

ASP.NET

与网站开发编程

实战

陈 湘 主编



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



ASP.NET 与网站 开发编程实战

陈 湘 主编

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书详细介绍了微软公司重点推广的 ASP.NET(ASP+)Web 应用程序开发技术的基本框架。本书的主要内容包括:ASP.NET 的安装、VB 快速参考、ASP.NET 常用组件对象、网页表单、服务器控件、数据库访问、缓存服务、跟踪服务及应用程序调试等,覆盖了 ASP.NET 这门新技术的所有主要内容。为了便于广大读者学习掌握,本书在讲解中使用了大量编程实例及其代码,最后还提供了两个完整的经典实例供读者学习和借鉴。

本书适合于探索 Web 应用程序开发的广大用户,也可作为有经验的 Web 开发人员的参考资料。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:ASP.NET 与网站开发编程实战

作 者:陈 湘 主编

出 版 者:清华大学出版社(北京清华大学学研大厦 A 座,邮编 100084)

责任编辑:宋 韬

印 刷 者:北京市清华园胶印厂

发 行 者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:27.75 字数:642 千字

版 次:2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

书 号:ISBN 7-302-05152-6/TP·3019

印 数:0001-5000 册

定 价:41.00 元

前 言

ASP.NET 是建立在公共语言运行环境(COMMON LANGUAGE RUNTIME)上的,ASP.NET 是 .NET 框架应用程序的执行引擎,可以用在服务器端构造一个强大的 Web 应用程序的编程框架。这是微软公司对 ASP.NET 的定义。它表明了两点意思:ASP.NET 的运行环境是公共语言运行环境;ASP.NET 是 .NET 框架的一个组成部分。

.NET 框架是创建、部署和运行 Web 服务及其他应用程序的一个环境,它致力于消除“巴别塔效应”。以前,令许多程序开发人员比较头疼的一个问题就是:怎样将用多种编程语言写出的程序块连接起来,以使它们能协同工作。开发人员必须依据所选编程语言来使用不同的命令集,这些命令被称作“框架”。当不同编程语言发生冲突时,就会导致“巴别塔效应”。也就是说,当各种编程语言混杂在一起时,任何一种编程语言的效力都会被削弱。另外,开发人员在学习一种新的编程语言时,总会经历一段艰难的学习过程。有了 .NET 框架后,开发人员便可对选用的任何编程语言一律使用统一的命令集,这些命令集都被统一封装到 .NET 框架提供的框架类中。这样,.NET 框架就成功地糅合了各种编程语言。无论您是一个传统的 C++ 程序人员,还是一个 VB 编程人员,都可以在 .NET 框架上使用您所喜欢的语言工作,各种语言在 .NET 框架上一律是等同的。

笔者不想也不可能在这里详细讨论关于 .NET 框架的更多内容,因为那是需要几本书才能完成的工作,还是回到我们本书的中心:ASP.NET。

与以往的使用 ASP 进行 Web 开发的模式相比,ASP.NET 展现了几个重要的优势:

性能增强 在服务器上运行的 ASP.NET 代码是经过编译的公共语言运行环境代码,与以往解释性的代码相比,它无疑会有快得多的执行速度。此外,通过预先绑定、及时编译、本地优化及缓存服务等技术,使开发工作既高效又轻松。

世界范围的工具支持 通过 Visual Studio.NET 集成开发环境,ASP.NET 可以获得丰富的工具支持,如拖放控件、自动化配置等。

强大和灵活 公共语言运行环境基本类库、消息处理和数据访问解决方案都已经无缝集成到 ASP.NET,可以使用您喜欢的语言进行开发。另外,公共语言运行环境还保证了您在已经做的基于 COM 的开发可以移植到 ASP.NET。

安全性 使用授权和鉴定技术可以确保应用程序的安全。

以上几点只是 ASP.NET 比较突出的几个优点,ASP.NET 还具有更多特点在这里没有被列举。这些可到 ASP.NET 的具体内容中去寻找,也许您会有更多惊喜的发现。

