

.NET开发丛书

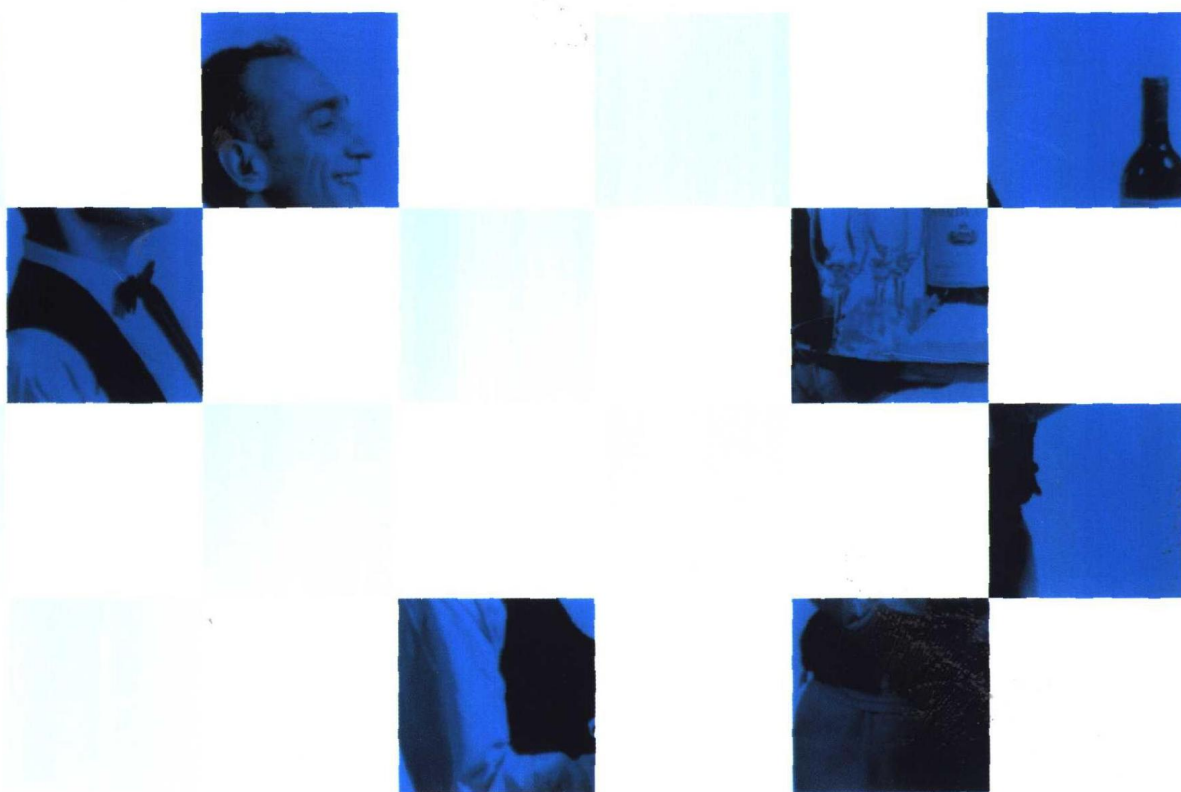


ASP.NET 基础教程

—— C# 案例版

Essential ASP.NET

with Examples in C#



(美) Fritz Onion 著
施 诺 译



清华大学出版社

.NET 开发丛书

ASP.NET 基 础 教 程

——C#案例版

(美) Fritz Onion 著
施 诺 译

清华大学出版社
北 京

内 容 简 介

本书结合用 C#语言编写的可实际运行的示例代码, 讨论了 ASP.NET 的构架、Web 窗体、配置、HTTP 管道、故障诊断和错误处理、验证、数据绑定、自定义控件、缓存、状态管理和安全性, 阐述了用 C#构建基于 Web 应用程序的最佳实践。

本书既可作为软件工程专业学生的参考书, 也可作为 C#程序员学习 ASP.NET 的参考书。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Essential ASP.NET with Examples in C#, 1st Edition by Fritz Onion, Copyright © 2003

EISBN: 0-201-76040-1

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-3651

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。

图书在版编目(CIP)数据

ASP.NET 基础教程——C#案例版/ (美) 奥尼恩著; 施诺译. —北京: 清华大学出版社, 2003

(.NET 开发丛书)

