光盘上就附带了一个名为 ud_cart.mxp 的插件打包文件。下面我们就以 ud_cart.mxp 插件为例，来介绍如何安装该插件到 DW UltraDev 系统中。

14.2.1 安装插件

启动插件的安装过程通常有三种方法：

第一种方法是在 Windows 的资源管理器中找到需要安装的插件打包文件，然后用鼠标左键双击该文件，于是 Extension Manager 插件管理器窗口就会自动打开，并且同时会开始插件的安装过程。

第二种方法是在 Windows 的资源管理器中找到需要安装的插件打包文件，然后在其上单击鼠标右键，于是会弹出一个快捷菜单，在该菜单中选择“Install Extension”菜单命令，如图 14-2 所示。

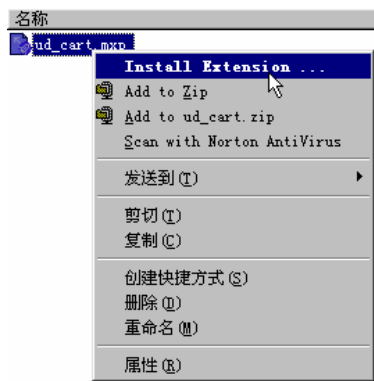


图 14-2
通过快捷
菜单安装
插件

第三种方法是在 DW UltraDev 中，选择“Commands / Manage Extensions”菜单命令打开 Extension Manager 插件管理器，图 14-3 中所示的就是打开的 Extension Manager 插件管理器窗口。

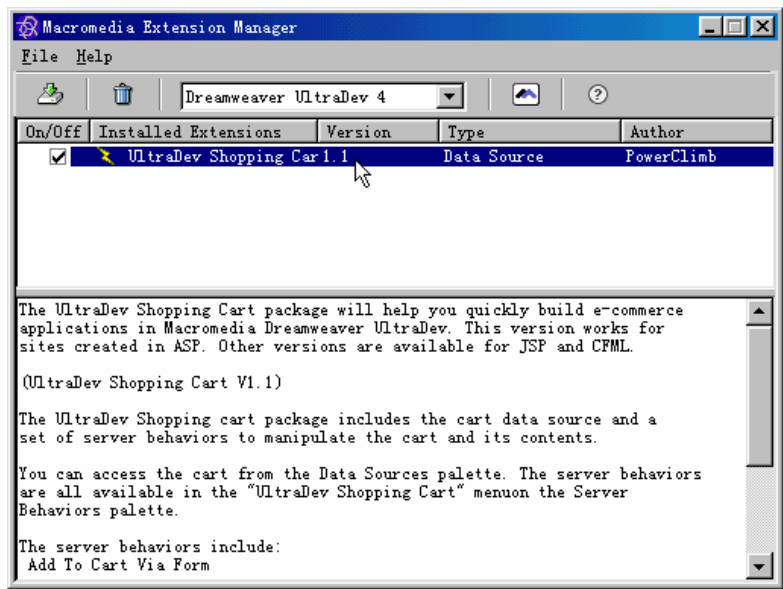
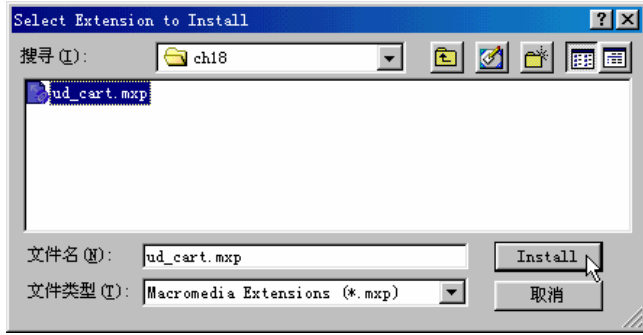


图 14-3
Extension
Manager
插件管理
器窗口

请确认在 Extension Manager 窗口的工具栏的下拉列表中选择了 Dreamweaver UltraDev，表示将把插件安装到 DW UltraDev 中。之所以要进行选择，是因为 Extension Manager 插件管理器是许多 Macromedia 软件所共用的插件管理器，例如 Dreamweaver 也是通过这个 Extension Manager 插件管理器来对插件进行管理的。

然后请选择 Extension Manager 窗口中的“File / Install Extension”菜单命令，或者选择工具栏上的  按钮，于是会出现如图 14-4 所示的文件选择窗口。在其中找到需要安装的插件打包文件，然后单击 Install 按钮即可开始安装。

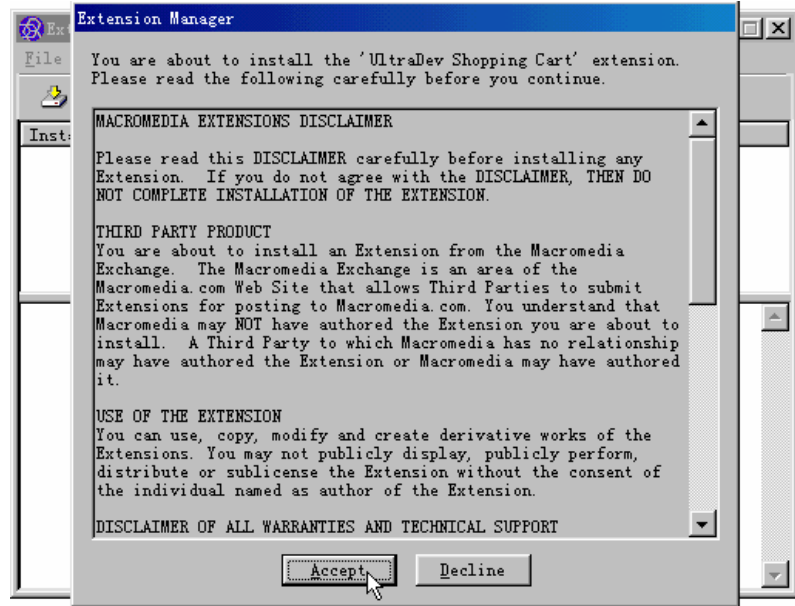
图 14-4
选择插件
打包文件



由于 DW UltraDev 已经包括了 Dreamweaver 的所有功能,因此,为 Dreamweaver 所设计的插件也能够用于 DW UltraDev 中。但是,在将一个为 Dreamweaver 所设计的插件安装到 DW UltraDev 时,必须采取上面所介绍的第三种方法。即首先打开 Extension Manager 窗口,然后在工具栏的下拉列表中选择 Dreamweaver UltraDev,最后选择“File / Install Extension”菜单命令进行安装。

不管采用上述三种方法的哪一种,最终所得的结果都是一样的,那就是会出现如图 14-5 所示的安装界面。该窗口中列出的是一些关于 Macromedia 插件的声明,请单击 Accept 按钮接受这些声明,以便开始正式的安装过程。

图 14-5
插件安装
界面



如果读者在系统中已经安装了另外一个版本的相同的插件，或者新安装的插件与某个已经安装的插件中同名，则安装程序会弹出一个对话框询问是否首先删除系统中已经安装的插件。如果单击 Yes 按钮，则老的插件将被删除而新的插件将被安装；如果单击 No 按钮，则安装过程将取消。

如果没有遇到其他任何问题，则通常会弹出一个如图 14-6 所示的对话框，表明插件已经正确地安装到了系统中。

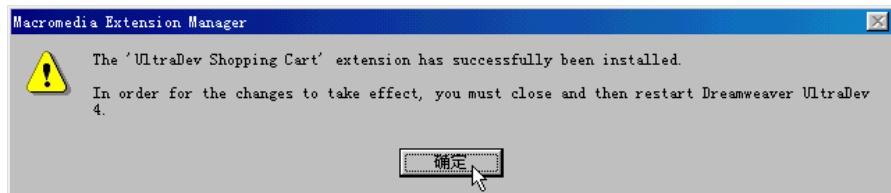


图 14-6
插件已经
正确地安
装到了系
统中



在开始安装一个插件之前，Extension Manager 管理器会对 DW UltraDev 中的系统菜单所使用的 menus.xml 文件创建一个名为 menus.xbk 的备份。这样做的目的是，如果万一在安装过程中插件将原来的 menus.xml 文件破坏了，导致 DW UltraDev 无法正常工作，我们就可以通过用备份的 menus.xbk 文件替换掉现在的 menus.xml 文件，以便系统恢复到原来的状态。

如果要想使安装的插件生效，通常需要关闭 DW UltraDev 应用程序，然后重新打开它。

14.2.2 测试插件

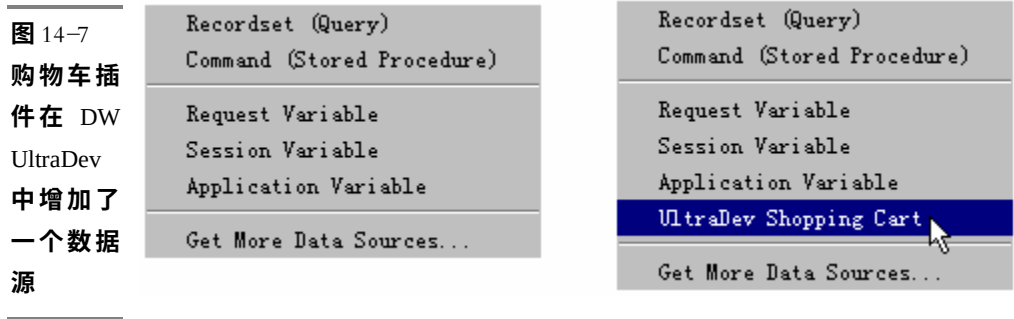
我们为读者所提供的名为 ud_cart.mxp 的插件是一个购物车插件。使用这个插件我们可以非常轻松地在 DW UltraDev 创建一个时下非常流行的电子商务应用。它可以实现的功能包括：

- ✧ 整体上控制购物车的外观。
- ✧ 轻松地对购物车的属性进行定制，如增加颜色和大小字段等。
- ✧ 快速计算并显示所购商品的数量、单价、总价等数据。
- ✧ 自动将货币数据格式应用到数据中。

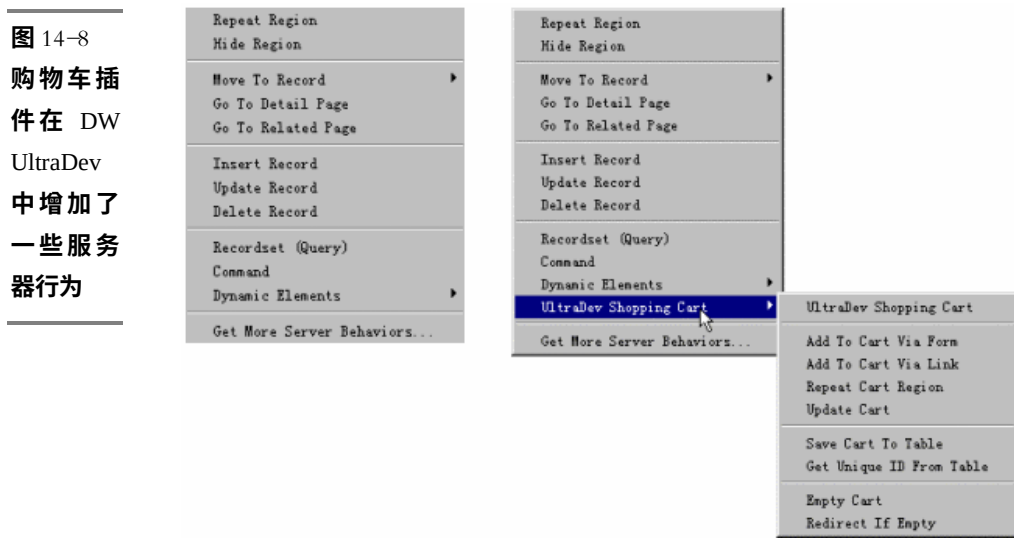
假设读者刚才已经将该插件安装到了 DW UltraDev 中，并且关闭并重新打开了 DW UltraDev，现在我们来看一下这个插件给 DW UltraDev 带来了什么样的变化。

首先，这个购物车插件会在 DW UltraDev 中增加一个数据源。打开 Data Bindings

面板，然后单击其中的按钮，读者就会在下拉菜单中列出的数据源中找到新增加的“ UltraDev Shopping Cart ”菜单项，如图 14-7 所示。其中左图是原来的数据源菜单，而右图是安装了购物车插件后的数据源菜单。



另外，和这个新增的数据源相配合，购物车插件还增加了一系列的服务器行为来处理购物车及其中的内容。打开 Server Behaviors 面板，然后单击其中的按钮，读者就会在下拉菜单中列出的服务器行为中发现新增加的“ UltraDev Shopping Cart ”子菜单项，如图 14-8 所示。其中左图是原来的服务器行为菜单，而右边是安装了购物车插件后的服务器行为菜单。



- 从图中可以看出，购物车插件增加了下列服务器行为：
- ※ Add To Cart Via Form：通过表单增加到购物车中。
 - ※ Add To Cart Via Link：通过超链接增加到购物车中。

- ※ Repeat Cart Region：重复购物车区域。
 - ※ Update Cart：更新购物车。
 - ※ Save Cart To Table：将购物车中的数据保存到数据库表中。
 - ※ Get Unique ID From Table：从数据库表中得到商品的惟一 ID 标识符。
 - ※ Empty Cart：清空购物车。
 - ※ Redirect If Empty：如果为空则对页面进行重定向。
- 感兴趣的读者不妨试一试使用购物车插件来创建一个网上商店的 Web 应用。

14.3 管理插件

14.3.1 插件管理器的主要功能

Extension Manager 插件管理器不仅可以轻松地安装插件，还可以完成许多其他的管理工作，如删除已安装的插件、对插件进行排序、下载更多的插件、提交新插件、报告插件的 Bug 等。

值得注意的是，Extension Manager 插件管理器中并不会显示出所有在 Extension Manager 插件管理器安装之前已经安装的插件，如已经安装到 DW UltraDev 系统中的插件。因此通常读者在第一次打开 Extension Manager 插件管理器时会发现窗口中的插件列表为空。

14.3.2 对插件进行排序

对插件进行排序的方法很简单。和在资源管理器中对文件进行排序一样，如果要对插件进行排序，只需在窗口列表中单击相应的排序按钮即可，如图 14-9 所示。

On/Off	Installed Extensions	Version	Type	Author
--------	----------------------	---------	------	--------

图 14-9
排序按钮

从图中可以看出，对插件排序可以采取以下几种方法：

- ※ On/Off：根据插件是允许还是禁止来排序。
- ※ Installed Extensions：根据插件的名字来排序。
- ※ Version：根据插件的版本来排序。

- ※ Type：根据插件的类型来排序。
 - ※ Author：根据插件的作者来排序。
- 例如,如果要以插件的作者的字母为序对插件进行重新排列,只需单击其中的 Author 按钮即可；如果要逆序排列，只需再次单击该排序按钮即可。

14.3.3 激活或禁止插件

如果用户通过 Extension Manager 插件管理器在系统中安装了数量众多的插件，则 UltraDev 在启动时就要调用更多的 XML 语句来启用这些插件，通常这会大大影响 UltraDev 的性能。为了解决这个问题，Extension Manager 插件管理器可以让我们激活或禁止某些插件。

从图 14-10 中可以看出，在每一个插件名的左边、On/Off 列中都有一个复选框。如果该复选框被选中，则表明相应的插件被激活了；反之，如果该复选框没有被选中，则表明相应的插件被禁止了。缺省情况下，当一个插件被安装到系统中时，它是被激活的。

图 14-10
允许或禁止插件

On/Off	Installed Extensions	Version	Type	Author
☒	UltraDev Shopping Car	1.1	Data Source	PowerClimb
☒	Sound	2.2.1	Object	Rabi Sunder Raj

图 14-11 中所示的 Extension Manager 插件管理器列表中有两个插件，其中上面一个插件 UltraDev Shopping Car 就被禁止了，而下面一个插件 Sound 是被激活的。

图 14-11
激活或禁止插件的示例

On/Off	Installed Extensions	Version	Type	Author
☐	UltraDev Shopping Car	1.1	Data Source	PowerClimb
☒	Sound	2.2.1	Object	Rabi Sunder Raj

当一个插件被禁止时，它会被系统重新进行打包，并存储在 UltraDev 主程序的 Configuration\Extensions\Disabled 目录下，以便将来可以被允许重新使用。

在激活或禁止插件的操作过程中，如果 UltraDev 已经运行了，则系统还会提示一个如图 14-12 所示的对话框，表示只有关闭当前运行的 UltraDev 并重新启动它，才可以使激活或禁止插件的操作生效。通常只需在对话框中单击“确定”按钮，然后重新启动 UltraDev 就可以看到效果。

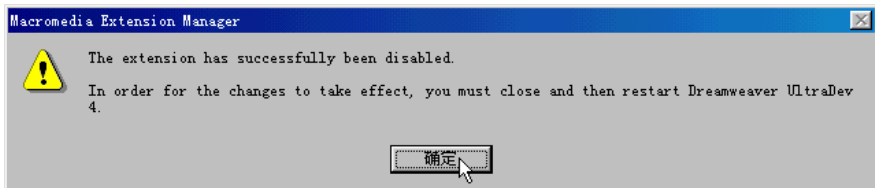


图 14-12

激活或禁
止插件时
的提示框

14.3.4 删除插件

如果要删除某个已经安装的插件，请在列表中选择该插件，然后选择 **File / Remove Extension** 菜单命令，或者单击工具栏上的  按钮，于是会出现一个如图 14-13 所示的对话框，确认是否真的要删除该插件。如果单击了“是”按钮，即可将插件删除。

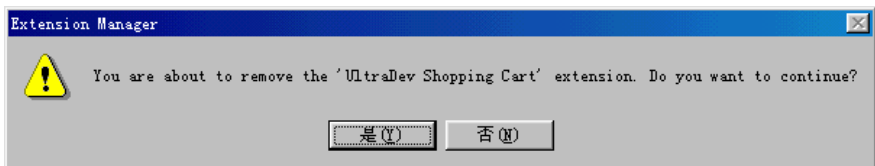


图 14-13

是否要删
除插件



如果用户想要删除一个被禁止的插件，请一定要使用上面的方法进行删除，而不要直接手工删除位于 `Configuration\Extensions\Disabled` 目录下的插件文件，否则，系统就不会对 `menus.xml` 文件作适当的修改，从而可能会造成一些错误。

如果要查找有关一个插件的更多信息，可以选择 **File / Go to Macromedia Exchange** 菜单命令，或者单击工具栏上的  按钮，于是系统就会在浏览器中打开 Macromedia Exchange 网站（假设读者已经连接到了互联网）。在该网站上你不仅可以看到所有其他插件的详细信息，也可以对自己使用过的插件做出适当的评论（甚至可以及时发现某个插件中的 Bug，给出修改意见），以供其他用户参考，当然还可以查找并下载更多的插件。

如果读者对 JavaScript、VBScript、Java、ASP 或 ColdFusion 等非常熟悉的话，还可以自己动手编写一些实用的插件，并上传到 Macromedia Exchange 网站中，为广大 DW UltraDev 用户贡献你的一份力量。

最后，作为一个示例，我们再给读者介绍一下如何使用一个实际的声音插件。

14.4 一个声音插件的使用

14.4.1 安装声音插件

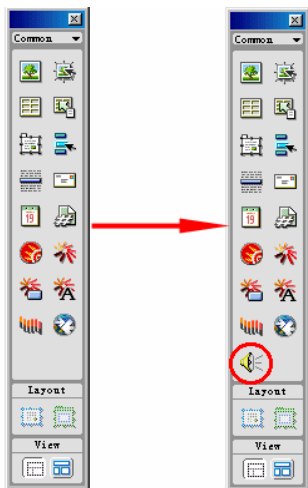
在 UltraDev 中可以安装的插件类型是非常多的。在上面的介绍中，我们安装的 ud_cart.mxp 插件是一个数据源类型 (Data Source) 的插件，插件的类型读者可以从图 14-11 中所示的 Type 列中查看出来，从该图中可以看出，除了 Data Source 类型的插件以外，我们还安装了一个名为 Sound 的 Object (对象) 类型的插件。下面我们介绍这个声音插件的使用。

同样我们在光盘上提供了这个名为 Sound.mxp 的插件，它的作用是可以往页面中插入背景音频文件。大家在实际使用过程中可能已经感觉到了，UltraDev 对于音频文件的支持，好像有天生的“缺陷”。而这个插件可以很好地支持声音文件。

这个插件的安装方法同 ud_cart.mxp 插件的安装方法是完全一样的 (实际上所有的插件的安装方法都是一样的)。读者将 Sound.mxp 插件安装到系统中以后，重新启动 UltraDev，就会发现 UltraDev 又发生了某些变化。

既然 Sound.mxp 插件是一个对象类型的插件，起变化的当然应该是对象面板了。打开对象面板，读者会发现其中多了一个  按钮，如图 14-14 所示。其中左图是安装 Sound.mxp 插件之前的对象面板，右图是安装了 Sound.mxp 插件之后的对象面板。可以看出 UltraDev 中现在多了一个处理声音的对象，下面我们就介绍如何通过这个新增的对象在页面中加入声音。

图 14-14
对象面板
发生变化



14.4.2 使用声音插件

首先打开要处理声音的页面，然后在对象面板中选择  按钮。于是系统弹出一个如图 14-15 所示的对话框。单击 Browse 按钮打开文件选择窗口，在其中选择要插入的声音文件。这个声音插件可以使用 MIDI、WAV 和 AIFF 这三种格式的声音文件。

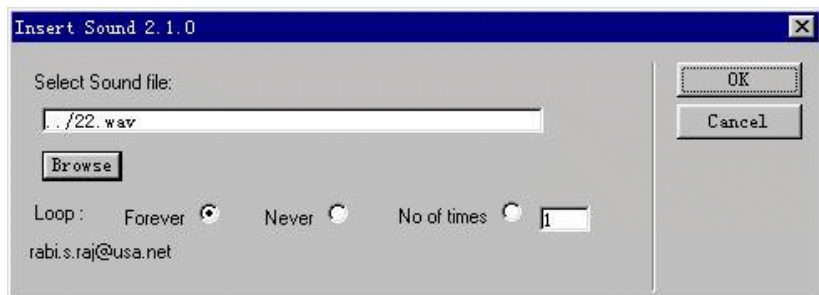


图 14-15
插入声音
文件

Loop 域用来控制循环次数。其中，Forever 为永久，Never 为不循环，No of times 表示循环次数，具体次数请在后面的文本框中输入。

设置完成后单击 OK 按钮，于是页面中就会出现一个  形状的对象，选择该对象，还可以在属性面板中修改它的设置。

保存该网页，然后按下 F12 快捷键在浏览器中预览。浏览器在调入该页面后，就会自动播放刚才添加的声音文件。

从这个例子中，读者可以看到插件对增强 UltraDev 性能的巨大作用。现在在互联网上有许多这样的免费插件可以供大家使用，在 Macromedia 的网站上就有数百个功能强大的插件。为了方便读者，我们在光盘上也选择了一些比较实用的插件，供大家参考使用。

第 15 章 定制 DW UltraDev 4 界面

本章概要：

- ☒ DW UltraDev 4 界面定制概述
- ☒ 定制对象面板
- ☒ 创建一个简单对象
- ☒ 改变缺省文件类型
- ☒ 定制 DW UltraDev 菜单
- ☒ 定制对话框界面
- ☒ 编辑 HTML 源码格式化配置文件
- ☒ 浏览器配置文件
- ☒ 扩展 DW UltraDev
- ☒ 第三方标记的定制

我们每个人都有自己的操作习惯，如果是在自己熟悉的界面下操作，将令人感到舒适愉快，工作效率也会提高，所以对每一个软件，如果能让用户定制界面，也是其吸引用户的一个卖点之一。

15.1 定制 DW UltraDev 4 界面概述

DW UltraDev 4 允许我们以多种方式定制其界面，这样，每个用户都可以在自己熟悉、舒服和有效的方式下工作。

DW UltraDev 为我们提供了以下一些定制方法：

- ※ 重新安排对象面板中的对象，将那些经常使用的对象放置在面板中。我们也可以创建新的面板以便重组对象，或者在已有的对象面板上增加新对象。具体方法参见 15.2 节。
- ※ 改变菜单项的键盘快捷键或菜单名称、增加新的菜单命令，或者从菜单中移走已有

的菜单命令。具体内容参见 15.5 节。

- ✧ 编辑 HTML 源码格式化配置文件，以获得对 DW UltraDev 创建的 HTML 源程序的更好控制。SourceFormat.txt 文件包含 HTML 代码显示格式的所有设置。详细内容参见 15.7 节。
- ✧ 利用 JavaScript 创建用户自己的命令和浮动面板。详细内容参见 15.9 节。
- ✧ 改变第三方标记（如：ASP 和 ColdFusion 标记）在文档窗口上的显示方式。参见 15.10 节。
- ✧ 设置网站的所有特性，包括从颜色方案到站点的配置和浏览。这些内容已经在前面讲述过了。

15.2 定制对象面板

缺省情况下，对象面板分为 8 个：字符对象面板（Characters）、常规对象面板（Common）、表格对象面板（Forms）、框架对象面板（Frames）、表头对象面板（Head）、一些在以上这些面板中的对象的不可见信息组成的面板（Invisibles）、活动对象面板（Live）和几个特殊对象组成的特殊面板（Special）。与这些面板相对应的文件夹位于 DW UltraDev 应用程序安装后磁盘目录的 Configuration/Objects 文件夹下。

对象面板中的每一个面板上的每一个对象，在相应文件夹中都有两个或三个文件与之对应：

- ✧ 一个 GIF 文件，包含该对象的图标。
- ✧ 一个 HTML 文件，包含的内容要么是将要插入你的文件中的 HTML 源代码，要么是你将用于指定插入数据的 HTML 格式。
- ✧ 一个 JavaScript 文件（可选），用于产生插入你的文件中的 HTML 源码。

用户可以将对象从一个面板移到另一个面板、重命名面板，甚至可以从面板中移走所有的对象。一旦作了这些改变，用户可以重新装载 DW UltraDev 的扩展文件，以使这些变化在对象面板可见。

1. 将对象从一个面板移到另一个面板

如果需要将对象从一个面板移到另一个面板，方法是将对象的 HTML 和 GIF 文件、JavaScript 文件（如果存在的话）从当前文件夹移到 Configuration/Objects 文件下的另一个文件夹。确保对象的所有文件都被移走。

2. 重命名对象面板中一个面板

如果希望重命名对象面板中的一个面板,方法很简单,重命名 Configuration/Objects 文件夹下的相应文件夹。

3. 从对象面板中移走一个对象

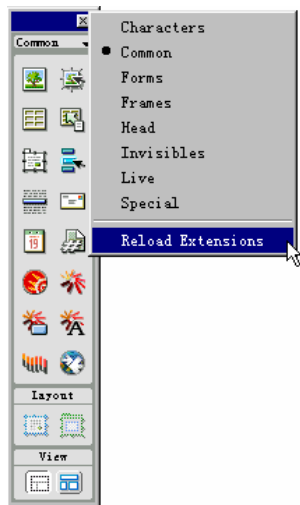
如果希望将对象从面板中移走,方法是将对对象的 HTML、GIF 和 JavaScript 文件从 Configuration/Objects 文件夹下移走。

4. 重载扩展文件

对 Configuration/Objects 文件夹进行改动后,需要重载 DW UltraDev 的扩展文件,才能看见改动后的效果。重载扩展文件的方法:

(1) 按下 Ctrl(Windows 系统)或 Option(Macintosh 系统)键,单击 Object 面板顶部的弹出菜单,将弹出如图 15-1 所示的菜单:

图 15-1
带有扩展
选项的快
捷菜单



(2) 选择 Reload Extensions 菜单项,即可完成扩展重载工作。你所作的修改在面板上能立即表现出来。

下面让我们来做个练习,删除 Common 对象面板上的 Insert Email Link 对象。首先删除 Configuration/objects/Common 目录下的与此对象相关的所有文件:E-Mail Link.gif、E-Mail Link.htm 和 E-Mail Link.js(这里只是一个示例,因为 E-mail Link 对象对我们编程很有用,所以读者应该将这些文件保存到另一个地方,以便恢复时用)然后利用 Reload Extensions 菜单项完成重载,结果如图 15-2 所示,我们发现 Insert E-mail Link 对象已经

