

精通.NET 编程系列

精通ASP.NET 编程

黎晓冬 李华飏 王福水 编著



科学出版社

www.sciencep.com

精通.NET 编程系列

精通 ASP.NET 编程

黎晓冬 李华飏 王福水 编著

科学出版社

内 容 简 介

ASP.NET 是微软公司推出的新一代 Active Server Pages,也是微软.NET 体系结构的组成部分,它由.NET 框架提供的构件产生,提供一个网络应用模型,使开发人员能轻松地创建网络应用程序。因为 ASP.NET 支持多种语言,本书选用了最常用的两种语言 Visual Basic.NET 和 C#,介绍 ASP.NET 的各种新应用。本书从基础开始讲解,从网络表单、自定义控件、HTML 控件、Web 控件、Web 服务、Web Application、缓存、本地化信息、.NET 数据库编程等方面系统地介绍了 ASP.NET 的知识。最后,给出了网上花店和 BBS 系统的制作实例。

本书内容丰富,知识全面,结构清晰,层次由浅入深,适合广大网络开发人员和设计人员使用。

图书在版编目(CIP)数据

精通 ASP.NET 编程/黎晓冬,李华飏,王福水编著. —北京:科学出版社,2003

(精通.NET 编程系列)

ISBN 7-03-011294-6

I. 精... II. ①黎...②李...③王... III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2003) 第 018340 号

责任编辑:孟战龙/责任校对:郝 岚

责任印制:吕春珉/封面设计:郭 建

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

2003 年 4 月第 一 版 开本: 720×1000 1/16
2003 年 4 月第一次印刷 印张: 28
印数: 1—4 000 字数: 546 000

定价: 46.00 元(含光盘)

(如有印装质量问题,我社负责调换〈路通〉)

前 言

2000 年 6 月, 微软公司发布了微软.NET 平台, 它可以简化高分布式环境下开发 Web 应用程序的过程, 是基于 Web 协议, 为实现不同语言和平台间的高度交互性而创建的新一代平台。ASP.NET 是微软.NET 体系结构的组成部分, 它不是单纯的 ASP 3.0 的升级版本, 而是继承了 ASP 3.0 的优点, 并在此基础上扩展了大量实用新功能, 同时, 修复了以前的错误。由此, ASP.NET 性能更稳定, 使用更便捷。

本书全面地介绍了 ASP.NET 的各项新特性和新功能的应用, 全书共分 12 章:

第 1 章 ASP.NET 概述。介绍了 ASP.NET 的发展历程、ASP.NET 的新特性及与以前版本 ASP 的比较, 让读者更深刻地了解 ASP.NET。

第 2 章 运行环境。介绍 ASP.NET 的软硬件配置及运行环境的安装。

第 3 章 C#语法基础。介绍了 C#语言的语法, 为 ASP.NET 编程做准备。

第 4 章 ASP.NET 网络表单。介绍 ASP.NET 网络表单的各种应用。

第 5 章 自定义控件与 HTML 控件。介绍自定义控件与 HTML 控件的使用, 也是 ASP.NET 编程中常用的技术。

第 6 章 Web 控件。Web 控件是抽象的、加强型对象模板的服务器端控件, 本章详细介绍了其功能和应用。

第 7 章 Web 服务。介绍 Web 服务的应用。

第 8 章 Web Application。介绍 Web 应用程序的概念及应用。

第 9 章 缓存。缓存技术的应用提高了计算机系统的性能, 本章讲述如何使用缓存技术提高 ASP.NET 程序的性能。

第 10 章 本地化信息。介绍在 ASP.NET 中如何使用本地化信息。

第 11 章 .NET 数据库编程技术。介绍在 ASP.NET 中如何应用数据库编程技术。

第 12 章 ASP.NET 实战篇。介绍网上花店和 BBS 系统实例的制作。

本书使用了丰富的示例, 将 ASP.NET 的功能与实际编程结合起来, 边讲边练, 从技术层次上由浅入深, 逐步深入地介绍 ASP.NET 技术, 在示例的编写中使用 Visual Basic.NET 和 C#代码, 从多方面解析 ASP.NET 的技术难点。本书还附有光盘, 书中示例的源代码都包含在光盘中, 减少了读者的重复劳动。

