

Windows 操作系统环境 混合语言程序设计

[美] Matthew A. Telles 著

胡衡沅 等 译

王 勇 校

- 了解不同程序设计语言的最佳用途,学习如何组合不同的程序设计语言进行编程
- 学习如何使用现有的 Windows 应用程序更快地构造用户自己的程序
- 学习基于组件的 Windows 程序设计
- 建立一个完整的 Windows 数据库系统
- 随附软盘,提供书中所有示例的代码及完整的 ISAM 库



电子工业出版社

Publishing House of Electronic Industry



Windows 操作系统环境

混合语言程序设计

[美] Matthew A. Telles 著

胡衡沅 黄小晗 张维存 译

王 勇 校

电子工业出版社

0030341

(京)新登字 055 号

内 容 提 要

本书主要介绍在 Windows 环境下的混合语言编程。所谓混合语言编程指的是利用各种组件进行编程,一个组件可以是一种程序设计语言,也可以是一般应用程序。

本书分为三部分。第一部分分别介绍了 Visual Basic、Borland C++、Visual C++ 和 Word for Windows 的特色(长处和不足);第二部分则以实例方式详细介绍了混合语言编程的整个过程,提出了 Windows 下进行混合语言编程需遵守的准则;第三部分(附录)则包含配套磁盘中软件的重要文档,以及一些重要技术说明。

本书适合于 Windows 程序员,也可供一般微机程序员参考用。

本书英文版由 M&T 出版公司出版。版权归 M&T 出版公司所有。本书中文版版权经美国远东图书公司(Far East Books Inc. U.S.A)授权电子工业出版社独家出版和发行。本书的翻译和文字处理工作由美国远东图书公司完成。未经出版者同意,不得以任何形式和手段复制或抄袭本书内容。

Copyright ©1994 by M&T Books, All rights reserved. Chinese Version Copyright ©1994 by Publishing House of Electronic Industry.

Windows 操作系统环境

混合语言程序设计

[美] Matthew A. Telles 著

胡衡沅 黄小晗 张维存 译

王 勇 校

责任编辑 胡毓坚

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

北京市顺新印刷厂印刷

开本:787×1092 毫米 1/16 印张:19.5 字数:472 千字

1995 年 1 月第 1 版 1995 年 1 月第 1 次印刷

印数:10100 册 定价:58.00 元(含盘)

ISBN7-5053-2835-2/TP·815

致 谢

衷心感谢在 CompuServe 工作的同行,没有他们的帮助,本书的写作不可能得以完成。此外,还要感谢 Internet——一个拥有丰富有用信息的有趣的地方。最后,还要感谢 Microsoft Developer Network 为一些不寻常的技术问题提供非常宝贵的答案。

本书的用途

无论你是技术作者、软件工程师或者是管理人员,都将在本书中找到令人感兴趣的东西。将学会如何利用别人的工作来改进自己的工作。此外,将学会如何借助于其他语言和开发环境的功能来扩充当前使用语言的开发环境。

作为技术作者,将了解如何把现存的应用程序结合到别人或你自己使用的其他应用程序中去。将学会混合语言编程中涉及到的术语和行话。最后,将看清产业的发展方向,跟上未来 Windows 应用开发的潮流。

作为软件工程师,将克服在众多软件机构中颇为流行的“not-invented-here”心理,将在使用其他 Windows 应用程序方面变得知识丰富,并且学会如何在自己程序中使用这些应用程序现有的功能。最后,将学会如何通过套用 C 和 C++ 的功能来扩展 Visual Basic 的应用程序,进而加快你的工作。

作为管理人员,将了解当前在 Windows 开发中什么是可能实现的。这样能够第一手地了解到通过把应用程序连接在一起,能够做什么,不能做什么,而不仅是根据应用程序开发人员的陈述。你将学会一些开发的捷径,从而大大缩短开发周期,节省宝贵的时间和资源。

作为其他人员,将进入一个令人兴奋而重要的 Windows 应用程序开发的新阶段,即混合语言编程阶段。无论你的兴趣是在于把别的语言结合到自己的应用程序中,还是想充分利用现有 Windows 应用程序提供的功能,都将在后面的内容中找到有用的东西。