在 Common 对象面板上消失了。

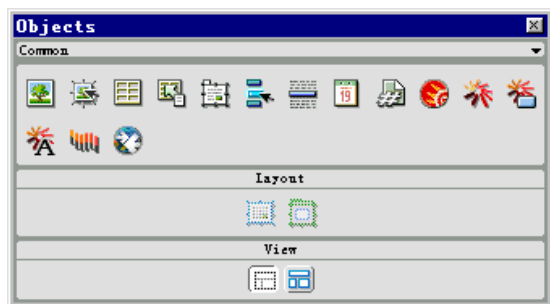


图 15-2
修改后的
对象面板

恢复 Insert E-mail Link 对象很简单，将以上提到的三个文件拷贝回原来的目录，再利用 Reload Extensions 菜单项，即可恢复以前的对象面板。

15.3 创建一个简单对象

许多简单的对象要求不用 JavaScript 编程，而是仅仅包含将插入文档的 HTML 语言。所以如果读者希望利用 JavaScript 语言创建更复杂的对象，请参见 15.9 节。下面让我们来创建一个简单对象：

1 在一个文本编辑器（例如：BBEdit 或 HomeSite）中创建一个新的空白文档。

用户可以将 HTML Source Inspector 作为自己的文本编辑器，但是首先你必须删除在你创建新文档时，在 HTML Source Inspector 中显示的所有标记。

2 在标记中键入或粘贴你需要插入你文档的对象，例如，键入以下代码：

```
<p>
&copy ; 2000 DCD Productions, Inc. <BR>
All Rights Reserved
</p>
```

3 保存文件。

如果你希望新建的对象显示在已有的任一对象面板上，则将其保存在已有的对象文件夹下（Characters、Common、Forms、Frames、Head 或 Invisibles）。如果希望创建一个新面板，则在 Configuration/Objects 目录下创建一新文件夹，并将文件保存至此文件夹下。

注意，在任意面板子目录下的文件夹（例如，在 Characters 子目录下创建的文件夹）都将被忽略。

4 在图像或图形编辑应用（例如，Macromedia Fireworks）中，创建一个大小为 18 × 18

像素的 GIF 图形，它将是新对象在对象面板中图标。

如果你创建了一个大图形，DW UltraDev 会自动将其缩放为 18×18 像素。如果你没有为此新对象创建一个图标，DW UltraDev 将会在对象面板上插入一个通用对象图标。

5 将图标文件设置为与对象文件同名，并保存在同一目录下。

例如，如果你的对象名为 Copyright_DCD.htm，它保存在 Common 目录下，那么你的图标文件就应该为 Copyright_DCD.gif，并且也将其保存在 Common 目录下。

6 重新启动 DW UltraDev 程序以便使用新对象。新建对象将出现在 Insert 菜单的最底部，同时也出现在相应的对象面板上。

15.4 改变缺省文件类型

缺省情况下，DW UltraDev 在 File / Open 对话框中显示它能识别的所有文件类型。用户可以利用此对话框下的弹出菜单，限定其仅显示某种特定类型的文件。如果你的工作经常涉及到一种特定的文件类型（如，ASP 文件），你可以根据此方法改变文件打开窗口的缺省显示。

1. 改变 DW UltraDev 的 File / Open 缺省文件类型

1 备份 Configuration 文件夹下的 Extentions.txt 文件，以便将来我们需要恢复缺省设置时使用。

2 在文件编辑器中打开 Extentions.txt 文件（不要在 DW UltraDev 中打开）。

3 剪切新缺省类型对应的文本行，并将其粘贴至文件的第一行。然后保存文件，重新启动 DW UltraDev，再打开“File / Open”，此时就能看到新的缺省类型。

2. 在 File / Open 对话框中增加新的文件类型

1 同上面一样，首先需要备份 Configuration 文件夹下的 Extentions.txt 文件，以便将来我们需要恢复缺省设置时使用。

2 在编辑器中打开 Extentions.txt 文件，同样切记不能在 DW UltraDev 中打开。

3 为每种需要增加的文件类型，在此文件中增加一新行。用大写字母表示新文件类型的扩展名，如果一行显示多个文件类型，它们之间用逗号分隔；然后紧跟着一个冒号，后面是對此文件类型简短的描述。例如，对 JPEG 文件，我们可以键入以下文本：

JPG,JPEG,JFIF:JPEG Image Files

4 保存文件，重新启动 DW UltraDev，我们就可以测试一下新的变化。

15.5 定制 DW UltraDev 菜单

DW UltraDev 利用一个 XML 文件 `menus.xml` 中定义的结构创建其所有的菜单，此文件位于 DW UltraDev 安装目录下的 `Configuration/Menus` 子目录下。编辑 `menus.xml` 文件中的内容，当我们重启 DW UltraDev 之后，就可以发现对菜单所作的变化。

通过编辑 `menus.xml` 文件，我们可以增加、改变和移走菜单项的键盘快捷键；我们也可以重组、重命名和移走菜单项。



在修改之前，一定要记住备份最初的 `menus.xml` 文件，或任意其他的 DW UltraDev 配置文件。在编辑配置文件时，难免会出错，但在 DW UltraDev 应用程序中却无法将其转换为先前的配置。幸运的是，在 `Configuration` 文件夹下包含了缺省 `menus.xml` 文件的备份，名字为：`menus.bak`，只需将其代替 `menus.xml` 文件，即可恢复缺省菜单设置。

15.5.1 编辑 Commands 菜单

我们可以不编辑 `menus.xml` 文件，就可以为 `Commands` 菜单增加特定的命令，或者改变其菜单命令名称。

如果希望创建新的命令并让它们能自动放置于 `Commands` 菜单下，那就使用 `History` 面板。

如果需要改变已创建的命令名称，或将其从 `Commands` 菜单中删除，我们就可以选择 `Commands / Edit Command List` 菜单命令（注意，当利用此方法删除一菜单命令时，同时也删除了包含此命令的文件）。

如果需要对 `Commands` 菜单下的命令重新排序，或者移走某菜单项，我们就必须编辑 `menus.xml` 文件。



术语 `command` 在 DW UltraDev 中有两种含义。严格地说，从一个扩展开发者的角度，一个命令（`command`）就是一种特殊的扩展。但在某些上下文中，“`command`”可以当作 DW UltraDev 菜单中任意菜单项的别称。

15.5.2 菜单语法

在改变 menus.xml 文件时一定要小心谨慎,因为 DW UltraDev 将忽略任何含有 XML 语法错误的菜单或菜单项。

menus.xml 文件包含菜单条、菜单以及菜单项的结构列表。一个菜单条(由一个表示开始和结束 MENUBAR 标记符标记)是一个菜单的集合,例如主菜单条、分离的 Site 窗口菜单条(仅对 Windows 存在),以及每个上下文菜单对应的菜单条。每个菜单条包含一个或多个菜单项,由 MENUITEM 标记以及它的属性描述。另外,一个菜单可以包含分隔符和子菜单。例如,下面这段代码是主菜单条(文档窗口)中的一部分定义:

```
<menubar name="Main Window" id="DWMainWindow">
  <menu name="_File" id="DWMenu_File">
    <menuitem name="_New" key="Cmd+N" enabled="true" command="dw.createDocument()"
      id="DWMenu_File_New" />
    ...其他菜单项、分隔符和子菜单
  </menu>
  ...其他菜单
</menubar>
```

15.5.3 菜单重排

我们可以移走菜单中的一个菜单项,或者将其从一个菜单移至另一个菜单、改变菜单条中菜单项的顺序,以及在菜单中增加或移走分隔符。

注意,上述功能对上下文菜单同样适用。

1. 移走菜单项

- 1 退出 DW UltraDev。
- 2 备份 menus.xml 文件。
- 3 在文本编辑器(如, BBEdit、HomeSite 或 Wordpad)中打开 menus.xml 文件,切记不要在 DW UltraDev 中打开。
- 4 剪切从<menuitem>标记开始到<menuitem/>标记结束的整个 MENUITEM 标记内容。
- 5 在新位置放置插入点。

 在新位置粘贴已剪切的菜单项。一定要确保是在<MENU>标记和相应的</MENU>标记之间粘贴。

如果需要创建子菜单，只需在一个菜单内再嵌套一对开始和结束 MENU 标记即可。也就是说，在一个菜单中插入一对新的<MENU>和</MENU>标记，然后在这对标记之间加入新的菜单项。

2. 在两个菜单项之间插入一个分隔符

这种功能实现起来比较简单，只是在两个 MENUITEM 标记之间输入一个<SEPARATOR />标记即可。

如果希望移走此分隔符，将相应的<SEPARATOR />行删除就行了。

15.5.4 菜单项语法

本节让我们来看看 MENUITEM 标记的各属性的含义：

1. NAME

这是在菜单上显示的菜单项名称。其中的下划线表示随后的字母是此菜单命令的记忆符（仅对 Windows 系统适用）。此属性是必须设置的。

2. ID

DW UltraDev 利用 ID 属性标识一个菜单项。每一个 ID 在整个菜单结构必须是惟一的。如果你希望在 menus.xml 中再增加新菜单项，建议你用你们公司的名字或其他可以惟一标识的字符串作为新菜单项 ID 的前缀，以确保 ID 惟一。此属性也是必须设置的。

3. KEY

KEY 是菜单命令的键盘快捷键。如果存在的话，使用下列字符串指定修饰键：

- ✱ Cmd 指定的是 Control 键（Windows）或 Command 键（Macintosh）。
- ✱ Alt 或 Opt 指定的是 Alt（Windows 系统）键或 Option（Mac 系统）键。
- ✱ Shift 指定 Shift 键，这对 Windows 和 Mac 系统都适用。
- ✱ Ctrl 指定 Control 键，与 Shift 字符一样，对两种系统都一样。
- ✱ 如果给定的快捷键由多个修饰键组成，则它们之间用加号“+”分隔。例如，KEY 属性为：Cmd + Opt + 5，这在 Windows 系统中表示当同时按下 Control+Alt+5 键，即可执行该菜单命令；在 Mac 系统中按下的是 Command+Option+5。
- ✱ 特殊的键是由名称指定的：从 F1 到 F12、PgDn、PgUp、Home、End、Ins、Del、

Tab、Esc、BkSp 和 Space。也可以利用这些特殊的键作修饰符。

4. PLATFORM

此属性用于指定该菜单项用于某种平台上。有效值为 win, 意味着仅适用于 Windows 系统; 如果有效值为 mac, 则意味着是用于 Macintosh 系统的。缺省情况是指该菜单对两个系统都适用。

5. ENABLED

这是 JavaScript 函数名, 用于指定该菜单项当前是否是处于激活状态。如果此函数返回 false, 表示此菜单项当前为变灰状态。

6. COMMAND

这是 JavaScript 的表达式, 当用户从菜单中选择此菜单项时执行此表达式。但是, 对复杂的 JavaScript 代码, 使用的是 JavaScript 文件(由 FILE 属性指定)。对一个菜单项, COMMAND 和 FILE 属性必须指定其中之一。

7. FILE

这是包含 JavaScript 代码的 HTML 文件的名称, 此文件用于控制菜单项。在此属性中指定的文件路径是相对 Configuration 文件夹的。

要注意的是, FILE 属性将覆盖 COMMAND、ENABLED 以及 CHECKED 属性。

8. CHECKED

这也是 JavaScript 的表达式, 指示是否在该菜单项的旁边加一个检查标志。如果表达式的值为 true, 则菜单项显示时旁边会出现一个检查标志。

9. DYNAMIC

如果此属性存在, 则表明该菜单项是动态显示的, 也就是说, 由包含有 JavaScript 代码的 HTML 文件设置该菜单项的文本和状态。用户要切记: 如果你指定了菜单项的 DYNAMIC 属性, 那就必须指定 FILE 属性。

15.5.5 键盘快捷键

如果觉得缺省的键盘快捷键使用起来不太方便, 你可以改变或移去已有的键盘快捷键, 也可以增加一个新的。

 退出 DW UltraDev 程序。

- 2 备份 menus.xml 文件。
- 3 在任意的文本编辑器中打开 menus.xml 文件，但不要在 DW UltraDev 中打开。
- 4 查看 Keyboard ShortCut Matrix，以找到目前还未使用的，或者是你希望重新分配的快捷键。如果你重新分配快捷键了，一定要在自己已打印出来的键盘快捷键矩阵页上将此快捷键用笔勾掉，以便将来参考。
- 5 如果你正在重新分配一个快捷键，并发现它已经分配给一个菜单项了，那么就将此菜单项的 KEY="shortcut" 属性删除。
- 6 找到希望分配此快捷键的菜单项。
- 7 如果此菜单项还没有快捷键，则在 MENUITEM 标记内的任意地方增加：KEY=""。
- 8 在 KEY 双引号之间加入新的键盘快捷键。
如果此键盘快捷键已经在别处使用过了，那么此快捷键仅对在 menus.xml 文件中位于前面的那个菜单项有效。
- 9 在键盘快捷键矩阵图的合适位置加上此新的快捷键。

15.5.6 改变菜单项名字

我们可以很容易地改变任意菜单项的名称。

- 1 退出 DW UltraDev 程序。
- 2 备份 menus.xml 文件。
- 3 在任意的文本编辑器中打开 menus.xml 文件，除了 DW UltraDev。
- 4 找到相应的 MENUITEM 标记，改变 NAME 属性的值。不要改变 ID 的值。

15.6 定制对话框界面

对话框中的各种对象、命令以及行为的布局都是由 HTML 文件中的 HTML 格式指定的，此文件位于 DW UltraDev 安装目录下的 Configuration 子目录下。我们可以利用 DW UltraDev 编辑这些格式。

- 1 在 Configuration/Objects 或 Configuration/Commands 或 Configuration/Behaviors 目录下找到合适的.htm 文件。
- 2 备份此文件，将其放置在 Configuration 文件夹之外的任意目录下。
- 3 在 DW UltraDev 中打开此备份文件，编辑格式，并保存。
- 4 退出 DW UltraDev 程序。
- 5 将已改变的备份文件拷贝回 Configuration 文件夹中，并代替已有的。最好的方法是备份原始的那个.htm 文件，以便将来在需要时恢复原状。

 启动 DW UltraDev，我们就可以发现变化了。

我们应该仅仅改变对话框的外表，而不是它的内部工作方式。它仍然应该包含有用同一名字命名的表格元素的相同类型，这样，DW UltraDev 从对话框获取的信息仍以同样的方式被使用。

例如，在一个对话框中的 Comment 对象取得文本域输入的文本，然后利用一个简单的 JavaScript 函数将此文本转换为一个插入到你的文档中 HTML 注释。描述此对话框的格式位于 Configuration/Objects/Invisibles/Comment.htm 文件中。你可以打开此文件并改变文本域的大小或其他属性，但是如果你将整个 TEXTAREA 标记移走，或改变它的 NAME 属性的话，Comment 对象将不能正确的工作了。

15.7 编辑 HTML 源码格式化配置文件

HTML 源格式化配置文件决定 DW UltraDev 如何为一个文档格式化 HTML 源代码。此配置文件包括单个标记以及标记组的格式属性，同时还有 HTML 格式化属性（由 Edit / Preferences 菜单命令设置）。我们可以在文本编辑器中编辑 SourceFormat.txt 文件，以精确控制 DW UltraDev 写 HTML 代码的方式。此配置文件是保存在 DW UltraDev 安装目录下的 Configuration 文件夹中的一个文本文件。

当 DW UltraDev 退出时，由 Preference 命令设置的 HTML 格式化属性将保存至 SourceFormat.txt 文件中，写入此文件的属性到下一次启动 DW UltraDev 时就会生效。因此，为了确保能访问新的属性设置，应在编辑此配置文件之前，退出 DW UltraDev 应用。

HTML 源格式化配置文件遵循一定的格式。注意下列协定：

- ※ 配置文件的每一小节都是以 <?keyword> 标记开始的。例如：<?OPTIONS>、<?ELEMENTS>和<?ATTRIBUTES>。
- ※ 每一小节的参数都是在此小节的最顶部的 HTML 注释：<!--和 -->之间定义的。
- ※ 在<?OPTIONS>小节中的 OMIT 行是为将来使用保留的，它并不影响目前的 HTML 源格式化配置。
- ※ 一个标记可以隶属于<?ELEMENTS>小节中的缩排组(IGROUP)。缺省时，IGROUP 1 包含表格行和列，IGROUP 2 包含框架集和框架。这些组与 HTML Format Preferences 对话框中缩排的 Table Rows 和 Columns,以及缩排的 Framesets 和 Frame 选项相对应。

例如，在 SourceFormat.txt 文件中的 P 标记的缺省设置为：<P BREAK="1,0,0,1"

```
<p> A paragraph of text that is indented from the left margin and that has a break
    before the opening P tag and after the closing P tag, but not after the opening P and
    not before the closing P.</p>

<p>Next paragraph.</p>
```


如果将其修改为：`<P BREAK="1,1,1,2">`，结果如下：

配置文件中还有一些标记和属性的设置，包括术语 `NAMECASE` 或 `SAMECASE`。其中，`NAMECASE` 是用于指定此标记或属性将按照一定的方式大写化。例如：`onClick` 的大小写形式是由 `<onClick namecase="onClick">` 语句指定的。另外，DW UltraDev 利用 `SAMECASE` 以确保能使用正确的大小写产生特定的属性。切记，不要从任何属性格式

```
<p>
A paragraph of text that is not indented from the left margin and that has one break
before and after the opening P tag, one break before the closing P tag, and two breaks
after the closing P tag.

</p>

<p>
Next paragraph.

</p>
```

化规范中移走 `SAMECASE`!

在 `SourceFormat.txt` 文件没有指定的任何属性,将使用你在 `HTML Format Preferences` 面板中指定的缺省格式化设置值。

15.8 浏览器配置文件

浏览器配置文件 (profile) 是当你运行目标浏览器检查时，你的文档将与之比较的文件。每项配置包括一特定浏览器支持的 `HTML` 标记和属性的信息。它也可以包含对标记替换的警告、错误消息和建议。

浏览器配置文件保存在 DW UltraDev 安装目录下的 `Configuration/BrowserProfiles` 文件夹下。你可以对其进行编辑，或重新创建一个新的配置文件。在进行这些工作时，并不需要将 DW UltraDev 程序关闭。

1. 浏览器配置文件格式

浏览器配置文件遵循一种特定的格式。为了避免在目标浏览器检查期间出现解析错误，我们在编辑或创建新配置文件时，应该遵守这些规则：

- ※ 配置文件的第一行是为文件名保留的。它必须跟着一个回车返回。这个名称将在 `Target Browser Check` 对话框和目标检查报告中显示。它必须是惟一的。
- ※ 第二行是为指示器 `PROFILE_TYPE=BROWSER_PROFILE` 保留的。DW UltraDev

- 利用此行决定哪个文档是浏览器配置文件。此行也不能修改或移动。
- ✧ 行首如果是两个连字符“ -- ”，则表示这是注释行，在进行目标检查进程中，此行是被忽略不计的。
 - ✧ 空格将出现在下列一些地方：在!ELEMENT 行中“ > ”括号之前，在左圆括号之后、右圆括号之前，并且在值列表的每个“ | ”符号之前。
 - ✧ 在下列这些词之前必须有一个没有空格的感叹号：ELEMENT、ATTLIST、Error、msg 和 htmlmsg，即，!ELEMENT、!ATTLIST、!Error、!msg 和!htmlmsg。
 - ✧ !Error 和!Warning 可以出现在!ELEMENT!和 ATTLIST 域。
 - ✧ !msg 消息可以仅包含简单的文本。!htmlmsg 消息可以包含任意有效的 HTML，包括超链接。
 - ✧ HTML 注释(<!-- -->) 不能作为标记在浏览器配置文件中列出，这是因为它将会干扰解析进程。DW UltraDev 不报告注释的错误，因为所有的浏览器都支持此项功能。一个标记项的语法为：

```
<!ELEMENT htmlTag NAME="tagName" >
<!ATTLIST htmlTag
  unsupportedAttribute1 !Error !msg="The unsupportedAttribute1 of the htmlTag is not supported. Try
using supportedAttribute1 for a similar effect."
  supportedAttribute1
  supportedAttribute2 ( validValue1 | validValue2 | validValue3)
  unsupportedAttribute2 !Error !htmlmsg="<b>Don't ever use this unsupportedAttribute2 of the
htmlTag !!</b>"
>
```

其中，htmlTag 是在 HTML 文档中显示时的标记。

tagName 是标记的称谓，例如，HR 标记称为“ horizontal rule ”。NAME 属性是可选的。如果指定，tagName 将用在错误消息中。

unsupportedAttribute 是指不支持的属性。对没有专门指出来的任意标记或属性都认为是支持的。仅当需要创建定制的错误消息时，才指定不支持的标记或属性。

SupportedAttribute 是指一个由 htmlTag 支持的属性。其中，只有那些没有!Error 指示符的标记才被浏览器支持。

validValue 是指属性支持的一个数值。

下面的例子将展示一个 APPLET 标记项，将适用于 Navigator 3.0：

<!ELEMENT APPLET Name="Java Applet" >
<!ATTLIST APPLET
Align(top middle bottom left right absmiddle absbottom baseline texttop)
Alt
Archive
Class !Warning !htmlmsg="This browser ignores the <CODE>CLASS</CODE> attribute for the
<CODE>APPLET</CODE> tag."
Code
Codebase
Height
HSpace
ID !Warning !htmlmsg="This browser ignores the <CODE>ID</CODE> attribute for the
<CODE>APPLET</CODE> tag. Use <CODE>NAME</CODE> instead."
Name
Style !Warning !htmlmsg="This browser ignores the <CODE>STYLE</CODE> attribute for the
<CODE>APPLET</CODE> tag."
VSpace
Width
>

2. 创建一个浏览器配置文件

通过修改已存在的配置文件创建一个新的浏览器配置文件。例如，为 Netscape Navigator 5.0 创建一个配置文件，首先，我们可以打开 Navigator 4.0 的配置文件，增加 5.0 版本中介绍的所有新标记或属性，然后将此文件当 Netscape Navigator 5.0 配置文件保存。

- 1 利用文本编辑器，打开一个与你打算创建的配置文件非常相像的配置文件，或打开你想修改的配置文件。
- 2 改变配置文件的名称。
配置文件的名称在文件的第一行。两个配置文件不能拥有同一个名称。
- 3 根据上面已经讲述的语法，将你所知道的浏览器支持的所有新标记或属性加到文件中。

如果你不希望接收到某一个不支持的标记的错误消息，将其增加到支持标记列表中。此时，应该将文件保存为另一个名称，这样可以保护原始文件。

4 删除那些不被浏览器支持的标记或属性。

如果你是为 Netscape 或 Microsoft Explorer 的下一版本创建配置文件,那么这一步就不是必要的,因为这两个浏览几乎支持所有的标记。

5 根据上述语法,增加定制的错误消息。

DW UltraDev 的配置文件列出所有支持的标记。它们也可以包含经常使用的注释行,但没有不支持的标记。为了增加定制错误消息,将行开头的两个连字符移走,并在!Error 之后加上!msg“ message ”或!htmlmsg“ <tag>message</tag> ”。例如,在 Netscape Navigator 3.0 的配置文件中会出现:

```
<!ELEMENT HR name="Horizontal Rule" >
<!ATTLIST HR
-- COLOR !Error
>
```

为了增加一个定制错误消息,将两个连字符移走,并加入自己的消息,结果如下:

```
<!ELEMENT HR name="Horizontal Rule" >
<!ATTLIST HR
COLOR !Error !msg "Internet Explorer 3.0 supports COLOR in horizontal rules, but Netscape
Navigator 3.0 does not."
```

6 可以对所有的错误情况都使用!Error, 或者使用!Warning 指出一个标记将被忽视,但它并不会引起一个错误。

15.9 扩展 DW UltraDev

1. 基础

DW UltraDev 的初衷是可扩展的。它包含一个 JavaScript 解释器,所以它可以读和执行 JavaScript 代码,并且它提供一个应用程序接口(API),这是一个 JavaScript 函数的大集合,允许开发者去扩展 DW UltraDev 的功能。DW UltraDev 也提供了一个文档对象模型(DOM),这将允许以后的扩展可以检查和修改一个文档结构和内容。

利用 JavaScript API,我们可以创建对象、行为动作、命令、属性检查器、浮动面板和数据转换器,以上这些统称为扩展。

我们也可以在编程一无所知的情况下,创建新对象和新的简单命令。但是对更深入一层的应用,如增加 DW UltraDev 的功能,就必须写它的扩展,这可以利用 JavaScript 或 C 语言。

2. 编辑 DW UltraDev 命令

在 DW UltraDev 菜单中的的所有命令，包括那些利用 Histroy 面板创建并保存的命令，都是用 JavaScript 语言编写的。这些 JavaScript 代码大部分都是通过调用 DW UltraDev 扩展 API 构成的。如果你熟悉 JavaScript 并懂得 DW UltraDev 扩展 API，就可以编辑 JavaScript 代码以改变 DW UltraDev 命令的功能。



不要试图改变任何 JavaScript 代码，除非你确保你熟悉你所要做的事情!

