



青松电脑系列丛书

Visual C++6.0 使用与开发

木林森 高峰霞 罗丽琼 奚红宇 编著



5.0



清华大学出版社

Visual C++ 6.0 使用与开发

木林森 高峰霞 罗丽琼 奚红宇 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书对 Visual C++ 6.0 的使用与开发作了详细介绍。全书内容共分 10 章,第 1 章介绍了 Visual C++ 6.0 开发环境的特点和构成;第 2 章对 C++ 语言的基本内容作了简要介绍;第 3 章至第 5 章详细阐述了如何运用 MFC 来开发 Windows 应用程序,这是本书最基本、最核心的内容;其余各章分别围绕一个主题论述,力求突出实用性,内容涉及到多线程、动态链接库、ActiveX 编程、数据库编程和 Internet 编程。为了帮助读者加深和巩固对每一部分的理解,作者精心设计了大量的编程实例,并将这些实例进行了严格的调试和运行。

本书内容新颖、语言浅显、通俗易懂,适用于 Visual C++ 初、中级程序设计人员学习使用,也可供大专院校师生阅读。

版权所有,翻印必究。本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Visual C++ 6.0 使用与开发
作 者: 木林森 高峰霞 罗丽琼 奚红宇 编著
出 版 者: 清华大学出版社(北京 清华大学校内,邮政编码: 100084)
因特网址: <http://www.tup.tsinghua.edu.cn>
责任编辑: 刘小峰
印 刷 者: 北京市清华园胶印厂
发 行 者: 新华书店总店北京科技发行所
开 本: 787×1092 1/16 印张: 23.75 字数: 563 千字
版 次: 1998 年 12 月第 1 版 1999 年 4 月第 2 次印刷
书 号: ISBN 7-302-03227-0/TP·1726
印 数: 10001—21000
定 价: 40.00 元

前 言

进入 90 年代以来,计算机可视化技术得到了迅猛发展并引起了广泛重视,越来越多的计算机专业人员和非专业人员开始研究并应用可视化技术。

美国微软公司推出的 Visual C++ 是目前使用极为广泛的可视化开发环境,Visual C++ 6.0 是当前的最新版本。作为可视化开发的强有力环境,Visual C++ 6.0 必将带给您一种全新的感受,使您工作起来更加简便快捷,帮助您梦想成真。

本书对 Visual C++ 6.0 使用与开发所应掌握的基本内容和方法作了全面系统的介绍。全书内容共分 10 章:

- 第 1 章介绍了 Visual C++ 6.0 的开发环境。
- 第 2 章对 C++ 语言的基本内容作了简要介绍,重点阐述了类、对象、类继承、重载和多态性等概念。
- 第 3 章介绍了 Windows 应用程序的编写方法,内容涉及到了 AppWizard 的使用、Windows 应用程序的构成、文档模板、窗口、消息与命令的处理、ClassWizard 和 WizardBar 的使用等。
- 第 4 章对边框窗口、文档和视图以及三者之间的关系作了介绍。
- 第 5 章介绍了对话框和控件,并对属性表、属性页和表单视图作了简要介绍。
- 第 6 章介绍了多线程。
- 第 7 章介绍了动态链接库。
- 第 8 章介绍了 ActiveX 编程,内容涉及到了容器、服务器、拖放、剪贴板、自动化、ActiveX 文档、ActiveX 控件以及 ActiveX 模板库(ATL)等。
- 第 9 章介绍的是数据库编程,重点介绍 ODBC 编程,并对 DAO,OLE DB 和 ADO 等也作了简要介绍。
- 第 10 章介绍了 Internet 编程,内容涉及 WinInet,ISAPI,异步 Monikers,MAPI 和 WinSock 等。

