HUSTP

用实例学 ASP.NET

使用VB.NET与ADO.NET

华中科技大学出版社
http://press.hust.edu.cn

台湾微软资深顾问 章立民 著

用实例学 ASP.NET

——使用 VB.NET 与 ADO.NET

章立民　著

华中科技大学出版社
中国·武汉

图书在版编目(CIP)数据

用实例学 ASP. NET——使用 VB. NET 与 ADO. NET/章立民 著
武汉:华中科技大学出版社, 2003 年 4 月
ISBN 7-5609-2924-9

Ⅰ. 用…
Ⅱ. 章…
Ⅲ. 主页制作-程序设计
Ⅳ. TP393. 092

用实例学 ASP. NET——使用 VB. NET 与 ADO. NET 章立民 著

责任编辑:周 筠 封面设计:潘 群
技术编辑:韩 睿 责任监印:张正林

出版发行:华中科技大学出版社
武昌喻家山 邮编:430074 电话:(027)87545012

录 排:华中科技大学惠友科技文印中心
印 刷:湖北新华印务有限公司

开本:787×1092 1/16 印张:42.75 插页:2 字数:700 000
版次:2003 年 4 月第 1 版 印次:2003 年 4 月第 1 次印刷 印数:1—5 000
ISBN 7-5609-2924-9/TP・504 定价:59.80 元(含 1CD)

序

就在世界杯足球赛进入尾声之际，本书终于完稿。眼睁睁看着实力与丰采兼具的球队纷纷提早打道回府，虽然心中为之惋惜，但还是得忍痛接受。虽然我不是狂热的足球迷，但是每四年一次的世足赛，总是绝对不放过。然而或许是因为职业的关系，每当看着人们对足球的痴狂，心想有哪种开发工具能让程序设计师由衷慑服，甚至奉为圭臬与至宝呢？我想，恐怕很难吧！

记得几年前，出版社的朋友就一直鼓励我写 ASP 的书籍。说句老实话，当时 ASP 的开发架构与程序编写环境还真让我有点不敢领教。直到微软推出 ASP.NET 之后，才令吾人刮目相看，真正掳获了我的心。为了让众多同好同享 ASP.NET 的威力，这半年来，我们焚膏继晷、日以继夜地努力，终在近日将本书完成，希望能借本书引领更多读者 参与此一飨宴，跳脱过去那段跌跌撞撞的日子。

不可否认的，Visual Studio .NET 确实是革命性的开发工具，它的基础架构与开发环境更符合开发人员心目中的理想，此外，其前瞻性与未来性更是令人耳目一新。相信在未来的日子里，它将被更多的专业人员所采用，学会它，肯定对您有所助益。

章立民，笔于埔里

2002/06/24

感　　谢

本书之所以能够如期完成，要感谢洪志豪先生的大力协助，在此向他致上十二万分的谢意。

作者小档案

姓　名：章立民

出生地：南投县埔里镇

性　别：男

经　历：
- 台湾微软公司资深顾问与专业讲师
- 工业研究院机械所制造资讯部顾问
- 资诚会计师事务所资讯系统服务部顾问
- 捷和建设资讯部顾问
- 盘天科技顾问
- 一日志工协会资讯顾问
- ComputerDIY 杂志专栏执笔
- RUN!PC 杂志专栏执笔
- 曾任教于台湾空中大学、联电、药物食品检验局、调查局、南亚塑胶、NIKE、核电一厂、大众电脑、台湾日立等台湾省内各大公民营企业与教学机构，乃省内最权威之专业咨询顾问与讲师。

专　长：关系数据库管理系统，目前致力于研究 ADO.Net，ASP.Net，Visual Basic.Net，Visual C#，SQL Server 2000，XML，Access 2002 与 FrontPage 2002。将有全系列书籍问世，敬请读者关注。

