

计算机软件开发与程序设计系列丛书



Visual C++ 6.0

编程实例与技巧

王 华 叶爱亮 祁立学 曹凌云 编著



机械工业出版社
China Machine Press

TP312
WH/1

计算机软件开发与程序设计系列丛书

Visual C++6.0编程 实例与技巧

王 华 叶爱亮 编著
祁立学 曹凌云



机械工业出版社
China Machine Press

050631

本书共分20章, 主要介绍了Visual C++集成开发环境、通用MFC编程、多媒体制作、数据库编程、多线程编程、网络编程、ActiveX控件以及一些常用编程方法和技巧等, 并配以大量例题。

本书读者对象是使用Visual C++开发工具的初、中级程序员。

图书在版编目(CIP)数据

Visual C++6.0 编程实例与技巧/王华等编著. - 北京: 机械工业出版社, 1999.4

(计算机软件开发与程序设计系列丛书)

ISBN 7-111-07082-8

I. V… II. 王… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(1999)第05210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: 陈剑甌

北京昌平第二印刷厂印刷·新华书店北京发行所发行

1999年4月第1版第1次印刷

787mm×1092mm 1/16·30.5印张

印数: 0 001-8 000册

定价: 48.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

微软开发工具套件Visual Studio 6.0企业版于1998年年底推出,其中包括了微软基于Windows和Web的“全面解决方案”:所有可视化开发工具、企业数据访问工具、生命周期开发工具和团队开发工具等。尤其是其中的Visual C++ 6.0,功能非常强大,支持面向对象编程、模块化、代码可重用、组件共享等技术,可以大大提高软件系统的设计、管理和开发的速度。

本书主要是针对使用Visual C++开发工具的初、中级读者而编写的。全书共分20章,主要介绍了VC++集成开发环境、通用MFC编程、多媒体制作、数据库编程、多线程编程、网络编程、ActiveX控件以及一些常用编程方法和技巧等。内容详实、新颖,语言简练,安排由浅入深,并且配以大量精练、完整的例程对知识点进行阐述,便于读者上机练习,逐步成为Visual C++ 6.0编程高手。

书中的例程是在Visual C++ 6.0集成开发环境中实现的。但同时也适合Visual C++4.0、Visual C++5.0等不同的版本。

本书是所有编著人员集体协作的结晶。参加本书编写工作的有王华、叶爱亮、祁立学、曹凌云。在本书的编写过程中,陈学林、张萍、韦蓉、张国强、麦云、谌廷政、梁勇、屠海蓉、刘淑霞等同志进行了细致的校对,并对例程作了验证。此外,张新贵、武斌、杨红兵、廖明景、王志清等同志在本书的编写过程中给予了大力的支持与帮助,在此表示衷心的感谢。