本书主要面向初级和中级读者, 对高级用户和高等学院的相关专业师生也有重要的参考作用。

2002/11/11

本书由逸枫工作室王福水、黎晓冬等编著，参与工作的还有李华飏、李方、王远庆、张力、王东、李立、张军、王江、唐洁、王永、李小军、彭兵、陆兵等，在此表示感谢。由于时间的关系，书中难免有不足之处，敬请广大专家和读者指正。

编者

目 录

第 1 章 ASP.NET 概述.....	1
1.1 ASP.NET 简介.....	1
1.2 ASP.NET 特点.....	2
1.3 ASP.NET 与 ASP 的比较.....	4
1.4 微软公司的 NGWS 系统.....	6
第 2 章 运行环境.....	7
2.1 软硬件环境需求.....	7
2.2 安装.....	7
第 3 章 C#语法基础.....	11
3.1 数据类型.....	11
3.1.1 数值类型.....	11
3.1.2 数值类型的转换.....	17
3.1.3 引用类型.....	19
3.1.4 装箱与解箱.....	23
3.2 语句.....	25
3.2.1 选择语句.....	25
3.2.2 循环语句.....	31
3.2.3 跳转语句.....	38
3.2.4 异常处理语句.....	42
3.2.5 checked 和 unchecked 语句.....	48
3.2.6 unsafe 和 fixed 语句.....	52
3.2.7 lock 语句.....	54
3.3 属性.....	57
3.3.1 属性的声明.....	57
3.3.2 属性的访问.....	57
3.4 方法.....	60
3.4.1 方法的声明.....	60
3.4.2 方法参数.....	62
3.5 索引.....	66
3.5.1 索引的声明.....	66

3.6	事件	70
3.6.1	事件的声明	70
3.6.2	事件的预定和撤消	70
3.6.3	事件的访问	71
3.7	继承	73
第 4 章	ASP.NET 网络表单	76
4.1	表单语法	77
4.1.1	执行代码	77
4.1.2	声明代码	81
4.1.3	服务器控件	82
4.1.4	HTML 服务器控件	85
4.1.5	数据绑定	87
4.1.6	对象标记	91
4.1.7	服务器端注释	93
4.1.8	服务器端#Includes	94
4.2	服务器端控件	95
4.2.1	声明服务器端控件	95
4.2.2	运行服务器端控件	95
4.2.3	处理控件动作事件	96
4.2.4	处理多个控件的动作事件	98
4.3	在控件中应用 CSS	102
4.3.1	在 HTML 控件中应用 CSS	102
4.3.2	在 Web 服务器控件中应用 CSS	102
4.4	服务器控件表单验证	104
4.4.1	客户端验证	105
4.4.2	显示验证错误	111
4.4.3	应用 CompareValidator 控件	117
4.4.4	应用 RangeValidator 控件	123
4.4.5	应用 RegularExpressionsValidator 控件	127
4.4.6	应用 CustomValidator 控件	132
4.4.7	登录系统综合验证	139
4.5	网络表单用户控件	145
4.5.1	用户控件属性	148
4.5.2	在用户控件中封装事件	150

