



精通

Visual C++ 6.0

黄庆生 汤毅 戴宁 编

人民邮电出版社

# 精通 Visual C++6.0

黄庆生 汤毅 戴宁 编

人民邮电出版社

## 内 容 提 要

Microsoft Visual C++ 作为 Microsoft Visual Studio 的重要组成部分, 包含了迄今为止功能最为强大的基于 Windows 的应用框架, 在同类产品中处于领先地位。利用它所集成的一系列强大功能, 用户就能够开发出实用高效的应用程序。

为了让广大的软件开发人员和编程爱好者能够早日了解并掌握 Microsoft Visual C++ 6.0 这一最新开发工具, 我们根据自己多年的实际编程经验和对 VC6.0 的使用心得, 编写了本书。对于 Windows 操作系统的消息循环的概念和 Windows 程序框架; VC++ 的编程环境; VC++ 所特有的编程工具以及可视化编程的优点等全书共分 12 章进行了讲述。

本书面向的对象主要是初级和中级的读者, 是学习 Visual C++ 6.0 的一本较好的参考书。

---

### 精通 Visual C++ 6.0

---

- ◆ 编 黄庆生 汤毅 戴宁  
责任编辑 须春美
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
北京顺义振华印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16  
印张: 28.25  
字数: 701 千字  
印数: 6 001 - 10 000 册
- 1999 年 6 月第 1 版  
2000 年 3 月北京第 2 次印刷

ISBN 7-115-07908-0/TP·1171

---

定价: 42.00 元

## 编者的话

进入 90 年代以来, 计算机可视化技术得到了迅猛发展并引起了人们的广泛重视, 越来越多的计算机专业人员和非专业人员开始研究和应用可视化技术。

Microsoft Visual Studio 是美国微软公司推出的目前使用最为广泛的可视化编程环境。它的出现, 改变了传统的编程手段, 使得程序员面对的不再是枯燥、单调的字符界面, 取而代之的是形象生动且极具可视性的开发环境。

Microsoft Visual C++ 作为 Microsoft Visual Studio 的重要组成部分, 包含了迄今为止功能最为强大的基于 Windows 的应用框架, 在同类产品中处于领先地位。利用它所集成的一系列强大功能, 用户就能够开发出实用高效的应用程序。有数据表明, 在用 Microsoft C/C++ 编写基于 Windows 平台的应用程序的开发人员中, 有百分之六十的人使用的都是 Microsoft Visual C++。

目前, Microsoft Visual C++ 的最高版本是 6.0。从 1.0 版本发展到今天, 它的每一次升级都是对软件开发的一次推动。随着 Internet 在全球的普及, 在 Visual C++ 6.0 版本中不仅包括了以前版本中各种功能, 更集成了全面强大的网络开发能力, 从而使用户也能方便地开发出网络程序。可以说, Visual C++ 6.0 使 C++ 语言的发展达到了一个全新的高度。

为了让广大的软件开发人员和编程爱好者能够早日了解并掌握 Microsoft Visual C++ 6.0 这一最新开发工具, 我们根据自己多年的实际编程经验和对 VC6.0 的使用心得, 编写了本书。

本书面向的对象主要是初级和中级的朋友们。我们假设了读者是下列几种情况之一:

1. 如果你是一位 Borland C&C++ 的程序员, 而今天成为了 VC 6.0 的使用者, 你注意的不应该是 C++ 语言的细节问题, 而是要重点了解 VC++ 相对于 Borland C++ 在编程的方式上的不同点, 认识基于 MFC 的编程优于基于 API 函数的编程的地方; 认识 AppWizard (第一章), ClassWizard (第六章), Spy++ (第十二章) 等 VC++ 所特有的编程工具, 以及这种可视化编程的优点。

