



Visual C++ 6

使用详解

朱东海 刘良华 董海宁 程进兴 编著

- 以Visual C++ 集成开发环境和语言基础为铺垫
- 以MFC基本类库、消息响应、MFC视窗与文档、MDI程序特性和动态链接库为重点
- 以Visual C++6 高级应用——线程创建、多线程应用程序开发、利用WinInet进行网络开发为终结



机械工业出版社
China Machine Press

Visual C++ 6

使用详解

朱东海 刘良华 董海宁 程进兴 编著
抖斗书屋 审



机械工业出版社
China Machine Press

Microsoft公司最新推出的Visual Studio 6.0 提供了众多的先进特性和强大工具,而Visual C++6是其核心开发工具之一,在网络应用方面增加了不少功能。本书旨在向广大的程序员介绍Visual C++6的使用方法和技巧,首先简要地介绍了Visual C++6的开发环境、C++的语言基础和利用Visual Studio开发应用程序的步骤;然后循序渐进地讲述了MFC的基本类库、键盘与鼠标消息响应、MFC视窗和文档、对话框的应用技巧;随后以SDI程序为辅,重点介绍MDI程序特性,并详细讲解动态链接库的创建和Visual C++6的高级应用,最后对利用WinInet进行网络开发做了简单介绍。

本书中文简体字版由机械工业出版社出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

版权所有,翻印必究。

图书在版编目(CIP)数据

Visual C++ 6 使用详解/朱东海等编著.-北京:机械工业出版社,1999.7

ISBN 7-111-07260-X

I.V… II.朱… III.C语言-程序设计 IV.TP312

中国版本图书馆CIP数据核字(1999)第20813号

出 版 人:马九荣(北京市百万庄大街22号 邮政编码100037)

责任编辑:姜 南

昌平环球印刷厂印刷·新华书店北京发行所发行

1999年7月第1版第1次印刷

787mm×1092mm1/16·26.5印张

印 数:0001-8000册

定 价:42.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

Microsoft公司最新推出的Visual Studio 6.0提供了众多的先进特性和强大工具，也是Visual系列开发软件包目前的最高版本，可完全满足现代企业机构创建企业级解决方案的要求。Visual Studio 6.0中的基于组件开发的工具可实现基于Windows DNA体系的应用方案。该套件所提供的对整个生命周期的支持可在设计、开发、分析、管理和部署等各个阶段中得到全面体现。新的企业级数据库工具采用Universal Data Access数据访问策略，结合ActiveX Data Object接口，为用户实现高性能的企业级数据处理。而套件中包含的集成化Web项目模型、组件管理和源代码控制使开发组功能得到了增强。除此之外，套件能与诸如Microsoft BackOffice家族等基于Windows NT的众多应用程序服务集成，加速组件应用在Internet和Intranet的部署。

作为其核心开发工具之一，Visual C++ 6在网络应用开发方面增加了不少功能，例如复杂的网络数据库引擎、访问等。但是从界面上看，Visual C++ 6似乎没有太大的变化，这样也使广大使用过Visual C++ 5.0或更早版本的编程爱好者能够很快地熟悉Visual C++ 6的开发环境。

为了向广大的程序员介绍Visual C++6的使用方法和技巧，我们编著了本书。考虑到本书读者对Visual C++知识掌握的不同情况，我们在该书中对基础知识和高级的应用技巧进行了较为全面的覆盖。

本书由中科辅龙计算机技术有限公司图书部策划，第2、3、4、5、6章由朱东海编写；第1、10、11、12章由刘良华编写；第7、8、9章由董海宁编写；第13章由程进兴编写。全书由朱东海组织编写，史惠康和郭美山统稿。另外，刘慧姜、袁英战完成了部分文字录入工作，谨此表示衷心感谢！

本书作者均有多年的商用软件开发经验，而且大多数具有丰富的计算机类图书的编写经验，相信读者在阅读的过程中可以跟随作者的思路在Visual Studio里愉快地漫游。

当然，由于本书成书时间仓促，加上作者水平和本书篇幅所限，书中难免挂一漏万，甚至有错误或不妥之处，望广大读者在使用过程中能不吝赐教。我们非常欢迎读者就本书内容跟我们联系，提出你们的宝贵意见或开展其他形式的合作。

最后，全体作者向本书的读者致意，愿知识能带给你神奇的力量！

如对本书内容有疑问的读者，可与抖斗书屋读者服务部联系，联系电话：(010)62565533转3301。

作者

1998年12月于北京中科院计算所

全书导读

本书是面向所有对Visual C++6感兴趣的广大读者的。因为读者情况的差别，我们在此对全书的基本内容作一概述，并且向各类读者提出作者推荐的阅读方法。

下表所示为作者对各种类型读者如何阅读本书的建议，当然这只是一个笼统的建议，如果读者认为并不恰当，可以根据自己的实际情况使用本书。

读者情况	建议阅读方法
没使用过Visual C++的以前版本	请仔细阅读本书的每一个章节，尤其是第4章对MFC基本类库的介绍，应当多加了解，至于高级的应用部分，如第11~14章，则可以在对基础知识有比较透彻的理解之后再学习
使用过Visual C++的以前版本	如果读者已经具备面向对象设计技术的基本知识，而且对Visual C++语言和MFC有相当的了解，可以跳过第2章，直接阅读后面的内容

