

Visual Basic Developer's Guide to ASP and IIS

# Visual Basic

## 开发指南

# — ASP和IIS篇

〔美〕 A. Russell Jones 著

王 军 孙永强 等译

●ASP(Active Server Pages)和IIS (Internet Information Server)是微软的Web开发工具，本书指导你利用Visual Basic、辅以ASP和IIS等工具开发功能强大的服务器端Web应用程序。



电子工业出版社  
Publishing House of Electronics Industry  
URL: <http://www.phei.com.cn>

*Visual Basic Developer's Guide  
to ASP and IIS*

# Visual Basic开发指南 ——ASP和IIS篇

〔美〕 A. Russell Jones 著

王 军 孙永强 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 提 要

这本书讲解如何使用Visual Basic来开发Web应用程序。它详细而深入地介绍了使用VB开发Web应用程序过程中所遇到的各种问题和所使用的新技术,包括WebClass、ASP、IIS、MTS、ADO等新技术,以及安全性、调试和错误处理、维持状态等实际问题。

本书共有十四章。第1章到第4章包括介绍性的内容,描述用VB开发Web应用程序相关的一些概念和技术。第5章到第13章结合实际应用中的问题展开讨论,每章讨论一个问题,并通过小的项目进行演示。第14章是作者的一个展望,提出了作者对一些新技术的独到见解。本书的重点是使用Visual Basic开发环境最新提供的WebClass来开发功能强大的服务器端的Web应用程序,包括WebClass、安全性、状态维持、存取数据库等内容。

本书适合于熟悉Visual Basic、HTML和Web的开发人员,如果你是其中的一员,那么这本书会极大地促进你的开发能力。



Copyright©1999 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版,SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

### 图书在版编目(CIP)数据

Visual Basic开发指南——ASP和IIS篇/(美)罗塞尔(Jones, R.)著;王军等译.-北京:电子工业出版社,2000.6

书名原文:Visual Basic Developer's Guide to ASP and IIS

ISBN 7-5053-6016-7

I. V… II. ①罗… ②王… III. Visual Basic语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2000)第63455号

书 名: Visual Basic开发指南——ASP和IIS篇

著 作 者: [美] A. Russell Jones

译 者: 王 军 孙永强 等

责任编辑: 陈 宇

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河金马印装有限公司

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036 电话: 68279077

北京市海淀区翠微东里甲2号 邮编: 100036 电话: 68207419

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 16.25 字数: 400千字

版 次: 2000年6月第1版 2000年6月第1次印刷

书 号: ISBN 7-5053-6016-7  
TP·3137

定 价: 27.00元

版权贸易合同登记号 图字: 01-2000-0345

凡购买电子工业出版社的图书,如有缺页、倒页、脱页,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。

我把这本书献给我的父母，Russell和Wanda Jones，他们不懈的支持和毕生对学习的热爱激励了我并使这本书得以完成。还有我的妻子，Janet，她总是在倾听我的诉苦并在适当的时候提出明智的语言上的建议。

## 致 谢

我想感谢InterCom公司的Bob Yeager和Margo Pearson，是他们使参与新技术成为我的主要工作。不是每一个人（实际上，很少有人）都能得到这种机会，因此我真正地感激这次机会。我想感谢《Visual Basic程序员期刊》的Susannah Pfalzer，是他建议我编写这本书。我还想感谢Sybex编辑队伍的杰出工作，特别是Denise Santoro、Colleen Strand、Sharon Wilkey以及Greg Guntle。他们的耐心指导使得编写本书成为一次宝贵的经历。

还有一些人极大地影响了我与这本书，我也想借这次机会感谢他们：Donald Pederson，是他把我带进编程世界；Charles Leonhard和Esther Steinberg，是他们给予我大量知识和建议；G. David Peters，他是我的一个榜样；以及David Northington，是他教会我如何实践。

最后，我想感谢你购买了这本书。我希望能从本书中学到大量知识，并且阅读时能够得到和我写作时一样多的乐趣。

A. Russell Jones

# 前 言

作为**Visual Basic**程序员，我们看到了从单个的、单片的可执行程序向分布式、基于组件的体系结构的明显转移。这种转移带来了思想上的改变。我们需要把应用程序看作是一组工作在一起的组件来提供所需要的解决方案。

随着技术的改变，我们不断修改我们的组件体系结构以支持这些改变。随着开发基于**Web**应用程序的需要日益增长，下一步是把我们所知道的基于组件的开发和今天的基于**Web**的技术结合起来。像超文本标记语言（**HTML, Hypertext Markup Language**）、活动服务器网页（**ASP, Active Server Pages**）以及**Internet**信息服务器（**IIS, Internet Information Server**）应用程序（也就是**WebClass**）之类的工具必须添加到我们的工具箱中。