本书的读者对象

也许您已经是一个有经验的 ASP 开发人员,那么您会从与旧的 ASP 的比较中得到许多 ASP.NET 为您提供的惊喜。然而本书最主要的任务是把那些对 ASP 开发有强烈兴趣,但是又因为各种原因没有去实现自己愿望的读者直接引入 ASP.NET。

本书没有对 HTML 进行专门介绍,但是示例代码中不可避免地要包括一些 HTML 的标记,因此在阅读本书之前您至少要有一些 HTML 的基本知识。本书也没有详尽地介绍书写脚本的某一种高级语言,但是在本书的第 2 章中包含了对 VB.NET 的快速参

考——它是本书示例代码书写脚本时采用的语言。因此,对已经至少了解某一种高级语言(C、JAVA、PASCAL、FORTRAN 等)的读者将会很容易看懂使用 VB.NET 书写的脚本。

在本书的命名中包含了“实战”这一个极具挑战性的词汇,因此通过大量的实例代码来展示 ASP.NET 的各种技术将是本书的一个重要特点。笔者一直深信,成为一个真正的程序员的过程,就是不断地阅读代码和书写代码的过程。只有在与代码打交道的过程中,某一种新技术才会真正地变成自己的财富。当然,这并不意味着本书对基本理论的忽略。在每一个具体的示例代码出现之前,都有相应的基本理论的介绍,尽管这些介绍很精简,但是已经足以使您能够读懂相关的示例代码。

致谢

在本书的编写过程中,得到了博嘉科技公司总经理王松先生的鼎力支持,在此对他表示衷心的感谢。另外,参加本书编写的还有孙忠、刘小伟、邓勇、欧阳劲、张云勇、卢军、唐寅、邹思轶、张炯明、李彬、卫星、贺玉龙、陈明、李宋琛、邓海、吴文锦、田茂敏、刘青松、巫文斌、刘洪海、刘昕、黄亮、范群、钟绪阳、张家彬、胡涛、杨峰、陈兴华等。

延伸服务:如果读者愿意参加“ASP.NET 与网站开发编程实战”的学习培训,或是在学习过程中发现问题,或需要本书相关光盘,欢迎来电。联系电话:(028)5404228;E-mail:bojiakeji@163.net。通讯地址:成都四川大学(西区)建筑学院成都博嘉科技资讯有限公司(邮编:610065)。

作 者

目 录

第 1 章 ASP.NET 简介与安装	1
1.1 从 ASP 到 ASP.NET	2
1.1.1 从静态网页到动态网页	2
1.1.2 从 ASP 到 ASP.NET	2
1.2 安装 ASP.NET	4
1.2.1 所需软件环境	4
1.2.2 安装 .NET Framework SDK	5
1.3 配置 IIS	6
1.3.1 创建虚拟目录	6
1.3.2 验证 ASP.NET 的安装	8
1.4 本章小结	9
 第 2 章 VB.NET 快速参考	10
2.1 注释与续行	11
2.2 常量与变量	11
2.2.1 常量	11
2.2.2 变量	13
2.3 运算符	16
2.3.1 数学运算符	17
2.3.2 赋值运算符	17
2.3.3 位逻辑运算符	17
2.3.4 比较运算符	18
2.3.5 连接运算符	19
2.3.6 逻辑运算符	20
2.3.7 混合运算符	20
2.3.8 运算符优先级	20
2.4 数组	21
2.5 控制语句	22
2.5.1 选择语句	22
2.5.2 循环语句	24
2.6 函数与过程	27
2.6.1 基本概念	27
2.6.2 常用函数	29
2.7 编码约定	39
2.7.1 变量命名约定	39

