

Visual C++ 6.0

多媒体开发指南

李博轩 编著

专题
开发指南



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



Visual C++ 6.0 多媒体 开发指南

李博轩 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书通过大量实例深入浅出地介绍了 Visual C++ 6.0 的多媒体编程技术。全书共 11 章,主要内容
包括 Visual C++ 6.0 集成开发环境及编程基础、MCI 多媒体编程技术和 DirectX 多媒体编程技术,最后实
现了一个有趣的多媒体游戏。

本书内容全面、深入,适合中高级读者、大专院校师生、企业技术开发人员学习参考,也适合各类学
习 Visual C++ 多媒体开发技术的培训班学员。

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 多媒体开发指南

作 者: 李博轩

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京密云胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 18 **字数:** 427 千字

版 次: 2000 年 2 月第 1 版 2000 年 6 月第 2 次印刷

书 号: ISBN 7-900622-82-9

印 数: 6001~11000

定 价: 35.00 元(含盘)

前 言

Visual C++ 6.0 是一种十分流行的软件开发平台,目前市场上多数有关 Visual C++ 的书籍都是快速入门一类,或对软件环境本身的介绍,很少涉及到实际应用。这样,读者在阅读完之后,只能对软件有一些零散的认识,很难用它实际地去完成独立应用程序的设计。

本书以 Visual C++ 6.0 在多媒体方面的应用为主题,重点介绍如何利用 Visual C++ 6.0 实现一个有趣的多媒体游戏。

本书以应用程序的设计思路贯穿全部内容,分别讲述多媒体原理、多媒体编程、多媒体游戏的设计和实现,内容涉及到波形音频、MIDI 音频、CD 音频、数字视频、DirectDraw、DirectInput、DirectSound 等。在这些内容中分别实现软件的功能模块,在全书结尾完成模块的集成,配套光盘中附有实例及源代码。

本书特点在于讲述如何用 Visual C++ 6.0 实现一个多媒体应用程序,而且将 Visual C++ 6.0 的多媒体特性和技术难点融入到具体的实例中,不强调“大而全”,而侧重“专而精”。

参与本书编写的还有吉尚戎、吉二源、吴灵、刘秀蓉、杜丽、何震声、宋森、陈明、李洪声、刘海涛、李敏、刘志诚、朱志言、刘兵等,王宇红、李兵参与了审校与录排工作,在此对他们的辛勤劳动表示感谢。

作 者

1999 年 9 月

目 录

第 1 章 Visual C++ 6.0 简介	1
1.1 Visual C++ 6.0 概述	1
1.2 Visual C++ 6.0 用户界面	2
1.2.1 工程工作区	2
1.2.2 Developer Studio 的菜单体系	6
1.3 使用 ClassWizard	20
1.3.1 Message Maps 标签	20
1.3.2 Member Variables 标签	21
1.3.3 Automation 标签	22
1.3.4 ActiveX Events 标签	23
1.3.5 Class Info 标签	24
1.4 建立第一个 Windows 应用程序	24
1.4.1 建立应用程序框架	25
1.4.2 应用程序代码分析	31
1.4.3 Hello, World!	34
1.5 本章小结	35
第 2 章 多媒体编程基础	36
2.1 多媒体文件	36
2.1.1 多媒体文件格式	36
2.1.2 从 RIFF 文件中获得信息	37
2.2 媒体控制接口(MCI)	38
2.2.1 MCI 设备类型	38
2.2.2 MCI 函数接口	38
2.2.3 MCI 命令消息	41
2.3 DirectX 概述	43
2.3.1 DirectX 的优越性	43
2.3.2 COM 与 DirectX 对象	43
2.3.3 DirectX 对象简介	44
2.4 本章小结	45
第 3 章 波形音频	46
3.1 波形音频概述	46
3.1.1 波形音频文件的质量	46

