

高职高专计算机系列规划教材

# Visual C++ 6.0 简明教程

赵冬玲 主编

田银磊 王仲英 副主编

朱乃立 主审

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书介绍 Visual C++的编程环境、编程基础(C++语言基础, Windows 编程, MFC 基础)以及常用控件的功能和用法;讲述如何给一个基于 MFC 创建的应用程序添加控件,如何改善用户界面(使用编辑器创建菜单、工具栏和状态栏)和创建新的用户界面;如何在文档/视图结构中添加合适的视图;如何创建和使用 ActiveX 控件以及数据库编程和多媒体应用等知识。书中给出了大量实例,并有习题辅导和上机操作指导,便于读者学习。

本书可作为高等院校、高等职业教育院校相应课程的教材,也可供读者自学使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Visual C++ 6.0 简明教程 / 赵冬玲主编. —北京:电子工业出版社, 2004.7  
(高职高专计算机系列规划教材)  
ISBN 7-5053-9997-7

.V... .赵... .C 语言 - 程序设计 - 高等学校:技术学校 - 教材 .TP312

中国版本图书馆 CIP 数据核字(2004)第 066931 号

责任编辑:王沈平 特约编辑:王宝祥

印 刷: 厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:16 字数:409 千字

印 次:2004 年 7 月第 1 次印刷

印 数:5000 册 定价:19.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

## 前 言

Visual C++是微软公司推出的优秀软件开发工具,它具有高效完备的 Windows 编程开发环境,将程序和资源的编辑、编译、调试和运行融为一体,并且提供了大量的程序开发工具,如应用程序开发向导 AppWizard、类向导 ClassWizard 等。这些工具都被高度集成在一个名为 Developer Studio 的开发平台中,利用该开发平台,用户可以自动生成应用程序框架,编写出各种各样的软件。从最简单的基于对话的程序到复杂的多用户程序,都可以用 Visual C++方便、快捷地进行开发。

在目前出版的 Visual C++的书籍中,一般性地介绍 MFC 类库的使用和面向对象编程知识的较多,而既有理论基础,又有开发手段,强调“理论够用、注重实践”的书较少。本书通过以通俗易懂的语言,先介绍每一章基本知识点(理论够用),接着举出一到多个综合实例,在实例中给出知识点的详细解释,从而帮助读者理解知识点、会用知识点。尤其重要的是,这些综合实例或承前,或启后,甚至进行一些必要的引申,从而将全书的内容贯穿起来,符合案例化教学的特点。相信读者通过对本书的学习,尤其是通过对本书实例的一步一步学习和分析,就可以掌握 Visual C++编程的基本方法,具备一定的开发能力。因此,我们强烈建议读者通过实例对本书进行学习。

本书是作者在参考大量的 VC 书籍与 VC 共享源代码的基础上,结合多年从事 VC 程序开发和教学的经验,针对高职教育特点,按照教学大纲要求编写而成的。本书由赵冬玲主编,王仲英和田银磊担任副主编。第 1 章、第 2 章及实验一、实验二由赵冬玲编写,第 3 章及实验三、实验四由李远杰编写,第 4 章、第 8 章及实验五由田银磊编写,第 6 章、第 7 章及实验七、实验八由王仲英编写,第 5 章及实验六由刘秋菊编写。另外,燕利霞、杨艳、梁国兰、卫艳荣和王东霞等对本书的编写提供了帮助,洛阳大学朱乃立教授担任本书的主审。在本书的写作过程中,还得到郝小会的大力支持,在此一并表示衷心的感谢。

