Osborne开发与应用技术丛

Developer's Headsta

Visual Basic.NET

开发快速入门

Developer's Headstart

构建强大的
Visual Basic.NET应用程序

使用Visual Basic.NET
最大限度地发展
面向对象的软件开发

从Visual Basic
迁移和升级到
Visual Basic.NET

Jeffrey Shapiro 著
天宏工作室 译

Mc Graw Hill

清华大学出版社
http:www.tup.tsinghua.edu.cn

麦格劳-希尔教育出版集
http:www.mheducation.co

Osborne 开发与应用技术丛书

Visual Basic.NET 开发快速入门

[美] Jeffrey Shapiro 著

天宏工作室 译

清华大学出版社

(京) 新登字 158 号

Visual Basic.NET 开发快速入门

Jeffrey Shapiro: **Visual Basic.NET Developer's Headstart**

EISBN:0-07-219581-9

图书在版编目（CIP）数据

Visual Basic.NET 开发快速入门/（美）夏皮诺著；天宏工作室译 .—北京：清华大学出版社，2002.10

（Osborne 开发与应用技术丛书）

书名原文：Visual Basic.NET Developer's Headstart

ISBN 7-302-05654-4

Ⅰ.V... Ⅱ.①夏 ...②天 ... Ⅲ.BASIC 语言－程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2002）第 067814 号

出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）

http://www.tup.tsinghua.edu.cn

责任编辑：冯志强

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：787×960 1/16 **印张**：13.25 **字数**：291 千字

版 次：2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号：ISBN 7-302-05654-4/TP·3334

印 数：0001～4000

定 价：22.00 元

致谢

这本书是于2001年6月23日在Tech Ed上开始构思的，并且必须在7月31日（这是Microsoft“无意中”在Beta 2版本中设置的“时间炸弹”）之前完成。如果没有一组尽责的人们所做的大量工作和协作，那么这本书就不可能完成，每一个人都不得不从自己繁忙的时间表中抽出时间来解决这一个突然出现的项目。

我需要感谢的第一个人是我的编辑Ann Sellers，在我说出“这个项目可行”并且我可以完成时，她相信我并且随后使之得以实现。

如果没有我的两个技术编辑Amir Liberman和Lou Boni的努力，这本书同样不会问世，他们都在Ziphex Consulting，Inc.公司工作。他们做了大量工作，超出了为技术上的准确而做的编辑和检查工作。他们还负责第6章中关于迁移和互操作的讨论及代码、第5章中的Web服务示例，还有许多检查事实和验证代码的工作。感谢你们！如果你们没有对我将这个项目带入出版社的请求作出快速响应，那么我们就不可能在这里看到这些文字。Lou和Amir是非常负责的VB、VB.NET专家（可以回答每一个问题），您可以通过http://www.ziphex.com与他们联系。

我还应该向产品组致以深深的谢意，特别是项目编辑Mark Karmendy，感谢他将所有内容整理在一起，并在许多夜晚和周末光顾办公室，等待最后几分钟的改动等等。本书还由资深编辑Lisa Theobald进行了认真的编辑，她对一些技术性很强（并且经常很含混）的写法具有出色的洞察力，使我免于犯太多错误。

最后但决非不重要的是，我还要感谢我的妻子Kim，她不断地为我提供动力、情感以及物理上的燃料（比如茶、咖啡、肉、维生素以及爱等等），使我能够艰难地写出每一句话。当然还要感谢我8岁的儿子Kevin，是他不断地提醒我自己正在做什么以及正在为谁做这些工作。

简介

在Visual Basic群体中流传着大量说法，其中的争论不亚于影片《角斗士》公演的场面。分歧是十分明显的。许多开发人员相信Visual Basic.NET与传统的VB相差太远，您可能是其中之一。或者，您可能已经准备好使用新的VB.NET及其新特性，如纯粹的OOP、结构化错误处理以及自由线程，它为VB开发人员提供了非凡的功能。我相信，使用VB.NET，您就能够在未来几年里为我们面临的最复杂的软件开发方案展开公平的竞争。

无论您是否喜欢Visual Basic.NET，它都是存在的。Microsoft对.NET Framework寄予了厚望，VB将成为较长一段时间内开发.NET软件的主要工具。将围绕.NET和VB.NET的所有预期和焦虑，以及许多有争议的看法——有些是我的看法——融合在一起，就编写出了这本书。

