



教育部人才培养模式改革和开放教育试点教材

实用软件工具与环境教程

——C++高级编程

谭德强 姜 波 主编 殷人昆 审



清华大学出版社

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

教育部人才培养模式改革和开放教育试点教材

实用软件工具与环境教程

——C++高级编程

谭德强 姜波 主编

殷人昆 审

清华大学出版社

(京)新登字 158 号

内 容 提 要

近年来软件开发工具日渐成为人们关注的焦点,而且它在软件开发中的作用非常重要。本书一共分为 13 章,依次介绍了如何利用 Visual C++ 这个软件开发工具来实现多线程、多媒体、动态链接库、数据库和网络编程;然后讨论了当今最流行的软件开发方法,即组件对象模型的理论以及如何使用软件开发工具和如何实现容器、服务器、自动化、ActiveX 控件;最后讲述了混合编程和自制 AppWizard 实现的编程。

作者根据多年的编程经验,把 Visual C++ 这个工具讲得深入浅出,易于读者掌握。本书实例丰富、理论清晰、可读性强,适合作为大学院校的教材,也可以供具有 Visual C++ 基础的读者自学使用。

版权所有,翻印必究

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:实用软件工具与环境教程——C++高级编程

作 者:谭德强 姜波

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>

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:21.75 字数:524 千字

版 次:2001 年 6 月第 1 版 2001 年 6 月第 1 次印刷

书 号:ISBN 7-302-04425-2/TP·2601

印 数:0001~8000

定 价:28.00 元

序

近年来计算机技术突飞猛进，尤其是硬件发展更快，而软件的发展相对要落后一些。但是人们对于软件的需求与日俱增，开发软件的难度不断地增加，如何解决这两者之间的矛盾呢？

因此，有人提出软件工程的思想，它从理论上可以解决这个难题，但是在实际工作中实现方式却是通过工具来实现的。正如人类历史的发展一样，人们不断地创造工具来推动社会向前发展。软件的发展也不例外，从当初没有任何工具，开发人员自己编写所有的代码，到现在依靠工具生成一系列的代码。在当今所有的开发工具中，可以说 Visual C++ 是最好的一种软件开发工具，它通过各种各样的方法，使 C++ 语言得以实现，而且能非常方便地进行 Windows 编程。

Visual C++ 提供了新颖的可视化设计工具，采用面向对象的方法将 Windows 编程的复杂性封装起来，实现了将可视化的编程与面向对象的程序设计的有机集成。作为现在最流行编程语言的 C++，其特点使其成为编程者最喜爱和最常使用的编程工具。所谓“真正的程序员用 VC，聪明的程序员用 C++ Builder”。

本书从软件开发工具开始，一步一步地引到 Visual C++ 这个工具中来，然后以理论和实践相结合的方法介绍当今最重要的一些编程和编程思想。

本书包括的内容是：软件开发工具和 Windows 编程的介绍、多线程编程、多媒体编程、动态链接库编程、数据库编程的思想、网络上的程序设计、组件对象模型的理论介绍、控件的实现、容器和服务器的实现、自动化编程、ATL 编程、混合编程和自定义 Visual C++。

在本书的编著过程中，我们尽量注意减少冗长无味的说明，代之以具体实用的例题演示；同时力争在理论和实践中找到一个平衡点。通过实例，引导读者把握 Visual C++ 的精髓所在；同时在讲述实例之后进行理论阐述，让读者不仅知其然，而且知其所以然。

本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于中级及中级以上的用户，尤其是用作教材。对其他有一定使用经验的读者，也可通过本书掌握本书深层次的开发方法，学会用更巧妙的办法开发出高水平的 Visual C++ 应用程序。

本书由中科辅龙计算机技术有限公司抖斗书屋策划，第 2、3、4、5 章由姜波编写，第 1、6、7、8、9、10、11、12、13 章由谭德强编写。另外参加编写的有：郭美山、徐平、郑红、魏红、王艳燕、刘小华等同志，全书由杨桂莲、石利文统稿。

