PROGRAMMER TO PROGRAMMER™

Professional Visual Basic Interoperability:
COM and VB6 to .NET
Written and tested for final release of .NET v1.0

Visual Basic
互操作高级编程
——从 COM 和 VB6 升级到.NET

Billy Hollis Rockford Lhotka 著　康博 译

Visual Basic 互操作高级编程

——从 COM 和 VB6 升级到.NET

Billy Hollis
Rockford Lhotka
著

康　博　译

清华大学出版社

(京)新登字 158 号
北京市版权局著作权合同登记号：01-2002-0261

内 容 简 介

在.NET 迅速崛起的时代，我们并不能完全抛弃 COM 时代使用 VB6 开发的各种应用程序，因此实现 Visual Basic .NET 和 VB6 之间的互操作就显得十分必要了。本书由浅入深介绍了互操作涉及的各种问题，详细讲述了.NET 与 COM 互操作的各种机制，并给出了一些实际的应用示例。在示范了具体的操作步骤后，本书还深入到系统内部讲述了互操作的实现细节，并对系统底层编程提出了相应的建议。

本书适用于那些想详细深入地掌握 VB6 与 Visual Basic .NET 互操作的开发人员，它对于把 VB6 系统升级为 Visual Basic .NET 提供了大量实用的操作方法。

图书在版编目(CIP)数据

Visual Basic 互操作高级编程——从 COM 和 VB6 升级到.NET/(美)荷利斯，(美)罗荷特卡著；康博译.
—北京：清华大学出版社，2002
书名原文：Professional Visual Basic Interoperability——COM and VB6 to .NET
ISBN 7-302-05746-X

Ⅰ.V… Ⅱ.①荷…②罗…③康… Ⅲ.BASIC 语言—程序设计 Ⅳ.TP312
中国版本图书馆 CIP 数据核字(2002)第 060529 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：郭东青
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：787×1092 1/16　**印张**：23.5　**字数**：601 千字
版　　次：2002 年 8 月第 1 版　2002 年 8 月第 1 次印刷
书　　号：ISBN 7-302-05746-X/TP・3394
印　　数：0001～5000
定　　价：42.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft，IBM，Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前　言

本书读者对象

本书适用于那些对 VB6 非常熟悉，但是想知道如何实现 VB6 与 Visual Basic.NET 互操作的程序开发人员。

本书假设您已经非常熟悉.NET 的整体结构及 VB6 与 Visual Basic.NET 的语法差异。本书并不介绍 VB6 与 Visual Basic.NET 之间的差异，也不讲述.NET 技术。如果您想了解.NET 的基础知识或者 Visual Basic.NET 的详细内容，请参考清华大学出版社翻译出版的下列书籍：

- 《.NET Framework 高级编程》
- 《VB.NET 高级编程》

本书主要内容

第 1 章“互操作功能的重要性”，讲述 VB6 及 COM 与新的.NET 平台之间的互操作存在的必要性。

第 2 章“在.NET 中使用 COM 组件”，讲述如何将 COM 组件导入到.NET 平台中，使其可用于.NET 环境，但此时它仍作为 COM 组件运行在 COM 当中。同时还将讲述如何设计 COM 组件和接口才能让.NET 轻松地以.NET 的方式处理这些组件。

第 3 章“在 COM 中使用.NET 程序集”，介绍和第 2 章相反的内容——如何创建一个能用于 COM 应用程序中的.NET 程序集，同时还将介绍如何有效地使用这个程序集。

第 4 章“从.NET 调用 MTS 和 COM+中的 COM 组件”，其中将重点介绍在 Visual Basic .NET 客户端应用程序中使用 COM+或基于 MTS 的 COM 组件。

第 5 章“COM+中的 Visual Basic .NET 和 COM”，介绍运行在 COM+中的 Visual Basic .NET 程序集是如何与同处其中的 VB6 组件进行交互的，以及相反的过程。