本书主要由木林森、高峰霞、罗丽琼和奚红宇负责编写,高星泉仔细审阅了全书。参与本书编写工作的还有高越明、王世航、叶伟东和王志伟。本书的录排工作由郭静负责。

由于时间仓促,加之水平有限,不当之处在所难免,尚望读者批评指正。

作 者

1998年10月

目 录

第 1 章 Visual C++ 6.0 环境	1
1.1 环境构成	1
1.1.1 工具栏	2
1.1.2 选单栏	3
1.1.3 快捷选单	11
1.2 项目与项目工作区	11
1.2.1 工作区窗口	12
1.2.2 项目配置	14
1.2.3 多项目	14
1.2.4 转换早期的 Make 文件	15
1.3 资源管理	15
1.3.1 资源管理器	15
1.3.2 资源符号	16
1.3.3 对话框编辑器	19
1.3.4 选单编辑器	21
1.3.5 加速键编辑器	23
1.3.6 串表编辑器	24
1.3.7 版本信息编辑器	25
1.3.8 图形编辑器	26
1.3.9 工具栏编辑器	29
第 2 章 C++ 语言基础	31
2.1 简单的 C++ 程序	31
2.2 标识符	32
2.3 基本数据类型	32
2.3.1 类型 char	32
2.3.2 类型 short	33
2.3.3 类型 int	33
2.3.4 类型 long	33
2.3.5 类型 _intn	34
2.3.6 浮点类型	34

2.3.7	常量	34
2.4	数据类型转换	34
2.5	C++ 存储类	35
2.5.1	auto 存储类	35
2.5.2	register 存储类	35
2.5.3	extern 存储类	36
2.5.4	static 存储类	36
2.6	基本运算	37
2.7	自定义数据类型	37
2.7.1	typedef	37
2.7.2	结构	38
2.7.3	联合	39
2.7.4	枚举	39
2.8	控制结构	40
2.8.1	条件语句	40
2.8.2	循环语句	42
2.8.3	转移语句	43
2.9	数组、字符串和指针	44
2.9.1	数组	44
2.9.2	字符串	45
2.9.3	指针	45
2.10	函数	46
2.11	类与对象	47
2.11.1	类的定义	48
2.11.2	对象的定义	50
2.11.3	构造函数与析构函数	51
2.11.4	内联成员函数	53
2.11.5	对象与成员函数	54
2.11.6	this 指针	55
2.11.7	静态数据成员和成员函数	55
2.11.8	友元函数	56
2.12	类继承	58
2.12.1	派生类与基类	58
2.12.2	派生类的对象和构造函数	59
2.12.3	多重派生类	60
2.12.4	类的多继承	62
2.12.5	多层派生类	64
2.13	重载	66

2.13.1	函数重载	66
2.13.2	操作符重载	68
2.14	多态性	71
2.14.1	虚拟函数	71
2.14.2	纯虚拟函数	74
2.15	I/O 流结构	75

第 3 章 编写 Windows 应用程序 77

3.1	使用 AppWizard 生成并建立应用程序	77
3.2	应用程序的类和源文件	86
3.2.1	文档类	86
3.2.2	视图类	86
3.2.3	主边框窗口类	86
3.2.4	子边框窗口类	87
3.2.5	应用程序类	87
3.2.6	其他有关的文件	87
3.3	应用程序的控制流程	88
3.3.1	InitInstance	88
3.3.2	消息循环	90
3.3.3	ExitInstance	90
3.3.4	OnIdle	91
3.3.5	Shell 注册	91
3.3.6	拖放功能	91
3.4	文档模板	91
3.5	窗口类和窗口对象	93
3.5.1	窗口对象	93
3.5.2	CWnd 派生的窗口类	93
3.5.3	注册窗口类	94
3.5.4	窗口的创建过程	95
3.5.5	销毁窗口	94
3.6	消息与命令	95
3.6.1	CCmdTarget 类	95
3.6.2	消息与消息处理	95
3.6.3	消息的种类	96
3.6.4	发送和接收消息	100
3.6.5	如何搜索消息映射	102
3.6.6	手工创建消息映射	104
3.6.7	用户界面对象的更新	104

