

按实例学编程语言丛书



按实例学

Visual C++ 6.0

韩世伟 等 编著



科学出版社

TP312
HSW/1

按实例学编程语言丛书

按实例学 Visual C ++ 6.0

韩世伟 等 编著



科学出版社

2000

053576

312

内 容 简 介

《按实例学编程语言丛书》的编写宗旨是把实例与概念融为一体,通过典型丰富的实例,介绍实际中最为常用的几种编程语言的操作方法和软件开发方法。

本书是《按实例学编程语言丛书》之一,书中内容由浅入深、循序渐进地介绍了 Visual C++ 6.0 的使用方法,从使用 AppWizard 建立基本程序开始,讲解了控件的使用,对话框的编程等基本内容,并对数据库编程与 ActiveX 控件,以及多媒体编程等技术作了详细的介绍。

本书提供了大量的实例,每一部分内容都包含详细的操作步骤。通过对本书的学习,读者在操作和使用过程中可轻松自如地学会用 VC 6.0 开发各种应用软件。

本书内容全面、叙述清晰,适合于 Visual C++ 6.0 的初学者作为入门学习的指导书,同时对已有使用 VC 经验的读者来说,书中的一些技巧也同样有借鉴意义。

图书在版编目(CIP)数据

JS364/18

按实例学 Visual C++ 6.0/韩世伟等编著.-北京:科学出版社, 2000

(按实例学编程语言丛书)

ISBN 7-03-007985-X

I. 按… I. 韩… II. C 语言-程序设计 N. TP312

中国版本图书馆 CIP 数据核字(1999)第 71277 号

科学出版社 出版

北京东黄城根北街 16 号
邮政编码:100717

新蕾印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 3 月第 一 版	开本: 787×1092 1/15
2000 年 3 月第一次印刷	印张: 19 3/4
印数: 1—5 000	字数: 449 000

定价: 26.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

丛 书 编 委 会

主 编	姜淑娟	李洪涛	张永平
副主编	王潜平	韩世伟	张 磊
编 委	顾 颖		
	张志丽	丁金芬	李洪涛
	阎大顺	刘建光	李向群
	林果园	曹天杰	夏战国
	刘 斗	吕凤霞	王茂光
	宋协武	朱 亮	刘晓东
	梁 勇	马元珠	邓 力
	朱宏博	孙晓静	李国强
	陆 宇	华 钢	王 慧
	左 明		

出版说明

用“日新月异”来形容计算机技术的发展再恰当不过了。计算机操作系统几乎一年更新一个版本,应用软件换代升级则更快。要想跟上新技术的发展,不要说非专业人士,就是计算机界的业内人士也颇感吃力。

自从一些大的计算机公司推出可视化编程系列软件以来,过去仅能由计算机专业人员开发的应用软件,现在非计算机专业人员也可轻松自如地编写了。编程方法从手工文字走向了可视化拖放技术。掌握并应用好这些最新编程技术一直是计算机爱好者的期望。

为了进一步向全社会普及计算机技术,适应计算机技术的发展趋势,我们组织编写了这套目前最为流行的可视化编程语言丛书——《按实例学编程语言丛书》。

本套丛书共分五本,它们是:

按实例学 Visual Basic 6.0

按实例学 Visual C++ 6.0

按实例学 Visual FoxPro 6.0

按实例学 Access 2000

按实例学 Delphi 5.0

本丛书具有以下特点:

- 以实例为基础,以操作为主线
- 包含完整的软件开发实例
- 以图形方式简明详细地讲解操作步骤
- 开发源码全部公开
- 面向初、中级水平的计算机爱好者
- 内容通俗易懂,采用最新版本