2. 如果你是一位 UNIX 系统的 C 语言程序员, 和 Borland C++ 的程序员一样, Borland C&C++ 语言本身对你们不是问题, 但是在直接进入角色之前, 还是应该先了解一下 Windows 操作系统的消息循环的概念和 Windows 程序框架 (第二、三、四章)。

3. 如果你是一位 VB 和 Delphi 的程序员, 那么一定具有基本的面向对象和可视化编程的概念, 但是可能对“类”还没有什么概念。所以我们建议你首先应该看看附录 B, 它讲述了“类”的一些基本概念和编程。如果你对 C 语言语法本身还不太熟悉, 那么你可以边学 VC++ 的基本概念, 边学 C 语言语法 (有了高级语言的基础, 这并不难)。本书在一些编程细节上, 也对 VC++ 和 VB 在编程方式上作了对比。

4. 学过 C 但是没有学过 C++ 的朋友很多, 你们的首要任务还是建立面向对象的编程概念, 也就是要建立“类”的概念, 请首先参看附录 B, 然后再进入 VC++ 的学习。

5. 学过一门高级语言的朋友, C 语言语法并不是你们的障碍, 你们只需花两三个星

期熟悉语法，再开始 C++ 的学习。VC++ 的难点还是在于对“类”这一概念的理解。

本书由黄庆生策划审校，汤毅、戴宁主编，参与本书编写和录排工作的人员还有：王俊峰、舒力、谢婷、黄颖硕、田友锋、陈果、赵仪、崔晶炜、罗玘、孙景华、黄瀚华、黄少棠、孙宝玉、黄昌明、王洪秀、马占荣、孙景利、田尊五、吴广志、刘世德、汪春军、张志明、王国戟、朱黎、李剑、董团结、顾云飞、刘贤轶、尹之恒、蒋伟峰、张旭等，在此一并表示感谢。

值得一提的是，在本书的编撰过程中，还得到了清华大学吴振华博士的指导与帮助，在此我们也向他致以深切的谢意。

鉴于时间仓促，加之编者水平有限，书中难免存在不妥之处，敬请广大读者批评指正。

编 者  
1999 年 5 月

# 目 录

序篇 Visual C++ 6.0 简介 .....	1
1. Visual C++ 6.0 新特性 .....	2
1.1 对编译器的改进 .....	2
1.2 对 AppWizard 的改进 .....	2
1.3 更完善的 MFC .....	3
1.4 数据库应用程序的增强支持 .....	3
1.5 相对于 Visual C++ 5.0 的新特性 .....	3
2. Visual C++ 6.0 开发环境简介 .....	4
2.1 Microsoft Developer Studio .....	4
2.2 Visual C++ 6.0 编程工具 .....	4
2.3 OLE 服务器 .....	5
2.4 MFC 类库 .....	5
2.5 Visual C++ 6.0 的帮助系统 .....	5
2.6 Visual C++ 6.0 附带工具简介 .....	5
3. 如何学习使用 Visual C++ 6.0 .....	5
第一章 建立一个 AppWizard 程序 .....	7
1.1 什么是 AppWizard? .....	7
1.2 迎接你的第一个 AppWizard 程序 .....	8
1.2.1 第一步 .....	9
1.2.2 第二步 .....	10
1.2.3 第三步 .....	11
1.2.4 第四步 .....	13
1.2.5 第五步 .....	14
1.2.6 第六步 .....	14
1.3 “I am a programmer.” 在哪儿? .....	17
第二章 学习简化的程序框架 .....	20
2.1 一个简化过的程序框架 .....	20
2.2 WinMain( ): 第一个动作 .....	22
2.3 登记窗口类 .....	24
2.4 创建一个窗口 .....	26
2.5 显示窗口 .....	28
2.6 显示消息 .....	28

