

Visual C++ 6.0

项目开发指南

——从入门到精通——



清华大学出版社

596 TP312e
W94

Visual C++ 6.0 项目开发指南

武 装 张碧霞 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载,
也可到视听部复制

国防工业出版社

·北 京·

图书在版编目(CIP)数据

Visual C++ 6.0 项目开发指南/武装,张碧霞编著.
—北京:国防工业出版社,2000.7
ISBN 7-118-02296-9

I.V… II.①武…②张… III.C语言-程序设计
IV.TP312

中国版本图书馆 CIP 数据核字(2000)第 19231 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 27 $\frac{3}{4}$ 640 千字

2000 年 7 月第 1 版 2000 年 7 月北京第 1 次印刷

印数:1—4000 册 定价:48.00 元

(本书如有印装错误,我社负责调换)

前 言

Visual C++ 6.0 是 Microsoft 发行的当今最流行的软件开发工具之一,是程序员的首选编程利器。其强大的功能能够大大加速程序员的工作、提高程序代码的效率。而且由于 Microsoft 在 PC 操作系统市场上的垄断地位,使用 Microsoft 发行的编程软件能够提供与 Windows 操作系统的最大兼容性。

本书并不像许多传统的介绍编程语言的书籍那样,只是进行知识点的简单罗列,且所举实例比较简单,大部分重要知识点的用法仍需要读者自己去艰难摸索,费时且收效甚微,也避免了一些项目开发类书籍的通常弊病,即项目与知识点脱节。本书以项目开发贯穿始终,并结合项目各个功能的实现过程,向读者全面讲解 Visual C++ 的语言知识点和编程技巧,读者只需依法而行,就完全可以掌握并灵活运用这些知识点,节省下大量的摸索时间用于新知识点的学习,进步当然会快,效率当然会高。同时由于项目的设计和实现都是从头起步,对读者项目开发能力和程序设计能力的提高肯定会大有帮助,这正是每个读者在学习 Visual C++ 语言时所追求的最终效果。

本书实例丰富,同时绝大多数实例有一定的深度和难度,读者最好能深入体会,相信对读者编程能力的提高会有一定的帮助。

本书所附的光盘包括“制图精灵”、“24 点游戏”和“计算器”的全部源代码和可执行文件,它们分别可以在光盘的 Painter、Point 24 和 Calculator 目录下找到。读者可以从光盘上打开文件和工程浏览程序,或直接运行可执行文件以测试程序的功能。

参与本书编写的作者除封面署名作者外,还有吴丽、刘蓉、刘兵、王键、李宁、杜江、何明声、宋明、陈力、李庆声、刘战涛、李霞、刘志东、华明、张蓉蓉、姚兵、哈文进、吴迪、周春雷、林云、龚剑、况颖、李小明、王爱玲、葛飞参与了校对与录排工作,在此对他们的辛勤劳动表示感谢。

第一篇

基础篇



本篇包括 3 章，分别向读者介绍了 Visual C++ 6.0 的用户界面、编程工具的使用、C++ 语言以及 Windows 编程机制等内容。初级读者通过本篇的学习，可以迅速掌握 VC 编程的基本知识，而对 VC 比较熟悉的读者可以跳过这部分内容直接进入下一篇的学习。



第 1 章 Visual C++ 6.0 操作速成

Visual C++ 6.0 是当今最流行的软件开发工具之一,是程序员的首选编程利器。选择 Visual C++ 6.0,一是因为其强大的功能能够大大加速程序员的工作、提高程序代码的效率。二是因为 Microsoft 在 PC 操作系统市场上的垄断地位,使用 Microsoft 发行的编程软件能够提供与 Windows 操作系统的最大兼容性。至于第三条原因,既然 MFC 类库都是使用 C/C++ 创建的,那么选择 Visual C++ 当然能够最方便地使用 MFC 所提供的强大功能。

Visual C++ 并不只是一个程序代码的编辑器和编译器。要想掌握使用 Visual C++ 编程的方法,首先需要掌握软件的基本操作,这样可以提高使用软件的效率。本章将向读者介绍其用户界面和编程工具,使读者能够快速掌握它的基本使用。

本章要点:

- * Visual C++ 6.0 的新特性
- * Visual C++ 6.0 常用操作
- * ClassWizard 的使用方法

1.1 Visual C++ 6.0 概述

Visual C++ 6.0 是 Visual Studio 98 (Visual Studio 6.0) 家族的一个成员,与 Visual Basic 6.0、Visual Foxpro 6.0、Visual J++ 6.0、Visual SourceSafe 6.0、Visual InterDev 6.0、MSDN Library 6.0 等产品共同发行。Visual C++ 6.0 包括标准版、专业版和企业版 3 个版本。

Visual C++ 6.0 标准版具有除了代码优化、代码编辑器以及静态链接 MFC 类库外的所有专业版的特点,适合于初学者使用。Visual C++ 6.0 专业版具有远程数据控制和数据打包控制等功能,并提供高质量的软件工具,适合于商用软件的开发。Visual C++ 6.0 企业版除具有专业版的所有功能外,还具有 SQL 数据库和调试工具以及可视化代码安检控制系统,适合于企业分布式数据库应用程序的开发。

