

Vehicle OTS

VIN

Validste VIN

程序設計

# Visual C++6.0

## 程序设计

乔林 杨志刚 何敏 编著

Color

Style

Year

Make

Model

Validste VIN

# 入门篇

# 序 言

Visual C++6.0 是一个全新的可视化的 C++ 编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具，它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的、可重用的和完整的面向对象程序（Object-Oriented Programming OOP）语言。对于广大的程序开发人员来讲，使用 Visual C++6.0 开发应用程序和数据库应用程序，无疑会大大地提高编程效率，而且随着应用的深入，我们将会发现编程不再是枯燥无味的工作，Visual C++6.0 的每一个设计细节都隐藏了值得我们挖掘的丰富的宝藏。

## Visual C++6.0 的新特征

坦率地说，一名优秀的编程人员需要一点小聪明。在 Windows 操作系统出现以前，编程实际上是一种激动人心的智力体验。而随着 Windows 操作系统的不断升级和普及，编程正在变成越来越枯燥无味的事情。然而如果要开发高级应用程序，我们将会发现，Visual C++6.0 几乎是唯一的选择。毕竟，市场上还没有一种 C++ 或其他语言的编译器的功能像 Visual C++6.0 一样强大，文档支持像 Visual C++6.0 一样详尽。Visual C++6.0 支持全部 ANSI C/C++2.0 标准的高级特性，例如模板（template）、名字空间（name space）和操作符重载（operator overloading），此外，Visual C++6.0 也完全支持 Windows API 函数，例如割边（cutting-edge）函数 DirectX、OLE 自动化（automation）和 ActiveX。

Visual C++6.0 实际上是 C++ 语言的一种版本，但它与传统的 C++ 语言有很大的差别。一个 Visual C++6.0 程序首先是应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，也仍然可以严格地按照设计运行。程序员的工作只是在“骨架”中加入自己的程序。缺省的应用程序是一个空白的主框架窗口（类 CmainForm），我们可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以放大缩小、移动、最大化和最小化等。因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。

## 本丛书的组织

本丛书包括三部分——《入门篇》、《提高篇》和《精通篇》。

《入门篇》是为初级用户编写的，主要讨论 Visual C++6.0 环境、编程的基本方法和 C++ 语言的基础知识，《入门篇》由乔林、杨志刚和何敏主编；《提高篇》是为中级用户编写的，主要讨论 Visual C++6.0 开发 MDI 应用程序、SDI 应用程序和位图与动画编程的基础知识，《提高篇》由乔林、杨志刚和刘文杰主编；而《精通篇》则为高级用户编写的，主要讨论使

用 Visual C++6.0 进行高级控件开发和深入编程的关键技术,《精通篇》由乔林、杨志刚和刘文杰主编。

此外,为了帮助读者理解本丛书所使用的 OOP 技术,《精通篇》还包含了几个较为详尽的附录,讨论了 OOP 的基本概念,并对 OOP 语法作了较为深入的分析。

我们这里要着重指出三点:

首先,本丛书是基于图形用户界面的,特别是基于 Windows 95/98/NT 界面的,这样就很容易掌握一些最重要的和最基本的概念。

其次,本丛书的内容没有一项是特别的技术。关于 Visual C++6.0 语言的精确定义可以从各种各样的资料上得到,其中最重要的一个就是 Visual C++6.0 的联机帮助系统。当然,只有理解了事情怎样做和为什么这样做时,这些介绍才能有所帮助,否则语言描述本身就太抽象了。

最后,本丛书通过大量由浅入深的实例讲解 Visual C++6.0 的核心内容。与简单地完成某项编程任务相比,我们更关心 Visual C++6.0 如何完成这项编程任务。因此,本丛书包含了大量的 Visual C++6.0 程序代码,这部分代码是深入了解 Visual C++6.0 和 MFC 的基础。

正是基于上述三点考虑,贯穿本丛书的是读者应该遵循什么样的步骤,应该采用什么样的思考方法,以及如何将自己的思考转化为正确的程序代码。因为只有在“干中学”才能有最大的收获,所以本丛书包含了大量实例,这些例子有的难,有的易,而且在很多情况下还有一定的关联。我们希望读者在阅读本书的同时也上机实践。

本丛书的内容由浅入深,没有阅读前面内容的读者可能会给其后的学习带来一定的麻烦,当然对相应内容有相当程度了解的读者又另当别论了。

本丛书主要由乔林、杨志刚、何敏和刘文杰负责编写,此外,王程铭、林杜和费广正参与了《入门篇》的编写,王程铭、费广正和乔林参与了《提高篇》的编写,周岚、林杜和金传恩参与了《精通篇》的编写。丛书由乔林和杨志刚统一规划组织,乔林同时审阅了丛书的全部内容。书中如有错误当是作者水平有限,恳请读者谅解。