2.6.1	改写源程序.....	28
2.6.2	编译、链接并运行该程序.....	32
2.6.3	与 DOS 或 UNIX 等价程序的比较.....	34
2.7	窗口类与窗口对象.....	35
2.7.1	窗口对象.....	35
2.7.2	CWnd 派生的窗口类.....	35
2.7.3	注册窗口类.....	36
2.7.4	窗口的创建过程.....	36
2.7.5	销毁窗口.....	36
<b>第三章</b>	<b>消息循环.....</b>	<b>37</b>
3.1	处理消息循环.....	37
3.1.1	程序设计中的基础事件.....	38
3.1.2	“I am a programmer” 的消息循环.....	39
3.2	对事件做出响应: WindowFun( ).....	41
3.3	响应不同的消息.....	41
3.3.1	用 WM_PAINT 消息跟踪屏幕.....	42
3.3.2	WM_DESTROY 消息.....	46
3.4	设备界面进行交互.....	46
3.4.1	HDC.....	47
3.4.2	其它可返回设备上下文的函数.....	53
<b>第四章</b>	<b>了解真正的程序框架.....</b>	<b>54</b>
4.1	WinMain( )函数.....	54
4.2	应用程序框架和源文件.....	59
4.2.1	文档类.....	59
4.2.2	视图类.....	64
4.2.3	主边框窗口类.....	69
4.2.4	子边框窗口类.....	74
4.2.5	应用程序类.....	77
4.3	工具条、状态条和打印等选项.....	97
4.3.1	工具条和状态条.....	97
4.3.2	打印选项.....	98
4.3.3	加入 MDI 支持.....	100
4.4	程序的控制流程.....	101
4.4.1	成员函数 InitInstance.....	102
4.4.2	成员函数 Run.....	104
4.4.3	成员函数 ExitInstance.....	104
4.4.4	成员函数 OnIdle.....	104

4.4.5 CWinApp 的专门服务 .....	104
<b>第五章 熟悉 Visual C++ 6.0 的编程环境</b> .....	<b>106</b>
5.1 Visual C++ 6.0 主窗口 .....	106
5.2 Visual C++ 6.0 工具栏 .....	107
5.3 Visual C++ 6.0 菜单栏 .....	108
5.3.1 “File” 菜单 .....	108
5.3.2 “Edit” 菜单 .....	114
5.3.3 “View” 菜单 .....	117
5.3.4 “Insert” 菜单 .....	120
5.3.5 “Project” 菜单 .....	121
5.3.6 “Build” 菜单 .....	123
5.3.7 “Debug” 菜单 .....	126
5.3.8 “Tools” 菜单 .....	128
5.3.9 “Window” 菜单 .....	131
5.3.10 “Help” 菜单 .....	132
5.4 项目与项目工作区 .....	133
5.5 资源与资源编辑器 .....	135
5.5.1 资源编辑器 .....	136
5.5.2 资源符号 .....	137
5.5.3 资源符号浏览器 .....	138
5.5.4 对话编辑器 .....	138
5.5.5 菜单编辑器 .....	142
5.5.6 加速键编辑器 .....	145
5.5.7 串编辑器 .....	147
5.5.8 版本信息编辑器 .....	148
5.5.9 图形编辑器 .....	149
5.5.10 工具栏编辑器 .....	152
<b>第六章 ClassWizard 编程</b> .....	<b>154</b>
6.1 使用 ClassWizard 添加消息处理函数 .....	154
6.1.1 消息映射 (Message Maps) 选项卡 .....	154
6.1.2 成员变量 (Member Variables) 选项卡 .....	156
6.1.3 自动化 (Automation) 选项卡 .....	157
6.1.4 ActiveX 事件 (ActiveX Events) 选项卡 .....	159
6.1.5 类信息 (Class Info) 选项卡 .....	159
6.2 ClassWizard 功能介绍 .....	161
6.3 传送鼠标消息 .....	164
6.3.1 Windows 系统最重要的事件: 鼠标的移动 .....	165