3.1.2	波形音频文件的格式	47
3.2	波形音频的处理函数	50
3.3	使用 MessageBeep 播放波形音频	50
3.4	使用 PlaySound 播放波形音频	51
3.5	使用 MCI 播放波形音频	54
3.5.1	波形音频常用的 MCI 命令和标志	54
3.5.2	波形音频的 MCI 编程步骤	55
3.5.3	建立处理波形音频的类	56
3.6	设计一个波形音频播放器	62
3.6.1	创建应用程序框架	62
3.6.2	制作应用程序界面	62
3.6.3	添加消息处理函数	63
3.6.4	完成应用程序功能设计	64
3.7	使用低级设备函数播放波形音频	71
3.7.1	波形低级设备函数	71
3.7.2	低级波形设备函数编程	72
3.8	本章小结	74
第 4 章	MIDI 音频	75
4.1	MIDI 概述	75
4.1.1	MIDI 消息	75
4.1.2	MIDI 文件的格式	76
4.2	使用 MCI 播放 MIDI	77
4.2.1	MIDI 常用的 MCI 命令和标志	77
4.2.2	MIDI 的 MCI 编程步骤	78
4.2.3	建立处理 MIDI 的类	80
4.3	设计一个 MIDI 播放器	89
4.3.1	创建应用程序框架	89
4.3.2	制作应用程序界面	89
4.3.3	添加消息处理函数	90
4.3.4	完成应用程序功能设计	91
4.4	使用低级设备函数播放 MIDI 音频	97
4.4.1	MIDI 低级设备函数	97
4.4.2	MIDI 低级设备函数编程	98
4.5	本章小结	99
第 5 章	CD 音频	100
5.1	CD 音频概述	100
5.2	使用 MCI 播放 CD	100
5.2.1	CD 常用 MCI 命令和标志	101

5.2.2	CD 音频的 MCI 编程步骤	101
5.2.3	建立处理 CD 音频的类	103
5.3	设计一个 CD 音频播放器	116
5.3.1	创建应用程序框架	116
5.3.2	制作应用程序界面	117
5.3.3	添加消息处理函数	118
5.3.4	完成应用程序功能设计	118
5.4	本章小结	125
第 6 章	数字视频	126
6.1	数字视频概述	126
6.2	使用 MCI 播放数字视频	126
6.2.1	数字视频的常用 MCI 命令和标志	127
6.2.2	数字视频的 MCI 编程步骤	129
6.2.3	建立处理数字视频的类	132
6.3	设计一个数字视频播放器	142
6.3.1	创建应用程序框架	142
6.3.2	制作应用程序界面	142
6.3.3	添加消息处理函数	145
6.3.4	完成应用程序功能设计	146
6.4	本章小结	153
第 7 章	制作媒体播放器	154
7.1	制作处理媒体文件的动态链接库	154
7.1.1	动态链接库概述	154
7.1.2	制作自己的动态链接库	155
7.2	设计媒体播放器	156
7.2.1	建立应用程序框架	156
7.2.2	制作应用程序界面	156
7.2.3	应用程序完成的功能	159
7.2.4	应用程序的设计	159
7.3	本章小结	168
第 8 章	DirectDraw 基础	169
8.1	DirectDraw 概述	169
8.1.1	DirectDraw 的优越性	169
8.1.2	DirectDraw 术语	170
8.1.3	DirectDraw 功能	174
8.2	DirectDraw 数据结构	174
8.2.1	DDBLTBATCH 结构	175
8.2.2	DDBLTFX 结构	177

8.2.3	DDCAPS 结构	179
8.2.4	DDCOLORKEY 结构	180
8.2.5	DDOVERLAYFX 结构	180
8.2.6	DDPIXELFORMAT 结构	181
8.2.7	DDSCAPS 结构	182
8.2.8	DDSURFACEDESC 结构	185
8.3	DirectDraw API	186
8.3.1	DirectDraw 对象	187
8.3.2	DirectDrawSurface 对象	194
8.3.3	DirectDrawPalette 对象	206
8.3.4	DirectDrawClipper 对象	207
8.4	本章小结	209
第 9 章	DirectDraw 应用	210
9.1	DirectDraw 基础编程	210
9.1.1	DirectDraw 对象编程	210
9.1.2	Hello, DirectDraw!	215
9.2	使用 DirectDraw 显示位图	222
9.2.1	调色板的创建和设置	222
9.2.2	将位图载入后备表面	224
9.3	离屏表面和 Blitting 编程	227
9.3.1	创建离屏表面	228
9.3.2	将位图载入离屏表面	228
9.3.3	Blitting 操作编程	229
9.4	颜色键和位图动画	231
9.4.1	设置颜色键	231
9.4.2	制作动画效果	233
9.5	调色板动画	237
9.5.1	载入调色板项	237
9.5.2	改变调色板	238
9.6	设计 DirectDraw 可重用函数	238
9.7	本章小结	241
第 10 章	DirectInput 和 DirectSound	242
10.1	DirectInput 概述	242
10.1.1	DirectInput 与 Windows	242
10.1.2	DirectInput 术语	242
10.1.3	DirectInput 设备常量	244
10.2	DirectInput API	245
10.2.1	IDirectInput 接口 API	245