## 对本丛书读者的忠告

倾向于使用 C++语言的程序员,总是欣赏 Visual C++6.0 编程时的深入性和编译后程序代码的运行速度与可优化性能。因此,鉴于 Visual C++6.0 语言的这种灵活性和高效率,如果要进行实用程序的开发,养成良好的编程风格是非常有必要的。无论是编写什么样的应用程序,如果对程序中任何一段代码的功能和要完成的任务不太明了,我们都推荐读者仔细思考并打上标记。事实上,笔者就是这么做的。

从某种程度上说,任何程序员都不是完美的。因此,本书所有的例子都基于这样一种假设,那就是笔者不仅会犯错误,而且很快就会犯错误。正是基于这样一种假设,本书才有了足够的自由度为实际的应用程序尝试一定程度的创造性解决方案。读者一定要注意,一定程度的创造性是必要的,但太专业或充满技巧的代码则是 bug 的温床,然后是个多个不眠之夜的前奏。

最优秀的程序员也会犯错误,而且是非常频繁地犯错误!笔者承认并接受这个事实,

然后试图寻找一种安全的编程方案，以使得程序在笔者无法预料的情况下也能够安全地运行。

正确的代码总是应该能够以快速的方式完成指定的功能。笔者倾向于编写既不太抽象（太一般）也不太敏感（太特殊）的代码，这种特殊与一般的编程折衷方案正是笔者所欣赏的。至少，这样的代码既能完成指定的功能，也能在笔者没有对它厌倦之前产生创造的欲望，这正是编程的辩证法。

优秀的程序员永远不要尝试成为一个完美主义者，苛求完美只会将你引入歧途。稍微降低一点你的目标，保证所编写的程序可以工作，然后再要求它“似乎是无 bug 运行的”就行了。程序越大越复杂，错误也就越多，Windows 95 的 5000 个 bug 就是最有力的例证（见：IEEE Computer, 1998, No. 7, pp107-109）。笔者从不期望自己编写的程序是完美无缺的，这也可以作为对读者的忠告。

编 者

1999 年 3 月

# 目 录

<b>第1章 Visual C++ 6.0 开发环境</b> .....	1
1.1 Visual C++ 6.0 的安装 .....	1
1.2 Visual C++ 6.0 主窗口 .....	4
1.3 工具栏.....	6
1.4 菜单栏.....	7
1.5 Visual C++ 6.0 项目与项目工具区 .....	32
1.6 Visual C++ 6.0 资源与资源编辑器 .....	34
1.7 小 结.....	48
<b>第2章 使用 Visual C++ 6.0 语言</b> .....	49
2.1 Visual C++ 6.0 语句与语句块 .....	49
2.2 Visual C++ 6.0 保留字和操作符 .....	51
2.3 Visual C++ 6.0 标识符 .....	57
2.4 常量与变量.....	58
2.5 Visual C++ 6.0 中的类型 .....	62
2.6 类类型.....	70
2.7 编译指令.....	73
2.8 注 释.....	78
2.9 小 结.....	79
<b>第3章 控制结构与数组</b> .....	80
3.1 分支控制结构.....	80
3.2 循环控制结构.....	86
3.3 数 组.....	90
3.4 小 结.....	92
<b>第4章 函 数</b> .....	93
4.1 函数和类方法的定义.....	93
4.2 内联函数.....	95
4.3 函数和类方法的参数.....	97
4.4 函数和类方法的返回值.....	101
4.5 函数和类方法的调用.....	102
4.6 小 结.....	105
<b>第5章 指针与引用</b> .....	106

5.1 指针的概念 .....	106
5.2 指针运算 .....	107
5.3 多级指针 .....	109
5.4 指针和数组 .....	110
5.5 指针和动态内存分配 .....	113
5.6 引 用 .....	115
5.7 类型定义 .....	118
5.8 指针和函数 .....	119
5.9 特殊的指针与复杂说明 .....	124
5.10 小 结 .....	127
<b>第 6 章 结构与链表 .....</b>	<b>128</b>
6.1 结构说明 .....	128
6.2 结构和指针 .....	129
6.3 结构与函数 .....	130
6.4 复杂结构与链表 .....	132
6.5 小 结 .....	139
<b>第 7 章 进入面向对象程序设计的新天地 .....</b>	<b>140</b>
7.1 构造函数和析构函数 .....	140
7.2 数据成员的初始化 .....	147
7.3 赋 值 .....	149
7.4 const 修饰的成员函数 .....	150
7.5 重 载 .....	152
7.6 继 承 .....	157
7.7 应用程序实例 .....	170
7.8 异常处理基础 .....	189
7.9 小 结 .....	196
<b>第 8 章 文件与输入输出流类 .....</b>	<b>197</b>
8.1 标准文件类型 .....	197
8.2 非缓冲文件 .....	201
8.3 标准 Windows 文件管理函数 .....	203
8.4 文件 I/O 流类 .....	206
8.5 MFC 文件类 CFile .....	219
8.6 小 结 .....	221
<b>第 9 章 Windows 和 MFC 应用程序 .....</b>	<b>222</b>
9.1 应用程序向导 AppWizard .....	222