内容介绍

本部分旨在将全书的各个章节的内容作一简要的介绍，以方便读者从中快速找到自己所需要和感兴趣的内容。

第1章 Visual C++6概述。本章主要介绍了Visual C++ 6的集成开发环境，包括工具栏、菜单栏等。通过本章的学习，即使用户从未接触过Visual C++系列开发环境，也能够很快地熟悉Visual C++6。

第2章 C++语言基础。本章主要介绍了C++语言的基础知识。首先介绍了C++语言中从C语言继承而来的一些基本概念，然后着重介绍了C++语言中的核心概念——类。

第3章 利用Visual Studio开发应用程序。本章主要介绍了如何在Visual Studio里创建应用程序。首先结合实例介绍了利用AppWizard来创建基于MFC的应用程序，然后着重介绍了在编辑应用程序中要注意的问题和一些编辑技巧。最后介绍了关于应用程序编译和链接的基本知识。

第4章 MFC基本类库介绍。本章主要介绍了MFC基本类库中的重要类及其成员函数和成员变量。用MFC基本类库开发应用程序，就要对MFC的基本类及其提供的功能进行必要的了解。本章通过对最重要的类的介绍使读者能对MFC基本类库有一个初步的了解，通过阅读本章可以使读者在开发应用程序中适应各种不同的功能要求，满足各种灵活的程序设计需要。但限于篇幅，本章仅介绍了一部分的基本类，而且对成员函数的介绍仅限于对功能的概述，至于其中各参数的取值及各成员函数的调用者和调用场合等均未予说明。

第5章 键盘与鼠标消息响应。本章对应用程序跟用户间最重要的交互方式——鼠标输入和键盘输入进行了较深入的介绍，而且对如何在应用程序开发中响应鼠标输入和键盘输入，结合实例进行了示范。

第6章 菜单栏、工具栏及状态栏应用技巧。本章主要介绍了菜单栏、工具栏和状态栏的应用技巧。首先结合实例介绍了如何在生成新项目的同时生成MFC缺省的状态栏和工具栏，并介绍了如何应用CToolBar类、CToolBarCtrl类、CStatusBar类和CStatusBarCtrl类来对状态栏

和工具栏进行动态修改。其次，结合实例介绍了如何在应用程序中增加和删除菜单及菜单命令，介绍了CMenu类的使用方法和注意事项。

第7章 MFC的视窗和文档。本章介绍了文档类和视窗类的概念，并且通过一个应用程序，介绍了两者之间的关系，并且向读者介绍了如何在文档中存储数据，如何将文档中的数据在用户窗口中显示输出，然后介绍了CStringArray类和CWordArray类。

第8章 对话框的使用。本章主要介绍了如何在程序中使用对话框。总共分为三个部分的内容：基于对话框的应用程序、模态对话框和非模态对话框。在基于对话框的应用程序的部分，通过一个简单的例程介绍了如何生成一个基于对话框的应用程序。在模态对话框与非模态对话框的部分，主要是立足于模态对话框与非模态对话框的比较。

第9章 单文档与多文档。本章向读者初步介绍了SDI程序和MDI程序的特性，并且通过例程的编制详细阐述了SDI程序和MDI程序的编制过程。在SDI部分，主要是突出了SDI与MDI程序的对比。在MDI部分，着重突出MDI程序的特性。

第10章 打印及打印预览。本章主要介绍了打印和打印预览的实现方法及其在MFC中实现的内部机制。除了简单的打印和打印预览的实现外，还介绍了一些高级功能的实现方法，如在打印预览（或打印）时不显示视图窗口中的背景图案（非文档部分）等。

第11章 定制帮助文件。本章主要介绍了用户帮助系统的实现方法和技巧，通过一个简单的帮助系统的实现，对包括帮助源文件的制作、编译以及在应用程序中的使用都作了较为详细的介绍。另外，在本章最后还介绍了Visual C++6中扩展功能的使用方法。

第12章 动态链接库。本章详细介绍了动态库的创建，包括普通Win32动态库、MFC动态库以及ARX程序（运行在AutoCAD14平台上）的创建。其实，ARX程序也是一种动态链接库形式的可执行模块，它的编写关键是设置好各种编译开关。除此之外，本章还通过一个简单的例程介绍了定制动态库的使用方法。

