



跟我学

VISUAL C++ 6

VISUAL C++ 6 Programming Blue Book

(美) Stephen D. Gilbert 著
Bill McCarty

赵军锁 龚波 李志 等译

 机械工业出版社
China Machine Press

 CORIOLIS

跟我学VISUAL C++6

Stephen D. Gilbert

(美) Bill McCarty 著

赵军锁 龚波 李志 等译



机械工业出版社
China Machine Press

《跟我学VISUAL C++ 6》面向初级和中级用户，是Visual C++ 6的全面而详尽的
导书！

本书的内容主要有：最新Microsoft 工具和组件的使用，Visual C++ Windows 程序
设计与MFC类库编程；图形和多媒体程序设计；精通控件和向导的使用；用软件组件构
造有用的应用程序；使用功能强大的Windows API；使用强大的Microsoft ODBC技术；
使用ActiveX建立数据库应用程序；文档-视图应用程序；创建交互式画图程序；开发关
系数据库应用程序。

掌握一门技术的最佳方式是进行大量的项目实践，本书的特点是新颖独特、基于项
目，使你可以快速精通Visual C++ 6。其结合实际项目并配以详尽解释的独特样式使你
可以快速应用所学知识，随心所欲地开发自己的应用程序。

Stephen D. Gilbert, Bill McCarty: VISUAL C++6 Programming Blue Book.

Authorized translation from the English language edition published by The Coriolis
Group.

Copyright 1999 by The Coriolis Group.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社出版，未经出版者书面许可，本书的任何部分
不得以任何方式复制或抄袭。

版权所有，翻印必究。

本书版权登记号：图字：01-98-2521

图书在版编目(CIP)数据

跟我学VISUAL C++6/(美)吉尔伯特(Gilbert,S. D.)，(美)麦卡蒂(McCarty,B.)著；赵
军锁等译.-北京：机械工业出版社，1999.4

书名原文：VISUAL C++6 Programming Blue Book

ISBN 7-111-07075-5

I.V… II.①吉…②麦…③赵… III.C语言-程序设计 IV.TP312

中国版本图书馆CIP数据核字(1999)第04648号

出版人：马九荣(北京市百万庄大街22号 邮政编码100037)

责任编辑：陈剑瓯

北京昌平第二印刷厂印刷·新华书店北京发行所发行

1999年4月第1版第1次印刷

787mm×1092mm 1/16·25印张

印数：0 001-5 000册

定价：58.00元(附光盘)

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译者序

Visual C++现已成为开发高级Windows 应用程序的主流产品。利用 VC++ 6和MFC 可以开发出强大的32位应用程序，尤其MFC的出现和逐步完善，继承了基本的Windows API，提供给开发者流行的开发技术。Visual C++的出现，使编写高深的Windows程序对于一般用户不再是高不可攀的事情。

《跟我学VISUAL C++ 6》面向初级和中级Visual C++用户，是VC++6的全面而详尽的指导书！作者是资深的编程大师，从使用者的角度来解释、介绍VC++6系统的基本组成、功能和使用方法，并提供许多有益的建议。该书对于那些有一定编程经验的人也有好处。本书以通俗易懂的语言、大量的图片和适当的提示对VisualC++编程的方方面面进行了较为全面的介绍。在翻译本书的过程中，我们也同样受益非浅。总的来说，本书有以下一些特点：

1. 结构清晰，阅读方便。在章节中，还有不少提示和警告信息，对于用户的实际操作很有帮助。本书的内容由浅入深，各个部分的内容既相互联系，又有比较大的独立性。使得用户能够根据自己的实际情况选择不同的章节进行阅读。

2. 覆盖面广。内容包括：最新Microsoft 工具和组件的使用，VC++ Windows 程序设计与MFC类库编程；图形和多媒体程序设计；控件和向导的使用；用软件组件构造有用的应用程序；使用功能强大的Windows API；使用强大的Microsoft ODBC技术；使用ActiveX建立数据库应用程序；文档-视图应用程序；创建交互式画图程序；开发关系数据库应用。并且每个专题都有较全面的论述。

3. 问题针对性强。本书的大部分内容是针对初学者和中级用户的，如果是一位VC++6的新用户，一定会有“相见恨晚”的感觉。如果你原来使用其他版本的Visual C++，你会发觉该书把VisualC++6的主要特色表达得淋漓尽致，让你能从整体上认识Visual C++6。

全书由赵军锁、龚波、李志组织翻译，此外参加翻译的人员还有张巧莉、李林、李士心、孙志刚、陈曙辉、时向泉、张飞、龚小丽、龚泽民、田韞、李琢林、李静、陈彦海、潇东等。此外，前导工作室的全体同仁共同完成了本书的录排、校对等工作。

由于时间仓促，且译者的经验和水平有限，译文难免有不妥之处，恳请读者批评指正。

译者

1998年12月

前 言

我们写本书的目的是为了帮助你掌握Microsoft的Visual C++以及Microsoft基本类库(MFC)应用程序框架。我们并不承诺,读过这本书之后你就会成为一个Visual C++/MFC的顶尖高手。但是这本书对你来说将是一个很好的领路人,至少你可以获得双倍的收益。Visual C++和MFC是非常巨大和复杂的,不可能仅仅靠读一本书就能掌握,这是个事实。唯一的技巧就是花时间,逐步逐步地来学。让我们先从MFC的“如何”(how-to)和“为什么”(what-for)这两个熟知且渐进的问题开始吧!

