


第 1 章

Visual C++ 6.0 集成开发环境

( 视频讲解：27 分钟)

Visual C++ 6.0 是由 Microsoft 公司推出的可视化的开发环境，是 Windows 下最优秀的程序开发工具之一。利用 Visual C++ 6.0，可以开发出具有良好的交互功能、兼容性和扩展性的应用程序。

本章致力于使读者了解 Visual C++ 6.0 的开发环境，掌握 Visual C++ 6.0 集成开发环境中各个部分的使用，知道如何设置自己需要的开发环境，并能编写一个简单的应用程序。

通过阅读本章，您可以：

- ▶▶ 了解 Visual C++ 6.0 的特点
- ▶▶ 掌握 Visual C++ 6.0 开发环境中各个部分的使用
- ▶▶ 掌握定制 Visual C++ 6.0 开发环境的方法
- ▶▶ 掌握应用程序的创建流程

1.1 Visual C++ 6.0 概述

视频讲解：光盘\TM\lx\1\Visual C++ 6.0 概述.exe

Visual C++ 6.0 是由 Microsoft 公司推出的基于 Windows 系统的可视化集成开发环境。同其他可视化集成开发环境一样，Visual C++ 6.0 集程序的代码编辑、编译、连接和调试等功能于一体，再加上 Microsoft 公司为 Visual C++ 6.0 开发的功能强大的 MFC (Microsoft Foundation Class, 微软基础类库)，使 Visual C++ 6.0 成为开发 Windows 应用程序的最佳选择。

Visual C++ 6.0 提供了对面向对象技术的支持，利用类将与用户界面设计有关的 Windows API 函数封装起来，通过 MFC 类库的方式提供给开发人员，大大提高了程序代码的可重用性；Visual C++ 6.0 还提供了功能强大的应用程序生成向导 (AppWizard)，能够帮助用户自动生成一个应用程序框架，用户只要在该框架的适当位置添加代码就可以得到一个满意的应用程序。

1.2 Visual C++ 6.0 开发环境介绍

视频讲解：光盘\TM\lx\1\Visual C++ 6.0 开发环境介绍.exe

在使用 Visual C++ 6.0 开发应用程序之前，需要了解 Visual C++ 6.0 的集成开发环境。本节将主要介绍 Visual C++ 6.0 的集成开发环境。

1.2.1 熟悉 Visual C++ 6.0 IDE 开发环境

Visual C++ 6.0 IDE 开发环境 (Integrated Development Environment, IDE) 拥有友好的可视化界面，并且布局非常紧凑，如图 1.1 所示。

从图 1.1 中可以看出，Visual C++ 6.0 IDE 开发环境由标题栏、菜单栏、工具栏、工作区窗口、编辑窗口、输出窗口和状态栏 7 部分组成。下面介绍各部分的具体功能。

- ☑ 标题栏：显示当前项目名称和当前编辑文件的名称。
- ☑ 菜单栏：是 Visual C++ 6.0 的核心部分，所有的操作命令都可以在这里找到。默认的菜单栏相当于一个工具栏，因为它可以拖拽到开发环境的任意位置。
- ☑ 工具栏：通常包括一些常用的工具按钮。除了在图 1.1 中可以看到 Standard、WizardBar 和 Bulid MinBar 外，Visual C++ 6.0 还提供了 Edit、Debug 等 11 个工具栏。右击工具栏，可以弹出相关的工具栏快捷菜单。
- ☑ 工作区窗口：该窗口包括类视图(ClassView)、资源视图(ResourceView)和文件视图(FileView) 3 个选项卡。
- ☑ 编辑窗口：用于显示当前编辑的 C++ 程序文件及资源文件。
- ☑ 输出窗口：当编译、链接程序时，输出窗口会显示编译和链接的信息。如果进入程序调试状

第 1 章 Visual C++ 6.0 集成开发环境

态，主窗口还将弹出一些调试窗口。

- ☑ 状态栏：用于显示当前的操作状态或所选择命令的提示信息等。

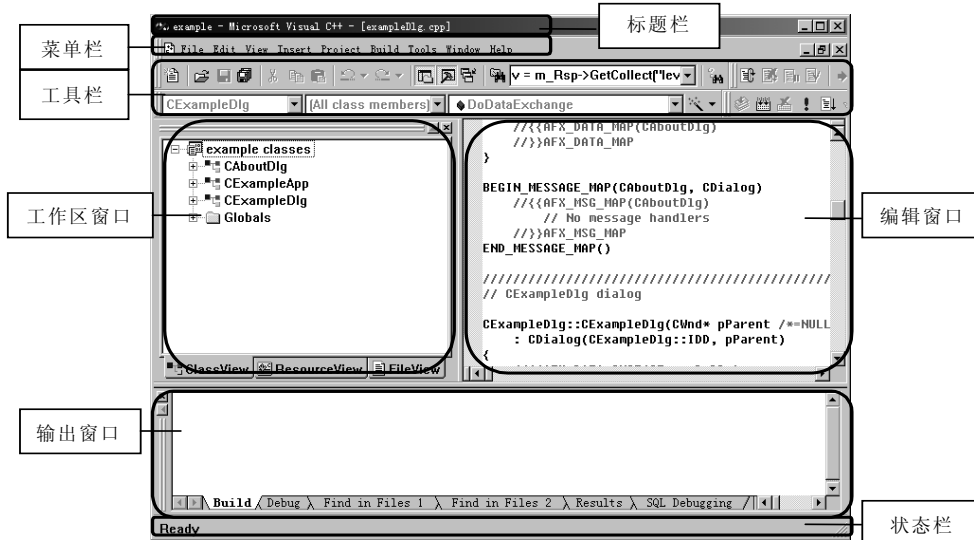


图 1.1 Visual C++ 6.0 IDE 开发环境

1.2.2 Visual C++ 6.0 IDE 菜单介绍

Visual C++ 6.0 IDE 集成开发环境的菜单栏中包括 File、Edit、View、Insert、Project、Build、Tools、Window 和 Help 9 个菜单，下面对这些菜单进行介绍。