第13章 Visual C++6高级应用。本章首先讲述了线程的基本概念、线程的创建与终止方法、开发简单的多线程应用程序的基本步骤、线程通信的概念，线程间通信的方法以及线程同步的基本概念、线程同步的几种基本方法。同时，本章还介绍了利用WinInet进行网络开发的基本概念、WinInet类的继承关系以及各类的基本功能，并对用WinInet类进行网络开发的基本过程作了简单的介绍。

目 录

前言

全书导读

第1章 Visual C++ 6概述1

1.1 Visual C++ 6用户界面1

1.2 Visual C++ 6工具栏3

1.2.1 标准工具栏3

1.2.2 Build工具栏4

1.2.3 Debug工具栏5

1.2.4 WizardBar工具栏6

1.3 Visual C++ 6菜单栏7

1.3.1 File菜单7

1.3.2 Edit菜单10

1.3.3 View菜单11

1.3.4 Insert菜单12

1.3.5 Project菜单12

1.3.6 Build菜单13

1.3.7 Debug菜单14

1.3.8 Tools菜单16

1.3.9 Window菜单22

1.3.10 Help菜单23

1.4 Visual C++ 6新特点25

1.4.1 编译、链接25

1.4.2 调试工具26

1.4.3 OLE对象和数据库27

1.4.4 编辑器27

1.5 本章小结28

第2章 C++语言基础29

2.1 C++语言的基础知识29

2.1.1 一个简单而完整的Visual C++例程29

2.1.2 标识符30

2.1.3 变量和常量31

2.1.4 数据类型32

2.1.5 结构32

2.1.6 枚举33

2.1.7 数组34

2.1.8 指针36

2.1.9 链表37

2.2 C++语言不同于C语言的新特点

——类的知识38

2.2.1 C++中的类与C语言中的

结构之异同38

2.2.2 类的成员函数40

2.2.3 类的数据成员40

2.2.4 类的构造函数41

2.2.5 类的析构函数42

2.2.6 特殊类型的构造函数42

2.2.7 生成类对象的几种方法43

2.2.8 访问类的数据成员和成员函数44

2.2.9 友元函数和友元类44

2.2.10 类继承45

2.2.11 类的多重继承46

2.2.12 类的多态性之一——函数

重载和运算符重载46

2.2.13 类的多态性之二——虚函

数和纯虚函数49

2.3 本章小结51

第3章 利用Visual Studio开发应用程序52

3.1 创建MFC应用程序52

3.2 编辑应用程序66

3.2.1 代码的编写格式66

3.2.2 充分利用各种编辑键68

3.3 编译和链接应用程序69

3.4 应用程序的调试70

3.4.1 设置断点72

3.4.2 使用Watch窗口进行变量查看74

3.4.3 使用Call Stack进行调用过程查看75

3.5 本章小结76

第4章 MFC基本类库介绍77

4.1 文档类77

4.1.1 CDocument类78

4.1.2 COleDocument类79

4.1.3 CDocItem类81

4.1.4 CDocObjectServer类81

4.1.5 COleLinkingDoc类	81	4.7.1 CDialog类	111
4.1.6 COleServerDoc类	81	4.7.2 CFileDialog类	112
4.1.7 CRichEditDoc类	82	4.7.3 CPrintDialog类	113
4.1.8 COleClienItem类	83	4.8 本章小结	114
4.1.9 COleServerItem类	87	第5章 键盘与鼠标消息响应	115
4.2 视图类	89	5.1 概述	115
4.2.1 CView类	89	5.1.1 窗口的客户区和非客户区	115
4.2.2 CScrollView类	91	5.1.2 鼠标消息和键盘消息	116
4.2.3 CCtrlView类	91	5.2 常用鼠标和键盘消息	118
4.2.4 CDaoRecordView类	91	5.2.1 常用的键盘消息	118
4.2.5 CEditView类	92	5.2.2 常用的鼠标消息	126
4.2.6 CFormView类	93	5.3 鼠标消息的高级应用	132
4.2.7 CListView类	93	5.4 本章小结	151
4.2.8 CRecordView类	93	第6章 菜单栏、工具栏及状态栏	
4.2.9 CRichEditView类	93	应用技巧	152
4.2.10 CTreeView类	96	6.1 概述	152
4.3 框架窗口类	96	6.2 如何创建工具栏和状态栏	154
4.3.1 CFrameWnd类	96	6.2.1 如何生成缺省的工具栏和状态栏	154
4.3.2 CMDIChildWnd类	98	6.2.2 如何在程序代码中对工具栏和	
4.3.3 CMDIFrameWnd类	99	状态栏进行动态修改	161
4.3.4 CMiniFrameWnd类	99	6.3 如何创建自定义的菜单栏	167
4.3.5 COleIPFrameWnd类	99	6.3.1 如何在资源文件中对菜单	
4.4 主要应用程序结构类	100	进行修改	167
4.4.1 CWinApp类	100	6.3.2 如何在程序代码中对菜单	
4.4.2 CWinThread类	103	进行动态修改	178
4.4.3 CDocTemplate类	104	6.4 本章小结	195
4.4.4 CMultiDocTemplate类	105	第7章 MFC的视窗和文档	196
4.4.5 CSingleDocTemplate类	106	7.1 一个例程try.exe	196
4.5 文件服务类	106	7.2 CDocument类	197
4.5.1 CFile类	106	7.3 CView类	198
4.5.2 CHttpFile类	108	7.4 用户的输入	200
4.6 绘图类及绘图对象类	109	7.4.1 在Windows下使用键盘	200
4.6.1 CDC类	109	7.4.2 在Windows下使用鼠标	201
4.6.2 CPaintDC类	109	7.5 try程序的改进	201
4.6.3 CClientDC类	109	7.5.1 try程序对键盘输入的处理	201
4.6.4 CFont类	109	7.5.2 try程序对鼠标消息的处理	211
4.6.5 CPen类	110	7.5.3 CStringArray类	216
4.6.6 CBrush类	110	7.5.4 CWordArray类	219
4.7 对话框类	111	7.5.5 关于Collection类的介绍	224