本书是这样组织的:它将向你展示如何使用Visual C++交互式开发环境(IDE)以及MFC来编程,并且一次前进一步。如果你通过其他的书学习MFC编程曾经失败过,请不要气馁。你可以先从本书的前几章开始!不久,你就将得到一个可以工作的MFC程序——NotePod,可以用它来编辑文本文件。不仅仅这样,你还可以了解到这个NotePod程序是如何工作的。接下来的几章将介绍一些更复杂的项目(Project),包括PaintORama和WordZilla。如果你喜欢从实践中学习,那么这本书正好适合于你。

尽管这是个简单的方法,但这本书并不适合于初学者:我们假定你已经掌握C++语言以及其标准库。如果不是这样,我们强烈建议你先看看Robert Lafore写的“Object-Oriented Programming in C++”(《C++的面向对象程序设计》,The Waite Group Press)。如果你对C++只是有一点点熟练,那么理解示例程序可能会有一点小小的麻烦,这个我们将详细解释。Richard Leinecker写的“The Visual C++ 5 Programmer's Reference”(The Coriolis Group)在C++语法或操作方面会对你有些帮助(此书中译本已由机械工业出版社翻译出版,中文书名《Visual C++5开发人员参考手册》)。

在阅读本书的时候,你应该把每个项目输进去并且运行,只有这样才能理解和巩固学过的东西。你可能会发现,某些章节大部分是面向项目的,而另外某些章节大部分都在解释前面章节中完成的项目。可能的话,可以在完成相关项目章节之后马上读对其解释的章节。这样的话,这些示例项目将会在你心中重新面目一新,帮助你更容易地理解这些解释。

本书重点放在基础上,但MFC基础不包括在内。我们将学习DocView结构、ActiveX控件、FormView、ODBC数据库、网络和Internet。

CD-ROM中包含有示例程序的全部源代码、一些很有用处的MFC实用程序以及最后的一些技术信息。我们希望你能有兴趣读此书,正如我们写本书的兴趣一样。

与原书出版社联系方法:

地址: The Coriolis Group, Inc.

14455 North Hayden Road, Suite 220

Scottsdale, Arizona 85260

电话: 602/483-0192

传真: 602/483-0193

网址: <http://www.coriolis.com>

目 录

译者序		第2章 Windows程序设计	19
前言		2.1 DOS的问题	19
第1章 创建第一个应用程序:		2.1.1 用户界面的问题	19
学习使用VC++	1	2.1.2 硬件相关性问题	20
1.1 审视VC++	1	2.1.3 协作问题	20
1.1.1 Visual C++是什么	2	2.2 Windows解决方案	20
1.1.2 使用Visual C++	2	2.2.1 统一的用户界面	21
1.2 NotePod项目: 初步介绍	3	2.2.2 Windows的消息体系结构	21
1.3 从AppWizard开始	4	2.2.3 设备无关的输入/输出体系结构	21
1.3.1 AppWizard第一步: 指定		2.2.4 进程间的通信和协作	22
应用程序风格	4	2.3 最初的Windows程序设计:	
1.3.2 AppWizard第二步:		API的使用	22
数据库支持	5	2.3.1 事件驱动是指什么	22
1.3.3 AppWizard第三步:		2.3.2 Windows应用程序体系结构	24
OLE支持	6	2.3.3 如何完成	25
1.3.4 AppWizard第四步:		2.3.4 WinMain()函数	26
更进一步的一些东西	6	2.3.5 WndProc()函数内幕	29
1.3.5 AppWizard“中场休息”:		2.3.6 Windows API程序设计小结	30
命名扩展名	7	2.4 Windows程序设计: MFC	30
1.3.6 AppWizard第五步:		2.4.1 何谓MFC	30
最后的一些事情	8	2.4.2 类库与应用程序框架	30
1.3.7 AppWizard第六步:		2.4.3 为什么使用MFC	31
具有视图的应用程序	8	2.4.4 一个MFC小程序	31
1.4 浏览该项目	10	2.5 马上转向MFC	36
1.4.1 丰富的文件: FileView	11	第3章 建立一个简单的基于对话框	
1.4.2 浏览类: ClassView窗口	11	的应用程序	37
1.4.3 ResourceView概览	12	3.1 资源和对话框	37
1.5 运行NotePod项目	13	3.2 启动对话框编辑器	41
1.5.1 创建NotePod	13	3.2.1 控件工具栏	41
1.5.2 运行NotePod	14	3.2.2 开始About对话框	42
1.6 开始新的一切	15	3.3 位图编辑器的使用	43
1.6.1 通过WizardBar简单了解	15	3.3.1 添加新图标	44
1.6.2 意料之外的帮助	17	3.3.2 修改应用程序图标	45
1.7 这究竟是怎么回事	17	3.4 对话框编辑器	48