丛书编委会

1999年10月

前 言

进入 20 世纪 90 年代以来,计算机可视化技术得到了迅猛发展,并引起广泛重视,越来越多的计算机专业人员和非专业人员开始研究并应用可视化技术。

微软公司推出的 Visual C++ 6.0 是目前使用极为广泛的可视化开发环境,它是一种功能强大而复杂的 C++ 编译器。它作为可视化开发的强有力环境,用来建立 Windows 95/98 和 Windows NT 应用程序。Visual C++ 提供了功能最强大的 Windows 应用程序框架,微软基本类库(MFC)封装了 Win 32 API 的函数,提供了大量编写好的代码,以实现多数 Windows 标准操作。在 Visual C++ 中,编写应用程序,即可以利用 MFC 封装的类库编写程序,也可以使用 SDK 调用 Windows 本身提供的 Win 32 API 函数编程,而最方便的是利用 AppWizard 向导建立程序的主体部分。这一方法简化了 Windows 编程,减轻了程序员的负担,使程序员将精力集中在创造新思想上,而将许多烦琐的 Windows 基本操作编程由 AppWizard 完成。使用 AppWizard 生成的应用程序框架将使 Windows 应用程序标准化,从而有利于程序员之间的相互理解、相互沟通。Visual C++ 是一个开放的体系,有很多第三方开发者在为 Visual C++ 提供控件,使程序员的编程更简单、更有效。Visual C++ 6.0 必将带给您一种全新的感受,使您工作起来更加简便快捷,帮助你梦想成真。

本书对 Visual C++ 6.0 使用与开发所应掌握的基本内容和方法作了全面系统的介绍,全书内容共分 18 章,循序渐进地讲解了 Visual C++ 6.0 的使用方法,从使用 AppWizard 建立基本程序开始,讲解了控件的使用、对话框的编程等基本内容,并对数据库编程与 ActiveX 控件以及多媒体编程等技术作了详细的介绍。

参加本书编写的人员还有林果园、曹天杰、刘晓东、梁勇、马元珠、邓力、朱宏博、孙晓静等。

在本书的编写和出版过程中得到了作者的朋友和同事很多支持和帮助,特在此表示衷心感谢。

由于编者水平有限,加之成书仓促,欠妥之处在所难免,恳切希望广大读者批评指正。

编 者

1999 年 10 月

目 录

第一章 创建最简单的 VC 程序	1
1.1 使用 AppWizard 示例	1
1.2 Visual C++ 的 AppWizard 程序向导介绍	9
第二章 对话框及 ClassWizard 编程示例	12
2.1 对话框知识简介	12
2.2 消息对话框	12
2.3 自定义对话框	14
第三章 菜单编程示例	26
3.1 菜单的相关知识	26
3.2 创建示例程序框架	27
3.3 设计自定义菜单	28
3.4 示例执行结果说明	37
3.5 添加键盘加速键	38
第四章 控件的简单使用——按钮和编辑控件使用示例	41
4.1 什么是控件	41
4.2 创建示例程序框架	41
4.3 设计示例程序界面	42
第五章 工具条的使用示例	49
5.1 创建示例工程文件	49
5.2 程序窗口设计	51
5.3 添加工具条图标	54
第六章 控件使用示例二	67
6.1 示例程序简介	67
6.2 创建示例工程	67
6.3 程序用户界面设计	68
第七章 按钮控件的高级示例——制作圆形按钮	90
7.1 建立示例工程文件	90
7.2 创建新类 CRoundButton	90
7.3 创建圆形按钮	96
第八章 控件高级示例——液晶显示时间	98
8.1 创建示例工程文件	98
8.2 创建新类 CDigitClock	99
8.3 进行液晶显示编程	106
第九章 鼠标使用示例	113

9.1	鼠标知识及示例工程创建	113
9.2	示例程序的编程	113
9.3	程序的改进	119
第十章	键盘使用示例	121
10.1	键盘码知识及示例工程	121
10.2	示例窗口设计及代码编程	122
第十一章	如何使用字体	125
11.1	示例工程框架创建	125
11.2	程序窗口设计及字体使用	125
11.3	使用字体对话框	132
第十二章	数据库编程示例	137
12.1	创建示例程序框架	137
12.2	设计程序界面	137
12.3	添加数据处理类 CCrack 类	139
12.4	数据库编程	147
第十三章	ActiveX 控件的使用示例	166
13.1	创建应用程序工程框架	166
13.2	提取 ActiveX 控件	167
13.3	使用 ActiveX 控件编程	168
第十四章	多媒体编程	175
14.1	创建多媒体程序的工程框架	175
14.2	创建多媒体处理类 CMciDevice	175
14.3	创建播放 CD 类 CCDAudio	184
14.4	多媒体程序设计	191
第十五章	Visual C++ 程序的调试	207
15.1	启动调试器	207
15.2	调试菜单介绍	207
15.3	调试要点	209
15.4	用于调试的宏和函数介绍	210
第十六章	帮助的制作	212
16.1	帮助系统介绍	212
16.2	创建示例程序工程框架	213
16.3	在 Word 中制作帮助文本	215
16.4	在 VC6 中制作帮助	221
第十七章	使用 InstallShield 示例	225
17.1	InstallShield 介绍及使用	225
17.2	InstallShield 工作区介绍	228
17.3	程序的卸载过程简介	235
第十八章	VC 编程技巧集锦	236