15.10 第三方标记的定制

服务器方的处理技术，如 ASP、PHP、ColdFusion 和 JSP，在 HTML 文件中使用特殊的非 HTML 代码指定。服务器基于这些代码创建并提供 HTML 内容。当 DW UltraDev 遇到非 HTML 标记时，它将此标记与第三方标记文件中的信息比较，该文件定义了 DW UltraDev 读取和显示非 HTML 标记的方法。

例如，ASP 文件包含服务器将解释的代码。这些代码是由一对分隔符表示的，由 “<%” 字符串开始，“%>” 字符串结束。Configuration/ThirdPartyTags 文件夹包含一个名称为 “ASP.xml” 的文件，它描述了 ASP 代码的格式，并且定义了 DW UltraDev 显示代码的方式。因为 ASP 代码的显示方式都是在 ASP.xml 文件中定义的，所以 DW UltraDev 并不去解释在两个分隔符之间的所有内容。

15.10.1 第三方标记的定制

用户可以定义自己的标记，并且创建标记数据库文件，以便 DW UltraDev 可以读取并显示这些自定义标记。如果服务器方的标识系统不是 ASP、ColdFusion 或 PHP，那么创建标记数据库就是非常有用的，为每一个标识系统创建一个新的标记数据库，让 DW UltraDev 知道如何去显示这些标记。

要注意的是，本节是解释在 DW UltraDev 中如何定义一个定制标识的外表，而不是如何去编辑一个定制标识的内容或特性。

每个标记数据库文件为一个或多个定制标记定义了名称、类型、内容模式、表现方案和图标。用户可以创建任意数目的标记数据库文件，但有一点必须记住，这些文件都必须保存在 Configuration/ThirdPartyTags 文件夹中，以便 DW UltraDev 能够读取和处理。标记数据库文件的扩展名为.xml。

如果你一次就需要工作在几个不同的并且不相关的站点上，那么就将一个特定站点的所有标记规范都放在一个文件中。然后利用定制图标简单地包含此标记数据库文件，并转交给维护此站点的人。

例如，我们利用名为 TAGSPEC 的一个 XML 标记定义了一个标记规范，下面的代码描述了 HAPPY 标记的规范：

```
<TAGSPEC TAG_NAME="happy" TAG_TYPE="nonempty" RENDER_CONTENTS="false"
CONTENT_MODEL="marker_model" ICON="happy.gif" ICON_WIDTH="18"
ICON_HEIGHT="18"></TAGSPEC>
```

存在两种不同的利用 TAGSPEC 定义的标记：标准 HTML 风格标记和字符串定界符的标记。字符串定界符的标记就像空的 HTML 标记（比如：IMG），它们并没有内容，并且没有一个结束标记。上面显示的 HAPPY 标记是一个标准 HTML 风格标记，它以开始<HAPPY>标识开始，然后是内容，接着是一个结束</HAPPY>标记。对一个 ASP 标记，以字符串<%开始，并以字符串%>结束，这就是一个字符串定界符标记。

表 15-1 TAGSPEC 的属性和可能的取值

属 性	值	描 述
TAG_NAME	tagName	定制图标的名称。必须设置。对字符串分隔符标记，TAG_NAME 仅用于决定此标记能否使用给定的一个 Property 检查器。如果此 Property 检查器的第一行包含此标记名称，并且此名称前后都有一个星号，则表示此检查器可以用于这种类型的标记。例如，用于 ASP 代码的 TAG_NAME 为：ASP，需要检查 ASP 代码的 Property 检查器应该在第一行包含：*ASP*
TAG_TYPE [*]	空或非空	决定标识自身是否就是内容（就像 IMG 一样），或者是否包围内容（如 CODE）。这对 HTML 标准风格类型标记是必须设置的
RENDER_CONTENTS [*]	TRUE 或 FALSE	决定标记的内容是否显示在文档窗口，或者由图标代替显示。该属性对非空标记是必须设置的

（续表）

属 性	值	描 述
-----	---	-----

CONTENT_MODEL	block_model head_model marker_model script_model	描述标记可以包含的内容和可以包含的标记。该属性必须设置。 block_model 指定标记可以包含块级元素，如：DIV 和 P，并且指定该标记仅能在 BODY 节显示，如：DIV、LAYER 或 TD marker_model 指定标记可以包含任意有效的 HTML 代码，并且它可以在任意地方显示 script_model 允许标记在一个文档的开始和结束 HTML 标记之间存在。当 DW UltraDev 遇到这种模式的标记，它将忽略标记的所有内容。该模式用于 DW UltraDev 不需要解释的第三方标识（像某些 ColdFusion 标识）
START_STRING	beginningOfTag	指定标志一个字符串分隔符标记的开始分隔符。字符串分隔符标记可以在能够显示注释的文档中的任意地方显示。DW UltraDev 不进行 START_STRING 和 END_STRING 之间的任何标记或实体的解析，或 URL 的解码。如果 END_STRING 被指定时此属性是必须设置
END_STRING	endOfTag	指定标志一个字符串分隔符标记的结束分隔符。如果属性 START_STRING 已经指定了，则此属性是必须设置的
DETECT_IN_ATTRIBUTE	TRUE 或 FALSE	指定是否忽略在 START_STRING 和 END_STRING 之间的所有内容（或者当这些字符串没有定义时，指开始和结束标记间的内容）。一般来说，你应该为字符串分隔符标记设置此属性为 TRUE。缺省为 FALSE
PARSE_ATTRIBUTES*	TRUE 或 FALSE	指示是否解析该标记的属性。如果此属性为 TRUE（缺省值），DW UltraDev 将解析标记的属性。如果为 FALSE，DW UltraDev 忽略直到下一个结束' >' 符号之前的所有内容。例如，像 CFIF 标记，此属性应该设置为 FALSE（<cfif a is 1>）
ICON	URL	指定与标记相关的图标的路径和文件名。对空标记和其内容不显示文档窗口内的空标记来说，这是必须设置的
ICON_WIDTH	widthInPixels	指定图标的宽度
ICON_HEIGHT	heightInPixels	指定图标的高度

*使用 START_STRING 和 END_STRING 的标记将忽略这些属性。

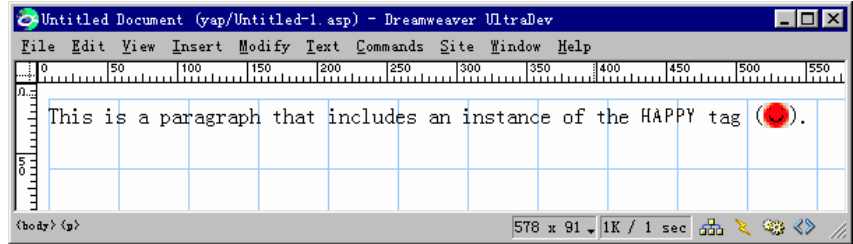
15.10.2 文档窗口显示定制标记的方法

文档窗口显示定制标记的方法依赖 TAGSPEC 标记的 TAG_TYPE 和 RENDER_CONTENTS 属性的值。如果 TAG_TYPE 的值为 empty，则显示在 ICON 属性中指定的图标。如果 TAG_TYPE 的值为 nonempty，但是 RENDER_CONTENTS 是 FALSE，当它是一个空标记时，显示图标。例如，上面定义的 HAPPY 标记可能在 HTML 中的代码显示如下：

```
<P>This is a paragraph that includes an instance of the <CODE>HAPPY</CODE> tag  
( <HAPPY>Joe</HAPPY> ) .</P>
```

并且在文档窗口中的显示如图 15-3 所示：

图 15-3
第三方标
记在文档
窗口的显
示



对一个 RENDER_CONTENTS 的值为 TRUE 的非空标记,在文档窗口将显示其在开始和结束标记之间的内容,而不是图标,例如：

```
<MYTAG>this is the content between the opening and closing tags</MYTAG>
```

那么文档窗口中将显示 “ this is the content between the opening and closing tags ”。

并且,如果 View / Invisible Elements 菜单项是激活的,那么此内容将以 Highlighting 特性中指定的第三方标记颜色高亮显示。



高亮显示仅适用于在标记数据库文件中定义的标记。

如果你觉得当前第三方标记的高亮显示颜色看起来不是很舒服,你也可以改变它,方法如下：

- (1) 选择“ Edit / Preferences ”菜单项。
- (2) 从 Category 列表中选择 Highlighting 项,如图 15-4 所示。
- (3) 单击 Third_Party Tags 颜色框,将弹出颜色调色板。
- (4) 从中选择一个你喜欢的颜色,单击 OK 退出 Preferences 对话框。

15.10.3 避免改写第三方标记

DW UltraDev 纠正 HTML 文档中某些类型的错误。缺省情况下,DW UltraDev 禁止修改 HTML 文档中具有某些特定扩展名的文件,包括.asp(ASP)、.cfm(ColdFusion) 和.php(PHP)。这是为了防止 DW UltraDev 意外修改包含在这些第三方标记中的代码。我们可以改变 DW UltraDev 缺省的改写行为,允许它能修改这些文件中的 HTML 代码,另外,我们也可以增加一些其他的文件类型,禁止 DW UltraDev 改写其内容。

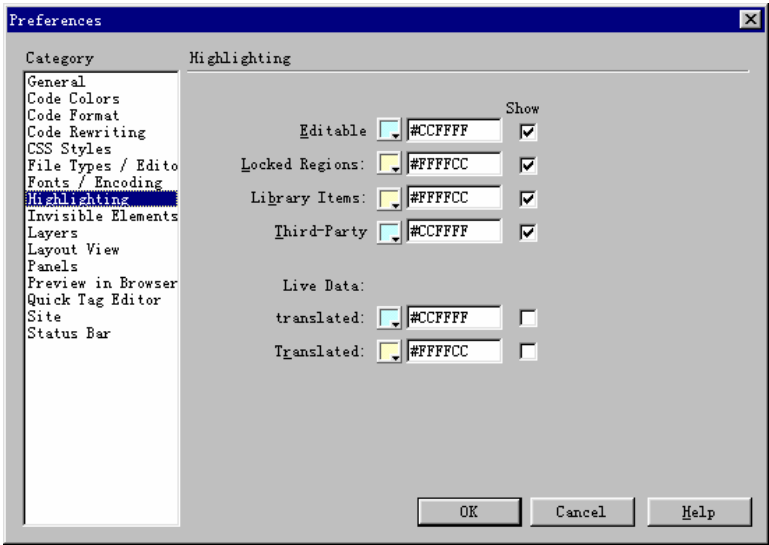


图 15-4
Preferences
对话框

当在 Property 检查器中输入某些特定字符时，DW UltraDev 也可以对之编码。允许 DW UltraDev 具有这种功能是一个非常好的观念，它使得这些特定字符在交叉平台和浏览器上显示的正确性更高。

1. 允许 DW UltraDev 改写所有文件中的 HTML 代码

 选择“ Edit / Preferences ”，并选择 Code Rewriting 类别，如图 15-5 所示：

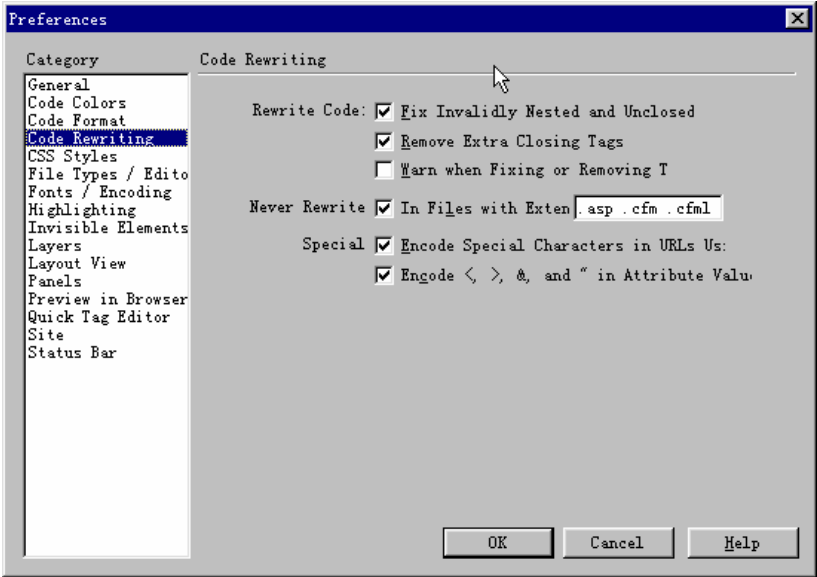


图 15-5
修改 DW
UltraDev
对 HTML
代码的改
写设置

 下面两种方法任选一种：

- ✱ 从扩展名列表中删除一个或多个扩展名。
- ✱ 取消 Never Rewrite HTML In Files with Extensions 选项的选定状态。

2. 增加 DW UltraDev 不能改写的文件类型

 选择“ Edit / Preferences ”，并选择 Code Rewriting 类别，如图 15-5 所示。

 确保选中 Never Rewrite HTML In Files with Extensions 选项，并在此选项的文本域中增加新的文件扩展名。这样，DW UltraDev 对那些具有新增文件扩展名的文件就不能进行改写了。

3. 关闭 DW UltraDev 编码选项

 选择“ Edit / Preferences ”，并选择 Code Rewriting 类别，如图 15-5 所示。

 根据需要取消图 15-5 中所示的最后两个选项中的任意一个或全部。

对于其他的 Code Rewriting 特性，读者可参见前面有关的内容。

第 16 章 DW UltraDev 4 的 DOM 模型

本章概要：

- ☒ DOM 概念
- ☒ DW UltraDev 4 的 DOM
- ☒ 树控制
- ☒ JavaScript 在扩展中的工作方式

DW UltraDev 4 允许对其功能进行扩展，从本章起我们介绍一些扩展 DW UltraDev 4 的方法。DW UltraDev 4 允许我们利用 HTML、JavaScript 和 C 语言对其功能进行扩展。我们可以编写能在 JavaScript 代码中调用的 Objects、Behavior Actions、Commands、Property Inspectors、Floating Palette、Data Translators，以及 C 语言编写的库。为了实现这些目标，要求我们必须精通 JavaScript 语言以及彻底掌握 HTML 原理。对 C 语言并没有很高要求，除非你需要实现一些功能性的东西，而这些功能又不能通过 DW UltraDev JavaScript API 或辅助 API 实现。

本章描述的 DW UltraDev 4 支持的 Document Object Model (DOM)，以及 DW UltraDev JavaScript API (定制 JavaScript 函数)，DW UltraDev 在安装时已经包含了，它们可以用于任何 Object、Behavior Actions、Commands、Property Inspectors、Floating Palette、Data Translator File。本章也描述了用于每种选定类型文件的 API，并提供了一些带有注释的代码例子供读者参考。

在此声明一点，本章从始到终，把 Object、Behavior Actions、Commands、Property Inspectors、Floating Palette、Data Translator Files 都统称为扩展 (extensions)。

DW UltraDev 扩展必须用 JavaScript 语言编写。本书要求读者对 JavaScript 语法和基础编程概念，如，函数、变量、语句、操作数、条件和循环，都比较了解和熟悉，所以这里不再介绍一般编程概念以及 JavaScript 编程方法了。

16.1 DOM 概念

HTML 文档是由显示文档结构的树状标记组成。树的根是 HTML 标记。树的两个最大分支是 HEAD 和 BODY。HEAD 的分支包括 TITLE、STYLE、SCRIPT、ISINDEX、BASE、META 和 LINK。BODY 的分支包括头(H1、H2 等)、块级元素(P、DIV、FORM 等)、文本级元素(FONT、BR、IMG 等) 和 ADDRESS 元素。这些分支的叶包括 :WIDTH、HEIGHT、ALT 和 HREF 等属性。

一个文档对象模块，即 DOM，也是一个揭示文档结构的树。然而，DOM 根据对象和特性报告此结构，而不是根据标记和属性。

DOM 树的根是文档自身，HTML 对象是树干，文档中剩余的对象作为 HTML 的分支，就像 HTML 标记和属性那样。

16.2 DW UltraDev 4 的 DOM

浏览器的 DOM 模型决定了在其平台上运转的 HTML 文档的 JavaScript 的工作方式。同样，DW UltraDev 的 DOM 模型也决定了扩展中的 JavaScript 是如何在 DW UltraDev 上工作的。

DW UltraDev 的 DOM 是 Netscape Navigator 4.0 DOM 的子集和 World Wide Web Consortium(W3C) 的 DOM Level 1 的子集的结合。因为与 DOM Level 1 的结合，HTML 页的每一部分已经变成一个对象——包括标记(在 W3C 中称之为元素)、注释和文本。

对象可以根据下标(document.forms[3].elements[1]) 或名称(document.myForm.myButton) 引用。具有相同名称的对象放在一个数组中。我们可以通过下标访问一组中的特定对象，例如，在 myForm 中名称为 myRadioGroup 的第一个圆按钮可以以此方式引用：document.myForm.myRadioGroup[0]。

表 16-1 给出了所有对象都支持的特性、方法和事件。带“*” 的表示只读特性。

表 16-1 对象支持的特性、方法和事件

对 象	特 性	方 法	事 件
window	document * navigator * innerWidth * innerHeight * screenX * screenY *	alert() confirm() escape() unescape() close() setTimeout()	OnResize

(续表)

对 象	特 性	方 法	事 件
window		clearTimeout() setInterval() clearInterval() resizeTo()	
navigator	platform *	None	None
document	forms *(表格对象的数组) images *(图像对象的数组) layers *(LAYER、ILAYER 和绝对定位的 DIV 和 SPAN 对象的数组) child objects by name * nodeType * parentNode * childNodes * documentElement * body * URL * parentWindow *	getElementsByTagName() hasChildNodes()	onLoad
所有标记/元素对象	nodeType * parentNode * childNodes * tagName * attributes by name innerHTML outerHTML	getAttribute() setAttribute() removeAttribute() getElementsByTagName() hasChildNodes()	
form	对所有标记都可用的附加特性: elements *(button 数组、checkbox、password、radio、reset、select、submit、text、file、hidden、image 和 textarea 对象) child objects by name *	仅这些方法对所有的标记适用	None
layer	对所有标记可用的附加特性: visibility left top width height zIndex	仅这些方法对所有的标记适用	None
image	对所有标记可用的附加特性: src	仅这些方法对所有的标记适用	onMouseOver onMouseOut onMouseDown onMouseUp
button reset submit	对所有标记可用的附加特性: form *	对所有标记都适用的附加方法: blur() focus()	onClick

(续表)

对 象	特 性	方 法	事 件
Checkbox radio	对所有标记可用的附加特性： checked form *	blur() focus()	onClick
password text file hidden image (field) textarea	对所有标记可用的附加特性： form * value	对所有标记都适用的附加方法： blur() focus() select()	onBlur onFocus
select	对所有标记可用的附加特性： form * options * (可选对象数组) selectedIndex	对所有标记都适用的附加方法： blur() (仅适用 Windows) focus() (仅适用 Windows)	onBlur (仅适用 Windows) onChange onFocus (仅适用 Windows)
option	对所有标记可用的附加特性： text	仅这些方法对所有的标记适用	None
array boolean date function math number object string regexp	与 Netscape 4 匹配	与 Netscape 4 匹配	None
text	nodeType * parentNode * childNodes * data	hasChildNodes()	None
comment	nodeType * parentNode * childNodes * data	hasChildNodes()	None
NodeList	Length*	item()	None
NamedNodeMap	length*	item()	None

16.2.1 DW UltraDev 对象及其特性

Dreamweaver 实现了由浏览器和 W3C 定义的标准对象，另外，也实现了两个定制对象：dreamweaver 和 site，其中 dreamweaver 对象有两个与它有关的只读特性：appName 和 appVersion。

appName 的值总是“ Dreamweaver ”，appVersion 的值的格式为“ versionNumber [languageCode](platform) ”。例如，Swedish Windosw 版本 Dreamweaver 3.0 中的 appVersion 特性值是“ 3.0 [se](win32) ”；而 English Macintosh 版本中对应的值为“ 3.0 [en](MacPPC) ”。

appName 和 appVersion 特性是在 Dreamweaver 3.0 才实现的，在以前的版本中是不可用的。如果你想测定一下当前使用的是否是 Dreamweaver 3.0 或更高版本，那么就可以通过简单地检查一下 appVersion 和 appName 特性是否存在就可以了。为了核对一个指定的 Dreamweaver 版本，首先需要检查 appVersion 特性是否存在，然后才检查版本号。例如：

```
if( dreamweaver.appVersion && dreamweaver.appVersion.indexOf( '3.01') != -1)
{
    //处理代码
}
```

site 对象没有特性。关于 dreamweaver 和 site 对象的方法的更多信息将在后面的 API 中介绍。

16.2.2 DOM

与 Netscape DOM 不同，众多的第三方书籍以及 Web 网站都没有介绍 DOM Level 1 的文档，因此，本节将详细描述 DOM Level 1 的特性和方法，和它们返回的值。

DOM Level 1 引入四个常量用以描述构成文档树的对象类型。这些常量通常作为 nodeType 特性的返回值显示，它们是：

- Node.DOCUMENT_NODE
- Node.ELEMENT_NODE
- Node.COMMENT_NODE
- Node.TEXT_NODE

16.2.3 文档对象的特性和方法

表 16-2 列出了 DW UltraDev 中 document 对象的新特性和方法，以及它们的返回值，并在适当的地方有一些解释。“*”号表示只读属性。

表 16-2 document 的新特性、方法以及返回值

方法或属性	返回值和解释
nodeType *	Node.DOCUMENT_NODE

(续表)

方法或属性	返回值和解释
parentNode *	null
parentWindow *	此 JavaScript 对象与文档的父窗口相对应 (此特性并不包括在 DOM Level 1 中, 然而, IE 4.0 支持此特性)
childNodes *	包含 document 对象的所有直接子对象的一个 nodelist。一般, 文档有一个惟一的子对象: HTML 对象
documentElement *	与 HTML 标记相对应的 JavaScript 对象。此特性是从 nodelist 中取得 document.childNodes 的值和提取 HTML 标记的简写
body *	与 BODY 标记相对应的 JavaScript 对象。此特性是从 nodelist 中调用 document.documentElement.childNodes 和提取 BODY 标记的简写。对 frameset 文档, 此特性将返回最外面的 frameset 节点
URL *	文档文件的://URL, 或, 如果文件还没有保存, 那么它将是一个空字符串
getElementsByTagName (tagName)	一个 nodelist, 用于遍历类型 tagName (例如, IMG、DIV 等) 的标记。提供与 dreamweaver.getObjectTags()函数相同的功能 如果 tag 参数是 LAYER, 那么此函数返回所有的 LAYER 和 ILAYER 标记, 以及所有绝对位置的 DIV 和 SPAN 标记 如果 tag 参数为 INPUT, 那么此函数返回所有的表格元素
hasChildNodes()	TRUE

16.2.4 HTML 标记对象的特性和方法

每一个 HTML 标记都是由一个 JavaScript 对象表示的。标记是按照树型层次结构组织的, 如果在标记 x 的开始和结束标记之间正好存在着另一个标记 y 的开始和结束标记 (即<x>x 内容包围着<y>y 内容</y></x>), 那么就可以称标记 x 为标记 y 的父标记节点。表 16-3 列出 DW UltraDev 中标记对象的特性和方法, 同时也列出了它们的返回值。“*”号表示只读特性。

表 16-3 HTML 标记对象的特性和方法

特性或方法	返回值和解释
nodeType *	Node.ELEMENT_NODE
parentNode *	父标记。如果这是一个 HTML 标记, 那么将返回文档对象
childNodes *	包含该标记所有直接子标记的 nodeList
tagName *	标记的 HTML 名称, 比如: IMG、A 或 BLINK。此特性值总是以大写字母形式返回

(续表)

特性或方法	返回值和解释
AttrName	含有指定标记属性值的字符串。如果 attrName 在 JavaScript 语言中是保留词 例如：class)，那么 tag.attrName 是不能使用的，此时，使用 getAttribute()和 setAttribute()
innerHTML	包含在开始标记和结束标记之间 HTML 代码源。例如，对代码： “ Hello,World! ”，p.innerHTML 将返回“Hello,World!”。 如果你重新设置此特性的值，那么 DOM 树会立即刷新并反映该文档的新结构(此特性并包含在 DOM Level 1 中，然而却被 IE4.0 支持)
outerHTML	此标记的 HTML 源代码，包含标记自身。例如，对于上面的代码，p.outerHTML 将返回“ Hello,World! ”。如果你设置了此特性的值，DOM 树立即刷新以反映文档的新结构 此特性并包含在 DOM Level 1 中，然而却被 IE4.0 支持)
getAttribute(attrName)	如果属性的值明确指定，将返回此属性的值，否则返回 null
getTranslatedAttribute(attrName)	指定属性的转换值，或者该属性的值没有被转换，那就返回与 getAttribute()相同的值(DOM Level 1 并不包含此特性，它是在 Dreamweaver 3.0 中增加的，以支持属性转换)
setAttribute(attrName,attrValue)	无返回值。设置指定属性的值。例如，img.setAttribute("src","image/roses.gif")
removeAttribute(attrName)	无返回值。从 HTML 对象中移走此标记指定的属性以及对应的值
getElementsByTagName(tagName)	一个 nodelist，可以用于遍历类型 tagName 的子标记，例如，IMG、DIV 等 如果 tag 参数是 LAYER，函数返回所有的 LAYER 和 ILayer 标记，以及所有绝对位置的 DIV 和 SPAN 标记 如果 tag 标记为 INPUT，函数返回所有的表格元素
hasChildNodes()	一个 Boolean 值，指示标记是否有子标记
hasTranslatedAttributes	一个 Boolean 值，指示标记是否有转换属性(DOM Level 1 并不包含此特性，它是在 Dreamweaver 3.0 中增加的，以支持属性转换)

16.2.5 文本对象的的特性和方法

HTML 文档中每一个邻近的块(例如，在 p 标记内的文本)都是由一个 JavaScript 对象表示的。文本对象从来不会有子对象。表 16-4 列出了 DW UltraDev 中文本对象的特性和方法。同时也列出了它们的返回值。“*”号表示只读特性。

表 16-4 文本对象的特性和方法

特性或方法	返回值和解释
nodeType *	Node.TEXT_NODE
parentNode *	父标记
childNodes *	一个空的 nodeList
data	实际的文本字符串。文本中的内容是由一个单独的字符表示的。例如，文本：Joseph & I 将返回：Joseph & I
hasChildNodes()	FALSE

16.2.6 注释对象的特性和方法

每一个 HTML 注释都是由 JavaScript 对象表示的。表 16-5 列出了 DW UltraDev 中注释对象的方法和属性。“*”号表示只读特性。

表 16-5 注释对象的特性和方法

特性或方法	返回值和解释
nodeType *	Node.COMMENT_NODE
parentNode *	父标记
childNodes *	一个空的 nodeList
data	在注释标记（<!--和-->）之间的文本字符串
hasChildNodes()	FALSE

16.3 树控制

DW UltraDev 利用两个新的控制——数据库树控制和可变的网格控制，扩展 HTML 选择标记。数据库树控制对显示数据库模式信息是非常有用的，即，显示组成数据库的表和列。可变网格控制对显示任意表格式的信息是非常有用的。在 DW UltraDev 4 中，高级 Recordset 对话框中利用数据库树控制和可变网格控制显示有关信息。如图 16-1 所示。

1. 增加一个数据库树控制

为了在对话框中增加一个数据库树控制，在你的页面增加以下的 HTML 代码：

```
<select name="DBTree" style="width:400px;height:110px" type="mmdatabasetree connection=""></select>
```

数据库树控制有下列属性：

- name="DBTree"，这是数据库树控制的名称。

- style="width:x;height:n", 这是控制的大小。我们可以以像素为单位指定控制的宽度和高度。
- type=mm databasetree, 这是控制的类型。值 mm databasetree 指示此标记应该作为数据库树显示, 并且应该显示在 connection 属性中指定的连接的内容。
- connection="connectionName", 指示一个 DW UltraDev 连接的名称 (与在 Connection Manager 中定义的一样)。如果 connection 属性是空的, 那么此控制也将是空的。
- 在 select 标记内容放置的任意标记都将被忽略。

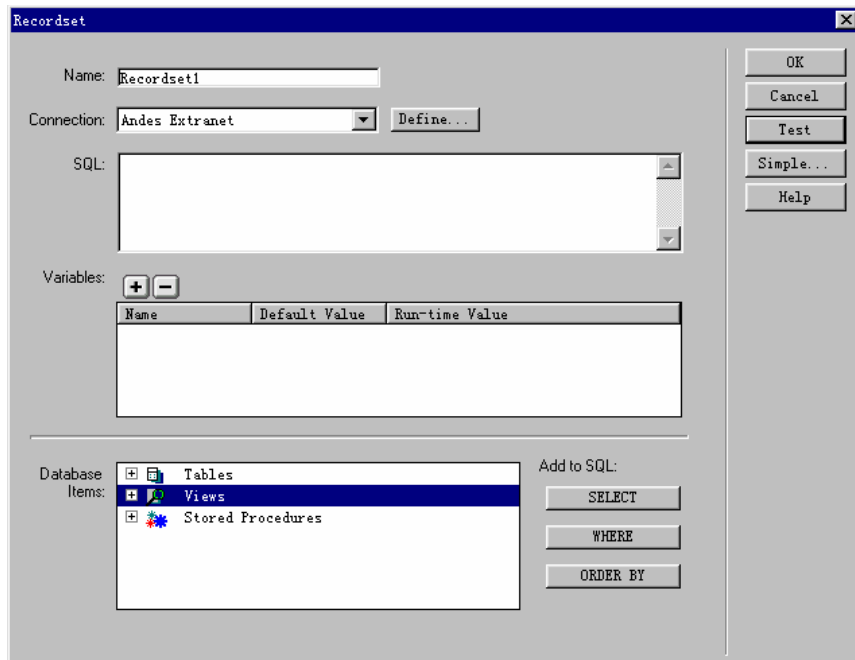


图 16-1
高级
Recordset
对话框

我们利用 JavaScript 代码改变 connection 属性, 并且检索树中选择的数据。我们可以利用 DBTreeControl 作为新标记的一个 JavaScript 包装对象。我们可以在 DBTreeControlClass.js 文件中发现这个对象的实现, 此文件位于 Configuration\Shared\UltraDev\Script 路径下。

2. 增加一个可变网格控制

为了在对话框中增加一个可变网格控制, 在你的页面增加以下的 HTML 代码:

```
<select name="ParamList" style="width:515px;" type=mmparameterlistcolumns="Name,SQL Data
Type,Direction,Default Value,Run-time Value" size=6></select>
```

可变网格控制拥有以下属性:

- name="ParamList", 这是可变网格控制的名称。
- style="width.x", 这是控制的宽度。
- type=mmParameterList, 这是控制的类型。值 mmParameterList 指示此标记将作为一个可变网格控制绘制。
- columns="colName1", "colNameX", 此属性指定列的名称, 各列之间用逗号分隔。因为每一列都必须有一个名称, 此属性不仅指定了列的名称, 同时也指定了控制中列的数目。也就是说, 指定的名称数目也就是列的数目。在上面的例子中, columns 属性被指定为: columns="Name,SQL Data Type,Direction,Default Value,Run-time Value", 此指定创建五个列: Name、SQL Data Type、Direction、Default Value 和 Run-time Value。
- size=n, 此属性指定可视行的数目。
- columnWidth="x,...", 此属性为每一列指定宽度, 它们之间用逗号分隔。如果没有指定宽度, 所有的列都具有相同的宽度。我们也可以只指定前面几列的宽度, 并不指定所有列的宽度, 那么剩余列的宽度都是相等的, 并且还得与控制减去前面几列宽度之后剩余的空间相匹配。例如, 假设我们指定下列 select 标记:

```
<select name="ParamList" style="width:500px;" type=mmparameterlistcolumns="Name,SQL Data Type,Direction, Default Value,Run-time Value" columnWidth="100,25,11," size=6></select>
```

此代码将创建一个 500 像素宽的可变网格控制。它有五列: Name、SQL Data Type、Direction、Default Value 和 Run-time Value。其中 Name 列的宽度为 100 像素, SQL Data Type 列 25 像素宽, 而 Direction 列是 11 像素宽。那么, 剩余的两列都是 182 像素宽。(指定列宽度总和为 136, 控制的总宽度为 500, 那么减去前三列之后剩余空间为 364 像素。现在还剩下 2 列, 364 除 2 等于 182。)