4.5.3 动态创建用户控件.....	153
4.6 数据绑定的服务器端控件.....	153
4.6.1 绑定概述和语法介绍.....	153
4.6.2 绑定到简单属性.....	154
4.6.3 绑定到集合和列表.....	156
4.6.4 绑定表达式或方法.....	162
4.6.5 DataBinder.Eval.....	165
4.7 服务器端数据访问.....	166
4.7.1 Connection、Commands 和 DataSets.....	166
4.7.2 访问基于 SQL 的数据.....	166
4.7.3 执行带参数的选择.....	167
4.7.4 向 SQL 数据库中插入数据.....	170
4.7.5 更新 SQL 数据库中的数据.....	176
4.7.6 删除 SQL 数据库中数据.....	185
4.7.7 在 SQL 数据库中排序数据.....	188
第 5 章 自定义控件与 HTML 控件.....	192
5.1 自定义控件.....	192
5.1.1 开发自定义控件.....	192
5.1.2 定义简单的属性.....	193
5.1.3 定义类属性.....	196
5.1.4 检索内部信息.....	199
5.1.5 开发合成控件.....	200
5.1.6 在合成控件中处理事件.....	203
5.1.7 从合成控件中激发自定义事件.....	205
5.1.8 维护状态.....	208
5.1.9 开发模板控件.....	211
5.1.10 定义一个自定义的控件构造器.....	215
5.2 HTML 控件.....	218
5.2.1 HtmlAnchor 控件.....	218
5.2.2 HtmlImage 控件.....	221
5.2.3 HtmlInputCheckBox 控件.....	227
5.2.4 HtmlInputRadioButton 控件.....	232
5.2.5 HtmlButton 控件.....	238
5.2.6 HtmlInputButton 控件.....	242

5.2.7	HtmlInputFile 控件	246
5.2.8	HtmlInputText 控件	252
5.2.9	HtmlTable、HtmlTableRow 和 HtmlTableCell 控件	256
第 6 章	Web 控件	264
6.1	AdRotator 控件	264
6.2	CheckBox 控件	267
6.3	CheckBoxList 控件	273
6.4	Button 控件	274
6.5	Calendar 控件	275
6.6	DropDownList 控件	277
6.7	HyperLink 控件	279
6.8	Image 控件	281
6.9	ImageButton 控件	284
6.10	Label 控件	287
6.11	LinkButton 控件	288
6.12	ListBox 控件	291
6.13	Panel 控件	294
6.14	Placeholder 控件	297
6.15	RadioButton 控件	301
6.16	RadioButtonList 控件	304
6.17	RegularExpressionValidator 控件	307
6.18	RequiredFieldValidator 控件	308
6.19	Table、TableRow 和 TableCell 控件	310
6.20	TextBox 控件	312
6.21	ValidationSummary 控件	313
6.22	XML 控件	314
第 7 章	Web 服务	316
7.1	Web 服务概述	316
7.2	Web 服务示例	320
7.3	在 Web Service 中使用数据	323
第 8 章	Web Application	326
8.1	概念	326
8.2	创建 Application	326
8.2.1	Application 的结构	326

8.2.2	设置 Application 的目录.....	327
8.2.3	创建一个 Application.....	328
8.2.4	Application 的生命周期.....	329
8.2.5	关于多线程.....	329
8.3	配置 config.web.....	330
8.3.1	配置概述.....	330
8.3.2	配置规则.....	330
8.3.3	配置语法.....	331
8.4	Application 对象.....	334
8.4.1	Application 对象的重要属性.....	334
8.4.2	Application 对象的重要方法.....	335
8.5	Session 对象.....	343
8.6	应用 global.asax 文件.....	348
8.6.1	Application 或 Session 作用范围事件.....	349
8.6.2	Application 或 Session 作用范围对象.....	350
8.7	管理 Application State.....	351
8.7.1	使用 Application State.....	351
8.7.2	使用 Session State.....	352
8.7.3	使用客户端 Cookies.....	354
8.7.4	使用 View State.....	355
8.8	安全访问限制.....	356
8.8.1	验证和授权 (Authentication And Authorization).....	356
8.8.2	基于 Windows 的验证.....	358
8.8.3	基于 FORM 的验证.....	358
8.8.4	授权用户和角色.....	360
第 9 章	缓存.....	364
9.1	缓存概述.....	364
9.2	输出缓存.....	365
9.3	数据缓存.....	367
9.3.1	数据缓存简介.....	367
9.3.2	数据缓存使用.....	368
第 10 章	本地化信息.....	373
10.1	本地化.....	373
10.1.1	本地化支持.....	373

