

Microsoft Microsoft Microsoft

Microsoft®

Visual C++™ 2.0

for Win32® 大全 (一)

—— 用户指南

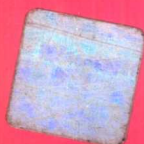
[美] Microsoft Corporation 著

赵人任 郑 峰 译

李 春 审校



清华大学出版社



Microsoft®
Visual C ++™ 2.0
for Win 32® 大全(一)
——用户指南

〔美〕 Microsoft Corporation 著
赵人任 郑峰 译
李 春 审校

清华大学出版社

(京) 新登字 158 号

Microsoft® Visual C++™ 2.0 for Win 32® 大全 (一) —— 用户指南

Microsoft Visual C++ User's Guide

Microsoft Corporation

Copyright 1995 by Microsoft Corporation.

Original English language Edition Copyright ©1994 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U. S. A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

中华人民共和国国家版权局著作权合同登记章 图字:01-95-313 号

未经出版者书面允许, 不得以任何方式复制或抄袭本书的内容。

本书封面贴有 Microsoft Press 激光防伪标签, 无标签者不得销售。

图书在版编目 (CIP) 数据

Microsoft Visual C++ 2.0 for Win 32 大全 (一): 用户指南/

美国微软公司著; 赵人任, 郑峰译。—北京: 清华大学出版社, 1995

ISBN 7-302-01934-7

I. M… I. ①美… ②赵… ③郑… III. C 语言, Visual C++ 2.0-手册 N. TP312

中国版本图书馆 CIP 数据核字 (95) 第 12854 号

出版者: 清华大学出版社 (北京清华大学校内, 邮编 100084)

印刷者: 北京清华大学印刷厂

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 21 字数: 521 千字

版 次: 1996 年 1 月第 1 版 1996 年 1 月第 1 次印刷

书 号: ISBN 7-302-01934-7/TP·883

印 数: 0001—5000

定 价: 38.00 元

引 言

Microsoft Visual C++ 2.0 Development System for Windows and Windows NT 是一个开发 C 和 C++ 应用程序的集成开发环境, 它支持多重平台和平台间的开发。Visual C++ 包括 C++ 应用程序的框架, Microsoft 基本类库 MFC3.0 等。它的推出极大地简化了 Windows 应用程序的开发以及应用程序到多重平台的移植。用户可以很容易地用 Visual C++ 和 MFC 在一个平台上开发 Windows 应用程序, 然后用同样的代码去为其它平台编制应用程序。

Visual 开发环境

Visual C++ 开发环境由一套集成工具所组成, 它们都是在 Windows 环境下运行的。你可以在同一个地方编制、测试并锤炼你的程序, 而毋须浪费时间去转换工作空间。让我们看一下 Visual C++ 的组成: 一个文本编辑器, 资源编辑器, 项目建立设施, 一个优化编译器, 一个增量链接器, 一个源代码浏览窗口以及一个集成调试器。在一个程序中你可以使用所有的工具。因为这些实用工具都在 Windows 环境下运行, 它们的操作方法具有风行的 Windows 风格, 用过 Windows 的用户会很容易接受。举个例子, 你在调试程序时可在编辑窗口中选择一个变量名, 然后把它拖动到观察窗口中某个位置, 调试器就会给该变量赋值并把结果显示在观察窗口中。或者你可以在对话框编辑器的工具条上选择一个控制, 然后拖动它到正被创建的一个对话框中, 并可随心所欲地改变它的大小或重新定位。Visual C++ 开发环境还提供一些工具条, 这样你只需单击按钮即可快速发出命令。而且, 当你把鼠标指针指向任一个按钮时, 它会显示一个描述性的标记, 这样有助于你正确选择。如果你不喜欢缺省的工具条, 你可重新定制或建立你自己的菜单条并把它们加到你所选择的菜单条按钮上去。

强有力的 Wizards

Visual C++ 提供两个强大的工具与 MFC 应用程序框架一起工作:

- Appwizard。Appwizard 产生一整套基于 MFC 类库的源文件和资源文件。通过选择 Appwizard 中的任选项, 你可用各种功能去定制开头文件。一旦你完成了所有选择, Visual C++ 就会从已建立的开头文件出发, 建立一个 Windows 应用程序的函数框架, 而无须用户费心。
- Classwizard。Classwizard 自动生成类的创建和编辑工作, 且可在 MFC 类库的基础上创建附加的新类。它为 new 类创建源代码, 成员函数以及消息映射 (易于把 Windows 的消息和类代码结合起来)。它还能把对话框中的数据映射到成员变量上并验证数据的合法性。

回顾一下用 Visual C++ 建立 Windows 应用程序的全过程。首先你应运行 Appwizard 以建立起程序的框架, 然后运行 Classwizard 以建立起程序的类、消息句柄和数据句柄, 最

后再把程序所要求的功能加到各个类中。

Visual C++ 项目

Visual C++ 以项目的形式来体现你的开发成果。一个项目包括应用程序的源文件以及建立项目的各项说明。每个项目都可以指定多个目标，而每个目标都对应着一段源代码。一个目标可以说明以下信息：应用程序的类型，基于运行的平台以及建立应用程序过程中所使用的工具设置。多重目标的蕴含允许你扩展项目的范围，但必须保持源代码的一致性以防出错。项目开发环境包括一个项目窗口，它以图的形式显示项目中的所有文件，这样用户可以很直观地检查这些文件之间的关系。甚至，用户还可以建立一些文件组，以便把某些关系更为密切的文件归为一组。另外，项目设置对话框允许你为整个项目，某个特定目标或特定文件指定工具设置。例如，你可以在全局范围内为整个项目指定某些编译器优化措施，但对某些目标或文件却可以摒弃之。

一旦你已指定了项目所包含的文件，所要建立的目标以及相应的工具设置，接下去就可以用项目菜单上的命令去建立该项目了。如果该项目所基于的平台不同于当前 Visual C++ 运行所附着的平台，请放心，开发环境会自动将编制好的应用程序传送到相应的远程机器上去。

建立错误修正

如果你的创建有错，Visual C++ 会帮你很快地找到出错位置。输出窗口显示一张错误表，记录了创建过程中产生的所有错误。按 F4 键，Visual C++ 显示源文件的编辑窗口，其中与输出窗口中第一个错误对应的那一行呈高亮度显示，这样你可以立即修正错误。另外，你也可利用菜单命令或相应热键迅速地定位并改正前面或后面的错误。

集成调试器

修正完所有创建过程中的错误，现在可借助于集成调试器来修正程序的逻辑错误了。Visual C++ 的调试器允许你监视程序的运行并可让它在你所选择的位置或环境下停止运行。例如，可以在程序中某一行设置一个断点，这样，当程序运行到这一行就会停止运行。也可以让程序在收到一个指定的 Windows 消息或在一个指定的异常发生时让程序挂起。如果你对某个变量的值感兴趣，也可让调试器中断程序运行，只要该变量的值发生了改变。

运用了对象链接与嵌入 (OLE) 技术的客户和服务器应用程序也可用集成调试器来调试。调试器可以逐行地执行客户 OLE 应用程序，而且，如果客户调用服务器 OLE 应用程序，Visual C++ 就会将它当成一个实例而让调试器从头开始执行它。这就让你很明确地了解到是否客户和服务器程序都运行正常。

任何运行失败的程序都可以用集成调试器来调试，不管该程序有没有调试信息，也不管它是否在前台运行。调试器能在程序存活时就启动工作。换句话说，这种“即时”调试允许你分析一个活动的程序而不是等它死亡后再来作事后剖析和检测。这样做的好处是，你可以找到问题所在处并改正它以让程序继续运行下去。

另外，调试器支持多平台和平台间的开发，这样，你就可以在一台远程机器上调试应用程序了。

源代码浏览窗口

当你开发和调试程序时，有可能需要查看在不同上下文中使用的类和各种符号。为此，当编制应用程序时，Visual C++会自动创建相应的浏览信息文件，其中包含了你所使用的全部符号的信息。浏览器窗口可以显示这些信息，并允许你在各实例间随意地移动那些在源代码中所用到的符号。另外，你可在浏览器窗口中查看类家族继承关系图以及函数调用关系图。并且可以很容易地查看任一个给定文件中所使用到的全部符号，继而显示任一个符号的定义，以及项目中所有的引用。若你想打开包含某个特定引用的文件，只须双击窗口内的对应条目即可。以上所陈述的种种都是 Visual C++ 为你所提供的方便，即你可以随意查看任一个关系并可随意在文件间移动这些关系，这样，最大限度地方便了你去维护，校对及调试源代码。

