

王建平 李团结 王宇红 编著

Visual C++ 6.0 编程



科学出版社

内 容 提 要

本书通过对大量实例的详细分析,循序渐进、深入浅出地介绍了微软最新的 32 位应用程序的可视化开发工具 Visual C++ 6.0。书中不但详尽细致地介绍了利用 Visual C++ 6.0 开发 Windows 应用程序的全过程,而且内容还涉及 Microsoft Visual C++ 工作平台、AppWizard、MFC 程序结构、图形设备接口、文档串行化、对话框、菜单、工具条和状态条、帮助系统的制作、打印和打印预览、常用的类、组件及 ActiveX 控件等多个专题。

本书适用于有一定 C++ 基础的初、中级程序设计和开发人员。利用本书提供的知识,读者可以迅速开发出功能强大的 Windows 应用程序。

图书在版编目(CIP)数据

Visual C++ 6.0 编程/王建平,李团结,王宇红编著.

-北京:科学出版社,2000

ISBN 7-03-008537-X

I.V… II.①王… ②李… ③王… III.C 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2000) 第 60223 号

科学出版社 出版

北京东黄城根北街 16 号
邮政编码:100717

北京双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2000 年 8 月第 一 版 开本: 787×1092 1/16
2000 年 8 月第一次印刷 印张: 18 1/2
印数: 1—5 000 字数: 422 000

定价: 35.00 元 (含盘)

(如有印装质量问题,我社负责调换<环伟>)

前 言

Visual C++ 6.0 是迄今为止 Microsoft 公司推出的最好的编译程序。它界面优美、功能强大,可以开发 Windows 95, Windows 98, Windows NT 的各种 32 位应用程序。Visual C++ 6.0 提供的:MFC 类库(Microsoft Foundation Class)可以大大简化和加速 Windows 应用程序的开发;功能完善的资源编辑器,可以用来设计复杂的对话框、菜单、工具条、位图,以及 Windows 应用程序常用的其它组件;可视化开发工具——微软开发工作室(Microsoft Visual Studio 6.0)提供了一个优秀的集成化开发环境,开发应用程序时可以通过图形化的视图来查看应用程序的结构。可以开发出功能强大、界面完善的 Windows 应用程序,而且,利用 Visual C++ 6.0,开发 Windows 应用程序异常简单。本书将举例向读者介绍 Visual C++ 6.0 的这些特点。

本书是为 Visual C++ 的初、中级读者而编写的。根据不同层次读者的需要,书中提供了大量精练、完整的实例,并对每一个实例进行详细分析,且全部实例都在 Windows 98 平台上用 Visual C++ 6.0 编译通过,这些例子也适用于 Visual C++ 4.0 和 Visual C++ 5.0。本书内容详实、编排新颖、语言简练、安排由浅入深,便于读者循序渐进学习,逐步成为 Visual C++ 的高手。

全书共分为十六章,第一章详细介绍了 Microsoft Visual C++ 工作平台的使用方法;第二章介绍了利用 AppWizard 创建了一个简单的基于 MFC 的 Windows 应用程序,并对 MFC 程序结构和 AppWizard 生成的类和代码作了简要的介绍;第三章对 MFC 应用类、框架类、文档类、视图类作了较为详细的说明,介绍了这些类的主要成员函数及其使用方法;第四章介绍了图形设备接口,包括设备环境类、字体、画笔及画刷等,对于每一种设备接口,都给出了具体的应用实例;第五章介绍了文档的串行化功能,并用实例具体说明;第六章介绍有模式对话框,并演示了资源编辑器的使用方法,最后编制了对话框的具体应用;第七章在前一章的基础上,进一步创建了无模式对话框的应用,从而说明了无模式对话框与有模式对话框的区别以及无模式对话框的操作方法;第八章对 MFC 菜单系统进行了介绍,并用实例说明了菜单的使用方法;第九章介绍工具条和状态条,并给出了具体的实例;第十章介绍帮助系统的制作;第十一章介绍打印和打印预览,并给出了具体应用实例;第十二章介绍了编程中经常遇到的几个常用的类,内容包括字符串类、数组类及时间类等;第十三章介绍组件,内容涉及制作软件封面、显示系统信息、添加进度信息、在视图中添加弹出式菜单等,最后是实例;第十四章是 ActiveX 控件编程,说明了给应用程序插入 ActiveX 控件的一般方法,具体介绍了 MSFlexGrid 控件,并演示了应用该控件编程的实例。在第十六章附录中列出了常见的 MFC 宏和全局变量、MFC 类结构等在编程中经常需要参考的资料。