由于作者水平有限,书中难免存在错误和疏漏之处,恳请读者批评、指正,在此表示感谢。

编著者

1999年1月

目 录

前言

第1章 Visual C++6.0开发环境.....1

1.1 主框架窗口1

1.1.1 菜单栏2

1.1.2 工具栏12

1.1.3 工作区15

1.1.4 客户区15

1.1.5 输出窗口15

1.2 资源编辑器16

1.2.1 加速键编辑器16

1.2.2 对话框编辑器16

1.2.3 图标编辑器17

1.2.4 菜单编辑器17

1.2.5 串编辑器17

1.2.6 工具栏编辑器18

1.2.7 版本信息编辑器18

1.3 Visual C++ 实用工具19

1.3.1 Spy++19

1.3.2 MFC Tracer20

1.3.3 Error Lookup20

1.3.4 WinDiff20

1.3.5 ZoomIn22

1.3.6 ActiveX Control Test Container.....23

1.3.7 Help Workshop23

1.3.8 HTML Help Workshop23

1.4 常用操作方法23

第2章 应用程序基本框架26

2.1 用MFC AppWizard自动生成第一个

Windows应用程序26

2.1.1 工作区和工程26

2.1.2 生成应用程序框架26

2.1.3 工程的文件组成32

2.2 程序的运行过程和应用框架34

2.2.1 应用框架34

2.2.2 程序运行过程35

2.3 应用类36

2.3.1 InitInstance()函数37

2.3.2 Run()函数41

2.3.3 ExitInstance()函数42

2.3.4 OnIdle()函数42

2.4 边框窗口类42

2.4.1 文档边框窗口42

2.4.2 主边框窗口类42

2.4.3 子边框窗口类43

2.5 文档类和视图类44

2.5.1 文档模板类44

2.5.2 文档类45

2.5.3 视图类45

2.6 上述各个类的对象之间的关系46

2.6.1 上述对象的构造46

2.6.2 上述对象之间的访问47

第3章 菜单与加速键48

3.1 消息48

3.2 消息映射48

3.3 命令消息处理过程和更新用户接口

对象状态命令消息处理过程50

3.4 命令消息的传递路径51

3.5 菜单和加速键资源53

3.5.1 菜单53

3.5.2 键盘加速键54

3.6 添加菜单54

3.7 添加键盘加速键59

3.8 菜单项的状态更新60

第4章 对话框与控件63

4.1 消息对话框63

4.2 一般对话框的创建与使用流程65

4.3 例程S4_1: 使用对话框与控件65

4.4 创建对话框资源66

4.5 创建对话框类	68	6.3 绘图程序实例	134
4.5.1 类的生成	68	第7章 文本与字体	153
4.5.2 往程序中添加代码	69	7.1 CFont类	153
4.5.3 消息响应函数	70	7.2 字体应用程序之一	153
4.5.4 数据交换和校验	70	7.3 字体应用程序之二	157
4.6 模态对话框	71	7.4 字体应用程序之三	159
4.7 非模态对话框	73	第8章 打印	163
4.8 属性页对话框	74	8.1 打印原理	163
4.8.1 CPropertySheet类	74	8.2 CPrintInfo	164
4.8.2 CPropertyPage类	77	8.3 CPageSetupDialog	166
4.8.3 创建一般属性页对话框	79	8.4 例程S8_1	171
4.8.4 创建向导对话框	80	8.4.1 创建工程	171
4.9 通用对话框	83	8.4.2 增加资源	171
4.10 控件	85	8.4.3 实现打印功能	172
4.10.1 Picture控件	86	第9章 文件操作	176
4.10.2 Static Text控件	86	9.1 串行化	176
4.10.3 Edit Box控件	86	9.1.1 MPC程序框架支持的文档结构	177
4.10.4 Spin控件	87	9.1.2 CArchive类	179
4.10.5 Combo Box控件	88	9.1.3 直接使用串行化	182
4.10.6 Group Box控件	90	9.2 CFile类	182
4.10.7 Radio Button控件	90	9.2.1 打开和关闭文件	182
4.10.8 Check Box控件	90	9.2.2 文件的定位	183
4.10.9 Button控件	96	9.2.3 文件的读写操作	183
4.10.10 Slider控件	98	9.2.4 文件的管理	184
4.10.11 Horizontal Scroll Bar与Vertical Scroll Bar控件	101	9.2.5 文件异常	184
4.10.12 Tab控件	104	9.3 文件对话框	184
4.10.13 Tree控件	107	9.3.1 使用CFileDialog	185
4.10.14 List控件	111	9.3.2 选择多个文件	186
4.10.15 Progress控件	114	9.3.3 设计自己所需的文件对话框	186
4.10.16 Animate控件	117	9.4 内存文件类	186
第5章 控件栏	119	9.4.1 CMemFile类	186
5.1 工具栏	119	9.4.2 CSharedFile类	187
5.2 例程S5_1	120	9.5 文件输入/输出流	188
5.3 状态栏	126	9.6 内存映射文件	188
5.4 对话框栏	129	9.6.1 创建和打开内存映射文件	188
第6章 绘图	132	9.6.2 使用内存映射文件	189
6.1 DC和CDC类	132	9.6.3 使用内存映射文件共享数据	190
6.2 GDI和CGDIObject类	133	9.6.4 VLM的内存映射文件操作	191
		9.7 文件操作例程	191

