Fundamentals of Web Applications Using .NET and XML

用.NET和XML构建Web应用程序

ERIC BELL, HAO "HOWARD" FENG
EDWARD L.W. SOONG, DAVID ZHANG 著
SHIJIA "SAM" ZHU

夏江 译

- 展示 .NET Web 开发的完整过程
- 深入探讨 Web 表单、Web 服务、SOAP 和 XML 技术
- 提供从传统 Web 编程方式向 .NET 开发环境过渡的指导

清华大学出版社

用.NET 和 XML 构建 Web 应用程序

Fundamentals of Web Applications Using .NET and XML

[美] ERIC BELL
HAO "HOWARD" FENG
EDWARD L. W. SOONG
DAVID ZHANG
SHIJIA "SAM" ZHU
著

夏 江
译

清 华 大 学 出 版 社
北 京

EISBN 0-13-041790-4

图书在版编目（CIP）数据

用.NET 和 XML 构建 Web 应用程序/（美）贝尔等著；夏江译.
—北京：清华大学出版社，2003.8

书名原文：Fundamentals of Web Applications Using .NET and XML

ISBN 7-302-07164-0

Ⅰ.用 … Ⅱ.①贝…②夏… Ⅲ.计算机网络-程序设计 Ⅳ.TP393

中国版本图书馆 CIP 数据核字（2003）第 077239 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：成昊
文稿编辑：何武
封面设计：杨月静
版式设计：关静
印 刷 者：北京市耀华印刷有限公司
发 行 者：新华书店总店北京发行所
开　　本：787×1092　1/16　　**印张**：27.5　　**字数**：675 千字
版　　次：2003 年 9 月第 1 版　2003 年 9 月第 1 次印刷
书　　号：ISBN 7-302-07164-0/TP・5221
印　　数：1～4000
定　　价：43.00 元

内 容 提 要

您想迅速成为微软的.NET 开发人员吗？本书作者采用 step-by-step 的风格，以其前瞻性的产业视角及技术上的专长，借助大量代码示例，向您展示利用.NET 构造 Web 应用程序的全过程，为您铺就一条通往.NET Web 开发的捷径。

书中通过对传统 Web 开发过程和.NET Web 开发过程的全面比较，突出.NET 在创建高级 Web 应用和服务方面的优点，并重点介绍了主要的.NET Web 技术：ASP.NET、ADO.NET、Web 服务和 XML 编程，以及一些先进技术，诸如 SOAP、对象远程部署、系统集成等。最后给出一个实例分析，展示如何利用.NET Web 和 XML 技术创建一种新型的商业模型。

本书面向有经验但尚不熟悉.NET 的程序员，对于具有 VB，C/C++，JScript 和 Java 编程经验的读者，本书也给出了向.NET 技术过渡的实用指导。

前　言

一种崭新的、打破商业集成壁垒的计算模型正在悄然形成。在这种模型下，数据可以在因特网上自由交换，应用程序则可以通过因特网无缝地进行协作。基于 XML Web 服务，这一崭新的计算模型将把我们的因特网经济带入一个新纪元。蕴含了一系列新技术的微软.NET 正是适合于搭建、部署、运营和集成这些 XML Web 服务的平台之一。

作为 Prentice Hall 的 Object Innovations .NET 系列丛书之一，本书面向有经验但尚不熟悉.NET 的程序员，从实用的角度介绍.NET Web 技术的基本原理，使他们掌握利用.NET 编写 Web 应用程序的基础。本书首先回顾现有的 Web 技术；然后介绍.NET 体系结构；讲述主要的.NET Web 技术：ASP.NET、ADO.NET、Web 服务和 XML 编程；之后，本书还探讨了一些高级技术，诸如 SOAP、对象远程部署、系统集成以及未来的 Web 技术。最后，通过案例研究展示了如何利用.NET Web 和 XML 技术创建一种新型的商业模型。

对于具有 VB，C/C++，JScript 和 Java 编程经验，但也许没有 Web 编程经验的读者，本书也给出了向.NET 技术过渡的实用指导。

组织

本书第一部分首先介绍了.NET 世界中的 Web 站点（第 1 章），每个读者都应该阅读这一部分，它揭示了由应用和产品到 Web 服务这一崭新概念和实践的模式变化。接下来是对.NET 框架的介绍（第 2 章），.NET 框架是.NET 环境中所有应用程序和服务的基本结构。.NET 框架去除了传统软件开发和 Web 开发之间的区别，这是对业界的一大贡献。第 3 章探讨.NET 框架下的编程环境。在这一章中，你可以找到跨语言互操作的详细介绍和例子。在这一环境中，包括继承和多态等技术在内的面向对象编程打破了语言间的障碍。这一章还将介绍为什么.NET 框架是开发大规模分布式系统和 Web 应用的最佳选择。