引 言

本书的总体内容是什么？

本书讲述 Windows 下的混合语言编程。混合语言编程究竟指的是什么呢？我们指的是利用各种各样的组件(component)进行编程。一个组件可以是一种程序设计语言,例如 C, Visual Basic 或者 C++,也可以是一个编程用应用程序(programming application),例如 Microsoft Word。一个组件也还可以是应用程序本身,例如 Windows 程序管理器(Windows Program Manager)。

本书的目的是教你怎样在不编写大量代码的前提下编制自己的 Windows 应用程序。毕竟,借用现存的代码总是比自己编写代码更好(也更快),我们为什么不利用这些现存代码的优势呢？

在微机编程的早些日子里,混合语言编程意味着利用汇编程序来加快应用程序中的关键部分。混合语言用于同样的目的,但它包含多种不同的内容。例如,我们可能利用动态数据交换(DDE)将一条请求发给一个字处理程序以格式化和打印一个文档。也可能向 Windows 程序管理器发送一条请求以便在用户工作台上安装一个新的程序项。其他可能的包括通过利用一些用户友好接口性能较差而功能更强的语言(例如 C)的功能来扩充当前使用的语言(例如 Visual Basic)。本书将讨论所有的这些可能性,并且教你如何借助别人的优势扩展你的 Windows 编程能力。

本书假定读者已具有 Windows 编程的最基本的经验。我们不打算讲述编程或语言语法。本书是为有经验的 Windows 程序员和程序管理人员准备的。为了从本书中学到尽可能多的东西,最好在学习本书的同时,利用与书本配套的磁盘,分析一下提供的例子。

目前已有很多优秀的书籍讨论了 Windows 编程环境的细节部分,对这些部分我们在此有所触及。对于每个环境,我们没有进行很深入的讨论,而忽略了各个环境中大量的细节问题。相反,我们把目标集中于更广的范围,对其中的各个环境提供了完整的开发景象。

如果你是一个程序员,将会发现大量可以被自己的应用程序借用的源代码行。仅这一点就值得去阅读本书。然而,我们希望,在学习本书后,你会考虑开发与其他应用程序协同工作的程序,而不是与其他应用程序竞争、完成相同工作的程序。尽管起初较为难于掌握,但从长远来,这些内容将为你节省大量的时间和精力。

假如你是一名程序管理人员或其他管理人员,将在本书中找到大量宝贵的信息,使你知道开发人员可以利用哪些开发环境,它们各自有哪些优、缺点。另外,还能获得一些有用的建议和技巧,并因此而使你的开发进程缩短几天或几星期的宝贵时间。

将从本书学到什么？

在本书中,你将学会以下内容:从 Visual Basic 到 C++(包括 Borland 和 Microsoft 编译器)的接口,在 C 或 C++中使用 Visual Basic 的控制,与 Windows 程序管理器的接口,与 Microsoft Word 的接口(作为与其他应用程序对话的例子),以及在代码中利用现存的库(包括静态和动态链接库)。

在书和配套磁盘中的程序代码涉及到以下几个方面:数据库管理,产生 Microsoft Word 格式文件,从 Visual Basic 和 C++中创建程序管理器组以及在你的应用程序中运行 Microsoft Word(和其他应用程序)。我们将讨论从你的程序中运行 Word 宏,还将讨论其他一些有趣而鲜为人知的 Win-

dows 细节。

各种信息

本书中的全部软件是利用以下编译器和应用程序开发的：

- Microsoft Visual Basic Version 3.0
- Microsoft Visual C++ Version 1.0
- Borland C++ Version 3.1
- Microsoft Word Version 2.0c
- Microsoft Windows Version 3.1

我们作了各种努力以使代码具有尽可能好的可移植性。另外，我们下了很大功夫来划分各部分，它们对应于不同的销售商。所有这些代码只需极小的改动，就可以在其他销售商的编译器下加以应用。

组织

本书分为三大部分。

第一部分介绍我们将要讨论的各种环境。在此，讨论了各种环境的优缺点，也提到了最好地使用该环境所需的一些有用的建议和技巧。对于每个环境，不仅讨论了它能做什么，也讨论了如何把它结合到一个更完整的系统中去。