由于作者水平有限,加之时间仓促,书中错误和疏漏之处在所难免,敬请读者批评、指正,在此表示衷心的感谢。

王建平 李团结 王宇红

2000 年 3 月 1 日

目 录

前 言

1 Microsoft Visual C++ 编译器	(1)
1.1 菜单系统	(2)
1.1.1 File 菜单	(2)
1.1.2 Edit 菜单	(6)
1.1.3 View 菜单	(9)
1.1.4 Insert 菜单	(11)
1.1.5 Project 菜单	(12)
1.1.6 Build 菜单	(14)
1.1.7 Tools 菜单	(15)
1.1.8 Window 菜单	(16)
1.1.9 Help 菜单	(16)
1.2 工具栏	(16)
1.2.1 Standard 工具栏	(17)
1.2.2 Build 工具栏和 Build minibar 工具栏	(18)
1.2.3 Debug 工具栏	(18)
1.3 工作区	(18)
1.3.1 按类查看代码	(19)
1.3.2 按文件查看代码	(20)
1.3.3 查看资源	(21)
1.4 代码编辑区	(22)
1.5 输出和出错信息	(23)
2 创建第一个 Windows 应用程序	(24)
2.1 用 AppWizard 生成一个简单的 Windows 应用程序	(24)
2.2 理解 AppWizard 生成的代码	(32)
2.2.1 应用类	(32)
2.2.2 边框窗口类	(34)
2.2.3 文档类	(37)
2.2.4 视图类	(37)
3 应用、框架、文档和视图	(38)
3.1 CWinApp 类	(38)
3.1.1 CWinApp::AddDocTemplate	(38)
3.1.2 CWinApp::Enable3dControls	(38)
3.1.3 CWinApp::InitInstance	(39)
3.1.4 CWinApp::LoadCursor	(39)
3.1.5 CWinApp::LoadIcon	(39)

3.1.6	CWinApp::Run	(40)
3.1.7	CWinApp::ExitInstance	(40)
3.2	框架类	(41)
3.2.1	CMDIFrameWnd 类	(41)
3.2.2	CMDIChildWnd	(41)
3.3	CDocument 类	(42)
3.3.1	CDocument::AddView	(43)
3.3.2	CDocument::CanCloseFrame	(43)
3.3.3	CDocument::Cdocument	(44)
3.3.4	CDocument::DeleteContents	(44)
3.3.5	CDocument::GetDocTemplate	(44)
3.3.6	CDocument::GetFirstViewPosition	(44)
3.3.7	CDocument::GetNextView	(44)
3.3.8	CDocument::GetPathName	(45)
3.3.9	CDocument::GetTitle	(45)
3.3.10	CDocument::IsModified	(45)
3.3.11	CDocument::OnChangedViewList	(45)
3.3.12	CDocument::OnCloseDocument	(46)
3.3.13	CDocument::OnNewDocument	(46)
3.3.14	CDocument::OnOpenDocument	(46)
3.3.15	CDocument::OnSaveDocument	(47)
3.3.16	CDocument::PreCloseFrame	(47)
3.3.17	CDocument::RemoveView	(47)
3.3.18	CDocument::SaveModified	(48)
3.3.19	CDocument::SetModifiedFlag	(48)
3.3.20	CDocument::SetPathName	(48)
3.3.21	CDocument::SetTitle	(48)
3.3.22	CDocument::UpdateAllViews	(49)
3.4	Cview 类	(49)
3.4.1	CView::DoPreparePrinting	(50)
3.4.2	CView::GetDocument	(51)
3.4.3	CView::IsSelected	(51)
3.4.4	CView::OnActivateFrame	(51)
3.4.5	CView::OnActivateView	(51)
3.4.6	CView::OnBeginPrinting	(52)
3.4.7	CView::OnDraw	(52)
3.4.8	CView::OnEndPrinting	(53)
3.4.9	CView::OnEndPrintPreview	(53)
3.4.10	CView::OnInitialUpdate	(54)
3.4.11	CView::OnPrepareDC	(54)
3.4.12	CView::OnPreparePrinting	(55)
3.4.13	CView::OnPrint	(55)

3.4.14	CView::OnUpdate	(56)
3.5	CEditView 类	(56)
3.6	CFormView 类	(57)
3.7	CScrollView 类	(57)
3.8	CRichEditView 类	(58)
3.9	实例	(58)
4	图形设备接口(GDI)	(61)
4.1	设备环境类	(61)
4.1.1	CDC 类	(61)
4.1.2	实例	(62)
4.2	字体	(64)
4.2.1	CFont 类	(64)
4.2.2	实例	(64)
4.3	画笔	(68)
4.3.1	CPen 类	(68)
4.3.2	实例	(71)
4.4	画刷	(74)
4.4.1	CBrush 类	(74)
4.4.2	实例	(75)
5	文件操作	(79)
5.1	关于串行化	(79)
5.1.1	CArchive 类	(79)
5.1.2	使类串行化	(80)
5.2	实例	(82)
6	有模式对话框	(110)
6.1	创建对话框资源	(110)
6.2	编写对话框类	(114)
6.3	使用对话框类	(117)
6.3.1	显示对话框	(117)
6.3.2	初始化对话框	(118)
6.3.3	响应编辑框	(119)
6.3.4	响应复选框	(121)
6.3.5	响应单选框	(122)
6.3.6	响应列表框	(124)
6.4	实例	(126)
6.4.1	创建对话框资源	(126)
6.4.2	编写对话框类	(126)
6.4.3	初始化对话框	(128)
6.4.4	设置矩阵的行数及列数	(130)
6.4.5	在静态文本框中显示矩阵元素	(132)
6.4.6	输入矩阵元素	(133)