3.7	ClassWizard 和 WizardBar 的使用	106
3.7.1	WizardBar 的使用	106
3.7.2	ClassWizard 的使用	107
3.8	CObject 类	110
3.8.1	串行化	111
3.8.2	运行时类信息	112
3.8.3	动态创建	112
3.8.4	调试支持	112
第 4 章	边框窗口、文档和视图	115
4.1	边框窗口	115
4.1.1	边框窗口和视图	115
4.1.2	主边框窗口与文档边框窗口	116
4.1.3	边框窗口的创建和销毁	117
4.1.4	管理当前视图	117
4.1.5	管理选单、控件栏和加速键	117
4.1.6	定制边框窗口	118
4.2	文档和视图	119
4.2.1	文档和视图的关系	119
4.2.2	使用文档管理数据	119
4.2.3	在视图中显示数据及与用户交互	122
4.2.4	多文档类型	123
4.2.5	多视图	123
4.2.6	MFC 各对象之间的关系	124
4.3	派生的视图类	124
4.4	滚动和缩放显示	125
4.4.1	滚动	125
4.4.2	缩放显示	125
4.5	分割窗口	126
4.6	快捷选单	126
4.7	绘图	127
4.7.1	设备文本	128
4.7.2	CPoint, CSize 和 CRect	128
4.7.3	绘图工具	128
4.7.4	CPalette 与调色板	132
4.7.5	CRgn 与区域	132
4.7.6	绘图函数	133
4.7.7	映射模式	137

4.7.8	绘图模式	140
4.7.9	设置背景颜色	141
4.7.10	设置背景模式	141
4.8	文本处理	141
4.8.1	文本输出函数	141
4.8.2	设置文本属性	142
4.8.3	获取字符属性	143
4.8.4	使用字体	145
4.8.5	文本处理范例	147
4.9	位图和位操作	148
4.9.1	装载预定义位图	148
4.9.2	从资源中装载位图	148
4.9.3	使用绘图函数建立位图	148
4.9.4	执行位操作	149
4.10	图标和光标	152
4.10.1	图标	152
4.10.2	光标	153
4.11	打印和打印预览	154
4.11.1	打印过程	155
4.11.2	打印循环	156
4.11.3	打印预览	156
4.12	编程范例	157
4.12.1	创建应用程序的基本框架	157
4.12.2	处理视图	158
4.12.3	处理文档	162
4.12.4	串行化处理	168
4.12.5	滚动处理	170
4.12.6	窗口分割	172
第 5 章	对话框、控件和控件栏	174
5.1	对话框	174
5.1.1	对话框的组成	174
5.1.2	模态和非模态对话框	174
5.1.3	创建对话框模板	175
5.1.4	用 ClassWizard 创建对话类	175
5.1.5	创建并显示对话框	175
5.1.6	设置对话框的背景颜色	176
5.1.7	初始化对话框	176

5.1.8	处理消息	176
5.1.9	对话数据交换和验证	177
5.1.10	对话框控件的类型无关访问	178
5.1.11	关闭对话框	179
5.1.12	管理对话框的 MFC 函数	179
5.2	通用对话类	180
5.2.1	CFontDialog 类	180
5.2.2	CColorDialog 类	181
5.2.3	CFileDialog 类	181
5.2.4	CPrintDialog 类	182
5.2.5	CPageSetupDialog 类	183
5.2.6	CFindReplaceDialog 类	183
5.3	控件	184
5.3.1	控件通知	186
5.3.2	消息反射	188
5.3.3	常用控件的使用	189
5.3.4	管理对话框控件的 MFC 函数	196
5.4	控件栏	198
5.4.1	工具栏	199
5.4.2	状态栏	201
5.4.3	对话框	202
5.4.4	CControlBar 成员函数	202
5.5	编程范例	202
5.5.1	创建对话框模板	203
5.5.2	创建对话类	205
5.5.3	定义成员变量	206
5.5.4	定义消息处理函数	209
5.5.5	修改对话类代码	210
5.5.6	创建对话类对象并显示对话框	212
5.5.7	添加工具栏按钮	217
5.5.8	建立并运行应用程序	218
5.6	属性表与属性页	221
5.6.1	创建属性表	221
5.6.2	交换数据	221
5.6.3	处理“应用”按钮	222
5.6.4	创建向导	222
5.7	表单视图	223

