

Visual C++

实用教程

张荣梅 梁晓林 编著

冶金工业出版社

Visual C++实用教程

张荣梅 梁晓林 编著

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

张荣梅 (CIP) 目录页

冶金工业出版社

2004

MJS76/07

内 容 提 要

本书从应用程序开发的不同阶段出发,分别系统地介绍了使用 Visual C++ 6.0 开发应用程序的基本知识、较复杂的和高级的编程技巧等。内容包括: Visual C++ 6.0 的开发环境, C++ 语言知识, 类和对象, 各种控件、菜单、工具栏、状态栏的创建, 图形应用程序, 单文档和多文档界面应用程序以及数据库应用程序的创建等。每章内容后均有实例和习题, 以便读者复习和上机实践。

本书通过大量的编程实践, 使读者能够尽快和牢固地掌握 Visual C++ 6.0 的各种编程技巧。

本书可作为大专院校计算机课程的教材, 也可作为各类计算机培训班和自学教材, 同时也适用于初学者和具有一定经验的 Visual C++ 6.0 用户使用和参考。

图书在版编目 (CIP) 数据

Visual C++实用教程/ 张荣梅, 梁晓林编著. —北京,
冶金工业出版社, 2004.2
ISBN 7-5024-3463-1

I. V… II. ①张…②梁… III. C 语言—程序
设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 123907 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 张登科 美术编辑 李 心 责任校对 卿文春 责任印制 牛晓波

北京兴顺印刷厂印刷; 冶金工业出版社发行; 各地新华书店经销

2004 年 2 月第 1 版, 2004 年 2 月第 1 次印刷

787mm×1092mm 1/16; 19.5 印张; 473 千字; 304 页; 1-3500 册

30.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

Visual C++是目前广为流行的新一代面向对象的程序开发环境。Visual C++ 6.0 提供了一个高效的 Windows 编程环境,将程序和资源的编辑、编译、调试和运行融为一体,具有优越的性能和强大的功能。本书通过大量的编程实践,使读者能够牢固地掌握 Visual C++ 6.0 的各种编程技巧。

本书各章安排如下:第1章是 Visual C++开发环境,让读者学习使用 Visual C++开发环境提供的各种工具和资源,并通过建立第1个应用程序去体验。第2章是 C++语言基础,对 C++编程语言进行了详细讨论。第3章是面向对象程序设计基础,介绍了类和对象的概念,并教读者在 Visual C++环境中如何定义类和使用类对象。第4章是 Windows 编程基础,读者将学习事件驱动程序设计的一些知识,Windows 消息、Windows 窗口以及 Win32 程序开发流程。第5章是 MFC 概述,读者将了解什么是 MFC, MFC 中类的组织结构和常用类,在 Visual C++中 MFC 应用程序框架结构, MFC 消息处理等。第6章是对话框与控件,介绍了常用控件的属性、操作、消息,并通过实例使读者学会使用它们创建用户自己的对话框,最后介绍了通用对话框及其使用方法。第7章是菜单、工具栏和状态栏,读者将学习如何在应用程序中创建菜单、工具栏和状态栏,并如何使用它们。第8章是图形设备接口与绘图,介绍了设备环境及设备环境类, GDI 绘图对象及其管理,并教读者如何绘图、如何绘制文本等。第9章是框架窗口、文档和视图,对框架窗口进行了概述,介绍了文档和视图及其关系,如何实现切分窗口和一档多视,最后讨论了文档序列化——文档的存取等。第10章是数据库编程,介绍了 ODBC 的概念,在 Visual C++中如何使用 ODBC 创建数据库应用程序等。

本书由张荣梅主编,书中第1、2、6、7、9、10章由张荣梅执笔,第4、5、8章由梁晓林执笔,第3章由赵宝琴执笔。参加本书大纲讨论及编写工作的还有孙洁丽、李福亮、白彦峰、甄增荣等。

在本书编写过程中,曾得到了同事们热心指导和无私帮助,在此向他们表示衷心的感谢。此外,在本书写作时,还参考了大量的文献资料,在此也一并向这些文献资料的作者深表谢意。