6.4.7	响应列表框	(133)
6.4.8	显示 CDialogMatrixInput 对话框	(134)
6.4.9	编译及运行	(135)
7	无模式对话框	(137)
7.1	创建和使用无模式对话框	(137)
7.2	实例	(138)
7.2.1	创建对话框资源	(138)
7.2.2	编写对话框类	(140)
7.2.3	使用对话框类	(143)
8	菜单与键盘加速键	(146)
8.1	Windows 消息映射	(146)
8.2	应用框架的内含菜单	(147)
8.3	菜单命令处理	(148)
8.4	菜单项的允许与禁止	(148)
8.5	键盘加速键	(148)
8.6	实例	(149)
8.6.1	添加菜单资源	(149)
8.6.2	在视图类中添加消息处理函数	(152)
8.6.3	菜单项的状态更新	(157)
8.6.4	添加键盘加速键	(159)
9	工具条、状态条	(163)
9.1	工具条	(163)
9.1.1	AppWizard 生成的缺省工具条	(163)
9.1.2	创建工具条的方法	(165)
9.1.3	工具条按钮的状态和风格	(165)
9.2	Visual C++ 6 中工具条的新特色	(166)
9.2.1	扁平钮和把手	(166)
9.2.2	文本标签	(167)
9.3	状态条	(168)
9.3.1	标准状态条	(168)
9.3.2	创建状态条的方法	(169)
9.3.3	更新状态条中的文字和风格	(170)
9.4	实例	(170)
9.4.1	创建工程	(170)
9.4.2	添加菜单资源	(170)
9.4.3	将新建按钮的位图资源添加工具条上	(170)
9.4.4	修改框加类	(173)
9.4.5	修改视图类	(174)
9.4.6	修改状态条	(181)
10	帮助	(185)
10.1	帮助文件的不同类型	(185)

10.2	设计用户 Help 的方法	(186)
10.3	编写 RTF 脚本文件	(187)
10.4	使用 Microsoft Help Workshop	(189)
10.5	实例	(196)
11	打印和打印预览	(204)
11.1	MFC 的打印和打印预览	(204)
11.1.1	参数缺省时的打印功能	(204)
11.1.2	映射方式	(205)
11.2	打印多页	(209)
11.2.1	滚动视	(209)
11.2.2	设置原点	(211)
11.2.3	CPrintInfo 类	(214)
12	几个常用的类	(218)
12.1	CString 类	(218)
12.2	CPoint 类	(224)
12.3	CSize 类	(225)
12.4	CRect 类	(226)
12.5	CTime 类	(229)
12.6	CTimeSpan 类	(232)
12.7	CArray 类	(234)
13	组件	(237)
13.1	添加组件的一般方法	(237)
13.2	实例	(238)
13.2.1	创建工程	(238)
13.2.2	制作软件封面	(238)
13.2.3	显示系统信息	(240)
13.2.4	空闲时间处理	(242)
13.2.5	在视图中添加弹出式菜单	(244)
13.2.6	切分窗口	(246)
13.2.7	添加日积月累	(248)
13.2.8	给控件添加小提示	(250)
14	ActiveX 控件	(253)
14.1	ActiveX 控件概念	(253)
14.2	给应用程序插入 ActiveX 控件	(253)
14.3	MSFlexGrid 控件的应用实例	(255)
14.3.1	创建工程	(255)
14.3.2	添加 MSFlexGrid 控件	(255)
14.3.3	添加资源	(256)
14.3.4	添加成员变量和成员函数	(258)
15	打包应用程序	(265)

15.1	收集文件	(265)
15.2	建立安装例程	(266)
15.2.1	InstallShield 向导	(266)
15.2.2	Scripts 标签	(271)
15.2.3	Components 标签	(271)
15.2.4	File Groups 标签	(274)
15.2.5	Resources 标签	(274)
15.2.6	Setup Types 标签	(276)
15.2.7	Setup Files 标签	(276)
15.2.8	Media 标签	(277)
16	附录	(279)
16.1	MFC 宏和全局变量	(279)
16.1.1	应用程序信息和管理函数	(279)
16.1.2	数据类型	(280)
16.1.3	运行时对象模型服务	(281)
16.1.4	诊断服务	(281)
16.1.5	异常处理	(282)
16.2	MFC 类结构	(284)