由于作者水平有限,时间仓促,书中错误在所难免,恳请广大读者给予批评指正。

编 者

2004 年 3 月

# 目 录

第 1 章 Visual C++ 6.0 概述 .....	( 1 )
1.1 Visual C++ 6.0 的发展 .....	( 1 )
1.2 Visual C++ 6.0 的特点和新特性 .....	( 1 )
1.3 Visual C++ 6.0 开发环境 .....	( 2 )
1.3.1 主窗口组成 .....	( 2 )
1.3.2 菜单功能 .....	( 3 )
1.3.3 工具栏 .....	( 7 )
1.4 项目和项目工作区 .....	( 9 )
1.4.1 项目 .....	( 9 )
1.4.2 项目工作区 .....	( 10 )
1.5 创建应用程序 .....	( 12 )
1.5.1 用 AppWizard 创建项目 .....	( 12 )
1.5.2 添加程序代码 .....	( 14 )
1.5.3 编译运行 .....	( 15 )
本章小结 .....	( 15 )
习题 .....	( 15 )
第 2 章 Visual C++ 编程基础 .....	( 16 )
2.1 C++语言基础 .....	( 16 )
2.1.1 简单的 C++程序 .....	( 16 )
2.1.2 C++基本数据类型 .....	( 16 )
2.1.3 运算符和表达式 .....	( 19 )
2.1.4 C++基本语句 .....	( 21 )
2.1.5 自定义类型 .....	( 26 )
2.1.6 指针 .....	( 30 )
2.1.7 函数 .....	( 33 )
2.1.8 类与对象 .....	( 42 )
2.1.9 类的继承和派生 .....	( 47 )
2.1.10 虚函数和多态性 .....	( 50 )
2.2 Windows 编程基础 .....	( 51 )
2.3 MFC 基础 .....	( 53 )
2.3.1 MFC 特点 .....	( 53 )
2.3.2 MFC 类库简介 .....	( 53 )
2.3.3 MFC 应用程序框架结构分析 .....	( 55 )
2.3.4 MFC 消息映射及处理 .....	( 56 )
本章小结 .....	( 56 )
习题 .....	( 56 )
第 3 章 对话框及常用控件 .....	( 58 )

3.1	对话框的组成与分类	( 58 )
3.1.1	对话框的组成	( 58 )
3.1.2	对话框的分类	( 58 )
3.2	在应用程序中添加对话框	( 59 )
3.3	通用对话框和消息对话框	( 64 )
3.3.1	创建通用对话框	( 64 )
3.3.2	消息对话框	( 65 )
3.4	Windows 常用控件	( 66 )
3.4.1	静态控件和按钮	( 66 )
3.4.2	编辑框	( 68 )
3.4.3	列表框	( 73 )
3.4.4	组合框	( 75 )
3.4.5	滚动条	( 78 )
3.4.6	旋转按钮控件和滑动条	( 80 )
	本章小结	( 86 )
	习题	( 86 )
第 4 章	设计用户界面	( 87 )
4.1	菜单	( 87 )
4.1.1	用编辑器设计菜单	( 87 )
4.1.2	菜单消息的处理机制	( 90 )
4.1.3	为菜单项添加成员函数 ( 消息处理函数 ) 以及相应的代码	( 91 )
4.1.4	动态菜单	( 95 )
4.1.5	快捷菜单	( 106 )
4.2	工具栏	( 109 )
4.2.1	工具栏简介	( 109 )
4.2.2	创建工具栏	( 109 )
4.2.3	工具栏实例	( 111 )
4.3	状态栏	( 113 )
4.3.1	状态栏简介	( 113 )
4.3.2	创建状态栏	( 114 )
4.3.3	状态栏类 CStatusBar	( 114 )
4.3.4	状态栏实例	( 115 )
	本章小结	( 119 )
	习题	( 119 )
第 5 章	框架窗口、文档和视图	( 120 )
5.1	框架窗口	( 120 )
5.1.1	框架窗口简介	( 120 )
5.1.2	框架窗口与视图窗口	( 120 )
5.1.3	框架窗口与文档边框窗口	( 120 )
5.2	文档与视图	( 121 )

5.2.1	文档类	(121)
5.2.2	视图类	(122)
5.2.3	文档与视图的关系	(124)
5.2.4	多文档	(125)
5.2.5	多视图	(125)
5.2.6	分割窗口	(126)
5.2.7	文档和视图的初始化及清理	(126)
5.3	SDI 应用程序文档和视图的实现	(126)
5.4	MDI 应用程序文档和视图的实现	(131)
5.4.1	创建 MDI 应用程序文档和视图	(131)
5.4.2	完善 MDI 应用程序文档和视图	(132)
5.5	文档和视图实现的综合实例	(133)
	本章小结	(144)
	习题	(145)
第 6 章	ActiveX 控件	(146)
6.1	ActiveX 简介	(146)
6.2	ActiveX 控件的属性、方法和事件	(146)
6.2.1	ActiveX 控件的属性	(146)
6.2.2	ActiveX 控件的方法	(147)
6.2.3	ActiveX 控件的事件	(147)
6.3	创建 ActiveX 控件	(147)
6.4	创建包含 ActiveX 控件的 MFC 应用程序实例	(151)
6.4.1	定制界面	(151)
6.4.2	添加库存属性	(152)
6.4.3	添加自定义属性	(154)
6.4.4	添加库存方法与添加自定义方法	(155)
6.4.5	添加库存事件与添加自定义事件	(156)
6.4.6	测试 ActiveX 控件	(157)
	本章小结	(158)
	习题	(158)
第 7 章	数据库编程	(159)
7.1	数据库概述	(159)
7.2	ODBC	(159)
7.2.1	ODBC 概述	(159)
7.2.2	Visual C++ 中的 ODBC 类	(160)
7.3	创建 ODBC 数据库的应用程序	(163)
7.4	DAO 与 ODBC	(178)
7.5	ADO 概述	(179)
	本章小结	(179)
	习题	(180)