10.2.2	IDirectInputDevice 接口 API	247
10.3	DirectInput 设备编程	251
10.3.1	DirectInput 键盘编程	252
10.3.2	DirectInput 鼠标编程	256
10.3.3	DirectInput 游戏杆编程	263
10.4	DirectSound 编程	268
10.4.1	DirectSound 对象初始化	268
10.4.2	创建 DirectSoundBuffer 对象	268
10.4.3	播放声音	269
10.4.4	销毁 DirectSound 对象	270
10.5	本章小结	271
第 11 章	StackUp	272
11.1	游戏概述	272
11.2	游戏设计要点	273
11.2.1	游戏程序结构设计	273
11.2.2	CStackUpApp 类	274
11.2.3	CStackUpWnd 类	276
11.2.4	CStackUp 类	276
11.3	本章小结	280

第 1 章 Visual C++ 6.0 简介

Visual C++ 6.0 是当今最流行的软件开发工具之一,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 用户界面
- ClassWizard 的使用
- Windows 编程入门

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 3 个版本:

1. 标准版

Visual C++ 6.0 标准版具有除代码优化、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。

2. 专业版

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。

3. 企业版

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。

与以前版本比较,Visual C++ 6.0 具有许多新特性。例如提供了 ODBC 类和高性能的 32 位 ODBC 驱动程序以方便用户对各种数据库管理系统的访问,从而建立强有力的数据库应用程序;可以使用 DAO(数据访问对象)类通过编程语言来访问和操纵数据库中的数据,并管理数据库、数据库对象与结构;为 Internet 提供了强有力的支持:WinInet 使 Internet 成为应用程序的一部分,并简化了对 Internet 服务,例如 FTP、HTTP 和 Gopher 的访问;可以使用 CHttpServer、CHttpFilter、CHttpSeverContext、CHttpFilterContext 和 CHttpStream 类来创建动态链接库,以便把功能添加到 Internet 服务器和 Web 页中。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。

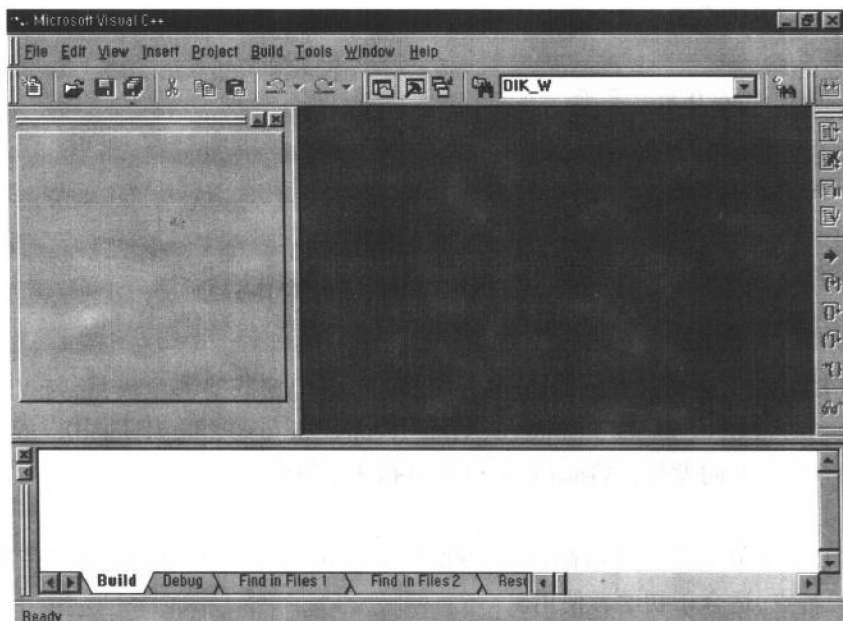


图 1-1 Developer Studio 的用户界面

Developer Studio 用户界面包括上端的菜单和工具栏、左边的工程工作区 (Project Workspace)、右边进行文件与资源编辑的主工作区以及下端的输出窗口和状态栏。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