第 6 章“定制编组”，使用编组的不同机制——非托管代码与托管代码之间互相传递信息。

第 7 章“线程问题”，介绍在 Visual Basic.NET 和 VB6 之间进行互操作时可能产生的线程问题。其中将讲述线程在 COM 环境下是如何在 VB6 中运行的，接着将讲述在.NET 平台上它是如何在 Visual Basic.NET 中运行的。

第 8 章“DCOM、远程处理和 Web 服务”，内容涉及通过 COM 互操作使用 DCOM，及.NET 平台提供的 DCOM 替代技术——XML Web 服务和.NET 远程处理技术。

第 9 章“共享配置信息”，讨论使用私有的 INI 文件、Windows 注册表和基于 XML 的配置文件来共享存在于 VB6 及 Visual Basic .NET 中的配置信息。

第 10 章“.NET 中的平面文件访问”，介绍从 VB6 升级到 Visual Basic.NET 时，二进制方式、随机方式和顺序方式访问平面文件各有哪些变化。

第 11 章“在.NET 中使用 ADO”，对比 ADO 和 ADO.NET，并将讨论在哪些情况下应该继续使用 ADO，包括在 Visual Basic .NET 中的新项目中使用 ADO 的情形。另外还将考察在.NET 及 VB6 中使用 ADO 的差别。

第 12 章“在.NET 中调用 API 或静态 DLL 函数”，介绍.NET 与不带 COM 接口的 DLL 之间的互操作性，此外还将介绍通常哪些情况允许 VB 程序来处理 Windows API 函数。

第 13 章“.NET 中 ActiveX 控件的互操作性”，介绍一种需要特殊处理的 COM 组件——ActiveX 控件。

第 14 章“VB6 代码迁移的准备工作”，如果继续使用或增强现有的 VB6 程序，本章将提供一些概要性的指导原则，这些原则将减少向 Visual Basic .NET 迁移的工作量。

本书的使用要求

使用本书需要您在计算机上能够运行下列软件：

- Windows 2000/XP
- Visual Studio .NET 专业版(或者更高版本)
- 带有 Visual Basic 6.0 的 Visual Studio 6.0

用户支持

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(地址为 feedback@ wrox.com)来向我们反馈意见。请确保反馈信息提到本书的书名。

如何下载本书的范例代码

当您访问 Wrox 公司站点(地址为 http://www.wrox.com/)时，通过 Search 工具或书名列表，可以方便地找到需要的书目。然后，单击 Code 栏中的 Download 超链接，或者单击本书的详细页面中的 Download Code 超链接，就可以下载相应的范例代码。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。将附件保存到本地磁盘上的文件夹中后，需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时，通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中，应确保解压缩程序(WinZip、PKUnzip, 其他)被设置为使用原有文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请反馈信息给我们，我们将不胜感激。

勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。您可以将 E-mail 信息发送到 support@wrox.com。您的反馈信息将被检查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 http://www.wrox.com/，并通过 Advanced Search 或者书名列表轻松定位本书页面。然后，单击 Book Errata 超链接即可，该链接位于本书的详细页面中的封面图解下面。

E-mail 支持

如果您希望直接向详细了解本书的专家咨询本书中问题，可以把电子邮件发送到 support@wrox.com。一个典型的电子邮件应包括下面的内容：

- 在主题字段中必须有本书的书名、ISBN 的后 4 位数字和问题的页数。
- 信息的正文应包括您的名字、联系信息和问题。

我们不会给您发送无用邮件，我们需要有用的详细资料，以便可节约您和我们的时间。当您发送一个电子邮件信息时，将得到下面一系列支持：

- 用户支持：首先，您的信息将被递送到我们的用户支持人员手中，并由他们阅读。对于一些被频繁提到的问题将被归档，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持：接着，一些有深度的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。
- 作者支持：最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 http://p2p.wrox.com/论坛的公共列表中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们惟一的系统将 programmer to programmer™(由程序员为程序员而作)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预定一个邮件列表：