第 6 章 多线程编程	224
6.1 线程类型	224
6.2 工作者线程	224
6.2.1 实现控制函数	225
6.2.2 启动线程	225
6.3 用户界面线程	226
6.3.1 初始化新线程	227
6.3.2 处理消息	227
6.4 终止线程	227
6.4.1 正常的线程终止	227
6.4.2 线程的提前终止	228
6.4.3 获取线程退出码	228
6.5 线程同步	228
6.5.1 同步的场合	229
6.5.2 同步对象	229
6.5.3 同步访问对象	230
 第 7 章 动态链接库	 232
7.1 从 DLL 导出	232
7.1.1 使用 .DEF 文件导出函数	232
7.1.2 使用关键字 <code>__declspec(dllexport)</code>	233
7.1.3 使用 <code>AFX_EXT_CLASS</code> 导出和导入	233
7.1.4 调用约定	234
7.2 链接 DLL 到可执行程序	235
7.2.1 隐式链接	235
7.2.2 显式链接	235
7.3 使用 AppWizard 创建 MFC DLL	235
7.3.1 静态链接 MFC 的 Regular DLL	236
7.3.2 动态链接 MFC 的 Regular DLL	237
7.3.3 Regular DLL 的初始化	237
7.3.4 MFC 扩展 DLL	237
7.4 创建非 MFC DLL	239
 第 8 章 ActiveX、OLE、COM 和 ATL	 240
8.1 OLE 概览	240
8.1.1 部件对象模型 (COM)	240
8.1.2 结构化存储	241
8.1.3 Monikers	241

8.1.4	统一数据传输	242
8.1.5	OLE 文档	243
8.1.6	在位激活	243
8.1.7	OLE 自动化	243
8.1.8	OLE 控件	244
8.2	ActiveX 概览	244
8.2.1	ActiveX 文档	244
8.2.2	ActiveX 控件	245
8.2.3	COM 和 DCOM	245
8.2.4	Internet monikers	245
8.2.5	ActiveX 超级链接	245
8.2.6	ActiveX 服务器扩展	245
8.2.7	ActiveX 脚本	245
8.3	OLE 容器	246
8.3.1	使用 AppWizard 创建一个基本的 OLE 容器	246
8.3.2	检查基本容器的功能	247
8.3.3	分析 AppWizard 生成的容器	249
8.3.4	实现 OLE 项矩形	253
8.3.5	激活和选择	254
8.3.6	绘制多个 OLE 项	258
8.3.7	添加删除功能	258
8.3.8	建立并运行容器	259
8.3.9	实现“剪切”、“复制”和“粘贴”命令	261
8.3.10	智能无效确认	262
8.3.11	与服务器协调	264
8.3.12	链接容器	266
8.4	OLE 服务器	266
8.4.1	服务器的类型	267
8.4.2	服务器的实现方法	267
8.4.3	使用 AppWizard 创建一个全服务器	268
8.4.4	检查 Full 服务器的功能	268
8.4.5	分析 AppWizard 生成的 Full 服务器	269
8.5	拖放和剪贴板	274
8.5.1	数据目标和数据源	274
8.5.2	拖放	274
8.5.3	剪贴板	275
8.6	OLE 通用对话类	276
8.7	自动化	276