1. File 菜单

File 菜单中包含用于对文件进行各种操作的命令，如图 1.2 所示。

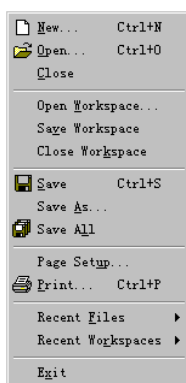


图 1.2 File 菜单

File 菜单中各命令的功能说明如表 1.1 所示。

Visual C++从入门到精通（第2版）

表 1.1 File 菜单中各命令的功能说明

命 令	功 能 说 明
New	创建新的文件、工程和工作区
Open	打开一个已存在的文件、工程和工作区
Close	关闭当前打开的文件
Open Workspace	打开一个已存在的工作区（Workspace）
Save Workspace	保存当前打开的工作区（Workspace）
Close Workspace	关闭当前打开的工作区（Workspace）
Save	保存当前打开的文件
Save As	将当前文件另存为一个新文件名
Save All	保存所有打开的文件
Page Setup	为打印文件的页面进行设置（打印机安装后有效）
Print	打印文件的全部或选定的部分（打印机安装后有效）
Recent Files	最近打开的文件列表，用户可以查看或重新打开
Recent Workspaces	最近使用的工作区（Workspace），用户可以查看或重新打开
Exit	退出开发环境



技巧

使用 Open 和 Open Workspace 命令都可以打开工程，但是使用 Open 命令打开工程时包括所有的文件类型，没有直接使用 Open Workspace 命令方便。

2. Edit 菜单

Edit 菜单中包含所有与文件编辑有关的命令（如复制、粘贴等操作），如图 1.3 所示。

Edit 菜单中各命令的功能说明如表 1.2 所示。

表 1.2 Edit 菜单中各命令的功能说明

命 令	功 能 说 明
Undo	撤销上一次的编辑操作。即使保存了文件，这个操作仍然有效
Redo	恢复被取消的编辑操作
Cut	将所选择的内容剪切掉，移到剪贴板中
Copy	将所选内容复制到剪贴板中
Paste	在当前位置插入剪贴板中最新一次的内容
Delete	删除被选择的内容
Select All	选择当前窗口中的全部内容
Find	查找指定的字符串
Find in Files	在多个文件中查找指定字符串
Replace	替换指定字符串
Go To	可将光标移到指定的位置
Bookmarks	设置书签或书签导航，方便以后查找

第 1 章 Visual C++ 6.0 集成开发环境

续表

命 令	功 能 说 明
Incremental Search	开始向前搜索
Format Selection	对选择对象进行快速缩排
Untabify Selection	在选择对象中用空格代替跳格
Tabify Selection	在选择对象中用跳格代替空格
Make Selection Uppercase	把选择部分改成大写
Make Selection Lowercase	把选择部分改成小写
a-b View Whitespace	显示或隐藏空格点
Breakpoints	编辑程序中的断点
List Members	显示出全部关键字
Type Info	显示变量、函数或方法的语法
Parameter Info	显示函数的参数
Complete Word	给出相关关键字的全称

3. View 菜单

View 菜单用来改变窗口的显示方式，如图 1.4 所示。

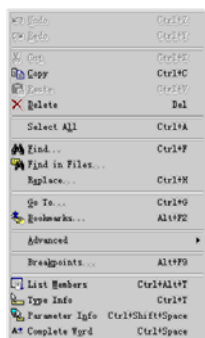


图 1.3 Edit 菜单

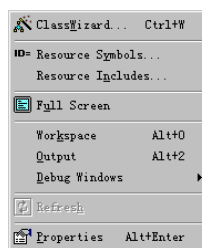


图 1.4 View 菜单

View 菜单中各命令的功能说明如表 1.3 所示。

表 1.3 View 菜单中各命令的功能说明

命 令	功 能 说 明
ClassWizard	打开类向导，用于编辑应用程序的类
Resource Symbols	浏览和编辑资源文件中的符号
Resource Includes	编辑修改资源文件名及预处理指令
Full Screen	在窗口的全屏幕方式和正常方式之间进行切换
Workspace	激活工作区窗口
Output	激活输出窗口
Debug Windows	激活调试窗口
Refresh	刷新选中区域
Properties	激活属性页窗口

Visual C++从入门到精通 (第2版)



只有在调试状态下才能通过 Debug Windows 命令激活相应的调试窗口。

4. Insert 菜单

Insert 菜单用于进行向当前工程中插入类、资源和插入文件等操作，如图 1.5 所示。
Insert 菜单中各命令的功能说明如表 1.4 所示。

表 1.4 Insert 菜单中各命令的功能说明

命 令	功 能 说 明
New Class	在工程中添加新类
New Form	在工程中添加新表单
Resource	创建各种新资源
Resource Copy	对选中的资源进行复制
File As Text	在当前源文件中插入一个文件
New ATL Object	在工程中添加一个新的 ATL 对象

5. Project 菜单

Project 菜单用于管理项目和工作区，如图 1.6 所示。

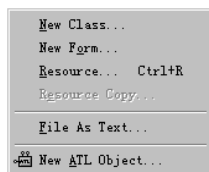


图 1.5 Insert 菜单

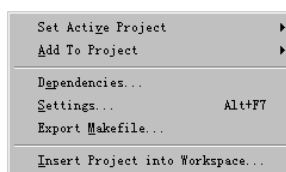


图 1.6 Project 菜单

Project 菜单中各命令的功能说明如表 1.5 所示。

表 1.5 Project 菜单中各命令的功能说明

命 令	功 能 说 明
Set Active Project	选择指定的项目为工作区中的活动项目
Add To Project	用于添加文件、文件夹、数据链接和部件
Dependencies	编辑工程组件
Settings	对工程进行编译及调试的设置
Export Makefile	以 Makefile 形式输出可编译项目
Insert Project into Workspace	将已存在的工程插入到工作区窗口中

