



Visual C++

程序员基础教程

丁有和 / 编著



12C-43

青 岛 出 版 社

TP312C-43
058

Visual C⁺⁺程序员基础教程

丁有和 编著

青 岛 出 版 社

鲁新登字 08 号

Visual C++程序员基础教程/丁有和编著 . - 青岛: 青岛出版社, 1999.10

ISBN 7-5436-2024-3

I. V...

II. 丁...

III. C 语言-程序设计-教材

IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 38224 号

书 名	Visual C++程序员基础教程
编 著 者	丁有和
出版发行	青岛出版社
社 址	青岛市徐州路 77 号(266071)
邮购电话	(0532)5835844 5814750 5814611-20
责任编辑	樊建修 金利鹏
装帧设计	申 尧
印 刷	胶州市印刷厂
出版日期	1999 年 10 月第 1 版, 1999 年 10 月第 1 次印刷
开 本	16 开(787×1092 毫米)
印 张	15.75
字 数	355 千
印 数	1-5000
ISBN	7-5436-2024-3/TP·231
定 价	26.00 元

第一章 快速入门

第一节 快速浏览 Developer Studio

Visual C++ 6.0 是 Microsoft 最新版本的 C++ 编译器，它是 Microsoft 的 Windows 程序开发工具的延续。Visual C++ 软件包除有一个编译器外，它还包括所有相关的库、例子和为 Windows 创建应用程序所需要的文档。Visual C++ 和 Developer Studio 构成了一个完整的集成开发环境，它使创建一个 Windows 程序变得很简单；通过使用 Developer Studio 中的工具、向导以及 MFC 类库，你能在很短时间内创建一个程序。

如何启动 VC 呢？从 Windows 系统中单击任务栏的“开始”，选择“程序”中的“Microsoft Visual Studio 6.0”，再选择“Microsoft Visual C++ 6.0”便可运行。第一次运行将显示 Tip of the Day 对话框，单击[Next Tip]按钮，则可以看到有关各种操作的提示；如果不选中 Show tips at startup 开关项，则以后运行 VC 就再不会出现此窗口，如图 1.1 所示。

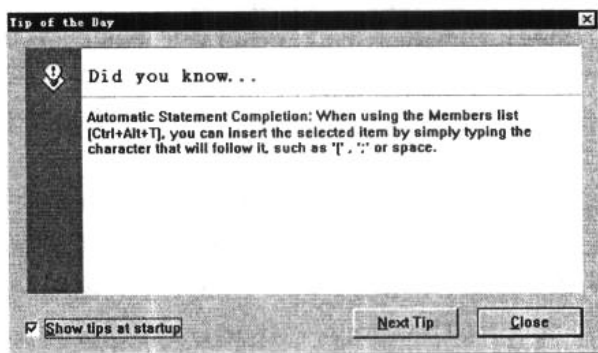


图 1.1 Tip of the Day 对话框

单击[Close]关闭此提示窗口，让我们进入正式的开发环境，快速浏览一下。

一、Developer Studio 窗口

Developer Studio 界面有三个窗口：

- 一个在左边的项目区窗口：这个窗口包含关于正在开发的这个项目的信息。
- 一个在右边的文档窗口：这个窗口将包含源文件和组成项目的项目资源。
- 一个在下边的输出窗口：这个窗口将输出程序编译的过程、错误以及其他信息。