这一部分没有大量的代码。我们给出了一些小的例子程序帮助你了解系统，还有一些代码段用以说明我们提供的建议和技巧。

第二部分包含了本书的大部分代码。在这一部分，我们将开发一个完整（尽管相当简单）的数据录入系统，此系统包含 Visual Basic 中的窗体（form）、一个数据库（在 C 和 C++ 中），一个报告打印/显示工具（利用 Microsoft Word）和一系列使我们能结合任何应用程序的工具。利用这一部分所提供的代码，你可以做到：

- 从 Visual Basic 中调用 C 和 C++ DLL；
- 产生与 Microsoft Word 兼容的文件；
- 执行其他 Windows 应用程序或与之通信；
- 用几乎任一种语言存储和查找在永久 ISAM 文件中的数据项；
- 学会如何结合各种不同的 Windows 环境创建尽可能好的编程环境。

最后，第三部分（附录）包含配套磁盘中的软件的重要文档，以及一些重要的技术说明。这一部分主要用于参考，基本上由事实（fact）和图表组成。

目 录

致谢

本书的用途

引言

第一章 Visual Basic	(1)
1.1 引言	(1)
1.1.1 什么是 Visual Basic	(1)
1.1.2 功能	(4)
1.1.3 不足之处	(6)
1.1.4 功能与不足之处的小结	(6)
1.1.5 小结	(7)
第二章 Borland C++	(9)
2.1 引言	(9)
2.1.1 什么是 C++	(10)
2.1.2 如何与 Windows 关联	(18)
2.1.3 Borland C++ 与其他语言相比如何?	(20)
2.1.4 功能	(24)
2.1.5 不足之处	(25)
第三章 Visual C++	(29)
3.1 引言	(29)
3.1.1 什么是文档和视图	(44)
3.1.2 这如何有助于混合语言编程	(45)
3.1.3 功能	(45)
3.1.4 缺点	(47)
3.1.5 小结	(49)
第四章 Word For Windows	(51)
4.1 引言	(51)
4.2 从程序中启用 Microsoft Word	(52)
4.3 在 Visual Basic 中同 Word 接口	(61)
4.3.1 一个更加复杂的例子	(63)
4.3.2 创建 Word 的自然格式文件	(64)
第五章 用混合语言设计	(101)
5.1 到目前为止我们已知道了些什么?	(101)
5.2 懒惰程序员原则	(102)
5.3 设计一个 Windows 应用程序	(103)
5.4 设计用户接口	(106)

5.5	混合语言编程的设计问题	(113)
5.6	下一步做什么?	(114)
第六章	定义的项目	(115)
6.1	我们要完成些什么?	(115)
6.2	项目概观	(115)
6.3	创建产品的步骤	(116)
6.4	设计项目	(117)
6.5	设计	(120)
第七章	数据录入屏幕	(133)
7.1	下一步做什么?	(133)
7.1.1	设计一个数据录入屏幕	(133)
7.1.2	数据录入流程	(133)
7.1.3	首次联系日期	(134)
7.2	设计数据库	(147)
7.2.1	与用户一起复审设计	(149)
7.2.2	往数据库中增加新记录	(166)
7.2.3	往数据库中增加记录	(173)
第八章	与用户打交道	(181)
8.1	如何检索数据	(181)
8.2	编辑数据库中的数据	(184)
8.3	实现处理	(185)
8.4	从数据库中取回记录	(190)
8.5	下一步做什么?	(201)
第九章	报告和打印	(203)
9.1	如何报告我们的行动	(203)
9.2	处理给客户的回信	(203)
9.3	创建 Microsoft Word 邮件归并文件	(204)
9.4	所有这些对程序员有何好处?	(208)
9.5	生成、显示和打印报告	(210)
9.6	重用、重用、重用	(212)
9.7	总结	(217)
第十章	安装	(219)
10.1	什么是程序组和项	(219)
10.2	与程序管理器接口	(221)
10.3	建立安装程序	(229)
10.4	解释批处理命令	(229)
附录 A	ISAM 用户指南	(239)
A.1	“ISAM”的含意是什么?	(239)
A.2	使用高级 ISAM 函数	(239)
A.2.1	设计数据库	(240)

