

新版软件步步高

基

础

本

Visual C++ 6.0

基础教程

捷新工作室 编著



2
1/2

国防工业出版社

77.12
C862
新版软件步步高(基础本)

Visual C++ 6.0 基础教程

捷新工作室 编著



国防工业出版社

·北京·

023908

图书在版编目(CIP)数据

Visual C++ 6.0 基础教程/捷新工作室编著. —北京:国防工业出版社, 1999. 10

(新版软件步步高·基础本)

ISBN 7-118-02125-3

I. V… II. 捷… III. C语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 45598 号

JS396/18

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码: 100044)

北京怀柔新华印刷厂印刷

新华书店经营

*

开本 787×1092 1/16 印张 17 1/4 399 千字

1999 年 10 月第 1 版 1999 年 10 月北京第 1 次印刷

印数: 1—4000 册 定价: 24.00 元

(本书如有印装错误, 我社负责调换)

总 序

在信息时代,知识成为推动社会生产力发展的一个最重要的因素,知识经济的轮廓在一些发达国家已经出现。以知识为基础的知识经济,其重要特点是信息产业的迅猛发展和产业的信息化,其内核是创新。我国是发展中国家,应该更加重视运用世界最新技术成果,有重点有选择地引进先进技术,增强自主创新能力,逐步实现技术发展的跨越。

在充分估量未来科学技术,特别是以计算机技术为先导的高技术发展对综合国力、社会经济结构和人民生活的巨大影响的基础上,为实现科教兴国战略多干实事,国防工业出版社组织了由数十位在计算机应用开发第一线工作的年富力强的博士、硕士组成的捷新工作室,编写出版《新版软件步步高(基础本)》和《新版软件步步高(提高本)》丛书。前者重在普及,后者追求提高,两者结合起来,力图满足多层次读者的需要。

《新版软件步步高(基础本)》的出版目的是普及新版软件的基本知识、基本操作技术,主要面向初学者,包括初次涉足该领域的机关、公司、企事业单位技术人员、大专院校师生及各类培训班学员,也可作为广大再就业职工理想的培训和学习教材。另外,对新技术感兴趣的读者也可将本丛书作为了解世界最新技术的窗口。

《新版软件步步高(提高本)》的出版目的是为有一定基础的读者找到提高专业技术水平和最新软件开发、操作技术的最佳途径,主要面向中高级读者,包括对该软件有一定基础知识,又希望提高自己专业技术水平的机关、公司、企事业单位技术人员,大专院校师生及各类高级培训班学员。

丛书的共同特点是突出一个“新”字,强调一个“精”字,力争一个“快”字。“新”是指软件的版本新;“精”是指精选的国内外流行最广、叫得最响的新版软件;“快”是指在保证质量的前提下,实现周期短,面市快。

丛书的内容覆盖最新高级语言开发环境(Visual J++ 6.0, Visual Basic 6.0, Visual FoxPro 6.0, Delphi 4.0),微机操作系统(中文 Windows 98, 中文 Windows NT 5.0),图形图像处理软件(Photoshop 5.0, 中文 CorelDRAW 8.0),Internet 浏览器(Internet Explorer 5.0),Web 页面设计环境(FrontPage 98),Internet 开发环境(Visual Inter Dev 6.0),大型数据库客户端开发工具(PowerBuilder 6.0)等。它们都是 1998 年以来推出的最新版软件。同时,我们还将把握新技术的脉搏,适时充实新的内容。

我们相信,丛书的出版必将为广大读者开辟跟踪、掌握、运用、进而创造新技术的最佳途径。

由于时间仓促,书中疏漏之处,敬请广大读者指正。

前 言

微软公司(Microsoft)的 Visual 系列开发工具,以它自己独到的特点和优势赢得了越来越多的开发者的厚爱,其编程的可视化是最容易为人们所感知的优点之一。Visual C++ 6.0中文版是 Microsoft 公司最新推出的 6.0 版 Visual 系列产品。