与以前的版本相比,Visual C++ 6.0 具有以下新特性:

- 改进的编译器。Visual C++ 6.0 中改进的编译器功能有:延迟的动态库联接输入,新的各种编译参数,生成更短小的编译程序代码。

- 增强的编辑器。Visual C++ 6.0 的编辑器中增加的功能有:关键字提示,更好的初始化控制,新的警告机制,支持操作符 delete 的自动放置,支持运行时的错误检查。此外还能够进行自动声明完成、快速宏记录和支持 IE4.0 的源编辑器。

- 更方便的跟踪调试工具。Visual C++ 6.0 中新增的调试跟踪功能包括:新的诊断映射机制,无须重编译的调试,软件形式的错误寄存器,改进的模拟输出,改进的函数指针显示,进程的远程调用,MMX 寄存器显示以及显示动态链接库的对话框等。

- 新的对象模板。新的对象模板提供了一些新的属性和方法,可以使工程的生成更为灵活。

- 改进的工程操作。改进的工程操作包括:增加了基于命令行的编译方式,动态更新视类,可直接进入对话框编辑器,对 .idl 文件编译的新支持。

- Wizard 工具的新支持。包括:AppWizard 增加了对网络工具条的支持,支持不基于 Document/View 结构的 SDI/MDI,新增的对 OLE 支持,help 文件的客户编译规则,ATL 对象中的数据路径支持,WizardBar 和 ClassView 中直接应用 delete 命令,支持 IE 控制等。

- OLE 数据库模板。

- 新的 MFC 类库支持。包括:活动的文档内容,CHtmlView 类对 HTML 控制的动态支持,支持 IE4.0 的一般控制,加入了支持 OLE 数据库的类等。

- 新的数据库支持。包括:ADO 的数据捆绑对话框工具,ADO 和 OLE 的数据捆绑控制支持,数据工具中的 Oracle 支持。

- 新的例程。包括:ATL 中添加了 4 个新例程,MFC 中添加了 11 个新例程等。

- 新的工具。包括:对象管理器,网络 HTML 帮助,更新的安装界面等。

这些新特性使 Visual C++ 6.0 更加适应现在计算机网络化、运行速度快以及加强数据传输的趋势,成为软件开发的首选工具。

1.2 Visual C++ 6.0 用户界面

Visual C++ 6.0 是 Microsoft Developer Studio 的组件之一,而后者通常被称为集成开发环境(IDE),亦即 Visual C++ 6.0 的用户界面。Developer Studio 具有包括源码创建、资源编辑、编译、链接和调试等在内的许多功能。图 1-1 是一个完整的 Developer Studio 用户界面,其开发环