本书的第二部分列举了利用 ASP.NET、ADO.NET、Web 服务和 XML 进行.NET 编程的主要功能。第 4 章是关于 ASP.NET 的完整综述。ASP.NET 就是基于.NET 框架的主动服务器页面，对比 ASP，ASP.NET 得益于 Web 表单编程模型以及服务器控件等新功能。通过一个完整的 Web 站点样例，这一章将展示如何搭建和部署 ASP.NET 应用以及如何处理那些真实环境中影响 Web 应用的一些问题，如会话状态以及安全性等。最后，我们简要地介绍一下 ASP.NET 的语法和从 ASP 过渡的途径。第 5 章介绍任何一种软件开发（无论是独立的应用程序或是 Web 应用）都要遇到的任务——访问数据。ADO.NET 是基于.NET 框架的主动数据对象，提供一种对各种不同的数据源进行一致且可扩展访问的方法。这一功能有助于提高一些设计目标，诸如互操作性、可维护性、可编程性和性能等。第 6 章介绍了 Web 服务这一新的计算模型的核心，包括如何搭建服务，如何由客户端进行访问等等。第 7 章介绍可扩展标记语言（XML）并提供一些样例程序。至此，你应该对使用.NET 平台进行 Web 应用开发有了相当程度的了解。

本书的第三部分将进一步介绍.NET 环境中的高级 Web 编程技术，如对象远程部署，简单对象访问协议（SOAP）等，同时还将介绍如何和其他平台进行交互。这一部分将向你展示如何基于对象的远程部署构建客户/服务器应用，并进一步构建足够灵活和强大的分布式系统（第 8 章）。第 9 章是针对那些不想利用 XML，但会直接编写基于 SOAP 的客户端的读者。当然，SOAP 将会帮助你搭建 Windows 和非 Windows 应用间的桥梁，有效地实现跨越整个因特网的应用间通信。是不是微软对开放系统的互操作支持太完美了，以至于你无法相信？第 10 章将向你介绍如何利用 SOAP，UDDI（通用描述、发现和集成）以及 WSDL（Web 服务描述语言）等技术实现这一看似不可能的任务。

第 11 章，也就是本书的第四部分，将演示本书的最终目标——使用.NET 进行 Web 编程。这一章本身是一个案例研究，它将教会你通过实践才能学到的知识。这一章向你展示如何通过因特网使用.NET 和 XML 以简洁的方式创建基于 Web 的分布式应用。

本书的附录是内容丰富的参考资料。附录 A 是针对没有 Web 编程经验的程序员的 Web 技术入门，它同时也可以为有经验的程序员提供快速参考。附录 B 讨论基于.NET 框架的 Visual Basic，比较新 VB 与旧 VB 间的差异（如果想了解有关 VB.NET 的更多内容，请参考“*Introduction to Visual Basic Using .NET* 和 *Application Development Using Visual Basic and .NET*”）。事实上，C#内容（附录 C）可以自成一章，但由于本书的主要内容并不是编程语言，所以只将其作为一个附录。（如果想了解更多内容，请参考本系列丛书中的“*Introduction to C# Using .NET* 和 *Application Development Using C# and .NET*”。）你还可以在附录 D 中学习有关 JScript 中的新内容。附录 E 介绍 Visual Studio.NET，微软最流行的 Visual Studio 开发环境的最新版本。新的 Visual Studio 有很多功能，可以使你应用开发更容易更愉快。了解这一环境有助于你的.NET 编程。

样例程序

学习一种语言的惟一方法就是不断地阅读和编写程序，包括一些相当规模的程序。本书提供许多小程序，它们分别展示了 Web 编程的相关功能，而且非常易于理解。

此外，本书还有一个案例研究，它展示了如何使用.NET 和 XML 编写 Web 应用。该案例研究对于理解本书的内容尤为重要。

所有样例程序均以自展文件的形式提供。当文件被展开时，在目录 c:\OI\Web 下会创建一个新目录。样例程序分别在 Chap1，Chap2 等目录中。Prentice Hall/Object Innovations .NET 系列丛书中其他书籍的样例程序在 c:\OI 下的其他目录中，所有这些丛书的样例程序均在你所安装的目录下。