. . .

第 8 章 多媒体应用	( 181 )
8.1 图形的绘制	( 181 )
8.1.1 基本概念	( 181 )
8.1.2 MFC 的图形设备接口类和设备环境类	( 182 )
8.1.3 字体与文字处理	( 192 )
8.1.4 位图	( 193 )
8.2 多媒体程序开发方法	( 196 )
8.2.1 常见多媒体文件格式	( 196 )
8.2.2 多媒体文件的播放	( 196 )
8.2.3 媒体控制接口的使用	( 197 )
8.2.4 使用 MCIWnd 窗口类	( 201 )
本章小结	( 205 )
习题	( 205 )
实验一 Visual C++开发环境	( 206 )
实验目的	( 206 )
实验内容	( 206 )
实验二 Visual C++编程基础	( 209 )
实验目的	( 209 )
实验内容	( 209 )
实验三 常用控件 ( 一 )	( 213 )
实验目的	( 213 )
实验内容	( 213 )
实验四 常用控件 ( 二 )	( 218 )
实验目的	( 218 )
实验内容	( 218 )
实验五 菜单、工具栏和状态栏	( 224 )
实验目的	( 224 )
实验内容	( 224 )
思考与分析	( 224 )
实验六 框架窗口、文档与视图	( 225 )
实验目的	( 225 )
实验内容	( 225 )
实验七 ActiveX 控件	( 228 )
实验目的	( 228 )
实验内容	( 228 )
实验八 ODBC 访问数据库	( 231 )
实验目的	( 231 )
实验内容	( 231 )
附录 A 操作符的优先级和结合性	( 235 )
附录 B MFC 类	( 236 )
参考文献	( 245 )

# 第 1 章 Visual C++ 6.0 概述

Visual C++是 Microsoft 公司在 Microsoft C/C++的基础上开发的基于 Windows 平台的可视化编程环境。“可视化”指的是一种开发图形用户界面 (GUI) 的方法,所以 Visual C++是基于 C++的可视化的程序设计语言。本章将介绍 Visual C++的发展和技术特点,并通过实例介绍应用程序的建立、编译和运行,使用户很快认识 Visual C++。

## 1.1 Visual C++ 6.0 的发展

虽然 Microsoft 公司推出的 Windows 支持多种应用程序编程接口 (Application Programming Interface, 简称 API),但是当开发人员把 C/C++程序直接用于 Windows 提供的一系列 API 函数上时,要耗费大量的精力和时间学习 Windows 编程方面的详细知识。为此,Microsoft C/C++ 7.0 给出了 C++ API,即微软基础类库(Microsoft Foundation Class, 简称 MFC) MFC1.0,目的是简化在 Windows 下的编程过程,为程序员提供面向对象的 Windows 编程接口。此后,Microsoft 公司对 MFC1.0 进行了扩充,增加了许多功能,即 MFC2.0 库。更重要的是提供了一个基于 Windows 的调试器,而且它所提供的 Windows 风格的编辑器也很受欢迎。至此,Visual C++诞生了。

最新版本的 Visual C++ 6.0 是在以往版本不断更新的基础上形成的,由于其功能强大、灵活性好、完全可扩展,以及具有强有力的 Internet 的支持,已成为目前最为流行的 C++语言集成开发环境。

## 1.2 Visual C++ 6.0 的特点和新特性

### 1. Visual C++ 6.0 继承了 Visual C++版本的特点

(1) 充分利用了 MFC 类库的代码。采用 MFC 开发的 Windows 程序之所以能够大幅度提高开发速度和效率,主要是在 MFC 类层次封装了大量 Windows 函数和 Windows 应用的默认处理。这样,软件开发人员只需要编写其特定的代码实现自己的开发任务即可,而一些通用的代码不用自己编写,系统已经自动生成。