任务

带着许多关于VB.NET的悬而未决的不确定性，我尽可能使本书的写法适合更多的读者。它的目标读者是需要尽快了解.NET软件开发的软件设计师和开发项目经理。如果您对公共语言运行时（Common Language Runtime，CLR）的出现以及它在Windows操作系统上的作用感到困惑，那么请阅读本书的前两章。

如果您是一位软件工程师，希望迅速了解为Windows操作系统编写程序的方式将要发生的变化，那么前两章也适合您。

本书还适合那些具有一些基本（但不是最基本）的编程技巧并打算加入我们的行列的新成员。是的，这意味着您通常需要事先具有一些编程知识，这样才能充分利用本书——如果您是一位VB程序员，那么就更好了。但是，对掌握所讨论的概念而言，事先具有VB知识不是必需的。

另一方面，如果您是一位经验丰富的专业VB程序员，则可能希望直接进入与您最相关的部分。第3章介绍了VB.NET的构造模块，并专门清楚地讨论了OOP。我们介绍了使用数组、字符串、流控制等等。我的方法并不是“告诉”您有哪些不同、有哪些新特性或哪些内容没有发生变化，而是向您“展示”有哪些不同、有哪些新特性或哪些内容没有发生变化，同时提供一些关于语言、及其语法和术语的基础知识和深入分析。

第4章和第5章都介绍Visual Basic.NET中的面向对象编程。这两章不但针对纯粹的VB程序员，也针对任何主流OOP语言的追随者。这两章还向您展示了如何

使用 VB.NET 进行数据库、Web 应用程序和 XML Web 服务开发。

第 6 章适用于任何希望知道如何移植传统 VB 代码以及 VB.NET 如何适应当前开发环境的人。第 7 章将 VB.NET 与 C#、Delphi 和 Java 等软件语言进行比较，如果您不能确定应该转移到 VB.NET 还是学习 C# 或 Java，那么本章将提供一些看法。它还可以减少您对丧失在 VB 6 技术中进行的现有投入所产生的恐惧。

开始学习所需的条件

要想翻开本书并感受 VB.NET 的功能，您需要安装 .NET Framework SDK Beta 2 或更高版本以及 Visual Studio.NET（包含 Visual Basic.NET）。虽然在 Windows 9x、Windows ME 和 Windows NT 上都支持 .NET，但是我建议您使用 Windows 2000 或 Windows XP。

Web 上的代码

为了便于学习，我们将为本书编写的代码放在了 Osborne 的 Web 站点上，其地址为 http://www.osborne.com。但是我强烈建议您创建在第 3、4、5 章中演示的类并编写代码。只是下载并运行示例不能让您完全体会 Visual Studio.NET 提供的优秀开发环境，也不能领略 VB.NET 的真正功能。

与我联系

本书就像它所宣称的那样，是 Visual Basic.NET 的入门资料。这是一个详尽而又复杂的题目，因此您不应该期望在有限的篇幅内找到自己问题的所有答案。我并没有引入一些技术细节来实现它，而是让您参考在 Microsoft 最终发布 Visual Basic.NET 和 .NET Framework 之前由 Osborne/McGraw-Hill 出版的篇幅更长的参考书，从而使您能够入门，这就是 *Visual Basic .NET: The Complete Reference*。它比较深入地展开了这里讨论的主题，并且增加了很多主题。在本书中，我不断地引用了一些在该书中介绍或扩充了的资料。如果您希望看到在该书中进行专门介绍的内容，请不要犹豫，及时与我联系。

我非常感谢您提供的任何反馈：表扬、批评或其他意见。您可以给我发电子邮件，地址为 jshapiro@codetimes.com。也可以与 Amir Liberman 和 Lou Boni 联系，以获得与 VB 有关的任何信息，他们的地址分别为 amir@ziphex.com 和 lou@ziphex.com。

目录

第 1 章

.NET Framework 和 VB.NET 简介

本章内容：

- .NET 的起源
- COM 技术
- 回顾 Visual Basic 的发展历程

使用计算机是为了解决问题，使我们的生活更轻松，允许我们远距离通信，挽救生命以及使生活充满乐趣。但是，在计算机能够做这些事情之前，必须对它们进行编程。不幸的是，除了一些有天赋的人之外，对计算机进行编程是一件非常困难的事情。