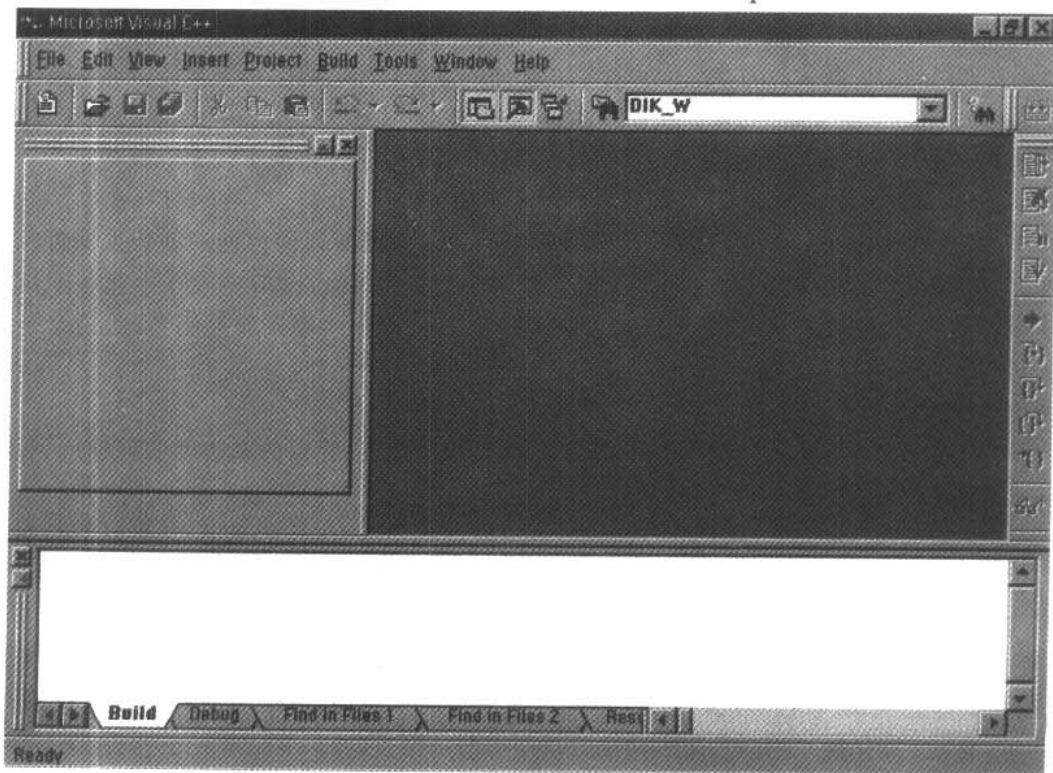


图 1-1 Developer Studio 的用户界面

境外观与以前版本相比只有细微的改变,它的风格和许多命令仍维持不变。如果读者以前从未用过 Visual C++, 就会发现如同任何大的 Windows 应用程序一样,需要一些时间来掌握软件的操作。它的界面是智能化的,而且非常宽容,鼓励用户去实践和尝试。

Developer Studio 的用户界面由上端的菜单和工具栏、左边的工程工作区(Project Workspace)、右边的进行文件与资源编辑的主工作区以及下端的输出窗口(Output)和状态栏组成。在调试时,平台还提供各种窗口,包括观察窗口、变量窗口、寄存器窗口、存储器窗口、调试堆栈窗口和反汇编窗口。下面主要通过对工程工作区和菜单体系的介绍,使读者对 Developer Studio 开发环境有进一步的了解。

1.2.1 工程工作区窗口

图 1-2 为打开一个工程后的工程工作区窗口。工程工作区窗口显示了项目各个方面的信息。在窗口底端选择相应的选项栏来显示项目的类、资源、数据源和文件的列表。在窗口中单击小的加号(+)或减号(-)来展开或折叠列表。例如,展开类的列表会显示出成员函数的名字。双击列表开头靠近文件夹或书本形图标,与单击表头的加号或减号有同样的效果。

在工程工作区的窗口下端有 3 个标签:ClassView、ResourceView 和 FileView。这 3 个标签提供了 3 个不同的查看工程信息的视图。在 ClassView 中可以查看项目中的类和成员函数,在 ResourceView 中可以查看工程中的资源数据,而在 FileView 中则可以查看工程的源文件。其中 ClassView 和 ResourceView 是开发应用程序时,最常使用的视图。

1.2.1.1 ClassView

ClassView 视图中以树形结构显示了此工程创建的所有类,并在每个类中列出了成员变量和成员函数,图 1-2 即显示了工程工作区的 ClassView 视图。每一个类首先列出带有紫色图标的成员函数,然后是才有绿蓝色图标的成员变量,每个成员的图标左边都有一个标志,以表示成员类型和存取类别的信息,保护型成员图标旁边的标志为一钥匙,私有成员的标志是一挂锁,而公有成员图标旁边没有标志。

在 ClassView 视图中双击类名,会在主工作区中打开这个类的头文件,显示出类的声明;而双击某个类的成员,则主工作区中会显示该成员的定义代码。使用鼠标右键单击某个类名,可以弹出一个快捷菜单对类进行操作,如图 1-3 所示。此菜单中包括下列菜单命令:

- Go to Definition——选择该菜单命令,将打开类的头文件。
- Add Member Function——选择该菜单命令,将弹出增加成员函数对话框。
- Add Member Variable——选择该菜单命令,将打开增加成员变量对话框。
- Add Virtual Function——选择该菜单命令,将打开虚函数重载对话框。
- Add Windows Message Handler——选择该菜单命令,将打开增加新窗口消息处理程序对话框。

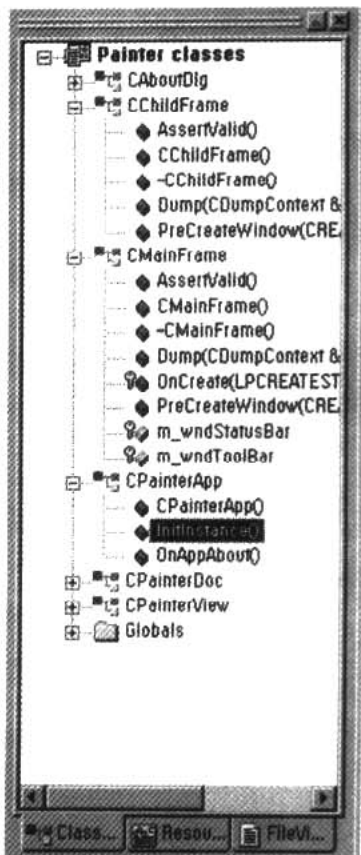


图 1-2 打开一个工程后的工程工作区窗口

框。

- Reference——选择该菜单命令,将产生应用程序中所有类名位置的列表。
- Derived Classes——选择该菜单命令,将产生此类中所有成员变量、成员函数以及以该类的所有派生类列表和附注信息。
- Base Classes——选择该菜单命令,将产生此类中所有成员变量、成员函数以及该类的所有基类列表和附注信息。
- Add to Gallery——选择该菜单命令,将该类添加到组件库中。
- New Folder——选择该菜单命令,将产生允许用户拖放类的文件类。
- Group by Access——选择该菜单命令,成员函数和成员变量分别以字母顺序排列;而关闭此选项时,成员函数和成员变量分别以其类型排列,即以公有型、保护型和私有型的顺序排列,而每一类型中又以字母顺序排列。
- Docking View——选择该菜单命令,将使工程工作区定位在主工作区边缘。
- Hide——选择该菜单命令,将隐藏工程工作区,使用 View | Workspace 菜单命令能够重新显示工程工作区。
- Properties——选择该菜单命令,将显示类的属性。