Web 站点

Prentice Hall/Object Innovations .NET 系列丛书还有一个 Web 站点：

www.objectinnovations.com/dotnet.htm，该站点提供一个下载本书样例程序的链接。

目　录

第 1 章　.NET 世界的 Web 站点

如果你想学习如何利用.NET 构建 Web 站点、控制远程应用，而且想通过例子来了解有关平台的内容，那么，本书就是为你而准备的。.NET 目前正以迅猛的势头涌向你身边的计算机，将来，不可避免地，它会出现在每一台 Windows 计算机中。

为了帮助你开始使用.NET 来构建 Web 站点，本书会对相关技术进行全面的介绍。但在你开始正式学习之前，本章首先向你展示“.NET 世界”发生了哪些变化，以及为什么会发生这些变化。本章将讨论：

- .NET 的分布式特性
- 目前的 Web 与.NET Web 间的比较
- 类似于“体验”这样的新概念

Web 站点可以而且将会以与今天不同的方式构建。在开始学习之前，先做一些指导也许会对你有所帮助。如果你想直接了解技术，可以进入第 2 章。

1.1　集成的 Web 站点

在.NET 出现以前，Web 站点大多是独立操作的。但某些网站不同，通过将其他 Web 站点集成到自己的网站上，它们形成了合作关系。这种现象在门户网站中最为常见，比如 Yahoo（www.yahoo.com）和 Snap （www.snap.com），这里不仅有天气预报还有新闻栏目。为了丰富门户的内容，这些门户网站获取许可可以使用或拥有这些服务，也就是成为它们自己的站点。这种行为可以称为集成或协作，或称之为协同工作。这种“分布式计算”具有很多优点，可以将网站的不同部分集成到另一个网站中，使它们一起运作，将不同的页面缝制在一起，而不是简单地给出页面的超链接。让网站在一起工作并且实现这些优点，正是.NET 的初衷。

但这种类型的协作比想象的要棘手得多。除了将两个或多个网站合在一起的技术细节外，还要考虑很多商业细节。必须筹备并提供服务，而且门户网站必须以特定方式推出这些服务。最后，其表现必须是无缝的。

对于 Web 开发人员而言，将不同网站的页面或部分页面集成在一起并不像日常工作那么简单，但大多数开发者都可以花一定精力做到这一点。不过，集成两个或多个网站并没有一定的框架，也没有标准，没有流行的工具集，更没有一本书籍完整地介绍这一主题。集成多个 Web 站点至今仍是需要自己动手解决的问题。

1.1.1　协作

首先的问题是就协作达成一个可行的定义。字典的定义很简单，“与他人一起工作”。

联网计算机通常都被认为可以使计算机在一起工作，但联网本身并不能达到这样的目标。联网为计算机共享数据提供了途径——这也是它能做的全部工作。在一起工作也就意味着每一台计算机都要为整个会话作出一定贡献，主动地完成它们各自的任务。

协作需要各方主动在一起工作，这意味着协作者以一种分布式的方式一起处理任务。每一台计算机按照所执行的任务提供其他计算机所需的数据。这种计算机间的协作是分布式计算的一种形式。关于分布式计算的理论和实践已有很多书籍，而且超出本书所讨论的范围。通过分布作业在若干计算机上构建 Web 站点或 Web 应用是分布式计算的一种形式。在.NET 出现之前，计算机信息技术界一直缺乏促进这种形式的有效机制。.NET 并不涉及分布式计算的所有形式，但它却非常适合于分布式计算中的一种关键形式——将多个应用有机地缝合在一起构成一个应用。这种分布式计算就是.NET 中的分布式计算，我们将其称为计算机间的协作。

现在我们对协作有了一个可行的定义，但为什么需要这种协作呢？原因就是它具有三点优势。首先，同时使用两台或多台计算机解决问题是更好地使用计算机的方法。这样，就可以充分利用很多计算机的空闲时间了。

其次，决定如何分解一个问题来进行分布式处理是很困难的。在计算机间分解一个问题需要花费一些工作，这样的过程尚不现实。.NET 的方法是重新描述问题，它提供了一个框架，该框架在问题某部分的解决方案出现之前就刺激它们的产生。.NET 用另外一种方式确定如何分解问题。不是空想问题并分解它，而是想象你需要什么，寻找各个解决部分并构成最终的解决方案。这是一种自里向外的思考方式。