书名原文: Essential ASP.NET with Examples in C#

ISBN 7-302-07280-9

I. A… II. ①奥… ②施… III. ①主页制作—程序设计 ②C 语言—程序设计 IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字 (2003) 第 084069 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

文稿编辑: 汤涌涛

封面设计: 立日新设计公司

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 21.25 字数: 457 千字

版 次: 2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

书 号: ISBN 7-302-07280-9/TP·5285

印 数: 1~4000

定 价: 39.00 元

读者对本书的赞誉

★★★★★ 绝非一般的 ASP.NET 著作

到现在为止，我已经教了 1 年多 ASP.NET 课，我认为本书将会成为学生们重要的参考书。我喜欢作者切实而慎重的写作风格。阅读这两本书真是一件赏心乐事（另一本是《ASP.NET 基础教程——Visual Basic .NET 案例版》）。更重要的是，本书自始至终在解决学生们在学习 ASP 时最常见的问题。完全摆脱了一些书下笔千言、离题万里的弊病。本书还针对高级开发人员，强化了许多表面上松散实际上根本不同的概念。

对于 ASP.NET 开发人员，无论是初学者还是高级开发人员，本书都是一本不可或缺的参考书。特别地，在你动手编写了一些 ASP.NET 程序，希望了解内部机理时，会觉得本书更加有用。

感谢作者如此出色的工作！

——来自美国田纳西州的读者，摘自亚马逊网站

★★★★★ 极力推荐高级开发人员阅读本书

我之所以订购本书，主要基于网友们的推荐以及 .NET 开发丛书中的每一本书都写得很好的事实。本书确实没有令我失望。Fritz 在理论和实践方面结合得非常好。他假定读者熟悉 .NET，并且有应用程序设计经验（我也希望更多的作者做出这样的假设）。

本书提供的基础知识足以使高级开发人员做出某些最重要的设计决策，诸如何时以及如何分离自定义控件中的表示逻辑；如何管理状态；使用什么数据缓存选项等。

本书的流程安排的确无懈可击。技术的每个细节和特征都在恰当的场所引入。我是如此喜欢本书的流程安排，以至于我按照它的流程向我们的团队介绍了 ASP.NET。

——Alexis Smirnov，加拿大魁北克省，摘自亚马逊网站

★★★★★ 每个 ASP.NET Web 开发人员必备之书

Fritz Onion 编写的本书是我曾经看过的最好的 ASP.NET 教程之一。它没有在繁琐的基础知识上花太多的功夫,但是也没有完全忽略它们。Fritz 在平衡实现的细节和讨论之间做到了恰如其分。我曾经读过许多这样的书,其中作者为了说明一个问题而反复陈述相同的事情。Fritz 在这方面确实做的不错,而且他的书非常值得一读。你不会后悔阅读本书的。

对于一本书还能要求什么呢?本书讨论了 ASP.NET 开发的所有重要的主题:DataGrid、验证、HTTP 管道、ASP.NET 构架。所有这一切都讨论到了!每个现实的 ASP.NET 开发人员都能从本书中获益良多!

如果只能购买一本 ASP.NET 教程,那么我肯定会选择本书!

——James J Edelen IV, 美国, 摘自亚马逊网站

★★★★★ 卓越的教材

我是法国巴黎 5 所大学的副教授,正在为 .NET 课程做准备。

我购买了几本美国版和法国版的 ASP.NET 教材。Fritz Onion 的教材是惟一一本帮助我清晰地理解 ASP.NET 内部工作机理的书。

——Michel Futtersack, 法国巴黎, 摘自亚马逊网站

序

我之所以接触微软，明确地讲就是为了有机会从事 ASP.NET 工作。当时还没有提出 ASP.NET 这个名称，但是承诺要构建一种语言中立的、已编译的 Web 平台，对初学者非常友好，而且在功能和性能方面足以构建全球最大的 Web 站点。这种承诺引起我极大的兴趣，的确，继续从事 ASP.NET 工作已经成为一种令人神往、获益良多的旅程。

Web 平台建立在崭新的微软开发者平台——.NET 框架和公共语言运行库（Common Language Runtime, CLR）之上。该平台提供了大量服务和功能，Web 应用程序模型建立在它们之上。该平台令我们改变了许多游戏规则。例如，使性能接近已编译的本机代码已经成为可能，而又不会失去与脚本化环境关联的快速开发经验的好处。