另外，Developer Studio 也包含大量的菜单、工具栏和其他用户接口，如图 1.2 所示。

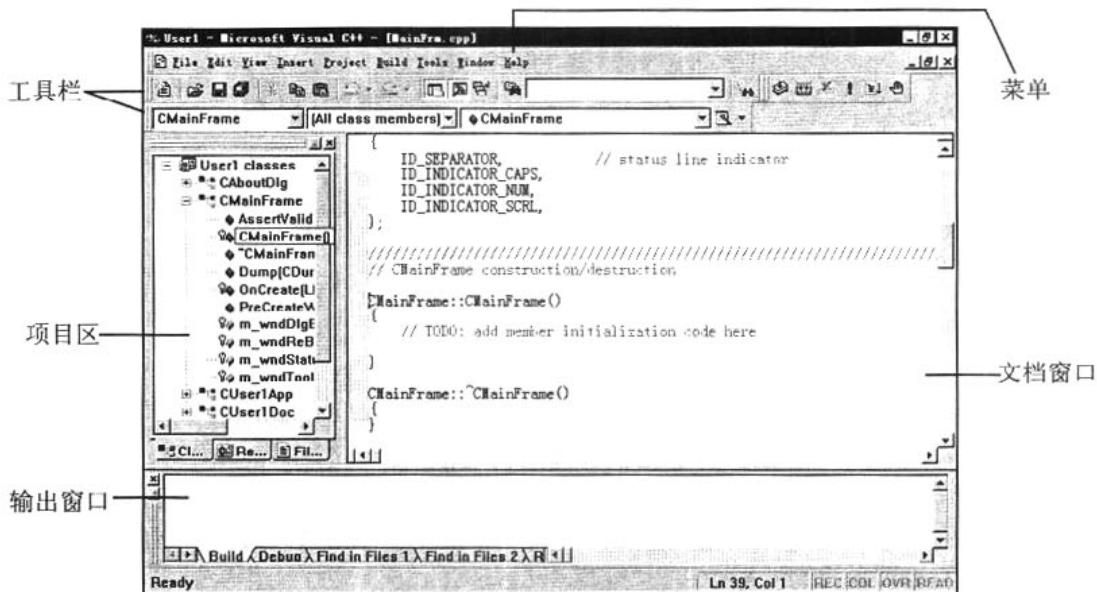


图 1.2 Developer Studio 环境

需要说明的是项目区(Workspace)窗口是一个浮动(docking)窗口，它既可以停靠在窗口的任一边上，也可浮动于其他窗口之上。项目区窗口由三栏组成：ClassView、ResourceView、FileView；它们从不同角度反映应用工程的内容。

若要显示或隐藏项目区窗口，可单击菜单项“View→Workspace”或在键盘上按下快捷键(缺省时为 Alt+0)，或单击相应工具栏按钮()。

若要显示或隐藏输出窗口，可单击菜单项“View→Output”或在键盘上按下快捷键(缺省时为 Alt+2)，或单击相应工具栏按钮()。

二、ClassView

ClassView 栏用以展示应用程序中所有类的信息。假设打开的工程名为 User1，单击项目区窗口底部的 ClassView，窗口中则显示出一个标题“User1 classes”，在它的前面是一个图标和一个套在方框中的“+”号，单击“”号或双击该图标，User1 中的所有类的名称将会被显示，如 CMainFrame、CUser1App、CUser1Doc、CUser1View，如图 1.3 所示。每个类名前也有一个图标和一个套在方框中的“+”号，双击类图标则可直接打开并显示类定义的头文件(如 CUser1App.h)；单击类前面的“”号则会显示该类中的成员函数和成员变量；双击成员函数前的图标，则可直接打开实现代码文件并显示相应函数体以供用户编辑。

注意此栏中的一些图标所表示的含义，如在成员函数的图标中，使用紫色方块表示公共成员函数，使用紫色方块和一把钥匙表示私有成员函数，使用紫色方块和一把锁表示保护型成员函数；又如用蓝绿色图标表示成员变量。

在 ClassView 中的任何标题或图标处单击鼠标右键都会弹出一快捷菜单，菜单中包含当前状态下对该元素的一些常用操作命令，使用这些命令能帮助我们快速编程。例如在类图标、类名称或类前面的“”处单击鼠标右键时，将弹出如图 1.4 中的快捷菜单。

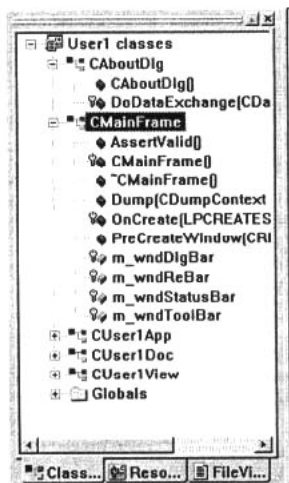


图 1.3 ClassView 显示了项目的所有类

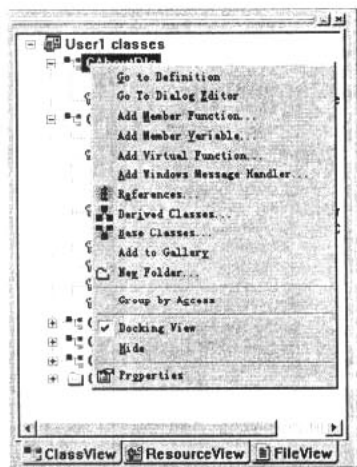


图 1.4 类的快捷菜单

三、ResourceView

ResourceView 包含了项目中所有资源的层次列表。在 Visual C++ 中，每一个图片、字符串值、工具条、图标或其他非代码元素都可以看做是一种资源。图 1.5 显示了一个典型的 ResourceView 窗口，当然，每一种资源都有自己使用的图标。



图 1.5 ResourceView 栏的内容



图 1.6 分类资源的快捷菜单

在 ResourceView 中的任何标题或图标上单击鼠标右键都会弹出一快捷菜单，它包含当前状态下的一些常用操作；如若在分类资源图标、名称或图标前面的“+”处单击鼠标右键时，可弹出如图 1.6 中的快捷菜单。

若在 ResourceView 中的最顶层的文件夹处单击鼠标右键将弹出一个快捷菜单，它允许你访问其中两个特殊对话框“Resource Include”和“Resource Symbols”。