9.7.1 创建程序框架和使用框架支持的串行化操作	192	11.4.2 增加、修改资源	258
9.7.2 使用CStdioFile类	199	11.4.3 增加CCrack类	260
9.7.3 使用CFile、CArchive类和内存映射文件	201	11.4.4 实现数据库功能	268
第10章 多媒体制作	205	第12章 多线程	286
10.1 声音	205	12.1 多线程概述	286
10.1.1 两个例子	205	12.2 用户界面线程	287
10.1.2 播放声音的三个简单函数	207	12.2.1 创建用户界面线程	287
10.1.3 用MCI控制波形声音的播放	209	12.2.2 启动用户界面线程	288
10.2 图像	223	12.2.3 使用用户界面窗口	289
10.2.1 Windows的BMP文件结构	224	12.3 工作者线程	289
10.2.2 显示DIB的程序实例	225	12.3.1 实现控制函数	289
10.3 视频	231	12.3.2 启动工作者线程	290
第11章 数据库编程	234	12.4 线程的终止	290
11.1 ODBC与DAO	234	12.4.1 正常终止	290
11.2 数据库操作	234	12.4.2 异常终止	291
11.2.1 异常处理	234	12.4.3 几种终止函数的比较	291
11.2.2 新建数据库	235	12.4.4 获取线程的退出码	292
11.2.3 打开数据库	235	12.5 线程的管理和操作	292
11.2.4 关闭数据库	236	12.5.1 优先级	292
11.2.5 新建表	236	12.5.2 控制运行状态	293
11.2.6 删除表	237	12.6 线程间通讯	293
11.2.7 打开表	237	12.6.1 使用同步类控制访问共享数据	293
11.2.8 修改表结构	238	12.6.2 使用消息队列	293
11.2.9 插入记录	238	12.7 使用同步类	294
11.2.10 删除记录	240	12.7.1 同步对象	294
11.2.11 修改记录	240	12.7.2 同步访问对象	296
11.3 DAO类	241	12.8 多线程的调试	297
11.3.1 CDaoWorkspace	241	12.9 多线程例程	299
11.3.2 CDaoDatabase	244	第13章 网络编程	311
11.3.3 CDaoRecordset	246	13.1 网络应用程序	311
11.3.4 CDaoQueryDef	251	13.1.1 网络应用程序的结构	311
11.3.5 CDaoTableDef	253	13.1.2 建立socket对象	311
11.3.6 CDaoRecordView	255	13.1.3 建立连接	312
11.3.7 CDaoFieldExchange	256	13.1.4 发送和接收数据	312
11.3.8 CDaoException	257	13.1.5 监听socket	312
11.4 例程S11_1	258	13.1.6 为客户端连接请求建立socket队列	312
11.4.1 创建工程	258	13.2 WinSock类	313
		13.2.1 CAsyncSocket	313
		13.2.2 CSocket	317