18.1	打开和编辑二进制文件	236
18.2	检测括号是否匹配	236
18.3	检测 #if 语句和 #endif 语句是否匹配	236
18.4	删除工程的中间文件以便于备份工程	236
18.5	快速显示当前函数的有关信息	236
18.6	如何优化工作区中的树目录	237
18.7	如何使工程链接时链入某些库文件	237
18.8	如何查看编译、链接时的出错信息	237
18.9	如何进行文件操作	238
18.10	调用 FILEOPEN 系统对话框	239
18.11	在菜单中添加新的菜单项	239
18.12	程序运行时间统计	240
18.13	取得系统时间	240
18.14	如何使文档一生成就最大化	240
18.15	设置窗口的初始位置和大小	241
18.16	如何移动窗口	241
18.17	如何改变视窗的背景颜色	241
18.18	去掉主窗口标题	242
18.19	包含位图和文本的按钮	242
18.20	在工具条中显示文本	251
18.21	从剪贴板上读写文本	252
18.22	从剪贴板上获取文本	253
18.23	把位图放置到剪贴板上	253
18.24	加载应用程序	254
18.25	退出应用程序	256
18.26	利用图标与用户交互	256
附录一 VC 常用控件的使用		260
1.	静态控件	260
2.	按钮控件	260
3.	列表框控件	261
4.	List 控件	264
5.	编辑框控件	266
6.	组合框控件	268
7.	滚动条控件	269
8.	拖放列表框	270
9.	动画控件的使用	271
10.	旋转按钮控件	272
11.	滑动块控件(Slider 控件)	273
12.	标签控件(Tab 控件)	275

13. Tree 控件	278
14. 进度条控件(Progress 控件)	283
附录二 MFC 常用类介绍	284
1. CPoint 类	284
2. CRect 类	285
3. CRgn 类	290
4. CSize 类	292
5. CString 类	294
6. CTime 类	301

第一章 创建最简单的 VC 程序

1.1 使用 AppWizard 示例

本章利用 Visual C++ 6.0 的 AppWizard 生成一个最简单的 VC 程序。实现步骤如下：

(1) 点击菜单 File 下的 New 选项, VC 会弹出如图 1-1 所示的对话框。系统弹出的对话框是让我们选择所要创建的文件类型, 这里的文件分为 Files, Project, Workspaces, Other documents 四种大类型, 每一个类型下面又包含许多具体的文件类型, 选中 Projects 标签, 标签下的工作区中列出的是各种不同的应用程序类型, 比如 dll 类型的动态链接库, exe 类型的可执行程序等, 这里选中 MFC AppWizard (exe) 选项, 表示要创建的是一个使用 MFC 基本类。