四、FileView

FileView 可将应用工程中的所有文件(C++源文件、头文件、资源文件、Help 文件)分类显示，如图 1.7 所示。

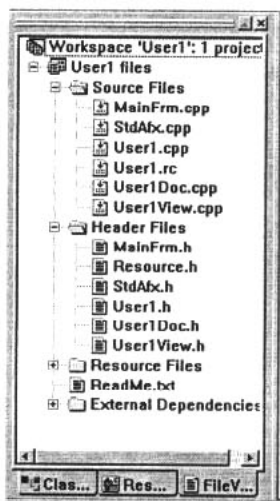


图 1.7 FileView 栏内容

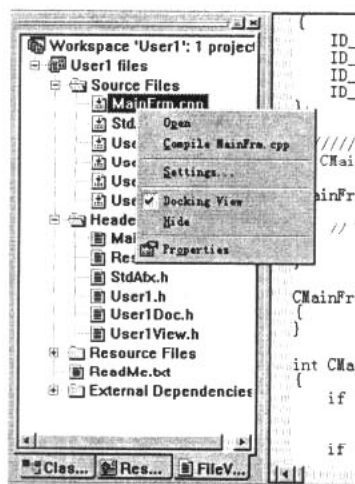


图 1.8 单击文件时的快捷菜单

每一类文件都有自己的文件夹，例如所有的 C++源文件都在 Source Files 文件夹中。你不仅可以在文件夹中移动文件，而且还可以创建新的文件夹以及将一些特殊类型的文件放在该文件夹中。

在 FileView 中的任何标题或图标处单击鼠标右键都会弹出一快捷菜单，它包含当前状态下的一些常用操作，如若在文件类型目录中的文件名(如 Source Files 目录中的文件 MainFrm.cpp)或图标处单击鼠标右键时，可弹出如图 1.8 中的快捷菜单。

若创建一个新文件夹，可在添加文件夹的地方单击鼠标右键，从弹出的快捷菜单中选择“New Folder”，将出现如图 1.9 所示的对话框，只要输入文件夹名称和相关的文件扩展名，单击[OK]即可。

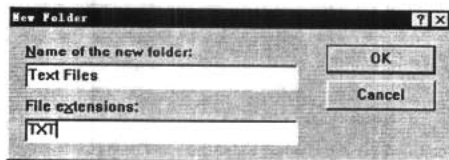


图 1.9 New Folder 对话框

五、MSDN 的在线帮助

Visual C++提供了三个不同的在线帮助来源：

- 由 WinHlp32 查看器显示的标准 HLP 文件。
- 对话框中的弹出式帮助信息。
- Microsoft Developer Network Library，即 MSDN。

标准的 HLP 文件内容包括环境的命令和窗口，当 Help(帮助)菜单的 Use Extension Help(使用扩展命令)命令被选中，或 Visual C++不能为帮助主题确定明确的上下文时，按 F1 键就会显示这些帮助文件。

在线帮助的第二个来源是弹出式消息，在这个环境中的许多对话框中都可以使用。选项卡和偶尔的提示会尽量解释对话框中编辑框和按钮的用途，但说明不充分时可以通过其他寻求到更多的解释。

在线帮助的第三个来源是你在 Visual C++ 中工作时最常用的，即 MSDN(Microsoft Developer Network)，所有的工具都可以共享它。MSDN 作为一个应用程序单独运行，不同任何单个开发环境结合在一起，故使用之前需要进行 MSDN 的安装。

要从 Visual C++ 中访问 MSDN，首先 Help(帮助)菜单中的 Use Extension Help(使用扩展帮助)命令必须没有被选中。图 1.10 展示了典型的 MSDN 界面，以左右两窗格显示。这两个窗格被设计成可以链接工作，左边的允许为你希望阅读的文章输入关键词，而右边的则显示找到的文章的内容。



图 1.10 中文版的 MSDN

按 F1 键，可以从 MSDN 库得到上下文的相关帮助。当选择一个基于当前窗口和光标位置的主题时，将在 MSDN 窗口右边看到相关帮助内容。若不止一个可能的帮助主题，将看到一系列的选项。

六、工具栏及其定制

工具栏是我们关于 Developer Studio 界面要讨论的最后一个话题，你首先要知道 Visual C++ 6.0 具有的工具栏要比第一次使用它时所见到的要多得多。在现有工具栏上任何地方单击鼠标右键，将弹出如图 1.11 所示的快捷菜单。

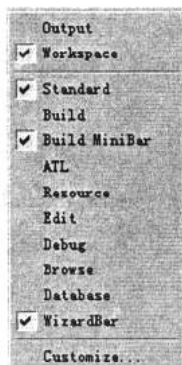


图 1.11 工具栏的快捷菜单

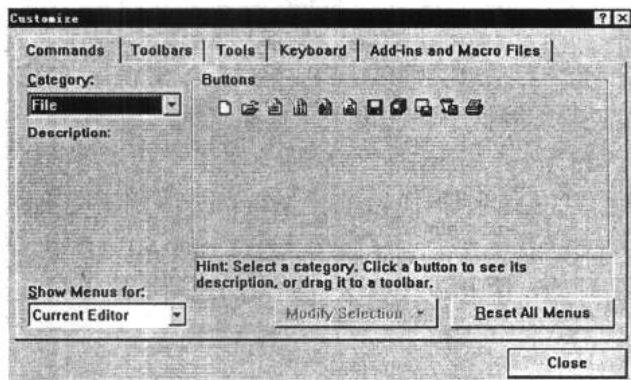


图 1.12 Customize 对话框

快捷菜单所显示的是 Visual C++ 6.0 提供的现有标准工具栏清单。任何带有选中标记的菜单都显示在工具栏区中。如果要在工具栏区中打开某工具栏,只需在快捷菜单中点选相应项即可。同样再操作一次即移去某个工具栏。