6.3.2	用鼠标绘图 .....	171
6.3.3	鼠标绘图: 改进型 .....	177
6.4	保存鼠标绘图的信息 .....	183
6.4.1	保存屏幕 MyProg2d .....	183
6.4.2	修改绘图程序: MyProg2d .....	193
6.4.3	测试和评价 .....	200
6.4.4	临近尾声 .....	201
<b>第七章</b>	<b>视图与文档 .....</b>	<b>210</b>
7.1	Document-View 模式 .....	210
7.1.1	为何要分成两类? .....	210
7.1.2	Documet-View 模式的 MFC 版本 .....	211
7.2	从视图中分离出文档 .....	211
7.2.1	文档 .....	211
7.2.2	视图 .....	218
7.2.3	运行程序 .....	225
7.3	保存文档 .....	225
7.3.1	将 Serialize( ) 补充完整 .....	226
7.3.2	检验结果 .....	229
7.3.3	注册一个文件类型 .....	230
7.3.4	关于 “no update” .....	230
7.4	再访 MyProg2.cpp .....	231
7.4.1	什么是 .INI 文件 .....	236
7.4.2	下一步怎么办? .....	237
<b>第八章</b>	<b>图形设备接口 .....</b>	<b>239</b>
8.1	设备环境类 .....	239
8.1.1	显示设备环境类 CClientDC 和 CWindowDC .....	240
8.1.2	构造和析构 CDC 对象 .....	240
8.1.3	设备环境的状态 .....	241
8.1.4	CPaint DC 类 .....	241
8.2	GDI 对象 .....	242
8.2.1	GDI 对象的构造与析构 .....	242
8.2.2	跟踪 GDI 对象 .....	242
8.2.3	库存的 GDI 对象 .....	243
8.2.4	GDI 选择的有效期 .....	244
8.2.5	持久的用于显示的设备环境——登记窗口类 .....	245
8.3	Windows 的颜色映射 .....	245
8.3.1	标准视频图形矩阵 (VGA) 显卡 .....	246

8.3.2	256 色显示卡.....	247
8.3.3	24 位颜色显示卡.....	247
8.4	映射方式.....	248
8.4.1	MM_TEXT 映射方式.....	248
8.4.2	“固定比例”的映射方式.....	249
8.4.3	坐标变换.....	250
8.5	字体.....	252
8.5.1	字体是 GDI 对象.....	252
8.5.2	选择字体.....	252
8.5.3	打印字体.....	252
8.5.4	显示字体.....	252
8.5.5	窗口中的点——逻辑 Twips.....	253
8.5.6	计算字符高度.....	253
8.6	MYPROG3 例程序.....	254
8.7	MyProg3B 程序.....	262
8.8	MYPROG3C 例程序：使用 CScrollView.....	266
8.8.1	MYPROG3C 程序的一些元素.....	270
8.8.2	CScrollView 的 SetScaleToFitSize 状态.....	271
<b>第九章</b>	<b>对象连接与嵌入 (OLE) 及其自动化.....</b>	<b>272</b>
9.1	公共对象模式 (COM).....	272
9.1.1	COM 解决的问题.....	272
9.1.2	COM 的实质.....	273
9.1.3	什么是 COM 接口.....	273
9.1.4	IUnknown 接口和 QueryInterface 成员函数.....	278
9.1.5	引用计数：AddRef 和 Release 函数.....	281
9.2	类厂 (class factory).....	281
9.2.1	COM 和 MFC——CCmdTarget 类.....	283
9.2.2	OLE 和窗口登记数据库.....	284
9.2.3	运行时刻对象登记.....	285
9.2.4	COM 客户程序是如何调用 DLL 侍者程序的.....	285
9.2.5	COM 客户程序是如何调用 EXE 侍者程序的.....	287
9.2.6	MFC 和 OLE.....	289
9.2.7	包容和继承.....	289
9.3	OLE 自动化.....	289
9.3.1	C++ 与 Visual Basic for Application (VBA) 之间的连接.....	289
9.3.2	自动化控制程序和自动化侍者程序.....	290
9.3.3	属性 (property)、方法 (method) 和集合 (collection).....	290
9.3.4	OLE 自动化所解决的问题.....	291

