特别赠送

本书实例和素材文件

Visual Basic 2005 基础与实例教程

郝春强 编著

- **内容全面** 涵盖了Visual Basic .NET程序设计的所有知识点
- **实例丰富** 从语法讲解到应用程序开发全部使用实例来讲解，让读者在实践中掌握所学知识
- **零 起 点** 从最基本的语法到数据库编程、网络编程、Web服务等进行了非常透彻地讲解
- **入门快捷** 采用知识讲解+实例练习+课后习题的教学方式，是初学者学习Visual Basic .NET进行程序设计的首选书籍

中国电力出版社
www.infopower.com.cn

Visual Basic 2005 基础与实例教程

郝春强　编著

中国电力出版社
www.infopower.com.cn

内 容 简 介

本书是一本 Visual Basic 2005 入门图书，书中以清晰的概念讲解和大量的示例相结合的方式，详细介绍了使用 Visual Basic 2005 进行程序设计的方法与技巧。除了对 Visual Basic 语言进行详细的讲解外，还对基本控件、消息框、对话框、工具栏、菜单栏等的使用，以及程序调试和异常处理，文件管理与注册表操作，数据库编程，网络编程和 Web 服务等进行了透彻地讲解。

本书内容由浅入深，语言简洁，示例丰富，非常适合 Visual Basic 初学者学习，也可作为高等院校和相关培训班的教材。

图书在版编目（CIP）数据

Visual Basic 2005 基础与实例教程 / 郝春强编著.北京：中国电力出版社，2007.8
ISBN 978-7-5083-5659-4

I. V… Ⅱ.郝… Ⅲ.BASIC 语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2007）第 088764 号

责任编辑：夏华香
责任校对：崔燕菊
责任印制：李文志

书　　名：Visual Basic 2005 基础与实例教程
编　　著：郝春强
出版发行：中国电力出版社
地址：北京市三里河路 6 号　邮政编码：100044
电话：（010）68362602　传真：（010）68316497
印　　刷：航远印刷有限公司
开本尺寸：185 × 260　印 张：21　字 数：513 千字
书　　号：ISBN 978-7-5083-5659-4
版　　次：2007 年 8 月北京第 1 版
印　　次：2007 年 8 月第 1 次印刷
印　　数：0001 — 4000
定　　价：32.00 元（含 1CD）

前　言

2000 年左右，微软开始了一个宏伟的战略，这个战略在很多场合被微软的高层和 20 世纪 80 年代的图形界面战略相提并论，这个战略就是.NET。.NET 是微软面向未来互联网的战略，微软把公司未来发展的赌注放在了.NET 上，如同 20 世纪 80 年代微软在图形界面上下的赌注一样。

.NET 战略所引发的重大变革同样在起初遭到各种非议和怀疑，但最终同样被证明是微软的又一次伟大创造。历经数年的发展，.NET Framework 从 1.0 到 1.1 再到 2.0 直到目前最新的 3.0，.NET 已经发展成为构建企业应用程序最重要的平台之一。

.NET Framework 是 Microsoft 为开发应用程序创建的一个富有革命性的新环境。伴随着.NET 的出现，微软对 Visual Basic 语言进行了一次革命性的重塑以适应.NET Framework，.NET 时代的 Visual Basic 语言被重新命名为 Visual Basic .NET，不应该把它看作是 Visual Basic 6.0 的简单升级，而应该把它看作是从 Visual Basic 语言演变而来的，一种为高效生成类型安全和面向对象的应用程序而设计的全新语言。目前 Visual Basic .NET 的最新版本是 Visual Basic 2005。

相对 Java（1995 年）、C#（2001 年）等年轻态语言而言，Visual Basic 可谓是源远流长，它的发展历史可以追溯到 40 多年前。

20 世纪 60 年代，计算机高级程序语言并不像今天这样琳琅满目，当时已经出现的高级语言有 Fortran（1954 年）、Algol（1958 年）和 Cobol（1961 年）等，尽管这些语言本身取得了很大的成功，但其固有的繁琐性使得程序的编写依然是一项艰难的工作。