单击快捷菜单的 Customize 选项(或菜单项“Tools—>Customize...”),将弹出如图 1.12 所示的对话框。Customize 对话框含有许多属性页,通过它们可重新配置 Developer Studio 的大部分界面。也许,我们最关心的是 Commands 属性页,它包含一系列菜单或工具栏的图标或文本命令。若想给一个已存在的工具栏添加某条命令时,只需选取该图标(或文本命令),并把它拖到工具栏上即可。这样使用起来就会快多了。若选择了一条命令但没有相关的图标,那么“Button Appearance”对话框将允许你为该命令选择一个图标。菜单也有相同的特性。你可以打开需要更改的下拉式菜单,从 Commands 属性页中选取想要添加的命令,把它拉到菜单中合适的位置上,但看到的将是命令的文本形式而不是图标。若从工具栏或菜单中删除无用图标,只需选取要删除的命令项,把它拉回到 Customize 对话框中,这条命令就会从工具栏或对话框中消失。若需要移动工具栏,只要点中工具栏左边的两条竖线,就可以把它拉到你需要的任何位置,对于菜单栏也可同样操作。

注意,VC++6.0 还提供了向导条(WizardBar),如图 1.13 所示。利用它可以更方便地处理类、成员函数和资源。WizardBar 含有三个相互关联的列表框(分别表示为类、标识、成员函数)和一个 Action 控制图标(包括一个下拉式按钮)。

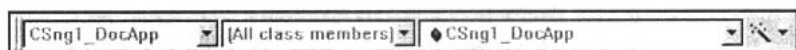


图 1.13 WizardBar

第二节 建立一个基本的应用程序

我们知道 C++是以类为基础的,通过类的派生和类成员函数的重载来反映事物之间的关系。在 Visual C++6.0 中,类同样是最重要的因素。如一个字符串可以是一个类 CString 的对象,一个窗口可以是类 CWnd 的对象,一个对话框可以是类 CDialog 的对象。

为了帮助程序员处理那些经常使用却复杂繁琐的有关 Windows 的各种操作,Visual C++6.0 设计了一套基础类库(MFC)。MFC 把 SDK 中的大多数内容封装成为各种类,使程序员从繁杂的 SDK 编程中解脱出来,提高了编程效率。

除了 MFC 外,Visual C++6.0 还定义了一些常用的标准程序结构和编程风格,我们可以称之为应用程序框架结构,如一般 Windows 应用程序结构、DLL 程序结构、基于对话框程序结构、单文档及多文档程序结构。在 Visual C++6.0 中,编程总是从生成一个适当类型的应用程序框架开始,因此,理解这些类型的应用程序框架是很必要的。

一、理解应用程序类型

Visual C++6.0 的 AppWizard 提供的应用程序框架类型如表 1.1 所示。

表 1.1 AppWizard 创建的应用程序类型

名 称	项 目
ATL COM AppWizard	创建 ATL(Active Template Library)应用模块工程
Cluster Resource Type Wizard	创建 Cluster Resource(用于 Windows NT 服务器)
Custom AppWizard	创建非标准的应用程序
Database Project	创建数据库应用程序
DevStudio Add-in Wizard	创建 ActiveX 组件或 VBScript macros
ISAPI Extension Wizard	创建 Internet Server 程序
MakeFile	创建独立于 Visual C++开发环境的应用程序
MFC ActiveX Control Wizard	创建 ActiveX Control 应用程序
MFC AppWizard(.dll)	MFC 的动态链接库
MFC AppWizard(.exe)	一般 MFC 的 Windows 应用程序
Utility Project	创建简单、实用的应用程序
Win32 Application	其他 Win32 的 Windows 应用程序
Win32 Console Application	Win32 的控制台应用程序
Win32 Dynamic-Link Library	Win32 的动态链接库
Win32 Static Library	Win32 的静态链接库

这里，我们主要讨论三种类型的应用程序框架：控制台应用程序、基于对话框的应用程序、单文档或多文档的应用程序。这几种是最基本的也是最常用的类型，随着内容的深入，其他应用程序框架的内容也会涉及到。

控制台应用程序是指那些需要与传统的系统 DOS 保持某种程度的兼容，同时又不需要为用户提供完善界面的程序。使用控制台应用程序的最大好处是不需要把原先 DOS 下的代码进行转换就可直接在 Windows 下运行。

“基于对话框的”通常针对的是实用程序，或者是不需要一个完整的菜单系统的小应用程序。构造一个基于对话框的应用程序的好处是：速度快、代码少，程序员所花费的开发和调试时间短。

单文档应用程序是用于类似于小型图形编辑器(如 Microsoft Painter)或小型文本编辑器(如 Notepad)的程序。由于它一次只能处理一个文档，因此它降低了编程的复杂性，减少了运行程序所需要的资源。单文档应用程序允许用户对创建的文档进行全面操作，但与多文档应用程序相比功能比较弱；例如，单文档应用程序的菜单比多文档应用程序的菜单少，没有选择编辑文档的 Windows 菜单。

