

Visual c++ 6.0

应用与 开发指南

宗 平 编著 ●
黄明军 审校 ●

人民邮电出版社

内容提要

本书全面系统地介绍了 Visual C++ 6.0 的应用与开发。全书首先对 Visual C++ 6.0 主要功能特点、软硬件配置及集成开发环境的使用做了详细介绍,对于 Visual C++ 的入门者,此部分不失为很好的学习向导;为了充分向读者展示 MFC 类库编程的特点,本书重点阐述了使用 MFC 开发 Windows 的应用程序;在本书的最后部分对动态链接库及 ActiveX 编程做了简单介绍。全书每章均有相配套的例程及编程方法的讲解,因此尽管内容繁多,读者并不会感到枯燥。

本书十分适合个人自学或作为企事业单位培训 Visual C++ 6.0 的中级教材,也可以作为大专院校计算机课程的参考书。

Visual C++ 6.0 应用与开发指南

-
- ◆ 编 著 宗 平
审 校 黄明军
责任编辑 刘华鲁
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本:787×1092 1/16
印张:27
字数:677 千字
印数:1-5 000 册
 - 1999 年 2 月第 1 版
1999 年 2 月北京第 1 次印刷

ISBN 7-115-07635-9/TP·995

定价:37.00 元

3500 册

前 言

近年来,随着计算机技术的发展,人们对计算机编程环境的要求也与日剧增。在这种情况下,可视化编程环境的产生是必然的。Visual C++是微软公司推出的应用非常广泛的可视化编程语言,它也是 Visual Studio 可视化编程环境中的一员。

从 Visual C++ 1.5 到 1998 年推出的 Visual C++ 6.0, VC 语言的发展也经历了一个不断提高和不断改进的过程。本书将要讲解的 Visual C++ 6.0 具有如下的主要特点:

- ◆ Visual Studio 环境有一整套工具用于 Win32 应用程序的开发。
- ◆ 功能强大的编程向导工具(MFC AppWizard、ClassWizard、MFC ActiveX ControlWizard、ATL COM AppWizard、Custom AppWizard 和 ISAPI Extension Wizard)。这些工具使得 Win32 应用程序的开发大大简化,并且也提高了编程效率。
- ◆ MFC 支持多线程应用程序的开发
- ◆ 提供对数据库快速高效的访问。可以使用 ODBC 驱动程序对大多数数据库进行访问,也可以使用 DAO(数据访问对象)类通过编写相应的代码来管理数据库。
- ◆ 对 Internet 编程的更强有力的支持。在 Visual C++ 6.0 的类库中添加了 Internet Explorer 4.0 的通用控件类。MFC 类库封装了 ActiveX 技术,利用 MFC 可以很方便地开发 Internet 和桌面应用程序。MFC 还使用 CHttpServer、CHttpFilter、CHttpServerContext 等类对 ISAPI(Internet Server API)进行了封装,这些类可以帮助我们创建动态链接到 Internet 服务器和 Web 页中。

可以看出,可视化编程语言相对于其他编程语言有着不可比拟的优越性,但另一方面,掌握可视化编程技术也比掌握普通的编程语言有着更大的难度。因此,广大计算机爱好者希望能够在计算机图书市场上看到更多更好的可视化编程的图书。

但是,目前市面上的有些可视化编程的图书往往过于注重“可视化”几个字,忽视了传统的计算机图书所注重的编程语言和编程技术原理的阐述,而只是简单地告诉读者应该如何如何做就可以生成一个怎样的应用程序。读过这类图书的读者可能都有这样的印象,读完书后对于书中介绍过的应用或与其相似的应用可以独立生成,但却很难做到举一反三。这显然是不适应多种多样的计算机程序设计的要求的。

Visual C++ 6.0 作为新一代的可视化编程工具,与其相关的内容是相当复杂和深奥的,仅仅通过一两本书不可能面面俱到。

因此本书的出发点是力求在浅显易懂的基础上多为读者留一些思考和钻研的线索。对于入门者可以通过本书初步掌握 Visual C++ 6.0 开发环境的使用及较简单的应用程序的生成与开发。对于已经具备了一定的可视化编程语言的知识的读者,可以通过阅读本书获得一些继续提高的突破口。