用户偏好

Visual C++ 也允许用户按照自己喜欢的方式去定制一些操作。例如，在文本编辑窗口中，开发环境允许你自己选择某种颜色去显示语言元素（诸如注释或关键字），或者以某种特定字体去显示文本。当你为一个特定项目安排好窗口布局时，Visual C++ 会记住这一布局，等你下一次启动该项目时，Visual C++ 会将你最近一次所使用的布局原封不动呈现给你，换句话说，所有打开的文件和窗口位置都与上一次相同。另外，当你调试一个应用程序的时候，可能会选中某些窗口和工具条，而 Visual C++ 会记住这些选择，在继后的子序列调试对话期间，这些选择会被继承下来。如果你不愿使用 Visual C++ 所提供的缺省命令，可以按照你的偏好去改变任一个热键设置，或增加一些热键，或为某个命令设置多重热键，或指定某些窗口，在其中任一热键都是活动的。在文本编辑窗口中，Visual C++ 也可帮助你记录下各次击键动作，并在必要时重新执行该命令序列。

扩展信息

Visual C++ 提供一些手段以获取有关开发环境或支撑软件的信息。*Introducing Visual C++* (Visual C++ 介绍) 包含了一系列的指导说明以帮助你熟悉开发环境，连同在开发环境中所使用到的方法和过程，以及如何用 Microsoft 基本类库去开发应用软件。当你在编辑窗口中编制源代码时，可以随时获取任一个 MFC 类的信息，或者 C 运行库中的任一个函数名，或者任一个语言元素（注释、关键字等）。只要你选中一个名字，按 F1 或 CTRL+F1，Visual C++ 就会立即显示有关该名字的引用信息。Help 菜单上的 Books Online 命令会显示一张表格，里面列出了所有 Visual C++ 系统所涉及到的书的信息。你可浏览这张表并可选择某个主题以求深入了解。另外，Search plus 命令可使你搜索查询联机书本的所有文本内容，只要你选择了一个主题并给出查找关键字或字的组合。若你需要了解某个打开的对话框的信息，只须单击 Help 按钮，Visual C++ 就会提供对话框中所有控制的描述，并且如果有必要的话，会提供给你获取更多深入信息的方法。

Visual C++ 用户指导

Visual C++ 用户指导中有一些例程以帮助了解如何用该开发环境去进行各种各样的开发工作。另外，还包括了有关命令行工具的引用信息。

文档约定

这本书所使用到的一些术语和图约定如下：

STDIO. H

大写字母表示文件名、寄存器以及在操作系统命令级上所使用的项。

Char, _alloca

黑体字表示 C 和 C++ 的关键字、操作符、特定语言的字符以及程序库。

在语法说明中，黑体字表示该文字必须按此种格式原样照搬。

有些常数、函数名以及关键字以一个或两个下划线打头。注意下划线也是名字的一部分并且是必不可少的。例如，要想使 `__cplusplus` 当作常数被编译器识别，开头的两个下划线必须带上。

expression

斜体字表示此处信息是由用户来提供的，例如文件名。另外，斜体字有时也用在文本中以示强调。

[[option]]

双重方括号内的项是可选的。

#pragma pack {112}

花括号连同竖线表示必须在两项或更多项中选一项，除非花括号外有中括号。

#include <io.h>

这种字体用于例子，用户输入、程序输出、以及文本中的错误消息。

CL [[option...]] file...