6. Build 菜单

Build 菜单中包含的命令用于编译、创建、调试及执行应用程序，如图 1.7 所示。
Build 菜单中各命令的功能说明如表 1.6 所示。

第 1 章 Visual C++ 6.0 集成开发环境

表 1.6 Build 菜单中各命令的功能说明

命 令		功 能 说 明
Compile		用于编译当前源代码编辑窗口中的源文件
Build		用于生成一个工程，即编译、连接当前工程中所包含的所有文件
Rebuild All		编译和连接工程及资源
Batch Build		一次编译和连接多个工程
Clean		用于删除当前项目中所有中间文件及输出文件
Start Debug	Go	开始或继续调试程序
	Step Into	单步运行调试
	Run to Cursor	运行程序到光标所在行
	Attach to Process	连接正在运行的进程
Debugger Remote Connection		用于编辑远程调试连接设置
Execute		运行程序
Set Active Configuration		选择激活的工程及配置
Configurations		编辑工程的配置
Profile		选择该命令，可以检查代码的执行情况



在调试状态下，Build 菜单会被替换成 Debug 菜单。

7. Tools 菜单

Tools 菜单用于选择或定制集成开发环境中的一些实用工具，如图 1.8 所示。



图 1.7 Build 菜单



图 1.8 Tools 菜单

Tools 菜单中各命令的功能说明如表 1.7 所示。

表 1.7 Tools 菜单中各命令的功能说明

命 令	功 能 说 明
Source Browser	启动源代码浏览器
Close Source Browser File	关闭打开的浏览信息数据库

Visual C++从入门到精通 (第2版)

续表

命 令	功 能 说 明
Visual Component Manager	激活可视化组件管理器
Register Control	启动寄存器控制器
Error Lookup	启动错误查找器
ActiveX Control Test Container	启动 ActiveX 控件测试器
OLE/COM Object Viewer	启动 OLE/COM 对象查看器
Spy++	启动 Spy++ 工具包
MFC Tracer	启动 MFC 跟踪器
InstallShield Wizard	为应用程序创建安装项目
Customize	定制 Tools 菜单和工具栏
Options	可以对集成开发环境的各项进行设置
Macro	创建和编辑宏
Record Quick Macro	记录宏
Play Quick Macro	运行宏

8. Window 菜单

Window 菜单用于进行窗口布局的调整、窗口间的跳转、打开和关闭等操作，如图 1.9 所示。

Window 菜单中各命令的功能说明如表 1.8 所示。

表 1.8 Window 菜单中各命令的功能说明

命 令	功 能 说 明
New Window	为当前文档打开另一窗口
Split	将窗口拆分为多个窗口
Docking View	启动或关闭 Docking View 模式
Close	关闭当前窗口
Close All	关闭所有打开的窗口
Next	激活下一个未停放的窗口
Previous	激活上一个未停放的窗口
Cascade	将所有打开的窗口重叠地排列
Tile Horizontally	将工作区中所有打开的窗口纵向平铺
Tile Vertically	将工作区中所有打开窗口横向平铺
Windows	管理当前打开的窗口

9. Help 菜单

Help 菜单为 Visual C++ 6.0 提供了大量详细的帮助信息，如图 1.10 所示。

Help 菜单中各命令的功能说明如表 1.9 所示。

第 1 章 Visual C++ 6.0 集成开发环境

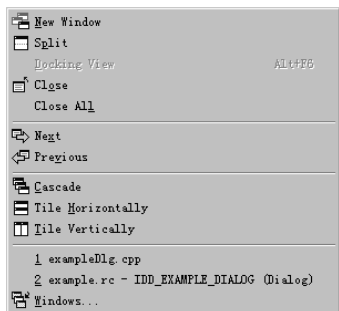


图 1.9 Window 菜单

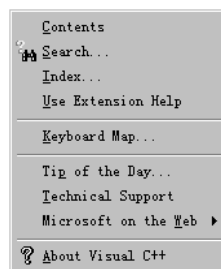


图 1.10 Help 菜单

表 1.9 Help 菜单中各命令的功能说明

命 令	功 能 说 明
Contents	显示所有帮助信息的内容列表
Search	利用在线查询获得帮助信息
Index	显示在线文件的索引
Use Extension Help	开启或关闭 Extension Help 工具
Keyboard Map	显示所有键盘命令
Tip of the Day	显示 Tip of the Day 工具
Technical Support	显示 Visual Studio 的支持信息
Microsoft on the Web	显示 Microsoft 产品页
About Visual C++	显示版本的有关信息



说明

在用户编辑对话框资源时，还会出现 Layout 菜单，该菜单中的命令主要用于对控件大小和位置的操作，在后面章节的应用中会进行介绍。

1.2.3 Visual C++ 6.0 IDE 工具栏介绍

工具栏是一种图形化的操作界面，与菜单栏一样也是开发环境的重要组成部分。工具栏中主要列出了在开发过程中经常使用的一些功能，具有直观和快捷的特点，熟练使用这些工具按钮将大大提高工作效率。在 Visual C++ 6.0 开发环境中包括 11 个标准工具栏，这些工具栏并不都显示在开发环境中，可以在工具栏上任意位置单击鼠标右键，在弹出的快捷菜单中选择要显示的工具栏，如图 1.11 所示。

下面介绍 3 个常用的工具栏。

1. Standard 工具栏

Standard 工具栏用于维护和编辑工作区的文本和文件，如图 1.12 所示。

Standard 工具栏中各命令按钮的功能说明如表 1.10 所示。

Visual C++从入门到精通 (第2版)