1

Microsoft Visual C++ 编译器

在学习利用 MFC 编程之前,我们先来了解一下 Developer Studio 的用户界面。Microsoft Visual C++ 6.0 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、工程创建工具和 Debugger 调试器等。用户可以在集成开发环境中创建、打开和编辑文件,也可以编译、链接、运行和调试应用程序。当使用 Visual C++ 编程序时,可以把 Visual C++ 的显示界面划分成 5 个功能区:菜单条、工具条、工作区、代码和资源编辑窗口、输出和出错信息窗口。每个功能区都可以独立于其它功能区来操作,因此,可以在各功能区之间自由地切换。图 1.1 显示了典型编辑器的情况以及这五个功能区的位置。

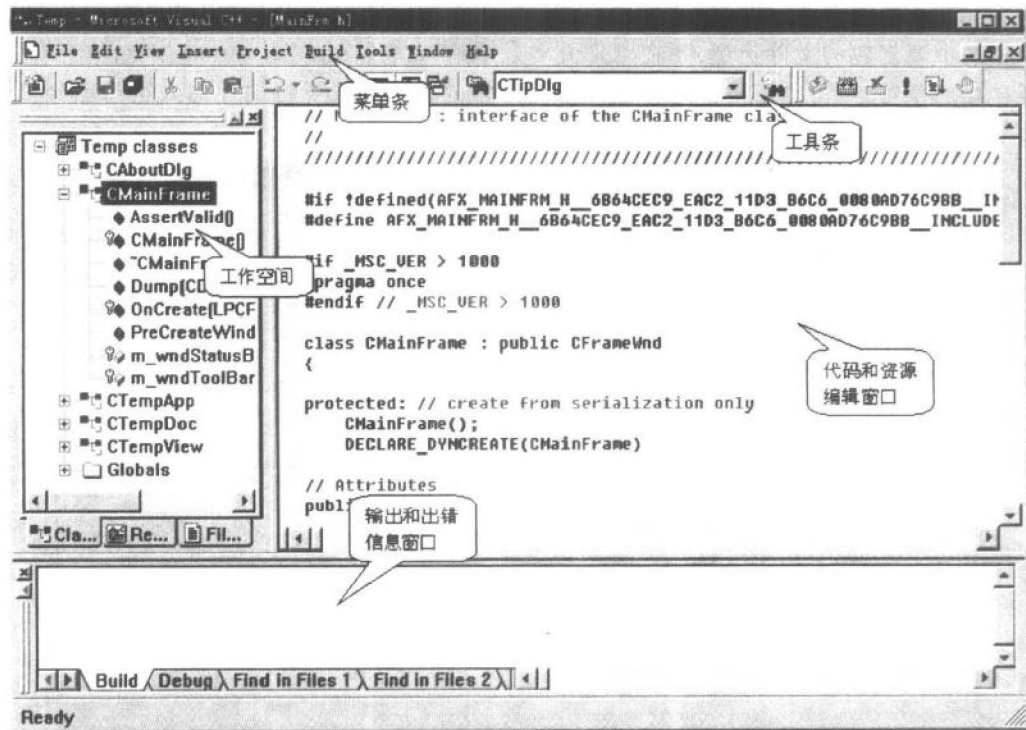


图 1.1 Microsoft Visual C++ 6.0 集成开发环境

注意: 用户可以利用鼠标拖动,将菜单条、工具条、工作区、代码和资源编辑窗口、输出和出错信息窗口任意地更换顺序与位置,也可以隐藏其中的一个或多个窗口。以下各节我们对这五个功能区分别加以介绍。

1.1 菜单系统

Microsoft Visual C++ 6.0 的系统菜单由多个子菜单组成,可以通过鼠标或键盘来选择子菜单。在选中某一个子菜单后,会弹出下拉式菜单,选择其中的菜单项可以执行相应的菜单命令。有些菜单还提供了加速键,在相应菜单项的右边为加速键的名称提示。下面对一些常用的和比较重要的菜单命令进行说明。

1.1.1 File 菜单

1. File | New...(Ctrl + N)

File | New... 菜单用于新建文件、工程、工作区和文档。用鼠标点击 File | New... 菜单后,将弹出 New 对话框,该对话框包含 4 个标签:File 标签、Projects 标签、Workspaces 标签和 Other Documents 标签。

File 标签

如图 1.2 所示,通过 File 标签可以创建多种类型的文件。在创建某种类型的文件时,须先选中要创建的文件类型,并在 File 文本框中输入文件名,在 Location 编辑框中输入文件存放的位置,单击 OK 按钮就可完成新文件的创建。如果要将文件加入当前已经打开的工程,只要选中 Add to project 复选框即可。