第三，外包你的网站的某些部分意味着其他人会为那些部分的维护而操心。这是简单的经营之道。

1.1.2 协作的代价

这种协作的代价是采用.NET。这意味着你将依托这个平台，而且学习使用支持它的工具和服务器。就构成.NET 的所有产品而言，学习任务很艰巨。类库非常庞大，编程语言也是新发明的或为.NET 进行了修改，而且还有新工具需要学习和使用，并且需要集成新的服务器。本书讨论所有这些问题[1]。关于服务器，这里仅介绍如何使用它们而不是如何管理它们。比如，ASP.NET 是因特网信息服务器（IIS）和 SQL 服务器（微软的工业级的数据库服务器）所支持的技术，所以本书关于 ASP.NET 的章节将使你能够启动并运行它，而无需考虑管理 IIS 的细节。尽管这种依托是可以预期的，但需要说明的是，由于.NET 的成功取决于人们是否接受这种变化，微软正在努力吸引开发人员使用这个平台，并正在改进.NET 使其更适合于在公司部署，但人力的代价始终都是存在的。

1.1.3 体验

由微软勾勒的“体验(experience)”一词正逐步成为人们的口头禅。许多文章都暗示.NET

1. 关于类库的讨论参见第 2 章和第 3 章。一个类库是一个预先编写好的代码的集合，你必须使用它们来利用.NET。关于 C#，受控 C++，VB.NET 和 JScript.NET，参见附录 B，C 和 D。

将预示着一个软件易用性的新纪元，它将丰富人们的日常生活，带给他们新的体验。

同时，计算机并不能隐藏你必须了解如何操作它们的事实。像朝圣者朝拜圣殿一样，你必须坐在计算机旁，知道如何操作它，并从中获得你所需要的内容。通常，如果有足够多的程序，一台计算机可以运行成千上万个程序，你必须一一学习如何操作它们才能从中获益。这种学习内容非常广泛，从不同软件包相似操作的不同命令到管理硬盘所需的操作。计算机需要用户屈从于它们的需求。

人们逐渐需要计算机做更多的工作。在浏览器中保存 Web 页面就是这样的一种需求。首先，你必须选择菜单项来保存页面，其次，你要为页面选择位置而且可能还需要更改页面的名字。这是大多数软件的一种常见行为模式。一旦熟悉了这些，你就接受了这种重复行为并且可以不假思索地执行它。如果要在硬盘上为 Web 页面寻找存放的位置，你可能仅仅需要点击几下鼠标或敲击几下键盘。当你保存若干个文档时，才遇上真正的麻烦，你不得不在每次保存页面时在硬盘中寻找同一个文件夹，程序无法记住你存放上一个文档的位置。这是一次相当难堪的体验。

作为 Web 用户的你对体验的理解非常重要。体验一词——直接观察事件——现已成为术语。微软使用体验来表示“你对事件感觉如何”，这种感觉会因人而异。.NET 的体验对用户与软件交互的方式非常关注。同一软件包由于它的外观、工作原理、操作速度以及其他更多因素的不同，也会带来完全不同的体验。

让个人计算机更易于使用一直是一种推动力量，让使用 Web 站点的细节更不易于察觉的工作也从未中断。最终，我们希望完全不用考虑技术细节。对技术考虑越少，效率就越高。这就是为什么计算机新手往往觉得不自在，甚至是害怕的原因。计算机会带给新手们前所未有的体验。计算机新手会花费大量的时间来将他们的需求转换成计算机的需求。这需要大量的工作且是一个容易出错的过程。

用户期望良好的体验。如果你不得不苦思冥想如何使用某个网站，那么你很可能不会使用这个网站。与某个网站进行交互必须相对比较简单。能够提供良好的体验正是.NET 浪潮的一部分，尽管可能还未发现对此有所支持。至今，.NET 平台尚没有工具、API 或者支持（或实施）体验的编程机制，也没有很多的文献可供参考，但这个词却在很多不同的地方出现，而且会越来越普遍。支持创建、维护和实施体验的工具最终会出现。

1.1.4　站点间协作

为了让 Web 站点间可以协作，必须有适当的标准机制。那些不能彼此协作的设备将使我们的生活变得一团糟，因为没有机制可以使它们在一起工作。一种成功的技术如果能为他人所用，那么它才有生命力，而且，如果有标准可以遵循的话，则会更好。

超文本标记语言（HTML）就是这样的机制，它可以让全世界的每个人都参与 Web 页面的编写。同样，其底层的超文本传输协议（HTTP），也就是 Web 服务器用来通信的协议，也是如此。HTML 主要用来描述如何表达数据，但不提供操作这些数据的方法。比如，货币对你和我可能意味着不同的内容，或者，你我可能按照不同的方式书写日期。渐渐地，像 HTML 一样流行的可扩展标记语言（XML）粉墨登场了。XML 是一种描述数据的通用方法。