3.4.1 组件的选取、重定义大小 以及移动	49	5.3.2 手工编写代码: 添加数据成员	85
3.4.2 对话框工具栏介绍	50	5.4 响应BN_CLICKED事件	86
3.4.3 处理静态文本	50	5.4.1 生成OnCancel()函数	86
3.4.4 创建位图资源	54	5.4.2 编写代码: 完成OnCancel() 函数	87
3.4.5 最后一个控件: 组框 (Groupbox)	55	5.5 编写OnDealCards()函数	88
3.5 最后的检查	56	5.5.1 OnDealCards(): 初稿	88
第4章 对话框	58	5.5.2 ClassWizard 和控件变量介绍	89
4.1 FourUp应用程序的结构	58	5.6 编写代码: 发牌、计算输赢	91
4.1.1 所需模块	58	5.6.1 创建 Card变量	91
4.1.2 其他方面	59	5.6.2 创建 Icon变量	91
4.1.3 FourUp家族	60	5.6.3 编写DealCards()成员函数	92
4.2 应用程序对象	61	5.6.4 编写 CalculateWinnings() 成员函数	94
4.2.1 CFourUp.h	62	5.6.5 最后一件事	95
4.2.2 FourUp.cpp	65	5.7 姑且一试	95
4.2.3 关于CWinApp	69	第6章 理解控件	96
4.3 窗口	69	6.1 CWnd的简短说明	96
4.3.1 关于CWnd	69	6.1.1 窗口的构造	96
4.3.2 CDialog类	70	6.1.2 CWnd::Create()函数	97
4.3.3 CAboutDlg浏览	70	6.1.3 窗口风格一览	97
4.3.4 遍历CFourUpDlg	71	6.1.4 创建子窗口	98
4.4 资源介绍	75	6.1.5 有用的CWnd函数	99
4.4.1 资源描述和资源编译器	75	6.2 CStatic再学习	100
4.4.2 图标资源	76	6.2.1 创建CStatic控件	100
4.4.3 关于位图资源	76	6.2.2 CStatic控件的使用	102
4.4.4 对话框资源	77	6.2.3 CStatic成员函数	103
4.5 小结	77	6.3 CButton家族	103
第5章 控件和ClassWizard: 实现你的 对话框	78	6.3.1 创建CButton控件	104
5.1 努力方向	78	6.3.2 CButton控件的使用	108
5.2 对话框编辑器回顾	79	6.4 小结	109
5.2.1 添加牌张	79	第7章 用数字绘图: 建立一个画图 应用程序	110
5.2.2 安排图标	81	7.1 LineOne上的图形	110
5.2.3 添加游戏区域	82	7.1.1 LineOne 项目	110
5.2.4 添加标题区	82	7.1.2 建立项目骨架	110
5.2.5 创建赔率表	82	7.1.3 加入一些线	111
5.2.6 标识按钮	83	7.2 LineOne内部	112
5.3 编写代码	84	7.2.1 OnPaint()是什么	113
5.3.1 事件计划	84		

7.2.2 获取画布	113	9.2.2 与代码建立联系	154
7.2.3 准备环境	114	9.3 预告	157
7.2.4 绘制直线	114	第10章 第二代PaintORama	158
7.3 LineTwo应用程序	115	10.1 PaintORama版本3: 颜色和风格	158
7.3.1 建立LineTwo	115	10.1.1 增加多种颜色	158
7.3.2 LineTwo是如何工作的	116	10.1.2 颜色选择: CColorDialog	158
7.3.3 LineTwo工作的缺陷	117	10.1.3 颜色框绘制	161
7.3.4 用ClassWizard添加一个窗口 消息处理函数	117	10.1.4 单选按钮和画笔风格	162
7.4 Windows的图案绘制	118	10.2 PaintORama: 线和外形	164
7.5 绘图工具	120	10.2.1 小巧而精美的组合框	164
7.5.1 库存画笔和画刷	121	10.2.2 线和外形	166
7.5.2 库存画笔和画刷的使用	121	10.2.3 橡皮圈的秘密	168
7.6 连续绘图	122	10.3 PaintORama: 最终的版本	169
7.7 PaintItGray	124	10.3.1 制作列表	169
7.8 勇往无前	126	10.3.2 编写画刷部分的代码	171
第8章 图形和文本	127	10.3.3 添加画刷预览和颜色支持	173
8.1 Windows和GDI的内幕	127	10.4 仍需改进	174
8.2 CDC家族介绍	129	第11章 建立文档和视图	175
8.3 GDI的画图工具箱	130	11.1 PaintORama: WM_PAINT 的再次处理	175
8.3.1 CPen、颜色和其他CDC属性	131	11.1.1 PaintORama 概览	176
8.3.2 GDI CBrush类	135	11.1.2 创建CMetaFileDC对象	176
8.4 CDC绘画的模式	137	11.1.3 用CMetaFileDC进行记录	177
8.5 创建你自己的屏幕保护程序	138	11.1.4 回放图元文件	180
8.6 下一步	141	11.1.5 小的补充	181
第9章 建立一个交互式的绘画程序	142	11.2 SDIOne: 转向DocView	182
9.1 PaintORama版本1	142	11.2.1 建立第一个SDI应用程序	183
9.1.1 建立PaintORama	143	11.2.2 在SDIOne 中处理文档	184
9.1.2 添加成员变量	144	11.2.3 SDIOne 中的视图处理	187
9.1.3 初始化成员变量	145	11.3 立竿见影	189
9.1.4 添加一个Clear按钮	147	第12章 DocView 环境	191
9.1.5 鼠标消息如何工作	148	12.1 DocView 概述	191
9.1.6 WM_LBUTTONDOWN: 启动画图	148	12.1.1 DocView 程序的类型	191
9.1.7 WM_LBUTTONUP: 画线结束	150	12.1.2 DocView 的优点	192
9.1.8 WM_MOUSEMOVE: 使所有像素排成一行	150	12.2 DocView 的组成	192
9.2 PaintORama版本2: 改进画笔	151	12.2.1 应用程序类	193
9.2.1 Spin控件	152	12.2.2 主窗口类	193
		12.2.3 文档类	194
		12.2.4 视图类	197