9.4	IDispatch 接口 .....	291
9.4.1	OLE 自动化程序设计 .....	292
9.4.2	MFC IDispatch 的实现 .....	293
9.4.3	一个 MFC OLE 自动化侍者 .....	293
9.4.4	一个 MFC OLE 自动化控制程序 .....	294
9.4.5	VARIANT 类型 .....	296
9.4.6	CVariant 类 .....	298
9.4.7	InvoDce 的参数和返回值的类型转换 .....	301
<b>第十章</b>	<b>动态连接库(DLLs).....</b>	<b>303</b>
10.1	为什么使用 DLL .....	303
10.2	传统的 DLL .....	303
10.3	MFC 库 DLL .....	304
10.3.1	MFC 库 DLL 的使用仅限于 Microsoft C++ 编译器 .....	304
10.3.2	作为 DLL 的 MFC 库类 .....	305
10.3.3	在应用中使用 MFC250D.DLL .....	305
10.3.4	MFC 库扩展 DLL .....	306
10.3.5	MFC 库 DLL 内存使用 .....	306
10.3.6	扩展 DLL 需要的一些代码 .....	306
10.3.7	搜索资源 .....	307
10.3.8	内联构造函数 .....	308
10.3.9	扩展 DLL 的引出 .....	308
10.3.10	类静态数据成员 .....	309
10.3.11	扩展 DLL 运行时刻类的确定 .....	310
10.3.12	创建 DLL .....	310
10.3.13	创建引入库 .....	310
10.3.14	DLL 的调试版和发行版 .....	310
10.4	MyProg4A: 编写自己的类库扩展 DLL .....	311
10.4.1	MyProg4AD.DEF 文件 .....	311
10.4.2	MyProg4AD 工程的 Visual 工作平台选项 .....	313
10.4.3	创建引入库并拷贝 DLL .....	314
10.5	MyProg4B: 使用 MFC 库扩展 DLL .....	314
10.5.1	CMyProg4bApp .....	315
10.5.2	CMyProg4bDoc .....	315
10.5.3	CMyProg4bView .....	315
10.5.4	CStudentDialog .....	317
10.6	创建并测试 MyProg4B 程序 .....	317
10.7	资源访问 .....	318
<b>第十一章</b>	<b>对话框 .....</b>	<b>320</b>

11.1	在状态条上显示对话控件的帮助信息.....	320
11.1.1	问题的提出.....	320
11.1.2	实现技术.....	320
11.1.3	实现步骤.....	320
11.1.4	实现内幕.....	324
11.2	利用 FileOpen 通用对话框打开多个文件.....	325
11.2.1	问题的提出.....	325
11.2.2	实现技术.....	325
11.2.3	实现步骤.....	325
11.2.4	实现内幕.....	329
11.3	定制通用文件对话框.....	331
11.3.1	问题的提出.....	331
11.3.2	实现技术.....	331
11.3.3	实现步骤.....	332
11.3.4	实现内幕.....	336
11.4	扩展和缩减一个对话框.....	337
11.4.1	问题的提出.....	337
11.4.2	实现技术.....	337
11.4.3	实现步骤.....	338
11.4.4	实现内幕.....	341
11.5	显示一个模式或无模式对话框.....	342
11.5.1	问题的提出.....	342
11.5.2	实现技术.....	342
11.5.3	实现步骤.....	343
11.5.4	实现内幕.....	349
11.6	编写定制的 DDX/DDV 例程.....	350
11.6.1	问题的提出.....	350
11.6.2	实现技术.....	350
11.6.3	实现步骤.....	351
11.6.4	实现内幕.....	356
<b>第十二章</b>	<b>剖析工具 Spy++.....</b>	<b>359</b>
12.1	窗体.....	360
12.1.1	查看窗体.....	360
12.1.2	查看窗体详细信息.....	360
12.1.3	定位窗体.....	362
12.2	消息.....	363
12.2.1	“Windows”选项卡.....	364
12.2.2	“Messages”选项卡.....	364