可变网格控制对象也有一个 JavaScript 包装对象, 用于访问和操纵控制的数据。我们可以在 GridControlClass.js 文件中找到此对象的实现, 该文件位于 Configuration\Shared\MM\Scripts\Class\路径下。

16.4 JavaScript 在扩展中的工作方式

16.4.1 DW UltraDev 4 处理扩展的方式

当 DW UltraDev 4 处理扩展时, 它编译 SCRIPT 标记之间的所有内容, 并执行位于 SCRIPT 标记(但它不是一个函数声明的一部分, 例如, 全局变量的初始化)之内的任意代码。它也读入、编译并执行 SCRIPT 标记的 SRC 属性指定的外部 JavaScript 文件

中的 scripts。



如果在你的扩展文件中任意 JavaScript 代码中包含有字符串“</SCRIPT>”，那么 JavaScript 解释器将按照实际的结束 SCRIPT 标记读取它，并且报告一个无结束字符串文字错误。为了避免这个问题，将此字符串分成几部分，并将它们连接起来，就像这样：“<”+“ /SCRIPT>”。

如果 JavaScript 代码在 command 和 Behavior Action 文件中，当用户从菜单上选择命令或行为时，DW UltraDev 执行 onLoad 事件处理器中的代码。如果 JavaScript 代码在对象文件中，如果文档的 body 包含一个表格，那么 DW UltraDev 执行 BODY 标记上的 onLoad 事件处理器中的代码。DW UltraDev 忽略在 Data Translator、Property Inspector 和 Floating Palette 文件中的 BODY 标记上的 onLoad 事件处理器。在所有的扩展中，当用户与表格字段交互时，DW UltraDev 在另一事件处理器中执行代码，例如，onBlur=“ alter ('This is a required field.') ”。

DW UltraDev 扩展不支持 Links (一个标记，包括这些 JavaScript URL，例如：)，也不支持 document.write() 语句。Plug-ins 在扩展的 BODY 中支持，但是 Java Applets 和 ActiveX 控制却不被支持。

16.4.2 在启动和关闭时运行 scripts

在 Dreamweaver 3.0 中，如果我们在 Configuration/Startup 文件夹中放置一个命令文件，那么当 Dreamweaver 启动时此命令也将执行。Startup 命令是在 menus.xml 文件、ThirdPartyTags 文件夹下的文件以及其他 Commands、Objects、Behaviors、Inspectors、Floating Palettes 或 Translators 之前装载的。这样，我们可以利用 Startup 命令去修改 menus.xml 文件或其他扩展文件。我们也可以给用户显示警告或提示信息，但我们不能调用 dreamweaver.runCommand()。

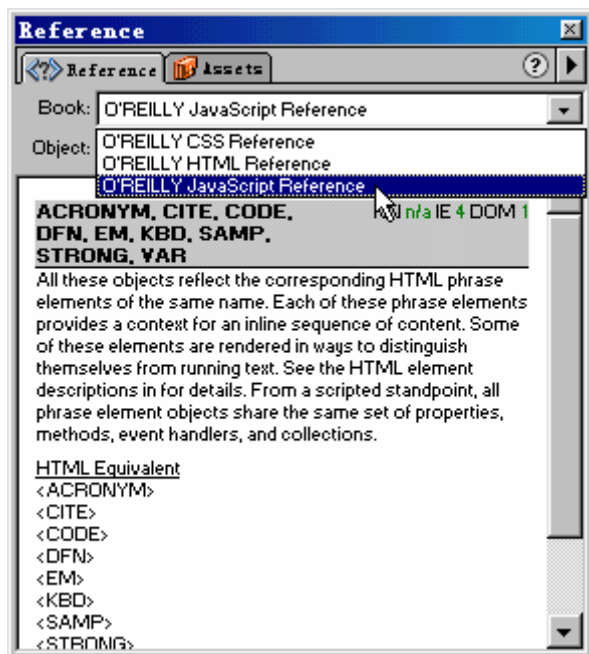
同样，如果我们在 Configuration/Shutdown 文件夹下放置了一个命令文件，那么当 DW UltraDev 4 关闭时，此命令也会被执行。我们可以从关闭命令中调用 dreamweaver.runCommand()，从而为用户显示警告或提示信息，但是这样的话，就不能关闭进程。

这里给读者一个小小的建议，如果你想实现 DW UltraDev 4 扩展功能，手边最好有一本好的 JavaScript 语言参考书，随时可以查阅一些语法问题。当然，如果读者的英文水平足够好的话，也可以借助于 DW UltraDev 4 新增的 Reference 面板，如图 16-2 所示。其中给出了三本书的参考：O'REILLY CSS Reference、O'REILLY HTML Reference 和 O'REILLY JavaScript Reference。当使用过程中，如果遇到不明白的地方，可以在这里得

到非常详细的英文解释。

图 16-2

Reference
面板



第 17 章 DW UltraDev 4 API 及其使用

本章概要：

- ☒ DW UltraDev 4 的 JavaScript API
- ☒ **文件** I/O API
- ☒ Design Notes API
- ☒ Database API
- ☒ Commands API
- ☒ Data Sources API
- ☒ Server Behaviors API
- ☒ **其他** API

对象、特性以及其他在本章介绍的方法为扩展 DW UltraDev 提供了一个良好的基础。为了创建更有价值的 DW UltraDev 扩展，以及使定制 DW UltraDev 的菜单成为可能，除了基于标准的 DOM 方法以外，DW UltraDev 4 还为用户提供了 400 多个 JavaScript 函数以及其他类型的 API，这是一个非常大的扩充。可以这么说，在 DW UltraDev 中用菜单、浮动面板、检查器、Site 窗口或 Document 窗口实现的所有任务，几乎都可以利用这些 JavaScript 和其他类型的 API 函数完成。本章为读者介绍这些函数的功能、使用方法和示例，由于函数太多，无法一一列举，所以对没有提到的函数，在编程时，请读者参考 DW UltraDev 提供的帮助。

17.1 DW UltraDev 4 的 JavaScript API

DW UltraDev 4 的 JavaScript API 函数是按下列顺序分组的：首先是功能，然后是对象的字母顺序，最后是方法名称。例如，有关处理创建、应用和删除 CSS 风格的方法都归类在 CSS Style 函数中。Dom 对象的方法首先列出来，接着是 Dreamweaver 对象的方

法, 然后是 `dreamweaver.cssStylePalette` 对象的方法。可选参数放在一对花括号“ {} ”内。

17.1.1 API 函数概述

1. 理解 API 中的对象

本章以下几节的所有定制函数都是 Dreamweaver 对象、Site 对象或表示一个文档的 DOM 对象的方法。这里, 适合后者类别的方法是按这种样式列出的 `dom.functionName()`。为了达到期望的结果, 你必须首先得到文档的 DOM, 并且作为该 DOM 的方法调用函数。而不能简单地键入 `dom.functionName()`。例如:

```
var currentDOM = dreamweaver.getDocumentDOM( 'document' );
currentDOM.setSelection( 100,200 );
currentDOM.clipCopy();
var otherDOM = dreamweaver.openDocument( dreamweaver.getSiteRoot()+ "html/foo.htm" );
otherDOM.endOfDocument();
otherDOM.clipPaste();
```

除非有特殊注释, DOM 对象的方法只能在打开的文档中工作。如果在没有打开的文档上运行一个函数, 将会导致 DW UltraDev 报告一个错误。

在 DW UltraDev 4 中, `dw` 是 `dreamweaver` 的同义词, 这样, 所有的 `dreamweaver` 对象方法都可以按照 `dw.functionName()` 样式引用。在以下几节的代码例子中都是以此符号表示的。

2. 菜单使能

因为 DW UltraDev 4 中的所有菜单项都是作为一个 JavaScript 函数实现的, 所以要求一种 JavaScript 机制以决定菜单项的使能, 这种机制是一系列函数, 称之为 `enablers`。

一个 `enabler` 决定了与它相关联的主函数是否能在当前上下文中运行。例如, `site.canGet()` 决定了 DW UltraDev 是否能完成一个 `Get` 函数 (`site.get()`)。

如果 `enabler` 可用的话, 以下的 API 函数说明都列出了此函数的 `enabler`。有一些函数没有 `enabler`, 这是因为要么与此函数相关联的菜单项总是处于使能状态的, 要么因为此函数与菜单没有关系。

17.1.2 最重要的函数

实际应用中, 与 DOM 操作有关的所有函数都要求首先决定将要改变的是哪个

DOM。此任务是由 `dreamweaver.getDocumentDOM()` 函数完成的，因此它也就成为最重要的函数。下面让我们来看看该函数的一些特性。

※ **dreamweaver.getDocumentDOM()**

适用版本 Dreamweaver 2.0 以上。

描述 为用户提供访问指定文档对象树的手段。调用者利用此函数得到对象树之后，就可以编辑该树从而改变文档的内容。

参数 `sourceDoc`。

此参数必须是下列几种形式之一：“`document`”、“`parent`”、“`parent.frames[number]`”，或是 `URL.document`，它指定了拥有光标焦点并包含当前选定内容的文档。其中，`parent` 指定了父框架集（如果当前选定的文档在一个框架集中），`parent.frames[number]` 和 `parent.frames['frameName']` 指定一个文档，该文档是在包含当前文档的框架集中一个特殊框架内。如果参数是一个相对的 URL，那么它是相对于扩展文件的。在 DW UltraDev 4 中，`sourceDoc` 的缺省值为 `document`。



如果参数为“`document`”，那么为了具有编辑文档的功能，调用者必须是 `applyBehavior()`、`deleteBehavior()`、`objectTag()`，或者是在命令或属性检查器文件中任意函数。

返回 位于树根处的 JavaScript 文档对象。

Enabler 无。

例子 下面的代码段利用 `dreamweaver.getDocumentDOM()` 函数访问当前文档的背景颜色。

```
var theDOM = dreamweaver.getDocumentDOM( "document" ) ;
theDOM.body.bgcolor = "#000000";
```

17.1.3 Behavior 函数

Behavior 函数允许我们为一个对象增加或移走动作、查找隶属于对象的动作、获取动作隶属对象的信息等有关动作方面的功能。`dreamweaver.behaviorInspector` 对象的方法可以控制或作用于 **Behaviors** 检查器中的选择对象，但不能是当前文档中的选择对象。下面就让我们来看看 **Behavior** 函数。

※ **dom.addBehavior()**

适用版本 Dreamweaver 3.0 以上。

描述 为选定元素增加一新的事件/动作对。此函数仅对活动文档有效。

参数 `event`，`action`，`{eventBasedIndex}`。

- 第一个参数是 JavaScript 事件句柄，用于将动作附属到元素上，例如，onClick、onMouseOver 或 onLoad。
- 如果动作是利用 Behaviors 检查器加入的，那么第二个参数是 applyBehavior() 返回的函数调用。例如：“MM_popupMsg('Hello Wordl')”。
- 第三个参数是该动作应该加入的位置。eventBasedIndex 是一个以 0 为基数的索引值，如果对某一特定的事件已经有两个动作与之相关，并且我们指定 eventBaseIndex 为 1，那么此动作将在其他两个动作之间被执行。如果忽略此参数，那么此动作将被加到指定事件已有的所有动作之后。

返回 无。

Enabler 无。

※ dom.getBehavior()

适用版本 Dreamweaver 3.0 以上。

描述 获取指定事件内指定位置处的动作。此函数作用于当前选定对象，并且它仅对活动文档有效。

参数 event, {eventBasedIndex}。

- 第一个参数是 JavaScript 事件句柄，通过它将动作附属到元素上，例如，onClick、onMouseOver 或 onLoad。
- 第二个参数是希望获取的动作所在位置。如果对某一指定的事件已经有两个动作与之相关，那么 0 表示第一个动作，1 表示第二个。如果忽略此参数，那么隶属于指定事件的所有动作都将被返回。

返回 返回一个表示函数调用的字符串 例如：“MM_swapImage('document.Image1', 'document.Image1', 'foo.gif', #933292969950')”，或者如果 eventBasedIndex 被忽略，那么将返回一个字符串数组。

Enabler 无。

※ dom.reapplyBehaviors()

适用版本 Dreamweaver 3.0 以上。

描述 检查以确保与任何在特定节点上的动作调用相关的函数必须位于文档的 HEAD 部分，如果在该位置处没有发现它们，那么就完成插入工作。

参数 {elementNode}。

该参数是在当前文档中的一个元素节点。如果忽略此参数，那么对单个动作调用，DW UltraDev 将检查文档中的所有元素节点。

返回 无。

Enabler 无。

※ dom.removeBehavior()

适用版本 Dreamweaver 3.0 以上。

描述 移走指定事件的指定位置处的函数。此函数作用于当前选定对象上，并且仅对活动文档有效。

参数 event,{eventBasedIndex}。

- 第一个参数是 JavaScript 事件句柄，通过它将动作附属到元素上，例如，onClick、onMouseOver 或 onLoad。如果此参数被忽略，那么元素上的所有动作都将被移走。
- 第二个参数指定将要被移走的动作所在的位置。例如，如果与指定事件相关的有两个动作，0 表示第一个动作，而 1 则表示第二个动作。如果忽略此参数，那么对指定事件的所有动作都将被移走。

返回 无

Enabler 无。

※ dreamweaver.getBehaviorElement()

适用版本 Dreamweaver 2.0 以上。

描述 获取与动作作用的标记相对应的 DOM 对象。此函数仅在动作文件内适用。

参数 无。

返回 一个 DOM 对象或 null。在下列情况下，此函数返回 NULL：

- 当前脚本不是在 Behaviors 检查器的上下文中执行。
- 利用 Behaviors 检查器编辑时间轴的动作。
- 当前执行的脚本是由 dreamweaver.popupAction()激活的。
- Behaviors 检查器正在将一个事件附属到一个链接封装对象上，并且该链接封装对象还没有存在。
- 此函数出现在一个动作文件外部。

Enabler 无。

例子 dreamweaver.getBehaviorElement()可以像 dreamweaver.getBehaviorTag()函数一样使用，以决定选择的动作是否适合于选择的 HTML 标记，除非它允许你访问有关该标记及其属性更多的信息。例如，如果你写了一个动作，仅适用于一个不是将其他框架或窗口作为目标的超链接(A HREF)，那么你可以利用 getBehaviorElement()作为针对参数对话框初始化用户接口的函数的一部分。

```
function initializeUI(){
var theTag = dreamweaver.getBehaviorElement();
var CANBEAPPLIED =( theTag.tagName == "A"
```

```

        && theTag.getAttribute( "HREF" ) != null
        && theTag.getAttribute( "TARGET" ) == null) ;
if( CANBEAPPLIED ) {
    // display the action UI
} else{
    // display a helpful message that tells the user
    // that this action can only be applied to a
    // hyperlink without an explicit target]
}
}
}

```

※ **dreamweaver.getBehaviorTag()**

适用版本 Dreamweaver 1.2 以上。

描述 获取动作所作用的源标记。此函数仅在动作文件内适用。

参数 无。

返回 表示源标记的一个字符串,与作为一个参数 HTMLelement)传递给 canAccept Behavior()函数的字符串相同。如果此函数并不在动作文件内,那么返回值将是一个空字符串。

Enabler 无。

例子 如果你编写了一个动作,它仅适用于一个超链接(A HREF),那么你可以利用 getBehaviorTag()作为参数对话框的用户接口初始化函数的一部分。

```

function initializeUI(){
var theTag = dreamweaver.getBehaviorTag().toUpperCase();
var CANBEAPPLIED =( theTag.indexOf( 'HREF' ) != -1) ) ;
if( CANBEAPPLIED ) {
    // display the action UI
} else{
    // display a helpful message that tells the user
    // that this action can only be applied to a
    // hyperlink
}
}
}

```

※ **dreamweaver.popupAction()**

适用版本 Dreamweaver 2.0 以上。

描述 为用户展现指定动作的参数对话框。对用户来说,这相当于在 Behaviors 检

查器中从 Actions 弹出菜单中选择该动作。此函数允许扩展文件而不是 Actions 将动作附属到用户文档中的对象上。此函数将挂起其他编辑，直到用户关闭此对话框。

注意，此函数仅可以在 `objectTag()` 内调用，或者在一个命令或属性检查器文件内的任意脚本中调用。

参数 `actionName,{funcCall}`。

- 第一个参数是一个包含一个 JavaScript 动作的文件名称，该文件位于 Configuration/Behaviors/Actions 文件夹下，例如，“Timeline/play Timeline.htm”。
 - 第二个参数是一个字符串，该字符串包含了一个对在 `actionName` 中指定的动作的函数调用。例如，“MM_playTimeline(...)”。
- 如果要指定此参数，那么它将由动作文件内的 `applyBehavior()` 函数提供。

返回 返回对动作的函数调用。当用户单击参数对话框的 OK 按钮后，动作就被加入到当前文档。其中，适当的函数被加入到文档的 HEAD 部分；HTML 会被加入到文档的 BODY 部分；其他编辑文本将会构成文档。此函数调用（例如，“MM_playTimeline(...)”）不会加入到文档，但将返回此函数的返回值。

Enabler 无。

※ `dreamweaver.behaviorInspector.getBehaviorAt()`

适用版本 Dreamweaver 3.0 以上。

描述 获取在 Behavior 检查器内指定位置处的事件/动作对。

参数 `positionIndex`。

返回 此函数返回一个数组，该数组只有两个元素：

- 一个事件句柄
- 一个函数调用或 JavaScript 语句

Enabler 无。

例子 因为 `positionIndex` 是基于 0 索引的，所以如果 Behaviors 检查器的列表如图 17-1 所示，那么调用 `dw.behaviorInspector.getBehaviorAt( 1)` 将返回一个包含两个字符串的数组：“onMouseOver”和“MM_changeProp('document.moon','document.moon','src','sun.gif',IMG)”。

※ `dreamweaver.behaviorInspector.getBehaviorCount()`

适用版本 Dreamweaver 3.0 以上。

描述 通过事件句柄计算附属于当前选定元素的动作的数目。

参数 无。

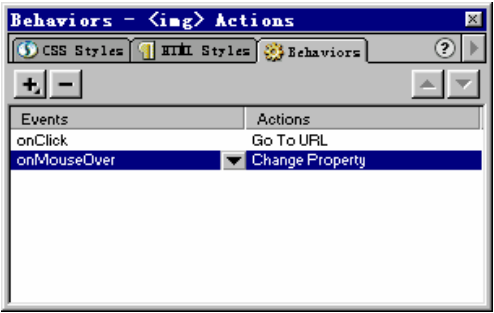
返回 返回一个整数，代表了隶属于选定元素的动作的数目。此数目与在 Behaviors 检查器中显示的动作数目相等，并且包括了 Dreamweaver 动作和定制 JavaScript 动作。

Enabler 无。

图 17-1

Behaviors

面板



例子 一个针对下面代码所示的选定链接的函数调用 `dw.behaviorInspector.getBehaviorCount()` 将返回数值 2。

```
<A HREF="javascript:setCookie()" onClick="MM_popupMsg( 'A cookie has been set.' );parent.  
rightframe.location.href='aftercookie.html'">
```

✳ **`dreamweaver.behaviorInspector.getSelectedBehavior()`**

适用版本 Dreamweaver 3.0 以上。

描述 获得在 Behaviors 检查器上选定的动作的位置。

参数 无。

返回 返回一个整数，代表了在 Behaviors 检查器上选定的动作的位置，如果没有选定动作，则该返回值为 -1。

Enabler 无。

例子 如果在 Behaviors 检查器上的第一个动作被选定，如图 17-2 所示的那样，那么对 `dw.behaviorInspector.getSelectedBehavior()` 的调用将返回 1 值。

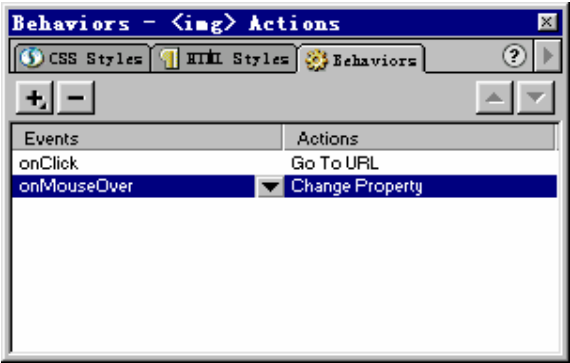
图 17-2

选定第一

个动作的

Behaviors

检查器



✳ **`dreamweaver.behaviorInspector.moveBehaviorDown()`**

适用版本 Dreamweaver 3.0 以上

描述 通过在事件范围内改变动作的执行顺序，从而将动作在 Behaviors 检查器中

显示的顺序下移一位。

参数 positionIndex。

此参数是动作在 Behaviors 检查器中的位置。检查器动作列表第一个元素位置为 0。

返回 无返回值。

Enabler 无。

例子 如果 Behaviors 检查器开始如图 17-3 所示, 那么调用 dw.behaviorInspector.moveBehaviorDown() 将交换事件 onMouseOut 上的 Change Property 动作和 Swap Image Restore 动作的顺序。本例中, 如果再以其他位置调用 dw.behaviorInspector.moveBehaviorDown() 函数将不会产生任何效果, 因为本例中, onClick 和 onMouseOver 事件都只有一个动作, 另外, 处于位置 2 的动作是已经位于 onMouseOut 事件动作集的底部, 不可能再向下移动。



图 17-3
Behaviors
动作检查
器

※ dreamweaver.behaviorInspector.moveBehaviorUp()

适用版本 Dreamweaver 3.0 以上。

描述 通过在事件范围内改变动作的执行顺序, 从而将动作在 Behavior 检查器中显示的顺序上移一位。

参数 positionIndex。

此参数是动作在 Behaviors 检查器中的位置。检查器动作列表的第一个元素位置为 0。

返回 无返回值。

Enabler 无。

例子 如果 Behavior 检查器开始如图 17-3 所示, 那么调用 dw.behaviorInspector.moveBehaviorUp() 将交换 Custom Script 动作和事件 onMouseOver 上的 Change Property 动作的顺序。本例中, 如果再以其他位置调用 dw.behaviorInspector.moveBehaviorDown() 函数将不会产生任何效果, 因为本例中, onClick 和 onMouseOut 事件都只有一个动作, 另外, 处于位置 2 的动作是已经位于 onMouseOver 事件动作集的顶部。

※ **dreamweaver.behaviorInspector.setSelectedBehavior()**

适用版本 Dreamweaver 3.0 以上。

描述 选定 Behaviors 检查器上指定位置处的动作。

参数 positionIndex。

此参数是需要选定的动作在 Behaviors 检查器中的位置。检查器动作列表的第一个元素位置为 0。如果要取消所有的选定，可以将 positionIndex 的值设置为-1。如果指定的位置处没有动作，这时，此函数作用的相当于 positionIndex 的值设置为-1。

返回 无返回值。

Enabler 无。

例子 如果 Behavior 检查器开始如图 17-3 所示，那么调用 dw.behaviorInspector.setBehaviorSelection(3) 将选定与 onMouseOver 事件相关联的 Change Property 动作。

17.1.4 剪贴板函数

所有的剪贴板函数都是与剪切、拷贝和粘贴有关。在 Macintosh 系统上，一些剪贴板函数也可以应用于对话框和浮动面板上的编辑域。那些可以作用于编辑域上的函数是既作为 Dreamweaver 对象的方法，也是 DOM 对象的方法实现的。其中，Dreamweaver 版本的函数的操作对象是拥有控制焦点的窗口上的选定对象，这里窗口可以是：当前 Document 窗口、HTML 检查器或 Site 窗口。对 Macintosh 系统，剪贴板函数也可以对拥有焦点的编辑域中的选定对象进行操作。DOM 版本的函数总是针对指定文档上的选定对象进行操作的。

※ **dom.clipCopy()**

适用版本 Dreamweaver 3.0 以上。

描述 向剪贴板拷贝选定对象，包括那些定义了选定对象的 HTML 标记。

参数 无。

返回 无返回值。

Enabler 无。

※ **dom.clipCopyText()**

适用版本 Dreamweaver 3.0 以上。

描述 向剪贴板拷贝选定文本，忽略所有的 HTML 标记。

参数 无。

返回 无返回值。

Enabler dom.canClipCopyText()。请参见 Enabler 函数部分。

※ **dom.clipCut()**

适用版本 Dreamweaver 3.0 以上。

描述 向剪贴板移动选定对象，包括那些定义了选定对象的 HTML 标记。

参数 无。

返回 无返回值。

Enabler 无。

※ **dom.clipPaste()**

适用版本 Dreamweaver 3.0 以上。

描述 将剪贴板中的内容粘贴到当前文档当前插入位置处，或代替当前选定对象。

如果剪贴板中的内容包含有 HTML，对它的操作也是同样的。

参数 无。

返回 无返回值。

Enabler dom.canClipPaste()。请参见 Enabler 函数部分。

※ **dom.clipPasteText()**

适用版本 Dreamweaver 3.0 以上。

描述 将剪贴板中的内容粘贴到当前文档当前插入位置处，或代替当前选定对象，并且用 BR 标记代替剪贴板内容中的任何换行符。如果剪贴板中的内容包含有 HTML 语句，它将被解释，“<”符号将被当作符号“<”粘贴，而“>”符号将被当作符号“>”粘贴。

参数 无。

返回 无返回值。

Enabler dom.canClipPasteText()，请参见 Enabler 函数部分。

※ **dreamweaver.clipCopy()**

适用版本 Dreamweaver 3.0 以上。

描述 向剪贴板拷贝当前选定的内容，选定内容的来源是拥有焦点的 Document 窗口、对话框、浮动面板或者 Site 窗口上的内容。

参数 无。

返回 无返回值。

Enabler dreamweaver.canClipCopy()，请参见 Enabler 函数部分。

※ **dreamweaver.clipCut()**

适用版本 Dreamweaver 3.0 以上。

描述 向剪贴板移动选定对象，选定对象可以是拥有焦点的 Document 窗口、对话框、浮动面板，或者是 Site 窗口上的内容。

参数 无。

返回 无返回值。

Enabler dreamweaver.canClipCut()，请参见 Enabler 函数部分。

✧ **dreamweaver.clipPaste()**

适用版本 Dreamweaver 3.0 以上。

描述 将剪贴板中的内容粘贴到当前文档、对话框、浮动面板，或者 Site 窗口。

参数 无。

返回 无返回值。

Enabler dreamweaver.canClipPaste()。请参见 Enabler 函数部分。

✧ **dreamweaver.getClipboardText()**

适用版本 Dreamweaver 3.0 以上。

描述 获取剪贴板上存储的所有文本。

参数 无。

返回 如果剪贴板存储了文本内容，那么此函数返回一个包含剪贴板内容字符串；否则，返回值为空。

Enabler 无。

17.1.5 命令函数

命令(Command) 函数将帮助我们构造 Configuration/Commands 文件夹下的大部分文件。它们管理命令菜单，并且从其他类型的扩展文件中调用命令。

✧ **dreamweaver.editCommandList()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Edit Command List 对话框。

参数 无。

返回 无。

Enabler 无。

✧ **dreamweaver.runCommand()**

适用版本 Dreamweaver 3.0 以上。

描述 执行指定的命令。对用户来说，此函数的效果相当于从菜单中选择菜单命令。如果一个对话框与指定的命令有关，那么将显示该对话框。此函数为用户提供了从其他扩展文件中调用命令的能力。

注意，此函数仅能在 objectTag() 函数内调用，或在命令的任意脚本、属性检查器文件或菜单中调用。

参数 commandFile, {commandArg1}, {commandArg2}...{commandArgN}。

- 第一个参数是 Configuration/Commands 文件夹下的一个文件名称。

- 第二个参数以及剩余的其他的参数都是作为参数传递给 commandFile 文件的。

返回 无。

Enabler 无。

例子 我们可以为表格编写一个定制属性检查器，以便允许用户通过从按钮的 onClick 事件句柄调用下列函数，就可以实现从检查器上的一个按钮得到 Format Table 命令。

```
function callFormatTable(){
    dw.runCommand( 'Format Table.htm' ) ;
}
```

17.1.6 转换函数

转换 (Conversion) 函数主要处理的是将表格转换为层，或相反，将层转换为表格，以及将 Cascading Style Sheet (CSS) 风格转换为 HTML 标记。每个函数都正确地复制了在文件 (File) 或修改 (Modify) 菜单中的一个转换命令的动作。