在编写过程中，得到了杨警、范小明和魏昆的极大支持，他们为本书提供了许多技术性的建议，在此表示衷心的感谢。另外在本书的编写和编排过程中，还得到了郭颖、王宇喜、邓伟、鞠盈、田勇强、袁园、施晓峰、杨志、张宇、李柳平、孟长新、李红军、黎明、熊刚、吴袁杰、杜林、李知量的很大支持和帮助，在此也表示由衷的感谢。

本书的编写大纲及内容均经清华大学计算机系殷人昆老师审阅。

由于时间仓促、作者水平有限，本书错漏之处在所难免，欢迎广大读者批评指正。
对本书内容有疑问的读者，可向抖斗书屋读者服务部提出咨询。

咨询电话：010-62346896 转 315

E-mail: replybook@126.com

网址: www.doudou.com.cn

最后祝大家都能如愿以偿地掌握 Visual C++!

作者

2000年8月

目 录

第 1 章 工具的选择和 Windows 程序设计的原理	1
1.1 软件和软件开发工具的概述	1
1.1.1 软件及其发展过程	1
1.1.2 软件工程	2
1.1.3 软件开发工具	3
1.2 开发工具的选择	4
1.2.1 Visual Basic	4
1.2.2 Delphi	6
1.2.3 C++ Builder	7
1.2.4 Visual C++	7
1.3 工具的实现——Windows 编程	9
1.3.1 Windows 操作系统特点	9
1.3.2 Windows 应用程序设计的特点	10
1.3.3 用面向过程技术进行 Windows 应用程序开发	12
1.3.4 用面向对象技术进行 Windows 应用程序开发	19
1.4 本章小结	25
1.5 思考和练习	26
第 2 章 多任务与多线程编程	27
2.1 程序、进程和线程的概述	27
2.1.1 多任务、进程和线程	27
2.2 线程的种类	28
2.2.1 MFC 中的线程类	29
2.2.2 用户界面线程(UI)	31
2.2.3 工作者线程	31
2.3 线程的创建、启动和终止	31
2.3.1 线程的创建	31
2.3.2 线程的启动	37
2.3.3 线程的终止	37
2.4 线程的操作和管理	38
2.4.1 线程运行状态的设置	38
2.4.2 线程的优先级	38
2.4.3 线程间的通信	40
2.5 在 VC++ 环境中使用同步对象	41
2.5.1 事件对象	41

2.5.2 临界区	46
2.5.3 互斥量	49
2.5.4 信号灯	52
2.6 本章小结	53
2.7 思考和练习	53
第3章 多媒体编程	54
3.1 多媒体概述	54
3.1.1 视觉类媒体	54
3.1.2 听觉类媒体	55
3.1.3 Win 32 的多媒体服务	55
3.2 VC++开发工具中声音的处理	56
3.2.1 波形声音的特点	57
3.2.2 波形音频文件的播放	57
3.3 VC++开发工具中图像的处理	72
3.4 VC++开发工具中视频的处理	76
3.5 本章小结	81
3.6 思考与练习	82
第4章 动态链接库	83
4.1 动态链接库 (DLL) 的概述	83
4.1.1 DLL 的定义	83
4.1.2 DLL 的种类	84
4.1.3 动态链接库和静态链接库的比较	84
4.2 DLL 的基本理论	84
4.2.1 DLL 入出口函数	84
4.2.2 DLL 的链接方式	86
4.2.3 动态链接库库函数的导出	87
4.2.4 从 DLL 中导出数据变量	89
4.3 用 Visual C++开发环境生成 DLL	89
4.3.1 用 Visual C++生成 Win32 DLL	89
4.3.2 用 Visual C++的 MFC 生成的 DLL	90
4.4 使用 DLL	92
4.4.1 DLL 的使用	92
4.4.2 DLL 的调试	92
4.5 用 VC++工具开发 DLL 的编程实例	93
4.5.1 用 MFC 生成 DLL	93
4.6 本章小结	104
4.7 思考和练习	104