ASP.NET 有一个宏伟的设计目标：成为一个可以开发动态内容并把它们发送到 Web 上的综合平台。必须承担的主要挑战是构建一个系统，能够吸引许多不同的背景和能力强的人，如 Web 开发者（用 Active Server Page 或者其他系统编写脚本）、Visual Basic 窗体开发者和 ISAPI 开发者。真正演变的是一个功能不断趋向完善的平台，利用个人已有的技能很快就可以变得非常高效，然后获取新技能以利用该平台的最新功能。

团队最初以 Active Server Pages 的诸多优点为基础，然后对它们进行扩展，不断地追求能使任务更容易的方法，而且用更少的代码表示出来。在良好的工具帮助下支持声明式设计（declarative design）是一个关键的设计目标。虽然此功能即将实现，但是还要时刻注意系统必须是可扩展的，而且支持许多现实世界中的、高度可扩展的 Web 站点要求的各种高级使用。

在开发 ASP.NET 的过程中，时刻挂在嘴边的一句话是“没有黑盒（No black boxes）！”。这是开发团队的目标，为之计划继续工作相当长一段时间，而且承诺实现一个分解的构架，可以进行扩展或者自定义以适应当前问题（而不管是什么问题）。因此，ASP.NET 核心原语是模块化的，且有一个相当可扩展的模型。

在本书中，你将会了解到扩展点在哪里，以及如何使用它们。Fritz 精心选择了关键概念，并说明了如何将它们组织成应用程序。真实的 Web 应用程序的组成部件都得到了很

好的表示：请求处理、页面和控件、配置、错误处理、安全性、缓存、数据表示及状态管理。

开发 ASP.NET 应用程序时，不需要理解大型的、复杂系统的全部。然而，当你开始构建更加复杂的具有挑战性需求的应用程序时，全面的基础知识及可靠的参考指南是必不可少的。

基于此因，我向大家推荐本书。本书非常适合作为一本 ASP.NET 的参考指南。它引领读者充分理解 ASP.NET 构架以及构建 Web 应用程序的核心原则。本书介绍了解决当前 Web 应用程序面临的许多现实问题时，必不可少的技术的更加高级的应用。我相信读者会认同本书的确是充分理解 ASP.NET 的基本指南。

Erik Olson

微软公司程序主管
华盛顿，雷德蒙

前言

这是在 2000 年 8 月的一个深夜，地点是加利福尼亚州南部城市托兰斯。这天，我已经同 Mike Woodring 和 Jason Whittington 上了 12 小时关于 DevelopMentor's Guerrilla（开发导师的游击战术）COM 课程。课后 Don Box 来到我这里，同往常一样，在学生们业已入睡以后，我们还在讨论有关的技术和黑客问题，一直到深夜。Microsoft 公司 7 月份在 PDC 刚刚发布了 .NET 的测试版本，而这一年我们已经花了大量时间来研究“下一代 COM”，我们为它的最终发布而倍感兴奋，因此我们可以讨论它了。就是在这样的一个晚上，Don 以他特有的简洁的方式，让我第一次见识了 ASP.NET（当时称为 ASP+）。他首先在 emacs 中输入一个 .aspx 文件，如下所示：

```
<%@ Page Language="C#" src="TestPage.cs"
    Inherits="TestPage" %>

<html>
<h1 runat=server id=ctl1/>
</html>
```

然后编写了另一个文件，如下所示：

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;