为了让程序设计和学习程序设计的过程变得更加轻松和愉快，20 世纪 60 年代早期，Dartmouth 学院的 J. Kemeny 和 T. Kurtz 开始设计一种新的语言。他们奉行的理念是尽可能地简单和尽可能地接近自然语言（英语）。这种新的语言被命名为 BASIC，BASIC 是 Beginner's All-purpose Symbolic Instruction Code（初学者通用符号指令代码）的缩写。因为其简单、易学，BASIC 一经问世就成为国际上广泛使用的一种计算机高级语言，至今仍是计算机入门的主要学习语言之一。

历经 40 多年的发展，BASIC 语言的发展历程可以归结为以下 5 个阶段：

第一阶段：（1964 年～20 世纪 70 年代初）1964 年 BASIC 语言问世。

第二阶段：（1975 年～80 年代中）计算机上固化的 BASIC。

第三阶段：（80 年代中～90 年代初）结构化 BASIC 语言。

第四阶段：（1991 年～2001 年）Visual Basic。

第五阶段：（2001 年～今）Visual Basic .NET。

Visual Basic .NET 是 Visual Basic 在.NET 平台下的最新发展，也是一次革命性的变化。之所以称其为革命性变化，是因为首先.NET 平台是一个革命性的变化，Visual Basic .NET 用

于构建面向.NET 平台的托管代码，而不是像传统 Visual Basic 一样构建面向 Windows 操作系统的代码；其次，传统 Visual Basic 的目标是创建 Windows 客户端应用程序，而 Visual Basic .NET 则旨在创建 XML Web 服务应用程序（也可以创建 Windows 客户端、控制台等其他类型的应用程序）。Visual Basic .NET 的这种革命性的变化甚至使得微软放弃了与传统 Visual Basic 程序的兼容，大量的传统 Visual Basic 创建的代码必须经过改动才能在.NET 平台上成功运行。

Visual Basic .NET 现在是一门现代的、强大的、面向对象的、简单的可视化开发语言。本书将从.NET 与 Visual Basic .NET 的基本概念讲起，循序渐进、由浅入深地介绍了使用 Visual Basic 2005 进行应用程序开发的各方面知识。

开发工具同样重要，Visual Studio 2005 是微软.NET 平台上的一个功能强大的、集成多种开发语言的最新软件开发工具。通过该开发工具，大多数.NET 编程语言都可以实现 RAD（快速开发）。本书中所有代码都是在 Visual Studio 2005 中编写的，书中有专门的章节介绍 Visual Studio 2005 的使用。

本书特色

（1）本书定位于快速入门级，阅读本书前不需要读者具有任何 Visual Basic 方面的基础知识，甚至可以是对编程技术一无所知的新手。因此，本书非常适合作为自学教材。

（2）本书通过清晰的概念介绍与大量的代码示例相结合的方式，来介绍使用 Visual Basic .NET 语言进行程序设计的基础与方法，并且书中所有的示例代码均位于本书的配套光盘中，便于读者在学习本书的同时查看和运行示例。

（3）本书每章的最后都给出了一些思考与练习题，通过对这些问题的思考以及编码实践，可以进一步明确概念和掌握编程技巧。在本书的附录中，提供了所有思考题的参考答案，对于编程练习题，本书的配套光盘中给出了示例程序，通过参考研习这些示例，对快速提高编程水平很有帮助。这也是本书的一个特色，即配套光盘不是对书中内容的重复，而是书盘互为补充，这就使得本书的内容很充实。

（4）本书试图让读者在学习 Visual Basic .NET 语言的同时，能够掌握面向对象编程技术的一般思想和方法。

致谢

除了本书署名作者外，还有如下同志在本书的写作过程中给予了帮助和支持，他们是聂晓玲、王晓丹、骆成凤、滕石欣、郝春琴、袁立明、齐晓莉、宗海燕、尚志有、李治富、罗慧、张巧荣、郝春文、杜美玲、王玉河、郝春娜等。