(1) 登录 http://p2p.wrox.com/站点。

(2) 从左边的主菜单栏选择一个适当的类别。

(3) 单击您要加入的邮件列表。

(4) 按照说明订阅并填写自己的邮件地址和密码。

(5) 回复您收到的确认邮件。

(6) 使用预定管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最好支持的原因

您可以选择连接到整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目　　录

第 1 章　互操作功能的重要性 …… 1
1.1　Visual Basic .NET 概述 …… 1
1.2　迁移方案 …… 2
1.2.1　VB6 和 Visual Basic .NET 的不兼容之处 …… 2
1.2.2　迁移到 Visual Basic .NET 的好方案 …… 5
1.2.3　坏的迁移情况 …… 5
1.2.4　不能迁移的情况 …… 6
1.2.5　互操作性的含义 …… 6
1.3　设计互操作方案 …… 6
1.3.1　Web 服务 …… 7
1.3.2　智能型客户端 …… 7
1.3.3　新的 ASP.NET 前端 …… 7
1.3.4　扩充原有的组件 …… 8
1.3.5　分布式数据处理 …… 8
1.4　互操作应用的主要范围 …… 8
1.4.1　在.NET 中调用 COM 组件 …… 8
1.4.2　在 COM 中调用.NET 组件 …… 9
1.4.3　在新旧程序间共享数据访问 …… 9
1.4.4　共享配置信息 …… 9
1.4.5　使用静态入口点调用非 COM 的 DLL …… 9
1.5　小结 …… 10

第 2 章　在.NET 中使用 COM 组件 …… 11
2.1　创建 COM 组件 …… 11
2.1.1　构建 COM 组件 …… 12
2.1.2　二进制兼容性 …… 22
2.2　导入 COM 组件 …… 23
2.2.1　在设计阶段导入 COM 组件 …… 24
2.2.2　动态导入 …… 37
2.3　使用 COM 组件 …… 38
2.3.1　早期绑定 …… 39
2.3.2　后期绑定 …… 44
2.3.3　对象的生存期 …… 48

2.3.4 多重接口 …… 50
2.3.5 继承性 …… 53
2.3.6 在.NET 中检查 COM 组件 …… 55
2.4 小结 …… 62

第 3 章 在 COM 中使用.NET 程序集 …… 63
3.1 创建.NET 程序集 …… 63
3.1.1 创建.NET 程序集概述 …… 63
3.1.2 对互操作进行设置 …… 75
3.1.3 使用 ComClass()属性 …… 75
3.1.4 手工提供类 …… 77
3.2 在 COM 中注册.NET 程序集 …… 85
3.2.1 使用 Visual Studio .NET …… 86
3.2.2 使用命令行实用程序 …… 89
3.3 在 COM 和 Windows 中使用.NET 对象 …… 92
3.3.1 早期绑定 …… 93
3.3.2 后期绑定 …… 98
3.3.3 处理锁定的文件 …… 100
3.3.4 对象生存期 …… 100
3.3.5 多重接口和默认接口 …… 102
3.3.6 在 COM 中检查.NET 程序集 …… 106
3.4 小结 …… 109

第 4 章 从.NET 调用 MTS 和 COM+中的 COM 组件 …… 110
4.1 使用 COM+组件 …… 111
4.1.1 创建 COM+组件 …… 111
4.1.2 在 Visual Basic .NET 中调用 COM+组件 …… 124
4.1.3 排队组件 …… 132
4.2 传输数据 …… 140
4.2.1 使用 ADO 中的 Recordset 对象 …… 141
4.2.2 使用变量数组 …… 146
4.2.3 使用 XML 文档 …… 148
4.3 在 VB6 中调用 COM+中的.NET 组件 …… 155
4.3.1 在.NET 中创建 COM+组件 …… 156
4.3.2 使类可用于 COM 客户程序 …… 162
4.4 小结 …… 163