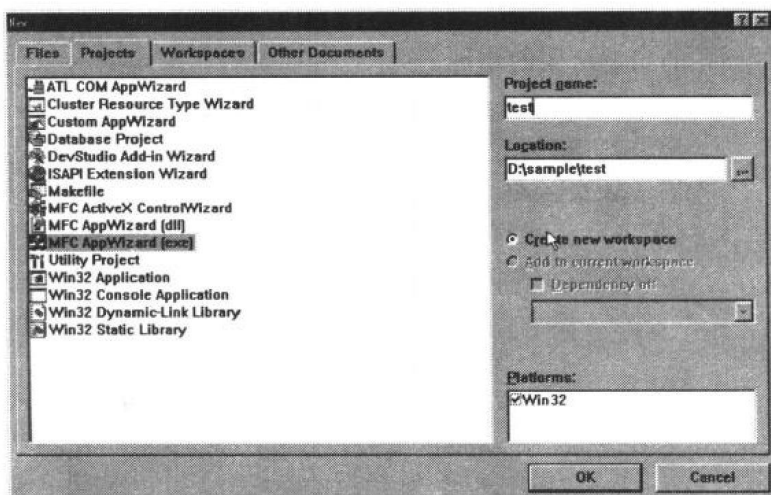


图 1-1 New 对话框 Project 标签

(2) 选好后在 project name 一栏中为程序起一个名字为 test, 在 location 一栏中为程序定义文件存放的目录, 对话框右下角的 platforms 一栏中的 Win32 项表示要创建的程序是建立在 32 位的 windows 平台基础上。如果要创建新的项目并将其添加到当前工作区中, 请选中 Add to current workspace 单选钮。需要的话, 还可以选中 Dependency of 复选框并指定项目名来使新项目成为已有项目的子项目。这里选中 Create new workspace 单选钮, 表示要创建一个新的工作区。从 platforms 列表框中选择可用的目标平台, 默认平台为 Win32。如果要选择其他目标平台, 那么首先要安装与该平台相关的 Visual C++ 6.0 的版本。点击 OK 进入下一步, 就启动了使用 MFC 方式开发应用程序的 AppWizard 功能。

图 1-1 上各项的意义如下：

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	创建资源文件向导
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 扩展
Makefile	创建独立于 VC++ 开发环境的应用程序
MFC ActiveX ControlWizard	创建 ActiveX Control 应用程序
MFC AppWizard(dll)	MFC 的动态链接库
Utility Project	创建工具程序
Win32 Application	Win32 的 Windows 应用程序(非 MFC)
Win32 Console Application	Win32 的控制台应用程序(非 MFC)
Win32 Dynamic-Link Library	Win32 动态库
Win32 Static Library	Win32 静态库

(3)AppWizard 弹出如图 1-2 所示的对话框,在该对话框中,AppWizard 让我们选择程序的类型和程序中的资源所用的语种。

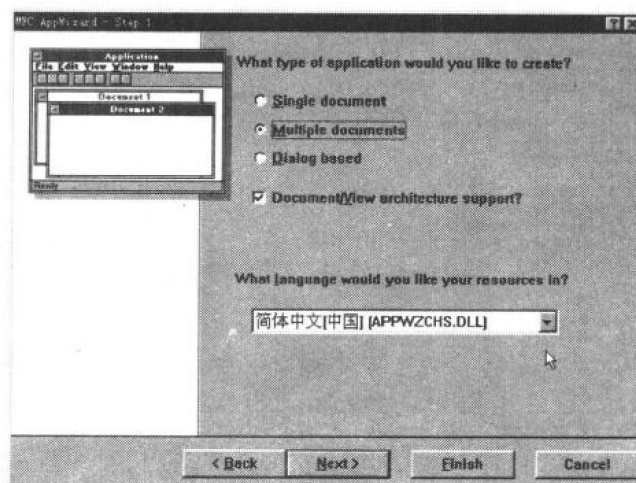


图 1-2 MFC AppWizard——Step 1 of 6 对话框

图 1-2 中各种选项是 AppWizard 可以创建三种不同类型的应用程序结构:

- Single document: 单文档界面(SDI),一次只允许打开一个文档。
- Multiple documents: 多文档界面(MDI),允许同时打开多个文档。如 Word 和 Excel 就是典型的 MDI 应用程序。
- Dialog based: 基于对话的应用程序,不支持文档/视图结构,而是仅仅显示一个简单的对话框。这对于小的应用程序是有用的。但如果要用到菜单、工具栏和打印等功能,那么就应该创建基于文档的应用程序。

选择应用程序结构时,对话框左上角会显示与所选择结构相对应的图表。例如,对于

多文档界面结构,在名为 Application 的窗口中将显示两个文档边框窗口。而对于单文档界面结构,则只显示仅有一个文件窗口的程序。

资源可用的语言中有中文、英语、法语、德语、西班牙和意大利语等。默认语言取决于当前计算机所用的 Windows。对于 Windows 简体中文版,默认语言就是中文。如果要选择其他语言,首先要确保已经安装了相应的动态链接库。比如要使用法语,则要安装 APP-WZFRA.DLL 等。

这里不妨选择 Multiple documents 单选钮,即程序类型为多文档界面,语种默认为简体中文,然后单击 Next 按钮,进入下一步操作。

(4)然后弹出如图 1-3 所示的对话框,AppWizard 让我们选择是否需要提供数据库方面的支持,以决定如何支持 ODBC 或 DAO(Microsoft Data Access Objects)。