10.1.2	配置设置	374
10.2	Culture 和编码的设定	374
10.2.1	编码	374
10.2.2	使用 CultureInfo	375
第 11 章	.NET 数据库编程技术	381
11.1	ADO.NET 概述	381
11.2	ASP.NET 的优势	382
11.3	Managed Provider	384
11.4	DataSet	384
11.4.1	DataSet 概述	384
11.4.2	处理数据集的步骤	386
11.4.3	创建数据集	387
11.5	通过 ADO.NET 访问数据库	387
11.5.1	ADO.NET 访问数据库的步骤	387
11.5.2	ADO.NET 对象	388
11.5.3	使用 DataSet	405
11.5.4	使用 DataTable	410
11.6	数据库基本操作示例	412
第 12 章	ASP.NET 实战篇	420
12.1	网上花店	420
12.1.1	设计综述	420
12.1.2	实施步骤	421
12.1.3	花店功能介绍	424
12.2	BBS 系统	426
12.2.1	设计综述	426
12.2.2	实施步骤	427
12.2.3	BBS 系统功能介绍	434

第 1 章 ASP.NET 概述

随着网络经济的到来,微软公司希望帮助用户能够在任何时候、任何地方、利用任何工具都可以获得网络上的信息,并享受网络通信所带来的快乐,在 2000 年微软公司发布了新一代平台——微软.NET, .NET 战略就是为着实现这样的目标而设立的。它是一项非常庞大的计划,也是微软公司今后几年发展的战略核心。

微软.NET 平台包括用于创建和操作新一代服务的.NET 基础结构和工具;可以启用大量客户机的.NET User Experience; 用于建立新一代高度分布式的数以百万计的.NET 积木式组件服务; 以及用于启用新一代智能互联网设备的.NET 设备软件。.NET 技术是一项革命性的技术,包含了分布式计算、XML、组件技术、即时编译技术等多种功能集成的环境。.NET 的核心产品包括 Visual Studio.NET、.NET Framework 和基于.NET Framework 的 ASP.NET。.NET 环境使用统一的 Internet 标准(如 XML)将不同的系统对接,并采用高度分布式应用服务架构,而且使用了一个综合的管理程序,该程序能管理平台中运行的服务程序,为运行程序强大的安全保护后台。

微软.NET 的策略是将互联网本身作为构建新一代操作系统的基础,对互联网和操作系统的设计思想进行合理延伸。这样,开发人员必将创建出摆脱设备硬件束缚的应用程序,以便轻松实现互联网连接。微软.NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

1.1 ASP.NET 简介

ASP 的第一个版本是 0.9 测试版。它给 Web 开发带来一阵暴风,它能够将代码直接嵌入 HTML,使得设计 Web 页面变得更简单,更强大,并且通过内置的组件能够实现强大功能,最明显的就是 ActiveX Data Objects (ADO),它使得建立一个动态页面如小孩子玩游戏一样简单。

最后出场的是 Active Server Page 1.0,它作为 IIS 的附属产品免费发送。并且不久就在 Windows 平台上广泛使用。ASP 与 ADO 的结合使开发人员很容易地在一个数据库中建立和打开一个记录集。这无疑使它如此快就被大众接受的因素,因为你现在能使用这些脚本建立和打开一个记录集,处理和输出任何数据,以任何顺序,几乎只要你能想到的,它就能完成。

1998 年, 微软公司又发布了 ASP 2.0。ASP 1.0 和 ASP 2.0 主要区别是外部的组件需要实例化。有了 ASP 2.0 和 IIS 4.0, 我们就有可能建立 ASP 应用了, 而且每个组件就有了自己单独的内存空间。内置的 Microsoft Transaction Server (MTS) 也使制做组件变得简单。

微软公司接着开发了 Windows 2000 操作系统。这个 Windows 版本给我们带上了 IIS 5.0 以及 ASP 3.0。此次并不是简单对 ASP 进行补充, 核心的不同实际上是把很多的事情交给了 COM 来做。在 Windows 2000 中, 微软结合了 MTS 与 COM 核心环境做出了 COM+, 这就让主机有了一种新的方法来使用组件, 同样给主机带来了更多的稳定性, 成了一个可以升级的效率高的工作平台。IIS 5.0 在表面上似乎没有改什么, 但是在接口上动的手术比较大。在内部, 它使用 COM+ 组件服务来对组件提供一个更好的执行的环境。