12.2.5 权利和义务	198	14.4.2 添加自己的指示器	235
12.3 CSDIOneApp 的InitInstance()	198	14.4.3 添加一个画笔颜色指示器	238
12.3.1 应用程序的建立	200	14.5 下一步的工作	243
12.3.2 连接各模块	202	第15章 保存MiniSketch:操作文档	
12.3.3 外壳	203	和文件	244
12.3.4 命令行	203	15.1 点和图形	244
12.3.5 窗口处理	204	15.1.1 类和属性	245
12.3.6 CSDIOneApp 的消息映射	204	15.1.2 Shape对象的行为	246
12.4 下面学习什么	205	15.2 定义Shape类	246
第13章 MiniSketch应用程序	206	15.2.1 创建类文件	246
13.1 名称中有什么	206	15.2.2 Shape类定义	248
13.2 往MiniSketch应用程序		15.2.3 Line类定义	249
中增加菜单	209	15.2.4 Squiggle 类定义	249
13.2.1 添加菜单标题:		15.2.5 FilledShape类定义	250
Clear Drawing	210	15.2.6. Box和Oval类定义	251
13.2.2 使用菜单编辑器	210	15.3 实现Shape类	252
13.2.3 添加一个菜单项处理过程	212	15.4 使用Shape类	254
13.2.4 幕后: ClassWizard的机制	214	15.5 保存MiniSketch中的数据	256
13.3 处理画笔	215	15.6 序列化	259
13.3.1 添加一个新菜单	215	15.6.1 MFC序列化细节	260
13.3.2 画笔颜色: 添加对话框选项	216	15.6.2 编写Serialize()	260
13.3.3 画笔宽度: 添加级联式菜单	219	15.7 还要查看哪些东西	262
13.4 下一步的工作	222	第16章 新的视野: 翻滚和打印	263
第14章 菜单、工具栏和状态栏	224	16.1 画刷颜色	263
14.1 命令UI	224	16.2 把文档和视图联系起来	265
14.1.1 方法1: 单独的UI处理函数	225	16.2.1 创建新的图形	265
14.1.2 方法2: 多个宏, 一个处理		16.2.2 完成该图形的绘制	266
函数	226	16.2.3 橡皮圈	267
14.1.3 方法3: 使用ON_UPDATE_		16.3 绘制图形	268
COMMAND_UI_RANGE		16.3.1 自我描绘图形	268
来代替	227	16.3.2 处理Update()	270
14.2 加速键	227	16.3.3 重新编写OnDraw()	270
14.3 工具栏	229	16.4 选择视图: 映射模式	271
14.3.1 打开颜色设置框——一个命令		16.4.1 MM_TEXT映射模式	272
按钮式的工具栏按钮	230	16.4.2 设备无关性映射模式	272
14.3.2 添加图形形状类按钮: 单选框		16.4.3 调整MiniSketch 应用程序	
按钮和工具栏按钮	232	中的映射模式	273
14.4 状态栏	235	16.5 翻滚视图	274
14.4.1 移去不期望的指示器	235	16.6 打印和打印预览	276