三点（省略号）后跟一个项表示更多相同格式的项可能会出现。

while () {...} {;}

一列或一行三点表示一段例程被故意省略掉了。

CTRL+ENTER

大写字母用来表示键盘上键的名字。如果有“+”连接着两个键名，表示必须先按住头一个键，再按下第二个键，这样才起作用。

“argument”

引号中的项是在文本中首次定义的新项。

“C string”

一些 C 语言的构造，例如字符串，需要用引号引起来。这些引号的格式是“ ” 或 ‘ ’ 而不是 “ ” 或 ‘ ’，并且是黑体的。

Dynamic-Link Library (DLL)

当字首组合词（如 DLL）第一次出现时，通常会将原文全部拼写出来。

x86 Specific→ **END x86 specific**

本书中陈述的信息有可能并非适用于所有的目标平台。于是，仅适用于 x86 目标平台处将会如此标记。

目 录

引 言	IX
-----------	----

第一部分 使用 Visual C++

第 1 章 用 AppWizard 创建一个新的应用程序	3
1.1 创建一个 AppWizard 项目	3
1.2 选择数据库和 OLE 任选项	5
1.3 选择应用程序和项目任选项	6
1.4 结束 AppWizard 过程	8
1.5 AppWizard 所创建的文件	8
第 2 章 项目的使用	9
2.1 项目类型	9
2.2 使用项目	10
2.3 在项目中使用元组	15
2.4 在项目内设置任选项	16
2.5 使用外部项目	19
2.6 使用预编译头文件	21
2.7 编制项目	21
2.7.1 编制当前目标	22
2.7.2 编制多重目标	22
2.8 编制单个源文件（不涉及项目）	23
2.9 运行程序	23
第 3 章 使用文本编辑器	25
3.1 管理文件	25
3.2 在源文件内漫游	28
3.3 源窗口的控制	29
3.4 用 Visual C++ 文本编辑器编辑文本	31
3.5 搜索及替换文本	32
3.6 记录并演示热键	36
3.7 设置 Visual C++ 文本编辑器选项	36
第 4 章 如何利用资源	40
4.1 Visual C++ 中的文件	40
4.2 使用已有的非 AppWizard 创建的资源文件	41
4.2.1 将已有的资源文件转换成 Visual C++ 格式	41

4.2.2	仅为 MFC 资源文件所支持的特征	42
4.3	使用资源编辑器	42
4.4	公共资源编辑过程	43
4.5	使用标识符	45
4.5.1	改变标识符或标识符的名称	46
4.5.2	改变标识符的值	46
4.5.3	用标识符浏览器来管理标识符	47
4.5.4	标识符名称限制	48
4.5.5	标识符值限制	49
4.6	使用高级资源文件技术	49
4.6.1	改变标识符头文件的名称	50
4.6.2	使用共享（只读）标识符或可计算标识符	50
4.6.3	用其它文件来包含资源	51
第 5 章	使用对话编辑器	52
5.1	改变对话特征	52
5.2	控制的类型	53
5.3	添加控制到对话框中	53
5.4	选取所要的控制	54
5.5	移动控制	55
5.6	删除及复制控制	56
5.7	决定控制的尺寸	56
5.8	改变制表次序	57
5.9	定义加速键	58
5.10	安排控制的布局	58
5.10.1	使用 Snap to Grid	59
5.10.2	排列控制	59
5.11	使用定制的控制	61
5.11.1	用户如何定制控制	61
5.12	创建格式视图对话框	61
5.13	测试对话框	62
第 6 章	使用菜单编辑器	63
6.1	创建菜单或菜单项	63
6.2	选取菜单和菜单项	65
6.3	创建热键菜单	65
6.4	移动或复制菜单及菜单项	66
6.5	以弹出式菜单的形式查看菜单资源	67
6.6	为菜单项定义加速键	67
6.7	为菜单项定义状态条提示	67
第 7 章	使用加速键编辑器	68
7.1	编辑加速键表	68