12.2.3	“Output”选项卡 .....	365
12.3	进程与线程 .....	367
附录 A	Visual C++ 6.0 的安装说明 .....	370
附录 B	走进 C++ 的世界 .....	378
B.1	类和对象的简介 .....	378
B.1.1	C 中用户自定义类型 .....	378
B.1.2	转向 C++ .....	379
B.1.3	构造函数 .....	380
B.1.4	析构函数 .....	382
B.1.5	其它的成员函数 .....	384
B.1.6	private 和 public 型类成员 .....	384
B.1.7	帮助函数 .....	385
B.1.8	对 C++ 封装的简要说明 .....	386
B.2	继承和多态性：一个具体的例子 .....	386
B.2.1	轨道物体的基类和虚函数 .....	386
B.2.2	纯虚函数 .....	388
B.2.3	派生类 .....	388
B.2.4	在基类中所调用的虚函数 .....	390
B.3	内嵌对象 .....	390
B.3.1	拷贝构造函数 .....	390
B.3.2	赋值运算符 .....	391
B.3.3	C++ 引用：提高效率 .....	393
B.3.4	内嵌对象创建的总结 .....	396
B.3.5	内嵌对象的析构 .....	397
B.4	在堆中申请对象 .....	397
B.4.1	C++ 的 new 和 delete 运算符 .....	397
B.4.2	通过指针来引用对象 .....	398
B.4.3	虚析构函数 .....	399
B.5	全程对象的申请 .....	400
B.6	对象之间的相互关系：指针数据成员 .....	401
B.7	this 指针的使用 .....	402
B.8	对指针的引用 .....	403
B.9	友元类和友元函数 .....	404
B.9.1	友元类 .....	404
B.9.2	全程友元函数 .....	405
B.10	静态类成员 .....	406
B.10.1	静态数据成员 .....	406

B.10.2	枚举类型——实现静态数据成员的捷径.....	407
B.10.3	静态成员函数.....	407
B.11	重载运算符.....	409
B.11.1	成员函数运算符.....	409
B.11.2	转换运算符.....	411
B.11.3	辅助运算符 ( helper operator ) .....	412
B.12	从代码中分离出类定义.....	413
附录 C	匈牙利表示法.....	416
附录 D	代码调试.....	418
D.1	TRACE.....	418
D.1.1	几种模式.....	418
D.1.2	使用 TRACE.....	419
D.2	调试框架.....	420
D.2.1	AssertValid( ).....	423
D.2.2	防御型的编程方式.....	424
D.2.3	关于 Dump( ).....	425
D.3	自我诊断.....	425
D.4	调试代码的作用.....	432
D.5	用 Dump( )显示对象的信息.....	433
D.6	检查内存.....	435

# 序篇 Visual C++ 6.0 简介

Visual C++ 6.0 是 Microsoft 公司提供的基于 Windows 95、Windows 98 和 Windows NT 的编程工具。如果想要深入掌握其编程机制，从而成为一个出色的 Visual C++ 程序员，就必须对 32 位 Windows 操作系统运行机制有深入的了解。这一点对于用好 Visual C++ 6.0，以及开发出高质量的 32 位 Windows 应用程序都是至关重要的。

在这里，让我们先对 Visual C++ 6.0 的新特性作一些简单介绍，以便读者能对其新增加的强大功能有所了解。

Visual，其英文含义是“可视的”，这里是指在用 Visual C++ 进行开发的过程中对应用程序界面（GUI）的开发，大多使用开发工具所提供的现成的组件，利用所见即所得的方式完成程序界面的设计（亦即通常所说可视化编程），因此大大减轻了程序设计人员的劳动强度，也提高了开发效率。