本书共分十章,其中主要对 Visual C++ 6.0 的基本类库(MFC)编程的一些基本问题,如窗口的生成和销毁、对话框的程序设计、多文档应用和单文档应用等进行了详细的介绍,并对每一章的例程都给出了具体的创建步骤、相应的源程序及注释。关于 MFC 类库

的高级应用，本书主要介绍了简单的 ActiveX 编程和动态链接库的编程。本书的附录为不熟悉 C++ 语言的读者编写了“C++ 语言速成”，并给出了 MFC 类库中最常用的一些类的定义及类成员的说明，相信这对于爱钻研的读者会有帮助。

由于时间仓促，加之水平有限，不当之处在所难免，尚请读者批评指正。

作者

1998 年 12 月

目 录

第一章 Microsoft Windows 95/NT 和 Visual C++	1
1.1 Windows 编程基础	1
1.1.1 基于消息的操作系统	1
1.1.2 Windows 的图形设备接口	1
1.1.3 全新的 32 位内存管理	2
1.1.4 动态链接库(DLL)	2
1.2 Visual C++ 6.0 概述	2
1.2.1 Visual C++ 6.0 的软硬件配置	2
1.2.2 Microsoft 基本类库应用框架	3
1.2.3 App Studio 资源编辑器	3
第二章 Visual C++集成开发环境	5
2.1 Visual C++ 6.0 的主窗口	5
2.2 Visual C++ 6.0 的工具栏	6
2.3 Visual C++的菜单栏	7
2.3.1 File 菜单	72
2.3.2 Edit 菜单	11
2.3.3 View 菜单	17
2.3.4 Project 菜单	18
2.3.5 Build 菜单	20
2.3.6 Tools 菜单	22
第三章 创建第一个 Windows 应用程序	26
3.1 用 MFC AppWizard 生成应用程序 MyApp	26
3.2 CWinApp 类和文档模板	32
3.3 窗口类和窗口对象	40
3.3.1 CWnd 及其派生类	40
3.3.2 窗口风格	41
3.3.3 窗口的客户区与非客户区	41
3.3.4 窗口对象	41
3.3.5 窗口的注册和创建	41
3.3.6 销毁窗口	42
3.3.7 MyApp 应用程序的边框窗口代码	42
3.4 MyAppView 视类	46

3.5 在视窗内绘图	47
3.5.1 OnDraw()成员函数	47
3.5.2 Windows 设备环境	47
3.5.3 往 MyApp 应用中加入绘图代码	47
3.5.4 视类代码清单	50
第四章 消息和命令	54
4.1 消息分类	54
4.1.1 标准 Windows 消息	54
4.1.2 控件通知(Control Notification)	55
4.1.3 命令消息(WM_COMMAND)	56
4.2 消息的处理——用 ClassWizard 添加消息映射	56
4.3 MyRectangle 应用的说明	60
4.3.1 m_Rect 数据成员	60
4.3.2 CPoint 和 CRect 之间的运算	60
4.3.3 设备坐标系和客户坐标系	60
4.3.4 使矩形无效	61
4.3.5 进一步熟悉消息映射	62
4.3.6 MyRectangle 应用源程序清单	63
4.4 发送和接收命令消息	78
4.4.1 命令消息的传递路线	78
4.4.2 加入消息映射入口	79
4.4.3 更新命令用户接口消息	79
第五章 Windows 的图形处理	81
5.1 图形设备接口	81
5.1.1 设备环境类	81
5.1.2 GDI 对象	85
5.1.3 映射方式	91
5.1.4 坐标变换	93
5.2 位图和位操作	94
5.2.1 设备相关和设备无关位图	95
5.2.2 位图文件结构	95
5.2.3 从资源中装入位图	96
5.2.4 读取 DIB 图像数据	99
第六章 对话框程序设计	106
6.1 对话框	106
6.1.1 对话框的组成	106