7.2	设置加速键的特征	69
7.3	将加速键与菜单项联系起来	69
第8章	使用串编辑器	70
8.1	在串表中查找串	70
8.2	添加或删除一个串	71
8.3	将串从一个段移到另一个段中	71
8.4	将串从一个资源描述文件移到另一个中	72
8.5	修改串及其标识符	72
8.6	向串中添加格式化字符或特殊字符	72
第9章	使用图形编辑器	73
9.1	编辑图形用的窗口和工具	73
9.1.1	图象编辑器窗口	73
9.1.2	Graphics 工具条	73
9.1.3	Colors 调色板	73
9.1.4	状态条	74
9.1.5	Image 菜单	74
9.2	编辑图形资源	74
9.2.1	设置位图特征	75
9.2.2	显示和隐藏 Graphics 工具条	76
9.2.3	指定前景色和背景色	76
9.2.4	绘制和擦除	76
9.2.5	画直线或画闭合曲线	77
9.2.6	填充有边界的区域	78
9.2.7	拾取颜色	78
9.2.8	选择一块位图区	79
9.2.9	切除、拷贝、清除以及移动	79
9.2.10	选择透明的或不透明的背景	80
9.2.11	翻转选择区域	80
9.2.12	反转选择区域的颜色	81
9.2.13	定制画刷	81
9.2.14	改变位图的尺寸	81
9.3	管理图形编辑器的工作空间	82
9.3.1	使用图象编辑器窗口	83
9.3.2	缩小或放大	83
9.3.3	改变放大率	84
9.3.4	显示及隐藏象素栅格	84
9.4	创建新的图标或光标	84
9.4.1	创建新的图象	85
9.4.2	选取显示设备	85
9.4.3	用屏幕色及其反转色绘制图形	85
9.4.4	设置光标热点	86

9.5	编辑 Graphics 工具条	86
9.5.1	设置平铺栅格的尺寸	86
9.5.2	创建新的工具条按钮	87
9.6	改变颜色	88
9.7	保存及安装 Colors 调色板	88
第 10 章	使用二进制数据编辑器	90
10.1	创建新的数据资源或定制资源	90
10.2	打开一个二进制资源	91
10.3	编辑二进制数据	91
第 11 章	使用版本信息编辑器	92
11.1	编辑版本信息	92
第 12 章	使用 Class Wizard	94
12.1	添加一个新的具有用户界面的类	94
12.1.1	添加一个新的类	95
12.2	输入一个类	96
12.3	将消息映射到函数上	97
12.3.1	添加消息句柄	97
12.3.2	删除消息处理函数	99
12.3.3	编辑消息处理程序	99
12.3.4	加载虚函数	99
12.4	处理对话框数据	100
12.4.1	对话数据交换	100
12.4.2	对话数据验证	102
12.4.3	定制数据的交换和验证	103
12.5	让 ClassWizard 随代码的改变而更新	103
12.5.1	删除类	103
12.5.2	改变类名或改变类的位置	103
12.5.3	重新编制 ClassWizard (.CLW) 文件	104
第 13 章	浏览程序中的符号	105
13.1	处理浏览信息文件	105
13.2	使用 Browse 工具条	107
13.3	显示文件中的符号	108
13.4	显示类的信息	110
13.5	显示函数信息	113
13.6	查找符号的定义和引用	115
13.7	修改浏览窗口的显示	117
第 14 章	使用调试器	118
14.1	使用断点	121

14.2	控制程序的运行	126
14.3	浏览和修改变量的值	128
14.4	使用 Memory 窗口及 Disassembly 窗口	134
14.5	高级调试	136
第 15 章	剖视代码	143
15.1	设置剖视器	143
15.2	为剖视建立代码	143
15.3	运行剖视器	144
15.4	剖视的类型	145
15.4.1	函数剖视	145
15.4.2	行剖视	146
15.5	选择性的剖视	148
15.5.1	修改 TOOLS.INI	148
15.5.2	指定要剖视的函数	148
15.5.3	指定要剖视的行	148
15.5.4	为剖视选择起始函数	149
15.6	其它剖视器特征	149
15.7	在 Win32s 下剖视	150
第 16 章	使用 Spy++	151
16.1	启动 Spy++	151
16.2	在 Spy++ 中工作	151
16.2.1	Spy++ 工具条	151
16.2.2	窗口查找工具	152
16.2.3	扩展 Spy++ 树	152
16.2.4	刷新视图	153
16.2.5	WinHelp 跳转	153
16.2.6	改变字体	153
16.3	Spy++ 视图	153
16.3.1	Windows 视图	153
16.3.2	Processes 视图	153
16.3.3	Threads 视图	154
16.3.4	Messages 视图	154
16.3.5	控制 Messages 视图	155
16.4	Spy++ 属性	157
16.4.1	Window Properties	157
16.4.2	Process Properties	159
16.4.3	Thread Properties	161
16.4.4	Message Properties	162
第 17 章	定制 Visual C++	163
17.1	用窗口工作	163