public class TestPage: Page
{
    protected HtmlGenericControl ctl1;
    void Page_Load(object src, EventArgs e)
    {
        ctl1.InnerText = "Hello!";
    }
}
```

然后把这两个文件放在机器上的 c:\inetpub\wwwroot 子目录中，并通过浏览器向我显示了 .aspx 页面的生成过程，然后大声对我说“明白吗”？也许是因为夜深了或者上了一整天课的缘故，但是我不得不承认，尽管我“明白了”Don 向我演示的问题的技术细节，然而

没有对能够从一个类中改变 h1 元素的 innerText 留下深刻印象。

经过两个晚上的休息后，我回想起那个.aspx 例子，并开始比较详细地研究 ASP.NET。经过一天的阅读和试验以后，我最终“明白了”，而且我已经迷上了 ASP.NET。这种技术的提出，从根本上改变了人们在 Windows 上开发 Web 应用程序的方式，而且它充分利用了最新的 .NET 运行库。接着，我用 6 个月时间研究和构建 ASP.NET 应用程序，并编写了 DevelopMentor's Essential ASP.NET 课程，在此后的一年半时间内，我一直从事 ASP.NET 的教学和讲座，以及撰写有关课程。本书是这些活动的结晶，希望能够在了解 ASP.NET 的过程中为你提供帮助。

C#与 VB.NET

在 .NET 推出之前，Visual Basic 不仅仅是另一种语言——它本身就是一个平台。例如，用 Visual Basic 6.0 构建应用程序，完全不同于用 C++ 和 MFC 构建应用程序。推出 .NET 以后，这种差别不再存在，而 Visual Basic 的确只是另一种 .NET 语言而已，与所有其他语言一样使用相同的函数库、相同的开发工具和相同的运行库。因此，我们现在可以从语言中立的角度谈论 ASP.NET 等技术。然而，代码示例必须用某种特定的语言表示，因此本书出版了两种版本：一个版本用 C# 编写例子，另一个版本用 VB.NET 编写例子。除了例子以外，这两本书的内容几乎是相同的。

示例代码、Web 站点、反馈

本书中的所有代码都取自可实际运行的示例，并且可以从本书的配套网站 <http://www.develop.com/books/essentialasp.net/> 上下载。本站点还包含本书出版后发现的所有勘误表，而且为本书中介绍的概念提供一系列更加丰富的例子供大家参考。作者欢迎大家通过该 Web 站点上的表单提出意见、勘误和反馈。

前提条件

本书专门讲述 ASP.NET，而不再花时间复习 .NET 编程、面向对象编程技术、数据库访问或者一般的 Web 应用程序开发技术。如果你已经花时间在所有这些领域中取得了经验，则可以获得本书中的大部分好处。

本书结构

本书从头开始研究 ASP.NET，首先在第 1 章中讨论 ASP.NET 构架的核心元素，然后在第 2 章中讨论服务器端控制模型。建议读者在阅读本书的所有后续章节之前，应先熟悉第 1 章和第 2 章的内容。然而，第 2 章以后的各章都是独立的，可以按你希望的任意顺序阅读。

第 1 章“构架”，讨论 ASP.NET 构架的基础，首先考虑.aspx 文件的解析以及随后将它们编译成程序集。然后详细解释了 Page 类，举例说明了最新的代码隐藏模型，并讨论了用来防止文件锁定的影子复制（shadow copy）机制。本章最后分析了 ASP.NET 中新增的类，它们取代了传统 ASP 固有的对象。

第 2 章“Web 窗体”，讨论 ASP.NET 所支持的基于控件的编程模型，即所谓的 Web 窗体。本章讨论使用 POST 主体数据和 ViewState 实现回送（post-back）的状态保持（state retention）的细节，并描述了如何有效地使用服务器端控件创建动态 Web 页面。本章最后讨论了 ASP.NET 中可用的各种服务器端控件。