在微机版 C++ 程序设计环境中,微软公司的 Visual C++ 可以说是独领风骚。微软公司于 1998 年推出了 Microsoft Visual Studio 6(Microsoft 可视化工作室),Microsoft Visual C++ 6.0 是其中的一个组件。它汇集了微软公司的包括“智能感知”在内的技术精华,不仅全面地贯彻了面向对象技术,而且在编译优化技术方面较其他同类产品具有明显的优势。它是一个彻底的程序员级的开发环境,“可视化”的设计减少了不少编程的工作量。利用 Visual C++ 6.0 几乎可以完成任何设计功能,小至普通的应用系统,大至复杂的应用开发工具。

Microsoft Visual C++ 6.0 有专业版、企业版和学习版三个版本,每一个版本都具有不同目的和特点。其中,专业版用于开发基于 Win32 平台(包括 Windows 95 和 Windows NT)的应用程序、业务和控制;企业版用于开发和调试 Internet 或 Intranet 的客户/服务器应用程序,企业版的 Visual C++ 6.0 包括专业版的所有特性,并且还增加了使用 SQL 数据库和调试 SQL 存储过程的工具;学习版是用于学习 C++ 语言的专业学习工具,学习版的 Visual C++ 6.0 包括除了代码优化器、Profiler 和与 MFC 类库的静态连接以外的所有 Visual C++ 6.0 专业版的特性。学习版的价格比较低,适用于个人学习 C++ 语言。

本书是《新版软件步步高(基础本)》丛书之一,本书面向初中级读者,侧重对基本概念、基本技能和简单的应用程序设计技术的介绍。有兴趣的读者还可以参考本书的姊妹篇《新版软件步步高(高级本)》丛书的《Visual C++ 6.0 高级教程》一书。

目 录

第一章 Microsoft Visual C++ 6.0 概述	1
1.1 Visual C++ 6.0 软件包的内容	1
1.1.1 Visual C++ 6.0 的工具栏和菜单	3
1.1.2 集成环境窗口	5
1.1.3 工作空间窗口和输出窗口	7
1.2 使用联机帮助	9
第二章 Visual C++ 6.0 集成开发环境	11
2.1 File 菜单	11
2.2 Edit 菜单	14
2.3 View 菜单	16
2.4 Insert 菜单	17
2.5 Project 菜单	17
2.6 Build 菜单	18
2.7 Layout 菜单	19
2.8 Tools 菜单	19
2.9 Windows 和 Help 菜单	20
第三章 C++ 语言简介	21
3.1 操作符和表达式	21
3.1.1 算术操作符	21
3.1.2 算术表达式	22
3.1.3 赋值操作符	22
3.1.4 增量和减量操作符	23
3.1.5 sizeof 操作符	23
3.1.6 关系和逻辑操作符	24
3.1.7 位处理操作符	25
3.2 C++ 的流程控制语句	25
3.2.1 if 语句	25
3.2.2 if...else 语句	26
3.2.3 switch 语句	26
3.2.4 for 循环语句	28

3.2.5 do_while 循环语句	28
3.2.6 while 循环语句	29
3.3 常量和变量	29
3.3.1 常量定义	29
3.3.2 变量定义	31
3.4 函 数	32
3.4.1 函数原型的声明	32
3.4.2 函数的定义	32
3.4.3 函数的局部变量和静态变量	33
3.4.4 内联函数	33
3.4.5 函数重载	34
3.5 数组简介	36
3.5.1 一维数组	36
3.5.2 多维数组	37
3.6 字符串及其操作	39
3.6.1 字符串的输入	39
3.6.2 字符串的赋值	39
3.6.3 字符串处理函数	40
3.7 C++ 的类型操作	43
3.7.1 C++ 类型定义	43
3.7.2 枚举数据类型	43
3.7.3 结构	44
3.7.4 联合	45
3.8 指针	45
3.8.1 指向变量的指针	45
3.8.2 指向数组的指针	46
3.8.3 指向结构的指针	47
3.9 输入/输出操作	48
3.9.1 输出	48
3.9.2 输入	48
3.9.3 注释	49
3.9.4 应用实例	49
第四章 面向对象程序设计技术	50
4.1 面向过程与面向对象程序设计的区别	50
4.1.1 面向过程的程序设计	50
4.1.2 面向对象的程序设计	50
4.2 面向对象的基本概念	51
4.2.1 对象	51