多文档应用程序允许同时打开多个文档，其功能最为丰富。正因为如此，多文档应用程序需要大量的编程工作，例如它不仅需要跟踪所有打开文档的路径，而且还需要管理 Windows 菜单以及增加一些特殊的功能(像“切分窗口”)。故选择编写多文档应用类型程序时，要考虑一下是否真的需要。

二、编制一个控制台应用程序

正如上面所述，控制台应用程序允许把程序中的一些语句从 DOS 直接移植到 Windows 中。虽然最新版本的 Visual C++6.0 提供的控制台应用程序类型可以让 DOS 程序员方便地进行 MFC 应用程序的开发，但我们不希望初学者选择这条途径，否则你将陷入无助的境地。尽管如此，我们还是来创建一个简单的控制台应用程序。

首先，我们按下面步骤创建一个项目 Console：

- ▼ 从主菜单中依次选择“File→New”，显示出 New 对话框。
- II 选择 Projects 标签，并从列表框中单击 Win32 Console Application 项目图标。



图 1.14 在 New 对话框中建立一个 Console 控制台应用程序

- II 输入项目名称：Console，第一次使用时最好选定一个项目所在的(Location)文件夹，如图 1.14 所示。
- II 单击[OK]钮继续。
- II 一个询问项目类型的 Win32 应用程序向导将出现，如图 1.15。选择 An empty project 单选钮。
- 单击[Finish]钮，系统开始创建此应用程序。