A. 2. 2	例子	(240)
A. 2. 3	创建一个 ISAM 文件	(242)
A. 2. 4	创建一个键描述符	(243)
A. 2. 5	更容易的方法	(245)
A. 2. 6	打开一个已有的数据库文件	(247)
A. 2. 7	向数据库添加数据	(249)
A. 2. 8	在数据库中查找数据	(253)
A. 2. 9	更新数据库中的数据	(255)
A. 2. 10	游历数据库	(256)
A. 3	顺序存取和 ISAM 系统	(259)
A. 3. 1	关闭数据库文件并将缓冲区内容储到磁盘	(260)
A. 3. 2	向数据库添加索引	(261)
A. 3. 3	使用多个索引	(261)
A. 4	移植和扩充 ISAM 系统	(267)
A. 5	低级 B 树函数	(267)
附录 B	ISAM 参考指南	(273)

第一章 Visual Basic

1.1 引言

当 Visual Basic 被引入到 Windows 程序设计领域时,人们马上反应出两种态度。一部分人(有经验的 C/SDK 程序员),对使用象 BASIC 那样琐碎的工具编写程序的思想表示轻蔑。另一部分人,把 Visual Basic 系统称为程序设计的第二次来临,他们认为该系统将取代所有其他的 Windows 程序设计形式,并且将使现在的程序员过时。

客观地说,我们会认识到,上面提到的两部分人都是错的。Visual Basic 绝不会取代 C 或者后继者 C++。另一方面,Visual Basic 已经产生了新一代的使用 Windows 环境的 Basic 程序员。更重要的是,Visual Basic 找回了临时的程序员,这些临时程序员过去在 Windows SDK 出现后曾经消失了。

在某个时期,那些做梦也不会想到为谋生而编程的人们通过编写简单的程序使他们的生活变得更轻松。只用几分钟就编出来的一个简单的 Basic 程序常常能缓解一个长期存在的问题。随着宏语言引入各种应用中,特别是当 Quick Basic 出现在 MS-DOS 操作系统中时,这种趋势更得以进一步发展。临时程序员现在几乎是随处可见。然而,Microsoft Windows 的出现改变了这一切。突然间,那些简单的小程序需要用到数百乃至上千行晦涩的 C 代码。过去人们仅用二、三十行 BASIC 代码处理一个简单的地址列表,在 Windows 中猛增到 20,000 行程序,而只完成同样的工作。虽然学习足够的 BASIC 知识以编写简单的程序只需几分钟时间,但要学习庞大的 Windows SDK 可能要几个月甚至几年。这使得在 Windows 环境下的临时性编程难于风行。

Visual Basic 却使临时性编程获得了新生。人们又能够在几分钟(而不是几个月)内编出解决一些问题的程序了。程序员在几天或几星期中就能开发出一些新的实用程序和游戏,Windows 的市场更加繁荣起来。

真正的程序员也同样感受这种影响。突然间,他们的同事不再让他们编制一些简单的实用程序以改进他们的商业工作了。这些同事们自己编写那些程序,而让程序员去作他们的本职工作。Visual Basic 可能并未改变这个世界,它当然也不会消失,但它对微机市场所产生的深远影响远远超过了起初人们的猜测。Microsoft 公司最近宣布类似 Visual Basic 的工具在他们的应用程序中将作为通用的宏语言。因此,Visual Basic 在将来很可能占据越来越重要的地位。

由于 Visual Basic 有两种可选用的风格——标准版本和专家版本,用户很容易购买一个简单的工具来扩展 Windows 环境。所有人都得益于这种易于使用的工具。Microsoft 无需在将来的操作系统中加入太多的东西,而临时程序员在程序设计领域仍会充分发挥作用。

1.1.1 什么是 Visual Basic

Visual Basic 是一个完整的编程环境。它用数据输入窗体(form)作为设计和实现 Windows 编程的基础,窗体采用了一种简单的拖放(drag and drop)方法。

用 Visual Basic 编程很简单。拖动一个窗口到你想要的大小,然后,把控制(control)(编辑文本,组合框、列表框等)放在窗口中你选定的地方。这些控制通过选取和改变相关的特性来调整。例如,

一个简单的静态文本控制在正常的情况下屏幕上的显示是,其中有一文本采用了默认的前景和背景颜色和标准的 Windows 显示字体。通过选择你想改变的属性,你能够很快且方便地改变字体、颜色控制中的文字。完成这些工作无需写任何代码。而且,你可以修改任意多次,而不会造成别的影响,也无需重新编译。