4.2.2 类	52
4.2.3 面向对象方法的主要特征	53
4.3 类与对象的关系	53
4.3.1 定义类的方法	53
4.3.2 定义对象的方法	55
4.4 构造函数与析构函数	56
4.4.1 构造函数	56
4.4.2 析构函数	60
4.5 对象的多态性	61
4.5.1 静态联编和动态联编	61
4.5.2 虚函数	62
4.5.3 纯虚函数和抽象基类	64
4.6 类的静态成员	65
4.6.1 静态数据成员	65
4.6.2 静态成员函数	67
4.7 友员	68
4.7.1 友员函数	68
4.7.2 友员类	70
第五章 掌握设计 Windows 程序的方法	72
5.1 DOS 程序和 Windows 程序的区别	72
5.2 变量的命名约定	73
5.3 创建一个完整的 Windows 程序	74
5.4 分析 C++ Windows 程序	84
5.4.1 CFirstApp 对象	88
5.4.2 CMainFrame 对象	97
5.4.3 CchildFrame 对象	104
5.4.4 FirstDoc 对象	107
5.4.5 FirstView 对象	111
第六章 了解 MFC 类库	136
6.1 MFC 类库简介	136
6.2 MFC 类库分类	136
6.2.1 根类 CObject	136
6.2.2 应用程序体系结构类	136
6.2.3 可视对象类	137
6.2.4 绘图和打印类	139
6.2.5 通用类	139
6.2.6 数据库类	140

6.2.7 Internet 和网络工作类	141
6.2.8 OLE 类	141
第七章 设计一个简单的应用程序	142
7.1 程序功能介绍	142
7.2 生成工程文件	144
7.2.1 工程	144
7.2.2 工程工作区	144
7.2.3 生成工程文件	144
7.3 可视化编程阶段	145
7.3.1 用 AppWizard 工具生成程序框架	145
7.3.2 熟悉工程工作区	150
7.3.3 使用 ClassWizard 工具	152
7.4 代码编程阶段	155
7.4.1 设置 Say 按钮的属性	155
7.4.2 为 Say 按钮添加程序代码	157
7.4.3 设置 Exit 按钮的属性	158
7.4.4 为 Exit 按钮添加程序代码	159
第八章 用文本编辑器实现基本文档操作	160
8.1 进入文本编辑器	160
8.2 文档的外部操作	161
8.2.1 打开文档	162
8.2.2 维护 MRU 列表	162
8.2.3 文档视图	163
8.2.4 保存文档	164
8.2.5 打印文档	165
8.3 文档内部的编辑操作	167
8.3.1 在空白区内移动	167
8.3.2 识别括号对	168
8.3.3 书签	168
8.3.4 搜索字符串	170
8.3.5 替换字符串	171
8.3.6 在磁盘文件中搜索文本	171
8.3.7 搜索常规表达式	172
第九章 文本编辑器的高级操作技巧	174
9.1 Visual C++ 6.0 的辅助编辑功能	174
9.1.1 查阅成员表	174

9.1.2 查阅参数信息	176
9.1.3 了解输入信息	176
9.2 其他辅助编辑工具	177
9.2.1 定制自己喜爱的编辑器	177
9.2.2 使用其它文本编辑器	178
9.2.3 管理组合键	180
9.2.4 使用高级编辑功能	183
9.2.5 使用宏	184
第十章 单文档界面程序设计	186
10.1 基本概念	186
10.2 应用程序功能介绍	187
10.3 生成工程文件框架	189
10.4 应用程序的可视化编程阶段	193
10.4.1 设计应用程序的主窗口界面	193
10.4.2 设计菜单系统	195
10.4.3 为对话框中的控件连接变量	195
10.5 应用程序的代码编程阶段	196
10.5.1 声明文档类中的数据成员	196
10.5.2 初始化文档类的数据成员	198
10.5.3 初始化视类的数据成员	199
10.5.4 连接文档类和视图类的数据成员	200
10.5.5 实现文件数据的输入输出操作	203
10.5.6 进一步完善程序	204
10.6 运行程序	206
10.7 扩充程序功能	206
10.7.1 对扩充功能的说明	206
10.7.2 程序的可视化编程阶段	207
10.7.3 程序的代码编程阶段	208
10.8 运行并测试增强功能后的应用程序	227
第十一章 多文档界面应用程序设计	228
11.1 应用程序功能介绍	228
11.2 生成工程文件框架	233
11.3 应用程序的可视化编程阶段	238
11.3.1 设计应用程序主窗口的界面	238
11.3.2 设计菜单系统	239
11.4 应用程序的代码编程阶段	241
11.4.1 声明视图类的数据成员	241