9.2	应用程序类与源文件的组织	228
9.3	程序运行过程分析	252
9.4	窗口类	257
9.5	更新用户界面	259
9.6	小结	260
<b>第 10 章</b>	<b>文档与视图结构</b>	<b>261</b>
10.1	窗口类	261
10.2	文档与视图结构	263
10.3	消息与命令的处理	268
10.4	MFC 的数组类	277
10.5	响应鼠标事件	279
10.6	小结	324
<b>第 11 章</b>	<b>GDI 与文本处理</b>	<b>325</b>
11.1	设备场境	325
11.2	绘图工具	326
11.3	绘图函数	329
11.4	映射模式与绘图模式	331
11.5	CString 类	335
11.6	文本处理	347
11.7	小结	360
<b>第 12 章</b>	<b>使用对话框与控件</b>	<b>361</b>
12.1	对话框	361
12.2	创建模式对话框	366
12.3	在对话框中使用控件	381
12.4	通用对话框	387
12.5	小结	392

# 第 1 章

## Visual C++ 6.0 开发环境

Visual C++ 6.0 是 Microsoft 公司开发的基于 Windows 95/98/NT 的编程工具。正是因为 Visual C++ 6.0 与 Windows 操作系统千丝万缕的联系，所以如果要深入掌握其编程机制，成为一名熟练的 Visual C++ 6.0 程序设计师，就必须对 32 位 Windows 操作系统的运行机制有深入的了解。这一点对于用好 Visual C++ 6.0 以及开发高质量的 32 位 Windows 应用程序是至关重要的。

Visual C++ 6.0 的最大特色是对面向对象技术的强大支持，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 Microsoft 基础类库（Microsoft Foundation Class, MFC）的方式提供给开发人员使用，提高了软件的重用性。

使用 Visual C++ 6.0 开发应用程序时，我们可以使用开发工具所提供的现成组件，利用所见即所得的方式完成程序界面的设计。这种方法大大增强了程序设计人员的效率。

作为学习 Visual C++ 6.0 的第一步，本章首先介绍 Visual C++ 6.0 的开发环境，使用户对其工作环境有较深入的了解。

### 1.1 Visual C++ 6.0 的安装

使用 Visual C++ 6.0 开发应用程序之前，必须先将 Microsoft Developer Studio 98 和 Visual C++ 6.0 安装到计算机上。我们可以使用安装程序进行新的安装或更新已有的安装。此外，安装程序还可以根据配置情况对安装进行裁剪。

在 Windows 95/98/NT 中安装 Visual C++ 6.0 企业版（Enterprise Edition）的过程如下：

- (1) 打开计算机，启动 Windows 95/98/NT。
- (2) 在 CD-ROM 驱动器中插入 Visual C++ 6.0 企业版系统光盘。
- (3) 单击“开始”按钮，从“开始”菜单选择“运行”命令，弹出“运行”对话框。

在“打开”文本框键入以下命令，如图 1-1 所示：

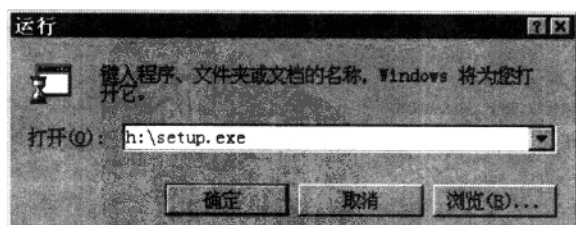


图 1-1 在“运行”对话框键入安装程序的名称



- (4) 系统将首先安装 Microsoft Internet explorer 4.0 和 DCOM。如果操作系统为 Windows NT 4.0 上, 则必须首先在 Windows NT 4.0 上安装 Service Pack 3。
- (5) 系统弹出“Welcome”对话框, 单击“Continue”按钮, 安装程序显示“Registration”对话框, 要求用户输入姓名 (User)、公司名称 (Organization) 和序列号。
- (6) Visual Studio 6.0 有四种安装类型, 即典型安装 (Typical)、最小安装 (Minimum)、CD-ROM 安装和定制安装 (Custom)。其中, 最小安装只安装开发环境、MFC 类库文件、编译器、库、工具以及产品支持服务帮助文件 (PSS 文件和 HLP 文件); 典型安装主要安装开发环境、项目建立工具 (编译器、链接器、资源编译器以及用于建立 Win32 应用程序的其他工具)、运行时间库 (Runtime Libraries)、MFC 类库与模板库、数据库驱动程序集、常用工具; CD-ROM 安装只将开发环境安装到计算机中, 其他部件从 CD-ROM 运行; 定制安装则允许用户选择要安装的部件。
- (7) 我们这里使用定制安装。系统弹出“Visual Studio 6.0 Enterprise Setup”对话框, 如图 1-2 所示。要求用户选择安装目录, 缺省安装目录为“C:\Program Files\Microsoft Developer Studio”。设置单击“Custom”按钮, 安装程序将开始安装过程。