Visual Basic 用窗体的概念来模型化 Windows 工作台上的任何输入和显示。控制的概念使数据录入和显示工具模型化。例如,控制可用于输入一文本串,一类 COBOL 的图形串,或者处理按下按钮、组合框等等。控制为你做了所有的工作。你所需要做的是指定控制需要使用的各种特性以及控制要处理的默认值。

除了上面提到的由用户作简单文本录入的控制,Visual Basic 还带有许多更复杂的控制。Visual Basic 环境提供了处理时间事件(后台处理)、串行通讯、电子表格表项的控制及其他一些控制。标准版本和专家版本都包含以上这些控制,但第三方销售商提供了许多别的控制。图 1.1 描绘了一个典型的 Visual Basic 屏幕。通过使用工具杆或菜单你可以方便地使用你所需要的工具。程序可在环境中设计、修改、测试和运行。象在后面的各章中将要看到的那样,Visual Basic 在许多情况下可以作为更复杂和更重要的编译器的原型开发环境。

Visual Basic 通常以二进制编码格式存储其窗体。对于普通用户这在一定程度上是难于理解的。然而,也可以用 ASCII 格式存储,这样用户可以看到文件中的内容,但取出时速度较慢。

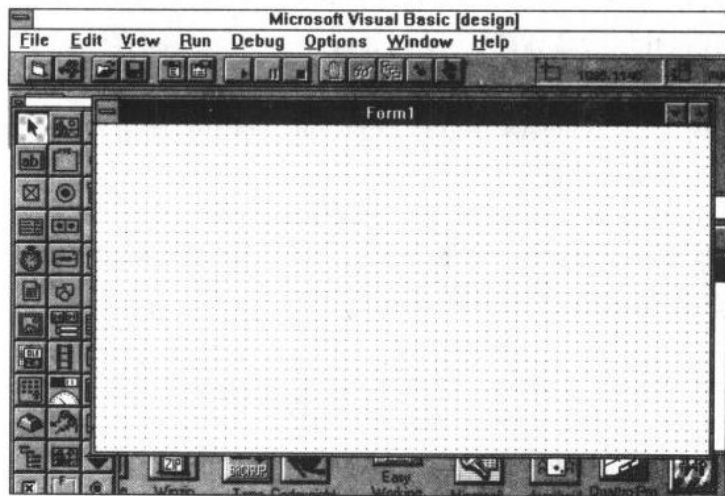


图 1.1 Visual Basic 程序设计环境

程序 1.1 描绘了一个典型的 Visual Basic。图 1.2 说明了该环境中携带的以普通图形方式显示的窗体。

程序 1.1 用 ASCII 格式的 Visual Basic 打印程序

```
VERSION 2.00
Begin Form Form1
    BackColor   = &H00808080&
```

程序 1.1 用 ASCII 格式的 Visual Basic 打印程序(续)

```
Caption      = "Form1"
ClientHeight = 4020
ClientLeft   = 1875
ClientTop    = 1920
ClientWidth  = 7365
FillColor    = &H00808080&
Height       = 4425
Left         = 1815
LinkTopic    = "Form1"
ScaleHeight  = 4020
ScaleWidth   = 7365
Top          = 1575
Width        = 7485

Begin TextBox Text1
    BackColor    = &H00808080&
    Height       = 285
    Left         = 2400
    TabIndex     = 2
    Text         = "Text1"
    Top          = 840
    Width        = 3615
End

Begin ComboBox Comb1
    BackColor    = &H00808080&
    Height       = 300
    Left         = 2400
    TabIndex     = 0
    Text         = "Comb1"
    Top          = 360
    Width        = 1215
End

Begin Label Label3
    Alignment    = 2.'Center
    Caption      = "this text is in blue on white"
    ForeColor    = &H00FF0000&
    Height       = 375
    Left         = 1680
    TabIndex     = 4
    Top          = 1560
    Width        = 3615
End
```