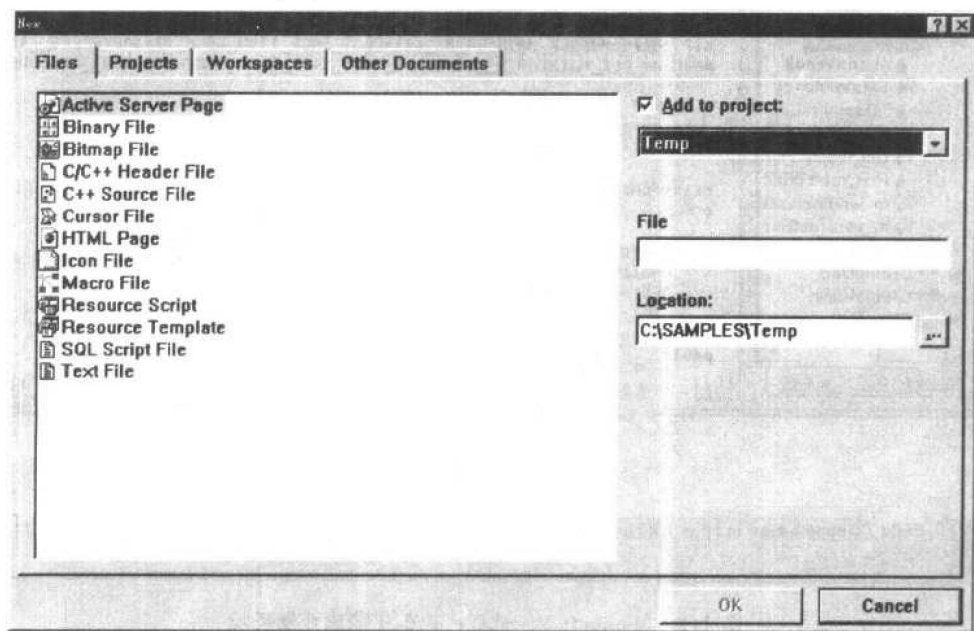


图 1.2 New 对话框的 File 标签

通过 File 标签可以创建的文件类型有:

- | | |
|---------------------|----------|
| Active Server Page: | 活动服务器文件。 |
| 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。
Text File:	文本文件。

• Projects 标签

如图 1.3 所示,通过 Projects 标签可以创建多种类型的工程。在创建某种类型的工程时,需先选中要创建的工程类型,并在 Project name 文本框中输入工程名,在 Location 编辑框中输入工程文件存放的位置,单击 OK 按钮就可完成新工程的创建。如果要 will 工程加入当前已经打开的工作区中,可以选择 Add to Current Workspace 单选框,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,可选中 Dependency of 复选框并需指定工程名。

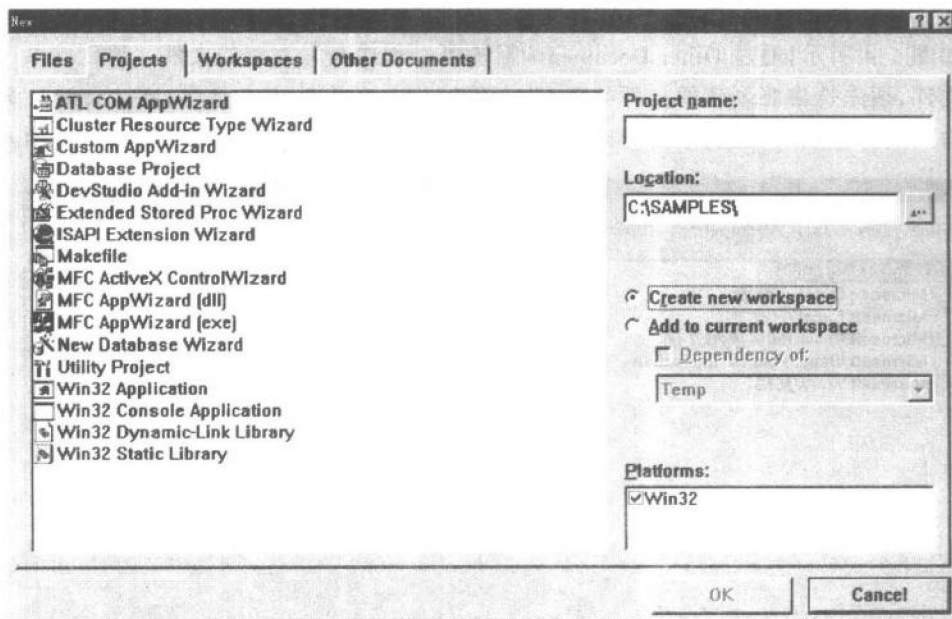


图 1.3 New 对话框的 Projects 标签

通过 Projects 标签可以创建的工程类型有:

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:	空白实用工具。
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。

• Workspaces 标签

通过 Workspaces 标签可以创建空白工作区。要创建空白工作区,须选择 Blank Workspaces,并在 Workspaces name 文本框中输入工作区名,在 Location 编辑框中输入工作区存放的位置,单击 OK 按钮就可完成新工作区的创建。

• Other Documents 标签

如图 1.4 所示,通过 Other Documents 标签可以创建其它类型的文件。要创建某一类型的文件,须先选中要创建的文件类型,并在 File 文本框中输入文件名,在 Location 编辑框中输入文件存放的位置,单击 OK 按钮就可完成新文件的创建。如果要将文件加入当