但是作为**Visual Basic**程序员，我们要如何适应这些工具？我们需要学习**HTML**吗？我们能够建立被**ASP**调用的组件吗？我们能够使用最喜欢的工具**Visual Basic**来建立**Web**应用程序吗？

本书将回答这些问题。作者**A. Russell Jones**，从**HTML**的简单介绍开始，然后他给出**ASP**和**IIS**应用程序的例子，并且详细讨论它们来帮助我们加快速度并跟上今天的技术。

**Deborah Kurata**

**InStep**技术公司（[www.insteptech.com](http://www.insteptech.com)）

《**Doing Objects in Visual Basic 6.0**》的作者

## 绪 论

这本书向你展示如何使用Visual Basic (VB) 来进入Web, Web是一种计算模式, 正迅速取代所有旧的建立程序的模式。这种取代并不是因为建立Web程序比建立标准网络程序更加容易(虽然在某些方面是这样), 也不是因为Web更加快捷(Web并不快), 或更加坚固(Web并不坚固——至少目前是这样)。这种取代是因为Web是一种更好的模式。

作为开发者, 我们将会看到出现一些新的适合这种新模式的语言。实际上已经有这样的语言——Java就是最好的例子。因为语言像文明一样崛起、昌盛和衰落, 所以我们将会看到其他一些语言的消失。仍具有充分活力的语言将被修改以适应新的模式, 就像过去被修改以适应其他模式的改变一样。

VB就是这些适应性语言中的一个。Basic深深扎根于个人电脑的起始时代。它随着PC的成长而成长, 获得速度和能力。Basic能够新的编程特性出现时加入它们而得到发展, 这些特性有: 命名子程序, 函数, 结构化编程, 强变量类型, 长变量名, 可选参数, variant和对象。当微软在1990年发布VB时, 它就迅速地成为创建Windows程序的首选方法。窗体, 事件驱动编程, 以及最重要的能够容易地封装市场上的第三方控件中的有用功能, 使得VB成为历史上最通用的编程语言; 而且VB还没有停止发展。

该语言的最新版本, 版本6, 把VB扩展到一个新的领域: Web编程。在这个过程中, 微软向开发人员提供了一种把他们的专长和VB的全部开发能力——包括它的调试器——应用到新的计算模式上的方法。

编程范围的增长几乎和计算机本身能力的增长同样迅速。仅仅几年之前, 开发人员的大部分目标还是运行于单个计算机上的程序; 应用程序的代码和数据全部保存在硬盘上。大部分的程序只用几张软盘就可以装下。为了共享信息, 人们把数据从一台计算机拷贝到另一台计算机。几乎就在这个模式变得通用时, 小型网络就开始普遍存在了。虽然它们开始的用处限制在扩展存储上, 而不是计算能力上, 但是人们开始了解在许多人之间共享数据能够提高商务的响应性。这个能力是如此的吸引人, 以至于很快大部分的商业程序都开始提供从单独PC对商务信息的访问。

这种客户访问中央数据的计算模式驱动了早期客户端-服务器应用程序的开发。含有所有的用户界面代码和所有的商业规则的客户端能够连接到一个中央数据存储上以让商务人员接近实时地访问公司的数据。不幸的是, 这个模式是昂贵的, 因为你必须把客户连接到一个专有网络上使用应用程序。另外, 所有的网络提供商都具有不同的、专有的方法来让远程客户连接到它们的网络上; 实际上, 网络通常是不兼容的, 因为计算机没有标准的通信方法。访问数据库和在屏幕上显示信息的代码是专有的。没有标准的界面。每一个公司和应用程序具有不同的击键动作和菜单项。程序员以不同的方式编码每个屏幕。没有下拉菜单、移动光标或剪切和粘贴信息的标准按键。你根本不能在程序之间剪切和粘贴信息; 而你必须使用纯文本文件来导出和导入信息。所有程序都具有专有的文件格式。你不能混合和匹配程序

组件，因为没有在语言之间进行通信的标准方法。甚至没有计算机打印文件的标准方法——你必须为你使用的每一个程序安装一个驱动程序。

就在混乱之时，**Windows 3**出现了，尽管它的界面有时令人困扰，但它在两年之内几乎占领了市场上每一台计算机的桌面。**DOS**时代过去了，**Windows**成为了王者。更进一步，**Windows**带来了一些标准：标准菜单，标准窗口界面；标准按键；每个打印机（而不是每个应用程序）有一个驱动程序；标准网络接口；在程序内和在程序之间都有剪切和粘贴的能力。命令行界面接近消失；可视化界面赢得了胜利。但是在所有这些美好现象和易于使用的背后是一个极度复杂的编程模式，全部都需要用**C**来编写。作为一个程序员，如果想要进入强大的**Windows**世界，则必须要学习**C**。幸运的是，**VB**及时出现了，它使我们远离困境并提出了一个新的编程模式——可视化编程。你不用再花费多个小时来计算像素值，重新编译，以及检查对象在屏幕上的位置，而只需在设计时把对象拖放到合适的位置上。现在，很难找到一个声称自己是“非可视化”的编程环境。

