

数据库应用开发技术丛书

ASP.NET

数据库高级教程 (C#篇)

李应伟
姚素霞 编著
景 丽



清华大学出版社

数据库应用开发技术丛书

ASP.NET 数据库 高级教程(C#篇)

李应伟 姚素霞 景丽 编著

清华大学出版社

北 京

前 言

Visual Studio.NET 是 Microsoft 公司推出的可视化开发工具，ASP.NET 作为 Visual Studio.NET 的组成部分之一，已经成为 Internet 中 Web 应用程序的新一代开发工具，并逐渐被广大程序员普遍采用。

在微软的 .NET 战略中，ASP.NET 是非常重要的一环，它相对于以前的 ASP 有了相当大的改进。相对于其他的 Web 应用开发模型来讲，ASP.NET 具有更大的优势，其主要特点包括：

- ASP.NET 与其前版 ASP 不同，它是在服务器上运行的编译好的公共语言运行时代码，可以更好地提高程序运行性能。
- ASP.NET 与语言无关，它可以采用 C#、VB.NET 以及 JScript 等支持 .NET 框架的语言来进行开发，开发者可以根据自身的情况进行选择。
- ASP.NET 支持开发 Web 服务，它对 XML 技术提供了更好的支持，使得 Web 应用程序的开发更具有可扩展性和跨平台性。
- ASP.NET 提供了很多功能强大的服务器端控件，使得程序的开发更趋于简单化。同时，它还提供了 HTML 设计代码和后台编程代码分离的技术，并在后台代码开发中提供了强大的智能化支持，使得开发工作更具有条理性。

C# 是一门崭新的语言，它具有开发效率高，应用范围广等特点，成为当前程序开发领域的一大热门。本书在开发 ASP.NET 应用程序的后台代码时就选用了 C# 语言，相信其优秀的开发性能一定能给读者带来耳目一新的感觉。

本书主要介绍 ASP.NET 的高级技术，尽可能帮助开发人员解决实际开发项目中遇到的问题。它主要适用于 Web 应用开发初学者及广大网络设计和开发人员阅读，对高级开发人员也有一定的参考价值。

全书共分 20 章，第 1~11 章主要对 ASP.NET 高级技术的各个方面分别进行说明和解析，并附以实例指导；第 12~20 章则通过一些详实的开发程序来帮助读者深入理解 ASP.NET，以及学习如何利用 ASP.NET 来构建功能强大的 Web 应用程序。

第 1 章介绍 ASP.NET 中 Global.asax 文件。

第 2 章介绍应用程序配置，主要对 web.config 文件的使用进行了详细的说明。

第 3~5 章详细讲解 ASP.NET 中的数据访问技术，使得开发数据驱动的网站不再那么神秘。

第 6 章介绍了 XML 在 ASP.NET 中的使用，反映了 ASP.NET 对 XML 技术的支持。

第 7 章主要介绍 Web 服务，对 ASP.NET 中如何开发和使用 Web 服务进行了详细的说明。

第 8 章介绍 Web 窗体控件, 自定义 Web 窗体控件等内容, 说明控件在 Web 程序开发中的作用。

第 9 章介绍组件服务。

第 10 章和第 11 章分别涉及 ASP.NET 中的安全性与性能问题, 可以用于开发更为高效、安全的 Web 应用。

第 12~20 章则通过几个大型的实例来分别介绍网站广告、图片处理、网络硬盘、邮件发送、统计图表、即时消息、BBS 系统的 Web 应用解决方案, 对前面介绍的内容进行了一些综合的应用。

参加本书编写工作的人员有郭斌、李应伟、姚素霞、李翔、韦敏宗、田龙、钟遥、宋明颢和王微等。其中郭斌完成第 1、2、3、7、9、16 章, 李翔完成第 5、11、12、18、19 章, 韦敏宗完成第 6、10、14、17 章, 田龙完成第 4、15 章, 钟遥完成第 8、20 章, 宋明颢完成第 13 章, 王微完成了附录部分的写作工作。在写作的过程中大家都付出了相当多的努力, 发扬了锐意进取的团队精神。