毫无疑问，计算机解决商业问题并使公司更具竞争力的能力已经成为了一个驱动因素，使工作站和服务器成为了世界各地不计其数的办公室中的永久固定设备。在过去十几年里，对在这些计算机上运行的商业软件的需求一直没有减弱，并且只要追求利润，这种需求就会继续增长。

自从个人计算机出现以来，Microsoft 拥有的操作系统市场已经扩大了很多倍。在为程序员提供创建在他们的计算机上运行的软件所需要的工具方面，或许没有哪一个商业策略能够比得上比尔·盖茨以及他的团队的成功。如果应用程序和软件解决方案不支持，那么操作系统或计算机平台（硬件）就不会有成功的机会。这是最基本的规则，Microsoft 从来没有打破这条规则。

1.1 .NET 简介

对于这种全新的产品，大多数人的第一个问题就是："什么是 .NET?" 在 Tech-Ed 2000 上，Microsoft 抢占了这个流行的 Internet 根域名，并将之作为以前所称的下一代 Windows 服务（Next Generation Windows Services，NGWS）的正式名称。

虽然整个 Microsoft 公司、全世界的大多数开发人员以及整个业界现在都对 .NET 寄予了厚望，但是这种产品的诞生是几年前的事情。这一节将介绍它的发展情况。许多人可能会说，"我为什么要关心这些？我只想开始编写代码。" 不过，扼要的回顾将帮助您了解 .NET Framework 引入的许多革新特性以及它为什么会改变您在未来几年内开发和部署应用程序的方式。

一些人可能认为 .NET 产品开始于 1995 年左右，大约是 Sun 发布 Java 的时候。但是从许多方面看，.NET 都开始于 DOS 流行的时候。

勾勒 .NET 的发展历程相当容易：图 1-1 显示了它在过去几年中的发展情况。首先是 DOS 时代，这是 Microsoft 与 IBM 之间的结合和分裂的历史。

然后是 80 年代后期的 16 位 Windows，它几乎是于 1995 年在一阵骚乱中结束的，当时，全世界都在关注图形用户界面，努力使用懒人式的不同实现方式开展工作。我们有了一个 16 位的处理命令堆栈、无法让人信任的多任务处理以及非常糟糕的错误保护（以至于术语 GPF 还不够大胆，不能作为 Webster 词典中的缩略词）。

Windows 95 是一个很大的改进（不过存在一个问题，即 DOS 仍然潜藏在彩色桌面和漂亮的图标下面）。Windows NT 改变了很多方面，使用了新的用户/内核模式抽象层、32 位内核、硬件抽象层、经过改进的驱动程序模型，并且更好地利用了

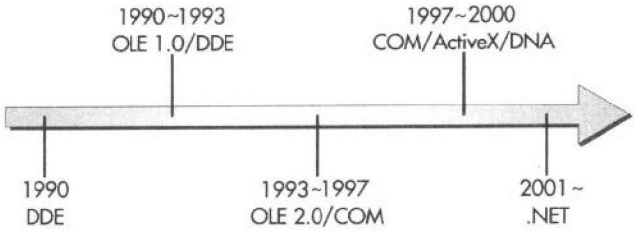


图 1-1　.NET 并不是一项新技术，更多的是从 70 年代的 DOS 开始的漫长的技术革新的结果

处理器，这大大改进了共享、容错以及线程处理。

然后是 Windows DNA（它代表 Windows Distributed interNet Applications Architecture，即 Windows 分布式网间应用程序结构），这是第 4 个时代。这可能不是 Microsoft 的时代，因为软件公司并没有停下来，眼睁睁地看着 Internet 大军占领它的后院（正如许多人所相信的）。在这个队列后面，比尔·盖茨签了两张十亿美元的支票，这就将我们带到了今天这个新的时代，即 .NET 和公共语言运行库（Common Language Runtime，CLR）。

作为一项新产品，Windows DNA 只不过是到达一个更好的方案的权宜之计。但是很显然，OLE 1.1 和动态数据交换（Dynamic Data Exchange，DDE），以及后面的 COM（Component Object Model，组件对象模型）和 DCOM（Distributed COM，分布式 COM）上的 OLE 2.0，它们为现在的软件开发人员所使用的产品作好了准备。当您回顾早期的时代时，就将看到共享内存、硬件、设备和处理空间及时间是应用程序开发人员所面临的问题的主要起因。