程序 1.1 用 ASCII 格式的 Visual Basic 打印程序(续)

```
Begin Label Label2
  BackColor   = &H00808080&
  Caption     = "Second Field:"
  Height      = 255
  Left        = 840
  TabIndex    = 3
  Top         = 840
  Width       = 1215
End
Begin Label Label1
  BackColor   = &H00808080&
  Caption     = "First Field:"
  Height      = 255
  Left        = 840
  TabIndex    = 1
  Top         = 360
  Width       = 1215
End
End
```

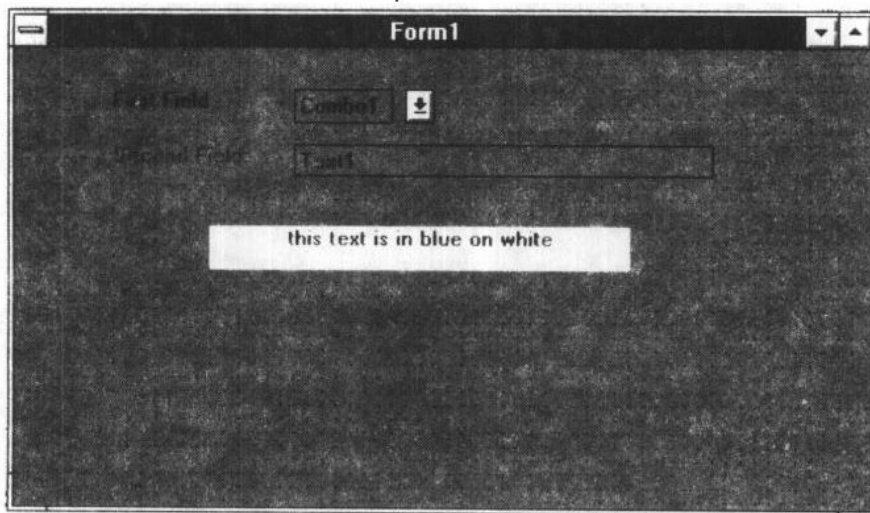


图 1.2 一个 Visual Basic 窗体窗口

1.1.2 功能

与其他程序设计语言和环境一样, Visual Basic 具有一定的功能,也有一些不足之处。类似地,它的某些特性使它适于完成某些工作,而另一些工作却根本不能或者无法用它完成。我们在此先集中讨论该语言的功能及其适应的领域。

第一个也是最重要的功能是在 Visual Basic 环境下,应用程序能快速地设计、实现、测试和重写。Visual Basic 重新定义了一些已有的概念,规定如何开发应用程序。在过去的几年中,我们在编写代码前,总是先在稿纸上作好设计工作,并且以一个一个模块作为基础,这几乎成了公理性的规则。

正是由于在计算机上开发程序的时间的代价提高,以及大多数语言采用自底向上开发方法,这种方法得以巩固。

Visual Basic 允许你先设计一个窗体,然后看看感觉如何,此时你无须写任何代码。大多数语言要在屏幕上完成同样的设计工作却需要几百甚至几千行代码。因此容易理解为什么要采用这种设计。哪个正常的人愿意先打一份草稿,再重写所有的代码呢?

编程简易性当然也是一个要考虑的因素。由于采用点拾取程序设计风格,Visual Basic 易于作用。甚至从没有编过程序的人也可能设计出带有菜单和对话框的输入窗体,甚至还能完成复杂的文件操作而完成这些工作却无需任何代码。这使 Visual Basic 充分展示了其简单性。由数百行晦涩的 C 代码或 C++ 代码所完成的操作只需要使用一两行这种易于学习的 Basic 语句。

Visual Basic 环境的可扩充性也应视作其最有价值的优点之一。自制控制正以惊人的速度涌入市场,这使得程序员只需把精力集中于应用程序应该做什么,而无需考虑代码应怎样实现最新的用户接口,或(不熟悉的内容)应怎样编写。当然,每一个有一两年以上编程经验的程序员都习惯于老板拿出一个别人的程序,并宣布这个系统要做一些“填空”(fill in the blank)工作。当别的程序员做“填空”工作时,应用程序设计人员和程序员只须关心他们想做什么,而不在意他们实现一个特有功能所需的时间。

无论如何,从总体上来看,可扩充性解决了两个大问题。首先,它终于允许真正代码重用了。由于通用的自制控制中的代码完成了应用程序的大部分工作,你完全不需要去考虑代码重用的问题,代码重用完全自动地实现了。第二,也是对最终用户更为重要的一点,所有程序倾向于包含同类的控制和特性。这使得在 Windows 环境从一个程序转移到另一个程序非常方便。Windows 提供了一个一致的用户接口。Visual Basic 的自制控制已开始使用这种特性。