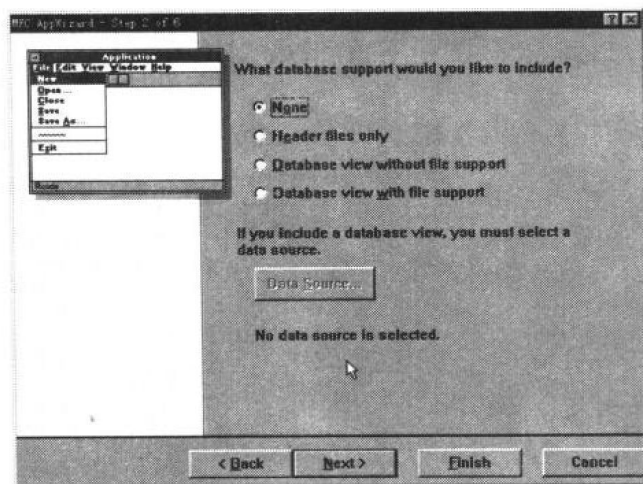


图 1-3 MFC AppWizard——Step 2 of 6 对话框

各选项的意义如下:

- None:不支持任何 ODBC 库。如果不使用数据库,那么选择该项将建立比较小的应用程序。
- Header files only:提供最低限度的数据库支持,包含数据库头文件(AfxDb. H)和链接库,但不创建任何与数据库相关的类,用户必须自己创建。
- Database view without file support:包含所有的数据库头文件和链接库,并创建与数据库相关的类。这个选项创建的应用程序允许用户查看和更新记录集中的记录,并具有文档支持但不支持串行化。
- Database view with file support:与上一选项唯一不同之处在于不仅有文档支持还有串行化支持。

如果应用程序包含数据库视图,那么还应定义数据源,方法是:单击 Datasource 按钮,从弹出的 Database Options 对话框来确定应用程序的数据源和记录集类型。

这里选中 None 单选钮,表示不带数据库支持。然后单击 Next 进入下一步。

(5)弹出可以选择复合文档(Compound document)支持选项对话框,如图 1-4 所示。

各项内容如下:

- None:默认设置,不带复合文档支持。

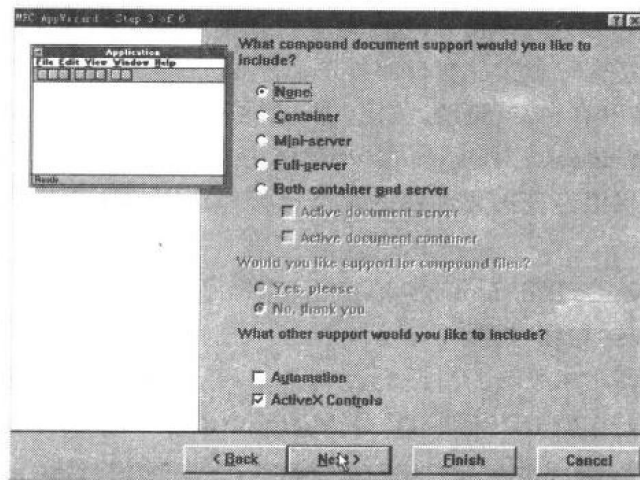


图 1-4 MFC AppWizard——Step 3 of 6 对话框

- Container: 应用程序作为容器,可以合并嵌入对象和链接对象到自己的文档中。
- Mini-server: 应用程序可以创建和管理复合文档对象,但不能单独运行,仅支持嵌入对象。
- Full-server: 应用程序既可创建和管理复合文档对象,又可单独运行,同时支持链接与嵌入对象。
- Both container and server: 应用程序既作为容器,又作为服务器。
- Yes, please: 使用复合文件格式来串行化容器的文档。复合文件格式能在一个文件中保存一个或多个自动化对象,并允许被单独的自动化对象访问。
- No, thank you: 不使用复合文件格式来串行化容器的文档。该选项必须一次将包含自动化对象的整个文件装入到内存中,不允许增量式保存单独的自动化对象。如果一个自动化对象被更改并加以保存,那么文件中所有的自动化对象都必须保存。
- Automation: 使应用程序支持自动化,这样应用程序就可以操作其他应用创建的对象,或者暴露自动化对象给自动化客户访问。
- ActiveX Controls: 使应用程序使用 ActiveX 控件。如果不选择该项,以后要插入 ActiveX 控件到项目中,就必须在应用程序的 InitInstance 成员函数中添加对 AfxEnableControlContainer() 函数的调用。