图 1-2 为打开一个工程或新建一个工程后的工程工作区窗口,在窗口下端有 3 个标签: Class View、Resource View 和 File View。

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。每一个类首先列出带有紫色图标的成员函数,然后是有绿蓝色图标的



图 1-2 打开一个工程后的工程工作区窗口

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等信息,例如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。用鼠标右键单击某个类名,可以弹出一个快捷菜单以对类进行操作,如图 1-3 所示。此菜单中包括下列菜单项:

- Go to Definition 打开类的头文件。
- Add Member Function 打开增加成员函数的对话框。
- Add Member Variable 打开增加成员变量的对话框。
- Add Virtual Function 打开虚函数重载对话框。
- Add Windows Message Handler 打开增加新窗口消息处理程序对话框。
- References 产生应用程序中所有类名位置的列表。
- Derived Classes 产生此类中所有成员变量、成员函数以及该类的所有派生类列表和附注信息。
- Base Classes 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 将该类添加到组件库中。
- New Folder 产生允许用户拖放类的文件夹。

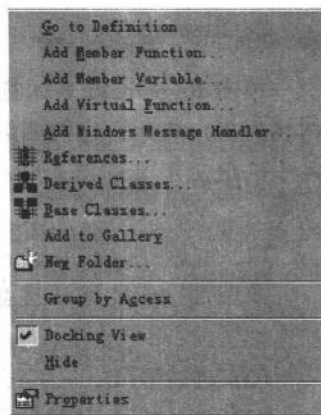


图 1-3 类的快捷菜单

- **Group by Access** 选择此选项时,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** 使工程工作区定位在主工作区边缘。
- **Hide** 隐藏工程工作区,使用 View|Workspace 菜单项能够重新显示工程工作区。
- **Properties** 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:

- **Go to Definition** 打开定义该成员函数的源文件。
- **Go to Declaration** 打开说明该成员函数的头文件。
- **Set Breakpoint** 设置断点。
- **References** 产生该成员函数的位置列表。
- **Calls** 产生该成员函数所调用的所有函数列表。
- **Called by** 产生所有调用该成员函数的函数列表。
- **New Folder** 建立一个允许用户拖放类的文件夹。
- **Group by Access** 选择此选项时,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。

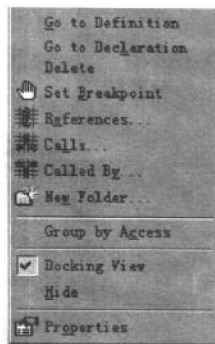


图 1-4 类成员的快捷菜单

2. Resource View

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。应用程序的资源包括快捷键、位图、图标、对话框、菜单、字符串表、工具栏以及版本信息。用鼠标双击 ResourceView 视图中的资源名称,就会在主工作区打开相应的资源。

(1) 加速键(Accelerator)

使用加速键能够从键盘执行菜单命令。在创建新的应用程序时,由 AppWizard 产生加速键资源,用户也可以为特定的菜单项自定义加速键。

(2) 对话框(Dialog Box)

对话框是应用程序与用户之间进行交互的最便捷的方法。当打开一个对话框资源时,旁边同时会出现一个控件栏,如果控件栏没有出现,那么用鼠标右键单击菜单栏,在出现的快捷菜单中选择 control 项即可。控件栏上的每个图标都代表一个可以插入对话框中的控件(如编辑框、列表框、按钮,等等)。用户能够简单地通过拖动将控件加入到对话框中,并调整其位置,使用 Layout 菜单中的命令能够很方便地调整对话框的布局。

将控件放入对话框资源中后,可以对这些控件的属性进行修改。用鼠标右键单击控件,在弹出的快捷菜单中选择 Properties,就可以打开属性对话框。在此对话框中能够对控件的属性值,如 ID 值、标题(Caption)等进行修改。

测试对话框时,可以通过单击工具条中的测试按钮来进行简单测试。

(3) 图标(Icon)

图标是一种 32×32 的点位图,通常可用图标表示最小化的应用程序,用户可以自己使用 Developer Studio 所提供的位图编辑器绘制应用程序的图标。

(4) 菜单(Menu)

菜单是应用程序的命令集,它体现了程序的功能,用户可在资源视图中对菜单资源进行编辑或增加新的菜单项。添加菜单项时,双击前一个菜单项的右部或底部,就会打开菜单项的属性对话框,以输入数字菜单标题、ID 值和各种风格信息。通过拖曳可以重新编排菜单项的顺序。