由于时间仓促和水平所限,书中不妥之处,敬请专家、读者不吝指正。

作 者

2003年10月

冶金工业出版社部分书目简介

书 名	定价 (元)
Photoshop 6.0 标准教程——计算机自学丛书	34.80
Flash 5 动画制作标准教程——计算机自学丛书	24.80
Dreamweaver 4.0 网页制作标准教程——计算机自学丛书	32.80
3D Studio MAX4.0 动画制作标准教程——计算机自学丛书	38.80
Fireworks 4 精彩网页标准教程——计算机自学丛书	24.80
Windows Me 标准教程——计算机自学丛书	23.80
计算机软件优秀培训用书——CorelDRAW 10 平面设计培训教程	24.80
2001 最新电脑装机教程	24.80
AutoCAD 2002 标准培训教程	38.80
最新 Office XP 三合一标准培训教程	30.00
Windows XP Professional 中文版与网络应用培训教程	19.80
一级 DOS 考试试题与模拟试题精选精析	24.00
一级 Windows 考试试题与模拟试题精选精析	23.00
二级 FORTRAN77 考试试题与模拟试题精选精析	26.00
二级 QBASIC 考试试题与模拟试题精选精析	25.00
二级 C 考试试题与模拟试题精选精析	27.00
计算机局域网	28
计算机新技术 DIY	36
高维稀疏聚类知识发现	14.50
AutoCAD 一级认证考试培训教程	16.00
Visual Foxpro 中 Windows API 调用技术与应用实例	49.00
网络原理与网络工程	35.00
超大规模集成电路衬底材料性能及加工测试	39.50
微型计算机控制系统	30.00
AutoCAD 2002 计算机辅助设计	29.50
智能控制原理及应用	29.00
数字电子技术基础	29.50
铝型材挤压模具 3D 设计 CAD/CAE 实用技术	28.00
80C51 单片机原理与应用技术	32.00
CAXA 电子图板教程	36.00
最优控制的结构化理论	15.00
智能决策支持系统研究开发及应用	16.00
多智能体计划调度系统的理论与应用	19.00
智能管理系统研究开发及应用	20.00
数据仓库与数据挖掘	30.00

目 录

1 Visual C++开发环境	1	2.1.3 C++程序的书写风格	18
1.1 工作平台概述	1	2.2 数据类型、运算符与表 达式	19
1.2 菜单栏	2	2.2.1 基本数据类型	19
1.2.1 File 菜单	2	2.2.2 常量与变量	20
1.2.2 Edit 菜单	2	2.2.3 运算符与表达式	24
1.2.3 View 菜单	3	2.3 C++语句的构造	28
1.2.4 Insert 菜单	3	2.3.1 顺序语句	28
1.2.5 Project 菜单	4	2.3.2 条件语句	31
1.2.6 Build 菜单	4	2.3.3 分支语句	32
1.2.7 Tools 菜单	5	2.3.4 循环语句	33
1.2.8 Windows 菜单	5	2.3.5 转移语句	34
1.3 工具栏	5	2.4 构造类型	37
1.3.1 工具栏的显示与隐藏	6	2.4.1 数组	37
1.4 项目和项目工作区	7	2.4.2 结构体	40
1.4.1 项目基本概念	7	2.4.3 共用体	42
1.4.2 Class View	8	2.4.4 枚举类型	43
1.4.3 Resource View	8	2.4.5 用 typedef 定义类型	44
1.4.4 File View	9	2.5 指针	44
1.5 资源	9	2.5.1 指针的定义	44
1.5.1 资源与资源标识	9	2.5.2 指针的基本操作	45
1.5.2 资源基本操作	10	2.5.3 指针与数组、结构体	47
1.5.3 资源文件的管理	11	2.5.4 new 与 delete	48
1.6 开发环境的初步实践	12	2.6 函数	49
1.6.1 用 AppWizard 创建	12	2.6.1 函数的定义和调用	49
1.6.2 理解程序框架	13	2.6.2 函数的参数传递	51
1.6.3 添加程序代码	14	2.6.3 函数的递归调用	54
1.6.4 编译运行	14	2.6.4 内联函数和重载函数	55
习题一	15	2.7 预处理指令与文件	57
2 C++语言基础	16	2.7.1 预处理	57
2.1 C++程序结构	16	2.7.2 文件	60
2.1.1 几个 C++程序	16	习题二	65
2.1.2 C++程序的基本组成	17		