这里选中 None 单选钮,表示不带复合文档支持。

(6)单击 Next 按钮,弹出如图 1-5 对话框,从中可以选择各种不同的用户界面选项,并可以选择 MAPI 和 Windows Sockets 支持。

- Docking toolbar: 默认设置,添加工具栏应用程序中。工具栏默认含有多个按钮(如“新建”、“打开”、“保存”等)用于完成常见的操作(如文档建立、打开和保存等)。
- Initial status bar: 默认设置,添加状态栏到应用程序中。状态栏包含与键盘上某些键(如 Caps Lock, Num Lock 和 Scroll Lock 等)相对应的自动指示器,同时指示器,同时状态栏还显示与选单命令和工具栏按钮有关的帮助信息。
- Printing and print preview: 默认设置,添加代码来调用 Cview 类的成员函数,以便处理打印、打印设置和打印预览等命令。

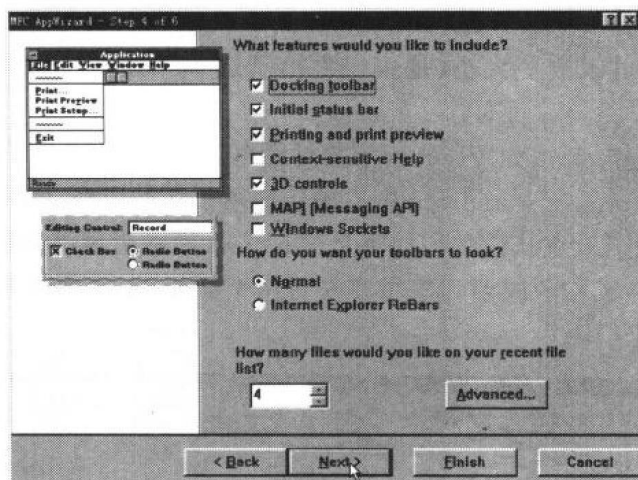


图 1-5 MFC AppWizard——Step 4 of 6 对话框

- Context-sensitive Help:生成帮助文件来辅助用户编写应用程序的帮助系统。
- 3D controls:默认设置,使应用程序中的界面具有漂亮的三维对观。
- MAPI(Messaging API):使应用程序可以创建、操作、传输和存储邮件消息。
- Windows Sockets:使应用程序可以通过 TCP/IP 网络通信。
- How do you want your toolbars to look:确定工具栏的风格。如果是传统风格,请选中“Normal”单选钮(默认设置);如果是 IE 风格,请选中 Internet Exploer ReBars 单选钮。IE 风格使工具栏可以包含任何标准 Windows 控件。
- How many files would you like on your recent file list:应用程序可以跟踪最近使用过的文件,默认时应用程序的“文件”选单底部列出 4 个最近使用过的文件。