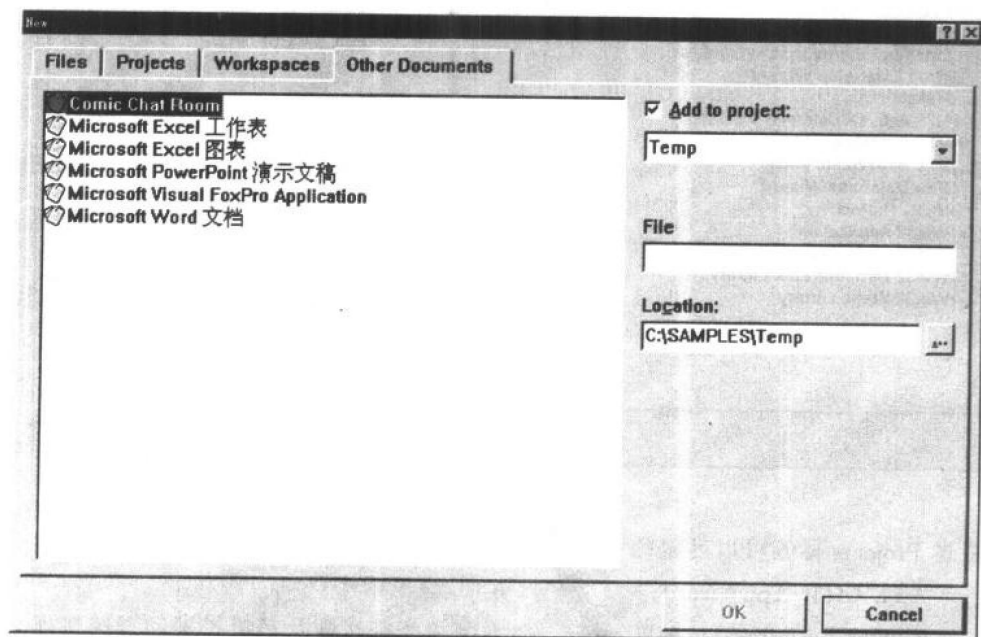


图 1.4 New 对话框的 Other Documents 标签

前已经打开的工程中,只要选中 Add to project 复选框即可。

2. File | Open... (Ctrl + O)

选择 File | Open... 菜单,该菜单将调用 Open 对话框,如图 1.5 所示。该对话框是标准的 Window File Open 对话框,缺省打开的文件类型是具有扩展名为:.c, .cpp, .cxx, .h 或者是 .rc 的普通文件。按下拉式组合框,可以打开可执行文件以及工作区文件等。

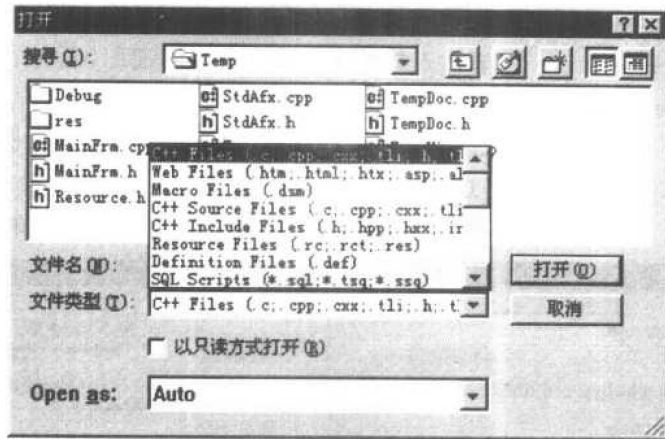


图 1.5 Open 对话框

3. 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,图 1.6 所示。该对话框也是标准的 Window File Open 对话框,缺省打开的文件类型是具有扩展名为:.dsw, .mdp 的工作区文件。

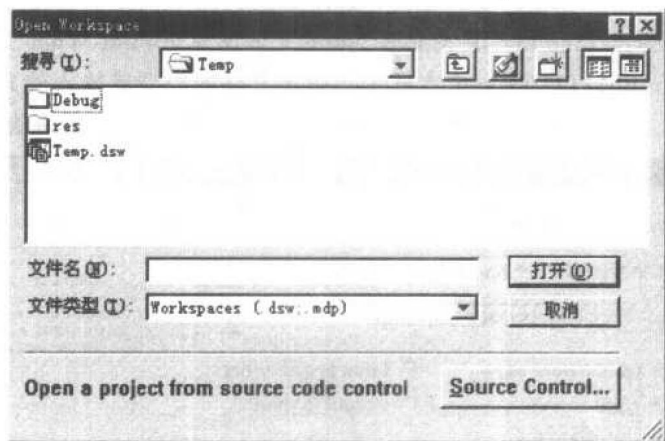


图 1.6 Open Workspace 对话框

1.1.2 Edit 菜单

1. Edit | Find...(Ctrl + L)

Edit | Find... 菜单用于查找指定的字符串。选择 Edit | Find... 菜单,将弹出 Find 对话框,如图 1.7 所示。Find What 文本框用于输入要查找的字符串,在以下的复合框中可以设置查找的选项。