2.7.2 对象命名约定	40
2.7.3 脚本编排约定	40
2.8 应注意的几个问题	41
2.8.1 VB.NET 代码在网页脚本中的位置	41
2.8.2 使用 VB.NET 函数和子程序时应注意的问题	41
2.9 本章小结	42
第 3 章 ASP.NET 常用组件对象	43
3.1 概述	44
3.2 Response 对象	44
3.2.1 利用 Response 对象发送信息	44
3.2.2 利用 Response 对象重定向浏览器	47
3.2.3 Response 对象属性与方法总结	48
3.3 Request 对象	49
3.3.1 利用 Request 对象获取 URL 传递变量	49
3.3.2 利用 Request 对象获取表单传递值	51
3.3.3 利用 Request 对象获取服务器变量值	54
3.3.4 Request 对象属性与方法总结	56
3.4 Server 对象	58
3.4.1 利用 Server 对象控制页面显示时间	58
3.4.2 利用 Server 对象创建晚绑定对象	61
3.4.3 Server 对象属性和方法总结	61
3.5 Application 对象	63
3.5.1 利用 Application 对象存取变量	63
3.5.2 创建一个 Application 对象	64
3.5.3 同步和互斥	66
3.5.4 Application 事件和 Global.asax 初探	66
3.5.5 Application 对象的属性和方法总结	68
3.6 Session 对象	69
3.6.1 利用 Session 对象存取变量	70
3.6.2 Session 的惟一性和 Session 的终止	73
3.6.3 Session 事件	76
3.6.4 Cookie 与 Session 状态配置	76
3.6.5 Session 对象的属性和方法总结	80
3.7 HttpContext 对象	81
3.8 实例:一个简单的网络聊天室	82
3.8.1 设计前的思考	82
3.8.2 在 IIS 上配置应用程序和 Global.asax 文件	82
3.8.3 创建登录页面	83

3.8.4 创建聊天室主页	85
3.8.5 创建信息发送页面	86
3.8.6 创建信息显示页面	88
3.9 本章小结	90
第4章 ASP.NET 网页表单基础	91
4.1 ASP.NET 网页表单简介	92
4.2 ASP.NET 页面语法	95
4.2.1 ASP.NET 页面语法组成	95
4.2.2 指令集合	96
4.2.3 代码声明块	98
4.2.4 代码集成块	99
4.2.5 HTML 服务器控件语法	99
4.2.6 自定义服务器控件语法	99
4.2.7 数据绑定表达式	100
4.2.8 服务器端对象标志	100
4.2.9 服务器端包含指令	101
4.2.10 服务器端注释	101
4.3 HTML 服务器控件	101
4.3.1 什么是 HTML 服务器控件	101
4.3.2 为什么要使用 HTML 服务器控件	102
4.3.3 常用的 HTML 服务器控件	104
4.4 ASP.NET 服务器控件简介	113
4.4.1 初识 ASP.NET 服务器控件	113
4.4.2 为什么选择 ASP.NET 服务器控件	113
4.5 本章小结	114
第5章 使用 ASP.NET 服务器控件	115
5.1 ASP.NET 服务器控件的基本形式	116
5.2 ASP.NET 服务器控件的基本属性	116
5.3 Style 对象属性参考	117
5.4 内部控件	119
5.4.1 标签控件 Label	119
5.4.2 按钮控件 Button、LinkButton 和 ImageButton	120
5.4.3 文本输入框控件 TextBox	125
5.4.4 选择类控件	125
5.4.5 超链接控件 Hyperlink	135
5.4.6 图形控件 Image	136
5.4.7 容器类控件	136