如果要设置其他高级选项,请单击“Advanced”按钮,弹出如图 1-6 所示的 Advanced Options 对话框。

对话框有两个选项卡。

其中,DocumentTemplateStrings(图1-7)选项页用于设置应用程序的文档类型,如

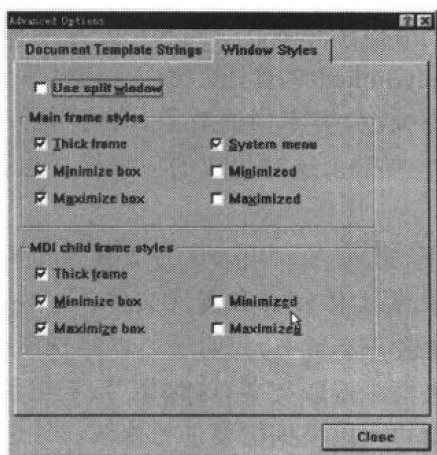


图 1-6 Advanced Options 对话框的
Windows Styles 标签页

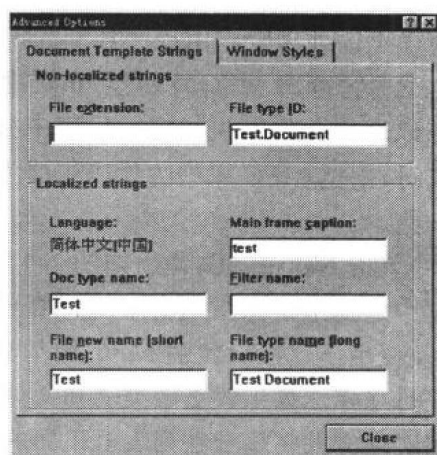


图 1-7 Advanced Options 对话框的
DocumentTemplateStrings 标签页

默认的文件名和扩展名等；而 Windows Styles(图 1-6)选项页用于确定窗口风格，如窗口是否有分割条，是否可以改变大小，是否包含控制选单和控制按钮等。设置后，请单击 Close 按钮返回图 1-5 所示对话框。

(7)单击 Next 按钮，弹出如图 1-8 所示的对话框，从中可以选择以下项目选项设置：

- What style of project would you like:确定应用程序的风格。如果是标准风格，请选中 MFC Standard 单选钮(默认设置)；如果是 Windows 资源管理器风格，请选中 Windows Explorer 单选钮，这种风格的应用程序带有左右两个面板，左边面板是一个树形结构，而右边面板是一个列表结构。

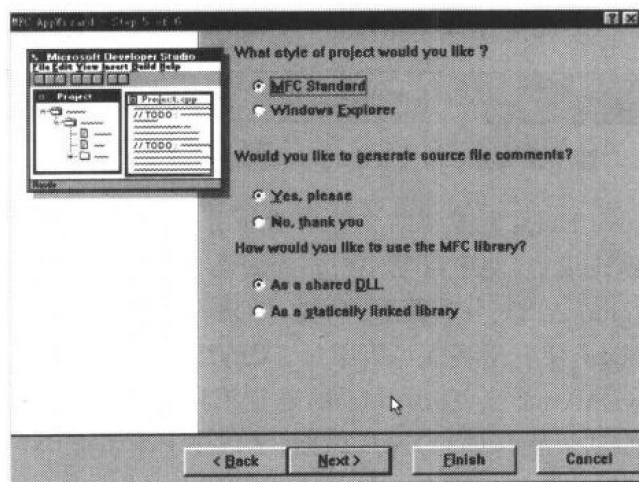


图 1-8 MFC AppWizard——Step 5 of 6 对话框

- Would you like to generate source file Comments:AppWizard 默认在源文件中插入相应的注释以帮助用户编写程序。源文件注释指示在何处需要添加用户自己的代码。如果不想插入，请选中 No,thank you 单选钮。

- How would you like to use the MFC library:选择 MFC 的静态链接库或动态链接库版本。如果应用程序是在运行时调用 MFC 库，请选中 As a shared DLL 单选钮(默认设置)，这样可以被多个应用程序共享，而且可以减少应用程序的磁盘空间和内存空间；如果是在建立时与静态 MFC 库链接，请选中 As a statically linked 单选钮。使用动态链接库方式时在以后生成的可执行应用程序中并不真正包含 MFC 类库中的对象，而使用静态链接方式时，则把 MFC 库中的代码生成为应用程序的一部分，这时生成的应用程序也就相对大一些。

(8)单击 Next 按钮，弹出如图 1-9 所示的对话框，进入 AppWizard 的最后一个步骤，对话框中的提示信息指明了系统将要自动创建的类、头文件(.h)和实现文件(.cpp)的名称，以及派生出这些对象的 MFC 的基类等内容。在这一步当中，我们还可以对视图类的基类进行选择。如果要更改类名和文件名，请从列表框中选取类名，这时类名、基类名、头文件和实现文件将显示在列表框下面的编辑框中，用户可以根据需要进行修改。