第 5 章 COM+中的 Visual Basic .NET 和 COM …… 164
5.1 COM+和 VB6 …… 165

5.1.1 为 COM+创建 ActiveX DLL …… 165
5.1.2 在 COM+中注册 DLL …… 170
5.1.3 创建 VB6 测试程序 …… 173
5.2 COM+和 Visual Basic .NET …… 173
5.2.1 在 COM+中注册 DLL …… 178
5.2.2 创建 Visual Basic .NET 测试程序 …… 180
5.3 从 COM+中的 Visual Basic .NET 中调用 VB6 组件 …… 182
5.4 从 COM+内的 VB6 中调用 Visual Basic .NET 组件 …… 185
5.5 小结 …… 188

第 6 章 定制编组 …… 190
6.1 四种不同的编组方法 …… 191
6.1.1 第一种编组方法：使用 Tlbimp.exe …… 191
6.1.2 第二种编组类型(简单数据类型编组)：将属性添加到使用标准类型的接口中 …… 192
6.1.3 第三种编组类型：对 COM 和.NET 之间传递的自定义数据结构进行编组 …… 193
6.1.4 第四种编组类型：实现 ICustomMarshaler 接口 …… 194
6.2 性能问题 …… 195
6.3 第二种编组类型 …… 196
6.3.1 Ildasm …… 196
6.3.2 Ilasm …… 197
6.3.3 使用带有第二种编组类型的 Ildasm 或者 Ilasm …… 198
6.4 第三种编组类型 …… 207
6.5 第四种编组类型 …… 221
6.6 托管代码到非托管代码 …… 233
6.7 小结 …… 233

第 7 章 线程问题 …… 235
7.1 线程的基础知识 …… 235
7.1.1 线程的复杂性 …… 236
7.1.2 使用单线程与多线程 …… 237
7.1.3 VB6 中的线程 …… 238
7.1.4 Visual Basic .NET 中的线程 …… 239
7.2 线程之间的互操作 …… 240
7.2.1 在 .NET 中调用 COM …… 240
7.2.2 在 COM 中调用 .NET …… 253
7.3 小结 …… 254

第 8 章 DCOM、远程处理和 Web 服务 …… 255
8.1 XML Web 服务 …… 255

8.2 .NET 远程处理 …… 256
8.3 DCOM …… 256
8.4 XML Web 服务和互操作 …… 257
8.4.1 从.NET 中调用 COM …… 258
8.4.2 从 COM 中调用.NET …… 263
8.5 .NET 远程处理和互操作 …… 268
8.5.1 从.NET 中调用 COM …… 269
8.5.2 从 COM 中调用.NET …… 277
8.6 小结 …… 281

第 9 章 共享配置信息 …… 283
9.1 专用的 INI 文件 …… 284
9.1.1 INI 文件的结构 …… 284
9.1.2 访问 VB6 中的 INI 文件 …… 285
9.1.3 访问 Visual Basic .NET 中的 INI 文件 …… 287
9.2 使用 Windows 注册表存储配置设置 …… 289
9.3 基于 XML 的配置文件 …… 291
9.3.1 访问带有.NET Framework 类的 XML 配置信息 …… 292
9.3.2 使用 Visual Basic .NET 中的 System.XML 访问设置信息 …… 293
9.3.3 使用 VB6 中的 MSXML 访问设置 …… 296
9.4 并行性问题 …… 298
9.5 小结 …… 298