6.1.2	CDialog 类	106
6.1.3	有模式和无模式对话框	110
6.1.4	创建对话框资源	110
6.1.5	通用对话框类	115
6.2	控件	118
6.2.1	标准 Windows 控件	118
6.2.2	往对话框中添加控件	124
6.2.3	对话框控件的访问	124
6.3	有模式对话框	126
6.3.1	有模式对话框的程序设计	127
6.3.2	一个实际的有模式对话框的创建——MyModelDlg 例程	127
6.3.3	对话框成员变量和控件的数据交换	133
6.3.4	CMyModelDlg 应用的源程序清单	137
6.4	无模式对话框	152
6.4.1	无模式对话框的创建和销毁	152
6.4.2	CFormView 类	153
6.4.3	一个实际的无模式对话框的例子——MyModelessDlg 例程	155
6.4.4	测试 MyModelessDlg 应用及完整的源程序	162
第七章	文档界面窗口	184
7.1	文档的序列化	184
7.1.1	CFile 和 CArchive 类	184
7.1.2	使类可序列化	184
7.1.3	添加文档中的数据成员	184
7.1.4	编写序列化代码	185
7.2	文档—视结构	187
7.2.1	在 MFC 应用中各对象之间的关系	187
7.2.2	文档和视的相互做用	187
7.3	单文档应用(SDI)	190
7.3.1	切分窗口和多视处理	190
7.3.2	一个实际的 SDI 应用	194
7.3.3	测试 MyCircles 应用和应用程序源代码	201
第八章	多文档接口程序设计	226
8.1	MDI 应用	226
8.1.1	文档模板类	226
8.1.2	CMDIFrameWnd 类	226
8.1.3	修改窗口风格	229
8.1.4	CMDIChildWnd 类	230

8.2 滚动处理	232
8.2.1 滚动处理函数	232
8.2.2 滚动消息响应函数	233
8.3 管理 Windows 菜单和加速键	234
8.3.1 键盘加速键	234
8.3.2 CMenu 类	236
8.4 一个实际的 MDI 应用	238
8.4.1 MDIApp 应用创建的简单过程	238
8.4.2 MDIApp 应用的源代码	242
第九章 动态链接库	251
9.1 普通 DLL 的编程与使用	251
9.1.1 DLL 与应用的连接	251
9.1.2 从 DLL 导出函数	252
9.1.3 利用 MFC AppWizard 创建普通 DLL	253
9.1.4 测试 MyDll 应用及源代码清单	258
9.2 MFC 扩展 DLL 编程	267
9.2.1 MFC 扩展 DLL 的内存使用	267
9.2.2 编写 MFC 扩展 DLL 所需要的一些代码	268
9.2.3 扩展 DLL 的引出	269
9.2.4 一个实际的 MFC 扩展 DLL	270
9.2.5 测试 ExtensionDll 应用	274
9.2.6 ExtensionDll 应用的完整源代码	278
第十章 ActiveX 编程简介	288
10.1 什么是 ActiveX	288
10.1.1 ActiveX 文档	288
10.1.2 ActiveX 控件	289
10.1.3 ActiveX 控件容器	314
10.1.4 ActiveX 自动化服务器	314
10.1.5 COM 对象	315
10.2 生成一个 ActiveX 控件	315
10.2.1 生成 ActiveX 控件基本框架	315
10.2.2 为 MyActiveX 控件增加属性	317
10.2.3 ActiveX 事件处理	322
10.2.4 控件的属性页处理	324
10.2.5 MyActiveX 项目的完整源代码	325
10.3 OLE 容器的设计	346
10.3.1 生成 MyContainer 应用程序框架	346

10.3.2 查看 MyContainer 代码	350
附录 A C++ 语言速成	361
A.1 C++ 语言基本概念	361
A.1.1 类定义	361
A.1.2 对象	362
A.1.3 构造函数和析构函数	363
A.1.4 成员函数的实现	364
A.1.5 内联成员函数	365
A.1.6 this 指针	366
A.1.7 静态类成员	366
A.1.8 友元函数	367
A.1.9 指针和引用	368
A.2 类继承	369
A.2.1 派生类与基类	369
A.2.2 父类有构造函数的继承	370
A.2.3 多重继承	372
A.3 多态性	374
A.3.1 函数重载	374
A.3.2 重载运算符	375
A.3.3 虚函数	376
附录 B MFC 类的概述	380
B.1 CObject 类	380
B.2 CFile 类	382
B.3 Exception 类	385
B.4 图形绘制类	386
B.5 集合类	396
B.5.1 数组类	397
B.5.2 列表类	398
B.5.3 映射类	401
B.6 窗口类	403