16.6.1	MFC打印函数	276	19.1	获得一些数据	321
16.6.2	MiniSketch应用程序例子	278	19.2	使用ODBC的非编码记录	322
16.7	谈谈一个崭新的话题	280	19.2.1	步骤1: 创建一个数据源名称	323
第17章	软件重用: 构造一个基于组件 的应用程序	281	19.2.2	步骤2: 使用AppWizard创建一个 ODBC应用程序	325
17.1	考察Gallery	282	19.2.3	步骤3: 创建你的表单	326
17.2	WordZilla 应用程序范例:	286	19.2.4	步骤4: 联系各部分	328
17.3	增强WordZilla应用程序	290	19.3	往OBos 中添加代码	329
17.3.1	安装缺省字体	290	19.3.1	添加一个记录过滤器	330
17.3.2	设置一个新字体	291	19.3.2	添加和删除记录	331
17.3.3	创建字符属性按钮	293	19.4	使用DAO	335
17.3.4	创建段落属性按钮	294	19.4.1	创建一个CDaoRecordView 应用程序	336
17.4	在对话框中应用ActiveX	297	19.4.2	改进DBos应用程序	338
第18章	ActiveX和基于组件的 应用程序	298	19.5	使用OLE DB	341
18.1	给WordZilla添加对话框	299	19.6	下一步: ActiveX和简便之途	342
18.1.1	画一个Select A Date对话框	299	第20章	关系数据库	343
18.1.2	创建对话框类	301	20.1	什么是关系数据?	343
18.1.3	联系该对话框	302	20.1.1	设计图解	343
18.2	模态对话框是如何工作的	304	20.1.2	关键字概念	344
18.2.1	构造一个对话框	305	20.2	使用DAO查看数据库	345
18.2.2	显示一个对话框	306	20.2.1	DBExplore例子	345
18.2.3	对话框的数据传输怎样实现	306	20.2.2	在OnFileOpen()中使用 DAO类	348
18.2.4	关闭对话框	307	20.2.3	探究其他的对象	349
18.2.5	处理OK	307	20.3	关系数据库: SQL	352
18.3	使用非模态对话框	308	20.3.1	SELECT命令	352
18.3.1	非模态对话框如何工作	308	20.3.2	选择域	352
18.3.2	WordZilla中的非模态对话框	308	20.3.3	选择行: 使用Where子句	353
18.4	ActiveX日期捕捉器	313	20.3.4	谓词This	353
18.4.1	添加ActiveX控件	313	20.4	ActiveX数据库控件	355
18.4.2	对话框编辑器中的ActiveX 控件	314	20.5	往DBExplore中添加ActiveX	356
18.4.3	ActiveX控件、代码和 ClassWizard	316	20.5.1	步骤1: 添加ActiveX数据控件	356
18.4.4	属性事件和方法	318	20.5.2	步骤2: 往主表单添加组件	358
18.5	再谈谈ActiveX	318	20.5.3	步骤3: 创建查询结果对话框	360
18.6	接下来的工作: 操作存储空间	320	20.5.4	步骤4: 激活查询按钮	361
第19章	创建数据库查寻和更新 应用程序	321	20.5.5	工作原理的简单说明	363
			20.6	下一步: Web	364
			第21章	网络编程: 浏览器和其他	

客户程序	365	21.4 开发导航工具	376
21.1 HTMLView	365	21.4.1 创建工具栏	376
21.2 提供Browser缺省主页	367	21.4.2. 添加代码	379
21.2.1 主页源代码	367	21.4.3 处理Open 对话框	381
21.2.2 理解HTML语言	370	21.5 使用WinInet类	383
21.2.3 帮助Browser 找到主页	371	21.5.1 WinInet是什么	384
21.3 Visual C++新功能: HTML资源	372	21.5.2 简单的WinInet应用程序	384
21.3.1 引入Browser.html 文件	373	21.5.3 WinInet其他功能	386
21.3.2 引入图形文件	374	21.6 保存自己的Browser	386
21.3.3 使用res: 协议	375	21.7 再见	388

第1章 创建第一个应用程序：学习 使用VC++

在VC++中，要创建一个和Windows NotePad一样的程序，可以不用写一行代码。由此可见，使用它创建自己的编辑器是如何的容易！

这个听起来很熟悉吗？

假设现在是圣诞节——或者是你的生日。蛋糕已经吃过了，贺卡也念完了，你也已经答谢过了，人们已忘记、宽恕或谅解了你应尽的义务。你开始坐下来考虑自己的事情。在房子地板的中间，到处都是包装纸和一些花生，还有你渴望得到的东西——Gizmo-2000。现在我们不知道Gizmo-2000是个什么东西也没关系——也许是早些时候的计算机、一台新的无线电、一台可以骑的剪草机，或者是一个电子显微镜。但更重要的是，它是这个地球上最大的、最棒的、最晚的、最令人敬畏的Gizmo，你一定会迫不及待地想开始干一些事情。

当然，还有一点小小的更复杂的因素。在你的Gizmo-2000的边上，还摆着很多很多你曾经读过的书。最上面的一本就是《The History and Operation of Gizmos:A Depth-First Approach》(Gizmo的发展历史和操作：入门)。

现在，你面临着一个选择：

a) 将Gizmo-2000整齐地放回纸箱里面，然后开始制定在合适时间将自己培养成一个Gizmo 专家的学习计划；

b) 给Gizmo接上电源，然后开始按动某些按钮，看这个宝贝究竟能干些什么？

如果你回答a，那么这本书可能并不适合于你。另一方面，如果你根本就不考虑a就回答b，那么这本书正是你所需要的。

这本书很适合于那些探索者——他们首先希望Gizmo能够动起来，然后再去研究它是如何以及为什么会工作。当然，探索是需要很多时间且有危险的。同18世纪晚期在西部定居的探险者们一样，你需要一个向导——这便是我们所扮演的角色。我们可以帮助你尽快地到达目的地，而不会被熊吃掉！

你准备好了吗？让我们开始吧！

1.1 审视VC++

要把Visual C++运行起来最快捷的方式就是从Start（开始）菜单上选择相应的快捷方式，如图1-1所示。我们注意到，Visual C++出现在Visual Studio 6的文件夹中，我们可以通过主Programs（程序）文件夹来访问（你可能已经作为单独产品购买了Visual C++ 6的拷贝，也可能是作为购买包括C++、Visual Basic以及其他一些开发工具的软件包的一部分获得）。

用来创建Visual C++程序的集成开发环境（IDE）类似于艺术家的工作室：它方便地将你所需要的工具集中到了一起。下面我们来解释一下Visual C++ IDE下的各种工具是如何一起工作的。