用鼠标右键单击成员函数名,也会弹出一个快捷菜单,如图 1-4 所示。此菜单包括下列菜单项:

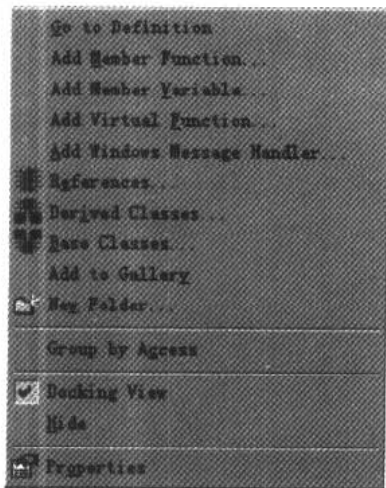


图 1-3 类的快捷菜单

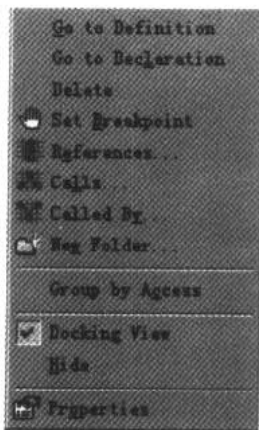


图 1-4 类成员的快捷菜单

- Go to Definition——选择该菜单命令,将打开定义该成员函数的源文件。
- Go to Declaration——选择该菜单命令,将打开说明该成员函数的头文件。
- Set Breakpoint——选择该菜单命令,将在该成员函数中设置断点。
- References——选择该菜单命令,将产生该成员函数的位置列表。
- Calls——选择该菜单命令,将产生该成员函数所调用的所有函数列表。
- Call by——选择该菜单命令,将产生所有调用该成员函数的函数列表。
- New Folder——选择该菜单命令,将建立一个允许用户拖放类的文件夹。
- Group by Access——选择该菜单命令,成员函数和成员变量分别以字母顺序排列;而关闭此选项时,成员函数和成员变量分别以其类型排列,即以公有型、保护型和私有型的顺序排列,而

每一类型中又以字母顺序排列。

1.2.1.2 ResourceView

单击 ResourceView 标签,会在工程工作区窗口中将应用程序中所有的可视资源进行树状列表,如图 1-5 所示。

在 Windows 编程中,经常会用到不同种类的资源,例如位图、光标、图标、菜单、工具栏和对话框模板等。在早些版本中所有的资源都被定义在资源描述文件中,必须用独立的资源编译器编译这些 .rc 文件,然后再连接到可执行文件中去。而在 Visual C++ 6.0 的开发平台提供了资源编辑器。这种资源编辑器可以提供可视化的资源编辑功能,从而大大简化了程序员的工作。

在编程过程中,程序员可能只是需要查看工程中资源文件的内容,而不是打开一个资源。例如,程序员可能希望在所有的对话框资源中搜索一个字符串,而不必逐个地打开每一个对话框资源。这时可以使用以文本形式打开资源文件的方法看到资源,并且执行文本编辑器支持的全局操作。

可以执行下面的步骤来以文本的形式打开一个资源文件:

- (1) 从项目工程的菜单项中选择 File|Open 菜单命令,并选择文件类型为 Resource Files。
- (2) 从下面的 Open as 下拉列表框中选择文件打开的形式为 Text。
- (3) 单击“打开”按钮。

这样就可以用文本文件的形式打开一个资源文件。

资源编辑器具有可视化的技术和界面,快速并且简单地创建和修改应用程序的资源。可以使用资源编辑器创建一个新的资源、修改一个已经存在的资源、拷贝一个已经存在的资源和删除一个旧的资源。资源编辑器主要就是为了简单而方便地进行上面的操作而设计的。当创建或者打开一个资源时,相应的资源编辑器就自动打开了。同时,也可以在一个文本格式的文件里打开一个资源文件。各个资源编辑器的使用,将在以后用到时分别介绍。