5.5 增强控件	143
5.5.1 日历控件 Calendar	143
5.5.2 广告交替控件 AdRotator	157
5.6 验证控件	163
5.6.1 概述	163
5.6.2 验证控件的基本属性	163
5.6.3 RequiredFieldValidator 控件	164
5.6.4 CompareValidator 控件	165
5.6.5 RangeValidator 控件	166
5.6.6 RegularExpressionValidator 控件	166
5.6.7 CustomValidator 控件	167
5.6.8 ValidationSummary 控件	168
5.6.9 验证控件综合示例	169
5.7 本章小结	175
 第 6 章 ASP.NET 数据访问基础	176
6.1 SQL 语言基础	177
6.1.1 准备工作	177
6.1.2 在数据库中创建、修改和删除表	179
6.1.3 SQL 常用语句	186
6.1.4 SQL 常用聚合函数	191
6.1.5 创建存储过程	193
6.2 ADO.NET 基础	194
6.2.1 ADO.NET 概述	194
6.2.2 有关数据访问的几个名称空间	196
6.2.3 数据提供者	196
6.3 连接、命令和数据集	196
6.3.1 使用 Connection 对象连接到不同的数据源	197
6.3.2 使用 Command 对象实现对数据的操纵	201
6.3.3 显示记录	202
6.3.4 示例:通过 ODBC 访问数据库	202
6.4 Command 对象和 DataReader 对象	204
6.4.1 Command 对象	204
6.4.2 DataReader 对象	205
6.4.3 示例:使用 DataReader 对象访问数据	206
6.5 DataSetCommand 对象和 DataSet 对象	208
6.5.1 DataSetCommand 对象	208
6.5.2 DataSet 对象	209
6.5.3 示例:使用 DataSet 对象访问数据	215

6.6 本章小结	219
第7章 使用数据列表控件	220
7.1 DataGrid 控件	221
7.1.1 基本语法	221
7.1.2 实现与数据源的绑定	221
7.1.3 自定义外观	222
7.1.4 实现记录的分页显示	223
7.1.5 对记录排序	227
7.1.6 定制列对象	229
7.2 Repeater 控件	251
7.2.1 基本语法	251
7.2.2 常用属性	251
7.2.3 处理事件	254
7.3 DataList 控件	257
7.3.1 基本语法	257
7.3.2 常用属性	257
7.3.3 处理事件	262
7.4 本章小结	269
第8章 ASP.NET 数据访问的高级技术	270
8.1 普通服务器控件的数据绑定	271
8.1.1 概述和语法	271
8.1.2 绑定到简单属性	271
8.1.3 绑定到集合	273
8.1.4 绑定到表达式或者函数	274
8.1.5 DataBinder.Eval()方法	276
8.2 使用参数对象与存储过程	278
8.2.1 使用参数对象	278
8.2.2 示例:使用参数对象检索记录	280
8.2.3 使用存储过程	282
8.3 增添、修改和删除数据库记录	283
8.3.1 增添记录	284
8.3.2 修改记录	289
8.3.3 删除记录	295
8.4 XML 数据访问	297
8.4.1 一个简单的 XML 文件	298
8.4.2 读取 XML 文件	299
8.4.3 示例:访问 XML 数据	300

8.5 本章小结	301
第9章 其他高级技术	302
9.1 缓存服务	303
9.1.1 概述 ASP.NET 的缓存技术	303
9.1.2 页面输出缓存	303
9.1.3 页面数据缓存	305
9.2 使用 Pagelet 建立可重用控件	309
9.2.1 什么是 Pagelet	309
9.2.2 应用一个简单的 Pagelet	309
9.2.3 创建包含属性的 Pagelet 控件	310
9.2.4 创建包含方法的 Pagelet 控件	314
9.3 文件上传	316
9.4 传送 E-mail	318
9.5 跟踪服务	322
9.5.1 概述 ASP.NET 的跟踪服务	322
9.5.2 页面级跟踪服务	323
9.5.3 应用程序级跟踪服务	326
9.6 应用程序调试	327
9.6.1 概述	327
9.6.2 使应用程序处于调试模式	328
9.6.3 调试应用程序	328
9.6.4 设置断点和测试变量的值	330
9.7 本章小结	331
第10章 实例一:在线考试系统	332
10.1 设计前的思考	333
10.2 文件结构	333
10.3 准备工作	334
10.3.1 创建“Examination”数据库	334
10.3.2 配置应用程序	335
10.4 创建考生登录页面	336
10.4.1 实现考生登录页面的页面显示	336
10.4.2 形成最终的登录页面文件	338
10.5 创建答题页面	339
10.5.1 实现答题页面的页面显示	339
10.5.2 计算和显示成绩	342
10.5.3 形成最终的答题页面文件	345
10.6 创建网站管理系统	347