3 面向对象程序设计基础	70	4.1.3 Windows 消息系统	120
3.1 面向对象的基本概念	70	4.2 Windows 窗口	123
3.1.1 类和对象	70	4.2.1 窗口	123
3.1.2 封装	71	4.2.2 定义窗口类的结构	123
3.1.3 继承	71	4.2.3 窗口类的注册与窗口建立	125
3.1.4 多态性	71	4.3 Win32 程序开发流程	127
3.2 类和对象的定义	72	4.3.1 Win32 程序开发过程	127
3.2.1 类的定义	72	4.3.2 窗口主函数 WinMain()	127
3.2.2 对象的定义	75	4.3.3 窗口函数 WndProc 和消息 处理	129
3.3 构造函数与析构函数	76	4.3.4 资源定义文件	130
3.3.1 构造函数	76	4.4 Hungarian 表示法	131
3.3.2 析构函数	77	习题四	131
3.4 静态类成员	78	5 MFC 概述	132
3.4.1 静态数据成员	78	5.1 MFC 简介	132
3.4.2 静态成员函数	79	5.2 MFC 中类的组织结构和常 用类	133
3.5 继承性与派生类	81	5.2.1 MFC 中类的组织结构	133
3.5.1 派生类的定义	81	5.2.2 COject 类	133
3.5.2 派生类的构造函数与析构 函数	86	5.2.3 应用程序体系结构类	134
3.5.3 多继承	87	5.2.4 可视对象类	134
3.6 多态性	91	5.2.5 通用类	136
3.6.1 编译时的多态性和运行时的 多态性	91	5.2.6 OLE 类	137
3.6.2 编译时的多态性	92	5.2.7 ODBC 数据库类	137
3.6.3 虚函数	93	5.3 MFC 应用程序框架结构	138
3.7 友元	102	5.3.1 单文档应用程序的建立	138
3.7.1 友元函数	102	5.3.2 理解 MFC AppWizard 创建的 程序框架	142
3.7.2 友元成员	103	5.3.3 MFC 应用程序的启动 流程	144
3.7.3 友元类	105	5.4 MFC 消息处理	146
3.8 模板	109	5.4.1 Windows 程序中的消息和消息 处理函数	146
3.8.1 函数模板	109	5.4.2 消息映射	148
3.8.2 类模板	112	5.4.3 使用 ClassWizard 管理消息和 命令	151
习题三	116	习题五	153
4 Windows 编程基础	119		
4.1 事件驱动与 Windows 消息 系统	119		
4.1.1 事件驱动程序设计	119		
4.1.2 Windows 消息	119		

6 对话框与控件	154	6.7.5 示例	192
6.1 对话框概述	154	6.8 通用对话框和消息对话框	200
6.1.1 对话框	154	6.8.1 通用对话框	200
6.1.2 使用对话框编辑器	156	6.8.2 消息对话框	204
6.1.3 对话框编程	159	习题六	205
6.1.4 控件的创建与使用	164		
6.1.5 访问控件	168		
6.2 静态控件	168	7 菜单、工具栏和状态栏	207
6.2.1 静态控件概述	168	7.1 菜单	207
6.2.2 静态控件属性	169	7.1.1 菜单概述	207
6.3 编辑框	170	7.1.2 用编辑器设计菜单	208
6.3.1 概述	170	7.1.3 编程控制菜单	211
6.3.2 属性和风格	170	7.1.4 快捷菜单(弹出式菜单)的设计 与使用	215
6.3.3 基本操作	171	7.1.5 示例	216
6.3.4 编辑控件的通知消息	171	7.2 工具栏	220
6.4 按钮类(CButton)控件	171	7.2.1 工具栏概述	220
6.4.1 按钮类控件概述	171	7.2.2 工具栏编辑器	220
6.4.2 按钮类的消息	172	7.2.3 工具栏与菜单结合	221
6.4.3 示例	172	7.3 状态栏	222
6.5 列表框控件	177	7.3.1 状态栏概述	222
6.5.1 概述	177	7.3.2 状态栏定义	222
6.5.2 属性	177	7.3.3 状态栏的常用操作	223
6.5.3 列表框(CListBox)的基本 操作	178	7.3.4 示例	224
6.5.4 列表框的通知消息	180	习题七	225
6.5.5 示例	180		
6.6 组合框类(CComboBox) 控件	183	8 图形设备接口与绘图	227
6.6.1 概述	183	8.1 设备环境和设备环境类	227
6.6.2 风格及类型属性	183	8.1.1 设备环境	227
6.6.3 组合框常见的操作	184	8.1.2 设备环境类	227
6.6.4 消息	185	8.2 GDI 绘图对象及其管理	229
6.6.5 示例	185	8.2.1 建设备环境	229
6.7 滚动类控件	188	8.2.2 GDI 对象的分类	229
6.7.1 概述	188	8.2.3 CPen 类	230
6.7.2 属性	188	8.2.4 CBrush 类	232
6.7.3 操作	190	8.2.5 CFont 类	233
6.7.4 消息 WM_HSCROLL 和 WM_VSCROLL	192	8.2.6 CRgn 类	236
		8.2.7 CPalette 类	239
		8.2.8 CBitmap 类	239