第一章 Microsoft Windows 95/NT 和 Visual C++

Microsoft Windows 已经被人们广泛接受,人们越来越多地体会到了图形用户界面给人们带来的好处,关于这点许多书籍都有过专门介绍。本章总结了 Windows 编程所必需具备的最基本的知识,并且介绍了 Visual C++ 的各组成部分是如何相互配合来帮助人们编写 Windows 应用程序的。通过本章的学习读者可以对 Windows 环境下的 Visual C++ 6.0 编程有一个全面的了解。

1.1 Windows 编程基础

尽管 Visual C++ 6.0 的编程环境比传统的 SDK 编程有着很大的优越性,作为一种开发工具,Visual C++ 仍然是基于 Windows 系统的,在以后的介绍中我们可以看到,Visual C++ 的编程过程将会大量直接或间接调用 Windows API 函数。因此,在本书的开头对 Windows 的编程特点做一个大概的介绍是完全必要的,它将十分有利于初学者对 Visual C++ 编程的学习。

1.1.1 基于消息的操作系统

在 Windows 中,每一个窗口与用户、系统及其他窗口进行通信的最主要和最直接的方式就是消息传递。在 Windows 中定义了上百种不同的消息,它们各自有不同的功能,这使得系统和用户想要窗口做的任何操作都可以通过向窗口发送一条或几条消息来实现。举一个很简单的例子,当使用鼠标在窗体上拖动时,系统会不停地向窗口发送 WM_MOUSEMOVE 消息,如果做这个动作时希望窗口进行相应的动作,只要在相应的消息映射函数中加入执行代码即可实现。

在 MFC 编程中,我们也经常调用 SendMessage 或 PostMessage 函数来发送消息,但更多的情况下,把 C++ 的代码转化为适当的消息的工作是由系统来完成的。

1.1.2 Windows 的图形设备接口

以往许多的 MS-DOS 程序都直接向打印机接口或视频存储区输送数据,这种做法的弊病是显而易见的,而 Windows 则在用户和硬件最底层之间设有一个图形设备接口(GUI)。

这样用户就可以通过 GDI 函数来同硬件打交道，而 GDI 函数会自动参照一定的称为设备环境的数据结构来对硬件进行操作。所以说，Windows 的图形设备接口使得用户在更高的层次上间接地与硬件进行相互通信，而不必去理会具体的一些细节问题。

1.1.3 全新的 32 位内存管理

可能读者还有印象，在 MS-DOS 系统下，应用程序只能加载于内存地址开始的 640K 地址空间中，这大大限制了应用程序的大小。尽管可以使用扩充和扩展内存管理器来弥补这一缺陷，但作为瓶颈的 640K 限制并没有完全消除。而 Windows 提供了全新的内存管理机制，它使得每个应用程序都有自己的独立的 4GB 的虚拟地址空间，每个应用程序可以在自己的虚拟空间中随意分配内存，Windows 系统将各个应用程序的虚拟空间映射到实际的物理内存并且保证不同的应用程序所占用的物理空间不会冲突。有关 Windows 的 32 位内存管理技术，我们将在以后的章节中做更详细地介绍。

1.1.4 动态链接库(DLLs)

在 Windows 环境下，动态链接库是在程序运行过程中被动态地装入和连接的，并且多个应用程序可以共享同一个动态链接库。这避免了冗余的代码，从而提高了程序开发的效率。

在 Visual C++ 中，动态链接库始终是一个很重要的部分，本书也将在第九章中做专门介绍。

1.2 Visual C++ 6.0 概述

近年来，随着图形操作界面的普及，可视化技术得到了越来越广泛的重视。特别是可视化的编程环境更是受到软件开发人员的青睐。Visual C++ 就是 Microsoft 公司推出的可视化编程的集成环境。

从 Visual C++ 1.0 推出到 Visual C++ 6.0 的诞生，特别是 Visual C++ 6.0 在 Visual C++ 5.0 的基础上又加强了对网络编程的支持，Visual C++ 的功能越来越强大，使用越来越方便。

1.2.1 Visual C++ 6.0 的软硬件配置

Visual C++ 6.0 运行所需的软、硬件配置应满足以下要求：

1. Windows95 或 WindowsNT
2. Internet Explorer 4.01
3. IBM PC 及其兼容机，最好是 Pentium 75 以上的微处理器
4. 16MB 以上内存
5. 400MB 左右的可用硬盘空间