13.3 客户端应用程序例程S13_1	318	14.9.1 DLL的使用	346
13.3.1 创建工程	318	14.9.2 如何调试DLL	347
13.3.2 增加、修改资源	319	14.10 DLL例程	348
13.3.3 增加CRequestSocket类	320	14.10.1 S14_DLL例程	348
13.3.4 实现网络功能	322	14.10.2 S14_APP应用程序例程	354
13.4 服务器端例程S13_2	326	第15章 组件	362
13.4.1 创建工程	326	15.1 添加组件的一般方法	362
13.4.2 增加、修改资源	326	15.2 例程S15_1	363
13.4.3 增加CListeningSocket类	327	15.3 空闲时间处理(Idle time processing)	363
13.4.4 增加CClientSocket类	328	15.4 往视图图中添加弹出式菜单(Pop-up Menu)	364
13.4.5 实现网络功能	329	15.5 制作软件封面(Splash screen)	366
第14章 动态链接库	333	15.6 切分视图窗口(Splitter Bar)	367
14.1 DLL基础知识	333	15.7 显示系统信息(System Info)	368
14.1.1 DLL概述	333	15.8 在软件启动时显示小提示(Tip of the day)	370
14.1.2 MFC中的DLL	333	15.9 给控件添加小提示(ToolTip)	372
14.2 DLL入出口函数	334	第16章 ActiveX控件	373
14.2.1 DllMain函数	334	16.1 往工程中添加ActiveX控件	373
14.2.2 MFC AppWizard生成的Regular DLL 的入出口	335	16.2 例程S16_1	374
14.3 从DLL中导出函数	335	16.2.1 创建工程	374
14.3.1 使用DEF文件导出函数	336	16.2.2 往工程中添加FlexGrid控件	374
14.3.2 使用关键字-declspec(dllexport)	337	16.2.3 使用FlexGrid控件	375
14.3.3 使用AFX_EXT_CLASS导出	337	16.3 自定义ActiveX控件	378
14.4 DLL中的数据 and 内存	338	16.4 例程S16_2	379
14.4.1 从DLL中导出数据	338	16.4.1 创建工程	379
14.4.2 多个进程共享DLL中的数据 and 内存	339	16.4.2 测试ActiveX控件	379
14.5 DLL中的资源	340	16.4.3 注册/取消注册ActiveX控件	381
14.6 程序链接	341	16.4.4 ActiveX控件的版本	381
14.6.1 隐式链接	341	16.4.5 定制ActiveX控件的图标	381
14.6.2 显式链接	342	16.4.6 定制ActiveX控件的界面	382
14.7 Visual C++6.0自动生成的Win32 DLL	342	16.4.7 添加栈属性	383
14.7.1 空的DLL工程	343	16.4.8 添加自定义属性	384
14.7.2 简单的DLL工程	343	16.4.9 添加栈事件	388
14.7.3 导出了变量、函数和类的DLL	343	16.4.10 添加自定义事件	388
14.8 MFC AppWizard生成的DLL	344	16.4.11 添加栈方法	389
14.8.1 静态链接MFC的Regular DLL	344	16.4.12 添加自定义方法	390
14.8.2 动态链接MFC的Regular DLL	344	第17章 帮助的制作	391
14.8.3 动态链接的MFC扩展DLL	345	17.1 HLP文件类型的帮助	391
14.9 DLL的使用和调试	346	17.1.1 RTF文件	392
		17.1.2 Help Workshop	395

17.1.3 例程S17_1	395	20.4 如何将工程中的MFC类添加到 ClassWizard中	444
17.1.4 WinHelp函数	399	20.5 Visual C++命名规则	445
17.2 HTML	402	20.6 如何使工程链接时链入某些库文件	446
17.2.1 HTML	402	20.7 如何查看编译、链接的出错信息	447
17.2.2 例程S17_2	405	20.8 如何在应用程序中使用中文	447
17.3 CHM类型的帮助	409	20.9 如何编写DOS控制台程序和C语言程序	447
17.3.1 HTML Help Workshop	411	20.10 如何在应用程序中启动其他应用程序	449
17.3.2 例程S17_3	412	20.10.1 调用system函数	449
17.3.3 HtmlHelp函数	414	20.10.2 调用WinExec函数	449
第18章 MFC常用类	415	20.10.3 调用ShellExecute函数	450
18.1 CArray	415	20.10.4 调用CreateProcess函数	450
18.2 CPoint	417	20.11 如何退出应用程序	451
18.3 CRect	419	20.12 用内存映射文件实现多个程序间 互斥运行	452
18.4 CSize	422	20.13 如何禁止运行一个程序的多个实例	453
18.5 CString	423	20.14 如何在程序中将其他窗口程序调至前台	454
18.6 CStringArray	429	20.15 设计通知栏图标程序	455
18.7 CTime	431	20.16 如何使MDI应用程序在刚启动时不 建立一个新文件	456
第19章 安装/卸载程序	434	20.17 如何删除指定目录及其目录下的所 有文件和子目录	457
19.1 安装/卸载程序	434	20.18 如何使用定时器	458
19.2 例程S19_1	434	20.19 如何改变对话框的颜色	458
19.2.1 InstallShield向导	434	20.20 如何改变控件的颜色	459
19.2.2 Scripts标签	436	20.21 如何获取对话框中控件的窗口指针 并进行一些常用操作	459
19.2.3 Components标签	436	20.22 如何制作图标按钮和位图按钮	459
19.2.4 Setup Types标签	438	20.23 如何明确对话框的UpdateData()函 数的参数含义	460
19.2.5 Setup Files标签	439	20.24 如何使对话框中的两个Edit Box控 件的内容相关	460
19.2.6 File Groups标签	439	20.25 如何从其他应用程序中“借用”资源	461
19.2.7 Resources标签	440	20.26 如何使用CDC类进行裁剪输出	461
19.2.8 Media标签	441	20.27 设置视图类的空背景画刷, 提高显 示刷新速度	462
19.2.9 程序的卸载过程	441	20.28 使用内存兼容位图对象绘制, 提高显 示刷新速度	463
第20章 方法与技巧集锦	443	20.29 实现全屏显示	464
20.1 Visual C++常用操作技巧	443		
20.1.1 打开和编辑二进制文件	443		
20.1.2 检测括号是否匹配	443		
20.1.3 检测#if语句和#endif语句是否匹配	443		
20.1.4 删除工程的中间文件以便于备份 工程	443		
20.1.5 快速显示当前函数的有关信息	443		
20.2 如何同时操作两个或两个以上的工程	444		
20.3 如何优化工作区中的树目录	444		