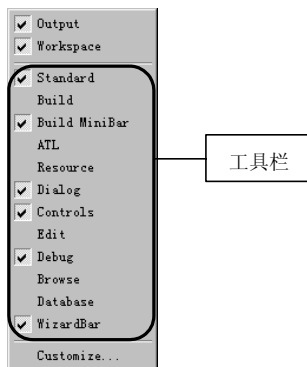


图 1.11 工具栏快捷菜单



图 1.12 Standard 工具栏

表 1.10 Standard 工具栏中各命令按钮的功能说明

命令按钮	功能说明
	创建一个新的文件、项目和工作区
	打开一个已存在的文件、项目和工作区
	保存当前打开的文件
	保存所有打开的文件
	将所选择的内容剪切掉，移到剪贴板中
	将所选的内容复制到剪贴板中
	在当前位置粘贴剪贴板中最新的内容
	取消上一次的操作
	恢复被取消的操作
	激活工作区窗口，用来管理工程中的文件和资源
	激活输出窗口，用来显示编译、调试和查找的信息
	管理当前打开的窗口
	在所有窗口中查找指定字符串
	在当前窗口查找指定的字符串
	调用 MSDN

技巧

通过 按钮可以调用 MSDN，但是搜索时需要输入要查询的内容。还有一种更简单的方法可以达到同样的效果，即在代码编辑器中选择要查询的内容，然后按 F1 键，即可快速调用 MSDN 进行搜索。

2. Build MiniBar 工具栏

Build MiniBar 工具栏用于运行程序和调试程序，如图 1.13 所示。

第 1 章 Visual C++ 6.0 集成开发环境



图 1.13 Build MiniBar 工具栏

Build MiniBar 工具栏中各命令按钮的功能说明如表 1.11 所示。

表 1.11 Build MiniBar 工具栏中各命令按钮的功能说明

命令按钮	功能说明
	用于编译当前在源代码编辑窗口的源文件
	用于编译、连接当前工程中的文件，生成一个可执行文件
	终止编译或连接的程序
	运行程序
	开始或继续调试程序
	编辑程序中的断点

3. Debug 工具栏

Debug 工具栏用于调试程序，如图 1.14 所示。



图 1.14 Debug 工具栏



Debug 工具栏中各命令按钮的功能说明如表 1.12 所示。

表 1.12 Debug 工具栏中各命令按钮的功能说明

命令按钮	功能说明
	放弃当前的调试，重新开始调试
	终止调试，返回到编辑状态
	暂停调试
	改变代码后调试
	显示将要运行的代码行
	单步执行程序，进入到函数内部
	单步执行程序，不进入到函数内部
	跳出当前函数
	运行到光标
	弹出 QuickWatch 窗口
	显示/隐藏 Watch 窗口
	显示/隐藏 Variables 窗口
	显示/隐藏 Registers 窗口
	显示/隐藏 Memory 窗口

Visual C++从入门到精通 (第2版)

续表

命令按钮	功能说明
	显示/隐藏 Call Stack 窗口
	显示/隐藏 Disassembly 窗口

1.2.4 Visual C++ 6.0 工作区窗口介绍








Visual C++ 6.0 工作区窗口包括 ClassView (类视图)、ResourceView (资源视图) 和 FileView (文件视图) 3 个选项卡, 下面分别进行介绍。

1. ClassView 选项卡

ClassView 选项卡用来显示当前工作区中所有的类、结构和全局变量, 如图 1.15 所示。

ClassView 选项卡中提供了工程中所有类的层次列表, 通过展开各个节点可以显示类中包含的细节。在层次列表的每个项目前面都有一个图标, 每个项目对应的图标含义如表 1.13 所示。

表 1.13 项目对应的图标含义

图 标	含 义
	类
	保护类成员函数
	私有类成员函数
	公有类成员函数
	保护类成员变量
	私有类成员变量
	公有类成员变量

当用户双击类或其成员的图标时, 光标自动定位到类或其成员定义的起始位置。在任意类名上单击鼠标右键, 将弹出一个快捷菜单, 其中各命令的功能如图 1.16 所示。

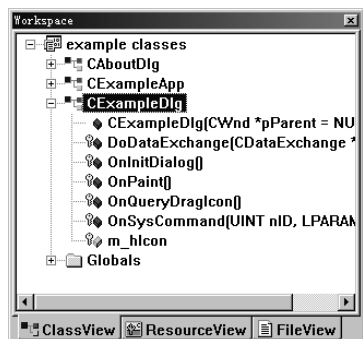


图 1.15 ClassView 选项卡

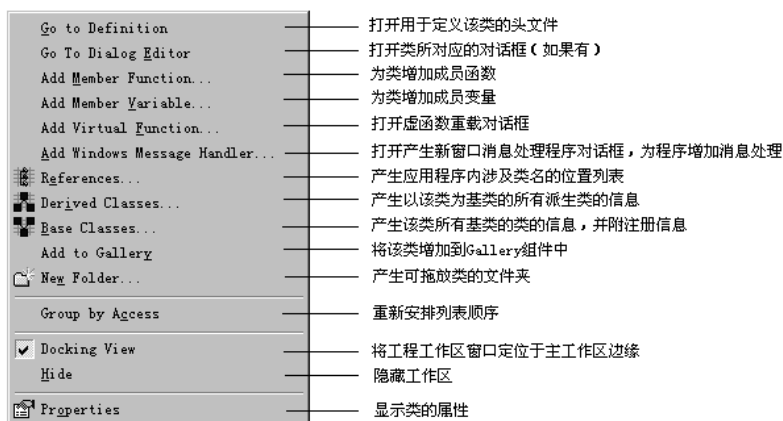


图 1.16 快捷菜单

第 1 章 Visual C++ 6.0 集成开发环境



注意

如果用户在头文件中手动添加函数声明，而没有设置函数的实现代码时，在当前类节点下也会显示函数名，但是双击时无法跳转到指定位置。

2. ResourceView 选项卡

ResourceView 选项卡在层次列表中列出了工程中用到的资源，图标、位图等都可以作为资源使用，如图 1.17 所示。

在 ResourceView 选项卡中的节点上单击鼠标右键，在弹出的快捷菜单中选择 Insert 命令，将弹出 Insert Resource 对话框，如图 1.18 所示。