1.2.1.3 FileView

在 FileView 视图中,工程文件以树状结构列表,列表文件分为源文件、头文件、资源文件和帮助文件 4 种类型,如图 1-6 所示。

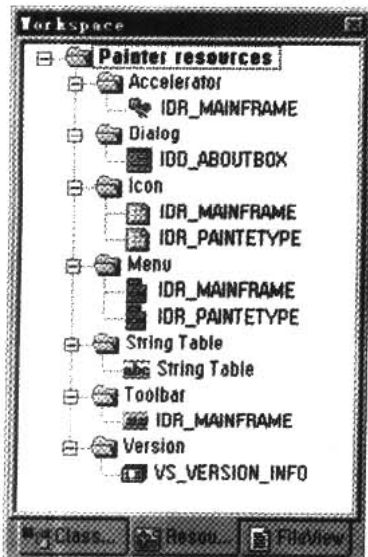


图 1-5 ResourceView 视图

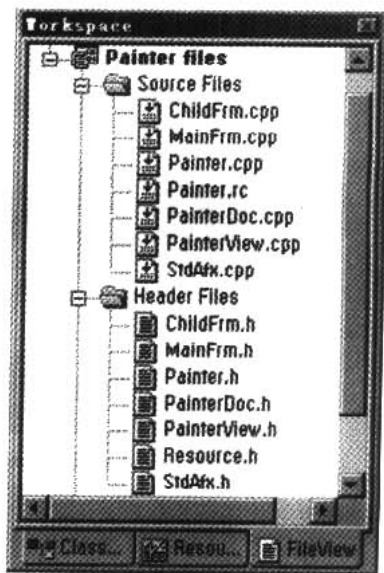


图 1-6 FileView 视图

在此视图中用鼠标右键单击文件夹,在弹出的快捷菜单中选择 New Folder,就会弹出如图 1-7 所示的新建文件夹对话框。

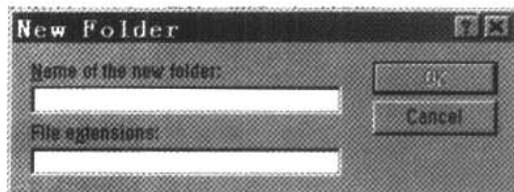


图 1-7 New Folder 对话框

在对话框中,用户可以定义文件夹的名称和其所包含文件的扩展名,从而将自定义的文件种类加入到工程中。在 FileView 视图中能够对工程文件进行添加、删除和拷贝,如果在视图中存在多个工程,则可以通过拖曳在工程之间交换文件。双击视图中列出的文件,就可以将此文件打开进行编辑。而删除文件时,只要选择要删除的文件,按下 Del 键即可。在 FileView 视图中单击鼠标右键,可以弹出快捷菜单,对项目工程和文件进行操作。其中,某些文件前向下的箭头表示该文件正在被当前设置使用。

1.2.2 主工作区窗口

用户在开发应用程序的过程中,源代码和资源的编辑工作都是在主工作区窗口中进行的。当双击工程工作区中的某一项时,在主工作区中就会出现相应的屏幕显示。例如,在工程工作区的 ClassView 视图中双击某成员函数,主工作区中就会出现该成员函数的源代码。用户可以像编辑文本那样,对源代码进行修改。如果在工程工作区的 ResourceView 视图中双击某一个资源,例如一个位图,在主工作区中就会出现位图图像。用户可以使用这时同时出现的绘图工具来编辑位图。

当在主工作区中进行代码编辑时,用户会注意到出现的代码使用了色彩调配,这就是所谓的语法着色。缺省情况下,常规代码为黑色,注释为绿色,而 C++ 关键字为蓝色。用户也可自行定义语法着色。选择 Tools|Options 菜单命令会出现如图 1-8 所示的 Options 对话框,在该对话框的 Format 标签下,可以定义语法着色的格式。

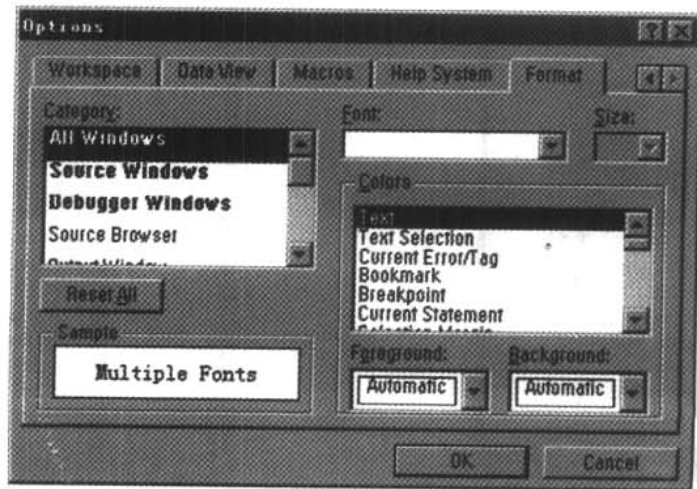


图 1-8 定义语法着色

语法着色可以帮助用户发现一些由于粗心或因拼写而造成的错误,这些错误往往不容易被发现。例如,用户在定义整型变量时,将 `int` 输入为 `itn`,由于 `itn` 不会变成蓝色,这就提醒了用户关键字拼写错误。因而,这在进行编译之前防止了许多编译错误发生。