20.30 如何在应用程序的整个窗口中获取鼠标的消息	466	20.32.3 DeleteGroup	473
20.31 如何使用注册表	467	20.32.4 DeleteItem	473
20.31.1 RegCreateKeyEx	468	20.32.5 ExitProgman	473
20.31.2 RegOpenKeyEx	470	20.32.6 Reload	474
20.31.3 RegQueryValueEx	470	20.32.7 ReplaceItem	474
20.31.4 RegSetValueEx	471	20.32.8 ShowGroup	474
20.31.5 RegDeleteKey	472	20.33 如何创建ODBC数据源	474
20.32 如何通过DDE利用Progman.exe创建程序组和程序项	472	20.34 如何在Windows NT下直接对并行口进行操作	475
20.32.1 AddItem	472	20.35 如何查询和设置系统参数信息	476
20.32.2 CreateGroup	473	20.36 如何获取系统的度量信息和相关的配置信息	476

第1章 Visual C++6.0开发环境

学习Visual C++ 6.0，首要的工作就是熟悉其开发环境。

Visual C++ 6.0秉承Visual C++以前版本的优异特性，为用户提供了一套良好的可视化开发环境，主要包括文本编辑器、资源编辑器、工程创建工具、Debugger调试器等。用户可以在集成开发环境中创建工程，建立、打开和编辑文件、编译、链接、运行、调试应用程序。

青出于蓝而胜于蓝的是，Visual C++ 6.0在用户界面、编译调试技术以及语言特性等方面都比其上一代有进一步的改进和发展。其中尤其值得一提的是Visual C++ 6.0的智能感应技术，该技术可以根据编辑状态下程序代码的输入情况，自动将属性、参数、数据类型和代码信息显示在列表框或文本框中，编程人员可以选择以自动完成输入或得到提示，从而大大减少访问联机帮助的次数，提高了编程效率。

本章的工作就是带领读者领略、熟悉Visual C++ 6.0的开发环境，以便读者在创建应用程序时能充分利用其强大的功能，从而达到事半功倍的效果。

对于初学Visual C++的读者来说，可能暂时用不到本章介绍的部分内容，所以在阅读这些地方的时候可以跳过去。等有了一些编程经验以后，再回过头来阅读本章，可能收获会更多一些。

1.1 主框架窗口

在Windows 95/98/NT 环境中运行Microsoft Visual C++ 6.0(主应用程序为Msdev.exe)后，即进入C++语言的集成开发环境。程序运行的主界面见图1-1。