在操作系统解决了与新的抽象层的冲突后，新的问题出现了——在不同的应用层上。目前，市场上有大量优秀的开发工具，并且为 Windows 操作系统开发复杂的软件也变得更容易。到发布 Windows 2000 时，每一个开发工具都可以创建自己的服务、接口和小程序。

虽然我们努力尝试共享一切，但是我们最终只共享了“混乱”。惟一稳定的计算机是那些专门执行一项单独的服务或应用程序的计算机。必须安装和管理我们所创建的垃圾的 IT 人员可以证明安装太多应用程序所导致的问题。使情况变得更糟糕的是，管理应用程序服务器现在已经变得像是在管理一群尝试共享酒吧中的高脚凳的醉汉。

在开发 Windows DNA 的几年中，Microsoft 开始了一个计划，以修复那些更改操作系统或更改硬件等等的人们所面临的问题。同时，情况变得就像天使的呼吸一样明显：Internet 将成为以后几十年里的计算燃料。毕竟，能够在计算机上而不是在打字机上写书、在电子表格中而不是在账簿表格中计账会使我们更有效率。当我们能够使用电缆将计算机连接在一起，并共享公共文件夹分层结构、打印机以及其他

设备时，我们的效率再次得到提高。

然后，我们发现发送电子邮件而不是发送传真或电报是多么有用和便宜。至于 World Wide Web，我们还在尝试它的其他用途，而不仅仅是网上银行。

Windows DNA 是一组激动人心的指导原则中的一个。简单地说，下面是Microsoft 及其可爱的对手们所设想的在千禧年的目标（让我们将之称为“大胆的六原则计划”）：

- **支持 Internet** 操作系统不可见而且无缝地连接到 Internet 上。Internet 协议、TCP/IP、FTP、HTTP 等应该使每一个应用程序和服务都可以完全与 Internet 集成，并使用远程服务需要的一切。
- **可互操作性** 开放式标准和协议必须允许各种平台上的应用程序进行互操作和通信。应该消除互操作的障碍。
- **集成** 操作系统的共享服务和工具或者它们所支持的服务应该非常简单，许多开发商都能够实现和集成它们。维护软件（特别是二进制文件、接口和库）也应该更容易。版本控制的实现必须很简单，安装开发商的新软件和功能应该不会破坏现有的功能。
- **不依赖于语言** 软件开发人员应该能够使用既适合手边的工作，又在开发人员的技术和知识范围之内的语言。换句话说，开发人员应该不必学习新的语言就可以开发出与市场上最新的软件功能竞争的软件。
- **更短的上市时间** 工具必须是现成的。开发人员应该不必重新去发明车轮或尝试建造更好的捕鼠器。发布新版本或新产品应该非常简单，就是“插入新功能并组装产品”。
- **拥有的总成本** 公司应该不必在管理、更改控制和支持上花费大量资金。不应该将安装新的应用程序或新版本看作努力将百夫长送上战场那样困难。

Windows DNA 的目标非常高。但是实现这种设想又是另外一回事情。这就为我们带来了一个问题：在这 6 个原则中，到 2001 年完全实现了多少？如果您继续阅读下去，就应该能够在本章结束之前回答这个问题。

1.1.1 划分和规则

这种发展（或革新）中的下一个阶段是建立结构。从最早开始，人们就在激烈争论将界面从实现中分离出来的主题，而本书也稍微触及了这个问题。另一方面，Windows DNA 需要在几个级别上实现一个分层结构。层（tier）是一个相当抽象的术语，但是它充分描述了将功能空间彼此分离的过程，在已经成为以网络为中心（广域）而不是以计算机或处理器为中心的计算环境中，这是非常重要的。

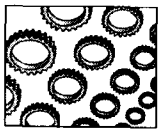
Windows DNA 模型并不是什么新的思想，它提出应该以三种方式逻辑地划分层：表示、逻辑和数据。但是，这个系统需要为每一层提供一些空间，以便与另一

层进行交互和通信。表示服务需要访问逻辑，而逻辑也需要访问数据（表示服务实际提供和使用数据，并且它们应该在很大程度上不知道完成其任务所需要的逻辑）。

大约在 Redmond 的工程师们认为自己彻底改造了早餐的同时，一个类似的分布式运动正在软件开发领域兴起。人们普遍接受这样的看法：软件服务和应用程序应该并且能够不再像北美野人那样在我们的硬盘驱动器上留下庞大的足迹。需要将软件分割为更小的相互联系的模块或组件。