第 10 章 .NET 中的平面文件访问 …… 299
10.1 对于平面文件的随机访问 …… 299
10.1.1 VB6 中的随机访问 …… 300
10.1.2 Visual Basic .NET 中的随机访问 …… 303
10.2 平面文件的二进制访问 …… 307
10.2.1 VB6 中的二进制访问 …… 307
10.2.2 Visual Basic .NET 中的二进制访问 …… 310
10.3 顺序的文件访问 …… 312
10.3.1 VB6 中的顺序访问 …… 312
10.3.2 Visual Basic .NET 中的顺序访问 …… 314
10.4 作为 Visual Basic .NET 中替代方案的流 …… 315
10.4.1 使用流技术的顺序访问 …… 315
10.4.2 使用流技术的二进制访问 …… 316
10.5 小结 …… 318

第 11 章　在.NET 中使用 ADO……319
11.1　ADO 和 ADO.NET 之间的比较……319
11.1.1　ADO 优点和缺点……320
11.1.2　在.NET 中何时仍然需要 ADO……320
11.2　在.NET 中使用 ADO……322
11.2.1　Visual Basic .NET 中的 ADO 代码……323
11.2.2　数据绑定限制……324
11.2.3　将一些 ADO 属性设置为字符串而引发的问题……324
11.2.4　将 ADO Recordset 转换为 ADO.NET DataSet……326
11.3　DAO 和 RDO 的意义……327
11.4　小结……329

第 12 章　在.NET 中调用 API 或静态 DLL 函数……330
12.1　.NET 对使用带有静态入口点的 DLL 的需求在减少……330
12.1.1　对访问 Windows API 的需求减少……330
12.1.2　较旧代码的退休……333
12.2　平台调用服务……334
12.2.1　声明 API……334
12.2.2　调用 API 函数……335
12.2.3　给函数指定别名……335
12.2.4　将结构作为参数传递……336
12.2.5　数据编组问题……338
12.3　更多的控制……340
12.3.1　控制数据编组……340
12.3.2　使用 DLLImport 而不是 Declare……341
12.4　性能因素……342
12.5　小结……342

第 13 章　.NET 中 ActiveX 控件的互操作性……343
13.1　何时在.NET 中使用 ActiveX 控件……343
13.2　Windows Forms 控件和 ActiveX 控件之间的差异……344
13.3　在.NET 中驻留 ActiveX 控件……345
13.3.1　示例——Windows Media Player 控件……346
13.3.2　关于示例的注意事项……347
13.3.3　从 Properties 列表中移出的 Custom 属性……347
13.3.4　使用 Aximp.exe 创建包装器……347
13.3.5　导入 VB6 UserControls……348
13.3.6　不能导入的控件……348
13.3.7　安全性……348

13.4 VB6 中的.NET Windows Forms 控件 …… 349
13.5 小结 …… 349

第 14 章 VB6 代码迁移的准备工作 …… 350
14.1 停止使用默认的属性和方法 …… 350
14.2 避免数组的非零下界 …… 351
14.3 让所有的参数显式地传值或引用(ByRef 或 ByVal) …… 351
14.4 将默认值放在所有可选的参数上 …… 352
14.5 在独立的代码行上声明所有变量 …… 352
14.6 注意声明变量的位置 …… 352
14.7 避免 UDT 中的固定长度字符串 …… 353
14.8 清除已作废的关键字 …… 354
14.9 删除隐式对象实例化 …… 355
14.10 停止隐式加载窗体 …… 356
14.11 将数据绑定转换到 ADO …… 357
14.12 尽可能使用固有的常量 …… 357
14.13 停止编写 DHTML 页面和 WebClass …… 357
14.14 包装 API 调用 …… 358
14.15 从 UI 中得到逻辑并放入组件和类中 …… 358
14.16 避免后期绑定 …… 358
14.17 小结 …… 359

第1章　互操作功能的重要性

当前，我们面临着近十年来软件开发中最大的转换问题。最终，一个能用于 Internet 世界的全新平台便应运而生了。这就是 Microsoft .NET。对于任一需要 Internet 集成的软件开发项目而言，它都将是一个极大的进步。