再次诚挚感谢我的家人，本书写作时，正值我的新家装修之际，是他们为我分忧解难，才使得我能有更多的写作时间。

感谢中国电力出版社给予了我编写本书的机会，并提出了很多有创意的建议，使得本书的内容更充实、更实用。

郝春强

2007 年 6 月于永丰

目　录

第 1 章　.NET 与 Visual Basic 2005

微软本身为了保持其领先地位，总是愿意抛弃旧的方式，积极创造新的方法。这方面很好的一个例子就是在 20 世纪 80 年代，微软把公司前途的赌注放在了图形界面上。尽管当时很多人认为老的字符界面已经非常完美，但是微软坚持认为图形界面要好得多，他们创造了 Windows 这个伟大的产品，从此个人计算机（PC）走进了千家万户，深刻地改变了人们的工作方式甚至生活方式，时至今日，人们几乎无法想象，没有 Windows，世界将会怎样？

2000 年左右，微软又开始了另一个宏伟的战略，这个战略在很多场合被微软的高层和 20 世纪 80 年代的图形界面战略相提并论，这个战略就是.NET。.NET 是微软面向未来互联网的战略，微软把公司未来发展的赌注放在了.NET 上，如同 20 世纪 80 年代微软在图形界面上下的赌注一样。

.NET 战略所引发的重大变革同样在起初遭到各种非议和怀疑，但最终同样被证明是微软的又一次伟大创造。历经数年的发展，.NET Framework 从 1.0、1.1、2.0，到目前最新的 3.0，.NET 已经发展成为构建企业应用程序最重要的平台之一。

本章作为全书的第 1 章，将从宏观角度介绍.NET 战略、.NET 平台以及 Visual Basic 2005 语言。对于初次接触.NET 的读者来讲，建立一个对.NET 和 Visual Basic 2005 的全局认识至关重要。

1.1　什么是.NET

关于.NET 最经常听到的问题是：“什么是.NET？”。

微软董事长兼首席软件设计师比尔·盖茨这样回答：“.NET 是指连接信息、人群、系统和设备的软件。”

微软原总裁兼首席执行官鲍尔默说：“.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。”

上述两种回答扼要地表述了.NET 的外在特征，从中可以初步得出.NET 的出现将带来这样的变化：软件将使不同的计算机以不同的方式相互交流，人们使用互联网的方式将与过去五六年使用互联网的方式大不相同。

在这里通过对.NET4 个关键特性的阐述，来进一步了解.NET 的概貌。

1．.NET 面向软件服务

今天的软件产品仅仅是一张或数张光盘，用户购买软件，亲自安装、管理和维护。软件服务则是来自因特网的服务，它替用户安装、更新和跟踪软件，这些软件的执行可能会跨越处于不同地理位置的不同计算机，同时用户的资料等各种数据也是存储在网络计算机上而不是存储在本地。这些就是软件产品和软件服务的不同之处。

伴随着被称为第三次 IT 革命的 Web 服务（Web Services）技术的出现以及 ASP（应用服务提供）产业的兴起，软件正逐渐从产品形式向服务形式转化，这是软件未来发展的趋势。.NET 正是为这一趋势所努力的成果。

Web 服务允许应用程序通过 Internet 进行通信和共享数据，而不管采用的操作系统、设备或编程语言是否相同。

.NET 是 Microsoft XML Web 服务平台，它提供了创建 XML Web 服务并将这些服务集成在一起所需要的功能。

2．.NET 依存于 XML

XML 是一种数据格式，它让数据容易理解并具有相当的灵活性。XML 是下一代产品的关键组成因素。微软的.NET 战略是依存于 XML 的，就像微软以前的产品依赖于图形界面一样。微软致力于把 XML 变成整个业界的标准，而微软.NET 战略的实施也许会成为最好的 XML 的实施案例，就像过去 Windows 是图形用户界面最好的实施案例一样。

XML 是.NET 的基础与灵魂，上述的软件服务和将要提到的融合多种设备和平台目标的实现，都离不开 XML 技术的支持。

3．融合多种设备和平台

在.NET 之前，软件是围绕一个系统而编写的，软件工程师当时是考虑一个系统而不是考虑用户来写软件的。如果用户换一台 PC，他们要做很多的工作才能把文档以及其他信息转到另一台 PC 上；如果他们想用另外一种终端工作，例如一种先进的电话或者手持便携设备，就必须运行一些协同软件以便让这两种不同的装置一起工作。