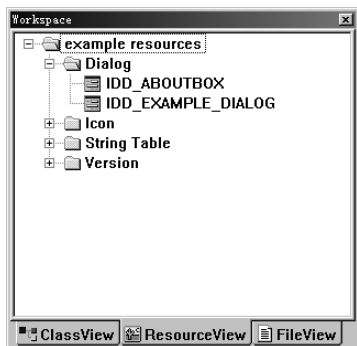


图 1.17 ResourceView 选项卡

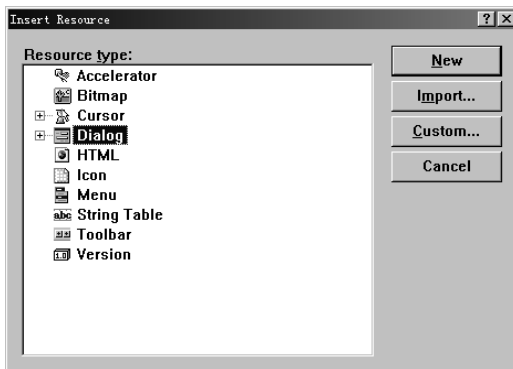


图 1.18 Insert Resource 对话框

在 Insert Resource 对话框中，可以根据需要对资源进行操作。其中，New 按钮用于新建资源，Import 按钮用于导入资源，Custom 按钮用于定制资源，Cancel 按钮用于退出对话框。

3. FileView 选项卡

FileView 选项卡与 ClassView 选项卡非常相似，可以显示和编辑源文件和头文件，如图 1.19 所示。通过 FileView 选项卡更容易进入类定义的文件，使打开资源文件和非代码文件更加简单。

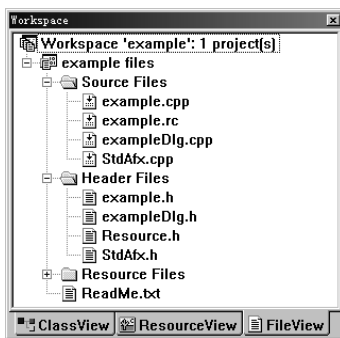


图 1.19 FileView 选项卡

在 FileView 选项卡中双击某一文件，将在开发环境的编辑窗口中打开对应的文件窗口。

1.2.5 Visual C++ 6.0 控件面板介绍

控件是一个独立的程序模块，用户可以利用对话框编辑器通过交互操作来创建，然后通过控件的ID与程序相连，并进行调用。另外，用户也可以直接编写代码创建控件，但是需要编写大量的代码。使用控件不仅能使界面美观、标准，还可以大大减少编程的工作量。在 Visual C++ 6.0 的工具栏空白处单击鼠标右键，在弹出的快捷菜单中选择 Controls 命令将显示控件面板，如图 1.20 所示。

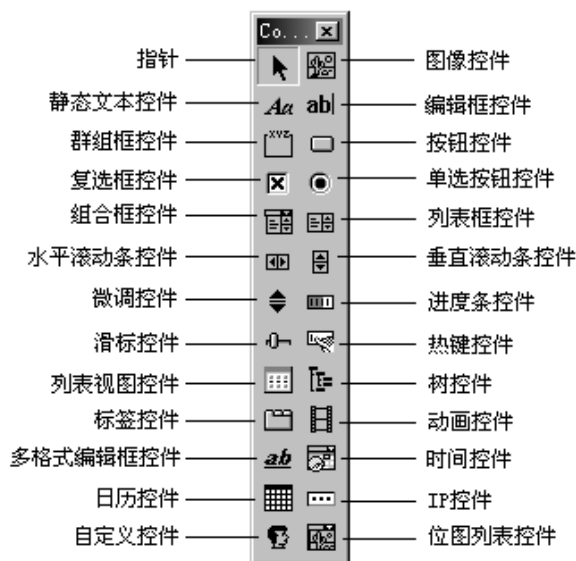


图 1.20 控件面板

1.2.6 Visual C++ 6.0 输出窗口介绍

Visual C++ 6.0 输出窗口位于开发环境的下部，在执行编译、连接和调试等操作时将显示相关的信息，如图 1.21 所示。

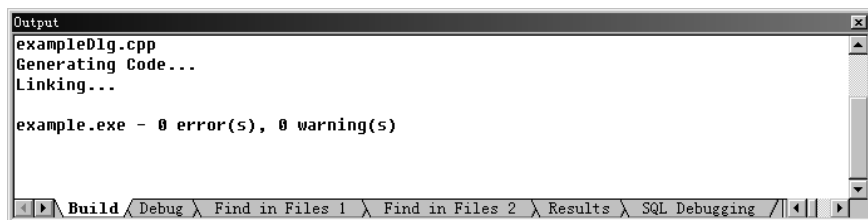


图 1.21 输出窗口

在输出窗口中，数据根据不同的操作显示在不同的选项卡中。各选项卡的功能如表 1.14 所示。

第 1 章 Visual C++ 6.0 集成开发环境

表 1.14 输出窗口中各选项卡的功能

选项卡	功能
Build	显示编译和连接结果
Debug	显示调试信息
Find in Files 1	显示在文件查找中得到的结果
Find in Files 2	显示在文件查找中得到的结果
Results	显示结果
SQL Debugging	显示 SQL 调试信息



说明

用户在进行编译、调试、查找等操作时，输出窗口会根据操作自动选择相应的选项卡进行显示，如果用户在编译过程中出现错误，只要双击错误信息，代码编辑器就会跳转到相应的错误代码处。

1.3 设置自己的开发环境

 视频讲解：光盘\TM\lx\1\设置自己的开发环境.exe

1.3.1 自定义工具栏

Visual C++ 6.0 为用户提供了 11 个预定的工具栏，此外用户还可以根据需要自己定义工具栏。自定义工具栏的步骤如下：

(1) 在 Visual C++ 6.0 开发环境中选择 Tools/Customize 命令，打开 Customize 对话框，选择 Toolbars 选项卡，如图 1.22 所示。