然而，新平台却经受着一个固有问题的困扰。所有公司都已经在其软件的基础结构上投入了巨资，如果单单为了支持一个新的平台，就将目前尚能运营的软件系统扔掉，这在资金方面是绝不可行的。于是出现一个过渡期就显得十分必要了(也许这个过渡期会很长)。在这个过渡期内，新的平台必须能同时运行目前已被启用的软件。

通常情况下，简单的共存是远远不够的。为了能够来回传递信息，新旧系统之间必须要能够交互作用。这种传递信息的能力称为互操作性，其间会有一个较高层次的综合层位于这两个平台之间。

VB6 是当前最为流行的编程语言，而在今后至少一段时期内，Visual Basic .NET 则很可能成为最常用的语言。因此，当软件开发环境转移到.NET 时，最重要的互操作情况就是：使得 Visual Basic .NET 中的程序和业已存在的 VB6 程序以及它们所使用的数据进行互操作。这就是本书将要讲述的全部内容。

1.1　Visual Basic .NET 概述

VB 是当前最为流行的编程工具。由 Microsoft 委托进行的调查数据显示：在 2001 年，使用 VB 编写的程序代码行数已经超过了用 COBOL 编写的代码。Microsoft 在全球卖出的 VB 使用许可证已经超过了 800 万份。

有迹象表明，VB 的流行趋势将延伸到.NET 时代。事实上，可以肯定地说，VB 将变得更加流行。.NET 去除了 VB 发展历史上曾有的众多限制，这将消除人们选择其他编程语言所曾有的那些理由。

当开发人员转移到.NET 环境时，他们仍然没有充足的理由舍弃 VB 而去选择其他编程语言。应用于.NET 中的编程语言，在其基本功能方面并没有太大的变化。它们仍然具备相同的功能和性能特征。VB 依然能适应大量的 Web 编程之需，仍能代替其他编程语言和工具。当大量的公司转移到.NET 之后，最可能的结果就是，他们将选择一种其开发人员已经熟悉的语言作为编程工具。为什么会在同一时刻面临一种新平台和一种不同语言的双重挑战呢？

这有两个重要的依据：

- VB 是当前正在使用的最受欢迎和最为流行的开发工具。
- VB 很可能成为一定时期内最流行的能用于.NET 中的开发工具。

这也就意味着，许多公司将会抓住机遇将他们当前的 VB6 编程环境升级到基于 Visual

Basic .NET 的环境中来。

两种策略——迁移和互操作

将开发环境转移到.NET 中，其理由是令人充分信服的，许多公司都将要进行这种变化。对于那些当前正在使用 VB 和正准备将 Visual Basic .NET 作为其开发工具的公司来说，有两种策略可以实现这种过渡。

- 将现有的 VB 程序迁移到 Visual Basic .NET 中。
- 继续使用现有的 VB 程序，同时在 Visual Basic .NET 中开发新的项目。设法使新程序(.NET 代码)和原有程序(非.NET 代码)进行互操作。

当然，这些策略可以混合使用，这样就可以仅迁移其中的一些程序，而另一些程序则保留下来。事实上，相对于单一某个策略，经常采用的是混合策略。

1.2 迁移方案

在.NET 之前，VB 的最新版本是 6.0。VB 1.0 发布于 1991 年，6.0 发布于 1998 年。仅仅 7 年时间就发布了 5 个新的版本。那么使用 VB 的企业是如何应付这种快速变化的？

实际上做到这一点并不困难。从早期的 VB 版本向新版本迁移相对无需费力。每一个新版本中都会增加一些新的语法，但是语法间的变化非常微小。因此只需将旧项目在新版 VB 中打开，并再保存一次即可。在早期大多数 VB 版本中，这就是要进行的全部升级工作。

其中例外的情况是从 VB3 到 VB4 的升级。VB3 生成的是用于 Windows 3.1 的 16 位代码，而 VB4 创建的是用于 Windows 95 和 NT 3.1 的 32 位代码。Microsoft 废除了 add-in 控件中的 VBX 窗体，并且用基于 COM 的 OCX 控件来取代了它。这使得 VB3 到 VB4 的迁移成为 VB 开发人员需要进行的最为艰难的一次版本转换。