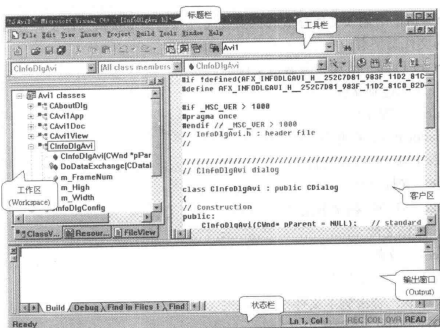


图1-1 Microsoft Visual C++ 6.0 集成开发环境

Visual C++ 6.0 集成开发环境的主框架窗口由标题栏、菜单栏、工具栏、工作区、客户区、输出窗口和状态栏等组成。主框架窗口的最上端为标题栏，用于显示应用程序名和当前打开的文件名，标题栏下面通常为菜单栏和工具栏。工具栏的下面一般为工作区和客户区。工作区用于显示当前打开工程的有关信息，包括工程的类、资源以及文件组成等内容。客户区用于文本编辑器、资源编辑器等进行文件或资源的编辑。输出窗口通常在工作区和客户区的下面，用于输出编译信息、调试信息和一些查询结果信息等。主框架窗口的最底端为状态栏，用于显示关于菜单栏、工具栏等的简单说明信息以及文本编辑器中当前光标所在的行列号等。

要注意的是，用户可以利用鼠标通过拖动，将菜单栏、工具栏、工作区、客户区和输出窗口等随意更换顺序和位置，也可以隐藏其中的某些窗口。

1.1.1 菜单栏

菜单栏由多个子菜单组成，可以通过鼠标或键盘选择子菜单。选中子菜单后，即弹出下拉式菜单，可以选择其中的菜单项以执行相应的菜单命令。有的菜单项提供了加速键，在相应菜单项的右边有加速键的名称提示。在开发环境中，可以用加速键直接执行相应的菜单命令。

下面对一些常用的、比较重要的菜单命令进行说明。对于一些通常通过工具栏实现的命令将在后面工具栏一节中进行说明。

1.1.1.1 File菜单

1. File | New...

File | New...菜单用于新建文件、工程、工作区和文档。选中File | New...菜单，将弹出New对话框，包含4个标签：Files标签、Projects标签、Workspaces标签和 Other Documents标签。

(1) Files标签 通过Files标签(见图1-2)可以创建多种类型的文件。如果要创建新文件，选中所要创建的文件类型，在File文本框中输入文件名，单击OK按钮即完成了新文件的创建。如果想将新文件加入当前已经打开了的工程，只需选中Add to project复选框并指定工程名。可以在Location编辑框中输入文件存放的位置或通过点击浏览按钮选择文件存放的位置。

可以创建的文件类型有：

- Active Server Page: 活动服务器页文件
- Binary File: 二进制文件
- Bitmap File: 位图文件
- C/C++ Header File: C或C++头文件
- C++ Source File: C++源文件
- Cursor File: 光标文件
- HTML Page: HTML超文本文件
- Icon File: 图标文件
- Macro File: 宏文件
- SQL Script File: SQL脚本文件
- Resource Script: 资源脚本文件
- Resource Template: 资源模板文件

• Text1 File: 文本文件

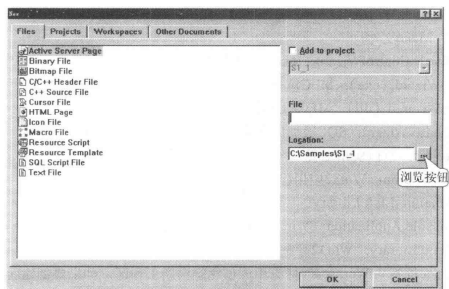


图1-2 New对话框Files标签

(2) Projects 标签 Projects 标签（见图1-3）用于创建多种类型的工程。如果要创建新工程，选中要创建的工程类型，在Project name文本框中输入工程名。如果想将新工程加入打开的工作区中，可以选择Add to current workspace，否则将创建包含新工程的新工作区。如果要使新工程成为已有工程的子工程，请选中Dependency of复选框并指定工程名。可以在Location文本框中指定工程文件存放的位置。