(2) 多种工具的支持。在 MFC 的基础上再配合提供给 Visual C++的 AppWizard 和 ClassWizard 等工具可以大大加快开发进程。

(3) 具有大量的基类可供程序员使用。例如,CWinApp 类、CFrameWnd 类、CMDIFrameWnd 类、CMDIChildWnd 类、CDocument 类和 CDC 类。通过从这些基类中派生出用户自己的类,然后重载几个虚函数就可以生成一个独立的应用程序。

(4) 产生了标准化的结构。因而使得采用 MFC 编写的程序在不同平台上的移植变得非常容易。

## 2 . Visual C++ 6.0 的新特性

(1) 最快的集成数据库访问。Visual C++ 6.0 允许用户建立强有力的数据库应用程序，可以使用 Windows 平台提供的 ODBC 类和高性能的 32 位 ODBC 驱动程序来访问各种数据库管理系统，也可以使用 DAO (数据访问对象) 类通过编程语言来访问和操作数据库中的数据并管理数据库、数据库对象与结构。

(2) 提供了对 Internet 强有力的支持。Visual C++ 6.0 是一种 32 位的编程环境。Win32 Internet API (WinInet) 可使 Internet 成为应用程序的一部分，并简化了对 Internet 服务 (如 FTP, HTTP, Gopher) 的访问。ActiveX 控件可以用在 Internet 和桌面应用程序中，其文档可以显示在整个 Web 浏览器中。另外，在 Visual C++ 6.0 的编程环境中，可以使用有关的类来创建动态链接库，以便添加功能到 Internet 服务器和 Web 页中。

这些新特点使基于 Visual C++ 6.0 的开发程序更完善地与网络、数据库等资源相结合，所以 Visual C++ 6.0 成为软件开发的理想工具。Visual C++ 6.0 分为标准版、专业版和企业版 3 种，其基本功能相同。

## 1.3 Visual C++ 6.0 开发环境

单击“开始”“程序”“Microsoft Visual Studio 6.0”“Microsoft Visual C++ 6.0”，即可启动 Visual C++ 6.0，进入 Visual C++ 6.0 的开发环境。

### 1.3.1 主窗口组成

Visual C++ 6.0 开发环境界面有标题栏、菜单栏、工具栏、项目工作区窗口、文档窗口、输出窗口以及状态栏等。标题栏、菜单栏、工具栏的组成与 Microsoft 的其他软件大同小异，如图 1.1 所示。项目工作区窗口包含用户项目的一些信息，包括类、项目文件以及资源等。在项目工作区窗口中的任何标题或图标处单击鼠标右键，都会弹出快捷菜单，其中包含当前状态下的一些常用操作。