此外, 蓝荣香、王昊亮、喻波、马天一、魏勇、郝荣福、李光龙、孙明、李大宇、武思宇、牟博超、付鹏程、高翔、朱丽云、崔凌、张巧玲、李辉、李欣、柏宇、郭强、金春范、程梅、黄霆、钟华、高海峰、王建胜、张浩、刘湘和邵蕴秋等同志在整理材料方面给予了作者很大的帮助, 在此, 对他们表示衷心的感谢。

但由于作者水平和经验有限, 书中难免有不足之处, 希望通过和广大的读者进行交流来解决, 以使得本书在再版时更为完美。

作 者

2004 年 3 月

目 录

第 1 章 ASP.NET应用程序设置	1
1.1 Global.asax 概述	1
1.2 ASP.NET 应用程序指令	6
1.2.1 @ Application 指令	6
1.2.2 @ Import 指令	7
1.2.3 @ Assembly 指令	8
1.3 Application 对象.....	8
1.4 Session 对象	13
1.5 脚本块.....	18
1.6 服务器端脚本标记	21
1.7 小结	21
第 2 章 ASP.NET应用程序配置	22
2.1 概述	22
2.2 配置文件格式.....	23
2.3 配置节.....	27
2.3.1 <configuration>节.....	27
2.3.2 <configSections>节	27
2.3.3 <appSettings>节.....	28
2.3.4 <compilation>节	29
2.3.5 <customErrors>节.....	29
2.3.6 <globalization>节	30
2.3.7 <sessionState>节.....	30
2.3.8 <trace>节.....	31
2.3.9 <authentication>节.....	31
2.4 使用位置和路径.....	36
2.5 扩展配置文件.....	37
2.5.1 扩展应用程序配置信息.....	37
2.5.2 扩展自定义的配置信息.....	38
2.6 使用配置文件.....	39
2.6.1 访问<browserCaps>节	39

2.6.2	访问扩展应用程序配置信息	40
2.6.3	访问自定义的配置信息	42
2.7	小结	43
第 3 章	ASP.NET数据库访问——ADO.NET	44
3.1	概述	44
3.1.1	ADO.NET 和 ADO	44
3.1.2	ADO.NET 特性	45
3.2	Connection 对象	46
3.2.1	Connection 对象概述	46
3.2.2	Connection 对象实例	46
3.3	Command 对象	47
3.3.1	Command 对象概述	47
3.3.2	Command 对象实例	48
3.4	DataReader 对象	51
3.4.1	DataReader 对象概述	51
3.4.2	DataReader 对象实例	52
3.5	DataAdapter 对象	52
3.5.1	DataAdapter 对象概述	52
3.5.2	DataAdapter 对象命令	53
3.6	DataSet 对象	53
3.6.1	DataSet 对象概述	53
3.6.2	DataSet 对象实例	54
3.7	小结	58
第 4 章	ASP.NET其他数据访问	59
4.1	概述	59
4.2	使用 ODBC.NET	59
4.2.1	ODBC.NET 简介	60
4.2.2	设置 ODBC.NET 数据源	61
4.2.3	访问 ODBC 数据源	62
4.3	使用 .NET Jet Driver	63
4.3.1	.NET Jet Driver 简介	63
4.3.2	Microsoft ODBC 桌面数据库驱动器简介	63
4.3.3	通过 Microsoft Jet 的 OLE 数据库提供者访问 Microsoft Jet 数据库	64
4.3.4	使用 ODBC 驱动器访问数据库	65