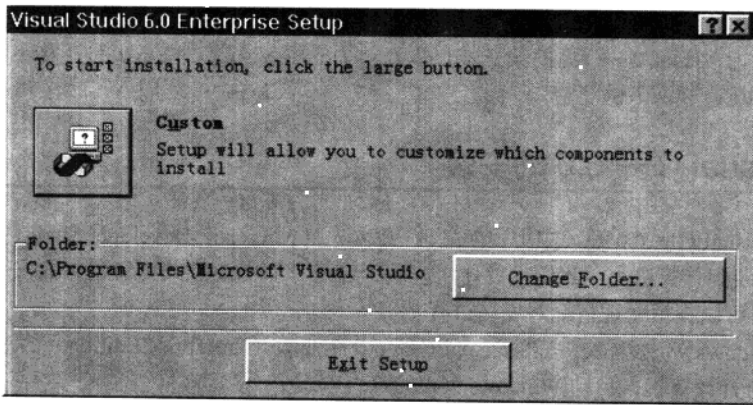


图 1-2 选择系统安装目录

- (8) 选择要安装的子系统, 包括“Microsoft Visual Basic 6.0”、“Microsoft Visual C++ 6.0”、“Microsoft Visual FoxPro 6.0”、“Microsoft Visual InterDev 6.0”、“Microsoft Visual SourceSafe 6.0”、“ActiveX”、“Data Access”、“Enterprise Tools”等, 如图 1-3 所示。
- (9) 选择“Microsoft Visual C++ 6.0”选项, 系统将弹出如图 1-4 所示的对话框。对于其中的每个选项, 都单击“Select All”按钮, 选择全部内容。

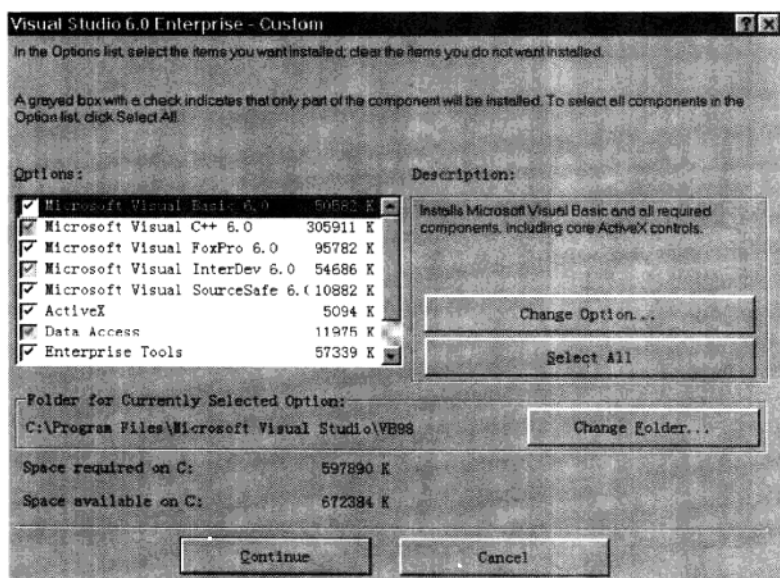


图 1-3 选择要安装的子系统

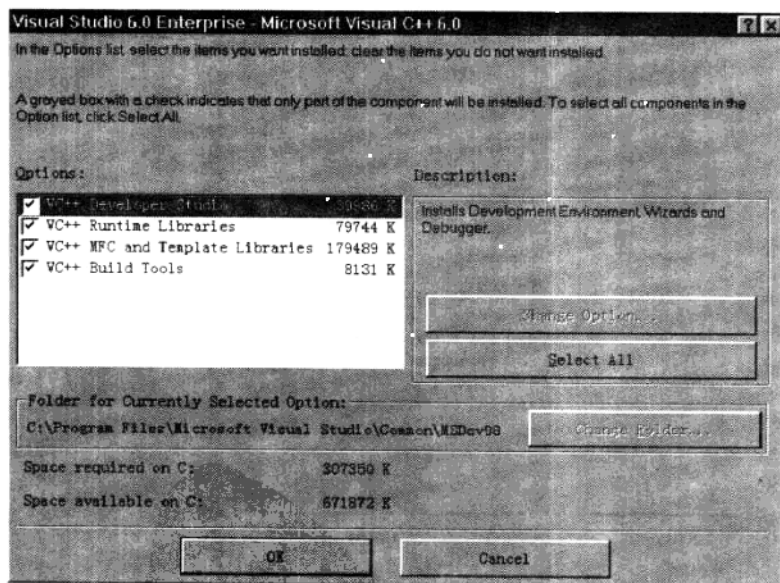


图 1-4 安装 Microsoft Visual C++ 6.0 子系统

- (10) 安装 VC++ 企业版工具, 如图 1-5 所示。全部安装选项选择完毕之后, 单击“OK”按钮, 系统将显示如图 1-6 所示的安装指示, 开始安装过程。整个安装过程大约需要 1 个小时。

