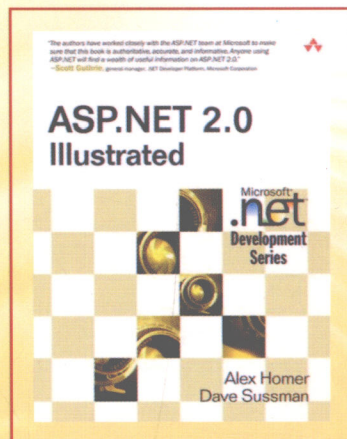


ASP.NET 2.0 Illustrated

ASP.NET 2.0 技术详解

[英] Alex Homer Dave Sussman 著
李胜权 苏金国 译

- 两位微软传奇技术大师联袂巨献
- ASP.NET 之父强烈推荐
- 迅速掌握 ASP.NET 2.0 精髓



TURING

图灵程序设计丛书

NET系列

ASP.NET 2.0技术详解

ASP.NET 2.0 Illustrated

[英] Alex Homer Dave Sussman 著

李胜权 苏金国 译

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

ASP.NET 2.0 技术详解 / (英) 霍默著; 李胜权等译.
—北京: 人民邮电出版社, 2007.11
(图灵程序设计丛书)
ISBN 978-7-115-16458-2

I. A… II. ①霍…②李… III. 主页制作—程序设计
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 093258 号

内 容 提 要

ASP.NET 已经发展成为一个相当成熟、丰富、全面的语言技术, 适合构建任何类型的 Web 应用。本书两位作者是仅有的来自英国的两位微软“软件传奇人物”, 一直与 ASP.NET 2.0 开发小组密切合作。本书简洁且详尽地提供了构建网站和 Web 应用时所需的 ASP.NET 2.0 信息, 并将重点放在 ASP.NET 2.0 的技术技巧上。书中所有的示例都在 Visual Studio 2005 中开发, 并且更新到 ASP.NET 2.0 的最新版本。

本书适用于各个层次的 ASP 开发人员阅读。

图灵程序设计丛书

ASP.NET 2.0 技术详解

-
- ◆ 编 [英] Alex Homer Dave Sussman
译 李胜权 苏金国
责任编辑 陈兴露
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 31.75
字数: 750 千字 2007 年 11 月第 1 版
印数: 1—5 000 册 2007 年 11 月北京第 1 次印刷
著作权合同登记号 图字: 01-2007-0543 号

ISBN 978-7-115-16458-2/TP

定价: 79.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Authorized translation from the English language edition, entitled *ASP.NET 2.0 Illustrated* by Alex Homer, Dave Sussman, published by Pearson Education, Inc., publishing as Addison-Wesley Professional. Copyright © 2006 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2007.

本书中文简体字版由 Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

对《ASP.NET 2.0技术详解》的赞誉

“作者与微软的ASP.NET开发小组的亲密合作保证了此书的权威性、准确性和丰富性。任何使用ASP.NET技术的人都会从中找到关于ASP.NET 2.0的大量有用的信息。”

——Scott Guthrie, .NET开发平台总经理, 微软公司

“对刚接触ASP.NET的人来说, 这本书是个不错的起点, 而已经熟悉ASP.NET老版本的程序员也可以把它作为一个全面的指南, 可以了解ASP.NET 2.0的新特性。开始的章节通过一个假想的公司详细介绍了有关的概念, 分析了大量的客户请求和场景。两位作者讨论起内容来生动传神, 又紧贴现实。很显然, 他们在这个领域中已身经百战, 具有丰富的实战经验。这本书把最基本到最复杂的任务都涵盖在内, 全面汇集了ASP.NET 2.0的有关信息。每个主题都提供了详细的参考资料, 以便读者了解更多信息, 与此同时, 各个主题还讲述了能有效工作的清晰示例。尽管本书技术性很强, 但是作者举重若轻, 写作简练清晰, 深入浅出。一些难于理解的概念采用了独特的方式来讲解, 相信在ASP.NET图书中, 这本书很快会成为你的最爱!”

——Ronda Pederson, 顾问, ASP/ASP.NET MVP

“此书出自当今顶尖的两位技术作者之手, 绝对精彩, 读者可以从中了解ASP 2.0 Framework的大部分信息。内容详尽的章节、简洁实用的示例、清晰透彻的解释, 这些都为读者充分利用ASP 2.0引入的新特性提供了坚实的基础和支持, 所有这些都采用一种逻辑清晰、直截了当的方法来阐述。相信与你共事的所有开发人员都会向你借阅这本书, 所以想办法保管好!”