4.3.5	使用 .NET Jet Driver For ACCESS 访问 MS ACCESS 数据库的实例	72
4.4	访问 MySQL	79
4.4.1	MySQL 简介	79
4.4.2	访问 MySQL 数据库	80
4.5	访问 Oracle	81
4.5.1	Oracle 简介	81
4.5.2	Oracle 的 ODBC 驱动器简介	82
4.5.3	访问 Oracle 数据库	82
4.6	访问 Excel	84
4.6.1	Excel 简介	84
4.6.2	访问 Excel 数据库	84
4.6.3	连接 Excel 数据源	85
4.7	访问 txt 文件	86
4.7.1	使用 ODBC 数据源	86
4.7.2	使用 System.IO 命名空间	87
4.8	小结	88
第 5 章	数据访问控件和自定义	89
5.1	概述	89
5.2	Repeater 控件	89
5.2.1	ItemTemplate 模板	90
5.2.2	AlternatingItemTemplate 模板	91
5.2.3	SeparatorTemplate 模板	93
5.2.4	HeaderTemplate 模板	94
5.2.5	FooterTemplate 模板	95
5.3	DataList 控件	95
5.3.1	SelectedItemTemplate 模板	96
5.3.2	EditItemTemplate 模板	98
5.4	处理控件的事件	100
5.4.1	Load 事件	100
5.4.2	ItemCreated 事件	103
5.4.3	ItemDataBound 事件	104
5.4.4	ItemCommand 事件	105
5.5	检索控件	106
5.5.1	界面设计	106
5.5.2	代码实现	112
5.6	小结	115

第 6 章 在 ASP.NET 中使用 XML	116
6.1 读写 XML	116
6.1.1 Xml 控件	116
6.1.2 XmlTextReader	119
6.1.3 XmlTextWriter	123
6.1.4 XmlDocument(W3C DOM)	125
6.2 XML 串行化	129
6.2.1 XmlSerializer	129
6.2.2 基本串行化	130
6.2.3 定制串行化	131
6.2.4 将 XML 映像到对象	132
6.2.5 将 XML 数据反串行化成对象	134
6.3 XML 的 XSL 转换	136
6.3.1 程序实例一	136
6.3.2 程序实例二	139
6.4 MSXML	140
6.5 XML 使用范例	141
6.6 小结	145
第 7 章 ASP.NET Web 服务	147
7.1 Web 服务概述	147
7.1.1 Web 服务技术架构	147
7.1.2 Web 服务体系结构	148
7.1.3 Web 服务协议集	149
7.1.4 对 Web 服务的深层理解	150
7.1.5 Web 服务带来的机遇	150
7.2 创建概述	151
7.2.1 新建 Web 服务工程	151
7.2.2 Web 服务创建	153
7.3 在 ASP.NET 中使用 Web 服务	156
7.4 自定义 SOAP	161
7.4.1 SOAP 扩展	162
7.4.2 自定义 SOAP 消息	162
7.4.3 传递复杂数据	163
7.5 异步化 Web 服务	164
7.6 小结	165

第 8 章 Web窗体控件自定义	166
8.1 Web 窗体控件概述	166
8.1.1 System.Web.UI.Control 类	168
8.1.2 System.Web.UI.WebControls.WebControl 类	172
8.1.3 System.Web.UI.HtmlControls.HtmlControl 类	177
8.2 创建 Web 窗体控件	180
8.2.1 用户控件	181
8.2.2 自定义服务器控件	182
8.3 公布 Web 窗体控件属性	184
8.4 封装 Web 窗体控件事件	185
8.5 使用 Web 窗体控件	186
8.6 小结	188
第 9 章 ASP.NET应用程序安全性	189
9.1 安全性概述	189
9.1.1 Web 应用程序的安全问题	189
9.1.2 Web 应用程序的“脆弱性”	190
9.2 ASP.NET 身份验证体系	191
9.2.1 身份验证	192
9.2.2 用户授权	193
9.2.3 模拟	194
9.3 使用 IP 级安全控制	195
9.4 使用 Windows 身份验证	198
9.4.1 基本身份验证	198
9.4.2 简要身份验证	199
9.4.3 集成 Winsows 身份验证	200
9.4.4 实现 Windows 身份验证	200
9.5 使用窗体验证	202
9.5.1 基于窗体验证的原理	202
9.5.2 配置窗体验证	203
9.5.3 一个窗体验证实例	204
9.6 Web 服务安全性	208
9.6.1 在 Web 服务中实现身份验证	209
9.6.2 自定义 SOAP 身份验证	212
9.7 小结	215