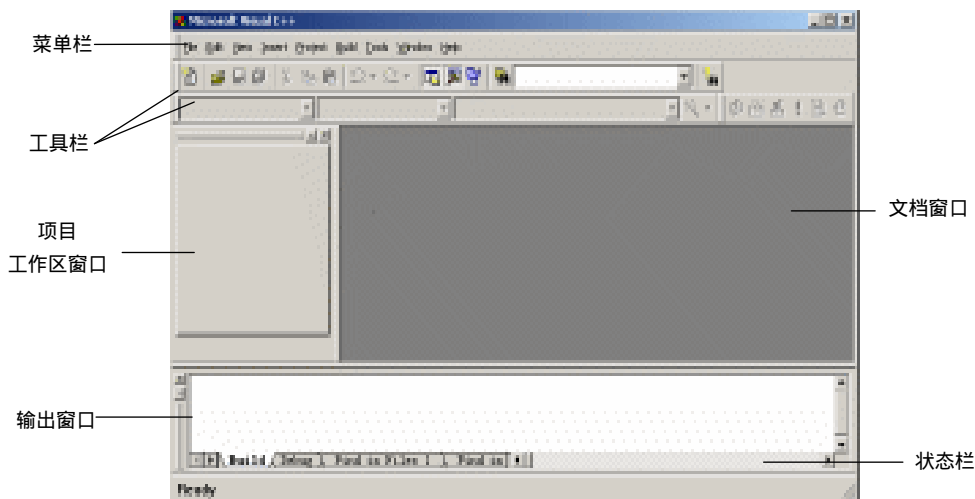


图 1.1 Visual C++ 6.0 开发环境界面

## 1.3.2 菜单功能

### 1 . File 菜单

File 菜单如图 1.2 所示，File 菜单中的各项命令主要用来对文件和项目进行操作，如“新建”、“打开”、“保存”、“打印”等。

- ? New 创建一个项目或文件，快捷键 Ctrl+N；
- ? Open 打开已有的文件，如 C++源文件，快捷键 Ctrl+O；
- ? Close 关闭当前被打开的文件；
- ? Open Workspace 打开一个已有的项目；
- ? Save Workspace 保存当前项目；
- ? Close Workspace 关闭当前项目；
- ? Save 保存当前文件，快捷键 Ctrl+S；
- ? Save As 将当前文件用新文件名保存；
- ? Save All 保存所有打开的文件；
- ? Page Setup 文件打印的页面设置；
- ? Print 打印当前文件内容或选定内容，快捷键 Ctrl+P；
- ? Recent Files 包含最近打开的文件；
- ? Recent Workspace 包含最近打开的项目。

### 2 . Edit 菜单

Edit 菜单如图 1.3 所示，主要是用来编辑文件内容。例如，用于撤销和恢复操作的 Undo 和 Redo 命令，用于复制、剪切、粘贴操作的 Copy、Cut 和 Paste 命令。此外，菜单中还有如下常用命令：

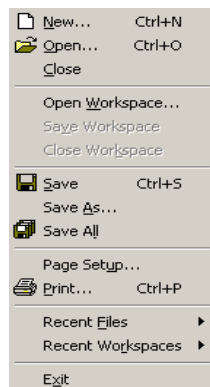


图 1.2 File 菜单

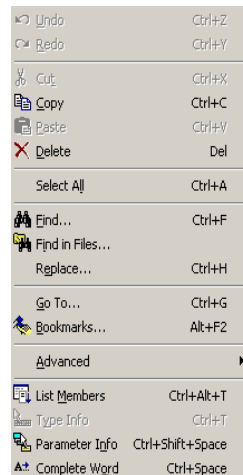


图 1.3 Edit 菜单

- ? Find 查找指定的字符串，快捷键 Ctrl+F；
- ? Find in Files 在指定的多个文件（夹）中查找字符串；

- ? Replace 将文件中字符串替换为指定的字符串，快捷键 Ctrl+H；
- ? Go to 将光标移到指定位置处，快捷键 Ctrl+G；
- ? Bookmarks 在光标当前位置处定义一个书签，快捷键 Alt+F2；
- ? Advanced 其他一些编辑操作，如将指定内容进行大小写转换；
- ? Breakpoints 在程序中设置、删除、查看断点，快捷键 Alt+F9；
- ? List Members 列出当前所有的成员变量，快捷键 Ctrl+Alt+T；
- ? Type Info 列出各种类型及其相关信息，快捷键 Ctrl+T；
- ? Parameter Info 列出所使用的参数及其相关信息，快捷键 Ctrl+Shift+Space；
- ? Complete Word 当输入单词的前几个字母时，智能列出可能的词语，快捷键 Ctrl+Space。

### 3 . View 菜单

View 菜单如图 1.4 所示，主要用来改变窗口和工具栏的显示方式，激活调试时所用的各个窗口等。

- ? ClassWizard 创建并编辑应用程序中的类，快捷键 Ctrl+W；
- ? Resource Symbols 打开资源符号浏览器，浏览和编辑资源文件中的符号；
- ? Resource Includes 用来修改资源符号文件名并编辑预处理代码；
- ? Full Screen 全屏显示当前窗口，按 Esc 键可切换回原来的显示方式；
- ? Workspace 显示并激活项目工作区窗口，快捷键 Alt+0；
- ? Output 激活输出窗口，显示程序建立过程的有关信息及运行结果，快捷键 Alt+2；
- ? Debug Windows 显示调试信息；
- ? Refresh 刷新选定的内容；
- ? Properties 激活属性对话框，设置或查看对象属性，快捷键 Alt+Enter。