——John Timney, 微软MVP, 英国电信公司高级Web服务顾问

“这本书会让你对ASP.NET技术有全新的认识。读完这本书后, 你会发现原有的网站突然变得黯然失色。”

——Chris Carpenter, L3通信公司高级软件工程师

前 言

那是1996年的一个傍晚，天气很冷，还下着雨，我们坐在英国伯明翰的办公室里，望着IE 3呈现的空白页，我们想知道桌子底下老迈的Gateway 386服务器什么时候能处理完那个dbWeb查询并得到结果。就在我们焦急等待的时候，微软公司IIS部门的一个小组给我们发来一封电子邮件，问我们是否有兴趣看看一个名叫Denali的新产品（尚在开发中）。干嘛不呢？我们正在写一本关于网站和数据库集成的书，最后一章的案例研究正毫无头绪呢。

就这样，我们安装了Denali 0.9，开始玩上了。这东西给我们的印象太深刻了。大约晚上10点的样子，我们做出了一个决定：那本书的最后一章就要介绍这个相当棒的新技术。这就是几个月后公开发布的ASP（Active Server Pages）。那时我们还没有认识到，这件事居然会决定我们今后写作生涯的方向。

将当前的ASP.NET 2.0与ASP 1.0相比较，10年来的巨变令人感叹。在我们所写的关于ASP的第一本书中，谈到了因特网的历史，详尽地介绍了HTML 3.2和一个名为CSS的新样式语言，提供了一个全面的VBScript教程，对ASP的所有对象、方法和属性给出了描述和示例，还介绍了相关的数据库访问技术（称为ADO，ActiveX数据对象）。在长约1000页的篇幅中，我们甚至还讨论了SQL语言、使用MTS和MSMQ、构建COM组件以及一些案例研究，并包括大量附录。

现在的ASP.NET，单是描述服务器控件就需要1000页的篇幅，ASP.NET已经成长为一个相当成熟、丰富而且全面的语言技术，适合构建任何类型的Web应用。如果还像第一本ASP书那样，详细地介绍ASP.NET 2.0和相关.NET Framework类的所有方面，这可能要占满你书架中整整一行！

我们不打算这么做，而只是希望简洁而详尽地提供构建好网站和Web应用时所需的ASP.NET 2.0信息。在此，你能分享我们对ASP.NET的热情、多年来使用的经验，以及长期以来与微软公司小组保持的硕果累累的良好关系。希望你也能像我们一样，对ASP.NET饱含热情。

为什么需要使用本书

本书面向的是对Web、HTML、CSS、JavaScript和C#语言相当熟悉的开发人员。我们尽量不使用含糊或过于复杂的代码或概念，而是把重点放在ASP.NET 2.0的技术技巧上。

本书中的示例都在Visual Studio 2005中开发，不过如果你只能得到Visual Web Developer(VWD)，也可以使用VWD来开发。另外，你还需要访问一个数据库，可以随Visual Studio 2005和VWD安装的SQL Server 2005或SQL Server Express Edition。可以从我们的服务器

(<http://www.daveandal.net/books/8344>) 下载这些示例，还能在线运行其中很多示例。或者也可以从Addison-Wesley网站 (<http://www.awprofessional.com/msdotnetseries>) 获得这些示例代码。

致谢

要感谢微软公司向我們提供支持和帮助的所有人，不只是这本书，多年来你们一直都是这样支持我们。能与Scott Guthrie（ASP之父）共事让我们觉得很荣幸，正是他和他的小组把这个技术从一个基于页面模板的脚本语言发展成为如今这个丰富而高效的开发语言。当然，我们还要感谢Addison-Wesley的编辑部门、出版部门、市场部门和销售部门的所有人员。特别要感谢Appan Annamalai，本书第14章有关本地化的许多内容都是他的贡献。

关于作者

Alex Homer是一个热爱ASP.NET的计算机专家和Web开发人员。虽然他会花些时间做一点实际工作（做顾问或培训，有时还会参加一些会议），但大多数时间都用来“玩”最新的微软Web技术，然后撰写相关的图书。他生活在英国风景如画的德比郡戴尔斯，隐于山野，不问世事，只是通过因特网与外界保持联系。可以通过Alex的软件公司（Stonebroom Limited）与他联系，他的电子邮件地址是alex@stonebroom.com。