第 3 章“配置”，描述 ASP.NET 所用的配置模型，首先讲述所有配置文件使用的 XML 格式，以及配置设置的层次化应用。本章将详细研究几个配置元素，包括 processModel 和 appSettings 元素。本章最后举例说明了两种在配置文件中添加自定义的配置节的方法。

第 4 章“HTTP 管道”，详细研究 ASP.NET 中服务 HTTP 请求所涉及的类。本章首先分析 HTTP 管道中用来处理请求的所有元素，然后讨论管道中的 3 个可扩展点：自定义应用程序类、自定义处理程序和自定义模块。本章最后讨论了管道中的线程和对象共享（object pooling）。

第 5 章“错误诊断和处理”，讨论 ASP.NET 的新的诊断特性，包括页面和应用程序跟

踪以及新的性能计数器。本章还讨论了用于调试 ASP.NET 应用程序和异常处理的技术。本章最后讨论如何为应用程序定义自定义的错误页面。

第 6 章“验证”，描述 ASP.NET 内建的新的验证构架。本章首先分析 Web 应用中一般是怎样执行验证的，然后说明 ASP.NET 的验证构架是怎样为验证用户输入问题提供一种通用方案的。本章详细分析了客户端和服务器的验证工作，而且还分析了所有可用的验证控件。

第 7 章“数据绑定”，分析将服务器端数据与 ASP.NET 页面中的控件进行绑定的过程。本章首先说明数据绑定在多种不同的数据源下（包括集合类、DataReaders 和 DataSets）是如何工作的，然后说明如何将数据与几个控件进行绑定，包括 DataGrid 类。本章最后讨论了模板，以及如何结合 Repeater, DataList 和 DataGrid 类有效地使用它们。

第 8 章“自定义控件”，讨论构建自定义控件以供 ASP.NET 应用程序使用的基础知识。本章阐明了如何构建自定义控件、如何使用 HtmlTextWriter 类实现独立的浏览器、如何进一步支持与浏览器无关的生成、如何定义属性和子属性、如何提取一个控件标记的内部内容、如何生成客户端脚本以及如何管理控件状态。本章还讨论了如何构建复合控件、用户控件、支持验证的控件以及支持数据绑定的控件。本章最后讨论了如何将控件与 Visual Studio .NET 设计器集成起来。

第 9 章“缓存”，讨论 ASP.NET 的输出缓存和数据缓存。本章讨论了输出缓存的机制，如何精确地控制把一个页面的哪个版本放入缓存中，以及如何以用户控件的形式用页片段缓存对一个页面的某些部分进行缓存。本章阐明了如何运用新的应用程序范围的数据缓存，并讨论了缓存数据时应当遵守的注意事项和准则。

第 10 章“状态管理”，讨论 ASP.NET Web 应用程序中各种类型的状态，并说明如何使用每种类型以及何时使用。本章首先讨论了应用程序状态，并说明了为什么在 ASP.NET 中通常应避开它。然后，讨论了会话状态的改进，包括进程外存储和无 cookie 密钥管理，以及优化使用会话状态的技术。本章最后讨论了用 cookie 和视图状态取代会话状态。

第 11 章“安全性”，描述了 ASP.NET 的安全特征，以及如何控制应用程序的客户认证和授权。本章首先回顾了 Web 应用程序的安全概念，然后说明了如何构建和管理需要用 ASP.NET 提供的窗体认证构架对客户进行认证的应用程序。本章还讨论了网络场（Web farm）中认证 cookie 的管理、安全的密码存储、构建基于角色的认证系统以及如何控制 ASP.NET 运用的过程同一性。

致 谢

首先感谢我的妻 Susan，我的孩子 Zoë 和 Sam，感谢他们在我撰写本书期间毫无保留的支持。还要感谢我的父母 Pat 和 Dan Onion，感谢他们的支持和指导。