### 4 . Insert 菜单

Insert 菜单如图 1.5 所示，主要用于项目及资源的创建和添加。

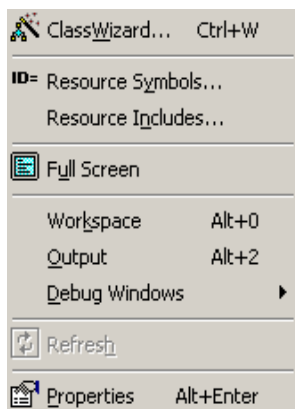


图 1.4 View 菜单

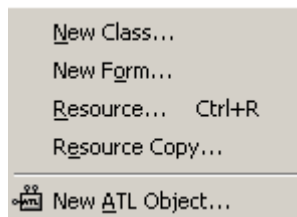


图 1.5 Insert 菜单

- ? New Class 创建一个新类并添加到项目中；
- ? New Form 创建一个新的表单并添加到项目中；

- ? Resource 创建一个新的资源或插入资源到资源文件中，快捷键 Ctrl+R；
- ? Resource Copy 复制选定的资源；
- ? File As Text 选定文本文件并插入到当前光标位置；
- ? New ATL Object 插入一个新的 ATL 对象。

## 5 . Project 菜单

Project 菜单如图 1.6 所示，主要用于项目的一些操作，如向项目中添加源文件。

- ? Set Active Project 选择指定项目为工作区的活动项目；
- ? Add To Project 将各种源文件、数据库描述文件添加到当前的项目中；
- ? Dependencies 编辑项目的从属关系；
- ? Settings 设置项目的各种属性，快捷键 Alt+F7；
- ? Export Makefile 生成当前可编译项目的.mak 文件；
- ? Insert Project into Workspacce 将已有项目加入到当前项目工作区中。

## 6 . Build 菜单

Build 菜单如图 1.7 所示，主要用于应用程序的编译、连接、调试、运行。

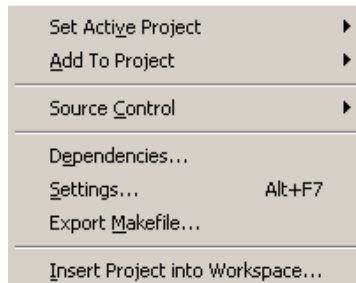


图 1.6 Project 菜单

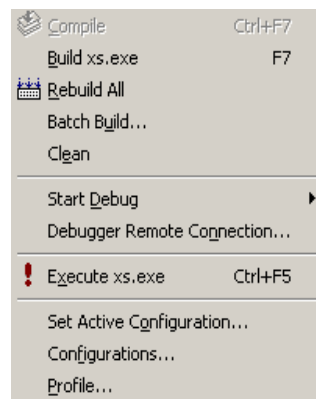


图 1.7 Build 菜单

- ? Compile 编译 C 或 C++源代码文件或资源文件，快捷键 Ctrl+F7；
- ? Build xs.exe 生成应用程序的.exe 文件（编译、连接），快捷键 F7；
- ? Rebuild All 重新编译和连接整个项目文件；
- ? Batch Build 编译和连接多个文件；
- ? Clean 清除当前项目的中间文件和输出文件；
- ? Start Debug 用于跟踪程序的调试和执行；
- ? Debugger Remote Connection 设置远程连接调试选项；
- ? Execute xs.exe 运行应用程序，快捷键 Ctrl+F5；
- ? Set Active Configuration 设置当前项目的配置，如 Win32 Release 和 Win32 Debug；
- ? Configuration 设置、修改项目的配置；
- ? Profile 设置与函数调用和运行时间等操作有关的一系列选项。

## 7 . Tools 菜单

Tools 菜单如图 1.8 所示，主要用于选择或定制集成开发环境中的一些实用工具，其中包括浏览程序符号、激活常用工具、更改选项设置、定制菜单与工具栏。

- ? Source Browser 浏览与程序中所有符号（类、函数、宏和类型）有关的信息；
- ? Close Source Browser File 关闭浏览文件；
- ? Visual Component Manager 可视化组件管理器，可方便地管理各种组件；
- ? Register Control 将 OLE 控件注册到 Windows 操作系统中；
- ? Error Lookup 检查 Win32 API 函数返回的标准错误代码；
- ? ActiveX Control Test Container 为测试 ActiveX 控件提供一个环境；
- ? OLE/COM Object Viewer 用来查看安装在系统上的所有 OLE 和 ActiveX 对象的信息；
- ? Spy++ 允许用户观察所有的进程、线程、窗口和窗口消息；
- ? MFC Tracer 调试程序时，可以激活各种级别的调试信息，再由 MFC 发送到输出窗口；
- ? Customize 定制命令、菜单、工具栏和快捷键；
- ? Options 更改开发环境的各种设置（如窗口、调试器、格式设置等）；
- ? Macro 创建和编辑宏；
- ? Record Quick Macro 录制新宏；
- ? Play Quick Macro 运行已经录制的宏。