- Match Whole Word Only: 表示查找时按整个单词匹配。
- Match Case: 表示查找时字母要按大小写匹配。
- Regular Expression: 表示查找的字符串中可以有通配符。
- Search all open documents: 表示在所有打开的文档中查找。
- Direction: 选择查找方向,可以向下查找(Down)也可以向上查找(Up)。
- Mark All: 单击此按钮,可以将所有包含指定字符串的行标记出来。

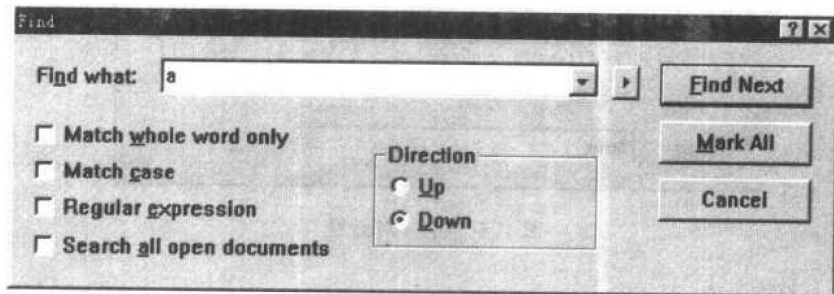


图 1.7 Find 对话框

2. Edit | Find in Files...

Edit | Find in File... 菜单用于在多个文件中查找指定的字符串。选择 Edit | Find in File... 菜单,将弹出 Find in File 对话框,如图 1.8 所示。Find What 文本框用于输入要查找的字符串,In files/file Type 文本框用于指定文件的类型或输入的文件名,In Folder 文本框用于指定目录或输入的目录名,在对话框下面的复合框中可设置查找的选项。

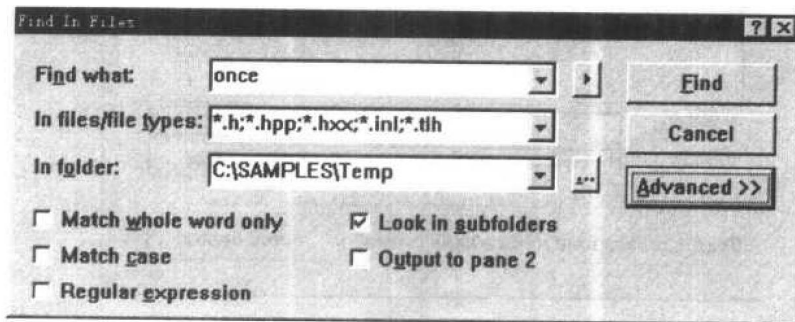


图 1.8 Find in File 对话框

- Match Whole Word Only: 表示查找时按整个单词匹配。
- Match Case: 表示查找时字母要按大小写匹配。

- Regular Expression: 表示查找的字符串中可以有通配符。
- Look in subfolders: 在所选目录的所有子目录中查找。
- Output to pane 2: 将查找的结果输出到窗口 Find in Files 2 中,从而不清除上一次查找的结果。

3. Edit | Replace...(Ctrl + H)

Edit | Replace... 菜单用于在文件中将指定字符串替换为另一个字符串。选择 Edit | Replace... 菜单,将弹出 Replace 对话框。该对话框与 Find 对话框基本类似,除 Find What 文本框用于输入要查找的字符串外,另外增加了一个 Replace with 文本框,用于输入进行替换的字符串。

4. Edit | Go to...(Ctrl + G)

Edit | Go to... 菜单用于将光标移动到活动窗口的指定位置,使用户能够快速地位于文件所指定的行、地址、书签、对象的定义位置或引用位置等。选择 Edit | Go to... 菜单,将弹出 Go to 对话框,如图 1.9 所示。可以在左边的 Go to What 列表项中选择需要的选项,如果 Line 被选中,则需要在右边的编辑框中输入行号,如果 Bookmark 被选中,则需要在组合框中挑选一特殊的书签。