8.7.1	使用 AppWizard 创建自动化服务器	277
8.7.2	分析 AppWizard 生成的自动化服务器	277
8.7.3	添加自动化属性和方法	278
8.7.4	自动化客户	285
8.8	ActiveX 文档	289
8.8.1	ActiveX 容器	290
8.8.2	ActiveX 服务器	292
8.9	ActiveX 控件	294
8.9.1	使用 ActiveX ControlWizard 创建 ActiveX 控件	294
8.9.2	事件	296
8.9.3	方法	300
8.9.4	属性	302
8.9.5	属性页	307
8.9.6	ActiveX 控件的显示和绘制	309
8.9.7	串行化	310
8.9.8	处理错误	311
8.9.9	处理特殊键	311
8.9.10	优化	312
8.9.11	注册 ActiveX 控件	316
8.9.12	ActiveX 控件的使用	316
8.10	ActiveX 模板库(ATL)	317
8.10.1	使用 ATL COM AppWizard 创建一个 ATL 项目	317
8.10.2	使用 ATL Object Wizard 添加对象和控件	319
第 9 章	数据库编程	321
9.1	了解数据库	321
9.2	开放数据库连接(ODBC)	323
9.2.1	ODBC 体系结构	323
9.2.2	CDatabase 类	325
9.2.3	CRecordset 类	327
9.2.4	CRecordView 类	334
9.2.5	编写一个简单的数据库程序	335
9.3	数据访问对象(DAO)	339
9.4	OLE DB	340
9.4.1	OLE DB 部件	341
9.4.2	应用程序的结构	342
9.5	ActiveX 数据对象(ADO)	345

第 10 章 Internet 编程	347
10.1 WinInet	348
10.1.1 MFC 的 WinInet 类	348
10.1.2 创建 Internet 客户应用程序	349
10.1.3 ftp 客户应用程序	351
10.1.4 gopher 客户应用程序	353
10.1.5 http 客户应用程序	353
10.2 ISAPI 服务器扩展和过滤器	354
10.2.1 MFC 的 ISAPI 支持	354
10.2.2 创建 ISAPI 过滤器	355
10.2.3 创建 ISAPI 服务器扩展	358
10.3 用异步 Monikers 实现 Internet 下载	360
10.3.1 MFC 对异步 Monikers 的支持	360
10.3.2 使用异步 Monikers 下载文件	360
10.3.3 实现异步 ActiveX 控件属性	361
10.4 MFC 的 MAPI 支持	363
10.5 MFC 的 WinSock 支持	364
10.5.1 CAsyncSocket 类	364
10.5.2 CSocket 类	366

第 1 章 Visual C++ 6.0 环境

进入 90 年代以来,随着多媒体技术和图形图像技术的蓬勃发展,可视化技术得到了广泛重视,越来越多的计算机专业人员和非专业人员都开始研究并应用可视化技术。一般讲,可视化技术包含两个方面的含义:一是软件开发阶段的可视化,即可视化编程,它使编程工作成为一件轻松愉快,饶有趣味的工作;二是利用计算机图形技术和方法,对大量的数据进行处理,并用图形图像的方式形象而具体地加以显示。

本书讨论的是可视化编程。要支持可视化编程,就需要相应的可视化开发环境,美国微软公司推出的 Visual C++ 就是这样的环境。Visual C++ 采用一种非常巧妙的方法将 Windows 的编程复杂性封装起来,使编程人员步入轻松容易地编写 Windows 应用程序的殿堂。使用 Visual C++,编程人员可以对 C/C++ 程序进行各种操作,如建立、打开、浏览、编辑、保存、编译、链接和调试等。