是的，**VB**引发了一场编程革命，但这场革命不是可视化编程，而是代码重用。代码重用说了20多年，但**Visual Basic**使代码重用成为了一个现实。**VB**通过推出基于**VBX**的控件来实现代码重用。**VBX**文件是特殊的**DLL**，它具有一个可视组件，你可以拖放这个组件到一个窗体上。这些**DLL**所封装的功能可以覆盖：从简单的文本控制，到数据库连接，到整个文字处理器，以及它们之间的每一件事物。因为**VBX**功能强大，易于使用，并且节省大量时间，所以围绕着它们的一个完整的领域迅速发展。

**VBX**不久就迅速地让位于**OCX**，**OCX**又迅速地让位于**ActiveX**控件。到**Visual Basic**版本5，你可以创建自己的**ActiveX**控件。更加重要的是，在版本5中，微软完全重写了**Visual Basic**并加入了对类和接口的支持。

最近，**VB 6**引入了一种特殊的称作**WebClass**的类，它被用来和**Internet**信息服务器（**IIS**，**Internet Information Server**）以及微软的活动服务器网页（**ASP**，**Active Server Pages**）技术一起工作。在和微软事务服务器（**MTS**，**Microsoft Transaction Server**）一起使用时，**WebClass**是一个用于建立强大的可扩展的**Web**站点的有力工具。

## 我为什么写这本书？

我之所以写这本书是因为我相信我们正处于一场在很多方面都比**PC**的发明更加重要的计算机革命之中。我从1979年就开始写计算机程序。除了使用**VB**，我还使用过很多语言（有**C**，**SmallTalk**，以及几种“写作”语言，例如**ToolBook**和**TenCORE**）来开发基于计算机的训练程序和企业应用程序。在过去的几年里，我开始写**Internet**程序——最初是**DOS CGI**程序，然后通过**ISAPI DLL**运行的**VB**程序，最后使用**Visual InterDev**和**ASP**。我对**VB**的新功能深感激动，因为我认为这个语言正在简化**Web**应用程序的开发。

迄今为止，开发**Web**应用程序的最好形式是格式化代码（**HTML**）与后端编程的笨拙结合。只有客户请求格式化内容页面并且服务器提供了它们之后，服务器和客户才能很好地相互通信。虽然格式化内容非常有用，但却丢失了一个有效的计算组件——用户和计算机之间的交互。静态网页还不够。**Web**开发现在正处于能够开始把一些交互性添加反馈给客户这个阶段。

另外，还没有出现能够帮助开发一个完整Web应用程序的所有组件的单一环境，这些组件包括：客户界面，商务逻辑，以及数据访问组件。我们已经拥有：HTML编辑器，例如HotMetal Pro、Hotdog和FrontPage；CGI类型的语言，例如Perl；客户端脚本语言，例如JavaScript和VBScript；以及很多把事情捆绑到一起的定制解决方案，包括InterDev和ColdFusion。然后还有Sun微系统公司，它推出Java作为每件事情的解决方案，包括为你运行烤面包机。Java是一个很棒的语言，但它的语法却不尽人意。幸运的是，Visual Basic开发小组已经前进了一步，他们通过提供一个统一的开发环境来使我们免于在各种语言之间切换，在这个开发环境中，我们可以把可运行于任何浏览器中的基于HTML的前端连接到一组运行于一个或多个服务器上的VB类和对象上面。

## 你应期望从本书中学到什么？

建立客户端-服务器应用程序需要与建立单机应用程序不同的模式，与此相同，为Web建立应用程序需要另外的模式。在某些方面，我们回到了主机时代，当时一个“聪明的”主机向一些终端提供和格式化数据。随着终端变成了PC并且变得聪明了，程序设计人员分解了这个模型。他们将处理从中央计算机移到客户端，但是数据（公司需要在许多人之间共享这些数据）仍位于中央计算机上。

不幸的是，分解后的客户端模型具有它自己的问题。客户端不会全部相同，因此程序经常需要重写多次以使它们能够在不同的客户端硬件配置上运行。当Windows变成主要的客户端平台时，它通过为客户端提供一个访问网络和外围设备的通用方法而稍微改善了这种情况。但是，仍有太多的不同使得客户端-服务器模型很难维护。