1.2.2 Microsoft 基本类库应用框架

Microsoft 基本类库和应用框架联系非常紧密，可以说应用框架就是类库的一个超集。在基本类库中，用户可以使用由软件公司销售的或是自己扩展的 C++类来实现各种功能，这使得编程人员能够借鉴和利用前人的成果并在此基础进行自己的扩展，也许这正是 MFC 类库编程倍受青睐的原因。举一个最简单的例子，如果想在窗口中创建一个可以输入文本信息的视区，可以直接从 MFC 的 CEditView 类继承你的视类，而不必像以往的 SDK 程序员那样在注册和创建窗口时指定繁杂的窗口属性以期获得所需要的窗口功能。在这里，是应用框架为你做了幕后的一切工作。因此，我们可以把应用框架定义为“在 Visual C++ 环境下的程序员和系统之间做接口的各种软组件的集合”。

1.2.3 App Studio 资源编辑器

Visual C++的编程是基于资源的编程方式。应用框架从资源中获得有关用户界面的信息。在 Visual C++ 6.0 中的资源有加速键(Accelerator)、位图(Bitmap)、对话框(Dialog)、菜单(Menu)、图标(Icon)、光标(Cursor)、串表(String Table)、工具栏(Toolbar)、版本信息 (Version Information)和主页(HTML)。

作为可视化编程环境下的资源编辑器，App Studio 对资源的编辑也是所见即所得的编辑。

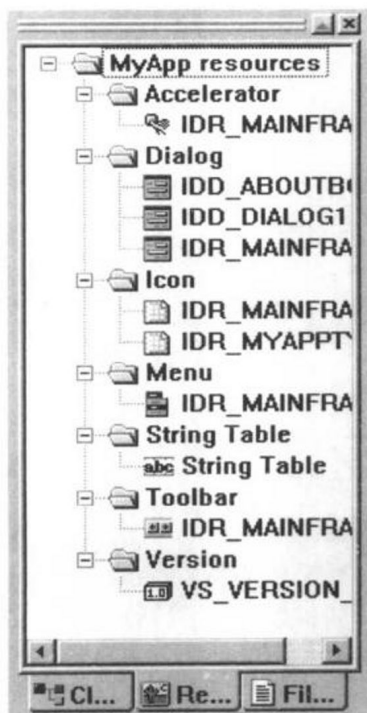


图 1.1 资源编辑器

如图 1.1 所示，在项目工作区中选中资源板面，将出现本项目中所有的资源列表，在

其中选中 **Menu** 下的菜单资源 **IDR_MAINFRAME** 并双击，就可以对此菜单进行可视化编辑了。可以说 App Studio 对资源的编辑方式充分体现了可视化编程环境的特点及优越性。