目前,Visual C++ 的最新版本是 1998 年推出的 Visual C++ 6.0。Visual C++ 6.0 较之以往版本具有更新、更强的功能。

1.1 环境构成

每次启动 Visual C++ 6.0 后,都将显示类似图 1.1 所示的 Visual C++ 6.0 环境。Visual C++ 6.0 环境的最上端是标题栏,标题栏左端为窗口图标,单击窗口图标或者按 Alt+空格键,将弹出窗口控制选单(欲称菜单)。利用窗口控制选单,可以控制窗口的大小和位置,如还原、移动、关闭、最大化和最小化等。标题栏右边有三个控制按钮,从左至右依次为“最小化”、“还原”和“关闭”按钮,利用这些按钮可以快速设置窗口大小,如将窗口最小化为图标、使窗口最大化充满整个屏幕或关闭窗口等。

标题栏下面是选单栏和工具栏。屏幕最底端是状态栏,给出当前操作或所选择命令的提示信息。工具栏下面有两个窗口:工作区窗口(Workspace Window)和源代码窗口。工作区窗口下面是输出窗口(Output Window),显示的是项目建立过程中所产生的信息。

此外,程序调试过程中,还将为不同的调试信息提供彼此独立的窗口,如观察窗口(Watch Window)、变量窗口(Variables Window)、寄存器窗口(Registers Window)、存储器窗口(Memory Window)、调用堆栈窗口(Call Stack Window)和反汇编窗口(Disassembly Window)等。

注:工作区窗口、输出窗口和观察窗口等默认为船坞化窗口(Docked Window)。船坞化窗口总是固定在某一角或边上,如果双击其边框,那么可以解除固定状态使其成为浮动窗口。浮动窗口可以随意拖放到任何喜欢的位置。如果拖动时按住 Ctrl 键,那么可以将浮动窗

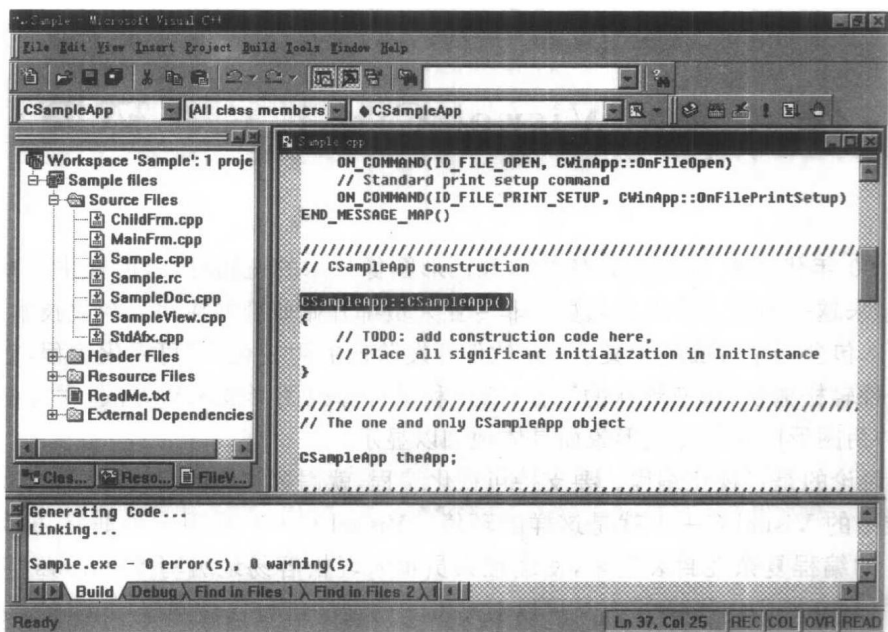


图 1.1 Visual C++ 6.0 环境

口拖出 Visual C++ 6.0 环境。再次双击边框又可使浮动窗口成为船坞化窗口。

1.1.1 工具栏

工具栏由某些工具按钮构成,提供对于常用命令的快速访问。用鼠标单击工具栏上的按钮即可快速执行该按钮所代表的操作。工具栏按钮具有提示功能,将鼠标指针指向某个按钮,稍待片刻,该按钮下面将出现按钮名称,而状态栏将显示按钮功能的简短描述。