10.6.1 确认管理员身份·····	347
10.6.2 指定考试科目和抽取试题·····	351
10.6.3 管理试题库·····	360
10.6.4 管理考生信息·····	365
10.7 本章小结·····	365
第 11 章 实例二:电子商务——网上书店 ·····	366
11.1 设计前的思考·····	367
11.2 文件结构·····	367
11.3 准备工作·····	369
11.3.1 创建“BookShop”数据库·····	369
11.3.2 在 IIS 上配置应用程序和 Global.asax 文件·····	370
11.4 创建主页·····	371
11.4.1 实现主页的页面显示·····	371
11.4.2 搜索引擎——“书籍查询栏”的代码实现·····	382
11.4.3 身份验证——“用户登录栏”的代码实现·····	386
11.4.4 购书篮的代码实现·····	387
11.4.5 形成最终的主页文件·····	393
11.5 创建注册页面·····	395
11.5.1 实现注册页面的页面显示·····	395
11.5.2 向数据库增添一个用户·····	400
11.5.3 形成最终的注册页面文件·····	401
11.6 创建书屋论坛·····	403
11.6.1 实现书屋论坛的页面显示·····	403
11.6.2 查看与搜索现有帖子·····	408
11.6.3 发表新帖子与回复帖子·····	413
11.7 创建书店管理系统·····	418
11.7.1 确认管理员身份·····	419
11.7.2 管理书籍仓库·····	423
11.7.3 管理注册用户信息·····	430
11.7.4 管理论坛信息·····	431
11.8 本章小结·····	431

第 1 章 ASP.NET 简介与安装

本章要点：

本章简要介绍 ASP.NET 的发展历史,有助于开始对 ASP.NET 的深入学习,同时也介绍安装 ASP.NET 的过程。为了便于学习本书,本章也讲述了配置 IIS 的一些基本内容。

本章主要内容：

- ☐ 从 ASP 到 ASP.NET
- ☐ 安装 ASP.NET
- ☐ 配置 IIS

1.1 从 ASP 到 ASP.NET

1.1.1 从静态网页到动态网页

相信对 Internet 有一点了解的读者,对 WWW(World Wide Web,万维网)这个英语单词的简写都不会陌生。正是 WWW 的出现,使人们把自己的信息放在网页上成为可能。它使用一种叫做“超链接”的技术,让网页访问者可以自由驰骋于各个不同的页面之间。但是,最初放在网页上的信息大多是以字符文本的格式出现的,网页的访问者只能是一个被动的信息接收者,网页的设计者也无法得到更多访问者的信息。也就是说,网页的作者与读者之间缺乏交互行为(当然,通过 e-mail 可以实现交互行为,但当某个网页访问者有几万人时,管理者怎么去处理收到的成千上万的邮件呢)。随着 HTML(Hypertext Markup Language,超文本标记语言)的不断发展,网页设计者可以在网页上放入诸如 Java 小程序、闪烁的 Flash 动画等让网页看起来漂亮美观、生动活泼的东西,好像网页已经开始“动”起来了,但这种“动”与笔者将要提到的动态网页的“动”有着根本性的差别。

正如笔者前面提到的一样,动态网页的“动”是指“交互性”这一概念。也就是说,区分一个静态网页与动态网页的最根本依据是看这个网页是否有交互性。什么是交互性呢?在这里笔者无法给出准确的定义,但是如果读者是一个上网爱好者,可能已经使用过基于这个概念的服务了。比如,当读者在中文雅虎的搜索条件框内输入“鲁迅选集”或者“海湾战争”后,开始搜索就会得到许多条包含读者输入的字符的信息。又比如,当读者第一次上联众网玩游戏或者到 OICQ 聊天时,需要注册成为这些网站的用户。当网页显示信息提示读者注册成功时,读者就已经成功地实现了与服务器的交互。