※ **dom.convertLayersToTable()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Convert Layers to Table 对话框。

参数 无。

返回 无。

Enabler dom.canConvertLayersToTable(), 请参见 Enabler 函数部分。

※ **dom.convertTablesToLayers()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Convert Table to Layers 对话框。

参数 无。

返回 无。

Enabler dom.canConvertTablesToLayers(), 请参见 Enabler 函数部分。

※ **dom.convertTo30()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Convert to 3.0 Compatible 对话框。

参数 无。

返回 无。

Enabler 无。

17.1.7 CSS 风格函数

CSS 风格函数处理 CSS 风格的应用、移动、创建和删除。其中，dreamweaver.css

StylePalette 对象的方法控制 Style 调色板上的选定对象，或者说这些方法只对 Style 调色板上的选定对象起作用，而不是当前文档中的。

※ dom.applyCSSStyle()

适用版本 Dreamweaver 3.0 以上。

描述 将指定的风格应用到指定的元素上。此函数仅对活动文档有效。

参数 elementNode,styleName,{classOrID}。

- 第一个参数是在 DOM 中的一个元素节点。如果指定 elementNode 为 NULL 或是空字符串，那么此函数对当前选定对象起作用。
- 第二个参数是一个 CSS 风格的名称。
- 第三个参数是风格应该应用的属性，可以是“class”或“id”。如果 elementNode 指定为 NULL 或空字符串，并且选定内容正好没有被标记包围，那么将利用 SPAN 标记应用指定风格。如果选定是一个插入点，DW UltraDev 能推断并决定风格应该应用于哪个标记。

返回 无。

Enabler 无。

例子 下列代码将 red 风格应用到选定内容上，方法是利用 SPAN 标记包围选定，要么将 CLASS 属性应用到包围选定的标记上。

```
var theDOM = dreamweaver.getDocumentDOM( 'document' );
theDOM.applyCSSStyle( "red" );
```

※ dom.removeCSSStyle()

适用版本 Dreamweaver 3.0 以上。

描述 从指定元素移走 CLASS 或 ID 属性，或者移走完全包围指定元素的 SPAN 标记。此函数仅对活动文档有效。

参数 elementNode,{classOrID}。

- 第一个参数是在 DOM 内的一个元素节点。如果 elementNode 被指定为空字符串，那么此函数将对当前选定内容起作用。
- 第二个参数是应该移走的属性，可以是“class”或“id”。如果 classOrID 没有被指定，它的缺省值为“class”。如果指定的 elementNode 节点没有定义 CLASS 属性，那么包围 elementNode 的 SPAN 属性将被移走。

返回 无。

Enabler 无。

※ dreamweaver.cssStylePalette.deleteSelectedStyle()

适用版本 Dreamweaver 3.0 以上

描述 从文档中删除当前 Style 调色板中选定的风格。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.cssStylePalette.duplicateSelectedStyle()**

适用版本 Dreamweaver 3.0 以上。

描述 复制当前在 Style 调色板中选定的风格，并且显示 Duplicate Style 对话框以允许用户为新风格设置一个名称或选择器。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.cssStylePalette.editSelectedStyle()**

适用版本 Dreamweaver 3.0 以上。

描述 为当前在 Style 调色板内选定的风格打开 Style Definition 对话框。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.cssStylePalette.editStyleSheet()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Edit Style Sheet 对话框。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.cssStylePalette.getSelectedStyle()**

适用版本 Dreamweaver 3.0 以上。

描述 得到当前在 Style 调色板中选定的风格的名称。

参数 无。

返回 返回表示选定风格的名称的一个字符串，或者如果当前没有任何一个风格被选定，那将返回一个空字符串。

Enabler 无。

例子 如果选定风格 Red，如图 17-4 示，那么调用函数 `dw.cssStylePalette.getSelectedStyle()` 将返回“ Red ”。

※ **dreamweaver.cssStylePalette.getSelectedTarget()**

适用版本 Dreamweaver 3.0 以上。

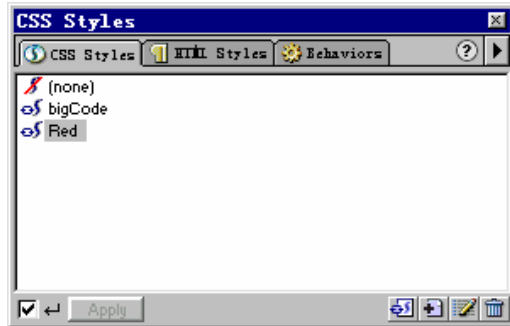
描述 得到 Style 调色板顶部的 Apply To 弹出菜单中选定的元素。

参数 无。

返回 返回应该应用风格的对象，或者，如果目标就是当前选定，那将返回 NULL。

Enabler 无。

图 17-4
CSS 风格
对话框



- ✧ **dreamweaver.cssStylePalette.getStyles()**
 适用版本 Dreamweaver 3.0 以上。
 描述 获得活动文档内所有类风格的列表。
 参数 无。
 返回 返回表示文档内所有类风格名称的字符串数组。
 Enabler 无。
 例子 假设 CSS Style 调色板初始如图 17-6 所示，那么调用函数 dw.cssStylePalette.getStyles()将返回包含下列字符串的一个数组：

“ bigCode ”
 “ red ”
 “ sectionHead ”

- ✧ **dreamweaver.cssStylePalette.newStyle()**
 适用版本 Dreamweaver 3.0 以上。
 描述 打开 New Style 对话框。
 参数 无。
 返回 无。
 Enabler 无。

17.1.8 外部应用函数

外部应用函数处理的是某些浏览器和外部编辑器有关的操作，这些浏览器和外部编辑器是在 Preferences 对话框中的 Preview in Browser 和 External Editors 特性中定义的。这些函数允许我们获得这些外部应用的有关信息，并可以利用它们打开文件。

- ✧ **dreamweaver.browseDocument()**
 适用版本 Dreamweaver 2.0 以上。

描述 在指定浏览器中打开指定的 URL。

参数 fileName,{browserName}。

- 第一个参数是将被打开的文件名称，是当作一个绝对 URL 表示的。
- 第二个参数是在 Dreamweaver 3.0 新加的，是在 Preview in Browser 特性对话框中定义的浏览器的名称。如果忽略此值，那么此参数缺省为用户的主浏览器。

返回 无。

Enabler 无。

例子 下面的代码利用函数 `dreamweaver.browseDocument()` 在浏览器中打开 Hotwired 主页。

```
function goToHotwired(){
    dreamweaver.browseDocument( 'http://www.hotwired.com/' );
}
```

在 Dreamweaver 3.0 中，我们可以利用下面的代码扩展此操作在 IE 中打开文档。

```
function goToHotwired(){
    var prevBrowsers = dw.getBrowserList();
    var theBrowser = "";
    for( var i=1; i < prevBrowsers.length; i+2) {
        if( prevBrowsers[i].indexOf( 'explore.exe' ) != -1) {
            theBrowser = prevBrowsers[i];
            break;
        }
    }
    dw.browseDocument( 'http://www.hotwired.com/';theBrowser ) ;
}
```

※ `dreamweaver.getBrowserList()`

适用版本 Dreamweaver 3.0 以上。

描述 获取 Preview in Browser 子菜单上的所有浏览器列表。

参数 无。

返回 返回一个数组，该数组的每个元素是表示列表中每个浏览器的一对字符串。

在每对字符串的第一个字符串是浏览器的名称，第二个字符串是该浏览器在用户机器上的安装位置，是用像 `file://URL` 这样的字符串表示。如果在子菜单上没有显示任何浏览器，那么此函数返回为空。

Enabler 无。

※ `dreamweaver.getExtensionEditorList()`

适用版本 Dreamweaver 3.0 以上。

描述 从 External Editors 性能配置中获取指定文件的编辑器列表。

参数 fileURL。

此参数可以是一个完整的文件定位字符串(file://URL)、文件名,或文件扩展名(包括扩展名前的点)。

返回 返回一个数组,该数组的每个元素是表示列表中每个编辑器的一对字符串。在每对字符串的第一个字符串是编辑器的名称,第二个字符串是该编辑器在用户机器上的安装位置,是用像 file://URL 这样的字符串表示。如果在性能配置中没有显示任何编辑器,那么此函数返回为空。

Enabler 无。

例子 函数 dw.getExtensionEditorList(".gif") 调用将返回包含下列字符串的一个数组:

“ Firewords 3 ”

“ file:///c:/Program Files/Macromedia/Fireworks3/Fireworks3.exe ”

说明: 这里,第二个参数显示是在本地机器上安装的 Firework3 路径,如果读者安装是选择是其他位置,那么显示的字符串也应该不同。

※ dreamweaver.getPrimaryBrowser()

适用版本 Dreamweaver 3.0 以上。

描述 获取主浏览器的路径。

参数 无。

返回 返回一个字符串,该字符串包含了用户硬盘上的主浏览器的路径,它的表达如 file://URL,但是如果没定义主浏览器,那么此函数将返回空字符串。

Enabler 无。

※ dreamweaver.getPrimaryExtensionEditor()

适用版本 Dreamweaver 3.0 以上。

描述 对指定文件,获取主编辑器的路径。

参数 fileURL。

返回 返回一个数组,该数组包含一对字符串,其中,第一个字符串是主编辑器的名称,第二个字符串是主编辑器在用户机器上的安装路径,表示方式如 file://URL。如果没有定义主编辑器,那么此函数返回一个空字符串的数组。

Enabler 无。

※ dreamweaver.getSecondaryBrowser()

适用版本 Dreamweaver 3.0 以上

描述 获取辅助浏览器的路径。

参数 无。

返回 返回一个字符串，该字符串包含了用户硬盘上的辅助浏览器的路径，它的表达如 file://URL，但是如果没有定义辅助浏览器，那么此函数将返回空字符串。

Enabler 无。

※ **dreamweaver.openWithApp()**

适用版本 Dreamweaver 3.0 以上。

描述 利用指定的应用打开指定的文件。

参数 fileURL,appURL。

- 第一个参数是将要打开的文件的的路径，格式为：file://URL。
- 第二个参数是打开指定文件的应用程序的路径，格式为：file://URL。

返回 无。

Enabler 无。

※ **dreamweaver.openWithBrowseDialog()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Select External Editor 对话框，以便允许用户选择打开指定文件的应用程序。

参数 fileURL。

返回 无。

Enabler 无。

※ **dreamweaver.openWithExternalTextEditor()**

适用版本 Dreamweaver 3.0 以上。

描述 在 External Editors 性能中指定的外部文件编辑器中打开当前文档。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.openWithImageEditor()**

适用版本 Dreamweaver 3.0 以上。

描述 用指定的图像编辑器打开指定的文件。



如果指定 Fireworks 为图像编辑器，那么此函数将激活一个特殊的 Fireworks 集成机制，将信息返回到活动文档。如果没有活动文档，那么应该永远都不要从 Site 窗口调用此函数。

参数 fileURL, appURL。

- 第一个参数是将要打开的文件的的路径，格式为：file://URL。
- 第二个参数是打开指定文件的应用程序的路径，格式为：file://URL。

返回 无。

Enabler 无。

17.1.9 文件处理函数

文件处理函数主要处理文档的创建、打开和保存，以及将 CCS 风格表单输出到外部文件。它们完成的任务如：浏览文件或文件夹、基于模板创建文件、关闭文档，以及获取最近打开的文件的信息。

※ dreamweaver.browseForFileURL()

适用版本 Dreamweaver 1.0 中首建，在 2.0 和 3.0 中进一步改进。

描述 打开指定类型的对话框，并且该对话框的标题条上带有指定标签。

参数 openSelectOrSave, {titleBarLabel}, {bShowPreviewPane}, {bSupressSiteRoot Warnings}。

- 第一个参数指示对话框的类型：open、select 或 save。
- 第二个参数是在 Dreamweaver 2.0 中加入的，它是应该在对话框标题条上显示的标签。如果此参数没有指定，那么 DW UltraDev 将利用操作系统提供的缺少标签。
- 第三个参数，也是在 Dreamweaver 2.0 中加入的，它是一个布尔值，用于指示是否在对话框中显示图像预览窗格。如果此参数为 TRUE，则对话框为图像文件过滤；如果没有指定，它的缺少值为 FALSE。
- 第四个参数是在 Dreamweaver 3.0 中加入的，是一个布尔值，用于检测所选文件是否在站点之外。如果此参数被忽略，它的缺省值为 FALSE。

返回 返回一个包含文件名称的字符串，格式为：file://URL。

Enabler 无。

※ dreamweaver.browseForFolderURL()

适用版本 Dreamweaver 3.0 以上。

描述 打开 Choose Folder 对话框，在该对话框的标题条上带有指定标签。

参数 {titleBarLabel}, {directoryToStartIn}。

- 第一个参数是应该显示在对话框标题条上的标签。如果没有指定，此参数缺省为“ Choose Folder ”。
- 第二个参数是在对话框中显示的开始路径，格式为：file://URL。

返回 返回一个包含文件夹名称的字符串，格式为 file://URL。

Enabler 无。

例子 下列代码将返回一个文件的 URL 值：

```
return dw.browseForFolderURL( 'Select a Folder', dw.getSiteRoot() );
```

※ **dreamweaver.closeDocument()**

适用版本 Dreamweaver 3.0 以上。

描述 关闭指定的文档。

参数 documentObject。

此参数是位于一个文档的 DOM 树的根处的对象，值由函数 dreamweaver.getDocumentDOM() 返回。如果 documentObject 指向的是一个活动文档，那么直到调用此函数的脚本结束执行时，此 Document 窗口才会关闭。

返回 无。

Enabler 无。

※ **dreamweaver.createDocument()**

适用版本 Dreamweaver 2.0 以上。

描述 在同一个窗口或新窗口中打开一个新的文档，并且此新文档就成为活动文档。



此函数只能从 menus.xml 或一个命令或属性检查器中调用。如果一个动作或对象试图调用此函数，DW UltraDev 将显示一个错误信息。

参数 bOpenInSameWindow。

此参数是一个布尔值，用于指示是否在当前窗口中打开新文档。如果 bOpenInSameWindow 为 FALSE，或者没有指定，或者是在 Macintosh 中调用此函数，那么新文档将在另一个单独的窗口的中打开。

返回 返回新建文档的文档对象，此值与 dreamweaver.getDocumentDOM() 函数的返回值相同。

Enabler 无。

※ **dreamweaver.exportCSS()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Export Styles as CSS File 对话框。

参数 无。

返回 无。

Enabler dreamweaver.canExportCSS()。

※ **dreamweaver.exportEditableRegionsAsXML()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Export Editable Regions as XML 对话框。

参数 无。

返回 无。

Enabler 无。

✧ **dreamweaver.getRecentFileList()**

适用版本 Dreamweaver 3.0 以上。

描述 获取 File 菜单底部列出的最近访问的所有文件的一个列表。

参数 无。

返回 返回一个字符串数组，数组元素分别代表的是最近访问的文件的的路径，格式为：file://URLs。如果没有最近访问的文件，那么函数返回为空。

Enabler 无。

✧ **dreamweaver.importXMLIntoTemplate()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Import XML 对话框。

参数 无。

返回 无。

Enabler 无。

✧ **dreamweaver.newFromTemplate()**

适用版本 Dreamweaver 3.0 以上。

描述 从指定模板中创建一个新文档。如果参数值没有指定，那么将打开 Select Template 对话框。

参数 {templateURL}。

此参数是一个模板在当前站点上的路径，格式为：file://URL。

返回 无。

Enabler 无。

✧ **dreamweaver.openDocument()**

适用版本 Dreamweaver 3.0 以上。

描述 在一个新的 DW UltraDev 窗口中打开一个文档以便对其进行编辑，并将当前焦点交给此文档。对用户来说，相当于选择了“ File / Open ”菜单，然后在对话框中选择一个文件。如果指定的文件已经打开，那么包含此文档的窗口将出现在屏幕的最前面。包含指定文件的窗口变成当前选定的文档。对 Dreamweaver 2.0 版本，如果 check in/check out 选项是激活的，那么文件在打开之前必须获得授权 (checked out)。对 Dreamweaver 3.0 版本，我们必须利用 dreamweaver.openDocumentFromSite()函数获得此动作。

注意，此函数不能从动作或对象文件中调用，否则将导致一个错误。

参数 fileName。

此参数是正要打开的文件名，格式为：file://URL。如果 URL 是相对路径，那么它是相对于包含调用此函数的脚本的文件的的路径。

返回 返回指定文件的文档对象。它与 dreamweaver.getDocumentDOM()函数返回的值相同。

Enabler 无。

※ **dreamweaver.openDocumentFromSite()**

适用版本 Dreamweaver 3.0 以上。

描述 在一个新的 DW UltraDev 窗口中打开一个文档以便对其进行编辑，并将当前焦点交给此文档。对用户来说，相当于双击 Sit 窗口上的一个文件。如果指定的文件已经打开，那么包含此文档的窗口将出现在屏幕的最前面。包含指定文件的窗口变成当前选定的文档。

注意，此函数不能从动作或对象文件中调用，否则将导致一个错误。

参数 fileName。

此参数是将要打开的文件名，格式为：file://URL。如果 URL 是相对路径，那么它是相对于包含调用此函数的脚本的文件的文件的路径。

返回 返回指定文件的文档对象。它与 dreamweaver.getDocumentDOM()函数返回的值相同。

Enabler 无。

※ **dreamweaver.openInFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Open In Frame 对话框。当用户选择了一个文档，它将在活动框架中被打开。

参数 无。

返回 无。

Enabler dreamweaver.canOpenInFrame()。

※ **dreamweaver.releaseDocument()**

适用版本 Dreamweaver 2.0 以上。

描述 从内存中显式地释放以前引用的一个文档。

当包含调用的脚本执行完毕时，由下列函数引用的文档将自动释放：

- dreamweaver.getObjectTags()
- dreamweaver.getObjectRefs()
- dreamweaver.getDocumentPath()
- dreamweaver.getDocumentDOM()

如果脚本打开许多文档，那么我们必须结束脚本运行之前，利用此函数显式地释放文档，以免出现内存不足的错误。



此函数仅与由 URL 引用的文档有关，并且此文档当前没有在框架或 Document 窗口中打开，另外它不能是扩展文件。扩展文件是在启动时装入内存的，并且直到

退出 DW UltraDev 时,才释放。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象,由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler 无。

※ dreamweaver.revertDocument()

适用版本 Dreamweaver 3.0 以上。

描述 将指定的文档回复到以前保存的版本。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象,由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler dreamweaver.canRevertDocument()。

※ dreamweaver.saveAll()

适用版本 Dreamweaver 3.0 以上。

描述 保存所有打开的文档,为那些以前没有保存过的文件打开 Save As 对话框。

参数 无。

返回 无。

Enabler dreamweaver.canSaveAll()。

※ dreamweaver.saveDocument()

适用版本 Dreamweaver 2.0 以上。

描述 在本地驱动器上保存指定的文件。



在 Dreamweaver 2.0 中,如果此文件是只读的,那么 Dreamweaver 将试图取得权限 (check out)。如果经过试验之后,此文档仍是只读的,或者不能创建此文档,那么将显示一个错误信息。

参数 documentObject,{fileURL}。

- 第一个参数是位于一个文档 DOM 树的树根处的对象,由 dreamweaver.getDocumentDOM()函数返回。
- 第二个参数是一个表示本地驱动器路径的 URL。如果 URL 是相对路径的,那么它是相对扩展文件的。在 Dreamweaver 2.0 中,此参数是必须的。在 Dreamweaver 3.0 版本中,如果忽略 fileURL 参数,那么如果此文档以前曾经保存过,那么它将

保存它当前的位置；否则，将显示 Save 对话框。

返回 返回一个布尔值，用于指示成功(TRUE) 或失败(FALSE)。

Enabler dreamweaver.canSaveDocumnet()。

※ **dreamweaver.saveDocumentAs()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Save As 对话框。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象，由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler 无。

※ **dreamweaver.saveDocumentAsTemplate()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Save As Template 对话框。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象，由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler dreamweaver.canSaveDocumentAsTemplate()。

※ **dreamweaver.saveFrameset()**

适用版本 Dreamweaver 3.0 以上。

描述 保存指定的框架集，或者如果框架集以前还没有保存过，那么打开 Save As 对话框。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象，由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler dreamweaver.canSaveFrameset()。

※ **dreamweaver.saveFramesetAs()**

适用版本 Dreamweaver 3.0 以上。

描述 为包含指定 DOM 的指定框架集文件打开 Save As 对话框。

参数 documentObject。

此参数是位于一个文档 DOM 树的树根处的对象，由 dreamweaver.getDocumentDOM()

函数返回。

返回 无。

Enabler dreamweaver.canSaveFramesetAs()。

17.1.10 Find/Replace 函数

Find/Replace 函数处理的是查找和替换操作。它们涵盖了一些基本的查找和替换功能，例如，查找一个查寻模式的下一个实例，以及不要求用户干预的情况下，就可以完成完成替换操作。

※ **dreamweaver.findNext()**

适用版本 Dreamweaver 3.0 以上。

描述 按照指定查寻字符串查找下一个实例，并选定文档中的实例。查寻字符串是由 dreamweaver.setUpFind()函数,或 dreamweaver.setUpComplexFind()函数,或用户在 Find 对话框中指定的。

参数 无。

返回 无。

Enabler dreamweaver.canFindNext()。

※ **dreamweaver.replace()**

适用版本 Dreamweaver 3.0 以上。

描述 首先验证当前的选定是否与指定的条件相匹配，如果匹配，就用查询中指定的内容代替当前选定。匹配条件是由 dreamweaver.setUpFindReplace() 函数，或 dreamweaver.setUpComplexFindReplace()函数，或用户在 Replace 对话框中指定的。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.replaceAll()**

适用版本 Dreamweaver 3.0 以上。

描述 用查询中指定的内容替换当前文档中与指定条件相匹配的所有内容。匹配条件是由 dreamweaver.setUpFindReplace()函数,或 dreamweaver.setUpComplexFindReplace()函数,或用户在 Replace 对话框中指定的。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.setUpComplexFind()**

适用版本 Dreamweaver 3.0 以上。

描述 通过装载指定的 XML 查询为一个高级文本或标记查找作准备。

参数 xmlQueryString。

此参数是一个以<dwquery>开始,以</dwquery>结束的一个 XML 编码字符串。为了获得正确格式的字符串,在 Find 对话框中建立查寻,单击 Save Query 按钮,在一个文本编辑器中打开查询文件,并拷贝从<dwquery>标记开始直到</dwquery>标记之间的所有内容,包括这两个标记。

返回 无。

Enabler 无。

例子 下面的第一行代码建立一人查寻标记,并指定查寻的范围应该是当前文档。第二行代码执行查寻操作。

```
dw.setUpComplexFind( '<dwquery><queryparams matchcase="false" ignorewhitespace="true"
useregexp="false"/><find><qtag qname="a"><qattribute qname="href" qcompare="="
qvalue="#"></qattribute><qattribute qname="onMouseOut" qcompare="=" qvalue=""
qnegate="true"></qattribute></qtag></find></dwquery>' );
dw.findNext();
```

※ dreamweaver.setUpComplexFindReplace()

适用版本 Dreamweaver 3.0 以上。

描述 通过装载指定的 XML 查询为一个高级文本或标记查找作准备。

参数 xmlQueryString。

此参数是一个以<dwquery>开始,以</dwquery>结束的一个 XML 编码字符串。为了获得正确格式的字符串,在 Find 对话框中建立查寻,单击 Save Query 按钮,在一个文本编辑器中打开查询文件,并拷贝从<dwquery>标记开始直到</dwquery>标记之间的所有内容,包括这两个标记。

返回 无。

Enabler 无。

例子 下面的第一行代码建立一人查寻标记,并指定查寻的范围应该是当前文档。第二行代码执行查找并替换操作。

```
dw.setUpComplexFindReplace( '<dwquery><queryparams matchcase="false"
ignorewhitespace="true" useregexp="false"/><find><qtag qname="a"><qattribute qname="href"
qcompare="=" qvalue="#"></qattribute><qattribute qname="onMouseOut" qcompare="="
qvalue="" qnegate="true"></qattribute></qtag></find><replace action="setAttribute"
param1="onMouseOut" param2="this.style.color=#000000;this.style.fontWeight=normal"/>
</dwquery>' );
dw.replaceAll();
```

※ dreamweaver.setUpFind()

适用版本 Dreamweaver 3.0 以上。

描述 通过定义查询参数，为接下来的 dw.findNext()操作准备一个文本或 HTML 查询源对象。

参数 searchObject。

此参数是一个对象，可以为此对象定义下列特性：

- searchString：这是查询的文本。
- searchSource：这是一个布尔值，用于指示是否查询 HTML 源代码。
- {matchCase}：也是一个布尔值，用于指示查询是否区分大小写，如果此特性没有显式地指定，它的缺省值为 FALSE。
- {ignoreWhitespace}：一个布尔值，指示查询中是否忽略空格。如果 useRegularExpression 特性是 TRUE，那么此特性缺省为 FALSE；相反，如果 useRegularExpression 特性是 FALSE，那么此特性缺省为 TRUE。
- {useRegularExpression}：一个布尔值，用于指示 searchString 是否使用常规表达式。如果此特性没有指定，它的缺省值为 FALSE。

返回 无。

Enabler 无。

例子 下面的代码以三种方法展示了创建一个 searchObject 对象的过程：

```
var searchParams;  
  
searchParams.searchString = bgcolor="#FFCCFF";  
searchParams.searchSource = true;  
dw.setUpFind( searchParams) ;  
  
var searchParams = {searchString: bgcolor="#FFCCFF", searchSource: true};  
dw.setUpFind( searchParams) ;  
  
dw.setUpFind( {searchString: bgcolor="#FFCCFF", searchSource: true}) ;
```

※ **dreamweaver.setUpFindReplace()**

适用版本 Dreamweaver 3.0 以上。

描述 通过定义查询参数和范围，为接下来的 dw.replace()或 dw.replaceAll()操作准备一个文本或 HTML 查询源对象。

参数 searchObject。

此参数是一个对象，可以为此对象定义下列特性：

- searchString：这是查询的文本。
- replaceStrng：这是用来替换选定内容的文本。
- searchSource：这是一个布尔值，用于指示是否查询 HTML 源代码。
- {matchCase}：也是一个布尔值，用于指示查询是否区分大小写，如果此特性没有

显式地指定，它的缺省值为 FALSE。

- {ignoreWhitespace}：一个布尔值，指示查询中是否忽略空格。如果 useRegularExpression 特性是 TRUE，那么此特性缺省为 FALSE；相反，如果 useRegularExpression 特性是 FALSE，那么此特性缺省为 TRUE；
- {useRegularExpression}：一个布尔值，用于指示 searchString 是否使用常规表达式。如果此特性没有指定，它的缺省值为 FALSE。

返回 无。

Enabler 无。

例子 下面的代码以三种方法展示了创建一个 searchObject 对象的过程：

```
var searchParams;
searchParams.searchString = bgcolor="#FFCCFF";
searchParams.replaceString = bgcolor="#CCFFCC";
searchParams.searchSource = true;
dw.setUpFindReplace( searchParams );

var searchParams = {searchString: bgcolor="#FFCCFF", replaceString: bgcolor="#CCFFCC",
searchSource: true};
dw.setUpFindReplace( searchParams );

dw.setUpFindReplace( {searchString: bgcolor="#FFCCFF", replaceString: bgcolor="#CCFFCC",
searchSource: true} );
```

※ dreamweaver.showFindDialog()

适用版本 Dreamweaver 3.0 以上。

描述 打开 Find 对话框。

参数 无。

返回 无。

Enabler dreamweaver.canShowFindDialog()。

※ dreamweaver.showFindReplaceDialog()

适用版本 Dreamweaver 3.0 以上。

描述 打开 Replace 对话框。

参数 无。

返回 无。

Enabler dreamweaver.canShowFindDialog()。

17.1.11 Frame 和 frameset 函数

框架和框架集函数仅覆盖了两个任务：获取一个框架集中的框架名称，以及将一个

框架分成两个。

※ **dom.getFrameNames()**

适用版本 Dreamweaver 3.0 以上。

描述 获取框架集中所有命名框架的一个列表。

参数 无。

返回 返回一个字符串数组，其中每个数组元素是在当前框架集中每一个框架的名称。没有命名的框架将被忽略。如果在框架集中命名的框架一个都没有，那么将返回一个空数组。

Enabler 无。

例子 对一个包含四个框架的文档，其中有两个是命名的，那么调用 `dw.getDocumentDOM().getFrameNames()` 将返回一个包含下列字符串的数组：