7.6 本章小结	225	9.1.6 声明视窗类的数据成员	264
第8章 对话框的使用	226	9.1.7 对文档类中的数据成员 进行初始化	264
8.1 对话框	226	9.1.8 对视窗类中的数据成员 进行初始化	265
8.1.1 对话框的组成	226	9.1.9 编写画圆的代码	266
8.1.2 模式对话框与非模式对话框	226	9.1.10 编写响应鼠标消息的代码	268
8.2 基于对话框的应用程序	227	9.2 多文档界面(MDI)应用程序	271
8.2.1 预览MyDialog1.EXE的运行状态	227	9.3 MDI应用程序的组成及特征	272
8.2.2 创建MyDialog1.EXE的工程	228	9.3.1 CMDIFrameWnd类	272
8.2.3 MyDialog1工程的设计	229	9.3.2 CMultiDocTemplate类	274
8.2.4 将一个变量附加于编辑窗口	230	9.4 一个MDI应用程序	276
8.2.5 向Exit按键附加代码	231	9.4.1 程序的特征与功能	276
8.2.6 处理OK and Cancel键的 BN_CLICKED事件的代码	231	9.4.2 创建应用程序的工程	277
8.2.7 处理Yes,No,and Cancel键的 BN_CLICKED事件的代码	234	9.4.3 试运行HELLO程序	279
8.2.8 处理Retry and Cancel键的 BN_CLICKED事件的代码	236	9.4.4 声明文档类的数据成员	280
8.2.9 处理Yes and No键的BN_CLICKED 事件的代码	238	9.4.5 声明视窗类的数据成员	281
8.3 模式对话框	240	9.4.6 初始化文档类中的数据成员	282
8.3.1 预览含有模式对话框程序的运行	240	9.4.7 初始化视窗类中的数据成员	282
8.3.2 创建MyDialog2程序	242	9.4.8 编写画圆的代码	283
8.3.3 设计模式对话框	242	9.4.9 使用HELLO程序的MDI特性	285
8.3.4 将Dialog1类加入MyDialog2程序	244	9.4.10 加入使用鼠标对圆进行 拖动的代码	288
8.3.5 向程序主窗口中添加菜单	245	9.4.11 编写存储和装入文档的代码	291
8.3.6 程序的文档类CMyDialog2Doc	246	9.4.12 同一文档的多个视窗	295
8.3.7 程序的视窗类CMyDialog2View	246	9.4.13 UpdateAllViews()函数和 OnUpdate()函数	296
8.3.8 程序的对话框类Dialog1	250	9.4.14 扩展HELLO程序	299
8.4 非模式对话框	253	9.5 本章小结	303
8.4.1 Dialog1类的修改	254	第10章 打印及打印预览	304
8.4.2 改写程序的视窗类	255	10.1 实现基本打印支持	304
8.5 本章小结	256	10.1.1 添加菜单命令	305
第9章 单文档与多文档	258	10.1.2 修改源代码	307
9.1 单文档界面应用程序(SDI)	258	10.2 实现高级打印支持	311
9.1.1 什么是单文档界面应用程序	258	10.3 本章小结	320
9.1.2 一个SDI应用程序	259	第11章 定制帮助文件	321
9.1.3 创建MyCircle程序的工程	260	11.1 F1和Shift+F1帮助	321
9.1.4 初步运行MyCircle程序	262	11.1.1 增加F1帮助支持	322
9.1.5 声明文档类的数据成员	263	11.1.2 增加Shift+F1帮助支持	325