(9)单击 Finish 按钮，弹出 New Project Information 对话框(如图 1-10)，显示应用程序的规范说明，即应用程序具有的特征。

(10)单击 OK 按钮，确认前面所做的选择都是正确的，AppWizard 根据这些选择自动

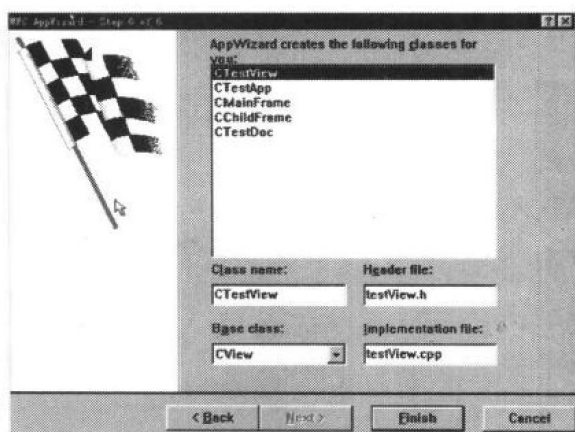


图 1-9 MFC AppWizard——Step 6 of 6 对话框

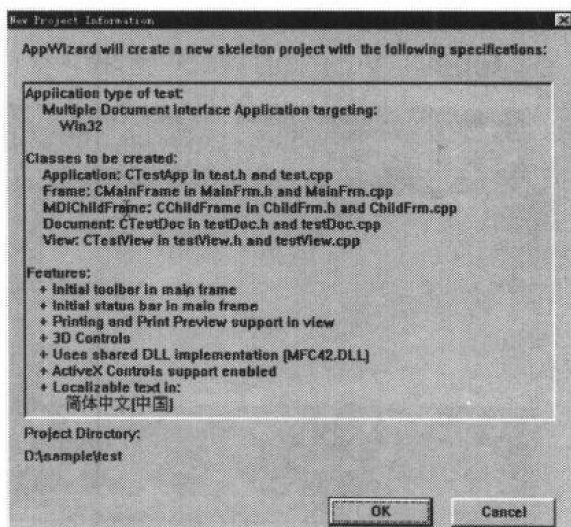


图 1-10 Test 工程文件的 New Project Information 对话框

生成应用程序的源文件。如果要重新更改某些选项,请单击 Cancel 按钮返回如图 1-7 所示的对话框。

(11)完成以上步骤后, AppWizard 为应用程序生成必要的框架文件并在工作区窗口打开生成的项目(如图 1-11)。其中,工作区窗口的 ClassView 面板显示的是 AppWizard 创建的类和成员函数,FileView 面板显示的是 AppWizard 创建的框架文件,ResourceView 面板显示的是 AppWizard 创建的资源。此外,AppWizard 还为每个项目建立 Win32 Debug 和 Win32 Release 两种配置。

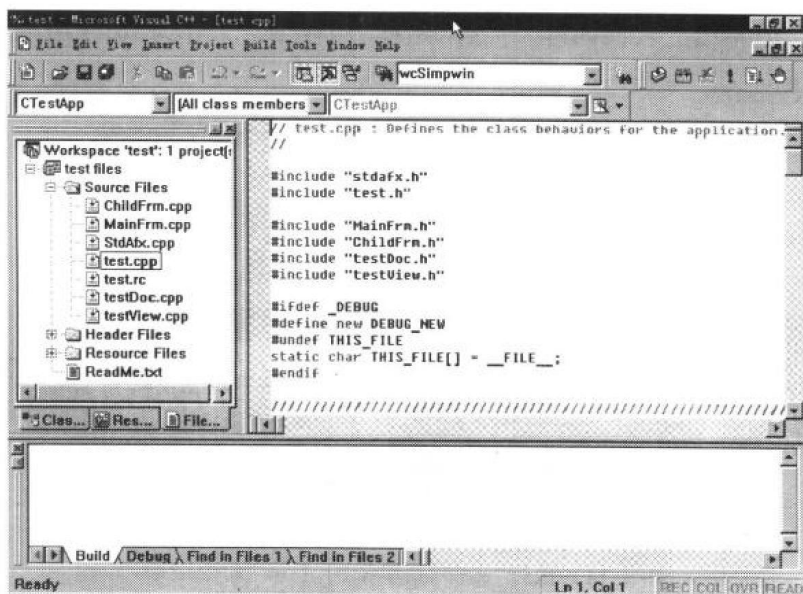


图 1-11 Microsoft Visual C++ 6.0 集成开发环境

(12)编译、链接、运行。点击 Build 菜单中的 Build Test.exe 或 F7 键进行编译、链接,若无错误,则生成 test.exe 文件。