感谢我在 DevelopMentor 的所有同事，无论在课程方面，还是在本书编写方面，都与他们进行了很多次讨论，总能从他们那里得到忠实的反馈意见。尤其要感谢 Bob Beauchemin，他总是及时地为我提供有用的反馈意见；感谢 Keith Brown 向我说明了如何删除无关紧要的信息，而要增强安全性一章；感谢 Simon Horrell 为本书提供的详尽反馈；感谢 Dan Sullivan，是他为我解决了众多难题；感谢 Ted Pattison，感谢他对我写作过程的关心和始终如一的明确的意见；感谢 Stu Halloway，是他使本书更加简洁；感谢 Mike Woodring，感谢他与我一同考虑同步处理程序蕴含的线程问题。

感谢我的评审：Justin Burtch，Amit Kalani，Daryl Richter，Martin Heller，以及 Matt Milner，他们都提供了宝贵的反馈意见。感谢 Microsoft 公司的 ASP.NET 团队的成员构建了如此有趣的一个产品，感谢他们为我解答了那么多问题。特别要感谢 Rob Howard 提供缓存部分的内容，以及 Erik Olson 在解释管道中线程分配和对象共享方面提供的帮助。感谢 Don Box 引领我进入了 ASP.NET 领域，在他帮助下我于 1995 年开始了写作生涯。

感谢我的编辑 Stephane Thomas，感谢他为本书付出的辛勤劳动；感谢我的文稿编辑 Debby English，不愧为 English 先生。还要感谢上千名上过 Essential ASP.NET 课程的学生：是你们提供的素材，而不是任何其他东西，形成了本书中的故事。特别要感谢 2002 年 10 月在华盛顿接受 ASP.NET 课程的学生们，是他们帮助选择了本书封面的颜色。

目 录

第 1 章 构架	1
1.1 基础	1
1.2 ASP 4.0	3
1.3 System.Web.UI.Page	6
1.4 代码隐藏	10
1.5 影子复制	16
1.6 指令	19
1.7 新的内部对象	24
1.8 小结	28
第 2 章 Web 窗体	29
2.1 服务器端控件	30
2.2 ViewState	34
2.3 事件	36
2.4 网页的生命历程	41
2.5 Web 窗体和代码隐藏	43
2.6 根路径引用语法	47
2.7 HtmlControl 控件	47
2.8 WebControl 控件	50
2.9 WebControl 控件与 HtmlControl 控件的区别	53
2.10 使用 Visual Studio .NET 构建 Web 窗体	54
2.11 小结	57
第 3 章 配置	59
3.1 web.config	60
3.2 配置数据	65
3.3 进程模型	67
3.4 附属设置	73
3.5 读取配置信息	74
3.6 创建自定义的配置部分的处理程序	77
3.7 小结	82

第4章 HTTP 管道	83
4.1 请求的生命期	83
4.2 上下文	86
4.3 应用程序	88
4.4 自定义处理程序	93
4.5 自定义模块	102
4.6 管道中的线程	112
4.7 小结	121
第5章 诊断和错误处理	123
5.1 ASP.NET 的诊断支持	123
5.2 调试	128
5.3 错误处理	130
5.4 小结	135
第6章 验证	137
6.1 窗体验证	137
6.2 验证控件构架	140
6.3 验证控件	145
6.4 小结	152
第7章 数据绑定	153
7.1 基础知识	153
7.2 数据绑定控件	155
7.3 绑定到数据库数据源	157
7.4 DataGrid	164
7.5 模板	175
7.6 小结	184
第8章 自定义控件	187
8.1 控件基础	187
8.2 状态管理	213
8.3 复合控件	221
8.4 用户控件	225
8.5 验证和数据绑定	226
8.6 设计器集成	233

8.7 小结	245
第 9 章 缓存	247
9.1 ASP.NET 中使用缓存的时机	247
9.2 输出缓存	248
9.3 数据缓存	262
9.4 小结	271
第 10 章 状态管理	273
10.1 状态类型	273
10.2 应用程序状态	275
10.3 会话状态	278
10.4 cookie 状态	291
10.5 视图状态	295
10.6 小结	297