Visual Basic 系统的另一个功能在于测试方面。当程序员一旦需要用到硬件调试器和令人头痛的消息框信息来查找程序故障时,运行时调试器能帮助程序员很轻易地找到问题所在。同时,内嵌的编辑器和解释器使得在运行时刻几乎不可能出现语法错误。因为 Visual Basic 基本上一一种解释执行语言,大多数错误在进入 Windows 系统前就已发现。这当然是最终用户所欢迎的特色。

当前,在程序设计领域,应用程序的快速原型设计已成了一个很寻常的行话。Visual Basic 使你能够迅速而容易地改变和测试数据录入屏、对话框和菜单,而本身作为一个“What-if”环境。管理人员,最终用户和测试人员能够很迅速地把那些仅仅完成了基本框架的应用程序放在一起,看起来就象是一个已完成的产品。事实上,目前许多公司正通过使用 Visual Basic 使得各设计小组能向程序员说明应用程序在屏幕上如何显示,应该完成什么工作。比起以前在纸上进行说明,用 Visual Basic 要方便得多。

对程序员来说,Visual Basic 在 BASIC 语言的基础上提供了高级串处理(string handling)另外,文件操作相当容易,而且迅速,并且允许使用随机存取文件或基于文本的文件。此外,可以很容易地向一个程序添加新的模块和全局信息,且无须顾及在此过程中会产生别的什么变化。

最后,应该考虑一下使用 Visual Basic 的思想是什么。作为一个可用于模拟演示,设计应用程序和向程序员说明应该建立什么的快速原型开发工具,此系统与其他多数系统不能比拟的。Basic 适用于基本面向用户接口的应用程序。除个别例外,象日历,人员信息管理器 and 计算器等应用程序能用极少的代码方便地编写出来。然而,Visual Basic 真正的作用在于作为别的运行于主机服务器上的重要应用程序的前端。一些大公司正使用简单的 Visual Basic 前端菜单和数据录入屏来从大的主机程序中存取数据(这些大型程序往往用户接口不那么友好)。

1.1.3 不足之处

任何语言的功能在于吸引程序员和管理人员选择并长期使用它。然而语言中的不足之处妨碍了它的流行并使它失去市场份额。

象其他所有语言一样, Visual Basic 也存在一些缺点。在此我们对其中的几个缺点作一些讨论, 并且说明它们对设计工作有何不良影响。

最大的缺点之一是 Visual Basic 根本上是一种解释执行语言, 这也是设计 Visual Basic 的本身的目的。尽管这种特性使 Visual Basic 具有极大的灵活性, 但在速度方面却降低了性能。Visual Basic 显然不可能象 C 或 Pascal 这类编译性语言那样快。它并不产生真正的机器码, 而只是通过解释器直接执行指令。速度性能较差是 Visual Basic 不能用于重要软件的唯一最大原因。对于用户接口软件, 你却更严格地受限于键入选择的速度。由于这个速度大大低于解释器处理执行指令的速度, 两种语言(解释执行语言性编译语言)的差别也就不那么重要了。

Visual Basic 是一个解释器。因此存在一个问题, 即它需要一个解释程序和运行(run-time) DLL。这个运行时间 DLL 要处理各个应用程序, 因而必须是分布式的(distributed), 这就带来了两个小问题。首先, 为了使你的软件运行, 必须确保 DLL 已正确安装好。第二, 用户不可能隐藏其实现的语言。这使得许多用户把 Visual Basic 编程视作儿戏。

从一个 Windows 程序员的观点来, Visual Basic 在一个更重要的特性方面存在一些其他的问题。即 Visual Basic 本身未提供指针。而指针是 C 和 C++ 编程的生命线。由于这个原因, 同时也因为 Windows 环境中的许多实例需用到指针来存取低层数据, 某些工作直接用 Visual Basic 根本不可能完成。作为代替, 必须通过 DLL 与 Windows 接口, 这进一步降低了系统的速度。