11.4.2 定义文档类中的数据成员.....	242
11.4.3 初始化视图类的数据成员.....	243
11.4.4 实现文件数据的输入输出操作.....	244
11.4.5 实现一个文档的多个视图.....	245
11.5 扩充程序的功能.....	249
11.5.1 增加分割条.....	249
11.5.2 在菜单中加入 split 选项	251
11.6 运行并测试增强功能后的应用程序.....	251
附录 A MFC 类的成员函数及其对应的函数功能	252
附录 B 安装 Microsoft Visual C++ 6.0	263

第一章 Microsoft Visual C++ 6.0 概述

与过去的面向过程的程序设计语言比较, Visual C++ 的最大特征在于它是一种面向对象(OOP, Object Oriented Programming)的程序设计语言。随着计算机软件技术的普及和发展,大量的程序设计工作都是在微机上完成的。因此,功能强大的微机版本 C++ 程序设计环境成为专业程序员的必修课。

在微机版 C++ 程序设计环境中,微软公司的 Visual C++ 可以说是独领风骚。微软公司于 1998 年推出了 Microsoft Visual Studio 6(Microsoft 可视化工作室),而 Microsoft Visual C++ 6.0 是其中的一个组件。它汇集了微软公司的技术精华,不仅全面地贯彻了面向对象技术,而且在编译优化技术方面较其他同类产品具有明显的优势。它是一个彻底的程序员级的开发环境,“可视化”的设计减少了不少编程的工作量。利用 Visual C++ 6.0 几乎可以完成任何设计功能,小至普通的应用系统,大至复杂的应用开发工具。

Microsoft Visual C++ 6.0 有三个版本:专业版、企业版和学习版,每一个版本都具有不同目的和特点。

(1) 专业版:用于开发基于 Win32 平台(包括 Windows 95 和 Windows NT)的应用程序、业务和控制。

(2) 企业版:用于开发和调试 Internet 或 Intranet 的客户/服务器应用程序。企业版的 Visual C++ 6.0 包括专业版的所有特性,并且还增加了使用 SQL 数据库和调试 SQL 存储过程的工具。

(3) 学习版:使用 Visual C++ 的专业工具学习 C++ 语言。学习版的 Visual C++ 6.0 包括除了代码优化器、Profiler 和与 MFC 类库的静态连接外的所有 Visual C++ 6.0 专业版的特性。学习版的价格比较低,适用于个人学习 C++ 语言。注意,禁止使用它开发用于发行的软件。

Visual C++ 6.0 开发工具的功能十分强大,本章将就其安装、软件包组件和联机帮助系统等内容作详细介绍。

1.1 Visual C++ 6.0 软件包的内容

Visual C++ 6.0 软件包包含了许多单独的组件,如编辑器、编译器、链接器、应用程序生成向导、调试器以及多种为开发 Microsoft Windows 下 C/C++ 应用程序而设计的工具。

Visual C++ 6.0 环境为帮助程序员开发和设计应用程序而提供的强大功能可概括为以下几点:

(1) 通过菜单和工具栏对命令进行合理和方便的访问。读者可以定制 Visual C++

6.0 中已有的菜单或创建新的菜单。

(2) 对不同的开发进程,从类和源文件的列表到编译器消息,提供了不同的观察窗口。

(3) 创建和维护源文件的文本编辑器,设计对话框用的优秀对话框编辑器,并创建其他界面组件,如位图、图标、鼠标及工具栏的图形编辑器。

(4) 为程序创建启动器文件的向导,在建立新项目的常规任务上提供一个良好的开端。Visual C++ 6.0 为不同类型的 Windows 程序提供向导,包括带有可选数据库和 Automation 支持的标准应用程序、动态链接库、基于对话框的应用程序、使用 Internet Server API(ISAPI)的 Web 服务器的扩展程序及 ActiveX 控件。