第 10 章	使用组件服务	216
10.1	组件服务概述	216
10.2	使用 .NET 组件	217
10.2.1	创建业务对象	217
10.2.2	使用业务对象	221
10.3	使用 COM 组件	223
10.4	小结	224
第 11 章	ASP.NET 应用程序性能优化	225
11.1	性能概述	225
11.2	性能优化方法	226
11.2.1	使用会话状态	226
11.2.2	使用 Page.IsPostBack	228
11.2.3	使用服务器控件	228
11.2.4	字符串操作	229
11.2.5	数据访问	231
11.3	性能测量	234
11.3.1	使用 WAS 测试网站性能	234
11.3.2	使用 ACT 测试网站性能	240
11.3.3	使用性能计数器监测网站性能	242
11.4	小结	244
第 12 章	网站广告	245
12.1	概述	245
12.2	问题分析	246
12.3	方案设计	246
12.3.1	使用 AdRotator	246
12.3.2	使用 XML	247
12.3.3	使用 DataSet	250
12.4	程序实现	253
12.4.1	界面设计	253
12.4.2	代码实现	257
12.5	小结	263
第 13 章	网络日历	264
13.1	概述	264
13.2	使用日历组件	265
13.2.1	使用 Calendar 属性	265

13.2.2	使用子标记	267
13.2.3	一个实例	268
13.3	使用数据库	269
13.4	增加记事功能	270
13.5	增加密码验证	271
13.6	定制日历	274
13.7	小结	276
第 14 章	图片处理及显示	277
14.1	概述	277
14.1.1	基本原理	277
14.1.2	界面设计	277
14.2	图片上传处理	280
14.3	图片管理	281
14.4	使用数据库	285
14.4.1	数据库建立	285
14.4.2	使用数据库	289
14.5	图片显示	292
14.5.1	分类显示	292
14.5.2	相册浏览	298
14.6	小结	308
第 15 章	网站在线管理	309
15.1	概述	309
15.2	在线文件管理	310
15.2.1	新工程创建	310
15.2.2	界面设计	311
15.2.3	代码编写	313
15.3	数据库管理	320
15.3.1	界面设计	320
15.3.2	显示数据库文件	322
15.3.3	在线修改数据	327
15.3.4	在线删除数据	329
15.3.5	数据库文件的管理	330
15.4	系统安全性	330
15.4.1	代码访问安全性	331
15.4.2	使用 SSL 进行加密和签名	331

15.4.3	身份验证	332
15.5	小结	333
第 16 章	网络硬盘	334
16.1	概述	334
16.2	File 类和 Directory 类	335
16.2.1	System.IO.File 类和 System.IO.FileInfo 类	335
16.2.2	System.IO.Directory 类和 System.DirectoryInfo 类	338
16.3	查看文件夹内容	341
16.3.1	新工程创建	341
16.3.2	主界面设计	342
16.3.3	代码实现	344
16.4	在创建新文件夹同时设置访问权限	346
16.4.1	界面布置	346
16.4.2	代码实现	347
16.5	上传文件到指定文件夹	348
16.5.1	界面布置	348
16.5.2	代码实现	348
16.6	下载文件到本机或在线查看文件内容	349
16.6.1	界面布置	350
16.6.2	代码实现	350
16.7	删除文件或文件夹	351
16.7.1	界面布置	351
16.7.2	代码实现	351
16.8	小结	352
第 17 章	发送邮件	353
17.1	概述	353
17.2	使用 SMTP 发送邮件	353
17.2.1	SMTP 协议的通讯模型	353
17.2.2	SMTP 协议的命令和应答	354
17.2.3	在应用程序中使用 SMTP 协议	356
17.3	使用 Socket 发送邮件	357
17.3.1	用 Socket 套接字为 SMTP 提供网络通讯基础	357
17.3.2	SMTP 会话应答的实现	358
17.4	增加附件	362
17.5	小结	368