第 5 章 数据库编程	106
5.1 数据库编程的概述	106
5.1.1 数据库及数据库管理系统	106
5.1.2 软件开发中有关数据库编程的几个关键概念	106
5.1.3 MFC 的 ODBC 类简介	108
5.1.4 数据库的基本操作	113
5.2 ODBC	115
5.2.1 ODBC 概述	115
5.2.2 ODBC 的实现	116
5.3 DAO	117
5.3.1 DAO 概述	117
5.3.2 DAO 和 ODBC 的异同	118
5.3.3 MFC 中的 DAO 类	120
5.4 VC 软件平台中的数据库编程实例	127
5.4.1 数据库的创建	127
5.4.2 配置 ODBC 环境	130
5.4.3 创建 ODBC 应用程序	132
5.5 本章小结	142
5.6 思考与练习	143
第 6 章 网络编程	144
6.1 网络概述	144
6.1.1 客户/服务器模型	144
6.1.2 网络协议	145
6.2 网络应用程序编写	147
6.2.1 网络编程的概述	147
6.2.2 客户端的编程实例	152
6.2.3 服务器端的编程实例	156
6.2.4 浏览器和 MAPI 的编程实例	158
6.3 DHTML	166
6.3.1 DHTML 对象模型	166
6.4 本章小结	170
6.5 思考和练习	171
第 7 章 组件对象模型	172
7.1 COM 的历史足迹	172
7.1.1 OLE 1.0	172
7.1.2 OLE2.0	176
7.1.3 分布式组件对象模型 (DCOM)	177

7.1.4	ActiveX	178
7.2	COM 的基本理论	180
7.2.1	COM 的组成元素	180
7.2.2	COM 的实现函数	181
7.2.3	COM 的服务	183
7.3	ActiveX 的组成	185
7.3.1	ActiveX 容器	185
7.3.2	ActiveX 服务器	186
7.3.3	ActiveX 自动化应用程序	186
7.3.4	ActiveX 控件	186
7.3.5	ActiveX 文档	186
7.3.6	ActiveX 平台	187
7.4	本章小结	187
7.5	思考和练习	188
第 8 章	ActiveX 控件	189
8.1	一个最简单的实例	189
8.2	ActiveX 控件的简介	191
8.3	ActiveX 控件的属性、方法和事件	193
8.3.1	ActiveX 控件的属性	193
8.3.2	ActiveX 控件的方法	194
8.3.3	ActiveX 控件的事件	194
8.4	ActiveX 的创建过程	194
8.5	ActiveX 的测试	200
8.5.1	方法一	200
8.5.2	方法二	201
8.5.3	方法三	202
8.6	一个经典的实例	203
8.6.1	定制界面的大小和形状	203
8.6.2	添加库存属性	204
8.6.3	添加自定义属性	209
8.6.4	添加库存方法	212
8.6.5	添加自定义方法	212
8.6.6	添加库存事件	213
8.6.7	添加自定义事件	215
8.6.8	在工程中调试 ActiveX 控件	216
8.6.9	将 ActiveX 加到 Internet 网上	219
8.7	本章小结	222
8.8	思考和练习	223

第 9 章 容器和服务	224
9.1 用编程工具实现容器应用程序	224
9.1.1 利用 Visual C++ 的 AppWizard 创建应用程序的基本框架	224
9.1.2 管理嵌入对象的大小和位置	227
9.1.3 有关鼠标操作的设置	230
9.1.4 删除有关的项目	235
9.2 服务器应用程序的编写	236
9.2.1 用 AppWizard 创建服务器应用程序的框架	237
9.2.2 实现特定的功能	239
9.2.3 服务器应用程序运行方式的详细讨论	245
9.3 本章小结	248
9.4 思考和练习	249
第 10 章 自动化的实现	250
10.1 自动化服务器的编写	250
10.1.1 用 AppWizard 生成服务器应用程序的基本框架	250
10.1.2 容器应用程序资源的定制	252
10.1.3 完善文档类	252
10.1.4 完善视图类	253
10.1.5 定义自动化服务器的属性和方法	256
10.2 自动化容器应用程序	258
10.2.1 创建自动化容器应用程序的基本框架	259
10.2.2 修改自动化容器应用程序的资源	260
10.2.3 从类库中创建新类	260
10.2.4 完善视图类	262
10.3 本章小结	265
10.4 思考和练习	266
第 11 章 活动模板库	267
11.1 创建活动模板库	267
11.1.1 COM 和 ATL 的简介	267
11.1.2 ATL 简介	269
11.1.3 ATL 的创建过程	269
11.2 增加 COM 对象	272
11.2.1 COM 对象和控件的种类	272
11.2.2 ATL 对象 (Objects)	272
11.2.3 ATL 控件	273
11.2.4 ATL 杂项	274
11.2.5 ATL 数据访问	274