(5) 帮助 MFC 应用程序创建和维护类的帮手——Class Wizard。

(6) Gallery 维护的内置可执行组件给程序增加方便的特征。

(7) 优秀的调试器。

(8) 通过宏和附加的动态链接库来添加自己的环境工具的能力。读者可以自己开发这些附加项,或从各种各样的供应商那里购买它们。

(9) 访问联机帮助的扩展系统的菜单。

图 1.1 显示的是一个典型的 Visual C++ 6.0 主窗口,其开发环境外观与以前版本相比只有细微的改变,它的风格和许多命令仍维持不变。如果读者以前从未用过 Visual C++, 会发现,像任何大的 Windows 程序一样,它需要一些时间来习惯。它的界面是智能化的,而且非常宽容,便于鼓励读者去实践和尝试。

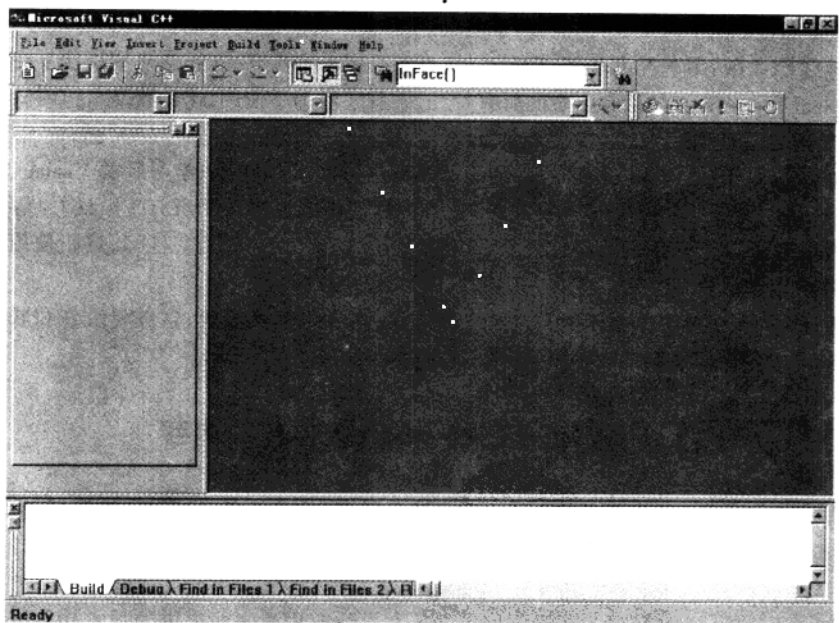


图 1.1 Visual C++ 6.0 主窗口

本节介绍 Visual C++ 6.0 的环境外壳,并描述开发项目时遇到的界面和窗口。在这里,我们集中讨论这个环境是如何应用到 Visual C++ 6.0 和 C/C++ 项目上的。

1.1.1 Visual C++ 6.0 的工具栏和菜单

Visual C++ 6.0 带有一个预先定义好的工具栏集,单击它们便可以访问它们。如果找不到需要的工具,可以用自己设计的定制工具栏来增大工具栏集。每个工具栏都由工具栏的标题栏上的名字标识,如图 1.2 所示。