Visual C++ 最大特色是对面向对象技术的支持，它利用类把大部分与用户界面设计有关的 Windows API 函数封装起来，通过 MFC（Microsoft Foundation Class）类库的方式提供给开发人员使用，极大地提高了软件的重用性。

另外，Visual C++ 内带的 AppWizard 可以帮助 MFC 类库的用户自动生成一个运行程序框架，即一个空的不能做任何事情的应用程序，而用户只需要对该应用程序框架进行扩充添加代码就可以得到一个结构优良的应用程序。

Visual C++ 提供的 ClassWizard 可以为用户提供方便而有效的使用管理 MFC 类库，它允许用户对 AppWizard 应用程序产生的类进行加工、编辑、甚至扩充。

Visual C++ 作为一个功能强大的软件开发工具，它对应用程序的开发支持是多方面的，它可以为以下程序开发提供支持：

- 利用 MFC 的 32 位 Windows 应用程序
- 使用 C 语言 SDK 编写 Windows 程序
- 编写控制台程序
- 编写 ActiveX 控件
- 编写动态链接库(DLLS)
- 开发数据库应用程序
- 开发 Internet 应用程序
- 开发多媒体应用程序

简而言之，Visual C++ 的优点是：

- 与操作系统配合紧密，不存在与操作系统不兼容的问题。
- 功能强大，用途广泛，不仅可以编写普通的应用程序，还能很好地进行系统软件及通信软件的开发。

■操作简便，熟练以后，开发效率将成倍提高。

## 1. Visual C++ 6.0 新特性

与以往的 Visual C++ 的各种版本相比较，Visual C++ 6.0 在编程环境、程序语言技术等方面做了许多改进，从而使 Visual C++ 更加适合专业程序员快速进行应用程序的开发。

### 1.1 对编译器的改进

Visual C++ 6.0 首先对编译器和链接程序进行了改进，不仅使其支持 COM（对象构件模型）应用程序的开发，并大大简化 COM 应用程序的开发工作。Visual C++ 6.0 还通过附带的大量演示例程来说明 Visual C++ 内嵌的编译器对 COM 程序的支持，阅读完这些程序，读者就会感到，复杂的 COM 程序其实并不神秘。而且，为了弥补 Visual C++ 5.0 编译器生成的可执行文件较为庞大的不足，Visual C++ 6.0 编译器在代码生成方面作了进一步优化，使生成的应用程序的大小仅是 Visual C++ 5.0 的 85%~90%。因而其编译产生的应用程序执行速度更快，效率更高。不仅如此，在 Visual C++ 6.0 集成开发环境中，我们可以通过设置链接程序的 /FIXED 选项开关，生成更小的、不含任何调试信息的 Release 版的可执行应用程序。集成调试环境通过增加 /G3, /G4, /G5, /G5, /G6 等编译器开关，允许开发人员根据自己处理器型号进一步优化所生成的应用程序。

为了进一步支持 C++ 语言中的异常处理能力，Visual C++ 6.0 新增添的 /EH 编译器开关选项，使得程序员能够充分利用 C++ 的强大异常处理能力，并使应用程序具有同步异常处理能力从而使开发出的应用程序更加强大。同步异常处理能力不仅可以有效处理异常，还能有效减小可执行应用程序的大小。

链接程序则通过提供的 /PDBTYPE 编译开关，使程序开发人员可以亲自对程序数据库进行设置，调整 PDB 中包含的程序的调试信息。高级程序员可以通过设置这一开关有效地节约宝贵的硬盘空间并提高链接的速度。