Dave Sussman曾经有份正式的工作，现在却再也不想出来工作了（想想就烦）。他大部分时间都投入到琢磨alpha和beta版的技术中，努力钻研这些技术，再把心得经验写出来，向大家讲解。他一直确信.NET是微软公司至今所发布的最好的技术，并对.NET的长足进步欣喜不已。像Alex一样，他也居住在一个宁静的小山村（有人说那里很古怪），不过他的房子还没有装修完（已经4年了）。可以通过Dave的公司（Ipona Limited）与他联系，他的电子邮件地址是dave@ipona.com。

目 录

第 1 章 技术与工具	1	2.1.8 使用母版页提供一致的外观	33
1.1 ASP.NET 2.0 的新特性	1	2.1.9 增加菜单和其他导航特性	35
1.1.1 外观一致的模板	1	2.2 小结	37
1.1.2 控件样式	3	第 3 章 数据源和数据显示控件	38
1.1.3 保护你的网站	4	3.1 数据库和连接	38
1.1.4 实现网站的个性化	6	3.1.1 自动关联数据库	38
1.1.5 创建门户	7	3.1.2 用户实例化	39
1.1.6 在网站中使用图像	9	3.1.3 连接串存放	40
1.1.7 在网站中使用数据	10	3.1.4 Visual Web Developer 和 Visual Studio 2005 数据库浏览器	41
1.1.8 国际化	11	3.2 数据源控件	41
1.2 相对于 1.x 的改变	12	3.2.1 SqlDataSource 控件	42
1.3 获得并安装 .NET 2.0	12	3.2.2 用 GridView 控件显示数据	42
1.3.1 与 1.x 一同运行	13	3.2.3 过滤和选择数据	43
1.3.2 安全账户	14	3.2.4 以声明方式选择数据	45
1.4 升级和部署应用	15	3.2.5 更新数据	46
1.5 安装示例	15	3.3 数据显示和编辑控件	49
1.6 小结	15	3.3.1 与数据源控件的数据绑定	49
第 2 章 真的就这么容易	16	3.3.2 定制 GridView 控件	50
2.1 开发人员的一天	16	3.3.3 BoundField 控件	52
2.1.1 使用数据源控件和 GridView 显示数据	17	3.3.4 ButtonField 控件	53
2.1.2 支持行的排序和分页	19	3.3.5 CheckBoxField 控件	54
2.1.3 提供行编辑特性	21	3.3.6 HyperLinkField 控件	55
2.1.4 增加过滤功能来选择特定的行集	22	3.3.7 使用 DataNavigateUrlFields 属性	56
2.1.5 在表单中显示一行进行编辑	24	3.3.8 ImageField 控件	57
2.1.6 处理通过业务对象提供的数据	28	3.3.9 DetailsView 控件	64
2.1.7 绑定数据来减少数据库访问	31	3.3.10 FormView 控件	68
		3.4 小结	70

第4章 业务对象的数据绑定	72	5.6.1 Visual Studio 2005 中 创建 UDT	118
4.1 两层和三层体系结构	72	5.6.2 在 SQL Server 2005 中 部署 UDT	119
4.1.1 业务层、数据层, 还是二者 都有	73	5.6.3 创建包含 UDT 列的表	120
4.1.2 对象数据映射器	73	5.6.4 在 UDT 列中插入数据	121
4.1.3 创建数据层	74	5.6.5 访问 UDT 列	121
4.2 ObjectDataSource 控件	79	5.6.6 补充读物	123
4.2.1 使用 ObjectDataSource 控件	81	5.7 小结	123
4.2.2 获得新插入行的标识值	83	第6章 数据和输出缓存	124
4.3 强类型数据集	84	6.1 Application、Session、Http- Context 和 ViewState 缓存	124
4.4 小结	91	6.1.1 使用应用状态	124
第5章 高级数据和页面技术	92	6.1.2 使用会话状态	125
5.1 DataSource 控件事件	92	6.1.3 使用 HttpContext	128
5.1.1 命令执行前修改参数	93	6.1.4 使用视图状态	129
5.1.2 更新之后同步 DataSource 控件	93	6.2 输出缓存	129
5.1.3 命令执行后访问输出参数	94	6.2.1 配置输出缓存	130
5.1.4 ObjectDataSource 控件的 输出参数	96	6.2.2 缓存部分页面	131
5.1.5 利用 ObjectDataSource 定制 分页	97	6.2.3 配置缓存	134
5.2 GridView 和 DetailsView 事件	100	6.3 数据缓存	135
5.3 处理数据编辑冲突	101	6.3.1 SQL Server 2005 的 SQL 缓存 失效	136
5.4 异步页面和数据	105	6.3.2 SQL Server 2000 的 SQL 缓存 失效	142
5.4.1 异步页面	106	6.4 缓存 API	146
5.4.2 异步任务	111	6.4.1 让缓存中的项到期	147
5.5 SQL Server 2005 用户自定义类型	112	6.4.2 使缓存条目依赖于外部因素	148
5.5.1 UDT 属性	114	6.4.3 多个缓存依赖关系	149
5.5.2 可串行化类型	115	6.4.4 缓存中的项到期时得到通知	149
5.5.3 构造函数	115	6.5 何时使用缓存	151
5.5.4 解析值	116	6.6 小结	151
5.5.5 输出值	117	第7章 处理 XML	153
5.5.6 处理 Null 值	117	7.1 Visual Studio 2005 中处理 XML	153
5.5.7 增加属性	117	7.2 数据绑定与显示 XML 数据	155
5.6 对 UDT 的补充	118	7.2.1 手动地绑定 XML 文件	155