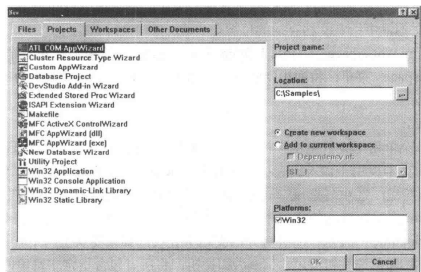


图1-3 New对话框Projects标签

可以创建的工程有：

- ATL COM AppWizard: ATL应用程序
- Cluster Resource Type Wizard: 资源动态链接库及超级扩展动态链接库
- Custom AppWizard: 自定义AppWizard
- Database Project: 数据库工程
- DevStudio Add-in Wizard: 自动化宏

- Extended Stored Proc Wizard: 用于访问SQL Server的动态链接库
- ISAPI Extension Wizard: Internet服务器或过滤器
- Makefile: Make文件
- MFC ActiveX Control Wizard: ActiveX控件程序
- MFC AppWizard (exe): MFC应用程序
- MFC AppWizard (dll): MFC动态链接库
- New Database Wizard: SQL数据库服务器
- Utility Project: 空白实用工程
- Win32 Application: Win32应用程序
- Win32 Dynamic-Link Library: Win32动态链接库
- Win32 Console Application: Win32控制台应用程序
- Win32 Static Library: Win32静态库

(3) Workspace标签 Workspace标签用于创建空白新工作区。如果要创建新工作区, 选择Blank Workspace, 在Workspace name文本框中输入工作区名, 单击OK按钮即完成了空白新工作区的创建。可以在Location文本框中指定工作区存放的位置。

(4) Other Documents标签 Other Documents标签用于创建一些其他类型的文档。比如, 如果机器中安装了Microsoft Office, 则在该标签下就会列出可以创建的文档类型有: Microsoft Excel工作表、Microsoft Excel图表、Microsoft PowerPoint演示文稿和Microsoft Word文档。

如果要创建新文档, 选择其中的一种类型, 在File文本框中输入文档名, 单击OK按钮即完成了新文本的创建。如果想将新文档加入打开的工程, 只需选中Add to project复选框即可。可以在Location文本框中指定文档存放的位置。

2. File | Open Workspace... File | Open Workspace...菜单用于打开工作区或工程。选中File | Open Workspace...菜单, 将弹出Open Workspace对话框, 文件过滤器为 (*.dsw, .mdp)。

1.1.1.2 Edit菜单

1. Edit | Find...

Edit | Find...菜单用于查找指定字符串。选中Edit | Find...菜单, 将弹出Find对话框 (见图1-4)。Find What文本框用于输入要查找的字符串; 选中复选框Match whole word only表示查找时按整个单词匹配; 选中复选框Match case表示查找时字母要按大小写匹配; 选中复选框Regular expression表示要查找的字符串中可以有通配符、可用的通配符见表1-1; 选中复选框Search all open documents表示在所有打开的文档中查找; 如果要从上向下查找, 单击Down单选按钮, 否则单击Up单选按钮; 单击Mark All按钮, 将所有包含指定字符串的行标记出来。

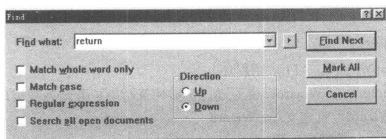


图1-4 Find对话框

5. Edit | Bookmarks...

Edit | Bookmarks...菜单用于在文件中定义书签。选中菜单Edit | Bookmarks..., 将弹出Bookmarks对话框, 其中的Name文本框用于输入书签名。

6. Edit | Advanced...

Edit | Advanced...菜单中包括下面的一些命令:

(1) Incremental Search: 增值查询, 选中后, 对任意位置出现的变量和对象均提示较详细的说明。

(2) Format Selection: 格式化选定区域的内容。

(3) Tabify Selection: 用TAB符号替换选定区域内的空格。

(4) Untabify Selection: 用空格替换选定区域内的TAB符号。

(5) Make Selection Uppercase: 使选定区域内的所有字符大写。

(6) Make Selection Lowercase: 使选定区域内的所有字符小写。

(7) View Whitespace: 显示或隐藏活动窗口的空格和TAB符号。

7. Edit | Breakpoints...

Edit | Breakpoints...菜单用于设置和删除断点。选中Edit | Breakpoints...菜单, 将弹出Breakpoints对话框, 对话框包含3个标签: Location标签、Data标签和Message标签。