第二章 Visual C++ 集成开发环境

2.1 Visual C++ 6.0 的主窗口

从 Windows 的开始菜单启动 Visual C++ 6.0 以后,即进入开发环境,如图 2.1 所示。

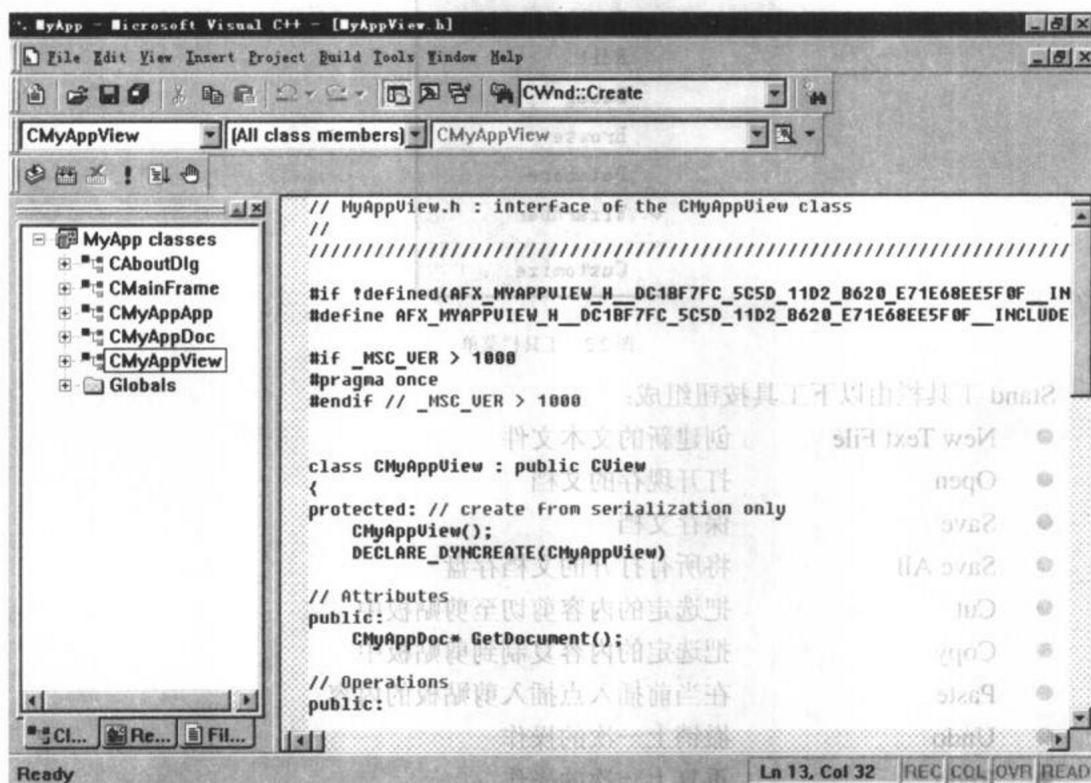


图 2.1 Visual C++ 6.0 主窗口

Visual C++ 6.0 的开发界面由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口及状态栏组成。在下面的几节中,我们将对整个开发界面做一个详细的介绍,相信通过本章的学习,读者可以熟练地掌握 Visual C++ 界面的使用方法。

2.2 Visual C++ 6.0 的工具栏

Visual C++ 6.0 的工具栏大大简化了用户的操作，并使操作过程可视化。在 Visual C++ 6.0 中工具栏有 11 种，如图 2.2 所示。缺省情况下，屏幕工作区域显示有两个工具栏，即 Stand 和 Builder MiniBar 工具栏。

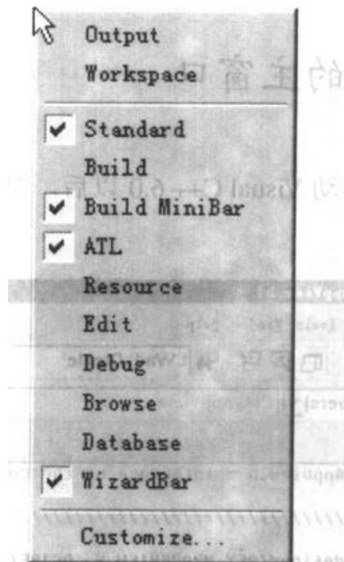


图 2.2 工具栏菜单

Stand 工具栏由以下工具按钮组成：

- New Text File 创建新的文本文件
- Open 打开现存的文档
- Save 保存文档
- Save All 将所有打开的文档存盘
- Cut 把选定的内容剪切至剪贴板中
- Copy 把选定的内容复制到剪贴板中
- Paste 在当前插入点插入剪贴板的内容
- Undo 撤销上一次的操作
- Redo 重复上一次的操作
- Workspace 显示或隐藏项目工作区
- Output 显示或隐藏输出窗口
- window List 管理当前打开的窗口
- find In Files 在文档中搜索字符串
- Find 搜索字符串
- Go Back 返回上一个主题列表
- Go Forward 前移到下一个主题列表

- Stop TopicRetrieval 停止主题搜索过程
- Refresh view 刷新 Info Viewer 窗口
- Home Page 返回 Info Viewer 主页
- Search 搜索联机文档

Build 工具栏由以下工具按钮组成:

- Select Active Project 选择活动项目
- Select Active Configuration 选择活动配置
- Compile 编译文件
- build 创建项目
- Stop Build 停止创建基础上
- execute Program 执行程序
- Go 继续执行程序
- Insert/Remove BreakPoint 插入或删除断点

在工具栏区域单击鼠标右键, 即可弹出图 2.2 所示的工具栏菜单, 用户可以从中选择或去除某些工具栏。如果不知道某一个工具栏按钮所要进行的操作, 只要将鼠标放在按钮上等待两秒钟, 系统会自动弹出有关此按钮的工具提示条。