7.2.2 使用 XmlDataSource 控件.....	155	8.2.2 ASP.NET 事件驱动体系结构.....	213
7.2.3 指定 TreeView 绑定.....	157	8.2.3 请求和响应信息.....	213
7.2.4 绑定表达式.....	158	8.3 小结.....	214
7.2.5 应用选择.....	159	第 9 章 Page 类和母版页.....	215
7.2.6 使用绑定表达式和选择.....	160	9.1 HttpRequest、HttpResponse、 HttpServerUtility 和 HttpContext 类.....	215
7.3 转换 XML.....	161	9.1.1 使用 HttpRequest 类.....	216
7.3.1 Visual Studio 2005 中调试 XSLT.....	163	9.1.2 使用 HttpResponse 类.....	216
7.3.2 利用 XmlDataSource 控件 转换.....	164	9.1.3 使用 HttpServerUtility 类.....	220
7.3.3 通过代码完成转换.....	164	9.1.4 使用 HttpContext 类.....	222
7.4 处理原始 XML.....	165	9.2 ASP.NET Page 类.....	222
7.4.1 流化 XML.....	166	9.2.1 Page 类的通用方法和属性.....	223
7.4.2 处理内存中的 XML 文档.....	169	9.2.2 访问内部 ASP.NET 对象.....	225
7.5 XML 与关系数据的转换.....	172	9.2.3 查找页面上的控件.....	226
7.6 处理 SQL Server 2005 XML 数据.....	173	9.2.4 写跟踪信息.....	230
7.6.1 创建 XML 列.....	174	9.2.5 皮肤和主题.....	230
7.6.2 创建模式.....	174	9.2.6 验证.....	230
7.6.3 在 XML 列中插入数据.....	175	9.2.7 页面级事件.....	230
7.6.4 从 ADO.NET 访问 XML 列.....	175	9.2.8 客户端脚本特性.....	233
7.7 小结.....	178	9.2.9 创建客户端脚本段.....	234
第 8 章 构建交互式 Web 表单.....	179	9.2.10 异步页面回调.....	235
8.1 ASP.NET 控件集.....	179	9.3 派生 Page 子类对象.....	238
8.1.1 标准 HTML 服务器控件.....	180	9.4 处理母版页.....	239
8.1.2 文本和图像显示控件.....	180	9.4.1 简单母版页.....	240
8.1.3 超链接和导航控件.....	185	9.4.2 嵌套母版页.....	241
8.1.4 输入和表单控件.....	190	9.4.3 动态设置母版页.....	241
8.1.5 ValidationSummary 控件.....	196	9.4.4 访问母版页中的值和控件.....	243
8.1.6 在代码中访问验证控件.....	196	9.5 小结.....	244
8.1.7 列表和表格控件.....	198	第 10 章 网站导航技术.....	246
8.1.8 富控件.....	204	10.1 利用超链接的简单导航.....	246
8.1.9 容器控件.....	210	10.2 利用 LinkButton 控件实现导航.....	249
8.1.10 移动控件.....	210	10.3 通过浏览器重定向实现导航.....	255
8.1.11 布局方法——表和样式表.....	211	10.4 通过服务器端重定向实现导航.....	257
8.1.12 选择适当的控件类型.....	212	10.4.1 HttpServerUtility. Transfer 方法.....	257
8.2 ASP.NET 页面和回送模型.....	212		
8.2.1 区别回送和导航.....	212		