Web重新推广了连接多个远程终端的聪明的中央处理单元——Web服务器的思想。但是远程终端现在成了PC——不同于老的终端——它们具有足够的能力来有效地处理图形显示。中央处理单元不再必须是昂贵的主机系统，而可以是相对便宜的PC服务器，或者更好的是连到一起提供请求信息的服务器集合。

本质上，一个成功的Web应用程序的关键在于用何种方法在服务器上存储信息，以能连续而又迅速地提取这些信息并将其送交给客户。如果不是这样，服务器将会因无法执行请求而陷入困境，客户因此将被迫等待，于是应用程序将会失败。在这本书里，我将提供一个模型，它能够使你建立一个不需要在结构上做明显改变就可以从工作组扩展到企业级的应用程序。我也将帮助你利用编程模式的另一个改变——不是新的，但在互连的Web编程世界中却有额外的重要性——这就是基于组件的编程。

## 谁应该看这本书？

我将此书定位于那些熟悉VB、HTML和Web的开发人员和程序设计人员。这本书不致力于VB编程，也不会教你很多HTML，DHTML或建立静态Web站点的知识。而它向你展示如何使用已有的VB技能来创建基于服务器的Web应用程序。使用这些新工具所创建的应用程序的种类由你决定。



如果你是一个想进入Internet编程世界的VB程序员，这本书将帮助你设计或重新设计你的应用程序以使它们能够成功运行。这本书不会教你Winsock编程的低级细节或其他的连接协议，如HTTP、FTP或TelNet。但是，它将向你展示如何使用已有的知识——你最精通的编程语言VB以及已有的基础结构——这样，你就能继续创建有用的、响应性好的、强健的程序。

微软使用活动服务器网页（Active Server Pages）技术把VB程序连接到它的Web服务器上——Internet信息服务器（在NT上）和个人Web服务器（在Windows 95/98上）。这本书不需要你精通ASP，但是如果你已经熟悉ASP编程，这样更好——你就可以跳过讨论ASP对象模型的部分。实际上，你将会看到ASP模型如何紧密地集成在VB中。

## 为什么选择书中的这些项目？

这本书包含两种类型的信息：

- 有关Web的知识框架，你可以使用VB或其他任何语言在其上建立应用程序
- 代码范例，以项目的形式给出，你将在大多数章节里建立这些项目

很多书都是通过在每一章节里建立一小部分应用程序，最后得到一个完整的应用程序的途径来讲述其内容。这个方法对作者很有用，因为它帮助作者讲述一个故事并且在各章之间保持连续性。我没有选择这种方法，主要是因为我发现这种项目在我的工作中起不到什么作用。相反，我提供了一系列小的项目，这些项目含有可重用的代码和思想，可以直接应用于你的项目中。有些项目仅仅是为了说明一个要点，而不是因为它们本身有什么作用——如果你能够透过现象看本质，而不是试着让它做一些有用的事情，则你可很容易地看出代码背后的推论。

这种小型的可重用代码段在很多方面都可以作为Web应用程序建立过程的中心，Web应用程序主要是由小型的、几乎独立的程序所组成。

## 你应具备哪些知识？

本书不是一本面向初学者的书。要想最大程度地掌握这本书，你必须熟悉VB。特别地，你应该熟悉类结构，VB的事件模型，以及COM。你不需要知道很多有关Internet的知识，虽然拥有一些HTML知识可以使你更加容易地阅读代码，建立项目，以及理解例子。

你应该熟悉关系数据库，或者是Access、或者是SQL Server之类的ODBC数据库。你还应了解ActiveX数据对象（ADO，ActiveX Data Object）；当然，如果你只使用过数据访问对象（DAO，Data Access Object）或远程数据对象（RDO，Remote Data Object），那么你在理解代码时也不会遇到多少问题。

## 需要哪些硬件/软件？

你将需要下列软件以完成本书中的项目：

- Visual Basic 6，专业版或企业版

- Microsoft Access或SQL Server 6.x及其更高版本
- Visual InterDev或其他好的HTML编辑器

建立这些项目，你需要Visual Basic 6的专业版或企业版。你也将需要Microsoft Access或（最好是）SQL Server 6.x及其更高版本。如果你使用Microsoft Access来建立网站，你将会遇到一些麻烦。有两个原因。首先，Microsoft Access引擎不是线程安全的；因此，如果使用Access作为Web应用程序的数据库，你将会遇到麻烦。虽然你不会马上遇到这种问题，但是如果你长时间地使用Access，这种问题就会出现。第二，我曾多次尝试过它，但它却没有很好地工作。因此，如果你想最初的开发中使用Access，这将没有问题。只是不要使用它来发布应用程序。在你准备对程序进行Beta测试时，将Access转换成Sybase，SQL Server，Oracle——哪种都行，只要不是Access。

本书中的一些（只是一些）项目使用SQL Server附带的Microsoft pubs数据库。如果你没有SQL Server，你可以从Sybex站点下载一个含有pubs数据库表格的Access数据库。