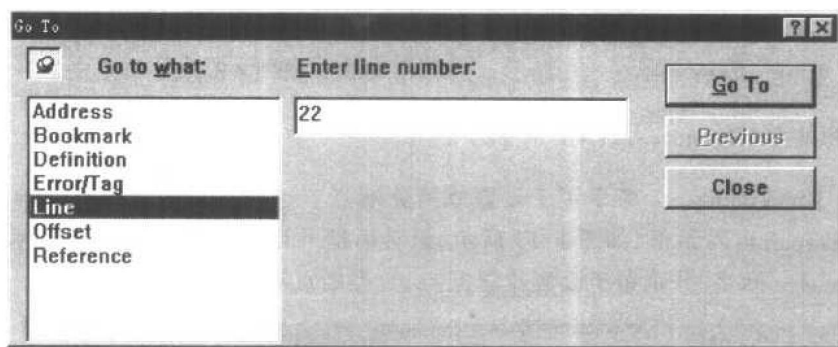


图 1.9 Go to 对话框

5. Edit | Bookmarks...(Alt + F2)

Edit | Bookmarks... 菜单用于在文件中定义书签。选择 Edit | Bookmarks... 菜单,将弹出 Bookmarks 对话框,如图 1.10 所示。为了给所在行增加一命名书签并保存在文件中,可以在 Name 文本框中输入书签名,并单击 Add 按钮;如果要进入一个命名书签内,可从列表框中选择该命名书签,再按 Go To 按钮。

6. Edit | Advance...

Edit | Advance... 菜单下包含了下面一些命令:

- Incremental Search: 对任意位置出现的变量或对象均提示较详细的说明。
- Format Selection: 格式化选定区域的内容。
- Tabify Selection: 用 TAB 符号替换指定区域的空格。

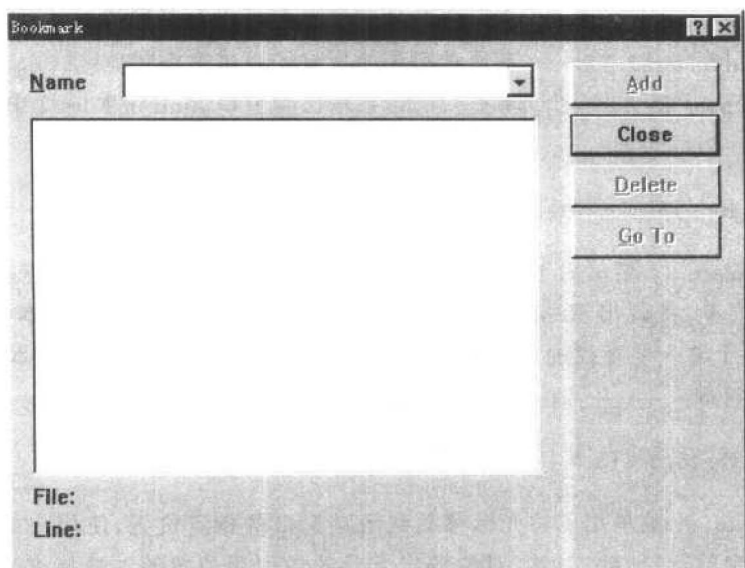


图 1.10 Bookmarks 对话框

- Untabify Selection: 用空格替换指定区域的 TAB 符号。
- Make Selection Uppercase: 使选定区域的所有字符大写。
- Make Selection Lowercase: 使选定区域的所有字符小写。
- View Whitespace: 显示或隐藏活动窗口的 TAB 符号和空格。

7. Edit | Breakpoints...(Alt + F2)

Edit | Breakpoints... 菜单用于设置或清除断点。选择 Edit | Breakpoints... 菜单，将弹出 Breakpoints 对话框，如图 1.11 所示，该对话框下有 3 个标签：Location 标签与 Data 标签与 Message 标签，分别用于设置位置断点、数据断点和消息断点。

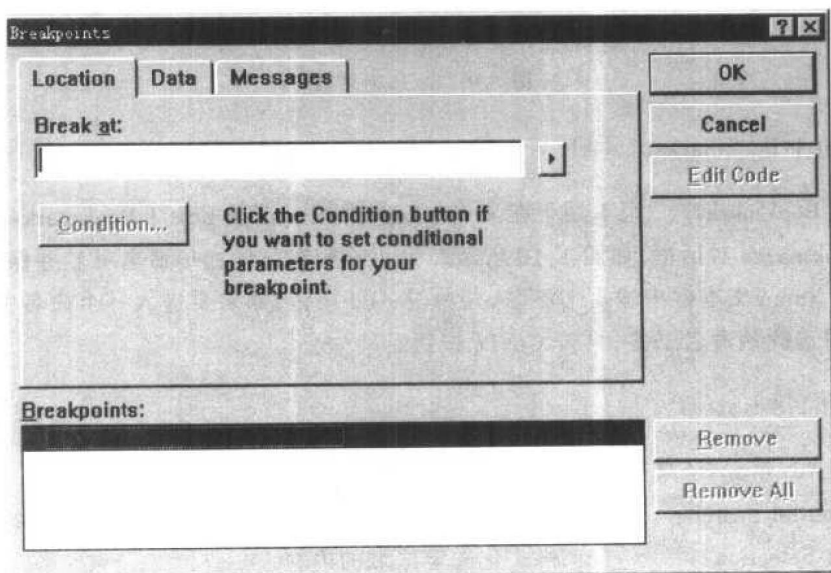


图 1.11 Breakpoints 对话框