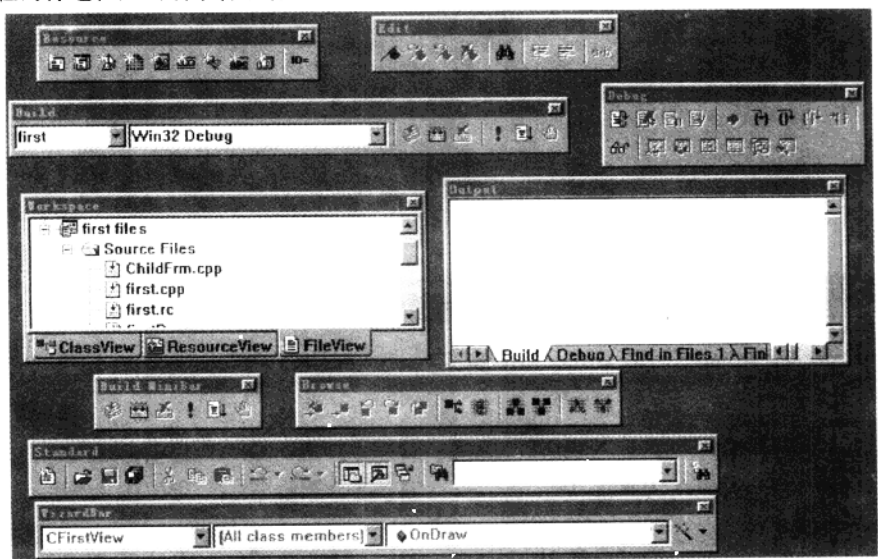


图 1.2 Visual C++ 6.0 工具栏集

正如下面部分所描述的那样,工具栏经常被定位在某个位置上,这时标题栏消失。例如,一般情况下,图 1.1 显示的 Standard、Build 和 Edit 工具栏定位在 Visual C++ 6.0 主窗口的顶端。工具栏在主窗口的位置由读者决定。读者可以在屏幕四周移动工具栏,通过拖曳边框来调整它们的矩形形状,并使任何一套工具可见或隐藏。读者可能愿意让某些工具栏(如 Standard 和 Build)一直都是可见的,至于其他工具栏,通常只有当读者需要工作在它们的窗口中时,才变得可见。例如,在默认设置中,Debug 工具栏只在调试过程中才是可见的,Colors 和 Graphics 工具栏仅在图形编辑器中才是可见的,因为这是读者需要它们的唯一地方。

图 1.3 显示了 Customize(自定义)对话框中包含的工具栏名称列表,在这里可以通过单击复选框来调整工具栏的可见性是打开或是关闭。在 Tools(工具)菜单中单击 Customize(自定义)命令即可打开这个对话框。

当鼠标停留在工具栏按钮上面时,按钮凸起。主窗口的底端的状态栏显示了对按钮的简短描述,并且,如果光标停留时间长一些,就会出现一个小的弹出式的“工具提示”窗口,它包含了按钮的名字。如果需要的话,Visual C++ 6.0 甚至可以显示放大的工具栏,如图 1.4 所示。

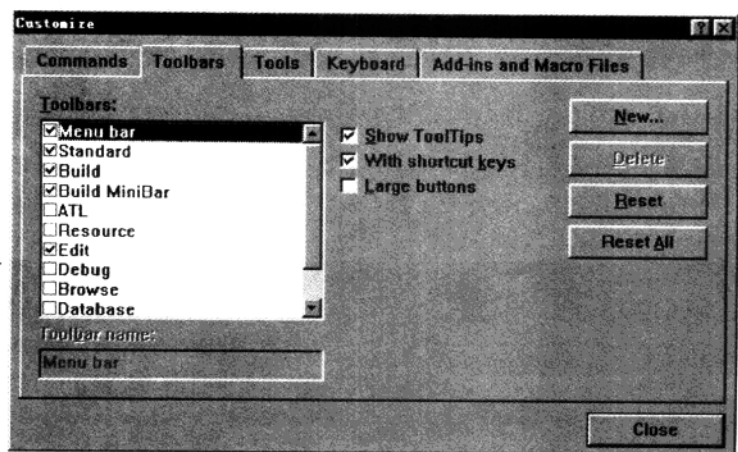


图 1.3 在 Customize(自定义)对话框中打开和关闭工具栏

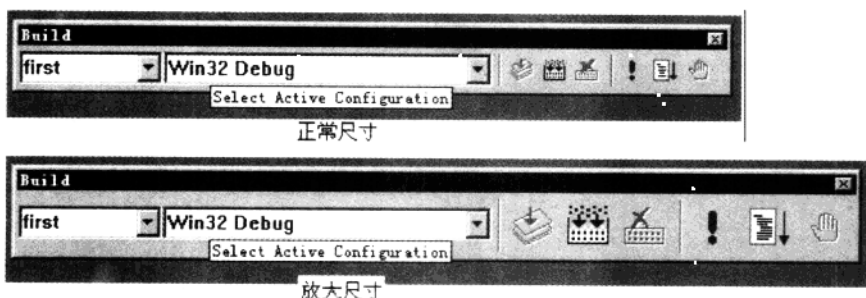


图 1.4 工具栏尺寸调节

工具提示和放大的选项都是在图 1.3 显示的 Customize(自定义)对话框中控制的。Visual C++ 6.0 菜单栏是一种特殊形式的工具栏,因为在全屏模式下菜单栏会隐藏起来,其他情况下,它就像一个普通的工具栏。当鼠标停留在 Visual C++ 6.0 的菜单栏上时,菜单名像工具栏一样呈凸起状。单击菜单名拉下菜单时,菜单名看起来像是凹进屏幕里。菜单打开后,把鼠标从一个菜单名滑动到另一个会拉下另一个菜单。