在编辑代码时,用户还会发现输入某一个类的成员函数时,会出现一个下拉列表框对用户的输入作出提示。例如,用户声明了一个 `CMyDoc` 的类对象 `m_doc`,而在 `CMyDoc` 类中定义了 `MyFunc` 和 `HerFunc` 两个函数。那么当用户输入 `m_doc.` 时,在输入焦点处就会出现一个下拉列表框,该列表框中有两个选项: `MyFunc` 和 `HerFunc`。这种输入提示也大大方便了程序员的编程工作,它使程序员不必再费心记忆哪个类中定义了哪些成员函数。而且输入提示同样也可以防止许多错误,例如,如果用户输入的类对象名不对,那么就不会出现输入提示,这提醒用户类对象名有误。

1.2.3 输出窗口

输出窗口中有 4 个标签: `Build`、`Debug`、`Find In Files 1` 和 `Find In Files 2`,如图 1-9 所示。`Build` 标签中显示编译器、链接器和其他工具的状态消息。`Debug` 标签中用于通知来自调试器的提示,这些提示对诸如未处理的异常和内存异常之类的情况提出警告。应用程序通过 `Output-DebugString` API 函数或 `afxDump` 类库产生的消息,也显示在 `Debug` 标签中。

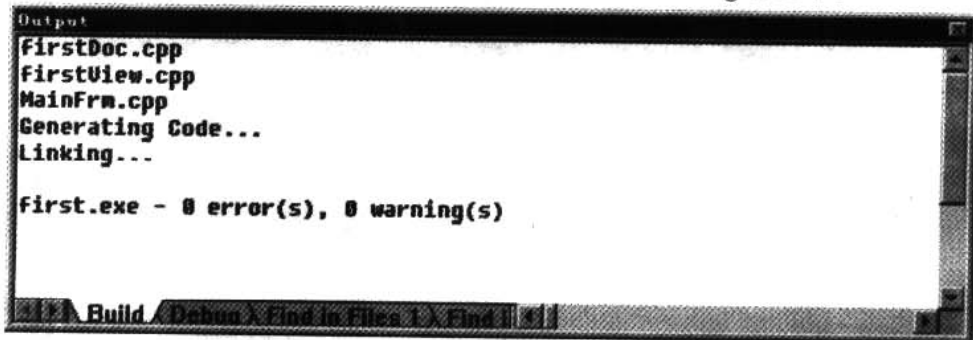


图 1-9 输出窗口

输出窗口中剩下的两个标签,显示从 `Edit` 菜单中选中的 `Find In Files` 命令的执行结果。默认情况下, `Find In Files` 搜索结果显示在输出窗口的 `Find In Files 1` 选项栏中,但 `Find In Files` 对话框中的一个复选项,允许把结果转移到 `Find In Files 2` 选项栏中。用户可以通过定制,在输出窗口中包含其他的标签。

1.3 Visual C++ 6.0 的菜单体系

Developer Studio 菜单栏包括 9 个菜单项,这些菜单项提供的命令可以完成几乎所有 Developer Studio 的功能,因此了解和掌握这些菜单命令是非常必要的。

1.3.1 File 菜单

图 1-10 所示为 `File` 菜单,包括以下菜单命令:

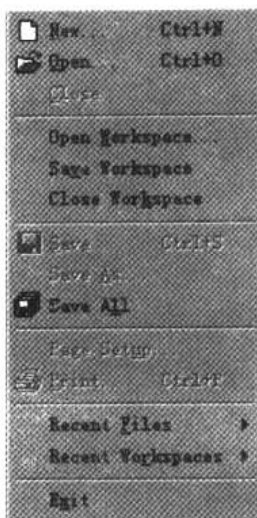


图 1-10 File 菜单

• New (Ctrl + N)——选择该菜单命令,会弹出一个带有标签的 New 对话框,如图 1-11 所示,这个对话框能够创建一个新的文件、工程、工程工作区或其他的文件。一般缺省为 Projects 标签,用于创建一个工程,这是用 Visual C++ 制作应用程序的第一步。