8.3 映射模式	240	9.5.3 文档序列化过程	271
8.4 CDC 中的绘图操作及其特点	241	9.5.4 文档序列化示例	273
8.4.1 设置绘图参数	241	习题九	279
8.4.2 绘图函数	242	10 数据库编程	280
8.5 绘制文本	243	10.1 数据库的访问和 ODBC	280
8.6 实例	244	10.1.1 数据库和 DBMS	280
习题八	246	10.1.2 开放式数据库接口 ODBC	280
9 框架窗口、文档和视图	247	10.1.3 MFC 和 ODBC	281
9.1 框架窗口	247	10.2 使用 ODBC	281
9.1.1 主框架窗口与文档窗口	247	10.2.1 CDatabase 类的用法	281
9.1.2 窗口风格的设置	247	10.2.2 CRecordset 类的用法	282
9.1.3 窗口状态的改变	249	10.2.3 CRecordView 类	288
9.2 文档和视图	251	10.2.4 了解 SQL	288
9.2.1 文档	252	10.3 使用 ODBC 创建数据库应用程序	291
9.2.2 视图	253	10.3.1 准备数据库	291
9.3 文档模板	255	10.3.2 创建应用程序外壳	293
9.3.1 文档模板类	256	10.3.3 设计主窗体	294
9.3.2 文档模板的构造函数	256	10.3.4 添加新记录	296
9.4 切分窗口与一档多视	260	10.3.5 删除记录	297
9.4.1 切分窗口	260	10.3.6 对记录进行排序	298
9.4.2 一档多视	262	10.3.7 查找	299
9.5 文档序列化	268	习题十	301
9.5.1 CArchive、CFile 类与 Serialize 函数	269	参考文献	302
9.5.2 使对象成为可串行化	270		

1 Visual C++开发环境

Visual C++是 Microsoft 公司推出的目前使用极为广泛的基于 Windows 平台的可视化编程环境。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 支持，在各种 C++语言开发工具中脱颖而出，成为目前最为流行的 C++语言集成开发环境。Visual C++ 6.0 分为标准版、专业版和企业版三种，但其基本功能是相同的，本书以企业版作为编程环境。

本章着重介绍 Visual C++开发环境及其为开发应用程序所提供的一些工具。Visual C++提供的工具很多，在任何一个应用程序中都不可能全部使用到这些工具。在此着重介绍一些主要工具，即在学习本书过程中和开发应用程序中经常使用的工具。

1.1 工作平台概述

当 Developer Studio 启动以后，可以看到一个如图 1-1 所示的窗口。开发工作室中每个区域都有其特定的用途。您可以重新布置各个区域，来定制开发工作室环境，使其适合自己的特殊开发需要。

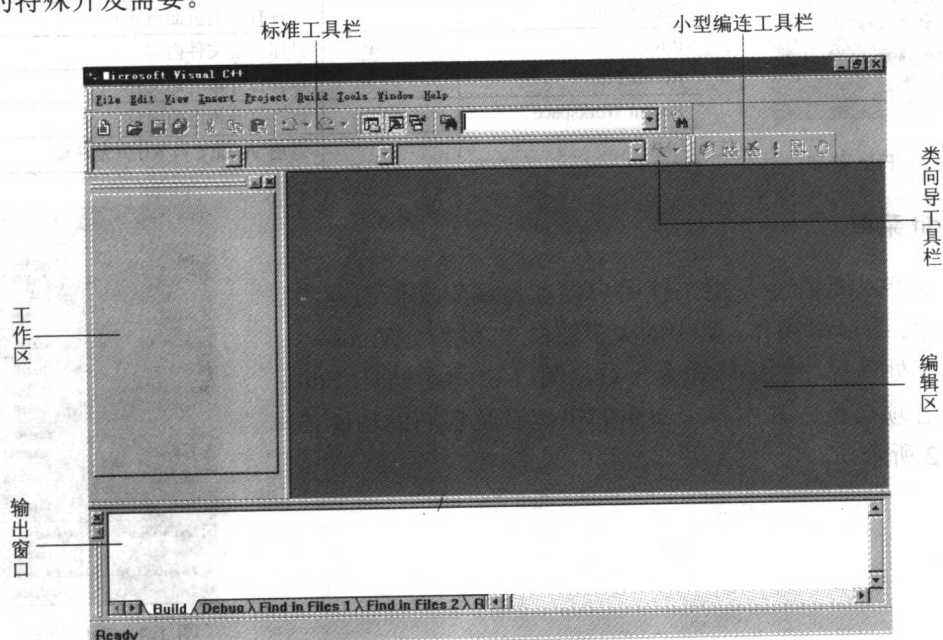


图 1-1 Visual C++的开始窗口

Visual C++开发环境界面由菜单栏、工具栏、项目工作区、编辑区、输出窗口以及状态栏等组成。

1.2 菜单栏

菜单栏包含了开发环境中几乎所有的命令，它为用户提供了文档操作、程序的编译、调试、窗口操作等一系列的功能。

1.2.1 File 菜单

File 菜单中的命令主要用来对文件和项目进行操作，例如“新建”、“打开”、“保存”、“打印”等。图 1-2 是 File 菜单中的各项命令，其中各项命令的功能描述如表 1-1 所示。