11.1.3 使用AppWizard选项	331	12.2 动态库的创建	354
11.2 建立帮助源文件	331	12.2.1 创建普通Win32动态库	354
11.2.1 RTF格式简介	332	12.2.2 创建MFC动态库	372
11.2.2 帮助文件的脚注形式	332	12.3 创建ARX程序	375
11.2.3 建立源文件	332	12.3.1 基于ObjectARX编程	376
11.3 编译帮助源文件	338	12.3.2 基于ObjectARX和MFC编程	382
11.3.1 建立工程文件并编译	338	12.4 本章小结	396
11.3.2 浏览帮助文件	342	第13章 Visual C++6高级应用	398
11.3.3 创建内容文件 (*.cnt)	344	13.1 用Visual C++6开发多线程应用程序	398
11.4 帮助文件的使用	347	13.1.1 线程概述	398
11.4.1 WinHelp函数	347	13.1.2 线程创建与终止	398
11.4.2 在PrintView例程中使用 帮助文件	348	13.1.3 线程的优先级	399
11.4.3 Visual C++ 6扩展帮助功能	350	13.1.4 创建多线程程序	400
11.5 本章小结	351	13.1.5 线程通信与线程同步	405
第12章 动态链接库	353	13.2 用Visual C++6开发网络应用程序	408
12.1 动态链接库简介	353	13.2.1 WinInet开发简介	408
12.1.1 动态链接与静态链接	353	13.2.2 WinInet类概述	409
12.1.2 动态库的使用	354	13.3 本章小结	411

第1章 Visual C++ 6 概述

Microsoft公司最新推出的Visual Studio 6.0提供了众多的先进特性和强大工具，也是Visual系列开发软件包的最高版本，可完全满足现代企业机构创建企业级解决方案的要求。Visual Studio 6.0中的基于组件开发的工具可实现基于Windows DNA体系的应用方案。套件所提供的对整个生命周期的支持可在设计、开发、分析、管理和部署等各个阶段中得到全面体现。新的企业级数据库工具采用Universal Data Access数据访问策略，结合ActiveX Data Object接口，为用户实现高性能的企业级数据处理。而套件中包含的集成化Web项目模型、组件管理和源代码控制使开发组功能得到了增强。除此之外，套件能与诸如Microsoft BackOffice家族等基于Windows NT的众多应用程序服务集成，加速组件应用在Internet和Intranet的部署。

作为其核心开发工具之一，Visual C++ 6(有时为简便计，也称VC6.0)在网络应用开发方面增加了不少新的功能和特点，例如从简单的闲聊室文档，到复杂的网络数据库引擎、访问，等等。但是从界面上看，Visual C++ 6似乎没有太大的变化，这样大大方便了广大使用VC5.0的编程爱好者，使他们能够很快地熟悉Visual C++ 6开发环境。不过，考虑到初次接触Visual C++的读者并不熟悉以前版本的开发环境，因此本章首先将详细地介绍VC6.0开发环境，包括菜单栏、工具栏；然后再介绍和分析Visual C++ 6的新特点和功能。

1.1 Visual C++用户界面

首先，启动Visual C++6，进入Microsoft Developer Studio 6.0开发环境，如图1-1所示，启动的方法与任何Windows应用程序一样。

与其它Windows应用程序一样，最上面的对话框是Tip of the Day(每日一贴)，该对话框显示的是有关应用程序(即VC6.0)的小技巧。按“Close”按钮即可关闭，也可以按“Next Tip”按钮以进入下一个小技巧(Tip)；当然，还可以取消“Show tips at startup”选项，这样在下次启动VC6.0时将不会出现“每日一贴”对话框。

默认状态下，Microsoft Developer Studio 6.0开发环境只显示标题栏、菜单栏、工具栏、工作区窗口(Workspace)以及状态栏。

屏幕的最上端是标题栏，标题栏用于显示应用程序名和所打开的文件名；标题栏的下面是菜单栏和工具栏；工具栏的下面是两个窗口，左边是工作区(Workspace)窗口，右边是源代码编辑窗口(未显示，只有当打开一个工程源文件时才有)。其次，工作区窗口和源代码编辑窗口的下面是输出窗口，第一次启动时未显示；为此，选择“View/Output”菜单命令或按Alt+2快捷键，这样输出窗口就显示出来了，如图1-2所示。

输出窗口主要用于显示项目创建、搜索等过程中生成的进度信息，它有多个不同的属性窗口(当前显示的是Build窗口)。输出窗口的下面(即VC开发环境主窗口的最底端)是状态栏。比较完整的Visual C++ 6开发环境主窗口如图1-3所示。