10.4.2	HttpServerUtility. Execute 方法	258	12.1.1	指定概要和个性化提供者	319
10.4.3	使用 Transfer 和 Execute 方法	258	12.1.2	创建和修改 ASP.NET 应用 数据库	321
10.5	表单内容的跨页面提交	263	12.1.3	启用概要和声明概要属性	323
10.6	站点图、菜单和导航路径控件	266	12.2	存储和使用动态概要数据	328
10.6.1	站点图提供者	266	12.2.1	读取概要数据	328
10.6.2	站点图配置文件	266	12.2.2	存储(写)概要数据	329
10.6.3	使用站点图文件	268	12.3	使用购物车的简单例子	330
10.6.4	深入站点图	270	12.3.1	设置文本大小	331
10.7	小结	274	12.3.2	显示购物车和增加商品	332
第 11 章	安全与成员	275	12.3.3	显示登录用户的详细信息	334
11.1	防止匿名访问网站	275	12.3.4	更新概要属性	336
11.1.1	ASP.NET 配置文件	276	12.3.5	移植匿名用户	336
11.1.2	IIS 和 ASP.NET 中的安全和 访问控制设置	277	12.3.6	Profile 类的其他特性	339
11.1.3	ASP.NET 认证与授权	279	12.4	使用主题实现网站个性化	340
11.2	ASP.NET 成员提供者与角色管理器	283	12.4.1	什么是主题和皮肤	340
11.2.1	ASP.NET 应用数据库	284	12.4.2	创建主题和皮肤	341
11.2.2	ASP.NET 成员提供者配置	284	12.4.3	应用主题和皮肤	343
11.2.3	ASP.NET 角色管理配置	288	12.4.4	避免使用主题和皮肤	345
11.3	ASP.NET 网站管理工具	289	12.4.5	使用主题的一个例子	346
11.3.1	创建 ASP.NET 应用数据库	289	12.5	小结	350
11.3.2	排除 ASP.NET 应用数据库 问题	291	第 13 章	构建 Web 门户	351
11.3.3	创建和管理用户和角色	292	13.1	ASP.NET 2.0 门户框架	351
11.4	ASP.NET 安全服务器控件	309	13.1.1	门户框架的目标	351
11.4.1	可用控件集	309	13.1.2	关于 WebPart 框架	352
11.4.2	使用安全服务器控件	309	13.2	WebPart 管理器	354
11.5	在代码中访问成员和角色特性	314	13.3	WebPart 和授权	356
11.5.1	Membership 类	314	13.4	区域和 WebPart	357
11.5.2	MembershipUser 类	316	13.4.1	WebPart	358
11.5.3	Roles 类	317	13.4.2	实现 WebPart	360
11.6	小结	318	13.4.3	作为用户控件的 WebPart	360
第 12 章	概要、个性化和主题	319	13.4.4	作为定制控件的 WebPart	362
12.1	配置概要和个性化	319	13.4.5	WebPart Chrome 和样式	363
			13.4.6	WebPart Verb	364
			13.5	编目区域	365
			13.6	编辑器区域	366