菜单项属性中的 Resource ID 唯一地标识了该菜单项,相应的加速键和工具栏按钮的 Resource ID 都与之相同。菜单项的标题(Caption)是在菜单栏中所显示的菜单项名称,在标题字母前面加入“&”,可以使该字母成为该菜单项的加速键。菜单项的 Prompt 文本是当菜单项被高亮显示或光标停留在和该菜单项相关联的工具栏按钮上时出现在状态条上的内容。新行分隔符(\n)之前的文本能在状态栏上显示,而其后的部分是在用户将鼠标停留在具有相同资源 ID 的工具栏按钮时出现的有关的工具信息。如果选择 Separator,则产生一个菜单间的分隔线。Checked 风格、Grayed 风格和 Inactive 风格可以决定菜单项的初始状态,但这些状态在应用程序运行时可以改变。Help 风格使菜单项处于菜单条的最右方;Pop-up 风格可以创建一个弹出式菜单,将菜单进行分层。

(5) 字符串表(String table)

字符串表是应用程序中串的列表。要改变字符串,需要打开属性对话框并改变标题,不过在主工作区内显示的字符串不能改变。

(6) 工具栏(Toolbar)

工具栏是一排小按钮,通常位于应用程序菜单下面。每个按钮都与相应的菜单项相连,并且其状态取决于菜单项的状态。如果菜单项为灰色,对应的工具栏按钮也为灰色;如果菜单项为复选,相应的工具栏按钮通常被画成压进按钮。工具栏按钮包括按钮位图和,并且其资源 ID 与相应菜单项的资源 ID 相同。

(7) 版本信息(Version)

AppWizard 为应用程序自动生成版本信息。在对已安装过的一个应用程序进行重安装时,版本信息可以提醒用户是否要覆盖旧版本。

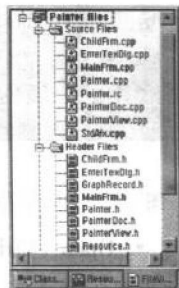


图 1-5 File View 视图

3. FileView

图 1-5 为 FileView 视图。

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。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,就会弹出一个如图 1-6 所示的对话框。

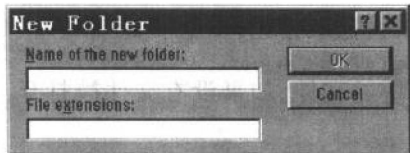


图 1-6 New Folder 对话框

在对话框中,用户可以定义文件夹的名称及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 Developer Studio 的菜单体系

Developer Studio 菜单栏包括下列 9 个菜单项,以下各节将依次进行介绍。

1. File 菜单

图 1-7 为 File 菜单,包括以下菜单命令:

- New (Ctrl + N) 选择该菜单项,会弹出一个带有标签的 New 对话框,如图 1-8 所示,这个对话框能够创建新的文档、工程或工作区。

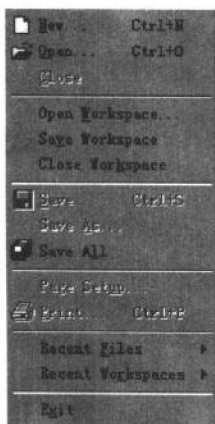


图 1-7 File 菜单

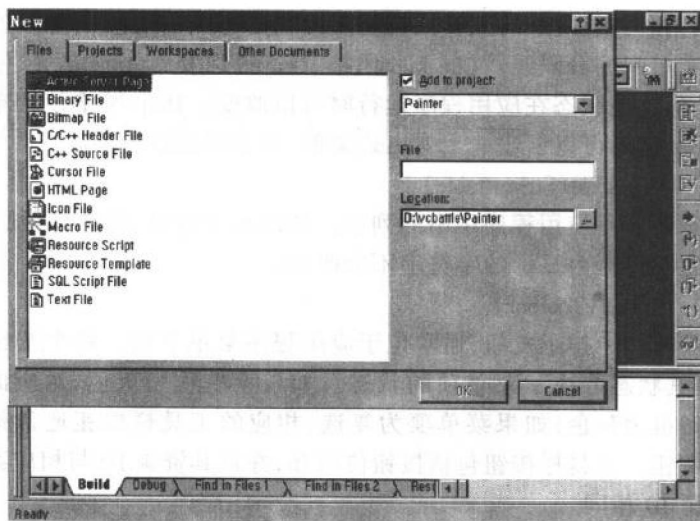


图 1-8 New 对话框

- Open (Ctrl + O) 选择该菜单项,会弹出 Open 对话框,如图 1-9 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。
- Close 选择该菜单项将关闭打开的文件,如果没有文件被打开,该菜单项为灰色。
- Open Workspace 打开已有的工作区文件,也可用 File|Open 命令,并将打开文件的类型变成 Project Workspace 的文件类型(.dsw, .mdp)。
- Save Workspace 保存打开的工作区。
- Close Workspace 关闭打开的工作区。
- Save (Ctrl + S) 保存当前活动窗口或选定窗口中的文件,如果没有文件被打开,该菜单项为灰色。
- Save As 将打开的文件用新的文件名加以保存,如果没有文件被打开,该菜单项为灰色。



图 1-9 Open 对话框

- **Save All** 保存当前打开的所有文件。
- **Page Setup** 选择该菜单项,会弹出如图 1-10 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)可包含一个或更多特殊域的文本。可通过鼠标单击编辑框后面的箭头增加数值,或者是自己输入。表 1-1 列出了每个格式码所对应的含义。

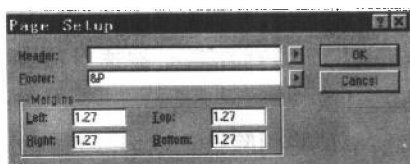


图 1-10 Page Setup 对话框

表 1-1 标题和脚注的格式码

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

- **Print (Ctrl + P)** 使用 Page Setup 所设置的格式打印当前活动窗口中的文件,图 1-11 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。

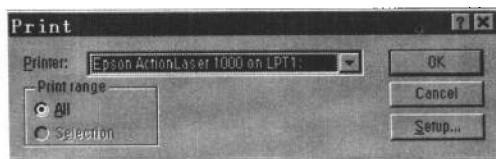


图 1-11 Print 对话框

- **Recent Files** 包含有最近打开的文件名,最多可列出 4 个文件名。
- **Recent Workspaces** 包含有最近打开的工作区名,最多可列出 4 个工作区名。
- **Exit** 退出 Developer Studio,如果还有尚未保存的修改,则会出现是否保存改变的提示。

2. Edit 菜单

图 1-12 为 Edit 菜单,它包括以下菜单命令:

- **Undo (Ctrl + Z)** 选择该菜单项能够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)** 选择该菜单项能够恢复被 Undo 命令取消的操作。
- **Cut (Ctrl + X)** 选择该菜单项将剪切当前活动窗口中选定的内容到剪贴板中。
- **Copy (Ctrl + C)** 选择该菜单项将复制当前活动窗口中选定的内容到剪贴板中。
- **Paste (Ctrl + V)** 选择该菜单项将剪贴板中的内容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 (Del)** 选择该菜单项将删除被选内容。
- **Select All (Ctrl + A)** 选择该菜单项将会选择当前窗口中的所有内容。
- **Find (Ctrl + F)** 选择该菜单项将会弹出如图 1-13 所

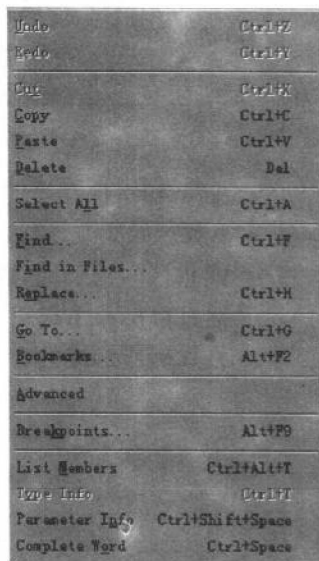


图 1-12 Edit 菜单

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,则将查询范围扩大到此时代打开的所有文档。

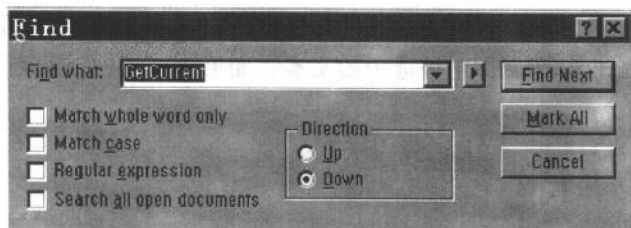


图 1-13 Find 对话框