“ navframe ”

“ main content ”

※ **dom.splitFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 将指定框架按水平或竖直方向分隔。

参数 `splitDirection`。

此参数必须是下列值之一：“ up ”、“ down ”、“ left ”或“ right ”。

返回 无。

Enabler `dom.canSplitFrame()`。

17.1.12 通用编辑函数

通用编辑函数处理发生在 Document 窗口内的一般编辑任务。这些函数可以完成文件、HTML 和对象的插入；应用、改变和移走字体和字符修饰特性；修改标记和属性，等等诸如此类的其他编辑功能。

※ **dom.applyCharacterMarkup()**

适用版本 Dreamweaver 3.0 以上。

描述 将指定的字符修饰特性应用到已经选定的内容上。如果当前选定是一个插入点，那么指定的字符修饰特性将对随后键入的文本起作用。

参数 `tagName`。

此参数是与字符修饰特性相关的标记名称。它必须是下列字符串之一：

“ b ”、“ cite ”、“ code ”、“ dfn ”、“ em ”、“ i ”、“ kbd ”、“ samp ”、“ s ”、“ strong ”、“ tt ”、“ u ”或“ var ”。

返回 无。

Enabler 无。

✳ **dom.applyFontMarkup()**

适用版本 Dreamweaver 3.0 以上。

描述 将 FONT 标记和指定的属性以及值应用到当前选定的内容上。

参数 attribute, value。

- 第一个参数必须是下列字符串之一：“face”、“size”或“color”；
- 第二个参数应该分配给 attribute 的值。例如，“Arial,Helvetica,sans-serif”、“5”或“#FF0000”。

返回 无。

Enabler 无。

✳ **dom.deleteSelection()**

适用版本 Dreamweaver 3.0 以上。

描述 删除文档中当前的选定内容。

参数 无。

返回 无。

Enabler 无。

✳ **dom.editAttribute()**

适用版本 Dreamweaver 3.0 以上。

描述 显示适当的接口以便编辑指定的属性。在部分情况，此函数都会显示一个对话框。此函数仅对活动文档有效。

参数 attribute。

返回 无。

Enabler 无。

✳ **dom.exitBlock()**

适用版本 Dreamweaver 3.0 以上。

描述 退出当前段落或文件头部块，将光标放置到所有块元素之外。

参数 无。

返回 无。

Enabler 无。

✳ **dom.getFontMarkup()**

适用版本 Dreamweaver 3.0 以上。

描述 为当前选定获取 FONT 标记指定属性的值。

参数 attribute。

此参数必须是：“face”、“size”或“color”。

返回 返回一个包含指定属性值的字符串，或者当此属性值没有设置时，将返回一

个空字符串。

Enabler 无。

※ **dom.getLinkHref()**

适用版本 Dreamweaver 3.0 以上。

描述 获取包围当前选定的链接文件。此函数相当于从当前节点的父节点开始遍历，直到它的祖父节点，直到发现一个链接，然后调用链接的 `getAttribute( 'HREF' )`。

参数 无。

返回 返回一个包含链接文件的字符串，格式为：`file://URL`。

Enabler 无。

※ **dom.getLinkTarget()**

适用版本 Dreamweaver 3.0 以上。

描述 获取包围当前选定的链接的目标文件。此函数相当于从当前节点的父节点开始遍历，直到它的祖父节点，直到发现一个链接，然后调用链接的 `getAttribute( 'TARGET' )`。

参数 无。

返回 返回一个包含链接的 `TARGET` 属性值的字符串，或者如果没有指定链接目标，那么此函数将返回一个空字符串。

Enabler 无

※ **dom.getListTag()**

适用版本 Dreamweaver 3.0 以上。

描述 获取选定列表的风格。

参数 无。

返回 返回一个字符串，该字符串中包含了与列表(“`ul`”、“`ol`”或“`dl`”)有关的标记，或者如果没有标记与列表相关，那么此函数将返回一个空字符串。另外，返回值总是小写形式。

Enabler 无。

※ **dom.getTextAlignment()**

适用版本 Dreamweaver 3.0 以上。

描述 获取包含当前选定的文本块的对齐方式。

参数 无。

返回 返回一个字符串，该字符串中包含了与选定块相关的 `ALIGN` 属性的值，或者如果 `ALIGN` 属性值没有被指定，那么此函数将返回一个空字符串。另外，返回值总是小写形式。

Enabler 无。

※ **dom.getTextFormat()**

适用版本 Dreamweaver 3.0 以上。

描述 获取当前选定文本的块格式。

参数 无。

返回 返回一个字符串，该字符串中包含了与选定文本相关的块标记（例如，“p”、“h1”、“pre”，等等），或者如果没有指定与此文本相关的块标记，那么此函数将返回一个空字符串。另外，返回值总是小写形式。

Enabler 无。

※ dom.hasCharacterMarkup()

适用版本 Dreamweaver 3.0 以上。

描述 检查当前选定是否已经具有了指定的字符修饰特性。

参数 markupTagName。

此参数是需要检查的标记的名称。它必须是下列字符串之一：“b”、“cite”、“code”、“dfn”、“em”、“i”、“kbd”、“samp”、“s”、“strong”、“tt”、“u”或“var”。

返回 返回一个布尔值，用于指示当前整个选定是否具有指定的字符修饰特性。如果当前选定只有部分具有指定特性，那么此函数返回 FALSE。

Enabler 无。

※ dom.indent()

适用版本 Dreamweaver 3.0 以上。

描述 利用 BLOCKQUOTE 标记缩进当前选定。如果当前选定位于一个列表内部，那么此函数将套叠当前列表内部其他相同类型的列表。

参数 无。

返回 无。

Enabler 无。

※ dom.insertHTML()

适用版本 Dreamweaver 3.0 以上。

描述 将 HTML 内容插入到文档的当前插入点处。

参数 contentToInsert, {bReplaceCurrentSelection}。

- 第一个参数是将插入的内容。
- 第二个参数是一个布尔值，用于指示是否用新内容代替当前选定的内容。如果此布尔值为 False，那么新内容将在当前选定内容之后插入。

返回 无。

Enabler 无。

例子 下面的代码将130插入到当前文档内：

```
var theDOM = dw.getDocumentDOM();
theDOM.insertHTML( '<b>130</b>' );
```

Document 窗口的显示将如图 17-5 所示：

图 17-5

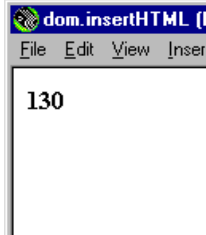
调用

insertHTML

函数之后的

Document

窗口



※ **dom.insertObject()**

适用版本 Dreamweaver 3.0 以上。

描述 插入指定对象，如果必要的话，为用户提示参数信息。

参数 objectName。

此参数是在 Configuration/Objects 文件夹下的一个对象的名称。

返回 无。

Enabler 无。

例子 调用 dreamweaver.getDocumentDOM().insertObject('Button') 之后，将在当前文档的当前插入点或在当前插入点之后插入一个 form 按钮。



尽管对象文件可以在另一个文件夹中保存，但是有一点非常重要：它们的文件名必须是惟一的。如果一个名为 Button.htm 的文件已经存在于 Forms 文件夹中，同时在 MyObjects 文件夹中也有一个同名的文件，那么此时 DW UltraDev 就不能区分应该使用它们之中的哪一个。

※ **dom.insertText()**

适用版本 Dreamweaver 3.0 以上。

描述 将文本内容插入到文档当前插入点位置处。

参数 contentToInsert, {bReplaceCurrentSelection}。

- 第一个参数是将要插入的文本内容。
- 第二个参数是一个布尔值，用于指示是否用新内容覆盖当前选定内容。如果此值为 False，那么新内容将在当前选定内容之后插入。

返回 无。

Enabler 无。

例子 下面的代码将把<lt;b>130插入到当前文档内：

```
var theDOM = dw.getDocumentDOM();  
theDOM.insertText( '<b>130</b>' );
```


Document 窗口的显示将如图 17-6 所示：

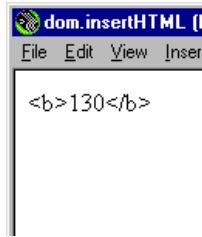


图 17-6
调用
insertText
函数之后的
Document
窗口

※ dom.newBlock()

适用版本 Dreamweaver 3.0 以上。

描述 用包含当前选定内容的块的标记和属性创建一个新块，或者如果当前光标在所有块之外，那么此函数将创建一个新的文本段。

参数 无。

返回 无。

Enabler 无。

例子 如果当前选定是在一个具有中心对齐格式的段内，那么调用 dreamweaver.getDocumentDOM().newBlock()函数之后，将在当前段后插入：<p align="center">。

※ dom.outdent()

适用版本 Dreamweaver 3.0 以上。

描述 突出选定内容。

参数 无。

返回 无。

Enabler 无。

※ dom.removeCharacterMarkup()

适用版本 Dreamweaver 3.0 以上。

描述 从选定内容上移走指定字符修饰类型。

参数 tagName。

此参数是与字符修饰属性相关的标记名称。它必须是下列字符串之一：“b”、“cite”、“code”、“dfn”、“em”、“i”、“kbd”、“samp”、“s”、“strong”、“tt”、“u”或“var”。

返回 无。

Enabler 无。

※ dom.removeFontMarkup()

适用版本 Dreamweaver 3.0 以上。

描述 从一个 FONT 标记中移走指定属性及其值。如果移走属性之后，此 FONT 标

记只剩余属性了，那么此标记也将被移走。

参数 attribute。

此参数必须是下列字符串之一：“face”、“size”或“color”。

返回 无。

Enabler 无。

※ dom.removeLink()

适用版本 Dreamweaver 3.0 以上。

描述 从选定上移走超链接。

参数 无。

返回 无。

Enabler 无。

※ dom.resizeSelection()

适用版本 Dreamweaver 3.0 以上。

描述 将选定内容改为指定的大小。为了改变一个层或热点的大小，请利用 dom.resizeSelectionBy()函数。

参数 newWidth, newHeight。

返回 无。

Enabler 无。

※ dom.setAttributeWithErrorChecking()

适用版本 Dreamweaver 3.0 以上。

描述 用指定值设置选定内容的指定属性，并且如果此值是一个错误的类型，或者它溢出了规定的范围，此函数将为用户显示提示信息。此函数仅对活动文档有效。

参数 attribute, value。

返回 无。

Enabler 无。

※ dom.setLinkHref()

适用版本 Dreamweaver 3.0 以上。

描述 为选定指定一个超链接，或者改变包围当前选定的链接值。

参数 linkHREF。

此参数选定内容将链接的文件 URL(相对文档的路径、相对根的路径，或是一个绝对路径)。如果没有指定此参数，那么将弹出 SelectHTML File 对话框。

返回 无。

Enabler dom.canSetLinkHref()。

※ dom.setLinkTarget()

适用版本 Dreamweaver 3.0 以上。

描述 设置包围当前选定的链接目标,此函数相当于从当前节点的父节点开始遍历,一直到它的祖父节点,直到发现一个链接,然后调用链接的 `setAttribute( 'TARGET' )`。

参数 {linkTarget}。

此参数是一个字符串,表示一个框架或窗口名称,或者保留目标之一(“_self”、“_parent”、“_top”或“_blank”)。如果没有指定此参数,那么将弹出 Set Target 对话框。

返回 无。

Enabler 无。

✳ `dom.setListBoxKind()`

适用版本 Dreamweaver 3.0 以上。

描述 改变选定的 SELECT 菜单的种类。

参数 kind。

此参数要么是“menu”,要么是“List box”。

返回 无。

Enabler 无。

✳ `dom.showListPropertiesDialog()`

适用版本 Dreamweaver 3.0 以上。

描述 打开 List Properties 对话框。

参数 无。

返回 无。

Enabler `dom.canShowListPropertiesDialog()`。

✳ `dom.setListTag()`

适用版本 Dreamweaver 3.0 以上。

描述 设置选定列表的风格。

参数 listTag。

此参数是与列表相关的标记。它必须是“ol”、“ul”或“dl”,或者是一个空字符串。

返回 无。

Enabler 无。

✳ `dom.setTextAlignment()`

适用版本 Dreamweaver 3.0 以上。

描述 设置包含选定的块的 ALIGN 属性值为指定值。

参数 alignValue。

此参数必须是“left”、“center”或“right”。

返回 无。

Enabler 无。

✳ `dom.setTextFieldKind()`

适用版本 Dreamweaver 3.0 以上。

描述 设置选定文本域的格式。

参数 fieldType

此参数必须是 “input”、“textarea” 或 “password”。

返回 无。

Enabler 无。

※ **dom.showFontColorDialog()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Color Picker 对话框。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.deleteSelection()**

适用版本 Dreamweaver 3.0 以上。

描述 删除活动文档中或 Site 窗口中的选定，在 Macintosh 平台上，将删除对话框中或浮动调色板上的拥有焦点的编辑域。

参数 无。

返回 无。

Enabler dreamweaver.canDeleteSelection()。

※ **dreamweaver.editFontList()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Edit Font List 对话框。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.getFontList()**

适用版本 Dreamweaver 3.0 以上。

描述 获取文本属性检查器以及 Style Definition 对话框中显示的所有字体组的一个列表。

参数 无。

返回 返回表示字体列表中每一个元素的字符串数组。

Enabler 无。

例子 对 DW UltraDev 的缺省安装，调用 dw.getFontList()将返回包含下列元素的一个数组：

“ Arial,Helvetica,sans-serif ”

“ Times New Roman,Times,serif ”
 “ Courier New,Courier,mono ”
 “ Georgia,Times New Roman,Times,serif ”
 “ Verdana,Arial,Helvetica,sans-serif ”

※ **dreamweaver.getKeyState()**

适用版本 Dreamweaver 3.0 以上。

描述 判断是否按下指定的修饰键。

参数 key。

此参数必须是下列值之一：“ Cmd ”、“ Ctrl ”、“ Alt ”或“ Shift ”。Windows 平台上，“ Cmd ”和“ Ctrl ”都是指 Control 键；Macintosh 平台上，“ Alt ”指的是 Option 键。

返回 返回一个布尔值，用于指示是否按下了指定的键。

Enabler 无。

例子 下面代码的功能是在执行下一操作之前，检查是否按下了 Shift 和 Control 键（Windows），或者 Shift 和 Command 键（Macintosh）键：

```
if( dw.getKeyState( "Shift") && dw.getKeyState( "Cmd")) {  
    // execute code  
}
```

17.1.13 全局应用函数

全局应用函数作用于整个应用范围。它们的提供诸如终止和访问特性的功能。

※ **dreamweaver.getShowDialogsOnInsert()**

适用版本 Dreamweaver 3.0 以上。

描述 检查在 General 特性中的 Show Dialog When Inserting Objects 选项是否设置。

参数 无。

返回 返回一个布尔值，用于指示是否打开了此选项。

Enabler 无。

※ **dreamweaver.quitApplication()**

适用版本 Dreamweaver 3.0 以上。

描述 当调用此函数的脚本执行完毕之后，退出 DW UltraDev。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.showAboutBox()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 About 对话框。

参数 无。

返回 无。

Enabler 无。

※ dreamweaver.showDynamicDataDialog()

适用版本 Dreamweaver UltraDev 4 以上。

描述 显示 Dynamic Data 或 Dynamic Text 对话框，然后等待用户关闭对话框。如果用户单击 OK 按钮，则此函数返回一个将要插入用户文档内的字符串（此字符串是从数据源 API 函数 generateDynamicDataRef() 返回的，并传递到数据格式化 API 函数 formatDynamicDataRef()；formatDynamicDataRef() 函数返回值就是 showDynamicDataDialog() 函数的返回值）。

参数 source, title。

- source 是包含 HTML 源代码的一个字符串，表示动态数据对象，它与先前调用此函数的返回值相同。此函数利用 source 的内容初始化所有的对话框控制，当用户单击 OK 按钮创建此字符串时，这些控制就准确地显示了。

UltraDev 将此字符串传递给 inspectDynamicDataRef() 以决定字符串是否与树中的某一节点相匹配。如果字符串与一节点匹配，那么当对话框显示时，该节点将处于选中状态。用户也可以传递一个空字符串，那就不会初始化对话框。例如，当对话框用于创建一个新项时，对话框就不会初始化。

- title 是包含对话框标题条上显示文本的一个字符串。此参数是可选的。如果没有指定此参数，UltraDev 将在标题条上显示 Dynamic Data。

返回 如果用户单击 OK 按钮，此函数就返回表示动态数据对象的一个字符串。

※ dreamweaver.showPreferencesDialog()

适用版本 Dreamweaver 3.0 以上。

描述 打开 Preferences 对话框。

参数 {whichTab}。

此参数必须是下列字符串之一：“general”、“external editors”、“floaters”、“fonts”、“highlighting”、“html colors”、“html format”、“html rewriting”、“invisible elements”、“layers”、“browsers”、“quick tag editor”、“site ftp”、“status bar”、“css styles”和“translation”。如果 Dreamweaver 不能正确地识别此参数，那么对话框将以上次活动窗格的状态显示。

返回 无。

Enabler 无。

17.1.14 全局文档函数

全局文档函数是作用于整个文档的。它们检查拼写、检查目标浏览器、设置页面属性，以及为文档内的元素决定正确的对象引用。

※ **dom.checkSpelling()**

适用版本 Dreamweaver 3.0 以上。

描述 检查文档内的拼写，如果必要的话，打开 Check Spelling 对话框，并且当检查完成后通知用户。

参数 无。

返回 无。

Enabler 无。

※ **dom.checkTargetBrowsers()**

适用版本 Dreamweaver 3.0 以上。

描述 运行一个目标浏览器以检查文档。为了运行一个目标检查器以检查一个文件夹或文件集，请使用 site.checkTargetBrowsers()函数。

参数 无。

返回 无。

Enabler 无。

※ **dom.showPagePropertiesDialog()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Page Properties 对话框。

参数 无。

返回 无。

Enabler 无。

※ **dreamweaver.getElementRef()**

适用版本 Dreamweaver 2.0 以上。

描述 获取 DOM 树中一个指定标记对象的 Netscape 或 IE 对象引用。

参数 NSorIE, tagObject。

- 第一个参数要么是“NS 4.0”或“IE 4.0”。Navigator 4.0 和 Internet Explorer 4.0 中 DOM 和嵌套引用规则是不同的。所以此参数指定特定的浏览器以返回一个有效的引用。

- 第二个参数是在 DOM 树内的一个标记对象。

返回 返回表示对象的一个有效的 JavaScript 引用的一个字符串，例如，document.layers['myLayer']。

- Dreamweaver 为下列标记返回正确的 Internet Explorer 浏览器引用：A、AREA、APPLET、EMBED、DIV、SPAN、INPUT、SELECT、OPTION、TEXTAREA、OBJECT 和 IMG 标记。
- Dreamweaver 为下列标记返回正确的 Navigator 浏览器引用：A、AREA、APPLET、EMBED、LAYER、ILAYER、SELECT、OPTION、TEXTAREA、OBJECT 和 IMG 标记，以及绝对定位的 DIV 和 SPAN 标记。对那此不是绝对定位的 DIV 和 SPAN 标记，Dreamweaver 返回“ cannot reference <tag> ”。
- Dreamweaver 不为没有命名的对象返回引用。如果一个对象没有 NAME 或 ID 属性，此时 Dreamweaver 将返回“ unnamed <tag> ”。如果浏览器不支持利用名称的引用，那么 Dreamweaver 将通过索引引用对象，例如，document.myform.applets[3]。
- Dreamweaver 为未命名表格和层内的命名对象返回引用，例如，document.forms[2].myCheckbox。

Enabler 无。

17.1.15 History 函数

History 函数处理 undo、redo、recode，以及 History 调色板中的历史操作步骤。一个操作步骤就是对文档或文档内一个选定进行的任意可重复的改变。dreamweaver.historyPalette 对象的方法控制或作用于 History 调色板内的选定，而不是当前文档内的选定。

✧ dom.redo()

适用版本 Dreamweaver 3.0 以上。

描述 恢复文档内刚刚取消的操作。

参数 无。

返回 无。

Enabler dom.canRedo()。

✧ dom.undo()

适用版本 Dreamweaver 3.0 以上。

描述 取消文档内刚刚进行的操作。

参数 无。

返回 无。

Enabler dom.canUndo()。

✧ dreamweaver.getRedoText()

适用版本 Dreamweaver 3.0 以上。

描述 如果用户选择 Edit / Redo 菜单项或按下 Ctrl+Y 键 (Windows) 或 Command+Y

键 (Machintosh), 那么将进行 redo 操作, 此函数就是获取与此 Redo 编辑操作相关的文本。

参数 无。

返回 返回包含与即将进行的 Redo 编辑操作相关的文本的一个字符串。

Enabler 无。

例子 如果用户最后一次操作是将选定内容变为粗体, 那么调用 `dw.getRedoText()` 函数将返回字符串: "Repeat Apply Bold"。

※ `dreamweaver.getUndoText()`

适用版本 Dreamweaver 3.0 以上。

描述 如果用户选择 Edit / Undo 菜单项或按下 Ctrl+Z 键 (Windows) 或 Command+Z 键 (Machintosh), 那么将进行 Undo 操作, 此函数就是获取与此 Undo 编辑操作相关的文本。

参数 无。

返回 返回包含与即将进行的 Undo 编辑操作相关的文本的一个字符串。

Enabler 无。

例子 如果用户最后一次操作是将一个 CSS 风格应用到一个选定文本, 那么调用 `dw.getUndoText()` 函数将返回字符串: "Undo Apply "。

※ `dreamweaver.playRecordedCommand()`

适用版本 Dreamweaver 3.0 以上。

描述 播放活动文档的记录的命令。

参数 无。

返回 无。

Enabler `dreamweaver.canPlayRecordCommand()`。

※ `dreamweaver.redo()`

适用版本 Dreamweaver 3.0 以上。

描述 恢复当前拥有焦点的 Document 窗口、对话框、浮动面板或 Site 窗口中刚刚 Undo 的操作。

参数 无。

返回 无。

Enabler `dreamweaver.canRedo()`。

※ `dreamweaver.startRecording()`

适用版本 Dreamweaver 3.0 以上。

描述 开始记录活动文档的操作步骤。以前记录的命令将立即被抛弃。

参数 无。

返回 无。

Enabler `dreamweaver.isRecording()` (必须返回 FALSE)。

※ **`dreamweaver.stopRecording()`**

适用版本 Dreamweaver 3.0 以上。

描述 并不提示用户就停止记录操作步骤。

参数 无。

返回 无。

Enabler `dreamweaver.isRecording()` (必须返回 TRUE)。

※ **`dreamweaver.undo()`**

适用版本 Dreamweaver 3.0 以上。

描述 取消当前拥有焦点的 Document 窗口、对话框、浮动面板或 Site 窗口中刚刚进行的操作。

参数 无。

返回 无。

Enabler `dreamweaver.canUndo()`。

※ **`dreamweaver.historyPalette.clearSteps()`**

适用版本 Dreamweaver 3.0 以上。

描述 清除 History 调色板中的所有历史步骤，并且使 Undo 和 Redo 菜单项处于 disable 状态。

参数 无。

返回 无。

Enabler 无。

※ **`dreamweaver.historyPalette.copySteps()`**

适用版本 Dreamweaver 3.0 以上。

描述 拷贝指定的历史步骤到剪贴板。如果这些指定的步骤包含一个不可重复的操作，那么 DW UltraDev 将警告用户将出现不可预料的顺序。

参数 `arrayOfIndices`。

此参数是在 History 调色板内位置索引的一个数组。

返回 无。

Enabler 无。

例子 下面的代码将拷贝 History 调色板中的前四个操作到剪贴板：

```
dw.historyPalette.copySteps ( [0,1,2,3] ) ;
```

※ **`dreamweaver.historyPalette.getSelectedSteps()`**

适用版本 Dreamweaver 3.0 以上。

描述 决定 History 调色板中当前被选中的部分。

参数 无。

返回 返回一个包含所有选定步骤的位置索引的数组。

Enabler 无。

例子 如果 History 调色板中的第二、第三、第四个操作是当前选中的步骤，就如图 17-7 所示的那样，那么调用 `dw.historyPalette.getSelectedSteps()`，将返回数组：`[1,2,3]`。

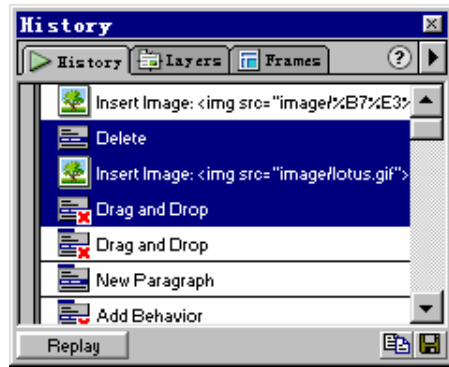


图 17-7
当前选中
步骤的
History 调
色板

※ **`dreamweaver.historyPalette.getStepCount()`**

适用版本 Dreamweaver 3.0 以上。

描述 获得 History 调色板内的操作步骤数目。

参数 无。

返回 返回表示当前在 History 调色板内列出的所有步骤数目的一个整形数。

Enabler 无。

※ **`dreamweaver.historyPalette.getStepsAsJavaScript()`**

适用版本 Dreamweaver 3.0 以上。

描述 获得与选定步骤对应的 JavaScript 语句。

参数 无。

返回 返回包含与选定步骤对应的 JavaScript 语句的一个字符串。

Enabler 无。

例子 对于图 17-10 所示的 History 调色板，下面的调用：

```
dw.historyPalette.getStepsAsJavaScript( dw.historyPalette.getSelectedSteps())
```

将返回字符串：

```
dw.historyPalette.getStepsAsJavaScript( dw.historyPalette.getSelectedSteps()) returns
"dw.getDocumentDOM().insertText( 'Hey diddle diddle, a cat and a fiddle, the cow jumped over the
moon.' );\ndw.getDocumentDOM().newBlock();\n dw.getDocumentDOM().insertHTML( '<img
src=\"../wdw99/50browsers/images/sun.gif\">'; true) ;\n":
ectedSteps())
```

※ **dreamweaver.historyPalette.getUndoState()**

适用版本 Dreamweaver 3.0 以上。

描述 获得当前 undo 状态。

参数 无。

返回 返回 History 调色板内 Undo 标记的位置。

Enabler 无。

※ **dreamweaver.historyPalette.replaySteps()**

适用版本 Dreamweaver 3.0 以上。

描述 在活动文档内重复指定的历史步骤。如果指定的步骤包含一个不可重复的动作，那么 DW UltraDev 将警告用户会出现不可预知的顺序。

参数 arrayOfIndices。

此参数是 History 调色板内所有步骤的位置索引的一个数组。

返回 返回 History 调色板内 Undo 标记的位置。

Enabler 无。

例子 调用：

```
dw.historyPalette.replaySteps( [0,2,3])
```

将进行 History 调色板内的第一、第三和第四步操作。

※ **dreamweaver.historyPalette.saveAsCommand()**

适用版本 Dreamweaver 3.0 以上。

描述 打开 Save As Command 对话框，允许用户将指定的步骤当作一个命令保存。如果指定的这些步骤中包含一个不可重复的动作，那么 DW UltraDev 将警告用户会出现不可预料的顺序。

参数 arrayOfIndices。

此参数是 History 调色板内位置索引的一个数组。

返回 无。

Enabler 无。

例子 下面的代码将把 History 调色板内的第四、第六和第八步的操作当作一个命令保存：

```
dw.historyPalette.saveAsCommand( [3,5,7]) ;
```

※ **dreamweaver.historyPalette.setSelectedSteps()**

适用版本 Dreamweaver 3.0 以上。

描述 选定 History 调色板内的指定步骤。

参数 arrayOfIndices。

此参数是 History 调色板内位置索引的一个数组。如果没有指定此参数,那么 History 调色板内的所有步骤都将被选中。

返回 无。

Enabler 无。

例子 下面的代码将选定 History 调色板内的第一、第二和第三步的操作：

```
dw.historyPalette.setSelection( [0,1,2] );
```

※ **dreamweaver.historyPalette.setUndoState()**

适用版本 Dreamweaver 3.0 以上。

描述 执行正确数目的 undo 或 redo 操作,直到达到指定的 undo 状态。

参数 undoState。

此参数是由 dw.historyPalette.getUndoState()调用返回的对象。

返回 无。

Enabler 无。

17.1.16 HTML 风格函数

HTML 风格函数处理的是应用、创建和删除 HTML 风格的功能。dreamweaver.htmlStylePalette 对象的方法控制和作用于 HTML Style 调色板内的选定对象,而不是当前文档内的选定对象。