有了这些, 微软公司推出了 ASP.NET, ASP.NET 又叫 ASP+, 它不是 ASP 的简单升级, 而是 Microsoft 推出的新一代 Active Server Pages。ASP.NET 是微软发展的新的体系结构 .NET 的一部分, 其中全新的技术架构会让每个人的编程生活变得更简单。

ASP.NET 不同于微软公司早期的 ASP, 它是新一代 .NET 平台的组成部分, 它修复了以前 ASP 运行时会产生错误, 同时, 包含了以前 ASP 的优点, 其全新的技术架构使编程变得更加简单。ASP.NET 是一个用于 Web 开发的全新框架, 包含了许多新的特性。ASP.NET 提供了更易于编写、结构更清晰的代码, 这些代码很容易进行再利用和共享; ASP.NET 使用编译后的语言, 从而提升性能和伸缩性; ASP.NET 使用 Web 表单使开发更直观, 利用面向对象技术促进组件的再利用。另外, ASP.NET 中还包括有页面事件、Web 控件、缓冲技术以及服务器控件和对数据捆绑的改进。供 ASP.NET 使用的库以及在微软 .NET 框架中允许通过 Web 使用客户商用函数, 为程序员提供了更多新的开发机会。

1.2 ASP.NET 特点

ASP.NET 是内建于通用语言运行环境中的编程架构, 用于创建功能强大的 Web 应用程序, 通过 Web 开发模块提供重要的特性及功能:

1. 增强性能

ASP.NET 是运行在服务器端的通用语言运行环境。和以前的解释运行方式不同, ASP.NET 充分使用了提前绑定、JIT (Just-In-Time) 编译、本地优化、缓存服务 (Cache) 等一系列技术来达到提高运行效率的目的。

2. 强大的开发工具支持

由于 ASP.NET 和 Visual Studio.NET 是集成在一起的, 所以, ASP.NET 拥有一套强大的开发工具, 包含了大量的工具栏和设计器, 拖放式服务器控件、自动配置等功能都为程序开发提供强大的支持。因此, 开发 ASP.NET 的页面或者应用程序也不需要专门的开发工具。ASP.NET 运行脚本只需要服务器担任系统和服务器软件 (ASP.NET 要求至少 Windows 2000 Server 和 IIS 5.0) 支持。

3. 功能强大适应性好

ASP.NET 是基于通用语言的运行环境 (Common Language Runtime) 的, 所以, 对于网络应用程序开发人员来说, 它功能强大, 并且适应性很强。所有一般语言具有的类库、消息和数据连接都能方便地连接到网络上。ASP.NET 同时也具有语言无关性, 所以用户可以选择自己所需要的开发语言 (如 C#、Visual Basic、JavaScript 等)。

4. 简易性

ASP.NET 使常用的网络开发变得相当方便, 从简单的提交表单和客户端验证的实现和整个网站的配置, 都可以轻松完成。ASP.NET 页面允许用户创建自己的界面, 从描述代码中分离出程序逻辑, 并且进行事件处理。另外, 通用语言运行环境利用像垃圾内存收集等功能来简化程序的开发流程。

5. 可管理性

ASP.NET 应用一个基于文本的分层配置系统, 可以简化对服务器环境和网络应用程序的设置。因为 ASP.NET 的管理是基于“无本地管理”的思想, 配置信息是以纯文本的方式存储的, 只要设置相应的文本文件就可以完成一系列的站点设置, 而不需要本地管理工具的协助。另外, 如果一个 ASP.NET 应用程序要配置到服务器上, 不必在服务器上使用一些特殊的软件进行设置, 只要简单地将制作好的文件复制到服务器上即可, 而不必运行编译代码。