11.2.6 属性的设置	275
11.3 实现接口	280
11.4 Web 上加 ActiveX 控件	280
11.5 编程实例	281
11.5.1 创建控件的基本框架	281
11.5.2 添加控件	282
11.5.3 添加属性和实现接口	284
11.5.4 添加属性页	290
11.5.5 添加事件	295
11.5.6 将控件放到网上	296
11.6 本章小结	297
11.7 思考和练习	297
第 12 章 混合编程	298
12.1 混合编程的介绍	298
12.2 VC 和 Fortran(Visual Fortran)的混合编程	299
12.2.1 Fortran 编程的优点	299
12.2.2 Visual C++的好处	300
12.2.3 优点+好处=混合编程的威力	301
12.2.4 Visual C++与 Fortran 混合编程的规则	302
12.3 VC 和 VB 的混合编程	308
12.3.1 VC 和 VB 混合编程方式	308
12.3.2 VB 与 VC 混合编程中处理消息的方法	309
12.4 编程实例	310
12.5 本章小结	318
12.6 思考和练习	318
第 13 章 自定义 Visual C++的工具	319
13.1 自定义 Visual C++编程环境	319
13.1.1 Options 选项	319
13.1.2 Customize 选项	320
13.1.3 工具栏	322
13.1.4 命令行参数	323
13.1.5 参数宏	323
13.2 定制 AppWizard	324
13.2.1 AppWizard 的工作原理	325
13.2.2 自定义 AppWizard	325
13.3 编程实例	328
13.3.1 创建应用程序基本框架	328

13.3.2 改变 AppWizard 宏	329
13.3.3 定义 CTestAppWiz 类	330
13.3.4 使用宏字典	332
13.4 本章小结	333
13.5 思考和练习	333

第 1 章 工具的选择和 Windows 程序设计的原理

本章将主要介绍软件和软件开发的过程，探讨软件工程上常用的一些工具，并对各种软件开发工具进行比较，提出各自的优缺点。然后将讨论如何借助工具实现软件的开发，其中还将讨论 Windows 操作系统的特点和 Windows 程序设计的特点，对以前的 Windows 程序设计及现在的 Windows 程序设计进行比较。

1.1 软件和软件开发工具的概述

1.1.1 软件及其发展过程

和人类社会其他生产工具的发展过程相类似，软件和软件开发工具的发展也是从无到有，从简单到复杂，而且软件开发工具的发展和软件的发展紧密相关，下面简单回顾一下它们的产生和发展过程。

计算机软件发展的三个时期及其特点如下：

1. 早期时代（60 年代中期之前）程序设计阶段

当时的软件（实际是程序）还依附在穿孔卡上，人们只能用第一代计算机语言，即机器语言进行程序设计。机器的每一个操作都使用二进制的字符串来进行，其缺点十分明显，使用起来很困难，而且还得记住那些机器代码。当初，人们必须把他们所想做的“事情”转化成机器代码指令，表达成机器可以识别和执行的 0 1 字符串。此外，那时程序的编程刚刚起步，人们主要集中精力实现最基本的运算和操作。

另外，编程被看作是计算机的附属工作，地位较低。硬件通用，软件专用，当时重硬轻软的思想很严重，编程远没有设计硬件那么风光。程序规模小，编写者和使用者为同一人（同组人）。