※ **dom.applyHTMLStyle()**

适用版本 Dreamweaver 3.0 以上。

描述 将指定的 HTML 风格应用到当前选定对象上。此函数仅对活动文档有效。

参数 htmlStyleName。

返回 无。

Enabler 无。

※ **dreamweaver.htmlStylePalette.deleteSelectedStyle()**

适用版本 Dreamweaver 3.0 以上。

描述 从 HTML Style 调色板上移走选定的风格。

参数 无。

返回 无。

Enabler dreamweaver.htmlStylePalette.canEditSelection()。

- ※ **dreamweaver.htmlStylePalette.duplicateSelectedStyle()**
适用版本 Dreamweaver 3.0 以上。
描述 复制选定的风格，并打开 Define HTML Style 对话框。
参数 无。
返回 无。
Enabler dreamweaver.htmlStylePalette.canEditSelection()。
- ※ **dreamweaver.htmlStylePalette.editSelectedStyle()**
适用版本 Dreamweaver 3.0 以上。
描述 为选定风格打开 Define HTML Style 对话框。
参数 无。
返回 无。
Enabler dreamweaver.htmlStylePalette.canEditSelection()。
- ※ **dreamweaver.htmlStylePalette.getSelectedStyle()**
适用版本 Dreamweaver 3.0 以上。
描述 获得 HTML Style 调色板中选定风格的名称。
参数 无。
返回 返回包含选定风格的名称的一个字符串。
Enabler 无。
- ※ **dreamweaver.htmlStylePalette.getStyles()**
适用版本 Dreamweaver 3.0 以上。
描述 获得 HTML Style 调色板中定义的所有 HTML 风格的名称的一个列表。
参数 无。
返回 返回一个字符串数组，数组中每个元素分别表示一个 HTML 风格的名称。如果当前一个 HTML 风格都没有定义，那么此函数将返回一个空数组。
Enabler 无。
- ※ **dreamweaver.htmlStylePalette.newStyle()**
适用版本 Dreamweaver 3.0 以上。
描述 为一个新的、无标题的风格打开 Define HTML Style 对话框。
参数 无。
返回 无。
Enabler 无。
- ※ **dreamweaver.htmlStylePalette.setSelectedStyle()**
适用版本 Dreamweaver 3.0 以上。
描述 选定 HTML Style 调色板上指定的风格。
参数 htmlStyleName。

返回 无。

Enabler 无。

17.1.17 层和影像地图函数

层和影像地图函数处理的是对齐、改变大小和移动层和影像地图热区的功能。函数描述符暗示了该函数是对层还是对影像地图热区进行操作的。

※ dom.align()

适用版本 Dreamweaver 3.0 以上。

描述 将选定的层或影像地图热区进行 left、right、top 或 bottom 方式的对齐。

参数 alignDirection。

此参数是选定层或影像地图热区进行对齐的方式：“left”、“right”、“top”或“bottom”。

返回 无。

Enabler dom.canAlign()。

※ dom.arrange()

适用版本 Dreamweaver 3.0 以上。

描述 按指定方向移动选定影像地图热区。

参数 toBackOrFront。

此参数是选定影像地图热区应该移动的方向：“front”或“back”。

返回 无。

Enabler dom.canArrange()。

※ dom.makeSizesEqual()

适用版本 Dreamweaver 3.0 以上。

描述 将选定层或影像地图热区的高度、宽度或整个大小都变成的相等的，改变是以最后选定的层或影像地图热区为标准。

参数 bHoriz, bVert。

- 第一个参数是一个布尔值，用于指示是否重新设置选定层或影像地图热区水平方向的尺寸。
- 第二个参数是一个布尔值，用于指示是否重新设置选定层或影像地图热区竖直方向的尺寸。

返回 无。

Enabler 无。

※ dom.moveSelectionBy()

适用版本 Dreamweaver 3.0 以上。

描述 向水平和竖直方向按指定像素移动指定层或影像地图热区。

参数 x, y。

- 第一个参数是选定对象应该向水平方向移动的像素数目。
- 第二个参数是选定对象应该向竖直方向移动的像素数目。

返回 无。

Enabler 无。

※ **dom.resizeSelectionBy()**

适用版本 Dreamweaver 3.0 以上。

描述 重设当前选定层或热区的大小。

参数 left, top, bottom, right。

- 第一个参数是选定层或热区的左边界的新位置。
- 第二个参数是选定层或热区的顶部的新位置。
- 第三个参数是选定层或热区的底部的新位置。
- 第四个参数是选定层或热区的右边界的新位置。

返回 无。

Enabler 无。

例子 如果选定层具有图 17-8 所示的 Left、Top、Width 和 Height 属性，那么调用：

```
dw.getDocumentDOM().resizeSelectionBy( -10,-30,30,10)
```

将重设层的大小：Left 为 44，Top 为 219，Width 为 106，Height 为 131。



图 17-8
选定层的
尺寸属性

※ **dom.setLayerTag()**

适用版本 Dreamweaver 3.0 以上。

描述 指定定义选定层（可以是多个）的 HTML 标记。

参数 tagName。

此参数必须是下列值之一：“layer”、“ilayer”、“div”或“span”。

- 第二个参数是选定层或热区的顶部的新位置。
- 第三个参数是选定层或热区的底部的新位置。
- 第四个参数是选定层或热区的右边界的新位置。

返回 无。

Enabler 无。

17.1.18 窗口函数

窗口函数处理与 Document 窗口和浮动调色板有关的操作。这些函数控制浮动调色板的显示和隐藏；决定拥有焦点的 Document 窗口中的窗格；以及设置活动文档。

※ **dom.getFocus()**

适用版本 Dreamweaver 3.0 以上。

描述 决定文档中拥有焦点的部分。

参数 无。

返回 返回下列字符串之一：

“ head ”：如果 HEAD 域是活动的。

“ body ”：如果 BODY 或 NOFRAMES 域是活动的。

“ frameset ”：如果一个框架集或它的任一框架被选定。

“ none ”：如果焦点不在文档内。例如，如果焦点在 Property 检查器或在另一浮动调色板上。

Enabler 无。

※ **dom.getWindowTitle()**

适用版本 Dreamweaver 3.0 以上。

描述 获得包含文档的窗口的标题。

参数 无。

返回 返回包含出现在文档的 TITLE 标记之间的文本的一个字符串，或者，如果文档没有在一个打开的窗口中，那么将返回空字符串。

Enabler 无。

※ **dreamweaver.getActiveWindow()**

适用版本 Dreamweaver 3.0 以上。

描述 获得活动窗口中内的文档。

参数 无。

返回 返回与活动窗口中的文档相对应的文档对象。或者，如果文档在一个框架中，那么将返回与框架相对应的文档对象。

Enabler 无。

※ **dreamweaver.getDocumentList()**

适用版本 Dreamweaver 3.0 以上。

描述 获得所有打开文档的一个列表。

参数 无。

返回 返回一个文档对象数组，每个元素与对应于一个打开的 Document 窗口。如

果一个 Document 窗口包含一个框架集，那么文档对象是指对应于框架集的，而不是框架的内容。

Enabler 无。

※ dreamweaver.getFocus()

适用版本 Dreamweaver 3.0 以上。

描述 决定拥有焦点的应用程序部分。

参数 bAllowFloaters。

返回 返回下列字符串之一：

“ document ”：如果 Document 窗口是活动的。

“ site ”：如果 Site 窗口是活动的。

· floaterName：如果 bAllowFloaters 是 TRUE，并且有一个浮动调色板拥有焦点，floaterName 是下列字符串之一：“ object ”、“ properites ”、“ launcher ”、“ library ”、“ css styles ”、“ html styles ”、“ behaviors ”、“ tielines ”、“ html ”、“ layers ”、“ frames ”、“ templates ”、“ history ”、“ data bindings ”或“ server behaviors ”。

· 在 Macintosh 平台上，如果既没有一个 Site 窗口也没有任一个 Document 窗口是打开的，那么返回一个空字符串。

Enabler 无。

※ dreamweaver.getFloaterVisibility()

适用版本 Dreamweaver 3.0 以上。

描述 查看指定的调色板或检查器是否可视。

参数 floaterName。

此参数是一个浮动调色板的名称。built-in 调色板必须是下列字符串之一：“ object ”、“ properites ”、“ launcher ”、“ library ”、“ css styles ”、“ html styles ”、“ behaviors ”、“ tielines ”、“ html ”、“ layers ”、“ frames ”、“ templates ”、“ history ”、“ data bindings ”或“ server behaviors ”。如果 floaterName 与这些 build-in 调色板没有一个是匹配的，那么 DW UltraDev 将在 Configuration/Floaters 文件夹下查找一个名字为 floaterName.htm 的文件。

返回 如果指定的浮动调色板可视并且在屏幕的最前面，那么返回 TRUE；否则返回 FALSE。当 DW UltraDev 不能找到 floaterName 指定的一个浮动调色板时，返回 FALSE。

Enabler 无。

※ dreamweaver.setActiveWindow()

适用版本 Dreamweaver 3.0 以上。

描述 将包含指定文档的窗口设置为活动的窗口。

参数 documentObject, {bActivateFrame}。

· 第一个参数是文档 DOM 树的树根处对象，由 dreamweaver.getDocumentDOM()函

数返回。

- 第二个参数是一个布尔值，仅当 documentObject 位于一个框架集内部时此参数有可用，它用于指示将包含框架集的窗口设置为活动的同时，是否将包含文档的框架设置为活动的。

返回 无。

Enabler 无。

※ dreamweaver.setFloaterVisibility()

适用版本 Dreamweaver 3.0 以上。

描述 指定是否将一个特定的浮动调色板或检查器设置为可视的。

参数 floaterName, bIsVisible。

- 第一个参数是一个浮动调色板的名称。built-in 调色板必须以下列字符串之一引用：
“ object ”、“ properites ”、“ launcher ”、“ library ”、“ css styles ”、“ html styles ”、
“ behaviors ”、“ tiemlines ”、“ html ”、“ layers ”、“ frames ”、“ templates ”、
“ history ”、“ data bindings ”或 server behaviors ”。如果 floaterName 与这些 build-in
调色板没有一个是匹配的，那么 DW UltraDev 将在 Configuration/Floaters 文件夹
下查找一个名字为 floaterName.htm 的文件。如果 DW UltraDev 不能找到指定的文件，
那么此函数将不起任何作用。
- 第二个参数是一个布尔值，用于指示是否将浮动调色板设置为可视的。

返回 无。

Enabler 无。

※ dreamweaver.showProperties()

适用版本 Dreamweaver 3.0 以上。

描述 将 Property 检查器设置为可视的，并且将焦点也交给它。

参数 无。

返回 无。

Enabler 无。

※ dreamweaver.toggleFloater()

适用版本 Dreamweaver 3.0 以上。

描述 显示、隐藏或将指定调色板或检查器放置到屏幕的最前面。



此函数仅在 menus.xml 文件中有意义。为了显示、隐藏或将指定浮动调色板的位置置为最前，请使用 dw.setFloaterVisibility()。

参数 floaterName。

返回 无。

Enabler 无。

17.1.19 Enabler 函数

Enabler 函数主要是用于决定是否使能某些菜单项,以便 DW UltraDev 可以在当前上下文内执行特定的操作。Enabler 函数规范描述了函数返回 TRUE 时的通用环境。但是这些描述并不是很广泛,可以没有描述某些函数返回 FALSE 时的情况。

※ **dom.canAlign()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行对齐操作,即 :Align Left、Align Right、Align Top 或 Align Bottom 操作。

参数 无。

返回 返回一个布尔值,用于指示是否有两个或更多层或热区被选中。

※ **dom.canApplyTemplate()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Apply To Page 操作。此函数仅对活动文档有效。

参数 无。

返回 返回一个布尔值,用于指示文档是否不是一个库 item 或一个模板,并且选定没有在 NOFRAMES 标记内。

※ **dom.canArrange()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行 Bring to Front 或 Move to Back 操作。此函数仅对活动文档有效。

参数 无。

返回 返回一个布尔值,用于指示是否有一个热区被选中。

※ **dom.canClipCopyText()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Copy as Text 操作。

参数 无。

返回 返回一个布尔值,用于指示当前选定是否是一个范围,也就是说,不是一个插入点。

※ **dom.canClipPaste()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Paste 操作。

参数 无。

返回 返回一个布尔值，用于指示当前剪贴板是否包含可以粘贴到 DW UltraDev 中的任意内容。

※ **dom.canClipPasteText()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Paste As Text 操作。

参数 无。

返回 返回一个布尔值，用于指示当前剪贴板是否包含可以当作文本粘贴到 DW UltraDev 中的任意内容。

※ **dom.canConvertLayersToTable()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Convert Layers to Table 操作。

参数 无。

返回 返回一个布尔值，用于指示文档的 BODY 中的所有内容是否都包含在层内。

※ **dom.canConvertTablesToLayers()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Convert Table to Layers 操作。

参数 无。

返回 返回一个布尔值，用于指示文档的 BODY 中的所有内容是否都包含在表格内，并且文档不是基于一个模板的。

※ **dom.canDecreaseColspan()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Decrease Colspan 操作。

参数 无。

返回 返回一个布尔值，用于指示当前单元是否有一个 COLSPAN 属性，并且此属性值是否大于或等于 2。

※ **dom.canDecreaseRowspan()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Decrease Rowspan 操作。

参数 无。

返回 返回一个布尔值，用于指示当前单元是否有一个 ROWSPAN 属性，并且此属性值是否大于或等于 2。

※ **dom.canDeleteTableColumn()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Delete Column 操作。

参数 无。

返回 返回一个布尔值，用于指示插入点是否在一个单元内，或者是否有一个单元或列被选中。

※ **dom.canDeleteTableRow()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Delete Row 操作。

参数 无。

返回 返回一个布尔值，用于指示插入点是否在一个单元内，或者是否有一个单元或行被选中。

※ **dom.canEditNoFramesContent()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Edit NoFrames Content 操作。

参数 无。

返回 返回一个布尔值，用于指示当前文档是否是一个框架集，或在一个框架集内。

※ **dom.canIncreaseColspan()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Increase Colspan 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在任意的单元正好位于当前单元的右方。

※ **dom.canIncreaseRowspan()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Increase Rowspan 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在任意的单元正好位于当前单元的下方。

※ **dom.canInsertTableColumns()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Insert Column(s) 操作。

参数 无。

返回 返回一个布尔值，用于指示选定是否在一个表内。如果选定是整个表格，那么此函数返回 FALSE。

※ **dom.canInsertTableRows()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Insert Row(s) 操作。

参数 无。

返回 返回一个布尔值，用于指示选定是否在一个表内。如果选定是整个表格，那

么此函数返回 FALSE。

※ **dom.canMakeNewEditableRegion()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 New Editable Region 操作。

参数 无。

返回 返回一个布尔值，用于指示当前文档是否是一个模板文件(.dwt)。

※ **dom.canMarkSelectionAsEditable()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Mark Selection as Editable 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在一个选定，并且当前文档是否是一个模板文件(.dwt)。

※ **dom.canMergeTableCells()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Merge Cells 操作。

参数 无。

返回 返回一个布尔值，用于指示选定是否是一个表格单元的矩形集。

※ **dom.canPlayPlugin()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Play 操作。此函数仅对活动文档有效。

参数 无。

返回 返回一个布尔值，用于指示选定是否可以用一个插件播放。

※ **dom.canRedo()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Redo 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在为 redo 操作保留的任意步骤。

※ **dom.canRemoveEditableRegion()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Unmark Editable Region 操作。

参数 无。

返回 返回一个布尔值，用于指示当前文档是否是一个模板。

※ **dom.canSelectTable()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Select Table 操作。

参数 无。

返回 返回一个布尔值，用于指示插入点或选定是否位于一个表格内。

※ **dom.canSetLinkHref()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以改变包围当前选定的链接，或者必要的话，创建一个新的。

参数 无。

返回 返回一个布尔值，用于指示选定是否是一个图像、文本、或一个链接内的插入点。

※ **dom.canShowListPropertiesDialog()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以显示 List Properties 对话框。

参数 无。

返回 返回一个布尔值，用于指示选定是否在 LI 标记内。

Enabler 无。

※ **dom.canSplitFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Split Frame[Left|Right|Up|Down]操作。

参数 无。

返回 返回一个布尔值，用于指示选定是否在一个在一个框架内。

※ **dom.canSplitTableCell()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Split Cell 操作。

参数 无。

返回 返回一个布尔值，用于指示插入点是否在一个表格单元的内部，或者当前选定就是一个表格单元。

※ **dom.canStopPlugin()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Stop 操作。

参数 无。

返回 返回一个布尔值，用于指示选定是否正在用一个插件播放。

※ **dom.canUndo()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Undo 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在为 undo 操作保留的步骤。

※ **dom.hasTracingImage()**

适用版本 Dreamweaver 3.0 以上。

描述 查看文档是否有一个跟踪图像。

参数 无。

返回 返回一个布尔值，用于指示文档是否有一个跟踪图像。

※ **dreamweaver.canClipCopy()**

适用版本 Dreamweaver 3.0 以上

描述 查看 DW UltraDev 是否可以执行一个 Copy 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在可以拷贝到剪贴板中的选定内容。

※ **dreamweaver.canClipCut()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Cut 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在可以剪切到剪贴板中的选定内容。

※ **dreamweaver.canClipPaste()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Paste 操作。

参数 无。

返回 返回一个布尔值，用于指示当前剪贴板是否存在可以粘贴到当前文档或 Site 窗口中活动窗格中的内容，或者，在 Macintosh 平台上，是指可以粘贴到浮动调色板或对话框中的编辑域。

※ **dreamweaver.canDeleteSelection()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以删除当前选定内容。删除可能会发生在 Document 窗口或 Site 窗口，这取决于拥有焦点的窗口。在 Macintosh 平台上，删除可能会发生在对话框或浮动调色板上的一个编辑域。

参数 无。

返回 返回一个布尔值，用于指示选定是否是一个范围，也就是说，不是一个插入

点。

※ **dreamweaver.canExportCSS()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Export CSS Styles 操作。

参数 无。

返回 返回一个布尔值，用于指示文档是否包含在 HEAD 部分定义的任意类风格。

※ **dreamweaver.canFindNext()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Find Next 操作。

参数 无。

返回 返回一个布尔值，用于指示是否已经建立了一个查寻模式。

※ **dreamweaver.canOpenInFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Open in Frame 操作。

参数 无。

返回 返回一个布尔值，用于指示选定或插入点是否在一个框架内。

※ **dreamweaver.canPlayRecordedCommand()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Play Recorded Command 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在可以播放的一个活动文档和一个先前记录的命令。

※ **dreamweaver.canRedo()**

适用版本 Dreamweaver 3.0 以上。

描述 查看在当前上下文中 DW UltraDev 是否可以执行一个 Redo 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在一个可以 undo 的操作。

※ **dreamweaver.canRevertDocument()**

适用版本 Dreamweaver 3.0 以上。

描述 查看在当前上下文中 DW UltraDev 是否可以执行一个 Revert(转换到上一次保存的版本) 操作。

参数 documentObject。

此参数是一个文档的 DOM 树的树根处的对象，是 dreamweaver.getDocumentDOM() 函数的返回值。

返回 返回一个布尔值，用于指示文档是否处在未保存状态，并且文档的一个保存版本在本地驱动器上。

※ **dreamweaver.canSaveAll()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Save All 操作。

参数 无。

返回 返回一个布尔值,用于指示是否存在一个或多个未保存的文档处在打开状态。

※ **dreamweaver.canSaveDocument()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以在指定文档上执行一个 Save 操作。

参数 documentObject。

此参数是一个文档的 DOM 树的树根处的对象,是 dreamweaver.getDocumentDOM() 函数的返回值。

返回 返回一个布尔值,用于指示文档是否存在未保存的修改。

※ **dreamweaver.canSaveDocumentAsTemplate()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以在指定文档上执行一个 Save As Template 操作。

参数 documentObject。

此参数是一个文档的 DOM 树的树根处的对象,是 dreamweaver.getDocumentDOM() 函数的返回值。

返回 返回一个布尔值,用于指示文档是否可以当作一个模板来保存。

※ **dreamweaver.canSaveFrameset()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以在指定文档上执行一个 Save Frameset 操作。

参数 documentObject。

此参数是一个文档的 DOM 树的树根处的对象,是 dreamweaver.getDocumentDOM() 函数的返回值。

返回 返回一个布尔值,用于指示文档是否是一个存在未保存的修改的一个框架集。

※ **dreamweaver.canSaveFramesetAs()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以在指定文档上执行一个 Save Frameset As 操作。

参数 documentObject。

此参数是一个文档的 DOM 树的树根处的对象,是 dreamweaver.getDocumentDOM() 函数的返回值。

返回 返回一个布尔值,用于指示文档是否是一个框架集。

※ **dreamweaver.canSelectAll()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Select All 操作。

参数 无。

返回 返回一个布尔值,用于指示一个 Select All 操作是否可以执行。

※ **dreamweaver.canShowFindDialog()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Find 操作。

参数 无。

返回 返回一个布尔值，用于指示是否 Site 窗口或一个 Document 窗口处于打开状态。

当选定位于 HEAD 部分，那么此函数返回 FALSE。

※ **dreamweaver.canUndo()**

适用版本 Dreamweaver 3.0 以上。

描述 查看在当前上下文中 DW UltraDev 是否可以执行一个 Undo 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在一个可以 Undo 的操作。

※ **dreamweaver.isRecording()**

适用版本 Dreamweaver 3.0 以上

描述 报告 DW UltraDev 当前是否正在记录一条命令。

参数 无。

返回 返回一个布尔值，用于指示 DW UltraDev 是否正在记录一条命令。

※ **dreamweaver.htmlStylePalette.canEditSelection()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以编辑、删除或复制在 HTML Style 调色板上的选定。

参数 无。

返回 返回一个布尔值。如果没有任何选定的风格，或者"clear"风格之一被选中，

那么此函数返回 FALSE。

※ **dreamweaver.timelineInspector.canAddFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Add Frame 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器是否有动画条或动作。

※ **dreamweaver.timelineInspector.canAddKeyFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Add Keyframe 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器内的选定是否是一个动画条的一部分。

※ **dreamweaver.timelineInspector.canChangeObject()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Change Object 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器内的选定是否是一个动画条的一部分。

✳ **dreamweaver.timelineInspector.canRemoveBehavior()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Remove Behavior 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器内的选定是否是一个动作。

✳ **dreamweaver.timelineInspector.canRemoveFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Remove Frame 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器是否有动画条或动作。

✳ **dreamweaver.timelineInspector.canRemoveKeyFrame()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Remove Keyframe 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器内当前帧是否是一个关键帧。

✳ **dreamweaver.timelineInspector.canRemoveObject()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Remove Object 操作。

参数 无。

返回 返回一个布尔值，用于指示 Timeline 检查器是否有动画条。

✳ **site.canAddLink()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Add Link to [Existing File | New File]操作。

参数 无。

返回 返回一个布尔值，用于指示站点地图中的选定文档是否是一个 HTML 文件。

✳ **site.canChangeLink()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Change Link 操作。

参数 无。

返回 返回一个布尔值，用于指示一个 HTML 文件或 Flash 文件链接到站点地图中

的选定文件。

※ **site.canCheckIn()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个获取权限并上传 (Check In) 操作。

参数 siteOrURL。

此参数必须是关键字 “site”，用于指明此函数应该作用在 Site 窗口内的选定，或者是一个文件的 URL。

返回 返回一个布尔值，用于指示下面所有的条件是否都是真的：

- 定义了一个远程站点。
- 如果一个 Document 窗口拥有焦点，那么文档文件已经在本地驱动器上保存过；或者，如果 Site 窗口拥有焦点，那么存在一个或多个文件或文件夹是处于选定状态。
- Check in/Check out 的开关是打开的。

※ **site.canCheckOut()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否有权对一个指定文件 (可以是多个) 执行下载操作。

参数 siteOrURL。

此参数必须是关键字 “site”，用于指明此函数应该作用在 Site 窗口内的选定，或者是一个文件的 URL。

返回 返回一个布尔值，用于指示下面所有的条件是否都是真的：

- 定义了一个远程站点。
- 如果一个 Document 窗口拥有焦点，那么文档文件是本地站点的一部分，并且还没有执行获取授权并下载操作 (Check out)；或者，如果 Site 窗口拥有焦点，那么存在一个或多个文件或文件夹是处于选定状态，并且至少存在一个选定文件还没有执行获取授权并下载操作 (check out)。
- Check in/Check out 的开关是打开的。

※ **site.canConnect()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以连接到远程站点。

参数 无。

返回 返回一个布尔值，用于指示当前远程站点是否是一个 FTP 站点。

※ **site.canFindLinkSource()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Find Link Source 操作。

参数 无。

返回 返回一个布尔值，用于指示站点地图中的选定链接是否不是主页。

※ **site.canGet()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Get 操作。

参数 siteOrURL

此参数必须是关键字“ site ”，用于指明此函数应该作用在 Site 窗口内的选定；或者是一个文件的 URL。

返回 如果参数是“ site ”，那么返回的布尔值指示在 Site 窗口中是否存在选中的文件或文件夹，并且定义了一个远程站点。如果参数是一个 URL，那么返回的布尔值指示文档是否属于一个已经定义了一个远程站点的一个站点。

※ **site.canLocateInSite()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Locate in Local Site 或 Locate in Remote Site 操作，至于执行这两个之中的哪一个取决于传入的参数。

参数 localOrRemote, siteOrURL。

- 第一个参数必须是“ local ”或“ remote ”。
- 第二个参数必须是关键字“ site ”，用于指明函数应该运行在 Site 窗口中的选定；或是一个文件的 URL。

返回 返回下列值之一：

- 如果第二个参数是“ site ”，那么返回的布尔值指示是否所有窗格都包含站点文件（而不是站点地图）。
- 如果第一个参数是“ local ”，并且第二个参数是一个 URL，那么返回的布尔值指示文档是否属性一个站点。
- 如果第一个参数是“ remote ”，并且第二个参数是一个 URL，那么返回的布尔值指示文档是否属性一个已经定义远程站点的站点，另外，如果服务器类型是 Local/Network，那么此布尔值也用于指示驱动器是否已经正确安装了。

※ **site.canMakeEditable()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Turn Off Read Only 操作。

参数 无。

返回 返回一个布尔值，用于指示是否存在一个或多个选定文件被锁定了。

※ **site.canMakeNewFileOrFolder()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以在 Site 窗口上执行一个 New File 或 New Folder 操作。

参数 无。

返回 返回一个布尔值,用于指示在 Site 窗口的选定窗格中是否存在可视的文件。

※ **site.canOpen()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以打开在 Site 窗口中当前已经选定的文件或文件夹。

参数 无。

返回 返回一个布尔值,用于指示在 Site 窗口中是否存在选定的文件或文件夹。

※ **site.canPut()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Put 操作。

参数 siteOrURL。

此参数必须是关键字“ site”,用于指明此函数应该作用在 Site 窗口内的选定;或者是一个文件的 URL。

返回 如果参数是“ site”,那么返回的布尔值指示在 Site 窗口中是否存在选中的文件或文件夹,并且定义了一个远程站点。如果参数是一个 URL,那么返回的布尔值指示文档是否属于一个已经定义了一个远程站点的一个站点。

※ **site.canRecreateCache()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Recreate Site Cache 操作。

参数 无。

返回 返回一个布尔值,用于指示 Use Cache To Speed Link Updates 选项是否对当前站点可用。

※ **site.canRefresh()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Refresh [Local |Remote]操作。

参数 localOrRemote。

此参数必须是“ local ”或“ remote ”。

返回 返回一个布尔值,如果 localOrRemote 是“ local”,那么返回 TRUE;否则此布尔值用于指示是否已经定义了一个远程站点。

※ **site.canRemoveLink()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Refresh Link 操作。

参数 无。

返回 返回一个布尔值,用于指示一个 HTML 或 Flash 文件链接到站点地图中选定

的文件。

※ **site.canSetLayout()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Layout 操作。

参数 无。

返回 返回一个布尔值，用于指示站点地图是否可视。

※ **site.canSelectNewer()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Select Newer [Remote | Local]操作。

参数 localOrRemote

此参数必须是‘ local ’或‘ remote ’。

返回 返回一个布尔值，用于指示文档是否属于定义了远程站点的一个站点。

※ **site.canShowPageTitles()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Show Page Titles 操作。