(1) Location标签 用于设置位置断点, Break at文本框用于输入位置, 包括源代码的行号、函数的起始或指定的内存地址。当程序执行到指定位置时, 执行位置断点中断程序。设置好位置断点后, 可以继续设置程序中断条件。这时只有程序执行到指定位置且程序中断条件为真时, 程序才执行中断程序。

(2) Data标签 用于设置数据断点, Enter the expression to be evaluated文本框用于输入表达式或变量。当表达式或变量的值改变时, 执行数据断点中断程序。

(3) Message标签 用于设置消息断点, Break at WndProc文本框用于输入窗口过程的句柄, Set one breakpoint for each message at文本框用于输入或选定消息。当窗口过程收到指定消息时, 执行消息断点中断程序。

1.1.1.3 View菜单

1. View | ClassWizard...

在集成开发环境中, 通过View | ClassWizard...菜单可以打开MFC ClassWizard对话框, 实现类向导的功能, 用于建立消息映射函数、为控件引入成员变量等。MFC ClassWizard对话框包含5个标签: Message Maps标签、Member Variables标签、Automation标签、ActiveX Events标签和Class Info标签。

(1) Message Maps标签 Message Maps标签(见图1-7)用于建立消息映射函数, Project组合框用于选定工程名, Class name组合框用于选定要建立消息映射的类名, Object IDs 列表框用于选定对象的标识号(ID), 鼠标左键双击Messages组合框选定对象的消息处理函数。也可以在Member functions列表框中选定消息处理函数, 用Delete Function按钮删除。

(2) Member Variables标签 Member Variables标签(见图1-8)为控件引入成员变量。Project组合框用于选定工程名, Class name组合框用于选定类名, 鼠标左键双击Control IDs 列表框中的控件可以为该控件引入成员变量。也可以用Delete Variable按钮删除为控件引入的成员变量。

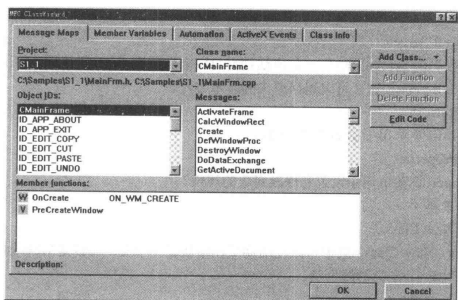


图1-7 ClassWizard对话框Message Maps标签

- (3) Automation标签 Automation标签用于添加自动化方法和属性到用户界面。
 (4) ActiveX Events标签 ActiveX Events标签用于添加ActiveX事务处理。

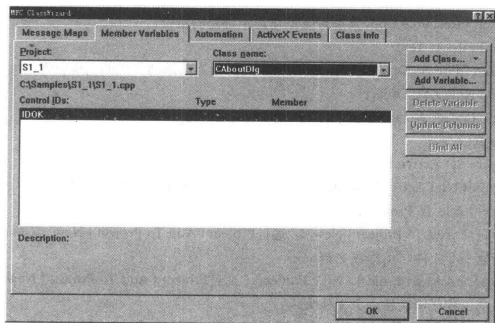


图1-8 ClassWizard对话框Member Variables标签

- (5) Class Info标签 Class Info标签用于显示指定类的基本信息。

2. View | Resource Symbols...

通过View | Resource Symbols...菜单可以打开Resource Symbols对话框。在Resource Symbols对话框中,可以浏览、新增和编辑资源标识号,也可以删除未使用的资源标识号。此外,还可以直接定位到具体使用标识号的地方。

3. View | Resource Includes...

View | Resource Includes...菜单用于修改资源头文件名,编辑预处理代码和修改包含的头