**说明：**要想下载代码，访问<http://www.sybex.com>。单击“Catalog”按钮并搜索本书的标题。单击“Downloads”按钮并接受许可协议。接受许可才能访问本书的下载页面。

你也将需要Visual InterDev（强烈推荐）或者一个HTML文本编辑器——我同时使用Visual InterDev编辑器和记事本。我也曾用过Homesite，Hotdog，HotMetal Pro，以及几个其他的编辑器，并且很喜欢它们。使用哪个编辑器没有关系，只要能够用它来阅读和编辑HTML。你并不需要一个功能强大的HTML编辑器来建立本书中的项目。你将需要稍微钻研HTML代码，本书将帮助你做到这些。

Web主要是一个文本文件的世界：带有标记的文本文件，带有代码的文本文件，含有XML的文本文件，等等。其思想就是，在一个标准的世界里，每个人都能够阅读其他人的文件格式。幸运的是，你不用对此担心太多，只要你坚持使用“标准的”HTML，不使用DHTML和ActiveX控件，并且为特定的浏览器使用了正确的语法，你所编写的几乎任何东西都会在任何浏览器中很好地工作。欢迎来到跨平台的世界。这对你是否很普通？不用担心，事情并没有我所描述的那样坏。在任何情况下，你都能使用几乎任何文本编辑器来编辑不同种类的内容：文本，格式化文本，数据库内容，代码。Web对于文本文件编辑器的制造者是很仁慈的。

## 本书中有哪些种类的代码？

在本书中你将看到至少三种代码：Visual Basic（或Visual Basic for Application，VBA）代码，VBScript代码和Access-SQL或Transact-SQL代码（虽然不是很多）。本书中的所有例子代码，如果是ASP中的脚本，则采用VBScript；Visual Basic模块或类中的代码则使用VB/VBA；而当代码应用于数据库时，则采用Access-SQL和Transact-SQL。

## 译者序

随着技术的发展，应用程序开发已经进入了**Web**环境，并且这也是未来的发展方向，它必将取代传统的客户机－服务器环境。同时，作为世界上最流行的开发环境，**Visual Basic**也紧跟技术的变革不断地调整自身，新增了适合开发**Web**应用程序的许多特性。这本书带领读者从头开始，使用**Visual Basic**开发环境在微软平台之上开发**Web**应用程序，因此，在紧跟新技术的同时，又紧紧地抓住了拥有最大用户群的开发工具，真是明智之举。

市面上有很多自称完全参考手册的书籍，它们所讨论的内容覆盖面很广，但是每一个主题都没有深入，所以实际使用价值不大。和这些书籍不同，本书的定位非常明确，它紧紧抓住使用**Visual Basic**开发**Web**应用程序这个主题来深入地进行讨论，并结合实际情况详细地解释和这个主题相关的每一部分内容，同时还给出了许多小的项目来帮助你理解。这本书讨论的最多的内容是**WebClass**，这个类是**Visual Basic 6.0**新增加的，用来与**IIS**和**ASP**技术一起协同工作。本书所讨论的主题有：**ASP**对象模型，如何建立**IIS**应用程序，**WebClass**，安全性，调试和错误处理，维持状态，使用**ADO**存取数据库，在**WebClass**中使用**ActiveX DLL**，**MTS**，部署应用程序等等。这些主题都和**Web**应用程序紧密相关。

本书的作者**A. Russell Jones**是一位经验丰富的专家级程序开发人员，同时又是一位专栏作家。他对本书中的每一个主题都有自己独到的见解。在本书中，他结合自己的经验对每个主题深入浅出地进行讨论，彻底给出每个问题的前因后果。相信读者在阅读本书时会时常发出“原来如此”的感慨。

本书原文内容丰富，论述透彻充分，非常适合于具有一定的**VB**开发经验同时又想快速进入**Web**世界的开发人员使用。在翻译过程中，译者忠实于原著的内容和特点，付出了辛勤的劳动。相信本书一定会给读者带来很大的收获。