参数 无。

返回 返回一个布尔值，用于指示站点地图是否可视。

※ **site.canSynchronize()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Synchronize 操作。

参数 无。

返回 返回一个布尔值，用于指示是否定义了远程站点。

※ **site.canUndoCheckOut()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 Undo Checkout 操作。

参数 siteOrURL。

此参数必须是‘ site ’或一个文件的 URL，如果是‘ site ’，那么是用于指明此函数应该运行在 Site 窗口内的选定对象上。

返回 返回一个布尔值，用于指示指定文件或至少有一个选定文件已经执行授权下载操作了。

※ **site.canViewAsRoot()**

适用版本 Dreamweaver 3.0 以上。

描述 查看 DW UltraDev 是否可以执行一个 View as Root 操作。

参数 无。

返回 返回一个布尔值，用于指示指定的文件是否是一个 HTML 或 Flash 文件。

17.2 文件 I/O API

Dreamweaver 3.0 包含一个称之为 DWfile 的 C 共享库，它为我们提供了读写本地文件的能力。本节就是描述这些文件 I/O API。

为了能正确访问 DWfile 库中的函数，必须将此库安装在 Configuration/JSExtensions 文件夹下，并且由 Dreamweaver 装载。因为 Dreamweaver 2.0 还没有装配 DWfile 库，用户可以通过下载得到并安装，所以在调用它的任意函数之前应该首先检验它的可用性。可以通过调用函数 `typeof( DWfile)` 检查此库的可用性，如果 DWfile 不存在，那么此调用将返回 `undefined`。例如，如果你正在一个命令的上下文中使用 DWfile，此时你可以作为 `canAcceptCommand()` 函数的一部分检查 DWfile 库的存在性：

```
// Returns TRUE if typeof( DWfile) is not undefined, FALSE otherwise.
function canAcceptCommand(){
    return( typeof( DWfile) != "undefined" ) ;
}
```

文件 I/O API 中的所有函数都是 DWfile 对象的方法。下面介绍的函数中花括号 { } 中的内容是可选参数。

※ DWfile.copy()

适用版本 Dreamweaver 3.0 以上。

描述 将指定文件拷贝到指定的 URL 处。

参数 originalURL、copyURL。

- 第一个参数是要拷贝的文件，格式为：file://URL。
- 第二个参数是拷贝的目标位置，格式为：file://URL。

返回 如果拷贝成功，此函数返回 TRUE；否则返回 FALSE。

例子 下面代码将文件 myconfig.cfg 拷贝为 myconfig_backup.cfg：

```
var fileURL = "file:///c:/Config/myconfig.cfg";
var newURL = "file:///c:/Config/myconfig_backup.cfg";
DWfile.copy( fileURL, newURL ) ;
```

※ DWfile.createFolder()

适用版本 Dreamweaver 2.0 以上。

描述 在指定位置处创建一个文件夹。

参数 folderURL。

此参数是要创建的文件夹位置，格式为：file://URL。

返回 如果创建成功，此函数返回 TRUE；否则返回 FALSE。

例子 下面的代码将试图在 C 盘根目录下创建一个 tempFolder 文件夹，并且显示一个提示信息对话框以指示操作是否成功：

```
var folderURL = "file:///c:/tempFolder";
if( DWfile.createFolder( folderURL)) {
    alert( "Created " + folderURL) ;
}else{
    alert( "Unable to create " + folderURL) ;
}
```

※ DWfile.exists()

适用版本 Dreamweaver 2.0 以上。

描述 测试指定文件是否存在。

参数 fileURL。

此参数是将要查寻的文件，格式为：file://URL。

返回 如果指定的文件存在，则此函数返回 TRUE；否则返回 FALSE。

例子 下面的代码检查文件名为 mydata.txt 的文件，并且显示一信息对话框以指示此文件是否存在：

```
var fileURL = "file:///c:/temp/mydata.txt";
if( DWfile.exists( fileURL)) {
    alert( fileURL + " exists!") ;
}else{
    alert( fileURL + " does not exist.") ;
}
```

※ DWfile.read()

适用版本 Dreamweaver 2.0 以上。

描述 将指定文件的内容读到一个字符串中。

参数 fileURL。

此参数是将要读取的文件，格式为：file://URL。

返回 返回一个包含文件内容的字符串。如果读取失败，那么返回的是一个空字符串。

例子 下面的代码读取 mydata.txt 文件，并且如果读取成功，则显示一个信息对话框

```
var fileURL = "file:///c:/temp/mydata.txt";
var str = DWfile.read( fileURL) ;
if( str) {
    alert( fileURL + " contains: " + str) ;
}
```

✧ DWfile.write()

适用版本 Dreamweaver 2.0 以上。

描述 将指定的字符串写入指定文件。如果指定文件还不存在，则先创建再写入。

参数 fileURL, text, {mode}。

- 第一个参数是将写入的文件，格式为：file://URL。
- 第二个参数是将写入的字符串内容。
- 第三个参数如果指定的话，必须是“append”。如果省略此参数，那么文件的内容则被字符串覆盖。

返回 如果字符串成功写入文件，则此函数返回 TRUE；否则返回 FALSE。

例子 下面的代码试图将字符串“xxx”写入文件 mydata.txt 中，并且如果写入操作成功则显示一个提示信息对话框。然后，试图将字符串“aaa”添加到该文件，并且如果第二个操作也是成功的话，也显示一个提示信息对话框。如果以下两个操作都成功执行，那么此时文件 mydata.txt 的内容将是：xxxxaaa。

```
var fileURL = "file:///c:/temp/mydata.txt";
if( DWfile.write( fileURL, "xxx")) {
    alert( "Wrote xxx to " + fileURL) ;
}
if( DWfile.write( fileURL, "aaa", "append")) {
    alert( "Appended aaa to " + fileURL) ;
}
```

17.3 Design Notes API

Dreamweaver 3.0 和 Fireworks 3.0 都为网站设计者和开发者提供了一些新方法以保存和检取文档信息的附加信息，例如：检查命令、改变记录，或者改变 GIF 和 JPEG 对象的源文件，提供的这些新函数称之为 Design Notes。

MMNotes 是一个 C 共享库，允许扩展用户读写 Design Notes 文件。与 DWfile 共享库相似，MMNotes 有一个 JavaScript API，使用用户可以调用库中的函数。

这两个库之间的下点就是 MMNotes 库还有一个 C API，允许其他应用程序有机会读写 Design Notes 文件。MMNotes 共享库可以独立使用，也就是说，即使没有安装 Dreamweaver，也可以使用 MMNotes 库。

17.3.1 Design Notes JavaScript API

在 Design Notes JavaScript API 中的所有函数都是 MMNotes 对象的方法。下面介绍的函数中，大括号“ {} ”里的是指可选参数。

※ **MMNotes.open()**

描述 打开与指定文件相关的 Design Notes 文件，或者如果此文件还不存在，则新建一个。

参数 filePath, {bForceCreate}。

- 第一个参数是与 Design Notes 文件相关的主函数路径，格式为：file://URL。
- 第二个参数是一个布尔值，用于指示站点的 Design Notes 处于关闭状态，或者 filePath 并不与任一站点相关时，是否创建 Note。

返回 返回 Design Notes 文件的文件句柄；或者如果文件没有打开或没有创建，返回 0。

※ **MMNotes.close()**

描述 关闭指定的 Design Notes 文件，并且保存所作的任何修改。

参数 fileHandle。

此参数是由 MMNotes.open() 函数返回的文件句柄。

返回 无。

※ **MMNotes.set()**

描述 创建或刷新 Design Notes 文件内的一个 key/value 对。

参数 fileHandle, keyName, valueString。

- 第一个参数是由 MMNotes.open() 函数返回的文件句柄。
- 第二个参数是一个包含 key 名称的字符串。
- 第三个参数是包含 value 的一个字符串。

返回 返回一个布尔值，用于指示操作是否成功。

例子 下面的代码打开与 dreamcentral 站点上 peakhike99/index.html 文件相关的 Design Notes 文件，然后增加一个新的 key/value 对，改变一个已经存在 key 的值，最后关闭此 Notes 文件。

```
var noteHandle = MMNotes.open( file:///c:/sites/dreamcentral/
peakhike99/index.html',TRUE) ;
MMNotes.set( noteHandle,"Author","M. G. Miller") ;
MMNotes.set( noteHandle,"Last Changed","August 28, 1999") ;
MMNotes.close( noteHandle) ;
```

17.3.2 Design Notes C API

除了 JavaScript API 以外, MMNotes 共享库也为用户提供了一个 C API, 以允许其他的应用程序创建 Design Notes 文件。如果你正在 DW UltraDev 内使用 MMNotes 共享库的话, 那么没有必要直接调用这些 C 函数, JavaScript 版本的函数会为你调用它们。

下面简单地为你介绍这些函数的一部分, 这些函数的所有定义和数据类型都包含在 DW Ultradev 应用程序文件夹的 Extending/c_files 文件夹下的 MMInfo.h 文件中。

函数的可选参数是由大括号‘ {} ’包围的。

※ FileHandle OpenNotesFile()

描述 打开与指定文件相关的 Design Notes 文件, 或者此文件还不存在, 则创建一个新的。

参数 const char * localFileURL, {BOOL bForceCreate}。

- 第一个参数是包含与 Design Notes 文件相关的主文件路径的一个字符串。格式为：file://URL。
- 第二个参数是一个布尔值, 用于指示站点的 Design Notes 处于关闭状态, 或者 filePath 并不与任一站点相关时, 是否创建 Note。

※ void CloseNotesFile()

描述 关闭指定的 Design Notes 文件, 并且保存所作的任何修改。

参数 fileHandle noteHandle。

此参数是由 OpenNotesFile()函数返回的文件句柄。

返回 无。

17.4 Database API

数据库函数允许程序员与结构化查询语言(SQL) 和存储过程打交道。利用这些函数, 我们可以检索特定数据库模式信息。一个数据库模式就是一个数据库的结构。数据的结构包括数据表和列名。利用这些数据库 API 函数, 我们可以通过 SQL 语句和存储过

程获得数据库表和列名；获得建立数据库链接的用户名和口令；并显示执行 SQL 语句或存储过程后的结果。

数据库函数可以用于任意扩展中。实际上，UltraDev 服务器动作、数据格式，以及数据源 API 函数都利用数据库函数。

下面的代码展示了服务器动作函数 `getDynamicBindings()` 是如何定义 `Recordset`（文件 `Recordset.js` 位于 `/Configuration/ServerBehaviors/ASP` 目录下）。注意使用了函数 `MMDB.getColumnList()`。

```
function getDynamicBindings( elementNode)
{
    var ss = findSSrec( elementNode, LABEL_Type)

    var connString = ss.activeconnection
    var connName = ss.connectionName
    var statement = ss.source
    var rsName = ss.rsName

    var pa = new Array()

    if( String( ss.ParamArray) != "undefined")
    {
        for( var i = 0; i < ss.ParamArray.length; i++)
        {
            pa[i] = new Array()
            pa[i][0] = ss.ParamArray[i].name
            pa[i][1] = ss.ParamArray[i].value
        }
    }

    var statement = ReplaceParamsWithVals( statement, pa)
    return MMDB.getColumnList( connName, statement)
}
```

下面描述了数据库函数中一些通用的参数：

- 大部分数据库函数利用一个连接名称作为一个参数。我们可以在 UltraDev 的 Connection Manager 中发现有效连接名称列表，或者也可以利用

MMDB.getConnectionList()函数获得所有连接名称列表。

- 存储过程经常要求参数。有两种方式可以为数据库函数指定参数值。第一种方式，我们可以提供一个数据值数组 (paramValuesArray)。如果仅指定一个参数值，那么该值必须按照存储过程要求的参数顺序指定。第二种方式是提供参数名称数组 (paramNameArray) (我们可以利用 MMDB.getSPParamsAsString()函数获得存储过程的参数)。如果提供的是参数名称，那么在 paramValuesArray 中指定的值必须与 paramNameArray 指定的参数名顺序一致。

※ MMDB.getConnectionList()

适用版本 DW UltraDev 4。

描述 获得 Connection Manager 定义的所有链接字符串列表。

参数 无。

返回 一个字符串数组，其中每一个元素是 Connection Manager 中定义的一个链接名称字符串。

例子 调用 MMDB.getConnectionList()函数可以返回字符串：["EmpDB","Test","TestEmp"]。

※ MMDB.getConnectionString()

适用版本 DW UltraDev 4。

描述 获得指定连接名称相对应的链接字符串。

参数 connName。

此参数是在 Connection Manager 中指定的一个 UltraDev 链接名称。它用于将一个数据库链接成为活动数据源。

返回 一个与指定链接名称相对应的字符串。

例子 代码：

```
var connectionString = MMDB.getConnectionString( "EmpDB")
```

将为 ADO 或 JDBC 链接返回不同的字符串。

对 ADO 连接，上面的代码可能会返回字符串：

```
"dsn=EmpDB;uid=;pwd=";
```

对 JDBC 连接，上面的代码可能会返回字符串：

```
"jdbc:inetdae:192.168.64.49:1433?database=pubs&user=JoeUser&password=joesSecret"
```

※ MMDB.getSPColumnList()

适用版本 DW UltraDev 4。

描述 获得调用指定存储过程后产生的结果集数据库列列表。

参数 connName,statement,paramValuesArray

- 参数 connName 是 Connection Manager 中指定的一个 UltraDev 连接名称。它用于

将一个数据库连接成为活动数据源。

- 参数 `statement` 是执行时返回结果集的存储过程名称。
- 参数 `paramValuesArray` 是包含一系列设计时参数测试值的数组。一定要按存储过程要求的参数顺序设置参数值。我们可以利用 `MMDB.getSPParamsAsString()` 函数获得存储过程的参数。

返回 一个表示数据库列列表的字符串数组。如果 SQL 语句无效, 或者连接字符串是错误的, 此函数将返回一个错误信息。

※ `MMDB.getSPParamsAsString()`

适用版本 DW UltraDev 4。

描述 获得以逗号为分隔符的一个字符串, 此字符串包含了存储过程所需的参数列表。

参数 `connName, procName`。

- 参数 `connName` 是 Connection Manager 中指定的一个 UltraDev 连接名称。它用于将一个数据库连接成为活动数据源。
- 参数 `procName` 是存储过程名称。

返回 返回以逗号为分隔符的字符串, 此字符串包含了存储过程所需的参数列表。参数后跟着一个分号, 分号后是参数的名称、传递方向以及数据类型。

例子 代码:

```
MMDB.getSPParamsAsString( "EmpDB", "getNewEmployeesMakingAtLeast")
```

可能会返回下列字符串:

```
startDate;direction:in;datatype:date,
salary;direction:in;datatype:integer
```

以上代码说明存储过程 `getNewEmployeesMakingAtLeast` 有两个参数: `startDate` 和 `salary`。对参数 `startDate`, 传递方向是输入参数, 数据类型为 `date`; 对参数 `salary`, 传递方向也是输入参数, 数据类型为 `integer`。

17.5 Commands API

命令可以用于完成各种各样的编辑, 包括用户当前文档、其他打开文档, 或者本地驱动器上的任意 HTML 文档。命令可以插入、移走, 或者重组 HTML 标记和属性、注释和文本。

命令是 HTML 文件。命令文件的 BODY 部分包含一个接受选项的 HTML 表格(例如, 一个表格应该怎样排序和以哪个列为标准进行排序)。命令文件的 HEAD 部分包含

JavaScript 函数，这些函数处理从 BODY 部分的表格输入，以及控制对用户文档所作的编辑类型。

※ **commandButtons()**

描述 定义应该显示在选项对话框右边的按钮，以及单击它们时产生的动作。如果没有定义此函数，那么将没有按钮出现，并且命令文件的 BODY 部分将扩展以填充整个对话框。

参数 无。

返回 返回一个包含偶数个元素的数组。第一个元素是最顶端按钮的标签字符串。第二个元素是定义单击顶端按钮时产生的动作的 JavaScript 代码字符串。其余元素以同样的方式定义了其他按钮。

※ **canAcceptCommand()**

描述 判断指定的命令是否适合于当前选定。

参数 无。

返回 如果命令允许，则此函数返回 TRUE；否则返回 FALSE，并且将相应的菜单命令置灰。

※ **receiveArguments()**

描述 处理菜单项传来的任何参数。

参数 {arg1},{arg2},...{argN}。

如果 arguments 属性是为 menuitem 标记定义，那么此属性值将作为一个或多个参数传递给函数 receiveArguments()。

返回 无。

17.6 Data Sources API

所有的数据源文件都保存在 Configuration 文件夹下的 DataSources 文件夹内。每一种服务器模型都有它自己的文件夹：ASP、ColdFusion 和 JSP。在每一种服务器模型子文件夹下都是与此服务器模型相关的数据源 HTML 文件。

数据源 API 函数允许我们可以操纵 Data Binding 检查器和 Dynamic Data、Dynamic Text 对话框中的“+”菜单中的选项。

※ **addDynamicSource()**

适用版本 DW UltraDev 4。

描述 增加一个动态数据源。因为每一个数据源文件中都有一个此函数的实现，所以当从“+”弹出菜单中选择一个数据源时，UltraDev 将调用一个合适的 addDynamicSource() 函数的实现。

返回 无。

※ **deleteDynamicSource()**

适用版本 DW UltraDev 4。

描述 当 UltraDev 用户在树中选中一个数据源，并单击“-”按钮时，调用此函数。deleteDynamicSource()函数会在某些情况下显示一个出错信息。例如，如果用户试图删除一个记录集列，此时该函数会显示一条信息以提示用户将删除记录集。

参数 sourceName, bindingName。

- sourceName 是与子节点相关的顶级节点的名称。
- bindingName 是子节点的名称。

返回 无。

※ **generateDynamicDataRef()**

适用版本 DW UltraDev 4。

描述 为子节点产生动态数据对象。

参数 sourceName, bindingName。

- sourceName 是与子节点相关的顶级节点的名称。
- bindingName 是子节点的名称。

返回 返回一个字符串，该字符串可以传给函数 formatDynamicDataRef()，以便在将其插入用户文档之前先将其格式化。

17.7 Server Behaviors API

为了创建 Web 应用程序，DW UltraDev 用户利用服务器动作完成相关工作。服务器动作可以创建 Web 应用程序需要的功能，例如，基于用户条件过滤记录、为记录分页、将结果列表链接到细节页面，以及将记录插入到结果集。服务器动作是用 HTML 和 JavaScript 语言编制的，所以你也可以创建自己的服务器动作。

※ **analyzeServerBehavior()**

适用版本 DW UltraDev 4。

描述 允许服务器动作设置它们的 incomplete 和 deleted 标志。

参数 behavior, [behaviorArray]。

- behavior 是一个 JavaScript 对象，表示将被分析的动作。
- [behaviorArray] 是一个 JavaScript 对象的数组，表示在页面发现的所有服务器动作。

返回 无。

※ **applyServerBehavior()**

适用版本 DW UltraDev 4。

描述 从 Parameters 对话框的表格元素取值，并将动作加入到用户的文档。当用户单击 Parameters 对话框的 OK 按钮时，DW UltraDev 调用此函数。如果此函数返回成功信息，那么 Parameters 对话框关闭；否则，Parameters 对话框没有关闭，并且显示一个出错信息。此函数可以编辑用户的文档。

参数 behavior|null。

- behavior 是一个表示动作的 JavaScript 对象，如果修改一个已存在的动作，则作为参数传递。
- null：如果增加一个新动作，则此参数为 null。

返回 如果成功，返回一个空字符串；如果失败，显示出错信息，并且不关闭 Parameters 对话框。

17.8 其他 API

本节将为读者简单地介绍一些 DW UltraDev 为我们提供的其他 API 函数，这里只作功能性的说明，对于具体的函数，读者在需要时可以查阅 DW UltraDev 自带的帮助文档。

17.8.1 Fireworks Integration API

FWLaunch 是一个 C 共享库，为对象、命令、动作以及属性检查器的作者提供与 Fireworks 通信的能力。

Fireworks 集成 API 中的所有函数都是 FWLaunch 对象的方法。

17.8.2 HTTP API

从 Dreamweaver 3.0 开始，扩展不再局限于本地文件系统内工作。DW UltraDev 现在为我们提供了通过 HTTP 从某个网站服务器上获取或向其发送信息。这就是 HTTP API 的功能。

HTTP API 中的所有函数都是 MMHttp 对象的方法。大部分函数都采用了一个 URL 作为其一个参数，并且大部分函数都返回一个对象。对 URL 的缺省端口是 80；如果你希望指定端口，那么你可以在 URL 后面添加一个冒号以及指定的端口号，例如：

```
MMHttp.getText( "http://www.myserver.com:8025" );
```

对于返回值为对象的那些函数，返回的对象有两个属性：statusCode 和 data。

其中 `statusCode` 属性指示操作的属性，它的取值可以是以下一些值，但并不局限于此：

- 200：状态 OK。
- 400：无效请求。
- 404：请求的 URL 未能找到。
- 405：服务器不支持请求的方法。
- 500：未知服务器错误。
- 503：已达到服务器的最大容量。

为了得到你的服务器的详细状态码信息，请咨询你的 Internet 服务器提供商或系统管理员。

`data` 属性值根据函数不同而变化。

返回对象的函数也有一个“`callback`”版本。当网站服务器在处理一个 HTTP 请求的同时，`Callback` 函数允许其他函数执行。当你正在从 DW UltraDev 制造多个 HTTP 请求时，这种功能是非常有用的。函数的 `callback` 版本将它的 ID 和返回值直接传递给它第一个参数指定的函数。

17.8.3 C-Level 扩展 API

C 级扩展机制允许程序员组合使用 JavaScript 和自己的 C 代码从而实现 Dreamweaver 扩展文件。程序员可以利用 C 定义函数，并将它们捆绑到 DLL 或共享库中，将库保存在 `Configuration/JSExtensions` 文件夹下，然后利用已内置到 Dreamweaver 的 JavaScript 解释器从 JavaScript 调用这些函数。

```
JSBool
readContentsOfFile( JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)
{
    char *fileName, *fileContents;
    JSBool success;
    unsigned int length;

    /* Make sure caller passed in exactly one argument. If not,
     * then tell the interpreter to abort script execution. */
    if( argc != 1) {
        JS_ReportError( cx, "Wrong number of arguments", 0) ;
        return JS_FALSE;
    }

    /* Convert the argument to a string */
    fileName = JS_ValueToString( cx, argv[0], &length) ;
    if( fileName == NULL) {
        JS_ReportError( cx, "The argument must be a string", 0) ;
        return JS_FALSE;
    }
}
```

数内的更多的 JavaScript 数据结构和函数信息，请看下面的代码：

```

}

/* Use the string( the file name) to open and read a file */
fileContents = exerciseLeftToTheReader( fileName) ;

/* Store file contents in rval, which is the return value passed
 * back to the caller */
success = JS_StringToValue( cx, fileContents, 0, *rval) ;
free( fileContents) ;

/* Return true to continue or false to abort the script */
return success;

```

17.8.4 Objects API

对象的功能是将指定代码字符串插入到用户文档。一旦对象的文件保存在 Configuration/Objects 文件夹的一个子文件夹内，那么此对象就会出现在某个 Object 面板上，以及 Insert 菜单中。

对象有两个组件：对象文件和 GIF 图像。其中对象文件定义了插入文档的内容；而 18 像素×18 像素的 GIF 图像则显示在对象面板上，表示此对象。

对象是 HTML 文件。对象文件的 BODY 部分可以包含一个接受参数的 HTML 表格（例如，将插入的表格的行数和列数）。对象文件的 HEAD 部分包含 JavaScript 函数，这些函数处理从 BODY 部分的表格输入，以及控制加入用户文档的内容。

注意，最简单的对象仅包含将插入的没有 BODY 和 HEAD 标记的 HTML 内容。

17.8.5 Menu Commands API

Dreamweaver 3.0 引入了一个新型命令，它使菜单变得更灵活，并且使动态菜单成为可能。与其他一般命令一样，菜单命令可以用于完成各种各样的编辑，包括用户当前文档、其他打开文档，或者本地驱动器上的任意 HTML 文档。菜单命令 API 扩展了常规命令 API，以完成几个与菜单系统显示和调用命令有关的任务。

菜单命令是 HTML 文件。菜单命令文件的 BODY 部分可以包含一个从命令中接受选项的 HTML 表格（例如，一个表格应该如何排序）。菜单命令文件的 HEAD 部分包含处理 BODY 部分输入的 JavaScript 函数，并且控制可以对用户文档进行的编辑。

菜单命令保存在 Configuration/Menus 文件夹下。

17.8.6 Property Inspectors API

Property 检查器可能是你对 DW UltraDev 界面中所有浮动面板中最熟悉的一个面板了。它对选定的定义、参考和改变名称、大小、外观，以及其他属性的设置是必不可少的面板，同时它也是为选定元素启动内部和外部编辑器绝对必要的面板。

DW UltraDev 对 Property 检查器有几个内置接口，允许程序员为许多标准 HTML 标记设置属性。对定制属性检查器文件，程序员可以覆盖这些内置接口，或者创建一个新接口以检查定制标记。

Property 检查器文件是 HTML 文件，保存在 Configuration/Inspectors 文件夹下。

17.8.7 Floating Palettes API

在 Dreamweaver 3.0 中，我们可以创建任意种类的浮动面板或是检查器，同时并不需要受属性检查器的大小和布局限制。

一个定制的属性检查器仍应该是你设置当前选定属性的首要选择。然而，定制浮动面板为整个文档或多个选定内容的显示信息提供了更多的空间和更大的灵活性。

Dreamweaver 也有几个内置浮动面板，它们可以从 Window 菜单中访问。我们也可以利用扩展菜单特性将自己的面板加入到此菜单中。

浮动面板文件是 HTML 文件，保存在 Configuration/Floater 文件夹下。浮动面板文件的 BODY 部分包含一个 HTML 表格。

17.8.8 Behaviors API

动作允许非程序员制作他们自己的交互式 HTML 页面。它们为 Web 设计者提供了一个简单的方法将动作分配到页面元素。

当你需要与非程序员共享函数时，或者你希望重复插入同一个 JavaScript 函数，并且每次都改变参数值时，你也可以编写动作。

注意，你不能利用动作直接插入 VBScript 函数，然而，你可以通过在 applyBehavior() 函数中编辑 DOM，间接增加一个 VBScript 函数。

术语“动作”是指事件（例如：onClick、onLoad 或 onSubmit）和行动（例如，Check Plugin，Go to URL 或 Swap Image）的组合。浏览器为每个 HTML 元素决定其接受的事件。每种浏览器支持的事件列表文件保存在 Configuration/Behaviors/Events 文件夹下。

动作是 HTML 文件。一般来说，动作文件的 BODY 部分包含接受动作参数的一个

HTML 表格(例如:参数指明应该显示或隐藏的层)。动作文件的 HEAD 部分包含处理 BODY 部分表格输入的 JavaScript 函数,并且控制插入用户文档的函数、参数,以及事件句柄。

17.8.9 Data Translators API

数据转换器将特定标记(例如,SSI、带条件的 JavaScript 语句,或其他非 HTML 代码,比如:ePerl、PHP3、JSP、CFML 或 ASP)转换为可以被 DW UltraDev 读取和显示的 HTML 代码。如果你使用的是 Dreamweaver 3.0 以上版本,你可以在标记内、整个标记或代码块内转换属性。

转换的标记或代码块必须包围在锁定区域以保存初始的标记。转换的属性不要求加锁。

所有的数据转换器——块/标记或属性——都是 HTML 文件。一个数据转换器的 HEAD 部分包含两个或更多的 JavaScript 函数:其中一个指定工作的转换器和作用的文件;一个执行实际的转换工作;另外是一些支持函数。一个数据转换器的 BODY 部分是空的。

数据转换器——尤其对整个标记或代码块——可以涉及复杂的操作,这些操作可能利用 JavaScript 还不能完成,或许用 C 更容易实现。

UltraDev 利用 `getServerLanguages()`、`getFileExtensions()`和 `getVersionArray()`函数决定为用户在 Site Definition 对话框的 App Server Info 面板上显示的选项。

17.8.10 Server Models API

服务器模式是指用于在一个服务器上运行脚本的服务器技术。当用 DW UltraDev 定义了一个站点,用户标识正在使用站点的模式。当用户在站点的页面中增加了动态元素后,DW UltraDev 利用服务器模式以决定使用的合适代码。

所有服务器模式都保存在 Configuration 文件夹下的 ServerModels 目录下。每一个服务器模式都有它自己的文件:ASP 2.0、Cold Fusion 4.0 以及 JSP 1.0。