第 11 章 安全性	299
11.1 Web 安全性	299
11.2 ASP.NET 的安全性	302
11.3 ASP.NET 中的系统标识	320
11.4 小结	321
术语表	323

第 1 章 构 架

ASP.NET 为 Web 应用程序开发者引入了许多新特性，这些特性包括已编译的服务器端代码、一种将服务器端逻辑与客户端布局相分离的代码隐藏（code-behind）技术、可扩展的服务器端控件模型、设计良好且易于使用的数据绑定模型、xcopy 部署、以及客户和服务端上的窗体身份验证支持。然而，除了上述的所有特性外，ASP.NET 还为我们提供了一种统一：语言、工具、库、部署模型、系统设计和诊断技术的统一。Web 应用程序开发者再也不用区别对待网页所用的组件与构架中其他地方所用的组件；再也不用借助于脚本调试器来诊断网页中存在的问题。他们可以摆脱许多不同规格的脚本语言（这些语言的细微之处经常使人迷惑）的束缚，使用他们拿手的.NET 语言构建网页。现在，在.NET 平台上构建 Web 应用程序犹如.NET 平台上的任何其他应用程序的开发。

本章介绍用 ASP.NET 构建 Web 应用程序的构架基础；讨论了用于处理请求的新的编译模型、如何构建代码隐藏类、如何用影子复制（shadow copy）机制实现 xcopy 部署，以及可用的新指令（directive）和内部对象（intrinsic）。

1.1 基 础

ASP.NET 的核心是许多.NET 类，它们相互间密切合作，对 HTTP 请求提供服务。在这些类中，一些类在系统程序集中定义，作为基础类库的一部分，并在.NET 运行时装入；一些类被载入全局程序集缓存（Global Assembly Cache, GAC），还有一些类从与应用程序关联的虚拟目录中的本地程序集中载入。所有这些类都被载入 ASP.NET 工作者进程内的应用程序域中，而且它们相互作用对给定的请求作出响应。图 1.1 显示了这一构架。

大多数开发者在从其他编程环境转向 ASP.NET 时都会面对基本的思想转变：所有要做的一切就是从一个程序集中装入一个类。类似于其他基于类的构架，在 ASP.NET 中构建应用程序的过程就是构造与基础框架中的类交互的类的过程。在用户编写的类中，有一些派生于基础框架中的基类，有一些可能会实现框架中定义的接口，而其余的类则只是简

单地通过调用框架中的基类的方法来实现与它们的交互。尽管 ASP.NET 还支持 ASP 风格的语法,但是 ASP.NET 文件中的服务器端代码被转化为类定义,并在第一次访问时被装入程序集中。因此,最终的结果是,许多类在一个进程中互相协作,对一个请求提供服务。

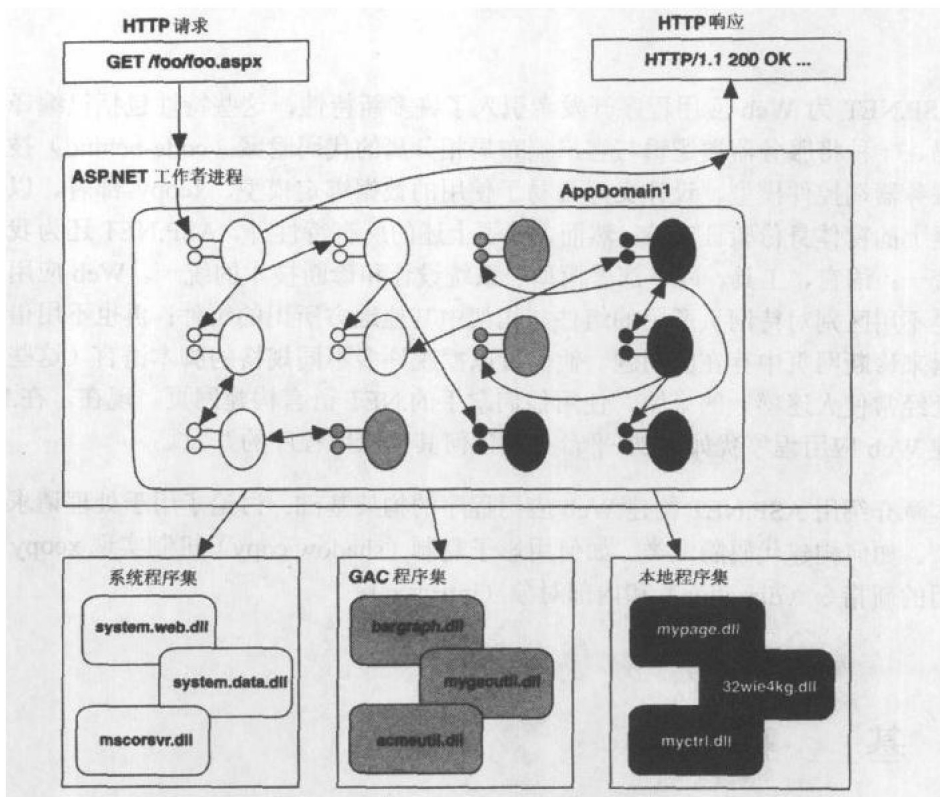


图 1.1 ASP.NET 的概要构架

另一个重大的转变是进程模型(process model)。ASP.NET 工作者进程(worker process)是从 inetinfo.exe (Internet 信息服务器(IIS)进程)中分离出来的一个单独的工作者进程——aspnet_wp.exe^①, ASP.NET 中的进程模型与 IIS 中有关的进程隔离设置无关。尽管 IIS 仍然是访问 ASP.NET 应用程序的典型入口点^②, 在物理上侦听相应的端口, 分发

① 在 IIS 6.0 (Windows Server 2003 带有该版本的 IIS) 中, 工作者进程模型有些不同, 我们将在第 3 章将对此进行讨论。这里值得注意的是, 如果不是以 IIS 5.0 隔离模式运行 IIS 6.0, 则 ASP.NET 驻留在一个称为 w3wp.exe 的进程中。

② 已有例子说明, 可以在没有 IIS 作为前端 Web 服务器的情况下使用 ASP.NET。Cassini 是由微软制作的一个样本 Web 服务器, 其完整的源代码可以从 <http://www.asp.net> 中下载。Cassini 与其他项目一起, 可以使我们在 Apache 中提供 ASP.NET 服务。