.NET 的出发点是：不把系统当作关键因素，诚然，会有不同的系统，但是它们应该能够自然地协同工作。在服务器层面，不把某个应用单纯地看作是在一种服务器上的一种应用，而是认为这个应用可以使用很多的服务器，并且能够自动地利用多个服务器带来的扩展的、更强的功能。以人为本的理念保证了由此产生的生产力和可靠性会超越大型机时代或者是 UNIX 时代的最好的应用，它所带来的巨大的可扩展性使得人们有很大的余地，这样，只要不断把新系统加入进来，就有了更大的能力。

.NET 的基本理念是：不再关注单个的网站和与 Internet 连接的单个设备，而是要让计算机组、相关设备和服务协同工作，提供更加广泛和丰富的解决方案以及服务，而不是像现在一样成为一座座信息孤岛。人们将能够控制何种信息、在何时、以何种方式传送给自己。.NET 的目标是把计算和通信带入一个丰富、合作和互动的环境中，远远胜过今天的单向网络。

在一些地方，这已经成为现实，例如为 Windows 平台设置的用于交易的 TPCC 基准，它的功效更为强大，同时性能价格比更高。

4．新一代的人机界面

.NET 是一个巨大的变化，它不仅是编程方面的巨大变化，也是用户界面的一个巨大变化。微软认为用户界面还可以更加自然，就是说人们坐在计算机（当然还可能是其他设备）前浏览信息的时候，不仅能够使用键盘，还能够使用一支笔来手写，也就是说计算机有手写识别的功能。还可能用声音来操作，就是说计算机还有语音识别功能。人们所需要的信息将展示在屏幕上，极高的分辨率使得屏幕的可读性非常强，即使是一个比较长的电子邮件也不需要打印。这些都是.NET 战略所推动的。

计算模式从终端主机时代、字符 PC 时代、GUI 时代发展到当前主流的 Internet 浏览器时代，今天的 Internet 与以前主机工作模式有许多相似之处，信息被存储在中央服务器内，而用户的所有操作都要依靠它们。让今天的网址之间相互传递有意义的信息，或者合作提供更广泛和更深层次的服务，是十分困难的。而用户要获得个性化的网络体验，或者得到“私人信息空间”来对网络信息进行编辑、分析和共享，则更加困难。现在，人们还在适应技术，但微软认为技术应当以人为本。包括 XML 和 SOAP（简单访问对象协议）在内的新行业标准将信息解放出来，使它们能够被重新组织、调整和编程，然后以任何可能的方式、在任何设备和系统上显示出来。以这些标准为基础的平台，将控制信息的权利重新交还给需要这些信息的人们。.NET 完全是为了实现这一目标而设计出来的，是微软公司提出的下一代互联网构想。

微软官方称：.NET 是微软的一个重要转折点。微软的产品在全球的应用越来越广泛，以服务为主的 MSN 也发展得越来越好，.NET 将以这些成功为基础。实现.NET 是一个较长的过程，类似从 MS－DOS 到 Windows 的转变。微软将继续提供和支持现有的平台和应用软件，包括不含.NET 技术的平台。但是，经过长时间的努力，微软的产品和服务将最终转化改进成订购式服务，通过 Internet 送货。

1.2 .NET 平台

.NET 是一种战略，.NET 平台则是实现这一战略的技术基础。它包括设备、基础设施、积木块服务、框架和工具等几个部分，这些部分所组成的一个垂直结构就是.NET 平台，如图 1.1 所示。

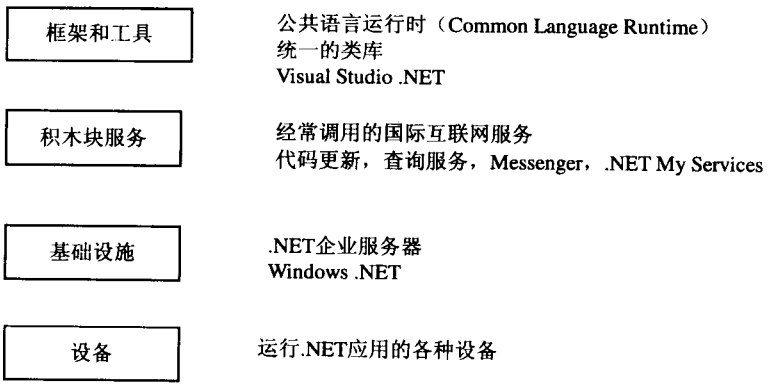

图 1.1 .NET 平台

1．设备

可以注意到处于.NET 平台底层的硬件部分使用了术语“设备”，而不再是“服务器”、PC 等字眼。这也是上面所提到的.NET 融合多种设备和平台理念的体现。这些设备可能是计算机，也可能是手持设备如手机等。