## 8 . Window 菜单

Window 菜单如图 1.9 所示，主要用于文档窗口的操作，如排列窗口，打开或关闭窗口，重组或切分窗口等。

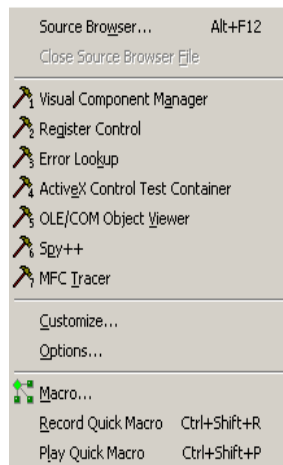


图 1.8 Tools 菜单

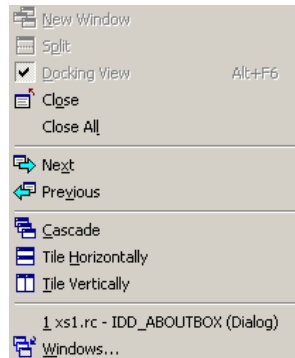


图 1.9 Window 菜单

- ? New Window 打开新窗口，以便在多个窗口显示当前文件内容；
- ? Split 切分文档窗口，以便查看同一文档的不同内容；
- ? Docking View 允许/禁止窗口的停靠特征，快捷键 Alt+F6；
- ? Close 关闭当前窗口；

- ? Close All 关闭所有打开的文档窗口；
- ? Next 使下一个文档窗口成为当前窗口；
- ? Previous 使上一个文档窗口成为当前窗口；
- ? Cascade 将当前所有打开的窗口在屏幕上重叠排放；
- ? Tile Horizontally 将当前所有打开的窗口在屏幕上横向平铺；
- ? Tile Vertically 将当前所有打开的窗口在屏幕上纵向平铺；
- ? Windows 文档窗口操作，可以关闭、保存和激活任何窗口。

## 9 . Help 菜单

Visual C++ 6.0 提供了大量详细的帮助信息，通过如图 1.10 所示的 Help 菜单便可得到这些帮助信息。在 Visual C++ 6.0 中，帮助窗口将不再是 Visual C++ 6.0 主窗口的一部分，它作为一个独立的应用程序具有独立的窗口。

- ? Contents 打开 MSDN Library 帮助系统，并进入其中的目录选项卡；
- ? Search 打开 MSDN Library 帮助系统，并进入其中的搜索选项卡；
- ? Index 打开 MSDN Library 帮助系统，并进入其中的索引选项卡；
- ? Use Extension Help 按 F1 或其他帮助命令将显示外部的帮助信息，若此命令没有选中，则启用 MSDN；
- ? Keyboard Map 弹出 Help Keyboard 对话框，查看所定义的快捷键；
- ? Tip of the Day 显示“Tip of the Day”对话框；
- ? Technical Support 用微软技术支持的方式获得帮助；
- ? Microsoft on the Web 微软网站。

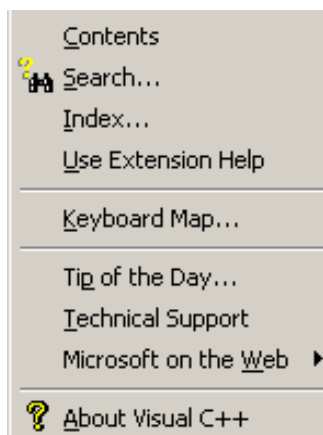


图 1.10 Help 菜单

### 1.3.3 工具栏

工具栏是一种图形化的操作界面，工具栏上的按钮分别和一些菜单命令相对应，提供了一种执行常用命令的快捷方法。

Visual C++ 6.0 显示的工具栏有：标准工具栏 (Standard)、类向导工具栏 (WizardBar) 及小型编译工具栏 (Build MiniBar)。

#### 1. 标准工具栏

标准工具栏如图 1.11 所示，其中的工具按钮命令大多数是常用的文档编辑命令，如新建、保存、撤销、恢复、查找等命令。



图 1.11 标准工具栏

## 2. 编译工具栏 (Build)

编译工具栏如图 1.12 所示，提供了常用的编译、连接操作命令。表 1.1 列出了编译工具栏的功能及对应的菜单命令。



图 1.12 编译工具栏

表 1.1 编译工具栏按钮的功能及对应的菜单命令

序号	对应菜单命令	功能描述
1	编译/Compile	编译程序
2	编译/Build	生成应用程序的.exe 文件
3	编译/Build Stop	停止编译
4	编译/ Build Execute	执行应用程序
5	编译/Go	单步执行
6	编译/Breakpoints	插入或删除断点