第 18 章 统计图表	369
18.1 概述.....	369
18.2 方案设计.....	370
18.2.1 使用 GDI+.....	370
18.2.2 使用 Office Web Components.....	373
18.3 公司盈利状况统计.....	378
18.3.1 数据库设计.....	378
18.3.2 界面设计.....	379
18.3.3 代码实现.....	380
18.4 公司收入分块图.....	384
18.4.1 数据库设计.....	385
18.4.2 界面设计.....	385
18.4.3 代码实现.....	386
18.5 报表输出.....	389
18.6 小结.....	390
第 19 章 即时信息	391
19.1 概述.....	391
19.2 问题分析.....	392
19.3 方案设计.....	392
19.3.1 数据库设计.....	392
19.3.2 实现即时信息.....	394
19.4 显示在线人数的即时信息.....	395
19.4.1 数据库设计.....	395
19.4.2 界面设计.....	397
19.4.3 代码实现.....	402
19.5 即时股价信息.....	412
19.5.1 数据库设计.....	413
19.5.2 界面设计.....	414
19.5.3 代码实现.....	416
19.6 小结.....	420
第 20 章 综合实例BBS系统	421
20.1 概述.....	421
20.2 问题分析.....	422
20.2.1 BBS 系统功能分析.....	422
20.2.2 数据库建立.....	422

20.3	程序实现	424
20.3.1	用户注册	424
20.3.2	用户登录	430
20.3.3	信息显示	433
20.3.4	信息发布	438
20.3.5	信息回复	445
20.4	小结	447
附录	System.Web.UI命名空间	448

第1章 ASP.NET应用程序设置

ASP.NET 应用程序被定义为可以从 Web 应用程序服务器上虚拟目录及其子目录中调用的所有文件、页、处理程序、模块和可执行代码的总和。在每一个 ASP.NET 应用程序里都包含一个名为 Global.asax 的文件。它主要负责一些高级别的应用程序事件，例如应用程序的开始和结束、会话状态的开始和结束等。

了解 Global.asax 文件的运行机制，将会对理解 ASP.NET 应用程序开发机理有很大的作用。本章将对 ASP.NET 应用程序以及 Global.asax 文件的配置进行介绍，希望这一章内容的讲述能为读者后面内容的学习奠定一个良好的基础。

1.1 Global.asax 概述

为了对 Global.asax 进行介绍，就必须首先安装 Visual Studio.NET 环境。本书的程序都是在 Visual Studio.NET 2003 中进行开发的，读者要先在自己的机子上搭建好开发环境。当然，也可以安装最新的 Visual Studio.NET 2003。为了创建一个 ASP.NET 应用程序，需要先安装并配置 IIS 服务器。一般来说，如果所安装的操作系统是 Server 版的话，则已经安装了 IIS 服务器；否则要先用安装盘来添加“IIS 服务组件”。安装好后，选择“开始”|“程序”|“管理工具”|“IIS 服务管理器”命令，就会弹出 IIS 配置对话框，并在这里完成虚拟目录的设置。另外还需要对相关文件夹进行设置，把其属性设置为“Web 共享”。这样就完成一个 ASP.NET 应用环境的设置。如果要添加一个新的 ASP.NET 应用，可以直接打开 Visual Studio.NET，添加新项目并设定项目名称来实现，如图 1-1 所示。