2.3 Visual C++ 的菜单栏

菜单在 Windows 风格的应用程序中是非常常见的, 几乎所有的 Visual C++ 6.0 的操作都可以通过选择菜单项来完成。在本节中, 我们将对 Visual C++ 6.0 开发界面中的主要菜单项做一下介绍。

2.3.1 File 菜单

File 菜单下有与文件存储和打印等功能有关的菜单项。由于其中大部分菜单项均为常用应用程序中的基本菜单, 因此本节只对其中部分涉及 Visual C++ 相关功能的菜单项做以介绍。

New 菜单项

使用此菜单命令, 可以创建新的文件和项目。当用户选择此命令项或按下 Ctrl+N 时, 会弹出如图 2.3 所示的对话框:

1. 创建新的文件

在 Visual C++ 6.0 中, 利用 New 命令可以创建更多类型的文件。如果要创建新的文件, 在图所示的 New 对话框中选择 Files 标签, 可以看到 Visual C++ 6.0 可以创建的文件类型有:

- Active Server Page 创建活动服务器页

- Binary File 创建新的二进制文件
- Bitmap File 创建新的位图文件
- C/C++ Header File 创建 C/C++ 的头文件
- C++ Source File 创建 C++ 源文件
- Cursor File 创建光标文件
- HTML Page 创建 HTML 文件
- Icon File 创建图标文件
- Macro File 创建宏文件
- Resource Script File 创建资源脚本文件
- Resource Template 创建资源模板文件
- SQL Script File 创建 SQL 脚本文件
- Text File 创建文本文件

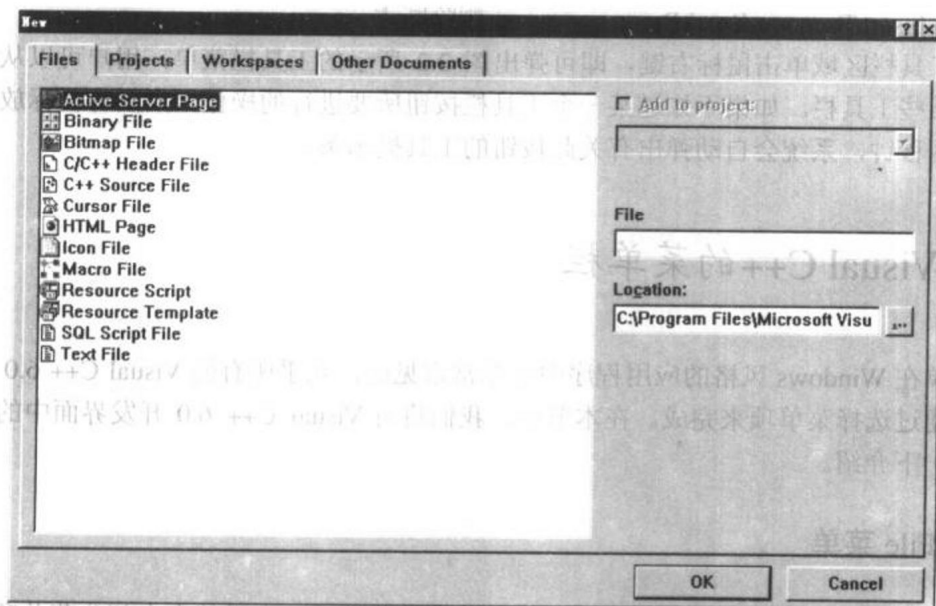


图 2.3 New 对话框

要创建新的文件，只要用鼠标点击所要选择的文件类型，在 **File** 文本框中键入所要创建文件的文件名，然后按下 **OK** 键即可。如果想要将创建的新文件加入项目中，请选择 **Add to Project** 复选框，并在其下的下拉框中选择要加入的项目名称。在 **Location** 下的文本框中可以指定新创建的文件所在的位置。可以手动键入目录名，也可以单击文本框右边的...按钮，弹出路径选择对话框(参见图 2.4)。选择新文件所在的路径，按下 **OK** 键完成选择。

2. 创建新项目

从 **New** 对话框中选择 **Project** 标签，可以看到 Visual C++ 6.0 可以创建如下类型的项目，如图 2.5 所示。

- ATL COM AppWizard 创建 Active 模板库(ATL)应用程序
- Cluster Resource Type Wizard 串资源类型向导