目　录

第 4 章 万丈高楼平地起：Web Form 基本的语法与技巧

第 5 章 网页的基本元素：ASP.NET 服务器控件

第 7 章 数据检验的利器：验证控件

第 8 章 现有资源的快速集成者：用户控件

第 9 章 来去一瞬间：谈 ASP.NET 的状态管理

第 10 章 应用程序的关键：Global.asax 文件

第 11 章 效能提升的催化剂：ASP.NET 缓存

第 12 章 让一切更完美：ASP.NET 配置设置

第 13 章 非请勿入：ASP.NET 安全性

第 1 章

一窥究竟：ASP.NET 速览

ASP.NET 是一个统一的 Web 开发平台，它提供生成企业级 Web 应用程序所需的各种服务。虽然 ASP.NET 有很大一部分的语法与 ASP（Active Server Pages）兼容，但是它更提供了一个新的程序设计模型与架构，以便让您生成出功能更强大且完善的应用程序。ASP.NET 完全在 .NET Framework 的支持之下，此举使得我们能够善加利用 Common Language Runtime（CLR）、类型安全性、继承性，以及该平台的其他各项特性。本章我们将带领大家速览 ASP.NET，以便让诸位读者先有一个基本的认识与观念。

本章将讨论下列主题：

- ASP.NET 平台的系统需求
- ASP.NET 与 ASP 的主要差异

1-1 ASP.NET 平台的系统需求

ASP.NET 的服务器平台必须是 Windows 2000 或是已安装 Service Pack 6a 的 Windows NT 4.0。至于 Web Services，则在 Microsoft .NET Framework SDK 所支持的所有平台上都可以使用，但是 Windows 95 除外。Microsoft .NET Framework SDK 所支持的平台包括：Windows 2000、已安装 Service Pack 6a 的 Windows NT 4.0、Windows ME、Windows 98、Windows SE 与 Windows 95。

1-2 ASP.NET 与 ASP 的主要差异

如果仅仅说 ASP.NET 是 ASP 的新版本，那不仅过于笼统，也太小看 ASP.NET 了。事实上 ASP.NET 已经彻头彻尾地重新改造，它采用了全新的观念与服务器端技术来开发动态网页，足以适应未来数年或更长时间的开发需求。

ASP.NET 是一个以 Common Language Runtime 为基础所生成的程序设计架构，并且用来在服务器上生成功能强大的应用程序。ASP.NET 提供许多比过去的 Web 开发模型更加优异的功能，以下我们将详细介绍其各项重要特性，而这些特性不仅代表它与

ASP 的主要差异，您还可以从中体会 ASP.NET 的精妙之处，进而了解为什么要从 ASP 转移至 ASP.NET。

多语言支持

正如大家所知道的，ASP 仅能使用 Script 语言编写服务器端脚本，最常用的不外乎是 VBScript 与 JavaScript。欲使用其他的 Script 语言，您的电脑必须安装其解译器。但是无论如何，Script 语言始终是解译式的，解译式语言的两大缺点就是缺乏严谨的类型（例如像 Visual Basic 或 Visual C / C++所支持的类型语言）与不具备编译环境。虽然 ASP 会缓存代码，但它仍然采用解译方式，因而不可避免地会发生效率低落与扩展性不足的问题。

ASP.NET 是一个编译式的 .NET 环境，这意味着您不仅可以使用任何 .NET Framework 兼容的程序语言来编写 ASP.NET，还可以充分利用 .NET Framework 的 Common Language Runtime（CLR）、类型安全性、继承性与其他各项特性。显然您现在可以使用 .NET Framework 下的 Visual Basic .NET、Visual C# 或 JScript .NET 来编写 ASP.NET 而开发出更快速且可靠的动态网页。当然，我们并不期待您要了解每一种程序语言，正确的做法应该是挑选您最熟悉的程序语言。比方说，如果您非常熟悉 Visual Basic .NET，就应该使用 Visual Basic .NET 来开发 ASP.NET；又假设如果您对 Visual C# 情有独钟，当然可以使用 Visual C# 来开发 ASP.NET。

更快的运行效率

ASP.NET 不仅能够享受到 .NET Framework 与运行时在效率上的诸多强化，它本身亦经重新设计，使其运行效率大幅超越 ASP 与其他 Web 开发平台。所有的 ASP.NET 代码皆以编译过的公共语言运行时代码运行于服务器上，而并非采用解译方式，此举使得 ASP.NET 能够使用早期绑定、严谨类型化、实时（JIT）编译、原始编译与缓存服务来大幅提升其运行效率。

ASP.NET 也很容易加以拆解，亦即开发人员能够轻易将与他们所开发的应用程序无关的模块（例如：一个 **session** 模块）加以移除。ASP.NET 内附 Performance Counters，开发人员与系统管理员可以使用它来监控与测试应用程序。