2．基础设施

基础设施包括构建企业应用所必须的操作系统和各种服务器，如数据库服务器等。

Windows .NET 是融入了.NET 技术的 Windows，它将紧密地整合.NET 的一系列核心构

造模块，为数字媒体及应用间协同工作提供支持，是微软公司的下一代 Windows 桌面平台。

在微软宣称的“第三代互联网”中，.NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础，它提供企业未来开展电子商务的高可靠性、高性能、高可伸缩性以及高可管理性。.NET 企业服务器的构成异常庞大而复杂，目前共包括 8 个各司其职的服务器，见表 1.1。

表 1.1 .NET 企业服务器一览表

服务器名称	功能描述
application center 2000	部署和管理基于 Windows 2000 之上的 Web 应用
biztalk server 2000	用于企业间交换商务信息
commerce server 2000	用于快速创建在线电子商务
exchange 2000	提供基于 Windows 2000 的通信和协作功能
host integration server 2000	为主机系统的组件集成提供方便
internet security &&acceleration server 2000	主要解决企业应用安全性和可管理性的问题
mobile information 2001 server	为移动解决方案提供可靠而具伸缩性的平台
SQL server 2000	提供完全的数据库和数据分析与数据挖掘解决方案

3．积木块服务

积木块服务（Building Block Services）是.NET 平台中的核心网络服务集合，它主要包括以下几个组成部分：Internet XML 通信，使 Web 站点变成灵活的服务来交换和处理数据；Internet XML 数据空间，为 Web 应用提供安全的和可编程的 XML 存储空间；Internet 动态更新，为快速开发和动态配置应用提供服务；Internet 日程安排，集成工作、社会和私人的日历；Internet 身份认证，提供从口令、钱包到生理数据等多级身份认证手段，还有 Internet 目录服务和 Internet 即时信息传递等服务。这些服务还在不断更新发展中。

4．框架和工具

.NET 框架（.NET Framework）处于.NET 平台的上层，它是.NET 平台的核心部分，是在.NET 平台上进行开发的基础，下一节将专门讨论.NET 框架。

.NET 平台上最好的开发工具当数微软发布的 Visual Studio.NET，它是 Visual Studio 6 的下一代产品，目前它的最新版本是 Visual Studio 2005，关于 Visual Studio 2005 的使用将在第 2 章中做详细介绍。

1.3 .NET 框架

.NET 框架（.NET Framework）是.NET 平台的编程模型，是创建、部署和运行 Web 服务及其他应用程序的一个环境。

1.3.1 .NET 框架的演化

任何技术的发展都是具有连续性的，.NET 也不例外。尽管.NET 框架富有革命性，但它同样是对过去技术的一种继承与发展，如图 1.2 所示展示了.NET 框架的演化过程。