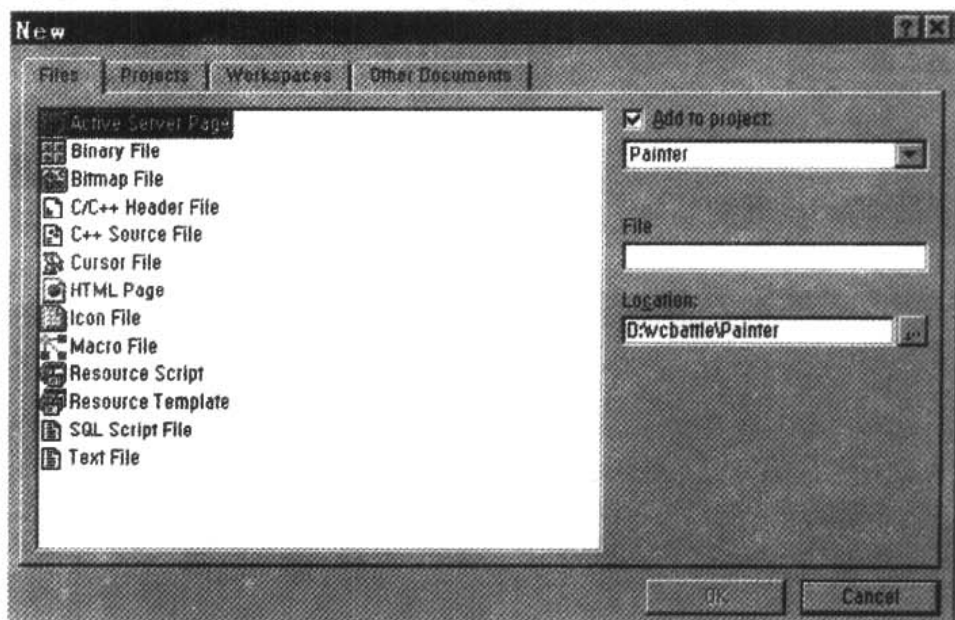


图 1-11 New 对话框

• Open (Ctrl + O)——选择该菜单命令,会弹出“打开”对话框,如图 1-12 所示,能够打开已有的文件,如 C++ 文件、宏文件、资源文件、项目文件等。对话框允许从任何已存在的目录和驱动器中选择文件,而且对话框还带一个通配符文件名文本框,用于在选择一个文件打开之前选取要查看的文件集合。对话中还有一个“以只读方式打开”的复选框,用于阅读一个文件,同时保护免遭意外的破坏。

• Close——选择该菜单命令,将关闭打开的文件,如果没有文件被打开,该菜单项为灰色。



图 1-12 “打开”对话框

如果该窗口的内容没有被保存或已经更新过,则在选择该项后,会弹出一个消息对话框,询问是否保存窗口中的内容。如果有多个文件打开,则选择该命令将关闭当前活动的或被选的窗口。

- Open Workspace——选择该菜单命令,将打开已有的工作区文件,也可用 File|Open 菜单命令,并将打开文件类型选择为 Project Workspace 的文件类型(.dsw,.mdp)。

- Save Workspace——选择该菜单命令,将保存当前工作区。

- Close Workspace——选择该菜单命令,将关闭当前工作区。

- Save (Ctrl + S)——选择该菜单命令,将保存当前活动窗口或选定窗口中的文件,如果没有文件打开,该菜单项为灰色。如果是对一个新建的窗口使用该选项,则将激活 Save As 对话框。

- Save As——选择该菜单命令,将保存一个新建的窗口或者将当前编辑窗口的内容存到一个不同的文件中去。使用这个选项将打开一个 Save As 对话框窗口,来让你指定要保存到哪个文件中,并且还可以指定目标目录和驱动器。如果没有文件打开,该菜单项为灰色。

这一项的作用主要是出于安全考虑。当并不想覆盖当前文本时,通过选择该项,可以将它存储到另一个文件中,然后在另一个文件中进行操作。如果一旦出现意外清空,还可使用原来的文件。

- Save All——选择该菜单命令,将保存当前打开的所有文件。

- Page Setup——选择该菜单命令,将会弹出如图 1-13 所示的 Page Setup 对话框,设置和格式化打印结果,标题(Header)和脚注(Footer)可包含包括一个或更多特殊域的文本。表 1-1 列出了每个格式码所对应的含义。

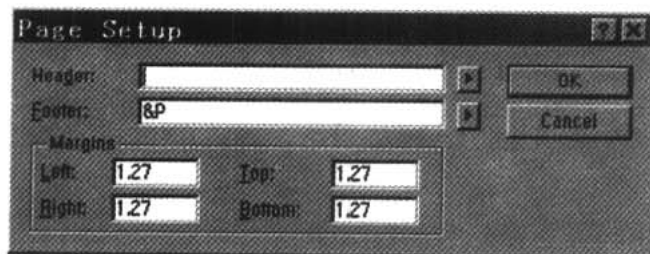


图 1-13 Page Setup 对话框

表 1-1 标题和脚注的格式码

格式码	含 义
&F	被打印的文件名称
&P	加入页号
&T	加入系统时间
&D	加入系统日期
&L	正文左对齐
&R	正文右对齐
&C	正文居中

•Print (Ctrl + P)——选择该菜单命令,将使用 Page Setup 所设置的格式打印当前活动窗口中的文件,如图 1-14 所示为 Print 对话框。如果文本被高亮显示,则 Selection 单选按钮被激活,如果选择该按钮,那么仅打印文件中被选中的文本。如果选择 All 按钮,则打印整个文件。



图 1-14 Print 对话框

•Recent Files——该级联菜单包含有最近打开的文件名,最多可列出 4 个文件名。选择其中的某一文件名,就会打开该文件。

•Recent Workspaces——该级联菜单包含有最近打开的工作区名,最多可列出 4 个工作区名。选择其中的某一工作区名,就会打开该工作区。

•Exit——选择该菜单命令,将退出 Developer Studio,如果还有修改后未经保存的文件,则会弹出提示对话框。

1.3.2 Edit 菜单

图 1-15 所示为 Edit 菜单,它包括以下菜单命令:

•Undo (Ctrl + Z)——选择该菜单命令,将恢复用户最近一次的编辑修改操作。如果 Undo 菜单项变灰,则说明没有操作需要恢复,或者是不能恢复最后一次操作。

•Redo (Ctrl + Y)——选择该菜单命令,将恢复被撤消的操作。

•Cut (Ctrl + X)——选择该菜单命令,将当前活动窗口中选定的内容剪切到剪贴板中,并覆盖掉剪贴板上原有的内容。

•Copy (Ctrl + C)——选择该菜单命令,将当前活动窗口中选定的内容复制到剪贴板中,并覆盖掉剪贴板上原有的内容。

•Paste (Ctrl + V)——选择该菜单命令,将剪贴板中的内容

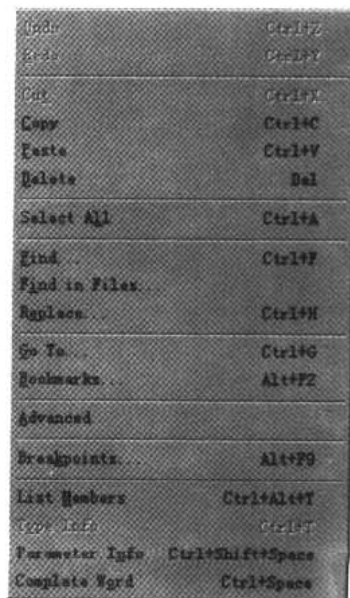


图 1-15 Edit 菜单

拷贝到当前光标所在位置,如果存在高亮被选对象,则用剪贴板中内容将其替换。

•Delete (Delete)——选择该菜单命令,将删除被选内容。

•Select All (Ctrl + A)——选择该菜单命令,将会选择当前窗口中所有内容。

•Find (Ctrl + F)——选择该菜单命令,将会弹出如图 1-16 所示的 Find 对话框,用以查找指定字符串。在 Find 对话框中,可以在 Find what 编辑框中输入字符串,并选择查找方向(Up 或 Down)。此外还可设置查询复选选项,如进行整词查找(Match whole word only)或大小写匹配查找(Match case);如果规则表述(Regular expression)被复选则对 Find what 内容作规则表述,表 1-2 为规则表达式中符号的含义;如果在所有打开文档中查询(Search all open documents)被复选,则将查询范围扩展到此时打开的所有文档。

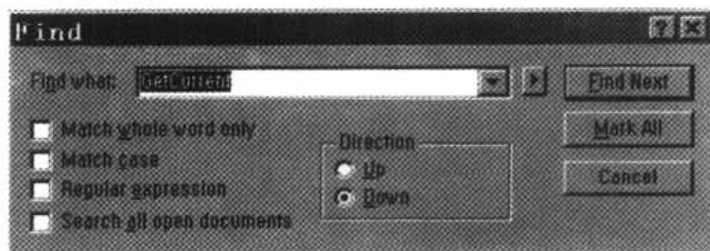


图 1-16 Find 对话框

表 1-2 规则表达式符号含义

符号	含 义	示 例
*	匹配任意多个字符	Const * 匹配 Const、Constant、Constants 等
.	匹配单个字符	Const . 匹配 Const1、Consts,但不匹配 Const11
^	匹配以指定字符串打头的每一行	Const ^ 匹配以 Const 打头的各行
+	匹配以指定字符串结尾的字符串	+ Const 匹配 123Const、EConst 等
\$	匹配以指定字符串结尾的每一行	Const \$ 匹配以 Const 结尾的各行
[]	匹配指定字符集中的字符	Const[abdefg] 匹配 Consta、Constb,但不匹配 Constl
\	匹配与指定字符一致的字符	Const[abcdE \ 1234]匹配 ConstabE1,但不匹配 ConstAd1
\{ \}	匹配花括号间指定字符串的任意顺序	Const _ \{ * # \} 匹配 Const _、Const _ * 和 Const _ # *

如果圆单击 Find 对话框中的 Mark All 按钮,将在每个匹配行中增加一个无命名书签,用蓝色的椭圆表示。

•Find in Files——选择该菜单命令,可以同时多个文件中搜索文本,图 1-17 所示为 Find In Files 对话框。可以在 In files/file types 编辑框中选择将要搜索的文件类型,并在 In folder 编辑框中选择搜索路径。对话框的下半部分复选框为搜索选项,其中 3 个选项与 Find 对话框中选项相同,而另外两个复选框的含义分别为:选中 Look in subfolders 复选框,则将在所选项目下的所有子目录下查找;选中 Output to pane 2 复选框,则将搜索结果输出到 Find in Files 2 窗格中,而且不消除以前搜索的结果。

单击 Find In Files 对话框的 Advanced 按钮,会出现如图 1-18 所示的对话框,可以允许同时在许多不同目录下搜索。

技巧:如果在选择 Edit|Find in Files 或 Edit|Find 菜单命令之前选择一个文本块,则该文本块将出现在 Find What 编辑框中。