2. 第二代（60 年代中期到 70 年代中期）程序系统阶段

在这个阶段，商业计算机公司如雨后春笋。一般来说，制造商自己提供程序系统（作为计算机的一部分），而由用户自己开发和使用应用程序。因此要购买计算机，还得招聘一组系统分析员和程序员，专门伺候机器。随着计算机应用的提高，人们对程序的需求越来越迫切。在这个时期，已经出现了一些高级语言，而作为硬件附属物的程序系统期待独立，并且出现“程序开发小组”、程序产品和“个体化”开发方法。

3. 第三代（70 年代中期之后）软件工程阶段

在 60 年代末，人们对于软件工作的认识大大深入，意识到了软件工作的困难性，这就是所谓“软件危机”问题的提出。

对于软件危机，它是指在计算机软件开发和维护过程中所遇到的一系列的严重问题。具体表现在以下几方面：

- 软件开发成本高，成本难以控制。
- 研制周期长，软件开发进度难以预计。
- 正确性难保证，软件质量差。
- 缺乏完整、正确的文档资料。
- 软件的维护困难，维护的工作量与费用不断增加。
- 软件的发展跟不上硬件的发展和用户的要求。

造成软件危机的原因有两方面：

- 客观原因 软件需求大，规模大。
- 主观原因 软件本身特点。

为了解决软件危机，出现了一门新兴的工程学科——软件工程。所谓软件工程是指指导计算机软件开发和维护的工程学科。它采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。下面我们还会详细讨论。

同时，在这三个时期的软件开发有如下特点：

1. 程序设计时期 使用汇编及机器语言编写程序，由程序设计者本人负责程序设计和维护。

2. 程序系统时期 使用高级语言编写程序，此外，还得编写程序说明书。在这个阶段由于程序需要迅速增长，软件开发人员已经不能满足软件的需求，出现软件危机。

3. 软件工程时期 产品软件（项目软件）使用先进的软件开发方法和工具开发，出现完整的开发周期，注重软件的重用和后期的维护工作。此时开发技术有了一定的进步，但是未获得突破性进展，软件危机未完全摆脱。

1.1.2 软件工程

对于软件工程的定义，我们已经进行了探讨。其实，软件工程的定义概括为“生产具有正确性、可用性以及开销合宜的产品。其开发步骤分为三个阶段。

1. 软件计划阶段

- (1) 问题定义，即要解决什么问题。
 - (2) 可行性研究，即在时间和资源的约束条件下，能否完成指定的任务。
- 其内容包括：技术可行性、经济可行性、法律可行性和社会可行性。

2. 软件开发阶段

软件开发阶段包括需求分析、软件设计、编码、测试和运行。

3. 软件维护阶段

为此，软件工程师们提出各种各样的软件开发模型，在此不再详细讨论，请读者参看有关的软件工程书籍。

1.1.3 软件开发工具

在软件开发的每一个阶段都有相应的工具支持，在此我们介绍一个最流行的开发过程管理软件 PVCS。

PVCS 是软件开发的基本支持软件，就像建造房屋或任何一个新的建筑物，大部分时间和精力要花在地基的构造，以及道路、布线、电力和水源等基础设施的建设上。PVCS 在软件开发过程中可以完善地管理软件系统中的多个版本；自动创建完整的文档，保障软件的维护；全面记载系统开发的历史过程，包括谁做了修改，修改了什么，为什么修改；管理和追踪开发过程中危害软件质量以及影响开发周期的缺陷和变化；管理需求分析等。

PVCS 为开发带来好处，同时强化其他软件工具的效率。

在 PVCS 的优势中，最为关键的一点是，它在以下几个方面给软件开发带来了益处：规范开发过程、缩短开发所需的时间、减少开发成本；完整详细地记载了开发历史过程，便于软件维护；通过排除开发中的错误，加强软件一致性和可重用性，提高软件质量。

今天的开发人员常常工作在包含有众多开发工具的环境中，如：编辑器、语言编译器、Debugger、数据库等。而在这样的环境中，PVCS 这种跨平台开发管理工具带来的效益十分明显。