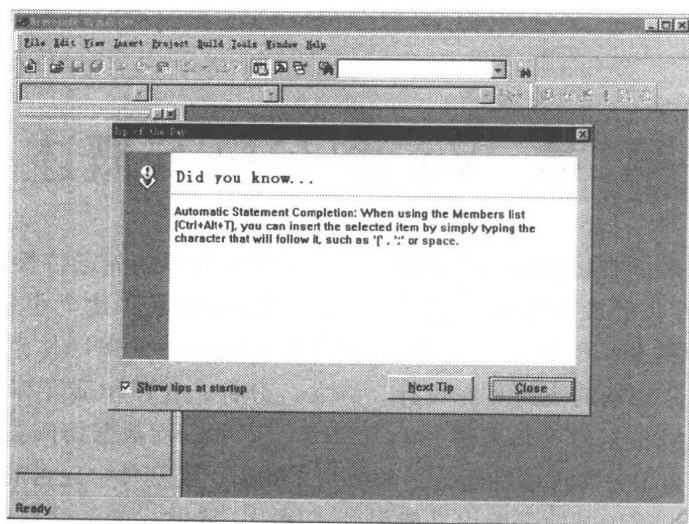


图1-1 Visual C++ 6 启动界面

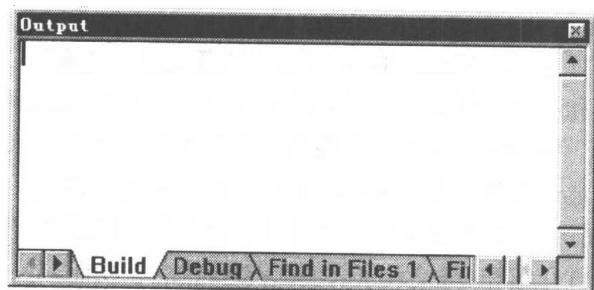


图1-2 输出(Output)窗口

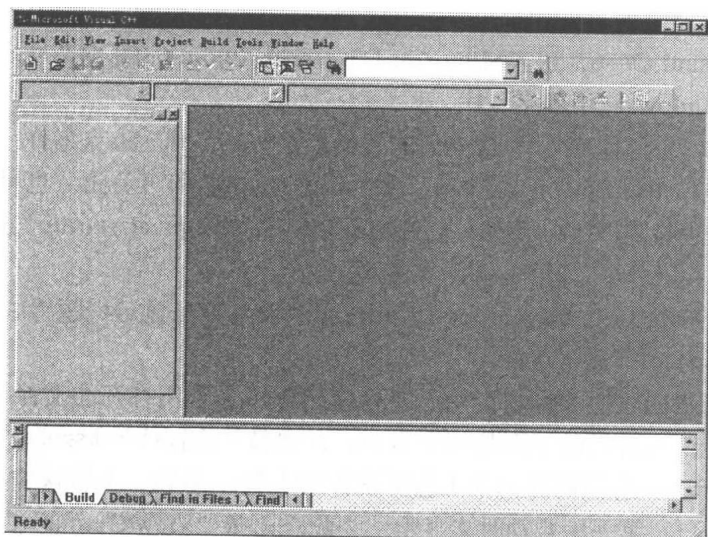


图1-3 Visual C++ 6 开发环境主窗口

与Visual C++ 5的界面不同的是, Visual C++ 6系统的联机帮助内容目录并不显示在工作区(Workspace)窗口。当用户按“Help/Contents”菜单命令时Visual C++ 6开发环境弹出一个联机帮助(即MSDN)窗口, 如图1-4所示。

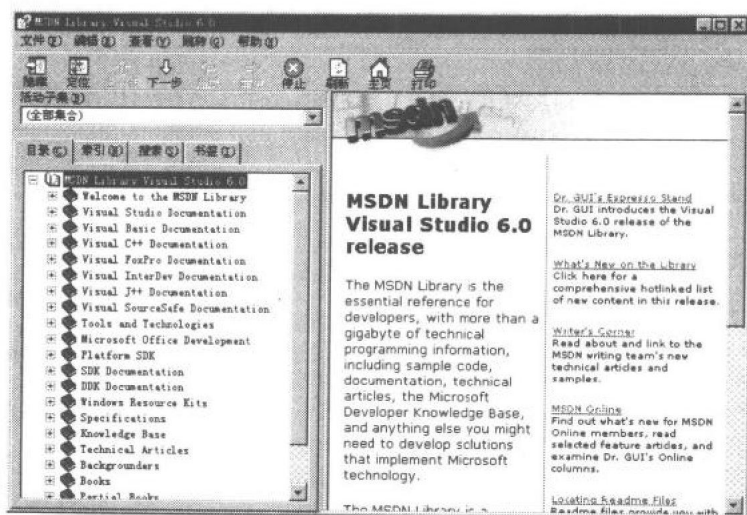


图1-4 Visual C++ 6 联机帮助窗口

在“活动子集”列表框中可以选择帮助主题查找的范围，该范围可以是安装在硬盘中的帮助文件，也可以是CD-ROM；紧挨着该列表框的下边是目录、索引等窗口，它们以属性标签(Tab)的形式集成在一起，其对应部分的使用方法与WinHelp.exe帮助窗口的相同。另外，图1-4所示窗口的右半部分显示的是当前查找的帮助信息。

1.2 Visual C++ 6工具栏

工具栏(ToolBar)对应于某些菜单选项或命令，可以直接用鼠标单击工具栏按钮来完成相应的命令。通常，在使用熟练之后使用工具栏按钮，比使用菜单项命令更加直接迅速。Visual C++ 6 包含10多种工具栏，在不同的工作环境下显示方式有所不同。在缺省状态下，屏幕上显示的工具栏是Standard(标准)、Build(创建)、Debug(调试)以及WizardBar(向导)等四个工具栏。