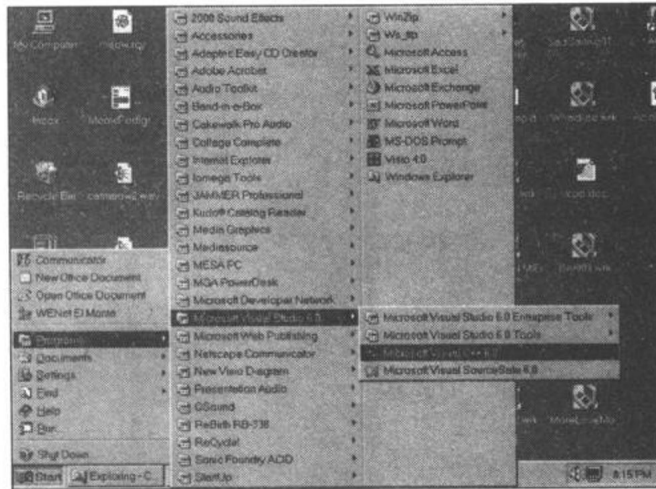


图1-1 从Start菜单上启动Visual C++ 6

1.1.1 Visual C++是什么

Visual C++包括以下这些主要的构件:

- 编辑器 (Editor): 用来输入、浏览以及修改C++源代码。
- 编译器 (Compiler): 用来将C++源代码编译成目标代码。
- 链接程序 (Linker): 用来链接目标代码和库模块以生成可执行文件。
- 库: 它提供一些预先写好的模块, 可以包含进编写的程序中。Microsoft基本类库(MFC)是最重要的库, 可以用它来编写运行在Microsoft Windows下的应用程序。另外, 标准C++库支持输入/输出以及C++语言的其他一些标准特征。

• 其他一些工具: 包括AppWizard、ClassWizard以及Resource Editor。本章将涉及到AppWizard, 后面的一些章节将用到其他一些工具。

1.1.2 使用Visual C++

启动Visual C++后, 可以看到一个五颜六色的屏幕, 该屏幕消失后, Visual C++便看起来有些朴素, 如图1-2所示。这就是无处不在的Tip Wizard对话框。它在Visual C++每次启动时出现。如果你不喜欢每次的自我提高, 可以通过清除Show Tips At Startup复选框来禁止该Tip Wizard。

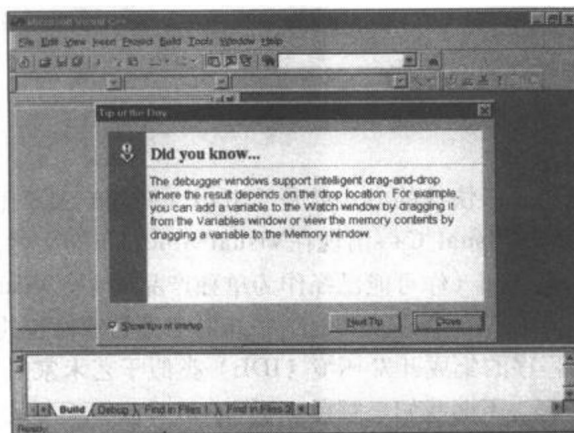


图1-2 显示Tip Wizard的Visual C++启动屏幕

Visual C++屏幕包括三个窗口：

- Project Workspace窗口显示在屏幕的左边，它有助于你管理具有多个源文件的大程序。
- Editor窗口显示在Project Workspace窗口的右边，可以使用Editor窗口输入和修改源代码。
- Output窗口显示在屏幕的下部。它显示一些有关发出的命令的进展信息、错误信息和结果的摘要。例如，如果你试图编译一个包含错误的程序时，将在Output窗口中显示编译错误。

1.2 NotePod项目：初步介绍

在创建项目之前，Visual C++里不会发生任何事情——所以，我们应该先创建一个项目。我们不必去创建一个毫无用处的示例，相反，我们可以创建一个能干某些有用事情的Visual C++程序。我们也不必去谈论老掉牙的“Hello World”程序。我们所要做的是一个真真正正的文本编辑器，它包含有菜单和工具栏——甚至还能处理打印和打印预览。我们的项目很类似于Windows的NotePad程序，只是相对而言更好一些。为了使得人们不与NotePad这个程序混淆，我们把这个程序取名叫NotePod（毕竟，从一般人的观点来看，这只是在原有的基础之上做了一些改进而已）。

要在Visual C++中创建一个新的项目，可以从主菜单上选择File/New菜单项来完成。在接下来弹出的对话框中，选择Projects选项卡，并且从列表框中选择项目的类型。

但是，Visual C++提供了很多不同种类的项目类型可供选择。更糟糕的是，这些不同的类型只是简单地依字母顺序进行了排列，这就使得选择正确的类型就如同在一个繁杂而不熟悉的菜单中挑选所需的菜一样——除非你对这家餐馆非常熟悉，否则就只能碰运气了！

同时，比较幸运的是，你并没有处在这种困境当中——你有一个向导，记住了吗？如图1-3所示，搜寻列表直到发现“MFC AppWizard(exe)”为止，并选择该选项。

除了选择项目类型之外，New对话框还要询问你项目的名称以及项目所存放的位置。请在Project Name 文本域处输入项目的名字（NotePod）。输入的名字会自动被添加到Project Name 文本域下面的Location文本区域中。