Visual C++ 6.0 含有十几种工具栏。默认时,屏幕上会显示 Standard, Build MiniBar 和 WizardBar 工具栏。其他工具栏一般并不显示,而是在适当的时候自动弹出,如 Debug 工具栏就是在调试程序时弹出的。此外,用户还可以在需要时按以下方法之一来显示或隐藏工具栏:

- 用鼠标右按钮单击工具栏的任意位置,弹出工具栏快捷选单(图 1.2 的左边),从中单击要显示的工具栏。如果工具栏名称前面带有选中标记“✓”,那么表示该工具栏已显示。此时,如果要隐藏该工具栏,请再次单击来清除相应的选中标记。
- 选择“Tools”选单的“Customize”命令,弹出“Customize”对话框,然后切换到“Toolbars”选项卡(图 1.2 的右边),再从列表框中单击欲显示的工具栏。

注: 工具栏一般紧贴在选单栏下面或以垂直条状形式紧贴在边框上。如果将其从选单栏或边框拖开,那么工具栏将成为浮动工具栏。浮动工具栏将悬浮显示在窗口中。可以将浮动工具栏随意拖放到屏幕的其他位置。

1. Standard 工具栏

Standard 工具栏主要由以下工具按钮组成:

• 2 •

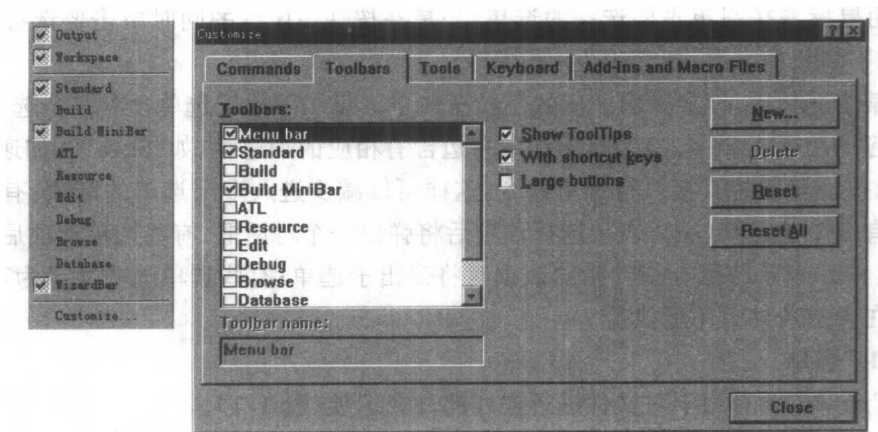


图 1.2 工具栏快捷选单和“Toolbars”选项卡

- New Text File: 创建新的文本文件。
- Open: 打开已有的文件。
- Save: 保存文件。
- Save All: 保存所有打开的文件。
- Cut: 剪切选定的内容到剪贴板中。
- Copy: 复制选定的内容到剪贴板中。
- Paste: 在当前插入点处插入剪贴板中的内容。
- Undo: 取消最后的操作。
- Redo: 重复先前取消的操作。
- Workspace: 显示或隐藏工作区窗口。
- Output: 显示或隐藏输出窗口。
- Window List: 管理当前打开的窗口。
- Find in Files: 在多个文件中搜索字符串。
- Find: 激活查找工具。
- Search: 搜索联机文档。

2. Build MiniBar 工具栏

Build MiniBar 工具栏主要由以下工具按钮组成:

- Compile: 编译文件。
- Build: 建立项目。
- Stop Build: 停止项目的建立。
- Execute Program: 执行程序。
- Go: 启动或继续程序的执行。
- Insert/Remove Breakpoint: 插入或删除断点。

1.1.2 选单栏

选单栏由多个选单组成,如“File”,“Edit”和“View”等。选择选单栏中的选单有两种方