17.1.1 用文档窗口工作	165
17.1.2 用船坞工具窗口工作	166
17.2 用工具条工作	169
17.3 定制键盘	175
17.4 定制 Tools 菜单	176
17.5 显示状态条	180
17.6 设置目录	180
17.7 为帮助文件指定搜索顺序	181

第二部分 Visual C++ 参考

第 18 章 工具条参考	185
18.1 缺省 Visual C++ 工具条	185
18.2 工具条的种类	190
第 19 章 键盘快捷键	194
第 20 章 设置编译器选项	202
第 21 章 设置链接器选项	226

第三部分 附录

附录 A CL 参考	237
A.1 CL 语法的描述	237
A.2 使用 CL	239
A.3 只能在命令行上使用的选项	240
附录 B LINK 参考	251
B.1 新的 LINK 特性	251
B.2 LINK 输入文件	253
B.3 LINK 输出	254
B.4 在命令行上运行 LINK	255
B.5 LINK 选项	256
B.6 模块定义文件	264
B.7 保留字	266
附录 C 剖视器参考	267
C.1 剖视器批处理	267
C.1.1 剖视器批响应文件	268
C.1.2 标准批处理文件	268
C.2 剖视器命令行选项	269
C.2.1 PREP	269
C.2.2 PROFILE	270

C. 2. 3	PLIST	271
C. 3	从剖视器输入数据	273
C. 3. 1	用制表符定界的记录格式	273
C. 3. 2	分析剖视器统计数据步骤	276
C. 3. 3	用 Microsoft Excel 处理剖视器输出	276
附录 D	LIB 参考	278
D. 1	LIB 概述	278
D. 2	运行 LIB	279
D. 3	管理一个库	280
D. 4	抽取一个库成员	281
D. 5	输入库和输出文件	281
附录 E	BSCMAKE 参考	284
E. 1	建立一个 .BSC 文件	284
E. 1. 1	创建一个 .SBR 文件	284
E. 1. 2	BSCMAKE 怎样建立一个 .BSC 文件	285
E. 1. 3	提高 BSCMAKE 的效率	285
E. 2	BSCMAKE 命令行	286
E. 2. 1	BSCMAKE 命令文件	286
E. 2. 2	BSCMAKE 选项	287
E. 2. 3	BSCMAKE 出口代码	288
附录 F	DUMPBIN 参考	289
F. 1	DUMPBIN 命令行	289
F. 2	DUMPBIN 选项	289
附录 G	EDITBIN 参考	291
G. 1	EDITBIN 命令行	291
G. 2	EDITBIN 选项	291
附录 H	NMAKE 参考	294
H. 1	运行 NMAKE	294
H. 2	MAKE 文件的内容	296
H. 3	描述块	297
H. 3. 1	目标	297
H. 3. 2	依赖	299
H. 4	Make 文件中的命令	299
H. 4. 1	命令修饰符	300
H. 4. 2	文件名部分语法	300
H. 4. 3	Make 文件中的内联文件	301
H. 5	宏和 NMAKE	302
H. 5. 1	定义一个 NMAKE 宏	302

H. 5.2	使用一个 NMAKE 宏	303
H. 5.3	特殊 NMAKE 宏	303
H. 6	推理规则	305
H. 6.1	定义一条规则	305
H. 6.2	预定义规则	306
H. 6.3	推理的依赖和规则	306
H. 6.4	推理规则中的优先级	306
H. 7	点命令	306
H. 8	Make 文件预处理	307
H. 8.1	Make 文件预处理命令	307
H. 8.2	Make 文件预处理中的表达式	308
附录 I	资源编译器	310
I. 1	在应用程序中蕴含资源	310
I. 2	创建一个资源定义文件	310
I. 2.1	预处理命令	311
I. 2.2	单行语句	311
I. 2.3	多行语句	311
I. 2.4	描述文件的例子	312
I. 3	RC 命令行	313
I. 3.1	为预处理器定义名字	313
I. 3.2	命名编译的资源文件	314
I. 3.3	查找文件	314
I. 3.4	显示进度消息	315
I. 3.5	公共语句参数	315
I. 3.6	公共控制参数	315
I. 3.7	公共资源属性	316
I. 3.8	内存属性	317
附录 J	修饰名	318
J. 1	使用修饰名	318
J. 2	察看修饰名	319

第一部分

使用 Visual C ++