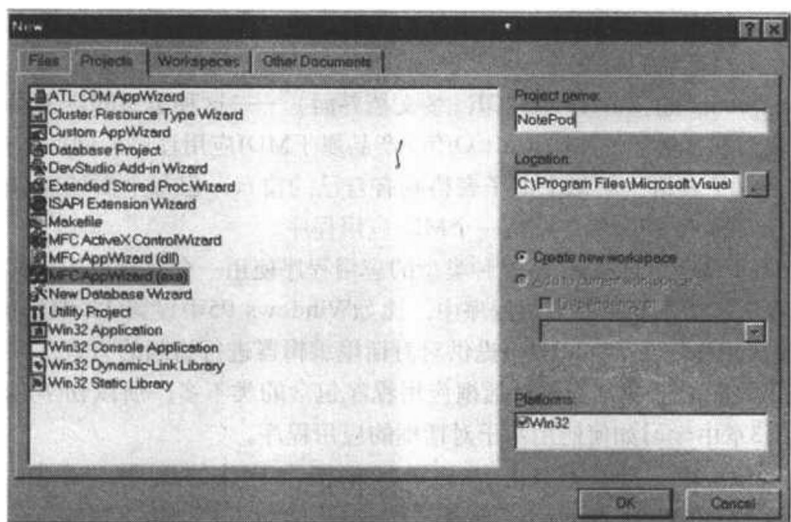


图1-3 开始一个AppWizard项目

输入项目的名字有多个目的。首先，在项目编译之后，该名字就是可执行文件的名字。其次，该名字还作为自动创建项目文件时存放项目文件子目录（在Visual C++的说法中称作workspace(工作区)）的名字。第三，Visual C++创建项目的时候，会使用项目名字来构造类的名字。

缺省情况下，创建新项目的时候，Visual C++将创建一个新的工作区。假若你愿意，你还可以在Location文本区域处输入一个不同的位置。对于NotePod项目来说，我们在Project Name 文本域处输入“NotePod”，并且对其他的选项使用其缺省值。完成时，单击OK按钮，后面的工作就由Visual C++ AppWizard来完成了。

1.3 从AppWizard开始

Visual C++的AppWizard实质上是一个代码生成器——你可以把它想象成为你的个人编程助手。AppWizard会给出一系列对话框，并询问一些有关你要创建应用程序的特征。获得这些必要的信息之后，AppWizard将产生一个定制的应用程序骨架。这个骨架程序具备你所指定的一些特征，你还可以更进一步地定制该应用程序，以使得它具备某些与其他Windows应用程序相区别的特征。然而，你也可以根本就不去管它。

请记住有关AppWizard的重要的一点：只有在项目开始的时候才能使用它。可以使用AppWizard创建一个应用程序的原型，但却不能使用它来维护或扩展应用程序。正因为这样，所以在开始应用程序之前，必须先计划好你希望应用程序来干什么事情。在完成之后，就无法再去运行AppWizard来增加打印预览或数据库支持的功能。

记住了这些之后，现在我们就来开始使用AppWizard构造NotePod程序。

1.3.1 AppWizard第一步：指定应用程序风格

如图1-4所示，AppWizard首先询问需要创建应用程序的类型。有以下这些选项：

- Single Document Interface(SDI, 单文档界面)——这种类型的应用程序一次只允许打开一个文档。文档自动充满应用程序的主窗口，不为其他的文档留下空间。Windows的NotePad程序是一个SDI应用程序。当从其File菜单打开一个文本文件时，新的文件代替了原来查看的文件。第11章我们将更进一步地讲解SDI。

- Multiple Document Interface(MDI, 多文档界面)——这种类型的应用程序允许同时打开多个文档。我们非常熟悉的Microsoft Office产品属于MDI应用程序。例如，在Excel中，可以同时打开多个电子表格——每个电子表格具有自己的窗口，但每个文档窗口又都在主应用程序窗口里。我们的NotePod程序就是一个MDI应用程序。

- Dialog based(基于对话框)——这种类型的应用程序使用一个对话框作为其主窗口。基于对话框的应用程序常用于简单的应用程序中，比如Windows 95中设置Date/Time Properties(日期/时间属性)的应用程序。Visual C++提供对对话框编辑器进行访问的方式，以使得可以设计基于对话框应用程序的外观。基于对话框应用程序包含的类不多，所以初学起来比较容易理解。我们将在第3章中学习如何使用基于对话框的应用程序。

在AppWizard第一步的对话框中还可指定两个特殊的选项。首先，可以不选中Document/View Architecture Support复选框——这是Visual C++ 6所支持的一个新的选项——这样可以使得AppWizard对某些应用程序而言只产生简单的代码。大多数情况下，可以让它保

持为缺省（选中）状态。在第12章中将介绍有关Document/View architecture更多的内容。其次，还可以采用英语以外的语言来保存资源，比如菜单、系统提示等，这个只需要简单地从下拉列表框中选择就可以了。

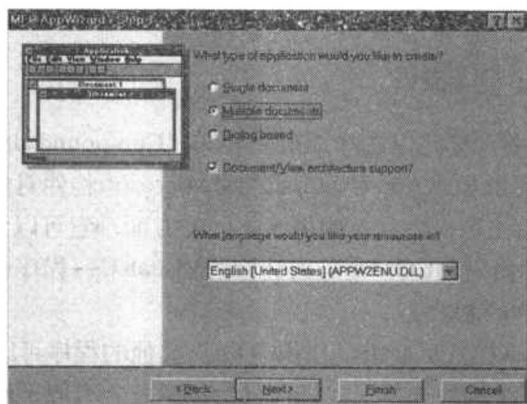


图1-4 AppWizard第一步：指定应用程序风格

对于NotePod程序，我们选择MDI应用程序、将Document/View复选框保持为选中状态，并将资源DLL设置为英语。完成了这些之后，单击Next进入第2步。