### 1.2 对 AppWizard 的改进

现在 Visual C++ 6.0 的 AppWizard 可以在基于对话框的应用程序中自动生成相应的对话框类，大大简化了创建基于对话框的应用程序的难度，并且在生成的基于对话框的应用程序中可以通过简单的选择，实现对 OLE 自动化的支持。此外，对话框类也可以像以往早期版本的 AppWizard 一样通过一个独立的代理类（proxy）实现 OLE 自动化。用户可以给生成对话框类添加合适的属性和方法逐步使应用程序功能完善。

Visual C++ 6.0 的 Custom AppWizard 允许用户自定义和修改用户创建工程的参数，例如用户可以修改编译器、链接器的参数，并且在使用自己创建的 Wizard 创建新工程的过程中可以自定义创建的步骤。



### 1.3 更完善的 MFC

Visual C++ 6.0 的类库在对 Internet 和数据库技术支持方面做了许多改进，使之比以前更为完善。首先它允许用户开发典型的基于 Internet 的应用，允许用户异步下载文件和设置应用程序的属性，并且在任务完成后，应用程序会自动释放系统资源供其他应用程序使用。此外 Visual C++ 6.0 开发的基于 Internet 的应用程序不仅可以在 Web 浏览器窗口中打开 ActiveX 文档，还可以在诸如 Microsoft Office Binder 之类的 OLE 对象包容器中打开 Active 文档。程序员使用完 MFC 类库提供的 Win32 Internet API (WinInet) 函数后就会发现 MFC 类库使得 Internet 与 Visual C++ 应用程序更加紧密地连接起来，并大大简化了编制 FTP、HTTP、Gopher 等协议应用程序的步骤。

### 1.4 数据库应用程序的增强支持

在对数据库应用程序的支持方面，VC6 中的 MFC 类库不但新添加了对 DAO (Data Access Objects, 即数据访问对象) 的支持，而且将原来的 ODBC API 函数进行了封装，提供了一系列 ODBC (Open Database Connectivity, 即开放式数据库连接) 类，以支持 ODBC3.0 标准。实际上，在 Visual C++ 6.0 中，集成调试环境还提供了一整套的 Visual Database 开发工具。Visual Database Tools 由 Query Designer 和 Database Designer 组成。利用 Query Designer，程序员可以利用可视化的方式创建 SQL 语句，或者对数据库进行修改。使用 Database Designer，程序员可以方便地对 SQL Server 6.5 进行添加、删除和编辑操作。通过 Active Template Library (ATL) 提供对 ActiveX 的完全支持。可以说，Visual C++ 6.0 在对数据库应用程序的支持方面丝毫不比 Visual Basic、Delphi 逊色。

### 1.5 相对于 Visual C++ 5.0 的新特性

Visual C++ 6.0 在界面和使用上与 Visual C++ 5.0 相比没有重大的变化，创建程序的过程、步骤及集成环境的操作上也没有什么新奇之处。但是它却带来了更为先进的技术和更为强大的工具。

微软开发工具的升级总是同它的系统平台的升级变化同步的。Visual C++ 1.52 称为 Windows 3.1 下 16 位应用程序开发的终结产品；Visual C++ 2.0 支持 Windows NT 及 Win32 程序的开发；Visual C++ 4.x 完善了对 Windows 95 及 Windows NT 4.0 的支持；当微软在 Windows 95 OSR2 版中捆绑 IE 3.0 时，Visual C++ 5.0 加强了对 Internet 及数据库的支持，提供了更新完善的 MFC 类库及 Office 97 式风格的界面，Visual C++ 5.0 和 Office 97 代表了新一代成熟的 Windows 95 应用程序，然而使用 Visual C++ 5.0 开发出的程序却落后于这一代的产品。当 IE4 推出后，即使 Service Pack 也未能跟上系统平台的进步，因为 Windows 98 及 Windows NT 5.0 即将出现。

Visual C++ 6.0 正好弥补了这一缺憾。Windows 98 及 Windows NT 5.0 是由 IE4 所带动，围绕 Internet 及相关的新技术所诞生的新一代操作系统。Visual C++ 6.0 的演化和特性与之相似。