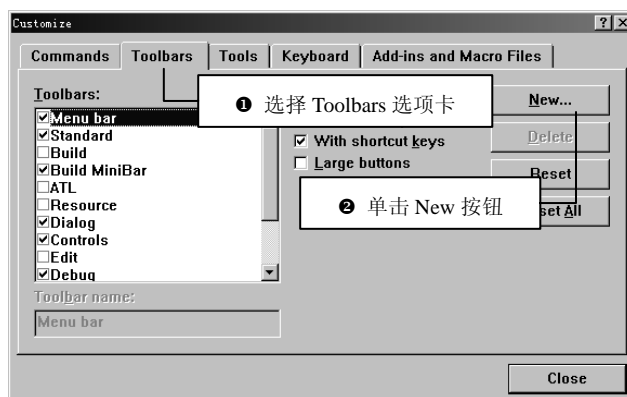


图 1.22 Customize 对话框

Visual C++从入门到精通 (第2版)

(2) 单击 New 按钮, 弹出 New Toolbar 对话框, 在 Toolbar name 编辑框中输入工具栏名称, 如图 1.23 所示。

(3) 单击 OK 按钮, 创建一个工具栏, 新创建的工具栏名称为“工具栏”, 如图 1.24 所示。

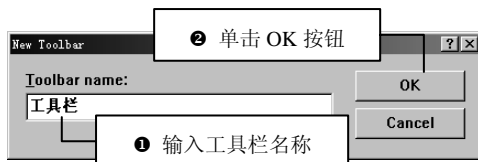


图 1.23 New Toolbar 对话框



图 1.24 新建工具栏

(4) 在 Customize 对话框中选择 Commands 选项卡, 在 Category 组合框中选择一个目录, 如图 1.25 所示。

(5) 在 Buttons 栏中会显示相应的按钮图标, 利用鼠标将其中的按钮拖动到新建的工具栏窗口中。根据需要在不同的目录中选择工具栏按钮, 将这些按钮都拖动到工具栏窗口以后, 单击 Close 按钮, 就完成了新工具栏的创建。新创建的工具栏如图 1.26 所示。

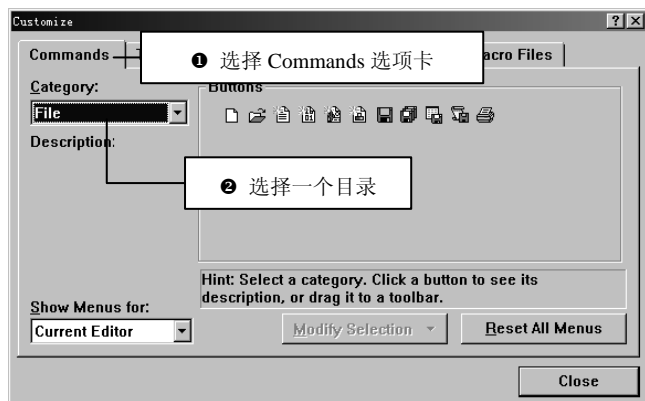


图 1.25 Commands 选项卡



图 1.26 新创建的工具栏



说明

如果用户想要删除自己创建的工具栏, 可以在 Customize 对话框中选择 Toolbars 选项卡, 在工具栏列表框中选择要删除的工具栏, 然后单击右侧的 Delete 按钮进行删除。

1.3.2 自定义代码编辑窗口

在 Tools 菜单中包含了許多编辑选项, 合理地设置这些选项可以提高程序的编译速度, 使程序代码更易于阅读和理解, 程序开发更加得心应手。用户可以设置代码编辑器中字体的大小、颜色等信息, 其中最主要也是开发人员经常设置的是数字、字符串和注释的颜色。下面就以设置这些信息为例来介绍如何自定义代码编辑窗口。步骤如下:

第 1 章 Visual C++ 6.0 集成开发环境

(1) 在 Visual C++ 6.0 开发环境中选择 Tools/Options 命令，打开 Options 对话框，选择 Format 选项卡，如图 1.27 所示。

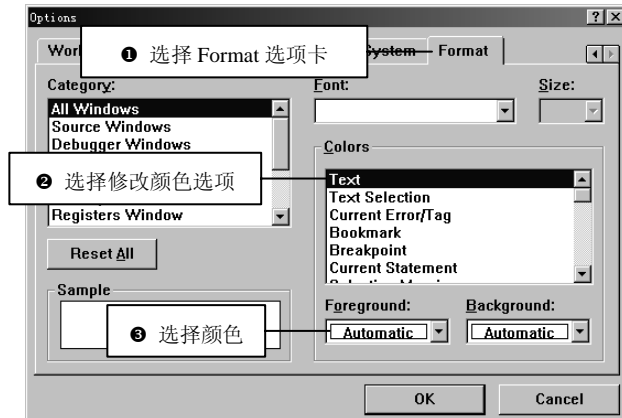


图 1.27 Options 对话框

(2) 在 Category 列表框中选择 Source Windows 选项；在 Colors 列表框中选择 Comment 选项，表示设置注释的颜色。在 Foreground 组合框中设置注释的字体颜色，用户可以选择自己喜欢的颜色，本例选择绿色。另外，还可以为注释设置背景色，方法是在 Background 组合框中选择一种颜色，但是通常情况下不要设置背景色，否则代码编辑器会显得很零乱。