在早些时候，Intersolv 向中国各大部委和院校赠送了这个工具软件，对于促进先进的软件开发管理技术在中国的普及、缩短国内软件产业与发达国家的差距大有好处。

PVCS 能为我国软件开发解决如下的问题。

1. 有效地保护用户投资，降低项目风险

PVCS 是开发项目成功不可缺少的守护神。在当前的软件应用开发中，往往一个应用系统投入运行后，由于开发过程缺乏有效的管理和控制，用户无法进行应有的维护，达不到预期的开发要求。PVCS Version Manager 则会完整地详细地记录开发过程中出现的变更和修改，并使修订版本自动升级，而且 PVCS Tracker 会自动地对上述变更和修改进行追踪。另外，PVCS Requisite Pro 提供了一个独特的 Microsoft Word 界面和需求数据库，从而可以使开发机构实时地、直观地对来自于最终用户的项目需求及需求变更进行追踪和管理，可以有效地避免重复开发，保证开发项目按期、按质、按原有的资金预算交付用户。

2. 加快投资回报

对开发过程有效地管理和控制，完整、明确地记载开发过程中的历史变更，形成规范化的文档，不仅使日后的维护和升级得到保证，而且更重要的是，还会保护宝贵的代码资源，积累软件财富，提高软件重用率，加快投资回报。

3. 支持 ISO9000 及 SEI CMM

标准市场竞争推动着标准化、规范化的软件开发模式,使越来越多的开发机构迫于市场和用户的压力,必须取得 ISO9000 质量认证并遵循 SEI Capability Maturity Model (CMM) 来改进自己的开发过程。这些质量标准为软件产品在开发过程中进行内部产品的评估和管理提供了框架。能否赢得更多的销售合同,取决于该企业是否已得到 ISO9000 及 SEI CMM 的质量认证。而软件开发管理工具 PVCS 是通过这些认证的基石。

PVCS 解决了曾在 70 年代波及西方发达国家的“软件危机”所出现的主要问题。事实已经表明,随着整个软件业的迅速发展,没有得到有效管理的软件开发过程中所出现的风险和挑战将越来越突出。加强软件开发管理,通过管理和追踪软件开发环境中产生的变更,建造一个规范、高效的软件开发基础结构,早已为西方发达国家软件产业的发展实践证明是解决上述问题的有效途径。

我们将在本书中重点讨论软件开发工具,别的软件工具请读者参考有关的书籍即可。

1.2 开发工具的选择

当今开发 Windows 应用程序有众多的工具供选择,这一小节中我们将对各种主要的工具: Visual Basic、Delphi、C++ Builder 和 Visual C++ 做比较。

1.2.1 Visual Basic

Visual Basic 是 Microsoft 的产品,是一个被软件界称之为划时代的革新产品,该软件改变了人们开发 Windows 程序的方式,它采用一种称之为交互式的可视化操作,使得软件开发人员开发 Windows 程序的每一过程都有直观形象的反馈,从而能加速整个开发进程,提高了开发效率。

Visual Basic 的优点如下:

1. Visual Basic 使得 Windows 程序设计人员不再只依赖于那些复杂的 SDK 编程,使得开发 Windows 程序变得容易多了,在所有的编写 Windows 应用程序的工具中,用户学用 Visual Basic 来开发 Windows 应用程序所需的时间是最短的。

2. Visual Basic 已成为 32 位的编程语言。Visual Basic 版本几经发展,目前已经发展到 6.0。在以前的 4.0 版本中,由于完全使用了面向对象的编程概念,而且它同时适用于 Windows 3.1 和 Windows 95 两种版本,因而使得其开发复杂程序的功能得到增强。到了 Visual Basic 5.0,则完全抛弃了 16 位 Windows 3.x 的用户,只能在 32 位 Windows 环境中使用。而到 Visual Basic 6.0 编程效率和运行速度又有了较大的改进。