13.6.1 AppearanceEditorPart	367	第 15 章 使用 Framework 类	430
13.6.2 BehaviorEditorPart	367	15.1 一个示例应用	430
13.6.3 LayoutEditorPart	369	15.1.1 在你的机器上配置示例	432
13.6.4 PropertyGridEditorPart	369	15.1.2 示例应用的用户界面	433
13.7 将 WebPart 连接在一起	370	15.2 利用 System.Collections 类存储 数据	436
13.7.1 实现 WebPart 连接	370	15.2.1 ArrayList	437
13.7.2 连接 WebPart	372	15.2.2 栈和队列	438
13.7.3 用户发起的连接	373	15.2.3 有序表、HashTable 和基于 字典的集合	438
13.7.4 连接到母版页中的 WebPart	375	15.2.4 如何返回值	440
13.7.5 转换器	376	15.2.5 串行化集合	440
13.8 小结	377	15.2.6 泛型集合	441
第 14 章 可用性、可访问性、移动设备 和本地化	379	15.3 利用 System.IO 类读、写和压缩 数据	443
14.1 页面验证、浏览器功能和替代内容	379	15.3.1 使用 Stream、StreamReader 和 StreamWriter	443
14.1.1 验证页面内容	380	15.3.2 读取驱动器、文件夹和文件 信息	445
14.1.2 检测浏览器功能	382	15.3.3 压缩数据	452
14.1.3 CSS 支持的特异性	386	15.4 利用 System.Drawing 类创建图形 和图像	454
14.2 使用页面更易于使用	387	15.5 利用 System.Net 类访问因特网	457
14.2.1 整体网站和页面设计	387	15.5.1 获取网页	457
14.2.2 控件、标题、快捷键和跳格 顺序	388	15.5.2 完成 DNS 查找	459
14.3 支持残疾用户和专用用户代理	389	15.6 利用 System.Net.Mail 类发送电子 邮件	460
14.3.1 最大化可访问性的通用技术	389	15.6.1 发送文本和 HTML 电子邮件	460
14.3.2 ASP.NET 控件中的可访问性 特性	392	15.6.2 发送带文本附件的电子邮件	462
14.4 为小屏幕和移动设备构建页面	408	15.6.3 发送有二进制附件的电子 邮件	463
14.4.1 页面设计和设备支持	408	15.7 利用 System.DirectoryServices 类访问活动目录	467
14.4.2 为移动设备创建页面	408	15.8 利用 System.Security. Cryptography 类加密数据	469
14.4.3 移动浏览器模拟器	416	15.9 小结	473
14.5 支持本地化和多种语言	417		
14.5.1 无代码实现本地化	417		
14.5.2 用户可选择的本地化	424		
14.5.3 通过编程访问资源	427		
14.5.4 可扩展性	427		
14.5.5 Web 资源	428		
14.6 小结	429		

第 16 章 Web 服务	474	16.3.2 处理错误	484
16.1 Web 服务	474	16.4 控制数据在 Web 服务中如何传输	484
16.2 创建 Web 服务	475	16.4.1 定制串行化	485
16.2.1 Web 服务性质	476	16.4.2 串行化集合	487
16.2.2 测试 Web 服务	478	16.4.3 手动地串行化	489
16.2.3 缓存和状态	480	16.5 Web 服务提升	490
16.2.4 创建异步 Web 服务	481	16.6 Web 服务的将来	495
16.3 使用 Web 服务	481	16.7 小结	495
16.3.1 异步调用 Web 服务	482		



2000年7月微软公司发布.NET Framework 1.0技术预览时,形势立刻明朗了,Web开发将发生变化。那时微软公司的最新技术 ASP 3.0功能相当强大,也很灵活,利用ASP可以很容易地创建动态网站。ASP催生了一系列相关的图书、文章、网站和组件,所有这些都使得开发过程更加容易。不过,ASP缺少一个应用框架,要作为一个企业开发工具它是绝对不能胜任的。ASP里所做的一切都面向代码,如果不写代码,什么也做不了。

设计ASP.NET就是为了解决这个问题,其核心目标之一是减少所要创建的代码量,从而能更快、更容易地编程。ASP.NET引入了声明式编程模型、丰富的服务器控件层次体系(支持事件)、庞大的类库,还提供了开发工具支持,不仅支持低级的记事本(Notepad),还支持高端的Visual Studio .NET。总而言之,ASP.NET向前迈出了很大的一步。

现在,随着2.0版的发布,我们看到ASP.NET又向前跃出了一大步,进一步提高了开发人员的生产力。ASP.NET的特性集更加丰富,增加了更多服务器控件和API包装功能,从而可以减少需要编写的代码量。工具集也得到改善,因为Visual Studio 2005拥有更多特性和一个更好的设计工具,另外对标准和验证提供了更好的支持。除了Visual Studio 2005外,还有一个Visual Web Developer Express Edition,这是Visual Studio的一个缩减版(不过缩减得不多),能提供同样优秀的开发环境,而且没有比它更低廉的价格了——它是免费的!