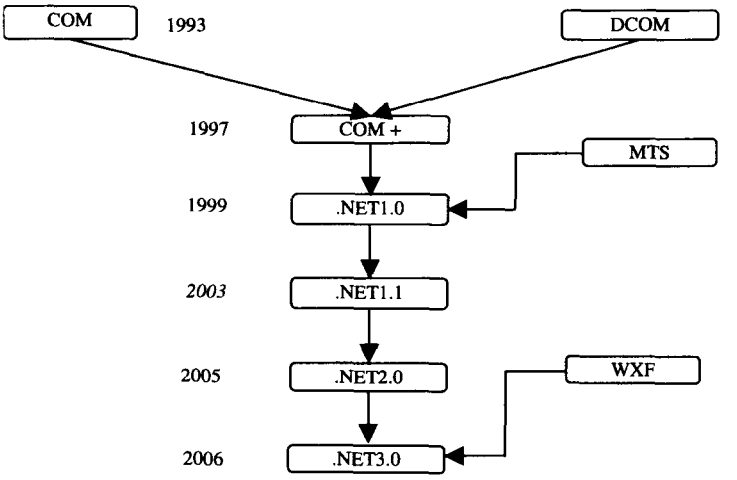

图 1.2 .NET 框架的演化

1.3.2 .NET 框架体系结构

.NET 框架体系结构如图 1.3 所示，它由以下 4 个主要部分组成：

（1）公共语言运行时（Common Language Runtime，CLR）

（2）统一类库（Base Class Library）

（3）ADO.NET

（4）活动服务器页面（ASP .NET）

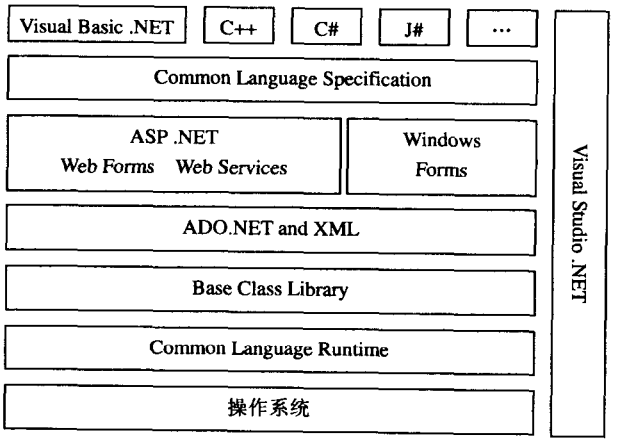

图 1.3 .NET 框架体系结构

1．公共语言运行时（CLR）

公共语言运行时是 .NET 框架应用程序的执行引擎。该名称不能准确反映它的全部功能。实际上，公共语言运行时在组件的开发及运行过程中，都扮演着非常重要的角色。在组件运行过程中，CLR 负责管理内存分配、启动或删除线程和进程、实施安全性策略，同时满足当前组件对其他组件的需求。与 COM 相比，CLR 的自动化程度大为提高（比如可自动执行内存管理），因而开发人员的工作变得非常轻松。尤其是反射功能将锐减开发人员

将业务逻辑程序转化成可复用组件的代码编写量。对编程语言而言，运行时这个概念并不新奇。实际上每种编程语言都有自己的运行时。Visual Basic 6.0 具有最为明显的运行时（名为 VBRUN），Visual C++跟 Visual FoxPro、JScript、SmallTalk、Perl、Python 和 Java 一样有一个运行时，即 MSVCRT。.NET 框架的关键作用在于，它提供了一个跨编程语言的统一编程环境，这也是它能独树一帜的根本原因。

2．统一的编程类库

.NET 框架为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集（API）。如今，C++开发人员使用的是 Microsoft 基类（MFC）库，Java 开发人员使用的是 Java 基类库，而 Visual Basic 用户使用的又是 Visual Basic API 集。.NET 框架统一了微软当前的各种不同类框架。这样，开发人员无需学习多种框架就能顺利编程。

远不止于此的是，通过创建跨编程语言的公共 API 集，.NET 框架可实现跨语言继承性、错误处理功能和调试功能。实际上，从 JScript 到 C++的所有编程语言，都是相互等同的，因此，可以说在.NET 中使用 C#语言还是使用 Visual Basic 语言已经无关重要了，它们最大的差别主要体现在语法层面，开发人员完全可以根据个人的习惯与喜好选择理想的开发语言。

3．ADO.NET

在开始设计.NET 框架时，Microsoft 就以此为契机重新设计了数据访问模型。Microsoft 没有进一步扩展 ADO，而是决定设计一个新的数据访问框架，但保留了缩写词 ADO。Microsoft 根据其成功的 ADO 对象模型经验设计了 ADO.NET。但 ADO.NET 满足了 ADO 无法满足的三个重要需求：提供了断开的数据访问模型，这对 Web 环境至关重要；提供了与 XML 的紧密集成；还提供了与.NET 框架的无缝集成。

4．活动服务器页面（ASP.NET）

ASP.NET 不仅仅是 Active Server Page（ASP）的下一个版本，它还提供了一个统一的 Web 开发模型，其中包括为开发人员生成企业级 Web 应用程序所需的各种服务和全新控件。ASP.NET 的语法在很大程度上与 ASP 兼容，同时它还提供一种新的编程模型和结构，可生成伸缩性和稳定性更好的应用程序，并提供更好的安全保护。