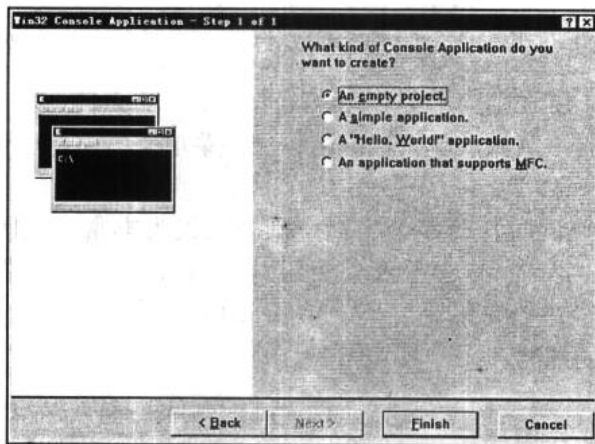


图 1.15 选择控制台应用程序类型

然后，在工具栏上单击 New Text File 图标，编写你的第一个 C++源程序，如下所示。
(加入的程序用右侧的文武线标明)

```
//我的第一个控制台应用程序
#include "stdafx.h"
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

接着，将刚才输入的源程序存贮起来。单击工具栏中的 Save 图标或从 File 菜单选择“Save”或按“Ctrl+S”快捷键，将源文件以 Console.cpp 为名(或其他名称，但必须以.cpp 为文件扩展名)存贮到项目目录中。

再从主菜单中依次选择“Project→Add to Project→Files”，将显示 Insert Files into Project 对话框，选择 Console.cpp 源文件，单击[OK]钮。

编译并运行程序。运行后屏幕显示“Hello World!”，这就是此程序所实现的功能。

三、编制一个单文档应用程序

在 Windows9X/NT 环境下，用 MFC 进行软件的开发一般是从单文档应用程序开始的。接下来，我们启动 MFC AppWizard 来创建一个单文档应用程序，这也是我们的第一个 Windows 程序。

开始：

从 File 菜单选择 New，弹出 New 对话框；选择 Projects 标签，在显示出的一系列应用程序项目类型中选择 MFC AppWizard(exe)作为项目类型。在 Project Name 字段中输入项目名，本例使用 Sngl_Doc 作为项目名，但你也可使用其他名称；单击[OK]钮。

第一步：

出现如图 1.16 所示的对话框，选择 Single Document 选项，然后按下[Next]钮。

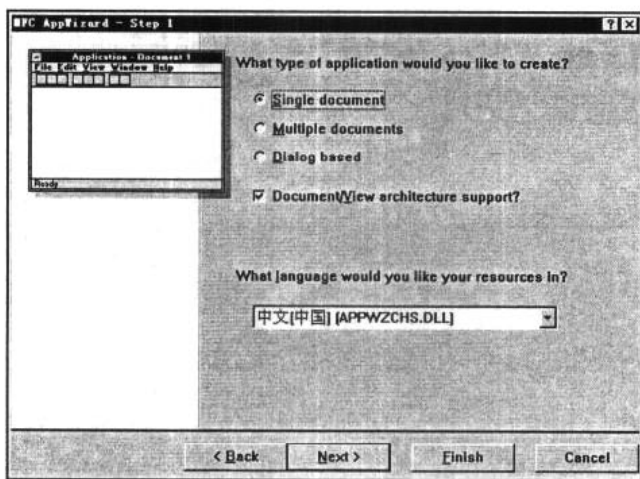


图 1.16 Step 1 对话框

第二步：

出现如图 1.17 所示的对话框要你进行一些选择。如果要在程序中加入数据库的支持，那么还需要选择数据库支持的层次。这部分内容将在本书的最后一章进行讨论。如图点选，单击[Next]钮进行下一步。

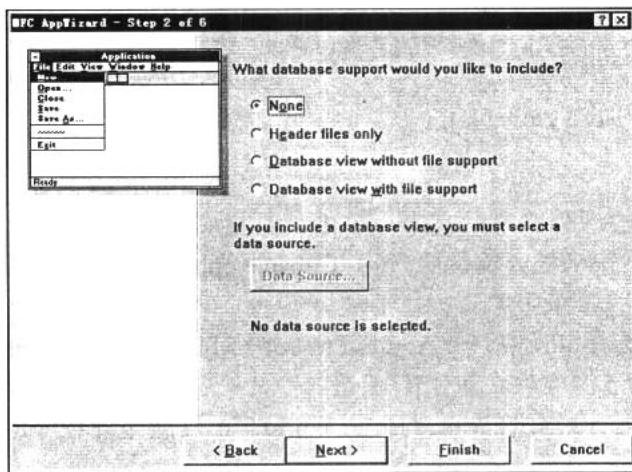


图 1.17 Step 2 对话框

第三步:

出现如图 1.18 所示的对话框, 这个对话框的上半部分(前五个选项)是设置向应用程序中加入 OLE 支持的层次。加入的层次越深, 支持越多, 程序也就越大。最基本的支持是作为一个容器(Container), 容器可以作为一个客户, 可存储链接和嵌入对象。下一个层次是小型服务器(Mini-server), 允许创建复合文档, 一个小型服务器不能单独工作, 它只能操作内嵌文档。再下一个层次是全服务器, 它具有完整的 OLE 功能, 可以像一个服务器那样工作, 但它不能完成容器的任务。最后的“Both Container and Server”选项给予应用程序一套完整的本地 OLE 支持, 它既可以是一个服务器又可以是一个客户端, 若将它应用于 Internet 浏览器中, 则必须将它设计成 ActiveX 文档服务器; 在“Both container and server”的下方有两个复选框, 我们可以将其设置为对 ActiveX 文档的支持层次(Server 和 Container)。