1.1 ASP.NET 2.0 的新特性

这一章并不打算深入介绍任何一个特定的特性,我们只是想让你对ASP.NET 2.0有所了解,并认识到利用ASP.NET 2.0能让Web开发更加容易。在以下新特性的概述中,将通过一些粗略的场景来分别描述。

1.1.1 外观一致的模板

ASP.NET 1.x提供了开发网站的一种简便方法,但是有一点很显然,它缺乏一种架构来应用一致的外观样式。为此产生了众多解决技术:

- 创建一个定制的对象(继承自Page),由这个定制页面预加载控件。
- 创建一个模板服务器控件,在这里模板为各个页面提供布局划分,然后在每个页面上使用这个控件。

□ 在网站的公共区使用用户控件，如标题、菜单和页脚。

在上述方法中，前两种方法要求开发人员懂得如何创建服务器控件，尽管大多数ASP.NET开发人员都能掌握这个用法，但是他们不一定都有经验。因此，最好避免使用定制服务器控件的方案。最后一种选择是一个简单的解决方案，很容易创建和实现。创建用户控件是为了提供可重用的功能，这也是用户控件一个很好的用途。不过，要应用一致的外观样式，需要先在各个页面上放置这些用户控件，然后确保它们放在每一个页面中的同一个位置上。换句话说，你真正需要的是一个页面模板，实际上它就是一个ASP.NET文件，只需为每个新页面复制这个文件就行了。但是这种方法存在一个危险，实际中太容易修改某一个页面，改动其布局。

为了提供一个模板解决方案，ASP.NET 2.0提出了母版页（Master Page）的概念，就是为页面的外观样式提供一个模板。母版页是为其他页面提供模板的ASP.NET页面，它提供了公共的页面级布局和功能。母版页为内容定义了占位符，这些占位符可以由子页覆盖。得到的结果页面将是母版页与子页的综合，如图1-1所示。

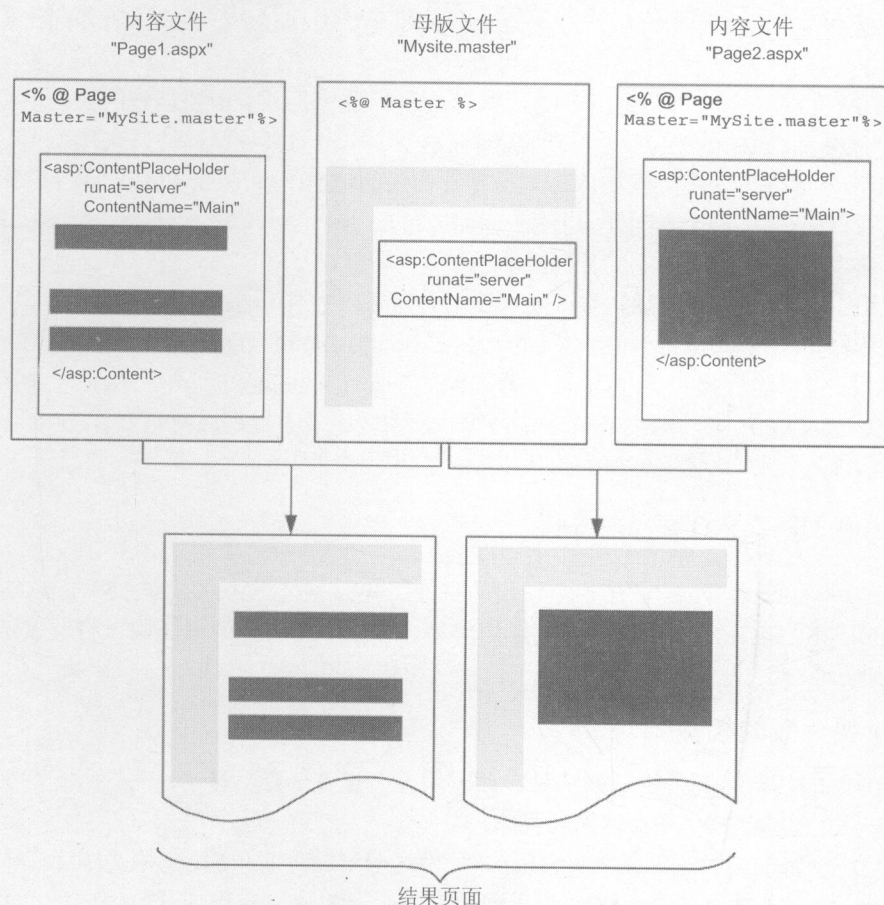


图1-1 合并母版页和子页

母版页将在第9章介绍。

1.1.2 控件样式

在处理网站外观样式方面，ASP.NET 2.0的第二个主要特性是主题（theme）。现在换主题或换“皮肤”很流行，允许用户为应用创建定制外观。Windows桌面应用中，最流行的两个可以换主题的应用是音频播放器（Winamp和Windows Media Player）；利用另外一些软件，甚至Windows XP也可以换主题。