1.2.1 标准工具栏

Standard(标准)工具栏提供了常用的文件操作(如新建、打开、保存等)、编辑、搜索以及浏览帮助主题等工具按钮，如图1-5所示。



图1-5 Standard(标准)工具栏

标准工具栏各工具按钮的功能说明见表1-1。

表1-1 Standard工具栏按钮功能描述

按钮图标	按钮名称	功能描述
	New Text File(新文本文件)	创建新的文本文件
	Open(打开)	打开已有的文件
	Save(保存)	保存当前活动窗口内的文件

(续)

按钮图标	按钮名称	功能描述
	Save All(全部保存)	保存所有窗口的文件内容
	Cut(剪切)	将当前活动窗口内选定的内容复制到剪贴板中, 并删除选定的内容
	Copy(复制)	将当前活动窗口内选定的内容复制到剪贴板中
	Paste(粘贴)	在光标当前所在的位置插入剪贴板中的内容
	Undo(撤消)	取消最近一次的编辑修改操作
	Redo(重复)	重复Undo命令取消的操作
	Workspace(工作空间窗口)	显示或掩藏工作区窗口
	Output(输出窗口)	显示或掩藏输出窗口
	Window List(窗口列表)	管理当前打开的窗口列表
	Find in Files(在多个文件中查找)	在多个文件中查找字符串
	Find(查找)	在当前文件中查找字符串
	Search(搜索)	对应于Help菜单的Search命令, 它弹出一个帮助主题搜索窗口, 如图1-6所示

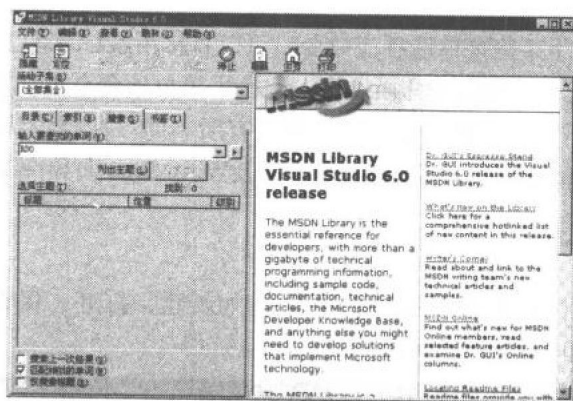


图1-6 Visual C++ 6 帮助主题搜索窗口

1.2.2 Build工具栏

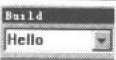
Build(创建)工具栏的工具按钮主要与Build(创建)菜单命令相对应; 除此之外, 还有Project菜单的一个命令, 即Set Active Project(设置当前工程), 如图1-7所示。



图1-7 Build(创建)工具栏

Build工具栏中各工具按钮的功能描述见表1-2。

表1-2 Build工具栏按钮功能描述

按钮图标	按钮名称	功能描述
	Set Active Project(设置当前工程)	选择指定的工程为工作区内活动的工程
	Set Active Configuration(当前配置)	选择活动工程的配置(Win32 Release或Win32 Debug)
	Compile(编译)	编译源代码窗口中的活动源文件
	Build(创建)	查看工程中的所有文件, 并对最近修改过的文件进行编译和链接
	Stop Build(停止创建)	停止创建工程
	Execute Program(运行程序)	运行应用程序
	Go(启动)	在调试过程中, 从当前语句启动或继续运行程序, 直到到达断点为止
	Insert/Remove Breakpoint(插入/删除断点)	插入或删除断点

1.2.3 Debug工具栏

Debug(调试)工具栏用于调试程序。当集成开发环境处于调试状态时, Debug工具栏处在突出位置, 如图1-8所示。该工具栏包括了Debug(调试)菜单栏的多数命令, 除此之外, 还有几个特有的工具按钮, 各工具按钮的功能描述见表1-3。

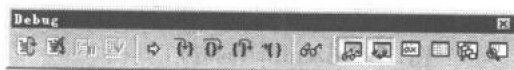


图1-8 Debug(调试)工具栏

表1-3 Debug工具栏按钮功能描述

按钮图标	按钮名称	功能描述
	Restart(重新运行)	从头开始对程序进行调试执行
	Stop Debugging(停止调试)	退出调试过程
	Break(暂停)	在当前位置暂停程序执行
	Apply Code Changes(代码修改)	该选项允许用户在调试程序的过程中修改代码
	Show Next Statement(显示下一行)	显示正在执行的代码行
	Step Into(进入函数)	单步执行程序, 当程序执行到某一函数调用语句时, 进入函数内部, 从函数的第一条语句开始单步执行
	Step Over(跳过函数)	单步执行程序, 当程序执行到某一函数调用语句时, 不进入函数内部
	Step Out(跳出函数)	与Step Into配合使用, 可以跳出Step Into进入的函数内部