以往的动态网页需要在服务器端编写可执行程序。如一个叫做 CGI(Common Gateway Interface,公共网关接口)的网络服务器接口,允许服务器上的可执行程序接收并处理客户请求的所有信息,然后将处理结果返回页面。而微软的 ISAPI(Internet Server Application Programming Interface,Internet 服务器应用程序编程接口)则允许编译过的 DLL(Dynamic Link Library,动态链接库)文件直接在服务器上执行。这两种方法都实现了动态网页的交互性,但对数据库的访问却很不方便,而且学习的难度也很大,所以提供一种高效易学的开发动态网页的技术成为一种必然。于是,微软在 1996 年提出了一种叫做 ASP(ActiveX Server Pages,活动服务器页面)的技术。

1.1.2 从 ASP 到 ASP.NET

ASP 最初受到人们的普遍关注,是因为它成功地融入了 ADO(ActiveX Data Object, ActiveX 数据对象)技术。使用过 Visual Basic 或者 Visual C++ 进行数据库开发的读者,一定知道 ADO 是一个多么令人振奋的技术。有了它,使用户对数据库的访问变得像小孩堆积木一样的方便。而现在,可以在自己的网页上使用这种技术。这一切都得归功于

ASP 的出现。

可以说,ASP 的出现掀起了网络开发的一阵热潮,特别是在它的 3.0 版本出现之后。现在,市面上关于 ASP 3.0 的书籍是很多的,而作为 ASP 最新版本的 ASP.NET(通常也叫 ASP+) ,它的流行自然是可以预见和理所当然的了。

从 ASP 的 1.0 到 3.0 版本,ASP 的性能提升主要体现在组件使用性能的提升上,其根本性的结构变化并没有出现。而 ASP.NET 从基础结构上就与以前的版本很不相同,虽然它们从界面上看起来很相似。下面是 ASP.NET 相对于以前 ASP 版本的一些显著优点。

1. 更强大的语言支持

与 ASP 只能用 VBScript 和 JavaScript 这样基本的非结构性语言编写相比,ASP.NET 能够使用种类更多,功能更强大的语言编写。Visual Basic 的开发人员在写 ASP 脚本时很喜欢用 VBScript,因为它是 VB 程序语言的一个子集,VB 开发人员不需要学习一种新的语言就可以直接进行 ASP 的脚本编写工作了。但 VB 开发人员也常常陷于这样一种困惑:自己熟悉的一些 VB 函数或者由多年从事 VB 开发得到的一些经验无法应用到 ASP 的开发工作中,VBScript 仅仅是 VB 的子集,使得 VBScript 不能够充分利用 VB 语言的强大功能。为什么 ASP 不能够使用 VB 进行开发呢? ASP.NET 很好地避开了对这一问题的回答,因为它用 VB 取代了 VBScript 作为脚本编写语言。同样,C++ 的程序员也可以不必再局限于使用 JavaScript 或者 Jscript。ASP.NET 也使用一种叫做 C#(读作:C sharp)的语言,它是一种从 C++ 语言改良的语言,号称同时具有 C++ 的灵活性与 VB 的简单易学性,同时避免了对 C++ 安全性方面的指责。听起来似乎是一种很完美的语言,假如读者一开始什么语言也不会或者拥有充足的时间和足够的耐心去学习,C# 是一个不错的选择。而且似乎可以预见的是,C# 可能会成为 ASP.NET 开发使用的脚本语言之中最流行的一种。但本书的主旨是讨论 ASP.NET,考虑到许多进行过 ASP 开发的人员大多使用 VBScript 或者说考虑到 VB 的支持者要多于 VC++ 的支持者(C++ 程序员一定不同意笔者的这种说法,但事实上大多数业余爱好者在面对 VB 与 VC++ 的选择时,都会一致选择 VB,原因是易学而已),同时也考虑到 ASP.NET 把 VB.NET 作为默认的服务器端脚本语言,所以本书使用 VB.NET 作为开发 ASP.NET 页面的脚本语言。JavaScript 作为大多数浏览器都支持的客户端脚本语言(支持 VBScript 的浏览器种类就少得可怜,不过,它得到了 IE 的支持。在我国,有多少人不使用免费的 IE 浏览网页呢?),在 ASP.NET 中仍然可以得到使用。