6. 可恢复性和有效性

ASP.NET 在设置时遵循了可扩展性的思想, 它对多进程环境下的应用程序进行了优化。进程可以动态地被 ASP.NET 调用和跟踪。如果一个进程失败, 其他的进程还可以照常运行, 从而保证应用程序不中断。

7. 可定制性和扩展性

ASP.NET 提供了可定制和重用的体系结构, 用户可以在一定的层次上修改 ASP.NET 提供的程序段, 甚至可以利用定制的组件重新编写 ASP.NET 的组件, 从而实现根据不同需要定制的目的。

8. 安全可靠

ASP.NET 基于 Windows 验证机制，它是为每个应用程序进行配置的方法，确保用户的应用是安全可靠的。

9. 语言支持

ASP.NET 即时地对 C#(读做“C Sharp”)、Visual Basic 和 Jscript3 种语言提供了支持。

1.3 ASP.NET 与 ASP 的比较

ASP.NET 和 ASP 的最大区别在于编程思想的转变，而不仅仅在于功能增强。ASP 使用 VBScript 或 JavaScript 这样的脚本语言混合 HTML 来编程，而脚本语言属于弱类型、面向结构的编程语言，而不是面向对象的，这就容易产生以下几个问题：

1. 代码逻辑混乱、难于管理

由于 ASP 是脚本语言混合 HTML 编程，因此很难看清代码的逻辑关系，并且随着程序的复杂性增加，使得代码的管理十分困难，甚至超出一个程序员所能达到的管理能力，从而造成出错或这样那样的问题。

2. 代码的可重用性差

因为 ASP 采用的是面向结构的编程方式，并且混合 HTML，所以可能是页面原型需要修改一点，导致整个程序都需要修改，更别提代码重用了。

3. 弱类型造成潜在的出错可能

尽管弱数据类型的编程语言使用起来更方便一些，但相对于它所造成的出错几率是远远得不偿失的。

ASP.NET 摆脱了以前 ASP 使用脚本语言来编程的缺点，理论上可以使用任何编程语言包括 C++、Visual Basic 和 JavaScript 等，但是最合适的编程语言还是 Microsoft Studio 为 ASP.NET Framework(框架)专门推出的 C#。尽管微软公司称 C#内核中更多的像 Visual C++，但实际上它和 Java 更像一些。C#是面向对象的编程语言，而不是一种脚本，所以它具有面向对象编程语言的一切特性，比如封装性、继承性和多态性等，这就解决了刚才谈到的 ASP 的那些弱点。封装性使得代码逻辑清晰、易于管理，并且应用到 ASP.NET 上就可以使业务逻辑和 HTML 页面分离，这样无论页面原型如何改变，业务逻辑代码都不必做任何改变；继承性和多态性使得代码的可重用性大大提高，程序员可以通过继承已有的对象最大限

度地减少不必要的工作，并且 C# 和 C++、Java 一样提供了完善的调试/纠错体系。

ASP.NET 与 ASP 相比具有以下新特性：

1. ASP.NET 基于全新的架构

ASP.NET 是完全基于 NGWS 运行时框架的，NGWS 运行时框架(NGWS runtime framework)是 Microsoft 提出的面向下一代 Web 的 .NET 架构的核心组成部分。NGWS 运行时是一种全新的针对 Windows 的底层基础架构，它位于 Windows 的运行和用户程序之间，被用来管理代码的执行。它在开发人员和操作系统之间提供一种封装，使 Windows 编程变得更加简单。NGWS 运行时是完全基于对象的，它扩展了 COM(Component Object Model, 组件对象模型)的结构，为可扩展分布式应用程序提供了更多的新的增强性服务。当 NGWS 运行时，我们不再必为忘记将 ASP.NET 页面中某个对象清零而担心，因为它会自动处理内存垃圾回收。同时一些对操作系统至关重要的底层服务的访问也变得更加轻松。