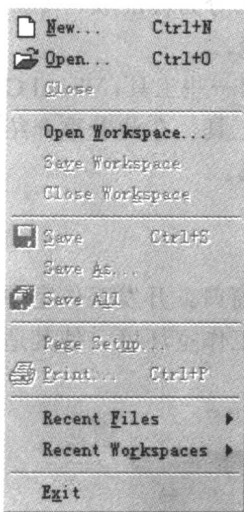


图 1-2 File 菜单

表 1-1 File 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
New	Ctrl+N	创建一个新项目文件
Open	Ctrl+O	打开已有的文件
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	Ctrl+S	保存当前文件
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	Ctrl+P	打印当前文件内容
Recent Files		选择打开最近的文件
Recent Workspace		选择打开最近的项目
Exit		退出 Visual C++ 6.0 开发环境

1.2.2 Edit 菜单

Edit 菜单中的命令是用来使用户便捷地编辑文件内容，如进行删除、复制等操作，其中的大多数命令功能与 Windows 中的标准字处理程序的编辑命令一致，图 1-3 所示的是 Edit 菜单中的各项命令，其中各项命令的快捷键及它们的功能描述如表 1-2 所示。

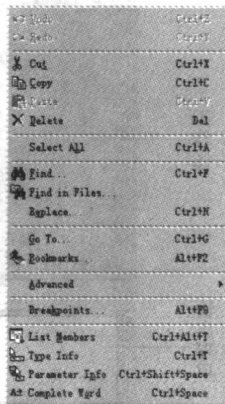


图 1-3 Edit 菜单

表 1-2 Edit 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
Undo	Ctrl+Z	撤销上一次操作
Redo	Ctrl+Y	恢复被撤销的操作
Cut	Ctrl+X	将当前选定的内容剪切掉，并移至剪切板中

续表 1-2

菜单命令	快捷键	功能描述
Copy	Ctrl+C	将当前选定的内容复制到剪切板中
Paste	Ctrl+V	将剪切板中的内容粘贴到光标当前位置处
Delete	Del	删除当前选定的对象或光标位置处的字符
Select All	Ctrl+A	选定当前活动窗口中的全部内容
Find	Ctrl+F	查找指定的字符串
Find in Files		在指定的多个文件（夹）中查找字符串
Replace	Ctrl+H	替换指定的字符串
Go to	Ctrl+G	将光标移到指定位置处
Bookmark	Alt+F2	在光标当前位置处定义一个书签
Advanced		其他一些编辑操作，如将指定内容进行大小写转换
Breakpoints	Alt+F9	在程序中设置断点
List Members	Ctrl+Alt+T	显示“词语敏感器”的“成员列表”选项
Type Info	Ctrl+T	显示“词语敏感器”的“类型信息”选项
Parameter Info	Ctrl+Shift+Space	显示“词语敏感器”的“参数信息”选项
Complete Word	Ctrl+Space	显示“词语敏感器”的“词语自动完成”选项

1.2.3 View 菜单

View 菜单中的命令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。图 1-4 所示的是 View 菜单中的各个命令，其中各项的功能描述如表 1-3 所示。

表 1-3 View 菜单命令的快捷键及功能描述

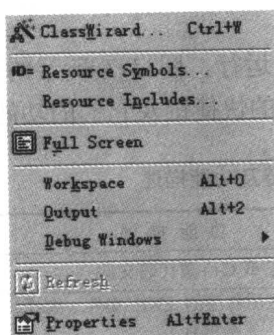


图 1-4 View 菜单

菜单命令	快捷键	功能描述
Class Wizard	Ctrl+W	弹出类编辑对话框
Resource symbols		显示和编辑资源文件中的资源标识符(ID号)
Resource Includes		修改资源包含文件
Full Screen		切换到全屏显示方式
Workspace	Alt+0	显示并激活项目工作区窗口
Output	Alt+2	显示并激活输出窗口
Debug Windows		操作调试窗口
Refresh		刷新当前选定对象的内容
Properties	Alt+Enter	编辑当前选定对象的属性