事实上,凡是可以编译成为 MSIL(Microsoft Intermediate Language,微软中间语言)的程序语言,都可以成为 ASP.NET 的脚本编写语言。ASP.NET 采用了一种类似 Java 编译的技术,先将脚本语言编译成为 MSIL 语言,再将 MSIL 语言编译成机器码执行。这样,利用 MSIL,ASP.NET 就实现了与 Java 一样的跨平台操作,只要有合适的编译器,任何语言都可以作为 ASP.NET 的脚本语言。

2. 更高的执行效率和开发效率

ASP.NET 在执行脚本时要经过两次编译:从脚本语言到 MSIL 和从 MSIL 到可执行

的机器码。而旧版本的 ASP 则只需要一次编译:从脚本语言直接到可执行的机器码。也就是说,MSIL 技术的使用使 ASP.NET 比旧版本的 ASP 多了一次编译过程,这肯定要多花一些时间。那么,ASP.NET 比旧版本 ASP 有更高的执行效率体现在什么地方呢?

当一个页面被第一次访问时,ASP.NET 确实要慢于旧版本的 ASP,但是当同一页面被多次访问时,ASP.NET 的执行速度则要快于旧版本的 ASP。因为页面在被访问过以后,它的脚本已经编译成了 MSIL 并被存储下来。这样,在以后的访问中,页面的脚本编译都是直接从 MSIL 到机器码,这比普通 ASP 页面每次都要从脚本语言到机器码的编译要快。ASP.NET 因此得到了比旧版本 ASP 更高的执行效率,而数据缓存和页面缓存技术使这一优势更加突出。

将 HTML 标记、文本、对象说明和代码等放在一个页面内是 ASP 开发人员的工作。某个开发人员开始时也许还津津有味地在这里放一个超链接,在那里放一段函数代码,但当内容变得“丰富”起来时,突然会发现面临的是一个多么庞大的 ASP 页面,有趣的工作顿时成为开发人员头疼的事。通过事先的计划和采用诸如“include 文件”等技术可以缓解烦躁的心情,但仍然得做好耐心和艰苦的准备。不仅如此,当编写一个有相似功能的页面时,又不得不重新开始,先前编写出来的页面除了使用“COPY”外,竟然无法重复使用已做过的工作。读者也许会这样问:“事先的规划怎样才是最好的?能否让一些页面共享已经编写好的一段代码?”ASP.NET 很好地回答了这些问题,它使用一种叫做“分离”的技术解决了这些难题,将内容和代码分离,甚至让内容和页面外观分离,这些都让页面的编码看起来井井有条并可以重用。ASP.NET 还通过使用服务器端控件等技术,使 ASP.NET 面向对象的特征更加明显,所有这些都让 ASP.NET 获得了更高的开发效率。

3. 其他优点

也许以上的两个突出优势已经让读者有了学习 ASP.NET 的充足理由,但作为一种先进的技术,ASP.NET 还拥有许多令人心动的优点。如更强大的错误处理和调试特性、更好的安全管理机制、更多的组件服务等。也许读者已经开始跃跃欲试了,但还必须要做一些学习前必要的准备工作。