缓存是 ASP.NET 非常重要的一项特性，并且能够大大帮助您生成高效能的 Web 应用程序。事实上要生成高效能且延展性佳的 Web 应用程序的重要因素之一就是要能够将相关元素在它们被取用后随即储存在内存中。您可以将这些元素储存在 Web 服务器本身或是取用通道中（例如：Proxy 服务器或浏览器）。如此一来，当一个访问请求与先前相同时，可避免再次生成信息而提升效率，当访问请求需要耗用大量的处理器

时间与其他服务器资源时，效率的提升更是明显。利用缓存技术，您可以将网页输出或跨越 HTTP 的应用程序数据访问请求储存起来，尔后再次使用时，服务器将不需再次生成它们，因此节省了时间与资源。

比方说，假设您的网页会去查询数据库并返回大量数据，并且需要花费 15 秒钟才能完成显示操作。在 ASP.NET 中，您可以加入以下的网页指令：

```
<%@ OutputCache Duration="60" %>
```

以上的 **@ OutputCache** 网页指令会使查询结果被缓存 60 秒钟，如此一来，当用户重新整理网页或另外一位用户探访它时，将会直接使用缓存中的网页而不用再次查询数据库。

ASP.NET 提供下列两种类型的缓存：

- **输出缓存**
  输出缓存允许您将访问请求所产生的动态网页与用户控制响应储存起来。如此一来，如果后续的访问请求与先前相同，将可直接使用缓存中的数据，而不需要再次动态运行网页或用户控制代码。
- **传统的应用程序缓存**
  借助此种缓存，您能够以程序控制方式将任意的对象储存至服务器的内存中，如此一来，您的应用程序便可省下重新生成这些对象的时间与资源。

世界级的工具支持

您可以在 Visual Studio 集成开发环境中使用丰富的工具箱与设计器来开发 ASP.NET，比方说，您可以使用具备“所见即所得”的编辑功能的 HTML 编辑器，将服务器控件拖放至网页，自动化开发等等。此外，您可以将自行开发的组件加入“工具箱”，使其在取用上与一般标准内置的控件完全相同。凡此种种，都将使您的开发效率大幅提升。

程序设计模型

当生成 ASP.NET 应用程序时，开发人员可以选用下列两种程序设计模型之一，抑或是将二者综合运用。

- Web Forms 允许您生成功能强大的窗体网页。当生成此类网页时，您可以使用 ASP.NET 服务器控件去生成公用的用户接口元素，并替它们编写程序来完成一般性工作。这些控件允许您利用可重复使用的内置或自定义控件来反复生成一个 Web 窗体，如此便能简化网页的代码。

- Web 服务是一种远程访问服务器功能的方法，它能够将数据或商业逻辑以程序化接口来呈现以便让客户端或服务器应用程序来访问。Web 服务使用 HTTP 与 XML 消息传递标准来跨越防火墙传递数据，打开了“客户端对服务器”或“服务器对服务器”的数据交换模式。Web 服务并未依附于特定的组件技术或对象调用协议，也就是说您所编写用来访问 Web 服务的程序能够使用任何的程序语言，使用任何的对象模型，并运行于任何的操作系统上。

上述两种做法皆能够善用 ASP.NET 的所有特性，以及 .NET Framework 及其 Common Language Runtime 的所有功能。

配置设置

ASP.NET 采用一个纯文本的分层式配置设置系统，以便能够轻易将各项设置应用至您的服务器开发环境与 Web 应用程序。由于配置设置信息是以纯文本的 XML 文件来储存的，故使用一般的纯文本编辑器（例如：“记事本”）即可加入新的设置。这种完全不需任何管理工具的做法反倒让我们很容易去部署 ASP.NET 应用程序。其实要将 ASP.NET 应用程序部署至一个服务器中，只需简单地将相关文件复制到服务器即可。值得一提的是，即使是部署或替换运行中的已编译代码也不需要重新启动服务器。

易用的状态管理

应用程序状态包含许多会影响应用程序的工作模式的信息与数据，包括：目录、购物车（shopping carts）、用户选项、确认清单、拜访人数（hit counters）等等。状态管理之所以可能变得非常复杂，主要原因是会有非常多不同的使用类型、数据类型及访问方法等信息可用来让程序设计师作为状态管理的基础。