全书由王军和孙永强翻译。在翻译过程中，译者力求准确地反映原著的内容，但由于水平有限，加上时间紧迫，错误之处在所难免，望广大读者批评指正。

译者

2000年3月于北京

## 目 录

|                                   |    |
|-----------------------------------|----|
| <b>第1章 Visual Basic与Web</b> ..... | 1  |
| 什么是Web应用程序？ .....                 | 2  |
| 为什么用VB编写Web应用程序？ .....            | 2  |
| IIS与DHTML应用程序有什么区别？ .....         | 3  |
| <b>第2章 IIS应用程序</b> .....          | 5  |
| 浏览器如何请求文件 .....                   | 5  |
| 如此多的请求，如此少的时间 .....               | 6  |
| ASP对象模型 .....                     | 7  |
| Web应用程序与客户端-服务器应用程序的比较 .....      | 13 |
| <b>第3章 创建ASP应用程序</b> .....        | 16 |
| 理解ASP应用程序的结构 .....                | 16 |
| 使用包含文件 .....                      | 17 |
| 理解语言无关性 .....                     | 18 |
| 在ASP中使用Cookie .....               | 19 |
| 使用Scripting.Dictionary对象 .....    | 21 |
| 简单介绍HTML和表单 .....                 | 24 |
| 创建一个自修改的ASP应用程序 .....             | 26 |
| 综合使用——缓存表格数据 .....                | 34 |
| <b>第4章 WebClass介绍</b> .....       | 44 |
| 理解WebClass如何工作 .....              | 44 |
| 理解HTML模板 .....                    | 47 |
| 不使用模板编写HTML .....                 | 48 |
| 研究WebClass事件序列 .....              | 51 |
| 优化，优化，优化 .....                    | 52 |
| 比较VB/WebClass与ASP的优缺点 .....       | 55 |
| <b>第5章 安全化一个IIS应用程序</b> .....     | 58 |
| 理解安全性如何工作 .....                   | 58 |
| 创建工程 .....                        | 58 |
| 创建一个HTML模板 .....                  | 61 |

---

|                                   |            |
|-----------------------------------|------------|
| 添加一个自定义事件 .....                   | 66         |
| 添加一个自定义WebItem .....              | 68         |
| 测试安全性 .....                       | 76         |
| 编译工程 .....                        | 77         |
| 改进站点安全性 .....                     | 80         |
| 利用NT安全性 .....                     | 81         |
| <b>第6章  WebClass调试和错误处理 .....</b> | <b>83</b>  |
| 研究WebClass事件序列 .....              | 83         |
| 调试客户端脚本 .....                     | 91         |
| 记录和显示错误 .....                     | 93         |
| 捕获错误 .....                        | 97         |
| <b>第7章  控制程序流程 .....</b>          | <b>105</b> |
| 理解程序流程 .....                      | 105        |
| 创建AccountInfo工程 .....             | 106        |
| 改用WebClass .....                  | 114        |
| 在一个应用程序中使用多个WebClass .....        | 118        |
| <b>第8章  在IIS应用程序中维持状态 .....</b>   | <b>130</b> |
| 选择一个状态维持方法 .....                  | 130        |
| 研究状态维持方法 .....                    | 140        |
| 缓存信息 .....                        | 151        |
| 总结状态维持 .....                      | 159        |
| <b>第9章  处理多个浏览器 .....</b>         | <b>160</b> |
| 理解不同浏览器的区别 .....                  | 160        |
| 确定浏览器的类型 .....                    | 161        |
| 编写浏览器特定的代码 .....                  | 166        |
| 用代码编写代码 .....                     | 169        |
| <b>第10章  在数据库中检索和存储数据 .....</b>   | <b>171</b> |
| 理解ADO .....                       | 171        |
| 介绍ADO对象模型 .....                   | 172        |
| 用ADO检索数据 .....                    | 173        |
| 从WebClass中访问数据库 .....             | 186        |

|                                           |            |
|-------------------------------------------|------------|
| <b>第11章 从WebClass中使用ActiveX DLL .....</b> | <b>190</b> |
| 把WebClass看作粘合剂 .....                      | 190        |
| 从WebClass中访问ActiveX DLL .....             | 191        |
| 和WebClass一起使用ObjectContext对象 .....        | 191        |
| 从一个WebClass中测试ActiveX DLL .....           | 193        |
| ActiveX对象的持久保存 .....                      | 197        |
| 建立一个基于浏览器的代码库 .....                       | 203        |
| <br>                                      |            |
| <b>第12章 IIS应用程序和微软事务服务器 .....</b>         | <b>213</b> |
| 什么是MTS ? .....                            | 213        |
| MTS如何适合IIS/Web应用程序模型 ? .....              | 215        |
| MTS如何帮助你 ? .....                          | 217        |
| 何时应该使用MTS ? .....                         | 218        |
| 如何使用MTS组件 ? .....                         | 218        |
| 如何把一个工程移植到MTS中 ? .....                    | 223        |
| <br>                                      |            |
| <b>第13章 部署WebClass应用程序 .....</b>          | <b>230</b> |
| 准备部署 .....                                | 230        |
| 创建一个安装软件包 .....                           | 233        |
| 配置目标服务器 .....                             | 235        |
| 处理权限问题 .....                              | 237        |
| <br>                                      |            |
| <b>第14章 下一步是什么? .....</b>                 | <b>239</b> |
| Web应用程序的方向在哪里 ? .....                     | 239        |
| 关于Java和CORBA .....                        | 241        |
| 关于数据库技术 .....                             | 242        |
| 关于XML .....                               | 243        |
| 结论 .....                                  | 246        |