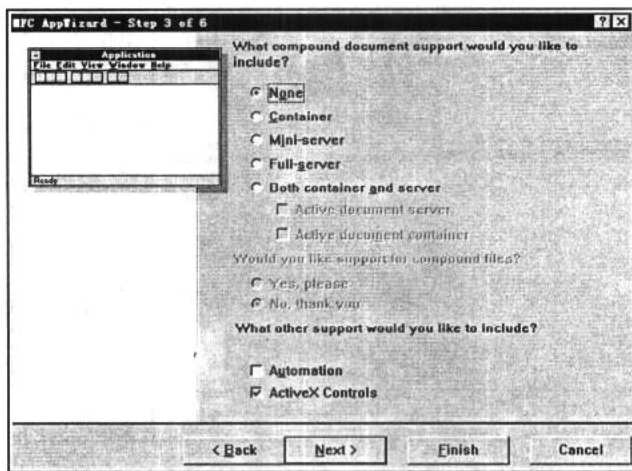


图 1.18 Step 3 对话框

最后的提问中有两个重要的复选框。第一个是 Automation。对于较大的应用程序应该选择这一项, 因为 Automation 允许操作其他应用程序创建的对象以及自动化客户中的对象。第二个选项是 ActiveX Controls, 它允许在程序中使用 OCX 控件。

保留缺省状态, 单击[Next]钮进行下一步。

第四步:

出现如图 1.19 所示的对话框。对话框的上半部分(前五项)是确定应用程序的界面元素,它们依次提供对浮动工具条、状态条的初始化、打印与预览、上下文帮助、3D 控件的支持。

接下来的复选框是对通信的支持,第一个是与 Email 有关的,第二个是支持 TCP/IP 网络的,若要程序加入 Internet 支持可选择此项。再往下是确定工具条的外观是标准的还是 IE4.0(及以后)的外观(即 Rebar)。

最后一项用于设置保留最近文件列表数目。单击“Advanced”按钮将出现如图 1.20 所示的“Advanced Options”对话框,这是为应用程序设置文档参数的地方。例如,若要处理的文件的扩展名为 xyz,则 File Open 和 File Save 对话框过滤栏显示的是“单文档文件 (*.xyz)”。设置如图 1.20 所示的参数。

单击[Close]按钮回到图 1.19 的对话框中,单击[Next]按钮进行下一步,进入图 1.21 所示的对话框中。



图 1.19 Step 4 对话框

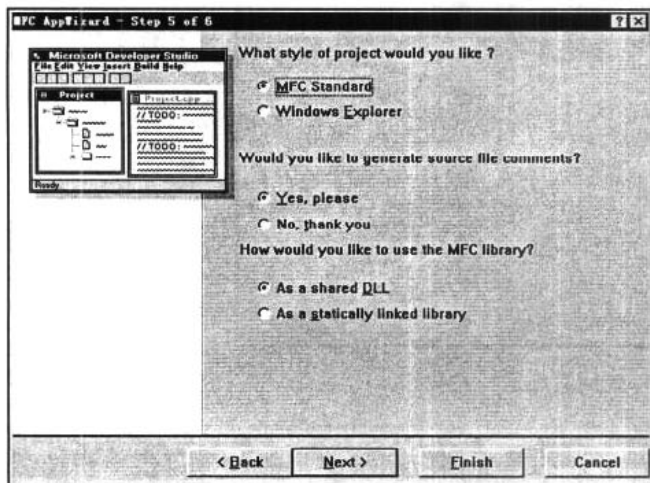


图 1.21 Step 5 对话框

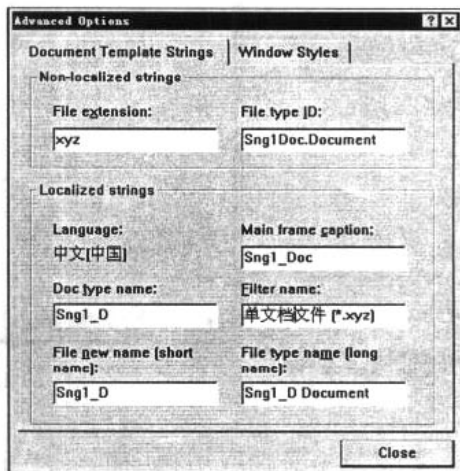


图 1.20 Advanced Options 对话框

第五步：

在此对话框中提出了三个问题：“项目窗口风格是 MFC 标准风格还是窗口左边有切分窗口的浏览器风格？是否包含资源文件组件？使用动态链接库还是静态链接库？”保留缺省状态，按[Next]进行下一步。

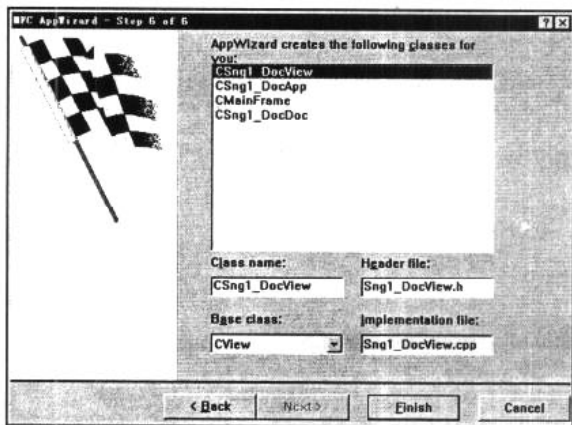


图 1.22 Step 6 对话框

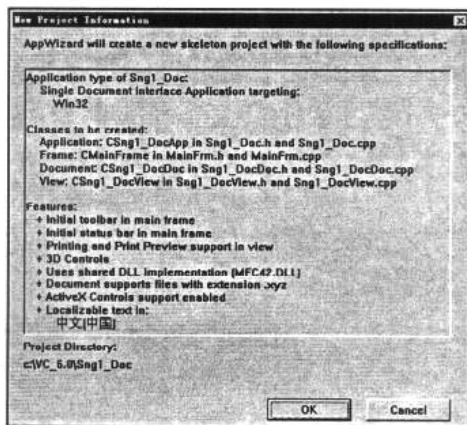


图 1.23 项目信息对话框

第六步：

你可以从此对话框中改变类的名称及基类，以实现应用程序的其他功能。也可将头文件、应用文件的名称修改。一切满意时，按[Finish]钮出现如图 1.23 所示的对话框，按[OK]钮。

上面过程中你没有编写任何程序代码，完全由向导生成的程序经编译并运行后出现如图 1.24 的结果。

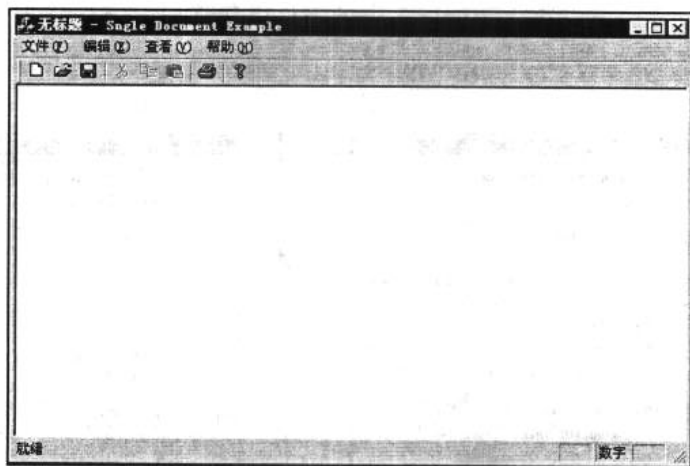


图 1.24 应用程序运行结果

这就是单文档应用程序框架。当然，上述的单文档应用程序只能包含可供用户选择的 MFC AppWizard 提供的功能，用户还可对其功能进行增加、更新。在这里我们把功能的增加和更新称之为定制；应用程序的定制(通常用 MFC)是任何程序员所必须掌握的方法和技巧，本书以后的章节将逐步对此进行详细深入地探讨。

第二章 菜单、工具栏和状态栏