另外, Visual Basic 不支持回调(callback)。回调是指 Windows 对在其环境中发生的事件作出响应而执行的用户编写的过程。尽管 Visual Basic 的特性和控制过程使许多回调变得不必要, 但仍存在着大量的必须使用真正的 C 回调函数的事件。例如, 事件划分为子类需要用户编写的回调。Visual Basic 需要一个高级语言编写的附加 DLL 来处理这样的事件。

Visual Basic 不能创建自制控制。这是一个奇怪且很严重的限制。Visual Basic 由于自制控制提供的可扩充性而兴起, 自制控制允许完成本未打算做的事情, 但如果不使用其他语言, 它本自无法扩充。自制控制必须用 C 或其他高级语言编写并要求你从 Microsoft 购买一个附加的组件(CDK)。

因为 Visual Basic 由 BASIC 语言发展而来, 因而其浮点数字运算功能也有所不足。BASIC 从不适用于数学运算, 不应用于编制需高精度和高速度的应用程序。

最后, Visual Basic 没有内嵌的浏览器(browser)以帮助你浏览所有用于处理各种事件的子程序。由于没有浏览器, 要想跟踪哪个事件正在被处理、哪个事件不需要处理或哪个事件完全被忽略, 几乎是不可能的。第三方的嵌入能缓解这一问题, 但需要花费更多的资金。由于缺乏浏览器, 用 Visual Basic 编写复杂的应用程序或者多人合作都是不可能的。

1.1.4 功能与不足之处的小结

总的说来, 与其他程序语言一样, Visual Basic 有一定的功能, 也存在着缺点。此外, 对某些应用程序来说, 它是一个理想的选择, 而对另外一些应用程序, 它却很不合适。有些应用程序需很快完成, 但较粗糙, 它们基本面向用户接口且作为与其他应用程序、数据库或设备通讯的应用程序的前端。对这类应用程序, Visual Basic 是相当精巧的语言。另一方面, 对于较大的科学计算程序, 或者高速度、实时响应的应用程序, Visual Basic 却显得很拙劣。

Visual Basic 编程环境鼓励单人编程小组。Visual Basic 环境产生的代码不可移植,也不容易在程序员间共享。由于对视察代码结构和模块没有内部的支持,一个程序员很难理解别的程序员的代码。象其他 BASIC 程序设计语言那样。Visual Basic 含有大量的 GOTO 和 GOSUB 代码。尽管作了一些扩充,使得比其他许多 BASIC 环境更为友好,但它不是结构化设计语言。

Visual Basic 向程序员、管理人员和测试人员提供了一个易于使用的前端生成器,用于生成数据录入屏,对应用程序以及 what-if 风格设计和编程问题的模拟和验证。仅仅由于它的快速原型开发特性,这个环境在任何编程机构都应受到重视。

运用自制控制的能力,完美地与 ODBC(和 Microsoft Access)数据库的结合能力和利用第三方工具扩充环境本身的能力使 Visual Basic 成为大多数数据机构所使用产品的有力竞争者。

最后,我们应注意到 Microsoft 在 Visual Basic 的最新版本(3.0)中加入了很多功能。Microsoft Access 关系数据库驱动(engine)已包含在该系统中,它支持 OLE 2.0 containers and automation。Visual 数据控制已加入到与数据库后端完美结合的前端中,并包含了 DDBC、SQL Server、Sybase 和 Oracle 关系数据库的驱动程序。

此外,该系统加入了几个新的工具,它们是 Crystal Reports Setup Wizard(用于产生自制的设置程序)。一个新轮廓(outline)控制允许你定义分层列表框(list-box),新的报告控制允许你在应用程序中嵌入报告。

最后,Visual Basic 程序员现在已开始运用 C 和 C++ 程序员使用了多年的普通对话控制了。现在,在 Windows 操作系统环境中标准的文件打开和存储对话(dialog),和打印对话以及一系列其他系统对话都可以使用了。这最后一项大大有助于应用程序的用户友好特性,也使应用程序符合于 Windows 标准。

1.1.5 小结

本章对 Visual Basic 作了介绍,从这一章你应该学到了 Visual Basic 如何工作,它有什么功能和不足之处。总的来说,我们已经知道,Visual Basic 在设计和构造用户接口方面是很出色的,但它不适于作为商业和科学应用程序的后端。