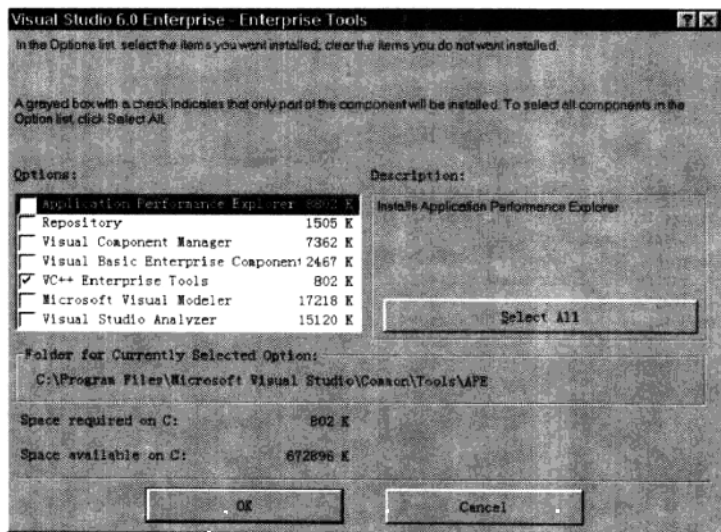


图 1-5 安装 Microsoft Visual C++ Enterprise Tools (企业版工具)

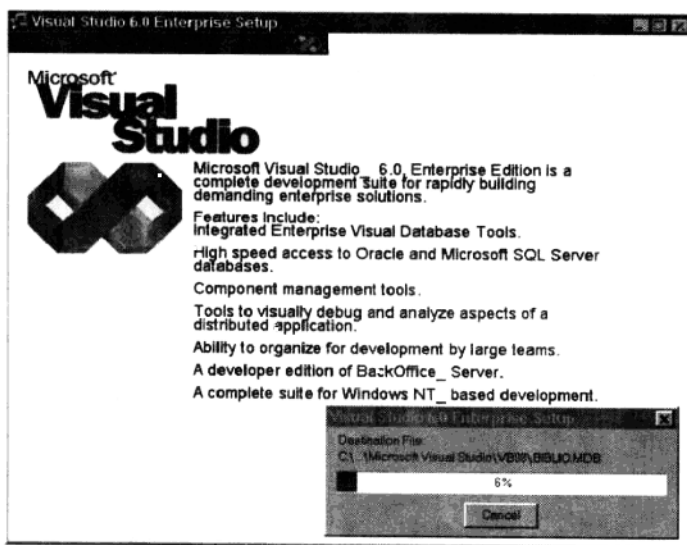


图 1-6 安装过程指示

## 1.2 Visual C++ 6.0 主窗口

首先按第一章所介绍的步骤将 Visual C++ 6.0 安装到 Windows 95/98/NT 中,接着单击“开始”按钮,从“开始”菜单启动 Visual C++ 6.0 进入开发环境 Developer Studio,如图 1-7 所示。