ASP.NET 是一个已编译的、基于.NET 环境的、可以用任何与.NET 兼容语言（包括 Visual Basic .NET、C# 和 JScript .NET.）创建的应用程序。另外，任何 ASP.NET 应用程序都可以使用整个.NET 框架，开发人员可以方便地获得.NET 技术的优点，其中包括托管的公共语言运行时环境、类型安全以及继承等。

ASP. NET 是使用.NET 框架提供的编程类库构建而成的，它提供了 Web 应用程序模型，该模型由一组控件和一个基本结构组成，有了它，Web 应用程序的构建变得非常容易。开发人员可以直接使用 ASP. NET 控件集，ASP.NET 还提供一些基本服务（例如会话状态管理和进程重启服务），这些服务极大地减少了开发人员需要编写的代码量，并使应用程序的可靠性得到大幅度提高。ASP.NET 还允许开发人员将软件作为一项服务（即 Web 服务）来提供。通过使用 ASP.NET Web 服务功能，ASP.NET 开发人员只需进行简单的业务逻辑编程，而由 ASP.NET 基本结构负责通过简单对象访问协议（SOAP）来提供服务。

1.3.3 .NET 框架编程模型

使用 .NET 框架编程不同于使用 Win32 API 编程，正如 Windows 编程与 DOS 编程

大相径庭一样。每一个 API 都是某一个时代的产品：DOS API 是 20 世纪 80 年代早期的产品；Windows API 是 20 世纪 80 年代中期的产品；而.NET API 是 20 世纪 90 年代后期的产品。

.NET 框架编程模型和传统编程模型有所不同，如图 1.4 所示。

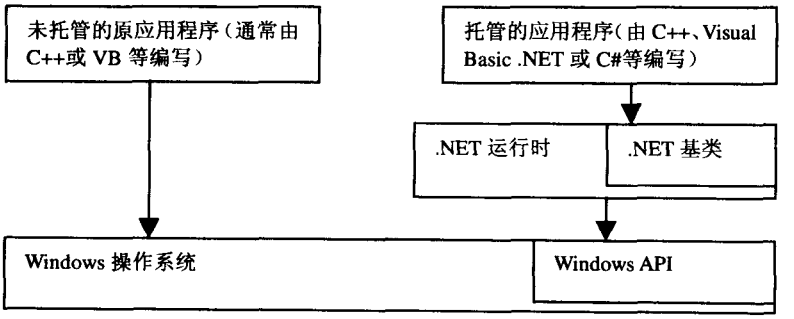

图 1.4 .NET 框架编程模型

传统的编程模型是上层应用直接依附在操作系统之上。例如在 Windows 中编程，程序运行于 Windows 之上。但在.NET 平台中，.NET 框架位于操作系统与上层应用之间，上层应用创建于.NET 框架之上，同时也运行于.NET 框架之上，而不像过去一样直接运行在操作系统之上。正是在操作系统和应用程序之间有了.NET 框架，才使得应用程序的平台独立性成为可能。

.NET 框架是 Microsoft 为开发应用程序创建的一个富有革命性的新环境。这句话最有趣的地方是它的含糊不清，但这是有原因的。注意这句话没有说“在 Windows 操作系统上开发应用程序”。尽管.NET 框架发布的几个版本都是运行在 Windows 操作系统上，但是不排除以后会推出运行在其他操作系统上的版本的可能，这些操作系统包括 FreeBSD，LinuxMacintosh，甚至个人数字助手类设备。这就是说.NET 具有平台独立性，是可移植的，这的确是一个很大的突破。微软为了保护其 Windows 操作系统的利益，一向在平台独立问题上非常保守，其开发的产品都只能运行于 Windows 环境中，这一直为人们所诟病。

要进一步理解.NET 框架编程模型，还需要认识两个新的概念：MSIL 和 JIT。下一节将对这两个概念做详细介绍。

1.3.4 .NET 程序的编译与运行

1．MSIL 和 JIT

在编译使用.NET 框架创建的代码时，不是立即创建成操作系统特定的本机代码，而是把代码编译为微软中间语言（Microsoft Intermediate Language，MSIL）代码，这些 MSIL 代码不专用于任何一种操作系统，也不专用于任何一种语言，有些类似于 Java 的字节码。Visual Basic .NET 及其他.NET 语言，如 C#等在编译阶段都编译为这种语言。