1.2.4 Insert 菜单

Insert 菜单中的命令主要用于项目及资源的创建和添加，图 1-5 所示的是 Insert 菜单中所包含的各个命令。表 1-4 列出了 Insert 菜单的各项命令的快捷键及它们的功能。

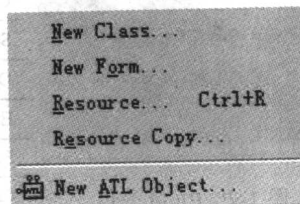


图 1-5 Insert 菜单

表 1-4 Insert 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
New Class		插入一个新类
New Form		插入一个新的表单类
Resource	Ctrl+R	插入指定类型的新资源
Resource Copy		创建一个不同语言的资源副本
File As Text		在光标位置处插入文本文件内容
New ATL Object		插入一个新的 ATL 对象

1.2.5 Project 菜单

Project 菜单的命令主要用于项目中添加源文件等, 图 1-6 所示的是 Project 菜单中的各个命令。表 1-5 列出了 Project 菜单的各项命令的快捷键及它们的功能。

表 1-5 Project 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
Set Active Project		激活指定的项目
Add To Project		将组件或外部的源文件添加在当前的项目中
Dependencies...		编辑当前项目的依赖关系
Settings...	Alt+F7	修改当前编译和调试项目的一些设置
Export Makefile...		生成当前可编译项目的 (.MAK) 文件
Insert Project into Workspace...		将项目加入到项目工作区中

图 1-6 Project 菜单

1.2.6 Build 菜单

Build 菜单中的命令主要用于应用程序的编译、连接、调试、运行, 图 1-7 所示的是 Build 菜单所包含的所有命令。表 1-6 列出了 Build 菜单的各项命令的快捷键及它们的功能。

表 1-6 Build 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
Compile ex_abc11g.cpp	Ctrl+F7	编译 C 或 C++ 源代码文件
Build ex_abc.exe	F7	生成应用程序的 EXE 文件 (编译、连接, 又称编连)
Rebuild All		重新编连整个项目文件
Batch Build...		成批编连多个项目文件
Clean		清除所有编连过程中产生的文件
Start Debug		调试的一些操作
Debugger Remote Connection...		设置远程调试连接的各项环境设置
Execute ex_abc.exe	Ctrl+F5	执行应用程序
Set Active Configuration...		设置当前项目的配置
Configurations...		设置、修改项目的配置
Profile...		为当前应用程序设定各选项

图 1-7 Build 菜单

1.2.7 Tools 菜单

Tools 菜单中的命令主要用于选择或制定开发环境中的一些实用工具，如图 1-8 所示，其中除了 Visual C++ 6.0 组件（如 Spy++ 等）外，其余的各项命令的快捷键和功能描述如表 1-7 所示。

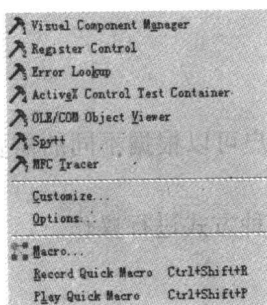


图 1-8 Tools 菜单

表 1-7 Tools 菜单命令的快捷键及功能描述

菜单命令	快捷键	功能描述
Source Browser	Alt+F12	浏览对指定对象的查询及其相关信息
Close Source Browser File		关闭浏览信息文件
Customize		定制菜单及工具栏
Options		改变开发环境的各种设置
Macro		进行宏操作
Record Quick Macro	Ctrl+Shift+R	录制新宏
Play Quick Macro	Ctrl+Shift+P	运行新录制的宏

1.2.8 Windows 菜单

Windows 菜单中的命令主要用于文档的操作，如排列文档窗口、打开或关闭一个文档窗口、重组或切分文档窗口等。图 1-9 所示的是 Windows 菜单中的各条命令，其中各项命令的快捷键及它们的功能描述如表 1-8 所示。

表 1-8 Windows 菜单命令的快捷键及功能描述

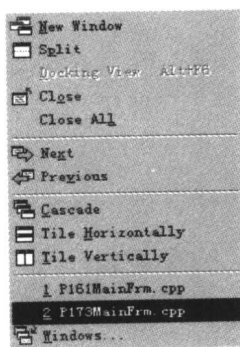


图 1-9 Windows 菜单

菜单命令	快捷键	功能描述
New Window		再打开一个文档窗口显示当前窗口内容
Split		文档窗口切分命令
Docking View	Alt+F6	浮动显示项目工作区窗口
Close		关闭当前文档窗口
Close All		关闭所有打开过的文档窗口
Next		激活并显示下一个文档窗口
Previous		激活并显示上一个文档窗口
Cascade		层铺所有的文档窗口
Tile Horizontally		多个文档窗口上下依次排列
Tile Vertically		多个文档窗口左右依次排列
Windows		文档窗口操作