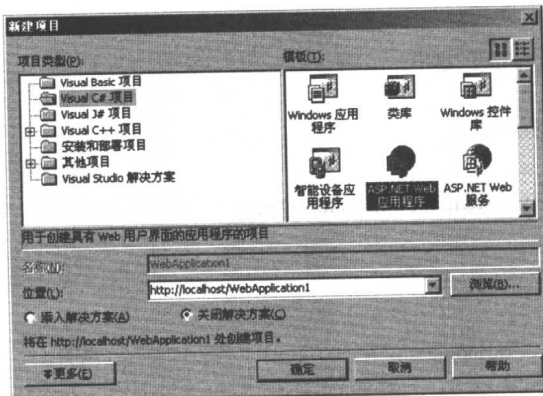


图 1-1 新建 ASP.NET 应用程序

在说明了如何创建 ASP.NET 应用程序后,就要来介绍 Global.asax 文件了。了解 ASP 的程序员都知道,在 ASP 中有一个 Global.asa 文件。该文件主要通过应用程序事件,如 Application_Start, Application_End, Session_Start 和 Session_End 等来处理应用程序逻辑。而在 ASP.NET 中则提供了 Global.asax。从概念上来说,这两者是非常类似的。当位于应用程序命名空间内的任何资源或者 URL 被首次访问时,ASP.NET 系统将自动解析 Global.asax 文件并把它编译为动态的 .NET 框架类——此类派生自 HttpApplication 基类并加以扩充。在创建 HttpApplication 派生类实例的同时,还将引发 Application_Start 事件。随后,HttpApplication 实例将处理页面的一个个请求或者响应,同时触发 Application_BeginRequest 或者 Application_EndRequest 事件,直到最后一个实例退出时才引发 Application_End 事件。

Global.asax 文件是经过设置的。任何关于该文件的直接 URL 请求都将被拒绝,从而可以保证外部的用户无法下载及查看其内容。当改变 Global.asax 文件的时候,.NET Framework 框架可以及时地检测出这种变化。此时,它会完成应用程序的所有当前请求,将 Application_OnEnd 事件传送给每一位聆听者,重新启动应用程序。当浏览器发出的下一个请求到达时,ASP.NET 应用程序将重新解析 Global.asax 文件,并激发 Application_OnStart 事件。

Global.asax 文件是一个可选的文件,它位于 ASP.NET 应用程序根目录下。如果不对它进行定义的话,则系统默认为用户未定义任何应用程序或会话事件处理程序。一般来说,如果用户不需要编写任何应用程序或者会话事件处理程序的话,可以选择不要 Global.asax 文件。

下面通过一个程序实例来说明 Global.asax 文件的应用。首先,新建一个 ASP.NET Web 应用项目,其项目名称为 WebGlobal。其中,Global.asax 文件的代码设置如下:

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Web;
using System.Web.SessionState;

namespace WebGlobal
{
    public class Global : System.Web.HttpApplication
    {
        public Global()
        {
            InitializeComponent();
        }
        protected void Application_Start(Object sender, EventArgs e)
        {
```



```
    }  
    protected void Session_Start(Object sender, EventArgs e)  
    {  
        Response.Write("Session Started<br>");  
    }  
    protected void Application_BeginRequest(Object sender, EventArgs e)  
    {  
        Response.Write("<h2>Global.asax Sample</h2>");  
        Response.Write("Begin Reques...<br>");  
    }  
    protected void Application_EndRequest(Object sender, EventArgs e)  
    {  
        Response.Write("End Request...<br>");  
    }  
    protected void Application_AuthenticateRequest(Object sender, EventArgs e)  
    {  
    }  
    protected void Application_Error(Object sender, EventArgs e)  
    {  
    }  
    protected void Session_End(Object sender, EventArgs e)  
    {  
    }  
    protected void Application_End(Object sender, EventArgs e)  
    {  
    }  
}
```

在上面的代码中可以看到，Global.asax 是由 System.Web.HttpApplication 类派生的。它提供了 Application_Start, Application_End, Session_Start, Session_End 等应用程序级事件。在相应的事件中，都添加了一些代码，以使得读者可以清晰地看到 Global.asax 文件的执行情况。

随后就要进行 Web 页面的加工了，主要对 WebForm1.aspx 进行设计。其中，界面设计如图 1-2 所示。

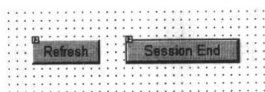


图 1-2 界面设计

在这里添加了两个按钮，分别命名为 Refresh 和 Session End，用于完成刷新和关闭