除了上面提到的菜单栏外,读者还可能碰到上下文相关菜单。Visual C++ 6.0 环境几乎都能响应鼠标右键的单击,通常是显示一个弹出式的带有与位置相适应命令的上下文相关菜单。甚至当 Visual C++ 6.0 没有打开窗口时,在空白区右击也会产生一个菜单,其中含有使窗口可见和调整工具栏开或关的命令。在工具栏上除标题栏外的任何地方右击,可打开同样的菜单。只要试一试右键,就会发现许多其他的快捷方式。

通过在工具栏或菜单栏上而不是按钮或菜单名的地方单击,并保持鼠标被按下,就可以把它们拖动到屏幕上新地方。如果工具栏的标题栏不可见,工具栏上的垂直分隔处正是“抓取”工具栏拖动的好地方。

1.1.2 集成环境窗口

除许多对话框外, Visual C++ 6.0 显示两种类型的窗口, 即文档窗口和停靠(Docking)窗口。文档窗口是一般的带边框子窗口, 其中包含有源代码文本和图形文档。Window(窗口)菜单中列出了在屏幕上以平铺方式还是以层叠方式显示文档窗口的命令。所有其他的 Visual C++ 6.0 窗口, 包括工具栏甚至菜单栏, 都是停靠的。开发环境有两个主要的停靠窗口——Workspace(工作空间)窗口和 Output(输出)窗口, 它们通过 View(查看)菜单中的命令变成可见的窗口。另一个停靠窗口是 Debugger(调试器)窗口, 它在调试过程中显示。在此首先看看所有停靠窗口的一些共同特征, 然后单独地讨论 Workspace(工作空间)窗口和 Output(输出)窗口。

停靠窗口可以固定在 Visual C++ 6.0 用户区的顶端、底端或侧面, 或者浮动在屏幕上任何地方。停靠窗口, 不论是浮动着的或是固定着的, 总是出现在文档窗口的上面。这样就保证了当焦点从一个窗口移到另一个时, 浮动的工具栏一直都是可见的, 但这也意味着, 文档窗口偶尔会看起来像消失了似的, 不过请放心, 文档窗口仍在那儿。这种情况刚开始时可能会让人觉得不习惯, 例如, 如果读者正在文本编辑器中编辑源代码, 此时打开一个占据整个 Visual C++ 6.0 用户区的停靠窗口, 源代码文档就会消失, 它隐藏在新窗口之下。如果覆盖的窗口被定位在该位置上, 读者就无法使文档窗口回到上面来。唯一的解决方法是要么关了覆盖的窗口, 要么把它拖到不挡眼的地方去。我们将看到如何快速地打开和关闭停靠窗口。

拖动一个停靠窗口时会出现一个移动的轮廓, 它显示了释放鼠标左键时窗口的新位置。这些轮廓线是灰色虚线, 直到它与环境用户区的边界或另一个停靠窗口的边界接触时才变成细黑线。这种变化提示读者, 若放下窗口, 会使它定位在边界附近的位置。工具栏在用户区的顶端或底端时定位为水平位置, 而在左右边界时则定位为垂直位置。在拖动工具栏时, 可通过按 Shift 键来改变工具栏的放置方向。

把窗口定位在希望的位置和尺寸上, 有时会花费一些时间。要使窗口占据整个用户区, 应向上拉动边界, 直到鼠标同用户区的上边界接触, 然后释放鼠标按键。要把窗口恢复到较小尺寸, 应拖动窗口, 直到光标接触到用户区的左边界。这迫使窗口成为浮动的, 可以通过其标题栏来把它拖放到其他地方。

在屏幕上移动一个停靠窗口时, 窗口看起来像是有主意似的, 顽强地紧贴着 Visual C++ 6.0 主窗口的某一边界, 或任何其他它接触到的定位窗口。有两个办法可防止这个问题发生。第一种办法就是在移动窗口时按住 Ctrl 键, 来暂时禁止它的停靠特征。第二种办法只对窗口有效, 对工具栏无效, 那就是禁止窗口的停靠能力, 直到再次使它生效。在窗口内部右击, 从上下文相关菜单中, 选择 Docking View(停靠视图)命令来关掉命令的复选标志。Window(窗口)菜单也提供对 Docking View(停靠视图)命令的访问, 如图 1.5 所示。