换主题之所以变得如此流行，这是由人的天性决定的。一个东西的外观如何，我们总是想有自己的选择，希望展现自己的个性。这在桌面应用中很容易做到，因为用户通常各自拥有一台机器。不过，对于网站，换主题就要困难得多，因为用户数量太多了。要跟踪哪个用户有哪个主题并管理这些主题，这很让人头疼，网站管理员都不想身陷其中。

有些网站提供了某些形式的主题，但是它们在定制化方面相对有限，只允许选择色彩方案。还有一些网站提供了一些样式表供用户选择（假设用户的浏览器支持这个特性），或者是在服务器上更改样式表。这样不仅允许选择色彩方案，还可以选择完整的样式，包括字体、边框样式，等等。

在ASP.NET 2.0中，主题的目标相当简单。

- ❑ 可以使用开发页面本身所用的设计工具和方法来定制网站或页面的外观。这说明，要向网站增加主题，没有必要去学习任何特殊的工具或技术。
- ❑ 可以把主题应用到控件、页面，甚至整个网站。例如，用户可以定制网站的某些部分，而确保其他部分（如公司标识）不能定制。
- ❑ 可以定制所有可见属性，确保在换主题时，页面和控件能得到一致的样式。

ASP.NET 2.0中，这个实现围绕着两个方面：皮肤和主题。**皮肤（skin）**是可以应用到控件的一组属性和模板。**主题（theme）**是一组皮肤和所有其他相关的文件（如图像或样式表）。皮肤特定于控件，所以，对于一个给定的主题，可能该主题中的各个控件分别有一个单独的皮肤。如果控件没有皮肤，则继承默认外观。主题有两类。

- ❑ 定制主题将覆盖控件定义，因此会改变控件的外观样式。应用定制主题要使用Page指令的Theme属性。
- ❑ 样式表主题不覆盖控件定义，因此控件可以使用主题属性，也可以覆盖主题属性。应用样式表主题要使用Page指令的StylesheetTheme属性。

这个实现很简单，因为皮肤与应用该皮肤的服务器控件使用同样的定义，而主题只是应用根目录下一个目录中的一组文件。例如，考虑以下示例目录结构：

```
default.aspx
\Themes
  \MyTheme
    MySkin.skin
  \YourTheme
    YourSkin.skin
```


每个主题由Themes目录下的一个目录组成。在各个主题中，有一个带.skin后缀的文件，其中包含该主题的皮肤信息。例如，MySkin.skin可能包含以下定义：

```
<asp:Label SkinID="Normal" runat="server"
  Font-Bold="True" BackColor="#FFC080" />
<asp:Label SkinID="Comic" runat="server"
  Font-Italic="True" Font-Names="Comic Sans MS" />
```

这里为Label控件定义了两个皮肤，分别有不同的外观属性。通过设置适当的页面级属性可以选择这个主题，皮肤可以通过设置一个控件级属性来选择，如以下代码段所示：

```
<%@ Page Theme="MyTheme" %>

<form runat="server">

  <asp:Label SkinID="Comic" Text="A Label" />

</form>

或

<%@ Page StylesheetTheme="MyTheme" %>

<form runat="server">

  <asp:Label SkinID="Comic" Text="A Label" />

</form>
```

页面级属性和控件级属性都可以在运行时设置，也可以在设计时设置，这就提供了一个极其强大的解决方案，特别是结合新增的个性化特性时，功能更为强大。

个性化和主题将在第12章介绍。

1.1.3 保护你的网站

随着越来越多的企业开展网上业务，安全性越发显得重要。不仅仅是保护像信用卡号之类的机密信息，用户的个人信息和喜好也需要保护。因此，你的网站中必须建有相应特性来认证用户。在ASP.NET 1.x中，这很容易做到，不过还是需要编写代码。为保证安全性，需要选择你最喜欢的安全机制（最常见的就是表单认证），然后在页面中增加控件来提供登录信息，如用户名、口令、“记住用户名”复选框，等等。没有内置的机制来存储个人信息，所以需要你自己动手来编写。

有了ASP.NET 2.0，我们就可以解脱了。它不仅可以保证安全，而且无需编写很多代码。对于登录功能，现在提供了以下特性：

- Login控件，可以为网站提供完备的登录功能。
- LoginStatus控件，指示登录状态，可以配置为提供登录和注销页面的自动链接。
- LoginName控件，显示当前（或匿名）登录名。
- LoginView控件，根据登录状态提供模板视图。