需要说明的是：随着开发环境当前状态的改变，菜单栏以及菜单中的命令项也会随之变化。

1.3 工具栏

第一次运行 Visual C++ 时，开发环境（参见图 1-1）显示的工具栏有：标准工具栏（Standard）、类向导工具栏（WizardBar）及小型编连工具栏（Build MinBar）。

(1) 标准工具栏。标准工具栏包括绝大多数标准工具：打开和保存文件、剪切、复制、粘贴，以及可能有用的其他各种命令。

(2) 类向导工具栏。类向导工具栏使您能执行许多类向导动作，而不用打开类向导。

(3) 小型编连工具栏。小型编连工具栏提供了常用的编译、连接操作命令。整个编连工具栏还允许您在多种编连配置之间切换。

上述这些工具栏不仅可以显示或隐藏，而且还可以根据自己的需要对工具栏的按钮及其命令进行定制。

1.3.1 工具栏的显示与隐藏

Visual C++所拥有的工具栏要比前面介绍的工具栏多得多，用户可以根据不同的需要选择打开相应的工具栏，或隐藏那些用不着的工具栏。

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。

1.3.1.1 “Customize”对话框方式

使用“Customize”对话框方式显示或隐藏工具栏的步骤如下：

(1) “Tools”菜单→“Customize”菜单项；

(2) 弹出“Customize”对话框，如图 1-10 所示，单击 ToolBars 标签项，将显示出所有的工具栏名称，那些显示在开发环境上的工具栏名称前面将带有选中标记（√）。

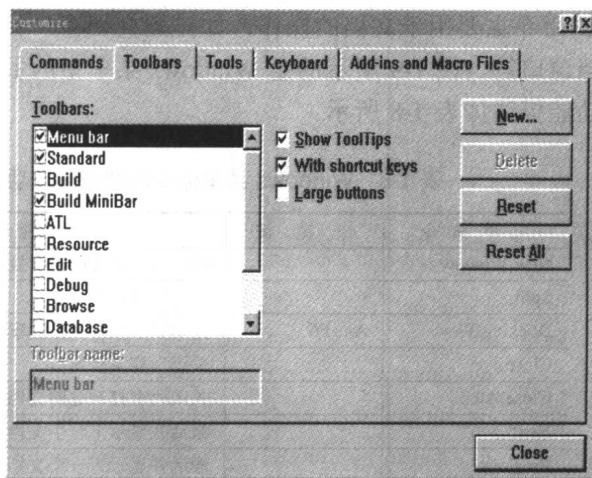


图 1-10 Customize 对话框

(3) 若要显示某工具栏，只需用鼠标左击该工具栏名称，使得前面的复选框带有选中标记即可；同样的操作再进行一次，工具栏名称前面的复选框的选中标记将去除，该工具栏就会从开发环境中消失。

1.3.1.2 快捷菜单方式

如果嫌上述操作不够便捷，那么可在开发环境中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 1-11 所示。表 1-11 列出了各个菜单命令的功能。

表 1-11 工具栏快捷菜单命令及其功能描述

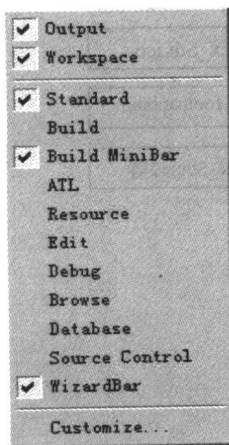


图 1-11 工具栏的快捷按钮菜单

菜单命令	功能描述
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	显示或隐藏编连工具栏
Build MiniBar	显示或隐藏小型编连工具栏
ATL	显示或隐藏 ATL 工具栏
Resource	显示或隐藏创建资源的工具栏
Edit	显示或隐藏编辑工具栏，它提供了书签、缩进量调整、显示或隐藏白字符等操作
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

1.4 项目和项目工作区

一个 Windows 应用程序通常有许多源代码文件以及菜单、工具栏、对话框、图标等资源文件，这些文件都将纳入应用程序的项目中。通过对项目工作区的操作，可以显示、修改、添加、删除这些文件。项目工作区可以管理多个项目。