但即使迁移工作量不算太多，在得到可替换的控件之后，对于大多数的项目而言，仍然需要适当地进行手工修改。例如，我所在的开发组中有 8 个开发人员，并且大约有 30 个大型程序需要迁移，那么完成这些工作则最多需要一周的时间。

后来这种情况就变好了。Microsoft 努力保持版本向上的兼容性，但是现在我们面临的是这样的一个局面，底层技术的改变迫使 VB 再次进行根本性的调整。如果希望仍像以前那样，仅仅将 VB“拿到”.NET 平台中就保持向上兼容，那则是不可能的了。

1.2.1 VB6 和 Visual Basic .NET 的不兼容之处

VB6 和 Visual Basic .NET 的不兼容之处，主要是由于 Visual Basic .NET 适应.NET Framework 的结果。另有一少部分是语言简化的结果，它的语法基础退回到了多年以前的那种简洁形式。

Microsoft 并没有仅仅为了变化而变化，理解这一点是很重要的。围绕这种变化有很多争论，到底哪些东西需要保留，哪些需要去除。要将 VB 带入到.NET 中，这种变化是十分必要的，这种必要性是因为.NET 可以将 VB 推进到一个新的开发时代。这种由不兼容所带来的负面效应，相对于将 VB 带入到.NET 世界所带来的正面效应而言，最终可被消除掉。

在某些书中所提及的有关如何升级到 Visual Basic .NET 中去的内容，以及它们所提供的转换的所有细节，我们在此就不再重述了。但是理解这些变化之大是很重要的，它可使得我们更好的对比迁移与互操作这两种方案。下面所列的是两者不兼容的主要所在，对每一处都给出了相应的示例。

1. 数据类型的变化

.NET Framwork 拥有一套一致的数据类型，它们可供运行在.NET 中的所有语言使用。这被称为通用类型系统，或简称为 CTS。

CTS 包括传统语言所共有的所有数据类型，例如，整型、浮点型和布尔型等等。它也包括长度可变的字符串类型，该类型早就是 Basic 的一部分，但是该类型在其他许多编程语言中却没有出现，例如 C++。

正如您可能想到的，创建一套能供所有语言使用的数据类型需要采用折衷方案。CTS 适用于 VB 是通过添加某些类型来实现的，例如添加可变长度的字符串类型，但是有些地方却需要对 VB 作出某些调整。

在 CTS 中，整数类型的名称发生了变化，另外还添加了一些新的数据类型。表 1-1 就总结了这些变化。

表　1-1

VB 中的数据类型	Visual Basic .NET 中的数据类型	数据宽度	表示范围	.NET Framework 中的相应类型名称
Integer	Short	16 bits	-32 768 ~ 32 767	Int16
Long	Integer	32 bits	-2 147 483 648 ~ 2 147 483 647	Int32
(N/A)	Long	64 bits	-9 223 372 036 854 775 808~ 9 223 372 036 854 775 807	Int64

上述的这些变化并非迁移的主要障碍，因为原有的代码可以非常轻松地进行相应的修改。即使是不修改代码，它们仍能正常运行，这些变量仅仅是可以获得比原先范围更大的值。

另一个变化则显得更为根本性一些。所有的.NET 中数组，其下标都是从 0 开始的。在 VB6 中可以编写如下所示的程序：

而在.NET 中却没有与之对应的数组存在。因此，任何使用这种数组的 VB6 代码若要在 Visual Basic .NET 中运行，都必须进行较大的修改。

另一个被删除的数据类型就是固定长度的字符串。在 VB6 中，可以如下声明一个固定长度的字符串：

在本例中，字符串长度被精确地固定为 16 个字符。而在 Visual Basic .NET 中却没有这样