若关掉窗口的停靠(Docking)模式, 则窗口会有以下几个特征:

- (1) 窗口就像一个普通的文档窗口, 标题栏上有最小化、最大化和关闭窗口按钮。
- (2) 从 Window(窗口)菜单中选择 Cascade(层叠)或 Tile(平铺)命令时, 窗口同其他打开的文档窗口一起排列位置。

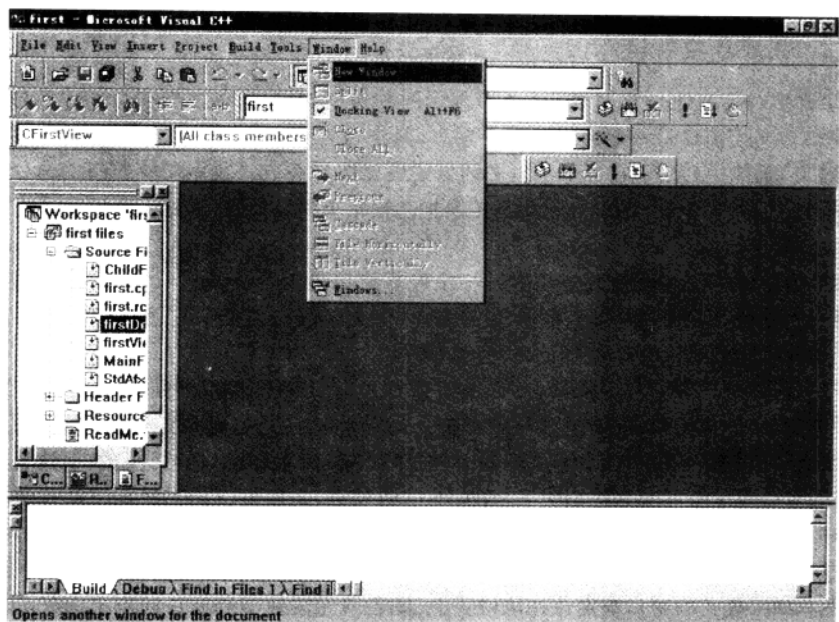


图 1.5 使用 Docking View(停靠视图)命令切换窗口的停靠模式

(3) 窗口不能移到 Visual C++ 6.0 主窗口用户区的外面,但当它处在停靠(Docking)模式下时是可以的。

(4) 输入焦点移到该窗口上时,它可以由 Window(窗口)菜单上的 Close(关闭)命令关闭。否则,Close(关闭)命令不能影响停靠模式下的窗口,即使它拥有焦点。窗口或工具栏被定位后,凸起的卷轴出现在窗口的顶端或左边界,如图 1.6 所示。双击卷轴使窗口变成浮动的,双击浮动窗口或工具栏的标题栏,使它回到原来的位置,也可以通过卷轴来把窗口拖动到另一个固定的或浮动的位置上。

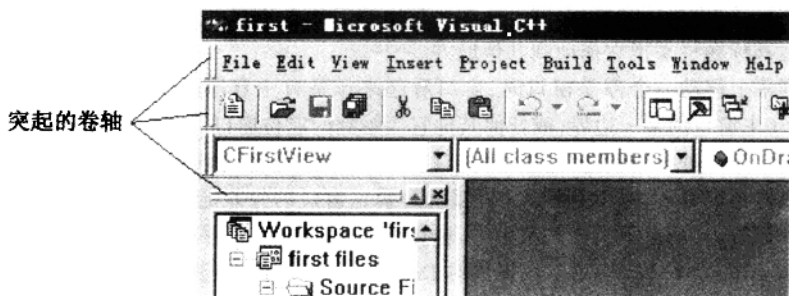


图 1.6 在停靠时,窗口、工具栏和菜单栏中出现一个凸起的卷轴

在 Visual C++ 6.0 中创建的窗口排列,将在项目的整个过程中维持,或直到读者改变窗口排列方式。下一次打开项目时,窗口像上次离开时的样子排列。然而,在环境中运行的实用程序的窗口并不遵守这些规则。这些窗口既不是文档窗口,也不是停靠窗口,它