1.4.1 项目基本概念

在 Windows 环境下，大多数应用程序除了许多源代码文件外，还包含菜单、工具栏、对话框、图标等，Visual C++称它们为资源，这些资源通常用资源文件保存起来。另外，还要包含应用程序代码源文件编连时所需要的库文件、系统 DLL 文件等。有效组织这些文件并维护各源文件之间的依赖关系是应用程序最先要达到的目的，Visual C++中的项目就起这样的作用。实际上，项目作为工作区中的主要内容已加入集成开发环境中，不再需要自己来组织这些文件。只需要在开发环境中进行设置、编译、连接等操作，就可创建可执行的应用程序文件或 DLL 文件。

在 Visual C++中，项目中所有的源文件都是采用文件夹的方式进行管理的，它将项目名称作为文件夹名，在此文件夹下包含源程序代码文件（.cpp，.h）、项目文件（.dsp）、项目工作区文件（.dsw）以及项目工作区配置文件（.opt），还有相应的 Debug（调试）或 Release（发行）、Res（资源）等子文件夹。例如用户创建的单文档应用程序项目名是 EX_Sdi，则各文件的布局如图 1-12 所示。当然，不同类型的项目的文件类型及数目会有所不同。

在开发环境中，Visual C++通过左边的项目工作区窗口对项目进行各种管理。项目工作区窗口包含三个页面，它们分别是 Class View、Resource View 和 File View。

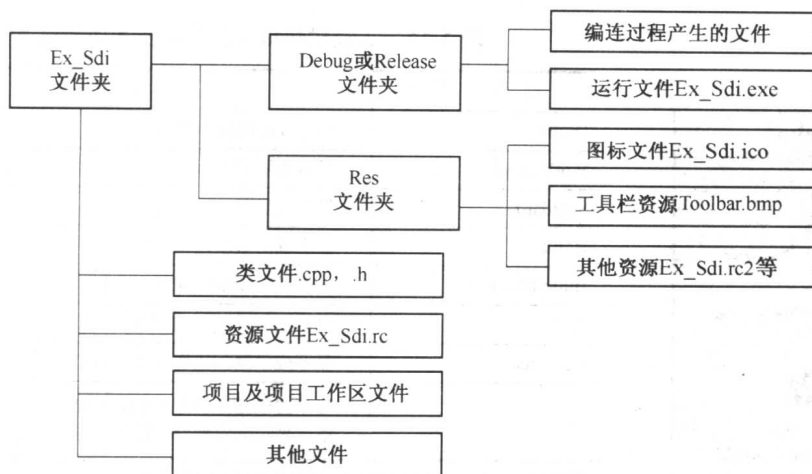


图 1-12 项目文件的布局

1.4.2 Class View

项目工作区窗口的 Class View 用于显示项目中的所有类的信息。假设打开的项目名为 Ex_Sdi，单击项目工作区窗口底部的 Class View，则会显示出一个标题为“Ex_Sdi Classes”的树状条目，在它的前面是一个图标和一个套在方框中的符号“+”，单击符号或双击图标，Ex_Sdi 中的所有类名将被显示，如 CMainFrame、CEX_SdiApp、CEX_SdiDoc、CEX_SdiView 等，如图 1-13 所示。

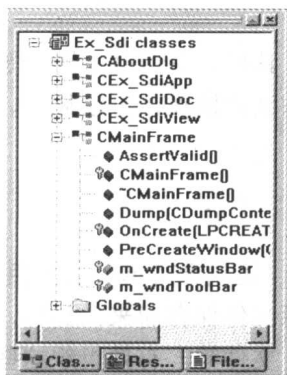


图 1-13 ClassView

在 Class View 页面中，每个类名前也有一个图标和一个套在方框中的符号“+”，双击图标，则直接打开并显示类定义的头文件（如 CEX_SdiApp.h）；单击符号“+”，则会显示该类中的成员函数和成员变量，双击成员函数前的图标，则在文档窗口中直接打开源文件并显示相应函数体代码。

这里，要注意一些图标所表示的含义。例如，在成员函数的图标中，

1.4.3 Resource View

包含了项目中所有资源的层次列表。在 Visual C++ 中，每一个图片、字符串、工具栏、菜单、对话框、图标等非代码元素等都可以看作是一种资源。图 1-14 显示了一个典型的 Resource View 窗口，当然，每一种资源都有自己使用的图标。

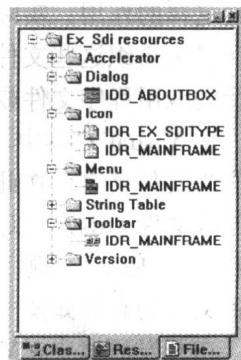


图 1-14 ResourceView