说明：表中的序号为图 1.12 工具栏按钮从左到右的顺序。

## 3. 类向导工具栏 (Wizard)

Visual C++ 6.0 提供了两个功能强大的向导工具，即应用程序向导 Appwizard 和类向导 ClassWizard。类向导允许用户在源文件中添加自己的类，为类的成员函数映射消息。Wizard 是配合类向导操作的工具栏，它包括三个相互关联的组合框和一个按钮，如图 1.13 所示。它们分别表示用户所选择的类信息 (Class)，消息标识 (Filter) 和相应的类成员函数 (Members)。单击 Actions 控制按钮可以将光标移动到指定的类成员函数在相应的源文件的定义和声明的位置处。通过 Wizard 工具栏的使用，使类的设计和维持实现了自动化，方便了程序的编辑和调试。



图 1.13 Wizard 工具栏

## 4. 工具栏的显示和隐藏

Visual C++ 6.0 的工具栏非常多，用户可以根据不同的需要选择相应的工具栏。显示或隐藏工具栏可以使用“Customize”对话框或快捷菜单两种方式进行操作。使用“Customize”对话框方式显示或隐藏工具栏步骤如下。

(1) “Tools”菜单 “Customize”菜单项。

(2) 弹出“Customize”对话框，如图 1.14 所示。单击“Toolbar”标签项，将列出所有的工具栏名称。

(3) 若要显示某工具栏，只需要用鼠标单击该工具栏名称，使得前面的复选框带有选中标记即可。去除复选框的选中标记，该工具栏就会从开发环境中消失。

在开发环境中的任何工具栏处右击鼠标，则会弹出一个包含工具栏名称的快捷菜单，如图 1.15 所示。

- ? Output 显示或隐藏输出窗口；
- ? Workspace 显示或隐藏项目工作区窗口；
- ? Standard 显示或隐藏标准工具栏；
- ? Build 显示或隐藏编译工具栏；
- ? Build MiniBar 显示或隐藏小型编译工具栏；
- ? ATL 显示或隐藏 ALT 工具栏；
- ? Resoruce 显示或隐藏创建资源的工具栏；
- ? Edit 显示或隐藏编辑工具栏；
- ? Debug 显示或隐藏调试工具栏；
- ? Browse 显示或隐藏项目信息浏览工具栏；
- ? Database 显示或隐藏数据库工具栏；
- ? WizardBar 显示或隐藏类向导工具栏；
- ? Customize 弹出“Customize”对话框。

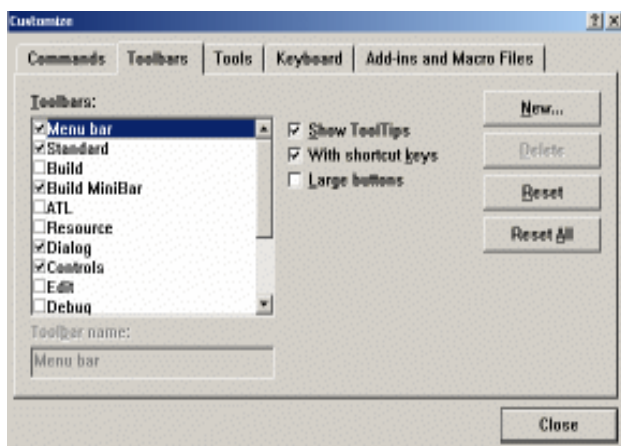


图 1.14 “Customize”对话框

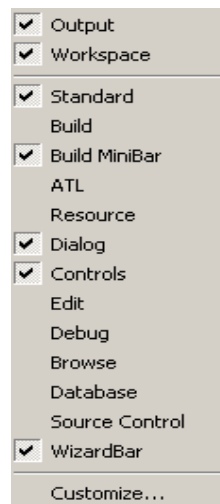


图 1.15 快捷菜单

## 1.4 项目和项目工作区

### 1.4.1 项目

Visual C++程序的核心是项目。现在大多数 Windows 应用程序除了源代码文件外，还包