## 第1章 Visual Basic与Web

- 什么是Web应用程序？
- 为什么用VB编写Web应用程序？
- IIS与DHTML应用程序有什么区别？

虽然Visual Basic已在Windows程序设计中风靡一时，但是迄今为止它还不是Web程序设计领域的主流语言，除了用于外围开发的VBScript（微软选用Visual Basic家族中具有挑战性的小子集VBScript，作为活动服务器网页（ASP，Active Server Pages）技术的缺省编程语言）。为什么VB没有成为Web程序设计的主流语言呢？有若干原因。

首先，Visual Basic只能运行在Windows平台上。直到微软的Web服务器，即Internet信息服务器（IIS，Internet Information Server）开始流行（大约在3年前，随着IIS 3.0的发布），VB才能够与Web服务器直接交流。这就为用VB开发Web应用程序提供了强壮的机制。

其次，无论何时你运行VB应用程序，即使是已编译过的VB应用程序仅有几KB代码，你都必须加载VB的一个大型运行时动态链接库（DLL）。这就使得VB程序在Web服务器上运行极其缓慢。每当客户端浏览器请求一个Web页面时，Web服务器都不得不忠实地装载运行时DLL、装载应用程序、处理请求，然后再卸载该应用程序和运行时DLL。因为Web只是处理一个进入、获取数据、快速离开的调度，所以加载一个大于1MB的应用程序支持DLL以提供一个很小的HTML文本，是极度的浪费。此外，在每次请求完成后应用程序将终止，因此在多个请求之间存储数据值是困难的——这意味着这种方法主要是用于格式化简单请求和数据库报表，而并不适用于全部应用程序。

为解决这些问题，微软提供了一个应用程序编程接口（API），即Internet Sever API（ISAPI）。通过ISAPI，程序可以与Web服务器通信。但是，微软没有提供VB访问ISAPI的方法。微软还“解决”了大型的支持DLL问题，其方法就是允许你把DLL直接装载到Web服务器地址空间中（毕竟，Visual C++也有同样的问题）。这样一来，这些DLL始终保持装载后的状态，可极大地加速响应时间。虽然ISAPI立即风靡，但是直到Visual Basic 5提供创建编译后的DLL功能，你才能够运用ISAPI。不幸的是，这种能力极大地降低了Web服务器的稳定性，因为如果某个进程内的DLL失效，Web服务器通常会因此而崩溃。

与此同时，出现了连接VB应用程序与Web服务器的一些其他方法。一种免费DLL，叫作对象链接和嵌入Internet服务器应用程序编程接口（OLEISAPI和后来的OLEISAPI 2，Object Linking and Embedding Internet Server Application Programming Interface），它提供了VB应用程序和Web服务器之间的连接。此外，相对完善的解决方案不仅提供了这种连接，而且还可以装载可配置的多个应用程序实例，提供这些实例之间的负载平衡，并使你能够将一个实例和某个用户连接起来，以便程序可以在服务器上保存多个请求之间的用户专门数据。这些解决方案还能够对服务器的多个客户端请求之间保留程序实例，因而你可以很容易地在程序变量中存储数据。

此外，你还能够直接对底层协议编程。Carl Franklin所编写的《Visual Basic 4.0 Internet 编程》一书极其出色。在这本书中他讨论了Winsock编程，多种Internet协议的使用，以及Internet的其他一些较低级的编程主题。所有这些信息现今仍然非常有效，并且如果你想深入研究，Carl的书（现已改版到VB 6）会对你更有帮助。

幸运的是，随着VB 6的发布，微软最终实现了Visual Basic与Web的直接连接。他们所提供的方法就是通过贯穿一个或多个ASP页面的组件对象模型（COM，Component Object Model）自动化来进行工作。你建立一个显露WebClass对象的DLL。客户端调用一个ASP页面，其装载此DLL的一份拷贝。从此以后，你就可以使用WebClass方法、属性和事件来把模板文件中的或临时创建的内容发送到客户端浏览器。

此时，如果你不能够理解上述所有内容，不必担心——读完本书，你就会理解。重要的是记住WebClass不是一个新技术，理解它们也并不难。它们所能做的事情，你使用Visual Basic 5和一个或多个ASP页面同样也能做到——它们只是使编程更加容易。