(续)

按钮图标	按钮名称	功能描述
	Run to Cursor(光标位置)	使程序运行到光标所在位置停止
	Quick Watch(快速查看)	弹出Quick Watch对话框, 用来查看、修改变量和表达式, 如图1-28所示
	Watch(查看窗口)	隐藏/显示查看窗口的显示与否, 在该窗口中可以添加、修改、删除查看对象的值
	Variables(变量)	隐藏/显示当前变量(局部和全局)的数值
	Registers(寄存器)	隐藏/显示寄存器变量的当前值
	Memory(查看内存)	隐藏/显示内存的使用情况
	Call Stack(调用堆栈)	隐藏/显示堆栈调用的情况
	Disassembly(汇编)	隐藏/显示源程序的汇编代码

1.2.4 WizardBar工具栏

如图1-9所示, WizardBar(向导)工具栏的功能与ClassWizard(类向导)菜单命令类似, 可以说前者是后者的简便方式; WizardBar工具栏主要用来向该工程各派生类添加消息处理函数。

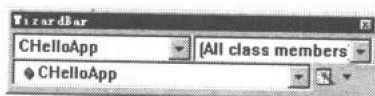


图1-9 WizardBar(向导)工具栏

Wizard Bar工具栏的各个窗口的功能(或作用)见表1-4。

表1-4 WizardBar工具栏按钮功能描述

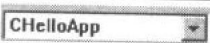
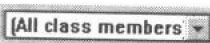
按钮图标	按钮名称	功能描述
	C++ Class(C++类)	可以在该列表框中选择一个应用程序所包含的(派生)类
	C++ Filter(C++过滤器)	可以在该列表框中选择添加成员函数的过滤器, 它直接影响C++ Members列表框中的选项
	C++ Members(C++成员函数)	可以在该列表框中选择需要添加的消息处理函数
	Actions(操作)	按下该按钮, 光标将停留在C++ Members列表框中所指定的消息处理函数模块里, 从而方便地添加代码
	Actions(下拉菜单操作)	按下该按钮, 将弹出一个下拉式菜单, 如图1-10所示

图1-10所示的下拉式菜单包含了一些常用的命令, 主要有利于快速编辑源程序。

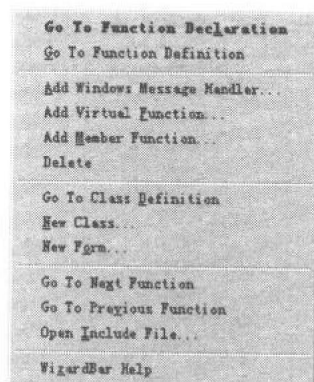


图1-10 WizardBar工具栏中的下拉式菜单

- Go To Function Declaration将插入符(Caret)移到函数声明位置。
- Go To Function Definition将插入符(Caret)移到函数定义体位置。
- Add Windows Message Handler添加窗口消息处理函数。
- Add Virtual Function添加MFC虚拟函数，即重载MFC虚拟函数。
- Add Member Function添加用户自定义的成员函数。
- Delete删除函数(如果插入符(Caret)当前所处位置在函数体内，被删除的将是该函数；如果插入符在函数体外，那么被删除的就是紧跟插入符位置的下一个函数；如果插入符(Caret)处在源文件末尾，那么该菜单命令不存在)。
- Go To Class Definition光标移到类定义位置。
- New Class创建一个新C++类。
- New Form创建一个新窗体。
- Go To Next Function将插入符(Caret)移到下一个函数位置(如果下一个函数不存在，则将插入符(Caret)移到该源文件的第一个函数位置)。
- Go To Previous Function同上，只是将插入符(Caret)移到前一个函数位置。
- Open Include File打开包含文件(只有当源文件包含了文件(即include语句)才可以；否则，将出现错误提示信息)，并且可以在随后的窗口中选择具体的包含文件。
- WizardBar Help向导工具栏联机帮助(此命令将弹出一个MSDN联机帮助窗口)。

1.3 Visual C++ 6 菜单栏

与Visual C++ 5.0一样，Visual C++6也采用停靠菜单栏(Windows的许多高版本软件都有此特性)。执行菜单命令的方法同标准Windows应用程序一致，这里不再重复。Visual C++6的菜单栏包括九个菜单，即File(文件)、Edit(编辑)、View(视图)、Insert(插入)、Project(工程)、Build(创建)、Tools(工具)、Window(窗口)以及Help(帮助)，如图1-11所示，下面分别对它们进行介绍。

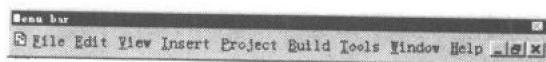


图1-11 Visual C++6的停靠菜单栏

1.3.1 File菜单

File(文件)菜单主要包含有关对文件或工程的创建、打开、保存以及源程序打印、打印预