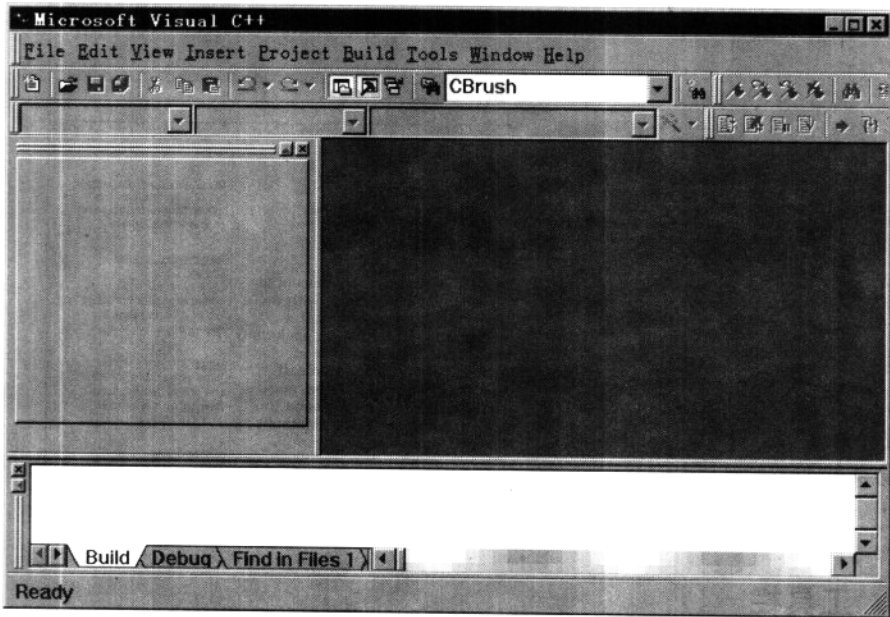


图 1-7 Visual C++ 6.0 主窗口

Developer Studio 由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态栏组成。

标题栏位于屏幕最上端，标题栏显示了应用程序名称和当前打开的文件名称，标题栏的颜色用于指明对应窗口是否为激活的。操作系统提供的控制菜单位于标题栏的左边，它用于控制窗口的大小和位置，以及进行还原、移动、关闭、最大化和最小化等操作。标题栏的右边有三个控制按钮，从左至右分别为最小化按钮、还原按钮和关闭按钮，这些按钮用于快速设置窗口大小。

应用程序最重要的菜单栏和工具栏位于标题栏的下面。工具栏的下面是两个窗口，一是工作区窗口，一是编辑窗口。在这两个窗口下面的是输出窗口，输出窗口用于显示项目建立过程中所产生的错误信息。屏幕最底端是状态栏，状态栏上给出了当前操作或命令的提示信息。

Visual C++ 6.0 的帮助信息显示在单独的与 Internet Explorer 4.0 相似的浏览窗口中，如图 1-8 所示。我们这里要提醒读者注意的是，所有 Microsoft Developer Studio 子系统都使用相同的帮助浏览窗口。

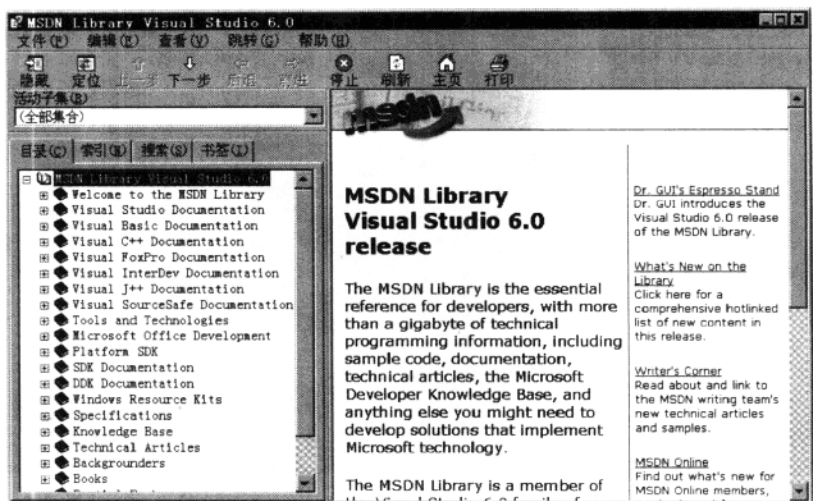


图 1-8 Visual C++ 6.0 帮助窗口

## 1.3 工具栏

Visual C++ 6.0 的工具栏由某些操作按钮组成，分别对应着某些菜单选项或命令功能，可以直接用鼠标单击这些按钮来完成指定功能。工具栏按钮大大简化了用户的操作过程，并使操作过程可视化。

Visual C++ 6.0 包含 10 个工具栏。缺省时，屏幕工具栏区域显示两个工具栏——Standard 工具栏（如图 1-9 所示）和 Build 工具栏（如图 1-10 所示）。

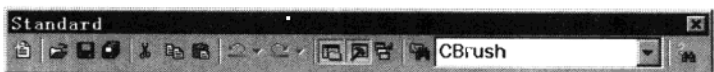


图 1-9 Visual C++ 6.0 的 Standard 工具栏



图 1-10 Visual C++ 6.0 的 Build 工具栏

Standard 工具栏中按钮和下拉列表框从左到右依次为：

- (1) New Text File: 创建新的文本文件。
- (2) Open: 打开已有的文档。
- (3) Save: 保存文档。
- (4) Save All: 保存所有打开的文件。
- (5) Cut: 剪切选定的内容到剪贴板。
- (6) Copy: 复制选定的内容到剪贴板。

- (7) Paste: 在当前插入点插入剪贴板中的内容。
- (8) Undo: 取消最后的操作。
- (9) Redo: 重复先前取消的操作。
- (10) Workspace: 显示或者隐藏工作区窗口。
- (11) Output: 显示或者隐藏输出窗口。
- (12) Window List: 管理当前打开的窗口。
- (13) Find in Files: 在多个文件中搜索字符串。
- (14) Find: 激活查找工具。
- (15) Search: 在指定的联机文档的子集中搜索字符串。