## 什么是Web应用程序？

Web应用程序不同于Web站点。Web站点所提供的信息都是预先建立好的并且能够存储在静态的超文本标记语言（HTML，Hypertext Markup Language）文件中。Web站点中的信息主要是从服务器传送到客户端。当用户必须输入信息时，服务器会提供一个一般的、固定的响应。在多个请求之间，服务器不会在意客户端所做的事情。客户端可以从一个地方跳跃到另外一个地方而不会影响Web站点，因为每个页面都是一个独立的单元；Web是由分离的多组超文本文档组成。与此不同，Web应用程序用来提供由单个用户或一组用户专门检索的和格式化的信息。信息的传送是双向的——用户的输入或身份通常决定了浏览器上所显示的内容。

Web应用程序提供动态信息，而不是静态的HTML文件。这种应用程序通常从一个数据库服务器上抽取所需要的内容。Web应用程序不仅提供信息，而且还接受你的信息并且以特定的自定义响应来响应你的行为。

Web应用程序很在意你从一个请求到另一个请求之间所做的事情。它必须跟踪你的活动以服务于你的需要。在Web站点中你可以一直来回跳跃，而在Web应用程序中你却不可以。有时，你需要从开始到结束一直遵循某个过程（例如，填写一个在线应用程序），否则的话，整个过程都不可信而必须放弃。

## 为什么用VB编写Web应用程序？

你可以用多种语言来编写Web应用程序，例如：Perl、Python、Java、C、C++、Quick Basic或Unix Shell Scripts。那么为什么要选用VB来编写Web应用程序呢？每当你需要作出决定时，首先总是要考虑需求。那么Web应用程序有什么需求，而VB又如何能够很好地满足这些需求呢？Web应用程序有以下5个主要需求：

**数据库连通性** Web应用程序通常要累积和管理关键信息，并把它存储到一个数据库中以备将来引用。它还使用这些信息存储来提供特定服务。因此，Web应用程序语言应



该擅长存储和检索数据库中的信息。自从第三版本以来，VB就有行业中领先的数据库连通性并且目前仍然是这样。

**速度** Web应用程序通常必须服务于很多客户。被编译成本机代码的Visual Basic有接近于C语言的速度。

**字符串处理** Web应用程序必须能够处理字符串搜索、连接及标识置换。Visual Basic具有卓越的字符串处理能力。

**安全性** Web应用程序必须能够处理安全性需求。任何一种编译语言都比解释语言有更好的安全性。Visual Basic和C一样也能够满足这种需求，并且比Java或脚本语言（例如VBScript或JavaScript）更好。

**事务** Web应用程序通常需要执行多个操作，这些操作必须成功完成以保持数据的有效性。当使用一个数据库时，通常你会将这些操作封装在一个事务中。事务就是一个契约，它保证整组操作要么成功，要么失败而不改变数据。事务的典型例子是检查帐户。当你存入一张支票时，银行从一个帐户中取出一定数量的金额，并把相同数量的金额存入另一个帐户中。这两个行为要么都成功，要么都失败。取出和存入的双重操作就是一个事务。通过微软事务服务器（MTS，Microsoft Transaction Server），你能够在你的Web页面中执行多个操作，甚至可以使用多个对象来完成事务。虽然你不总是需要事务，但是一旦你需要它们，则就是非常迫切的。

除了以上5个主要的需求外，还有其他一些不太重要的需求：

**熟悉程度** 如果你熟悉某种语言，你可以更快地编写出更好的程序代码。因为你是Visual Basic程序开发人员，所以在这里你已走在竞赛的前列。

**调试程序** 如果你有一个功能强大的调试器，你可以编写出更好的程序代码。Visual Basic和Visual Studio在这方面处于领先地位。查看并替换变量内容、临时重写代码、移动执行指针、单步执行DLL项目中的代码、单步执行ASP文件中的代码以及单步执行浏览器中的代码等能力——所有这些均来自Visual Basic项目组——都是卓越的调试能力。在微软去年发布Visual Studio 6之前，ASP调试只能通过Response.Write语句和Response.End语句来完成，这类似于在Visual Basic中你只能使用Debug.Print和Stop语句来调试。

**代码/对象重用** 当然，你能够用任何一种语言来实现这种需求，但是我敢断定没有任何一种语言能够像Visual Basic一样有如此多的第三方工具、库和例程。本书中也包含了大量的例程和对象，并且在适当的时候你可以重用它们。大部分例程在基于VB的Web应用程序中都可以正确地运行，故再次重申，你有可能正处在竞赛的前列。

正如你所看到的，Visual Basic不仅适合于其他类型的程序开发，同样也适合于Web应用程序的开发——它是一种通用语言并能够很容易地满足这些需求。

## IIS与DHTML应用程序有什么区别？

Visual Basic有两类内置的Web工程：

**IIS应用程序** 这些应用程序运行在Web服务器上，并运行在微软的Internet信息服务器（IIS，Internet Information Server）之下。利用IIS工程，你可以为任意类型的浏览器（甚至无浏览器）客户端编写Web应用程序。