菜单、工具栏和状态栏是 Windows 应用程序中不可缺少的界面元素。它们的风格和外观有时直接影响着用户对软件的评价,许多优秀的软件(如 Office 97,甚至是 Windows98)为增加对用户的吸引力,不惜资源将它们做得多姿多彩,甚至达到真三维的效果。可见,菜单、工具栏和状态栏在应用程序中是多么重要。本章将从它们最简单的用法开始入手,逐步深入地讲解如何对其进行编程控制(并列出它们常见的成员函数),最后用 CRebar 和 CRebarCtrl 定制一个具有 Windows 98(或 Windows NT 4.x)风格的用户界面(菜单、工具栏和状态栏)。

第一节 菜单

菜单是一系列命令的列表,用户能够选中命令并执行相应任务。除一些简单的基于对话框的应用程序外,所有的 Windows 程序都提供了一些不同类型的菜单。

为了使 Windows 程序更容易使用,许多程序员对于菜单的显示都遵循一些规则。这些规则是:

- 若点击某菜单项会弹出一对话框需要用户输入一些内容,那么在该菜单文本后添上“...”(三个小点)。
- 若菜单项需要助记符,则使用带下划线的字母。通常是菜单项文本的第一个字母,当然也可以是其他字母。使用时敲击该字母,对应的菜单项就会被选中。尤其注意:助记符不能重复,否则菜单不会按预期的进行。敲击重复的助记符会使第一个使用该助记符的菜单项高亮显示,一直等到按下回车键此菜单项才被选中,继续敲击该助记符,则下一个使用该助记符的菜单项高亮显示。
- 若某菜单项带有子菜单,可在菜单项文本后加上一个向右的箭头(▶)。
- 若某菜单项需要快捷键的支持,可将此快捷键显示在该菜单项文本之后。但快捷键要慎用,因为它占用资源,快捷键太多不宜记住,达不到预期的效果。

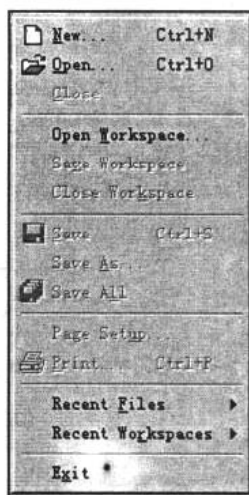


图 2.1 File 菜单

图 2.1 显示了 Visual C++6.0 的 File 菜单,请对应上文注意它们的使用规则。

一、用编辑器设计菜单

依次选择菜单“File→Open Workspace”，将我们的第一个单文档应用程序 Sng1_doc.dsw 调入(.dsw 是 Visual C++6.0 专用项目扩展名)。在项目区中选择 ResourceView 标签，双击 Menu 中的 IDR_MAINFRAME，则菜单编辑器窗口出现在右边。现在，我们就可以设计菜单了。

在编辑器窗口显示的菜单上，我们单击“帮助”(或其他)，这时在菜单的最后一项 Visual C++为我们留出了一个空位置，用来输入新的菜单项，如图 2.2 所示。

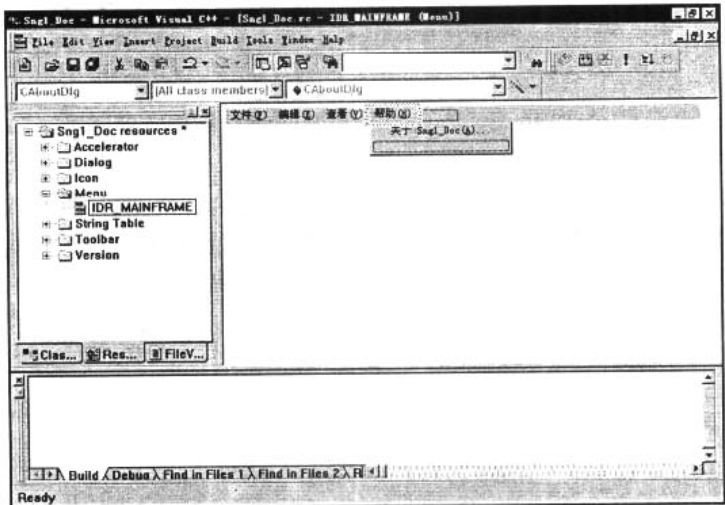


图 2.2 菜单编辑器

在空位置上双击鼠标左键，出现它的“属性”对话框，在此对话框中我们定义菜单项的显示文本和资源 ID 号。虽然 ID 号的定义是随意的，但通常按一定的规则来确定，即使使用时便于理解和记忆。

作为一个 VC++程序员，了解标识符 ID 的命令规则是有必要的，它的规则如下：

- 在标识符名称中允许使用字母 a~z、A~Z、0~9 以及下划线。
- 标识符名称不区分大小写字母，如 new_idd 与 New_Idd 是相同的标识符。
- 不能以数字开头，如 8BIT 是不合法的标识符名。
- 字符个数不得超过 247 个。

除了上述规则外，出于习惯，VC++系统提供了一些常用的定义标识符名称的前缀供用户使用和参考，见表 2.1。

表 2.1 常用标识符定义的前缀

标识符前缀	含义
IDR_	表示快捷键或菜单及相关资源
IDD_	表示对话框资源
IDC_	表示光标资源或控件
IDI_	表示图标资源
IDB_	表示位图资源
IDM_	表示菜单项
ID_	表示命令项
IDS_	表示字符串表中的字符串
IDP	表示消息框中使用的字符串