1.2 安装 ASP.NET

1.2.1 所需软件环境

在操作系统方面,ASP.NET 只能在带有 IIS4.0 的 Windows NT 4.0 Server(或更新版本)上运行,但是 Windows 2000 Server(带 IIS5.0)无疑是最适合的操作系统平台。在数据库方面,还应该有 SQL Server 7.0 以上的版本(最好是 SQL Server 2000)或者 Access 2000,这两者都是微软提供的进行数据库开发的软件。所不同的是 SQL Server 面向的是企业级的大型数据库开发,而 Access 则主要面向桌面数据库。由于 ASP 1.0~3.0 都是作为 IIS 的一个组件由相应的操作系统服务包(Service Pack)提供,安装了服务包自然也

就安装了 ASP 的操作平台,不需要对 ASP 操作平台进行专门安装。但是,由于现在使用的 Windows 操作系统的最新版本是 Windows 2000 系列,相应的 IIS5.0 附带的是 ASP3.0 的版本,所以需要在操作系统上安装 ASP.NET 操作平台(即 .NET Framework SDK),可以到微软的网站下载 .NET Framework SDK。网址是 <http://msdn.microsoft.com/downloads/default.asp>。进入这个页面后,可以看见这个页面的横向被分为两大部分,展开左边部分的“Software Development Kit”项,然后再单击其子项“.NET Framework SDK Beta1”,页面的右边部分就会出现十几项超链接文字。第一项是一次性全部下载安装文件(约 111MB),而下面的项则是把 111MB 的文件分成 11 份和一个合并程序文件(下载完毕后需要执行合并程序文件 setup.exe,以生成 .NET Framework SDK 的安装文件)。

在安装 .NET Framework SDK 前,还应该将 IE 升级到 5.5 或者更高的版本。

下面是推荐的软件配置:

Windows 2000 Server + SQL Server 2000(或 Microsoft Access 2000) + IE5.5 + .NET Framework SDK

1.2.2 安装 .NET Framework SDK

有了 .NET Framework SDK 安装软件,就可以进行安装了。安装的过程非常简单,只需要按照安装过程中的提示,一步一步进行就可以了。在这里要特别提到 .NET Framework SDK 范例网页的安装。

在安装完 .NET Framework SDK 后,假如运行 .NET Framework SDK 的操作系统在 C 盘,并且没有改变安装程序的默认设置,那么 .NET Framework SDK 应该安装在下面的目录中:C:\Program Files\Microsoft.NET\FrameworkSDK。

下面是安装 .NET Framework SDK 范例网页的步骤:

(1)进入“C:\Program Files\Microsoft.NET\FrameworkSDK\Samples”目录,执行 Samples.exe 文件。

(2)如果读者已经安装了 Microsoft SQL Server 7.0 以上的版本,则直接进入第三步。如果没有,则进入“C:\Program Files\Microsoft.NET\FrameworkSDK\Samples\setup”目录,执行 InstMSDE.exe 文件。

(3)进入“C:\Program Files\Microsoft.NET\FrameworkSDK\Samples\setup”目录,执行 Finish.exe 文件。

(4)打开 IE,在地址栏中输入如下地址:<http://localhost/QuickStart/ASPPlus/default.htm>。如果页面能够正常显示,表示 .NET Framework SDK 范例网页已经成功安装。

如果读者的机器上已经安装了 Microsoft Office2000 套装软件,.NET Framework SDK 的安装会造成一些问题。当读者每次打开 Office2000 应用程序中的一个时,会出现一个要求注册的对话框。如果注册不成功,就会出现 Office2000 的 50 次使用限制。解决这个问题的方法就是预先备份 mso9.dll。假如读者的 Office2000 安装在 C:\Microsoft Office\文件目录下,此文件应该位于 C:\Microsoft Office\Office\目录下。在安装完 .NET Framework SDK 后,用备份的 mso9.dll 文件覆盖此时位于 C:\Microsoft Office\Office\目录下的 mso9.dll 文件,就可解决问题。如果读者事先没有备份,那么可以用