在典型的 ASP 中，管理状态通常也就代表编写一大串的代码。尤其是当 ASP 应用程序需要使用在多部服务器中时，该项处理更加困难。状态值必须被写入隐藏字段并在网页间传递，抑或是写入数据库并在每一个网页视图中访问。

ASP.NET 的 **Application** 与 **Session** 状态管理不仅与 ASP 非常类似，而且能够轻易兼容于所有其他的 .NET Framework API，因此可以说是易学易用。

延展性与可使用性

ASP.NET 在设计之初就已将延展性纳入考虑范围，以便使其运行效率能够在聚集环境或多处理器的环境中有效提升。此外，各项处理亦会被 ASP.NET 运行期严密地监

控与管理，以确保当发生任何不正常的状况时（例如：死锁）一个新的处理会就地生成，这样的做法能够让应用程序持续运转并处理来自各方的工作需求。

自定义能力与扩充性

ASP.NET 允许开发人员将他们的代码嵌入 ASP.NET 架构的适当级别中。说得更明白些，也就是您能够以自己所编写的组件来替换 ASP.NET 运行期的子组件。这种扩充性将使验证与状态服务前所未有地简化。

安全性

NET Framework 与 ASP.NET 提供所有 Web 应用程序默认的验证与审核逻辑。您可以根据您本身的应用程序的需求方便地移除、加入与替换这些验证与审核逻辑。

数据库访问

ASP.NET 应用程序经常会去访问数据库并将数据显示在网页上，ASP.NET 提供前所未有的简易方式来帮助您完成此项操作，此外，它还提供管理数据库数据的功能。

应用程序逻辑

ASP.NET 提供了一个简易的架构来让程序设计师能够编写运行于应用程序级别的逻辑代码。程序设计师可以将此逻辑代码写在 **global.asax** 文本文件中，抑或是写在一个将以组件方式部署的已编译类中。逻辑代码可以包含应用程序级别的事件，而且程序设计师可以扩充该架构以便使其符合自身的 Web 应用程序。ASP.NET 完全支持您过去编写在 **global.asa** 文件中的 ASP 应用代码。当您将 ASP 移转至 ASP.NET 时只需将 **global.asa** 更名为 **global.asax** 即可。

兼容性

ASP 应用程序的语法与处理完全兼容于 ASP.NET。如果您想将您的 ASP 程序移植至 ASP.NET，只需将文件的扩展名从原来的 **.asp** 更改成 **.aspx** 即可。要替这些旧程序加入 ASP.NET 的功能也是轻而易举的，有时只需加入几行代码即可。

更加简易

在 ASP.NET 中，窗体提交、客户端验证、站点配置设置等许多例行性的操作都变得更加简易。比方说，ASP.NET 网页架构允许您生成能够将应用程序逻辑与展示代码清楚切割开来的用户接口，并以一个简易且类似 Visual Basic 的窗体处理模型来处理事件。除此之外，公共语言运行时使用自动引用计数与垃圾回收等托管代码服务来具体简化开发操作。

XML 与 SOAP 支持

XML（Extensible Markup Language：可扩展的标记语言）是一个纯文本的自我描述语言，在今日的开发环境中，XML 所扮演的角色已是无可取代的。不论您使用哪一种平台，XML 都能让您在一个开放格式的文本文件中打开、读取与储存数据，而使其成为一个传送数据的媒介。ASP.NET 与 ADO.NET 都使用 XML 作为数据的传送格式。其结果是，ASP.NET 能够与任何的数据源和既存的 COM 组件进行沟通与互动。

SOAP（Simple Object Access Protocol：简单对象访问协议）是一种以 HTTP 为基础的协议，用于交换 Web 上的结构与类型信息。一旦接收到数据，便会使用 XML 分析器将数据转换成 XML 文件，以便用户能够使用该数据。在过去，要生成该类型的应用程序是非常庞大且复杂的一项工作。

ASP.NET Web 服务分别使用 XML 与 SOAP 作为其数据格式与协议，使得要生成该类型的应用程序变成一件轻松事。

结束语

本章我们只是速览 ASP.NET 的各项重要特性，并借此让您了解它优于 ASP 之处。下一章我们将研讨 ASP.NET 的基本语法，请勿错过哦！