(3) 在 Colors 列表框中选择 Number 选项，表示设置数字的颜色。同样，在 Foreground 组合框中设置数字的颜色，本例选择蓝色。

(4) 在 Colors 列表框中选择 String 选项，表示设置字符串的颜色。在 Foreground 组合框中为字符串选择一种颜色，本例选择红色。

(5) 单击 OK 按钮完成设置，如图 1.28 所示。

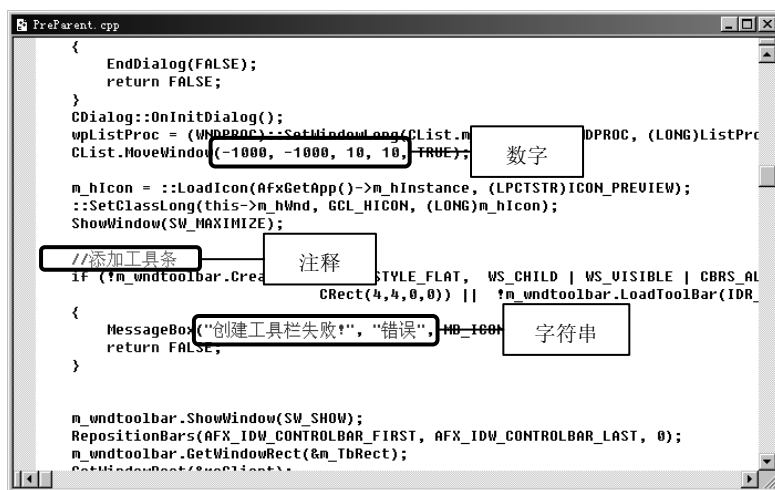



图 1.28 代码编辑窗口

1.4 创建一个简单的程序

 视频讲解：光盘\TM\lx\1\创建一个简单的程序.exe

【例 1.1】 一个简单的 MFC 应用程序。（实例位置：光盘\TM\sl\1\1）

通过前面的学习，读者已经对 Visual C++ 6.0 的开发环境有了大致的了解。下面来制作一个简单的 MFC 应用程序，希望通过这个程序使读者了解开发应用程序的一般过程。

1.4.1 创建工程

(1) 在 Visual C++ 6.0 开发环境中选择 File/New 命令，弹出 New 对话框。选择 Projects 选项卡，选择 MFC AppWizard[exe] (MFC 应用程序向导) 选项，在 Project name 编辑框中输入创建的工程名“Hello”，在 Location 编辑框中设置工程文件存放的位置为“D:\Hello”，如图 1.29 所示。

(2) 单击 OK 按钮，弹出 MFC AppWizard-Step 1 窗口，如图 1.30 所示。

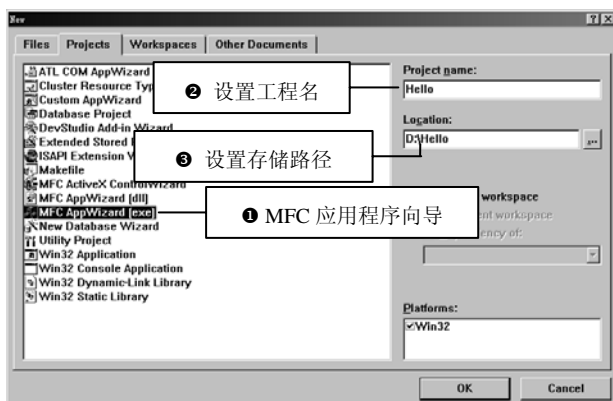


图 1.29 New 对话框



图 1.30 MFC AppWizard-Step 1 窗口

在 MFC AppWizard-Step 1 窗口中可以指定生成框架的类型。

- Single document: 生成单文档应用程序框架。
- Multiple documents: 生成多文档应用程序框架。
- Dialog based: 生成基于对话框的应用程序框架。
- Document/View architecture support: 选中该复选框，允许生成文档/视图和非文档/视图结构程序。

(3) 本例选中 Dialog based 单选按钮，创建一个生成基于对话框的应用程序。单击 Next 按钮，弹出 MFC AppWizard-Step 2 of 4 窗口，如图 1.31 所示。

MFC AppWizard-Step 2 of 4 窗口中各选项介绍如下。

第 1 章 Visual C++ 6.0 集成开发环境

- About box: 生成“关于”对话框。
- Context-sensitive Help: 生成支持上下文相关帮助的帮助文件。
- 3D controls: 具有 3D 效果的程序界面。
- Automation: 应用程序能够操作在其他应用程序中实现的对象，或者自己的应用程序可供 Automation 客户使用。
- ActiveX Controls: 支持 ActiveX 控件。
- Windows Sockets: 支持基于 TCP/IP 协议的网络通信。
- Please enter a title for your dialog: 设置应用程序主窗口的标题。



说明

如果用户没有其他的设置，则可以直接单击 Finish 按钮完成创建。

(4) 单击 Next 按钮，弹出 MFC AppWizard-Step 3 of 4 窗口，如图 1.32 所示。

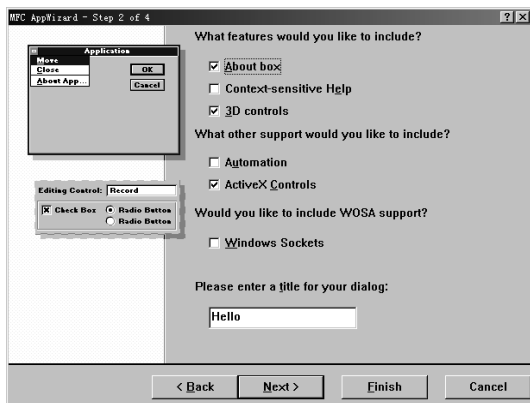


图 1.31 MFC AppWizard-Step 2 of 4 窗口

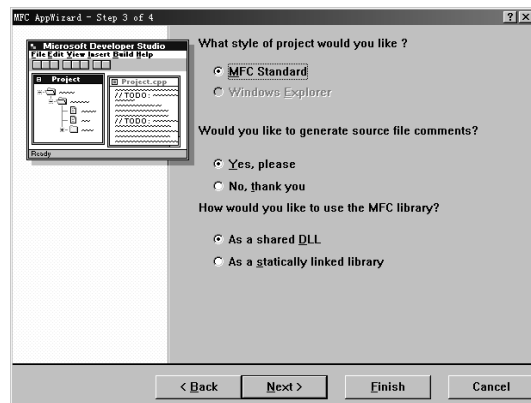


图 1.32 MFC AppWizard-Step 3 of 4 窗口

在 MFC AppWizard-Step 3 of 4 窗口中各选项介绍如下。

- MFC Standard: 标准 MFC 项目。
- Windows Explorer: “Windows 资源管理器”风格项目。
- Yes, please: 在源文件中添加注释。
- No, thank you: 不添加注释。
- As a shared DLL: 共享动态链接库。
- As a statically linked library: 静态链接库。