1.3.2 AppWizard第二步：数据库支持

AppWizard的下一步询问需要支持数据库的类型。AppWizard将自动书写代码以通过Open Database Connectivity (ODBC, 开放数据库互连) 标准访问数据库，并可支持上百种不同的数据库。AppWizard使得还可通过Microsoft 在Access和Visual Basic中使用的数据库引擎Jet Engine来访问数据库。在Visual C++ 6 中，还可通过ActiveX Data Objects(ADO, ActiveX数据对象)来访问Object Linking and Embedding(OLE, 对象链接与嵌入) DB的提供者（在第19和20章中我们将更进一步地讲解）。

当然，在图1-5中，我们可以看到由AppWizard提供的选项并没有ODBC、ADO、OLE DB。相反，却是含糊不清的对数据库的4个级别支持的选择。毕竟，如何能够了解你是需要“Header files only”还是“Database view with file support”呢？

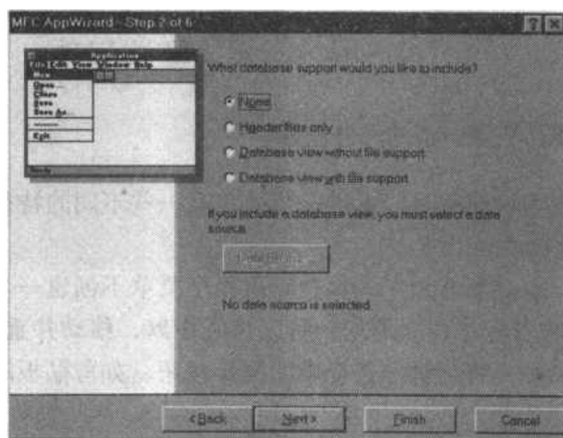


图1-5 AppWizard第二步：指定数据库支持选项

幸运的是，对于我们现在来说，答案是相当简单的。NotePod应用程序根本就不需要任何的数据库支持，所以我们还是到第19章再来讨论这些选项吧（在那一章中，我们将使用Visual C++数据库支持来创建一个数据库查询和更新的应用程序）。在这里，我们选择None并单击Next继续下一步。

1.3.3 AppWizard第三步：OLE支持

在AppWizard的第三步，你可以选择你所需要的“Compound document support”（复合文档支持）。在该屏幕中，可以要求AppWizard添加对Microsoft构件对象模型（COM）的支持。

COM是一个非常大的主题，它涉及到很多不同的方面，它可以用一本书（或者是10本书）来讲述。在第17章中，你将看到COM是如何让你的Visual C++程序可以使用Visual Basic 程序所使用的同种类型的ActiveX控件。

然而，COM比ActiveX控件更高级。使用COM，其他的程序可以自动控制你的Visual C++程序，而你的程序就像一个构件一样。或许你已经注意到，类似于Microsoft Word的程序允许在文档中嵌入一个由其他应用程序创建的电子表格或一幅图形：这就是COM作用的结果。图1-6显示的是在Visual C++中有效的COM选项。

我们在NotePod程序中不需要使用任何的复合文档，所以应该清除AppWizard第三步对话框底部的ActiveX Controls复选框。然后，单击Next以继续。

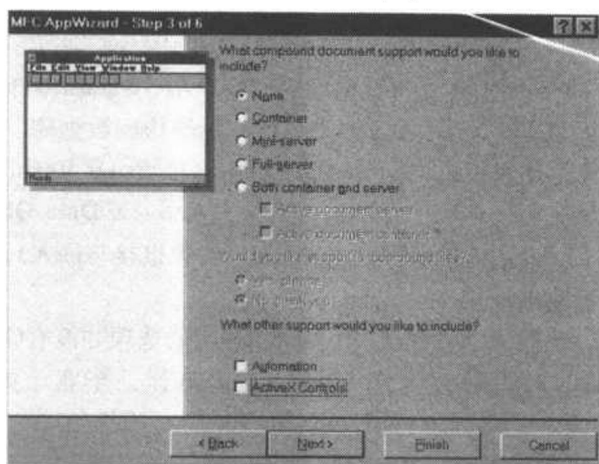


图1-6 AppWizard第三步：设置COM选项

1.3.4 AppWizard第四步：更进一步的一些东西

在第四步中，AppWizard使得可以更进一步地定义一些不同的特征。这些选项的其中四个在缺省状态下是选中的，如下：

- Docking Toolbar——告诉AppWizard在应用程序菜单下创建一个标准的工具栏。该工具栏是可浮动的，这就意味着应用程序的用户可以将它拖离、移动并重新附在屏幕的任一边上。AppWizard在创建工具栏时，将提供一些标准的操作按钮，如剪贴板的复制、粘贴，文件的打开和保存。

- Initial Status Bar——在应用程序窗口底部设置一个标准的Windows状态栏。激活状态栏