3. Visual Basic 的运行速度有明显的提高。由于在 Visual Basic 中引入了所谓的本地代码(Native Code)编译器,克服了以往版本由于执行文件采用 P-Code 代码而导致运行速度慢的问题,使得程序执行速度大大加快。根据 Microsoft 的声明,采用本地代码编译后得到的应用程序在某些情况下执行速度较以往提高了 10~20 倍,执行速度可以直逼与采

用 Visual C++ 编写的应用程序的执行速度, 而应用程序开发效率高则是 Visual Basic 的特长, 因此目前 Visual Basic 6.0 非常具有竞争力。

Visual Basic 广泛地用于开发各种 Windows 应用程序, 如界面制作, 多媒体应用。但是, 由于 Visual Basic 天生的特性, 注定了它有以下缺点:

1. Visual Basic 是一种解释型的语言, 运行速度慢, 前面已讨论了此问题, 采用 P-Code 代码虽然执行文件很小, 但是在运行时需要解释执行, 并且, 它的运行必须有对应的 VISUAL-BASICRUN.DLL 和所使用的 VBX 或者 OCX 支持。对于浮点操作密集或者循环嵌套很多的应用来说, Visual Basic 没有采取特别的优化, 因而执行速度远不如用 C/C++ 和 Fortran 开发的应用执行速度快。Visual Basic 5.0 虽然通过引入本地代码编译器大大弥补了这个缺陷, 但是由于其只能运行于 32 位 Windows 环境, 因而在 16 位 Windows 上速度问题仍然得不到解决。虽然目前转向 32 位 Windows 的趋势非常强劲, 但是不容忽视的是由于硬件的限制或者使用习惯等诸多原因, 还有许多用户仍然在 16 位 Windows 上工作。在计算机十分普及的美国, 1996 年使用 16 位 Windows 的用户仍然超过了使用 32 位 Windows 的用户, 任何进行系统软件设计的人员都应该照顾到这些仍然使用 16 位 Windows 的用户。

2. Visual Basic 来源于 Basic 语言, 虽然经过 Microsoft 的不断增强, 但是仍然缺乏非常灵活的数据类型, 因而在开发一些项目必须使用的复杂数据结构方面遇到麻烦, 如链表、图和二叉树等等。由于在中大型项目开发后期, 开发工作不再以界面为主, 而是在算法设计和底层软硬件工作上, 这就使 Visual Basic 开发项目的后期工作量大幅度增加, 为了达到项目要求, 经常需要再转向 C/C++ 开发一些专用的动态链接库来解决问题。

3. Visual Basic 不能灵活地使用系统资源。熟悉 Windows 编程的人都知道, 如果要直接访问硬件或者要编写对系统进行有效访问的应用程序, 没有 Windows API 函数的帮助则非常困难, 但是令 Visual Basic 程序员失望的是, API 函数是用 C 语言和汇编语言实现的, 是为 C 编程准备的, 如果要在 Visual Basic 里面使用这些上千个 API 函数则比较麻烦, 特别是如果设计人员不懂 C 语言则尤其困难。由于 API 函数的复杂性, 而其本身不是为了方便 Visual Basic 编程而提供的, 因此在 Visual Basic 里面调用 API 函数需要一定的技巧, 这些技巧足够用一本很厚的书来表述。Visual Basic 程序员可以从书店里找到很多本类似的书籍。可以说, 任何一个 Visual Basic 程序员发展到一定阶段都需要与众多的 API 函数打交道。另外, 由于 Visual Basic 不支持端口操作, 因此, 如果要编写类似数据采集等需要与硬件交互的程序则需要求助于 C/C++ 语言。

4. Visual Basic 项目分发和管理困难, 其原因如上所述, Visual Basic 应用的运行不能脱离 Visual Basic 的运行库和所使用的控件, 因此, 如果开发人员要将 Visual Basic 应用分发给用户, 那么一定要带上 Visual Basic 的运行库和所使用的控件, 并且要保证正确安装, 这就导致即使一个非常简单应用也需要附带大量其他相关支撑库程序, 对于 Visual Basic 4.0 及更高版本, 由于大量地使用了 OLE 控件 (在 Visual Basic 中称为 OCX), 其安装更为复杂。