(5) 单击 Next 按钮，弹出 MFC AppWizard-Step 4 of 4 窗口，如图 1.33 所示。

(6) 在 MFC AppWizard-Step 4 of 4 窗口中确定类的名称及所在文件的名称。单击 Finish 按钮，弹出 New Project Information 窗口，如图 1.34 所示。

(7) 在 New Project Information 窗口中显示了将要创建工程的文件清单，单击 OK 按钮完成工程的创建。创建的工程如图 1.35 所示。

Visual C++从入门到精通 (第2版)

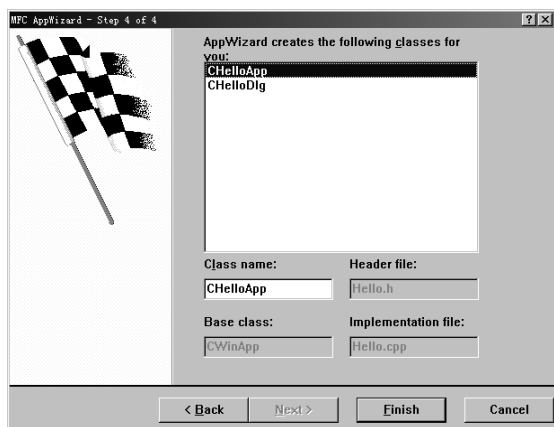


图 1.33 MFC AppWizard-Step 4 of 4 窗口

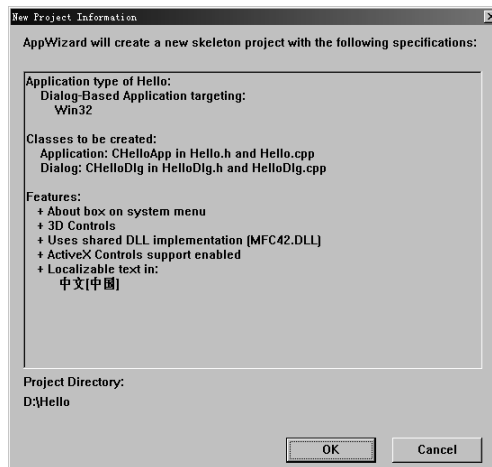


图 1.34 New Project Information 窗口

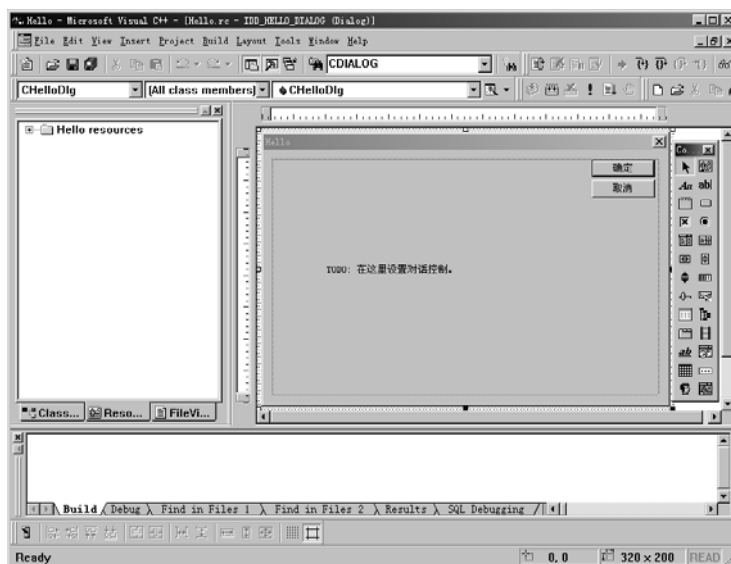


图 1.35 创建的工程

1.4.2 编辑程序

工程创建后会自动创建一个对话框，其中有两个“按钮”控件和一个“静态正文”控件，这是程序自动生成的控件。接下来实现新工程的编辑。步骤如下：

- (1) 调整控件的位置，并修改对话框的大小，如图 1.36 所示。
- (2) 双击“确定”按钮，为“确定”按钮处理单击事件，在该事件中为“静态正文”控件设置显示的文本和字体大小。代码如下：

```
void CHelloDlg::OnOK()
{
```

第 1 章 Visual C++ 6.0 集成开发环境

```
CFont font; //声明字体对象
font.CreatePointFont(200,"宋体"); //创建字体
GetDlgItem(IDC_STATIC)->SetWindowText("Hello Visual C++"); //设置静态文本控件的显示文本
GetDlgItem(IDC_STATIC)->SetFont(&font); //设置静态文本控件的显示字体
//CDialog::OnOK(); //注释掉程序生成的退出代码
}
```

(3) 按 F7 键编译程序，按 F5 键执行程序，单击“确定”按钮。程序运行结果如图 1.37 所示。

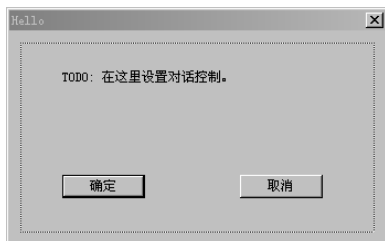


图 1.36 对话框



图 1.37 程序运行结果

1.5 小 结

本章主要讲解 Visual C++ 6.0 集成开发环境，对开发环境中的各个部分作了详细讲解，从中读者可以初步了解 Visual C++ 6.0。最后通过创建一个简单的应用程序，向读者演示使用 Visual C++ 6.0 开发应用程序的整个过程。俗话说“工欲善其事，必先利其器”，只有熟练掌握了开发环境这支利器，才能在以后的程序开发中事半功倍。