Build 工具栏中按钮和下拉列表框从左到右依次为:

- (1) Select Active Project: 选择活动项目。
- (2) Select Active Configuration: 选择活动配置。
- (3) Compile: 编译文件。
- (4) Build: 创建项目。
- (5) Stop Build: 停止创建项目。
- (6) Execute Program: 执行程序。
- (7) Go: 启动或者继续程序的执行。
- (8) Insert/Remove Breakpoint: 插入或删除断点。

如果要在屏幕上显示或者隐藏工具栏,请在屏幕工具栏区域单击鼠标右按钮,从工具栏快捷菜单选择或者清除相应的工具栏。

## 1.4 菜单栏

Visual C++ 6.0 的菜单栏由多个菜单组成。选择某项菜单后,系统将显示相应的下拉子菜单。在下拉式子菜单中,有些菜单选项的右边对应着相应的加速键。菜单中带有三个圆点符的菜单选项表示选择该项后,系统将自动弹出一个对话框。而带有黑三角箭头的菜单选项则表示选择该项后,系统将自动弹出级联菜单。若下拉式子菜单中的某些菜单选项显示为灰色,则表示这些选项在当前条件下不能选择。

注意,如果在窗口的不同位置上单击鼠标右键,系统将弹出快捷菜单,可以在快捷菜单中选择与当前位置最相关的命令。

### 1.4.1 “File” 菜单

“File” 菜单包含用于对文件进行操作的命令选项,如图 1-11 所示。

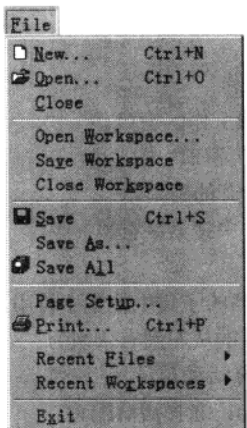


图 1-11 “File” 菜单

### 一、“New...” 菜单项

该菜单项用于打开“New”对话框，使用此菜单项可以创建新文档、新项目、新工作区或者其他文档。

#### (1) 创建新文件

如果要创建新文件，请从“New”对话框的“Files”选项卡中选择要创建的文件类型，然后在编辑框中输入文件的名称，如图 1-12 所示。如果要添加新文件到已有的项目中，请选择中“Add to project”复选框并选择项目名称。

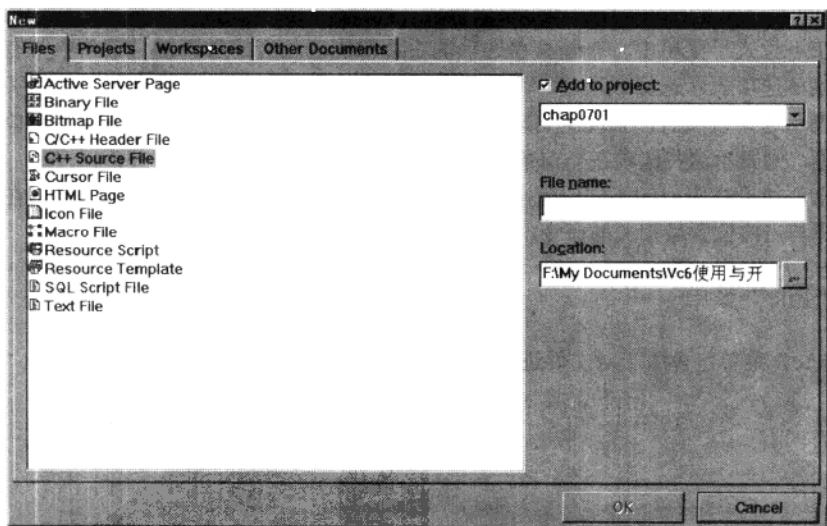


图 1-12 “New”对话框的“File”选项卡

Visual C++ 6.0 可以创建下述文件类型：活动服务器页（Active Server Page）、二进制文件（Binary File）、位图文件（Bitmap File）、C/C++ 头文件（C/C++ Header File）、C/C++ 源文件（C++ Source File）、光标文件（Cursor File）、HTML 文件（HTML Page）、图标文件（Icon File）、宏文件（Macro File）、资源脚本文件（Resource Script）、资源模板文件（Resource Template）、文本文件（Text File）和 SQL 脚本文件（SQL Script File）。

### （2）创建新项目

如果要创建新的项目，请从“New...”对话框的“Projects”选项卡中选择要创建的项目类型，在“Project name”文本框中输入项目名称即可，如图 1-13 所示。如果要向打开的工作区中添加新项目，请选择“Add to current workspace”选项，否则系统将自动创建包含新项目的新的工作区。如果要使新项目成为已有项目的子项目，请选中“Dependency of”复选框并指定项目名称。