数据库引擎是从单独的应用程序二进制文件中分离出来的第一个超大型组件，并被转移为操作系统服务和服务器的任务。网络访问、设备管理和表示服务几乎是在同时分离出来的。开发人员需要一种升级、修复和改进软件的方法，而不必向用户销售装在 5 磅重的盒子中的产品（接收到一个大的软件盒子可能很有趣，但是在将盒子里的聚苯乙烯填充物扔到废纸篓中，污染我们的地球之后，快乐很快就结束了）。

在 Microsoft Windows 领域中，用来对服务和应用程序执行这项工作的工具是一项被称为组件对象模型（Component Object Model，COM）的技术。



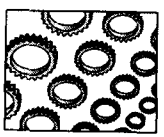
注意

COM 在 Java 领域中的竞争对手是 CORBA（Common Object Request Broker Architecture），它是 Java 主要的组件对象模型。

1.1.2　最早的 COM

COM 实际上是一个二进制标准，它允许两种算法彼此通话，并协商信息交换方式。它定义了组件的使用者和提供者之间的一个接口。Microsoft 积极地将 COM 宣传为分布式计算的最佳技术。要想理解.NET，您主要需要理解什么是 COM、它要实现的目标以及它的成功和失败之处。

没有 COM 接口的发布，COM 对象或组件的实现是没有用的。这个接口是分布式信息块———一种为应用程序访问某个位置的实现方式提供“全球定位系统”的规范。当应用程序使用者或客户使用一个 COM 组件时，应该只使用接口。实现方式应该保持完全独立，位于一个动态链接库（Dynamic-Link Library，DLL）中或作为一个可执行文件中。



注意

许多组件开发人员都在同一个模块中同时提供接口和组件。对于传统的 VB 而言，这是默认方式。

可以使用任何合适的语言来设计 COM 接口，但是最好使用面向对象（Object-Oriented，OO）语言。由于各种原因，C++ 是 COM 开发人员的第一选择。例如，C++ 允许您使用一个纯粹的抽象基类来设计 COM 接口，对于 COM 程序员来说，继承基类的能力是非常有用的面向对象的优点。在继承接口时，也将继承接口所需的内存蓝图（或映像）。这是 C++ 的一个显著的设计特点，这也是在设计组件时广泛使用它的原因之一。

接口包含一个虚函数集合的指针，它被称为虚函数表（virtual functions table，或 vtable）。vtable（参见图 1-2）指向 vtable 中列出的虚函数的实际实现方式。vtable 是独立于平台的，因此它可以位于任何操作系统或平台上，不过 vtable 的内存结构随操作系统的不同而不同。

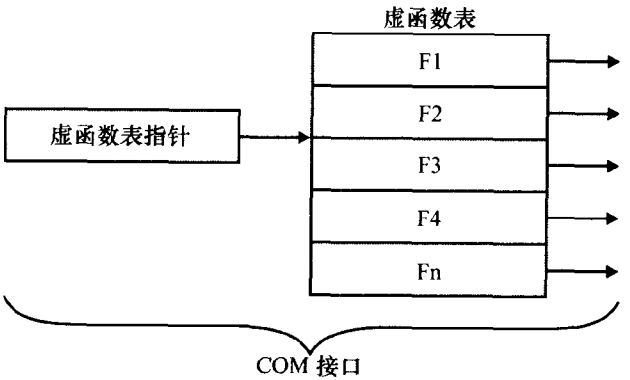
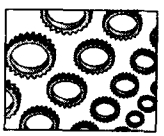


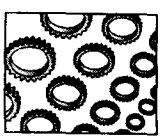
图 1-2 指向一个函数数组的指针被称为 vtable，反过来，vtable 又指向动态链接库或可执行文件（被称为 COM 服务器）中的函数实现方式



注意

虚函数是函数的定义，因此不是在 COM 接口中实现的。基类中的虚方法遵循相同的实现原则。它们是在派生或扩展的类中实现的。

COM 对象的实现是二进制文件，虽然可以在任何现代语言中开发 COM 实现方式，但是它是平台专用的，这意味着它只有在存在 COM 运行时环境的时候才能运行。



注意

Microsoft 允许您在 Visual J++ 中开发 COM 组件是使 Sun 感到烦恼的一个问题。