2. ASP.NET 通过编译来执行

ASP.NET 针对 ASP 的执行环节做了许多改进，页面脚本不再通过解释来执行，而是通过 NGWS(Next Generation Web Services, 下一代 Web 服务)兼容的编译器来编译执行。程序员仍然和以前一样编写代码，编译工作在脚本代码第一次收到请求并运行时由 NGWS 运行时自动完成，同时保存一份缓存的拷贝。当脚本再次收到请求时，系统直接运行缓存的拷贝，从而大大加快了执行速度，因为我们现在运行的是经过编译的代码。如果源脚本有了改动，运行时会自动从缓存中丢弃旧的编译版本，以便下次收到执行请求时重新编译代码。

3. 数据缓存技术

设想一下这样的情形：你有一个动态生成的 ASP.NET 页面，该页面的内容源于一个后台数据库，而这个数据库可能好几天才得以更新一次，而现在，假设你的服务器收到了来自客户端的一个请求，服务器会与数据库建立连接，执行查询，取回记录，反馈给用户，服务器执行的功能很简单，可是当服务器在同一时间要处理成千上万的相同的请求，会产生什么样的后果呢？毫无疑问，结果是无谓的重复运算与可怕的系统资源占用。所以，这就需要用到数据缓存技术。

在 ASP.NET 中，当服务器第一次从数据库中取回数据后，数据被保存在变量中，当服务器接到相同的请求后，经过缓存的数据直接从服务器传递到客户端，现在我们可以不必打搅数据库了。但是，必须在代码中设定数据的缓存期限，到了规定的时间或满足特定的条件，就清空缓存，取回新的数据。至于如何来设定缓存的期限，就完全取决于你的需求与技巧了。

4. ADO.NET, 更易于使用的 ADO

Microsoft 在推出 ASP.NET 的同时, 也推出了新一代的 ADO, 即 ADO.NET。ADO.NET 以一种 n 维 (n -dimensional) 的, 面向对象的方式来处理数据。在 ADO 中, 数据以二维的形式表现出来, 即行、列。但如果 RecordSet 中有一个特定的行列本身又是一个 RecordSet, 从而形成一个新的维度呢? 这样的情况在 ADO 中很棘手, 而用 ADO.NET 创建这样的多维 RecordSet 却只是举手之劳。事实上, 上面所说的 RecordSet 在 ADO.NET 中已不复存在, 取而代之的是 DataSet。在 ADO.NET 中, DataSet 相当于 RecordSet 的集合, DataView 则类似于 RecordSet。

1.4 微软公司的 NGWS 系统

微软公司正在构建一个为 Windows 平台提供服务的完全新的基础结构。当这个新的基础结构被使用的时候, 它将位于低级的 Windows 操作系统、用户应用和代码执行的管理之间。这个基础结构, 被称为 NGWS (Next Generation Web Service) 系统, 它提供了开发人员和操作系统之间的封装性, 因此这将使 Windows 程序更加容易编写。开发人员将不用再去考虑一些底层的细节问题, 如内存管理、打印机管理、语言差异以及其他一些琐事。

NGWS 系统是一个面向对象的系统。举例来说, 在 NGWS 系统中每一种数据类型是一个对象, 它由合适的方法和属性所组成。使用 NGWS 系统, 开发人员将不能够再使用指针。NGWS 系统同时自动地管理碎片集。如果忘了对一个对象赋初值, 也没有关系。最后对于一个操作系统而言, 非常重要的低层次的服务, 例如, 安全性、认证、文件系统存取以及网络工作信息等服务, NGWS 系统也提供了对它们进行操作的方法。

使用 NGWS 系统来编写的代码应该被称为“被管理的代码”更为合适。对于没有使用这个基础结构的代码应该被称为“未被管理的代码”。

ASP.NET 页面是需要被编译的, 而不再是进行脚本的解释。ASP.NET 页面将被 NGWS 系统的兼容编译器进行编译, 这个兼容编译器包括 Visual Studio 的所有组成部分(包括 Visual Basic、Visual C++、Java、C#), 以及其他编译器(如 NGWS 系统的兼容 Perl 编译器)。