因为代码在编译阶段没有直接编译成本机代码，所以在执行应用程序时，必须完成更多的工作，这就是 Just-In-Time（JIT）编译器的任务。

JIT 把 MSIL 编译为专用于某种操作系统和目标机器结构的本机代码，只有这样，操作系统才能执行应用程序。这里编译器的名称 Just-In-Time，反映了 MSIL 仅在需要时才编译的特性。

过去，常常需要把代码编译为几个应用程序，每个应用程序用于特定的操作系统和 CPU 结构，这通常是一种优化形式（例如为了让代码在 AMD 芯片上运行更快），但更多时候是必须的（例如分别运行在 Windows 和 Linux 操作系统上）。现在就不必要了，顾名思义，JIT 编译器使用 MSIL 代码，而 MSIL 代码是独立于机器、操作系统和 CPU 的。目前有几种 JIT 编译器，每种编译器都用于不同的结构，总能找到一个合适的编译器创建所需的本机代码。这样，用户需要做的工作就比较少了，实际上，用户不必考虑与系统相关的细节，只需要把注意力放在代码的功能上就足够了。

2．程序集

在编译应用程序时，创建的 MSIL 代码存储在一个程序集中，程序集包括可执行的应用程序文件（这些文件可以直接在 Windows 上运行，不需要其他程序，其扩展名为.exe）和其他应用程序使用的库（其扩展名是.dll）。

除了包含 MSIL 外，程序集还包含元数据（即程序集中包含的数据的信息）和可选的资源（MSIL 使用的其他数据，例如声音和图片文件）。元数据使得程序集是完全自描述的，不需要其他信息就可以使用程序集。也就是说，人们不再需要把应用程序所需要的数据添加到系统注册表中，因此，部署应用程序就非常简单，只需把文件复制到远程计算机上即可。

当然，不必把运行应用程序所需要的所有信息都安装到一个地方。可以编写一些程序集，执行多个应用程序所要求的任务。此时，通常把这些可重用的程序集放在所有应用程序都可以访问的地方。在.NET 框架中，这个地方是“全局程序集高速缓冲存储器”（Global Assembly Cache），有相应的工具可以帮助把程序集放在高速缓冲存储器中。

3．托管代码

在把代码编译为 MSIL，再用 JIT 编译器把它编译为本机代码后，CLR 的任务还没有完全完成。用.NET 框架编写的代码在执行时是托管的，即 CLR 管理着应用程序，其方式是管理内存、处理安全性，以及允许进行跨语言调试等。相反，不在 CLR 控制之下运行的应用程序是非托管的，某些语言如 C++可以用于编写这类应用程序，例如，访问操作系统的低级功能。使用 Visual Basic .NET 主要编写在托管环境下运行的代码，它们使用 CLR 的托管功能，让.NET 与操作系统进行交互，当然也可以编写在非托管环境下运行的代码，但需要特别标注。

如图 1.5 所示的是传统的代码编译与运行过程，如图 1.6 所示的是.NET 中代码的编译和运行过程。

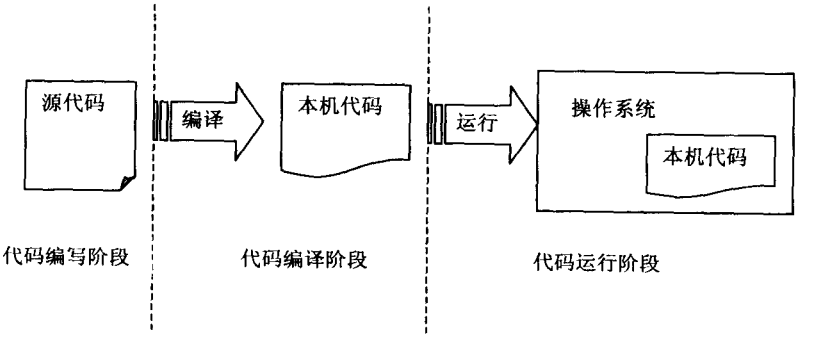

图 1.5　传统的代码编译与运行