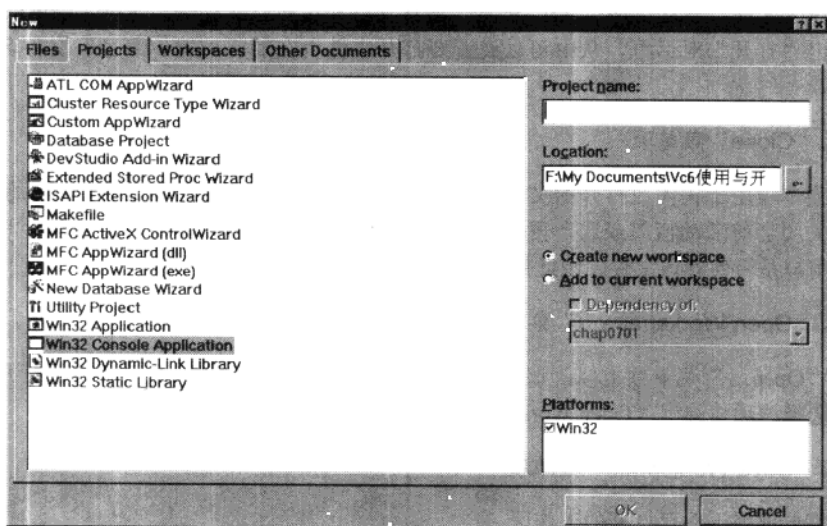


图 1-13 “New”对话框的“Projects”选项卡

Visual C++ 6.0 可以创建下述类型：ATL 应用程序（ATL COM AppWizard）、收集资源类型项目（Cluster Resource Type Wizard）、自定义 AppWizard（Custom AppWizard）、数据库项目（Database Project）、自动化宏（DevStudio Add-in Wizard）、扩展存储过程项目（Extended Stored Proc Wizard）、Internet 服务器或过滤器（ISAPI Extension Wizard）、Make 文件（Makefile）、ActiveX 控件程序（MFC ActiveX Control Wizard）、MFC 可执行程序（MFC AppWizard(exe)）、MFC 动态链接库（MFC AppWizard(dll)）、SQL 服务器数据库（New Database Wizard）、Skeleton 程序（Utility Project）、Win32 应用程序（Win32 Application）、Win32 动态链接库（Win32 Dynamic-Link Library）、Win32 控制台应用程序（Win32 Console Application）和 Win32 静态链接库（Win32 Static Library）。

### （3）创建新工作区



如果要创建新工作区，请从“New...”对话框的“Workspace”选项卡中选择工作区类型，然后在“Workspace name”文本框中输入工作区名称。

#### (4) 创建其他新文档

如果要创建其他新文档，请从“New...”对话框的“Other Documents”选项卡中选择要创建的文档类型，然后在“File name”文本框中输入文档名称。如果要向已有的项目中添加新文档，请选中“Add to project”复选框，然后选择项目名。

Visual C++ 6.0 可以创建的文档类型与安装的其他应用程序有关，例如“Microsoft Excel 工作表”、“Microsoft Excel 图表”、“Microsoft PowerPoint 演示文稿”、“Microsoft Word 文档”和“Visio 4 Drawing”等。

## 二、“Open...”菜单项

该菜单项用于打开已有的文件，如 C++ 文件、Web 文件、宏文件、资源文件、定义文件、工作区文件、项目文件、SQL 脚本、图形文件和可执行文件等。选择“Open...”菜单项将弹出“打开”对话框，从中可以设置要打开文件所在的驱动器、路径、文件名以及打开方式。

## 三、“Close”菜单项

该菜单项用于关闭已打开的文件。如果系统中包含多个已打开的文件，那么使用该菜单项将关闭当前活动窗口或选定窗口中的文件。如果某个文件在关闭时还未保存，则系统会提示用户是否保存该文件，因此用户不必担心数据会丢失。

## 四、“Open Workspace...”菜单项

与“Open...”菜单项类似，该菜单项也用于打开已有的文件，但主要用于打开工作区文件。选择该项将弹出“Open Workspace”对话框。

## 五、“Save Workspace”菜单项

该菜单项用于保存打开的工作区。

## 六、“Close Workspace”菜单项

该菜单项用于关闭打开的工作区。

## 七、“Save”菜单项

该菜单项用于保存活动窗口或当前选定窗口中的文件内容。如果该文件还未保存过，则系统将弹出“Save As”对话框，提示用户输入文件名。如果当前文件是以只读(Read-Only)方式打开的，那么与“Save”菜单项对应的操作按钮就会显示为灰色，表示该菜单项不可用。

## 八、“Save As...”菜单项

该菜单项的功能与“Save”菜单项类似，唯一的区别在于此菜单项以新文件名保存文