



# Visual C++ MFC

## 棋牌类游戏

### 编程实例

葛 垚 主编  
雷超然 罗勋阳 吴裕豪 陈 其 编著



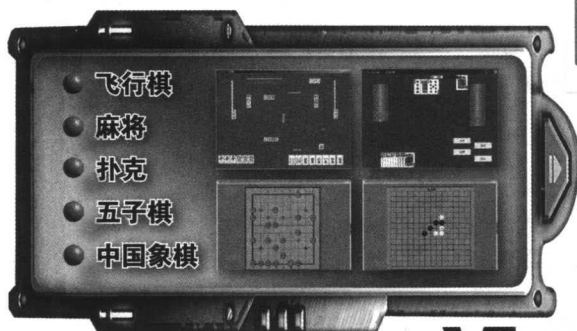
CD-ROM



人民邮电出版社  
POSTS & TELECOM PRESS

TP312/2872D

2008



# Visual C++ MFC 棋牌类游戏 编程实例

葛 垚 主编  
雷超然 罗勋阳 吴裕豪 陈 其 编著

人民邮电出版社  
北 京

## 图书在版编目 (CIP) 数据

Visual C++ MFC 棋牌类游戏编程实例 / 葛珪主编;  
雷超然等编著. —北京: 人民邮电出版社, 2008.7  
ISBN 978-7-115-17500-7

I. V… II. ①葛…②雷… III. C 语言-程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 004718 号

## 内 容 提 要

本书介绍了如何使用 Visual C++ 来设计各种棋牌类游戏, 主要针对想学习游戏设计的初学者。本书通过一些简单易懂的语句来讲述各种棋牌类游戏, 而一些实例更具体化了各种操作。本书循序渐进地讲解了游戏最核心的内容, 包括画面绘制、游戏动画、游戏消息处理、游戏声音处理、网络游戏处理和基本的人工智能思想等。各个游戏实例的安排也由浅入深; 从简单的单机双人游戏到复杂的多人的网络游戏, 对它们的制作过程都进行了详细的讲述。不但让读者在学习过程中深入了解了 Visual C++ 游戏制作的思想, 而且让读者有了自己的游戏作品。各个章节的内容有很紧密的联系, 但重点又有所不同。读者可以在学习后面章节的内容后回顾前面的程序, 从而进一步修改和优化, 使之成为一个非常完善的游戏系统。

本书适合游戏开发人员以及游戏爱好者学习与使用。

## Visual C++ MFC 棋牌类游戏编程实例

- ◆ 主 编 葛 珪
- ◆ 编 著 雷超然 罗勋阳 吴裕豪 陈 其  
责任编辑 陈 昇
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京艺辉印刷有限公司印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16  
印张: 26  
字数: 638 千字 2008 年 7 月第 1 版  
印数: 1—4 000 册 2008 年 7 月北京第 1 次印刷

ISBN 978-7-115-17500-7/TP

定价: 49.00 元 (附光盘)

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154



## 前 言

如今，在电脑如此普及的时代，设计游戏已经成为许多游戏爱好者的梦想。笔者相信有许多游戏爱好者凭着自己对游戏无比的热爱和丰富的想象力，想在游戏设计的领域里闯出一片天地，让自己脑海里的神奇世界在数字世界中得到实现。然而，更多的时候，许多人因为技术的原因，在程序实现上无能为力，以至于最后半途而废。

基于这个出发点，为了尽可能地让这些入能够重新燃起对游戏设计的热情，以及帮助更多的人实现自己游戏设计的梦想，本书将以各种层面的技术与范例，向读者介绍游戏设计的知识和技巧，邀请读者一同进入游戏设计这个神奇的领域，去实现那些以往认为不可能实现的梦想。

本书主要是为初学者入门和帮助有一定基础的人进一步提高来考虑的，通篇以实例贯穿，对游戏背景、算法设计以及游戏界面设计等方面进行详细的讲解。所以，本书非常适合游戏设计入门人员和想在一定基础上得到进一步提升的人员使用。

本书的内容以 Visual C++ 6.0 为基础进行编写，适合 6.0 以上的编程应用。随着 Windows XP、Windows Server 2003、Windows Vista 操作系统的发布以及 Visual Studio.NET 2003 在中国的同步上市，Microsoft 在整个软件市场上的主导地位日趋稳固，所以由 Microsoft 推出的 Visual 系列开发工具越来越受到程序员的青睐。但 Visual Studio 2003 过于庞大和复杂，对于初学者的入门并不适合。

而 Visual C++ 6.0 是一个非常稳定成熟的版本，并且该版本对系统的要求也比较低，开发的应用程序几乎可以满足各种类型项目的需求。因此我们决定以 Visual C++ 6.0 为基础进行编写。本书从介绍 Visual C++ 的基本环境开始，由浅入深，从简单的 MFC 控件使用，到 GDI 绘图以及复杂网络程序和算法设计，都进行了详细的说明。

在第 1 章中，我们将向大家介绍 Visual C++ 可视化集成开发环境，让读者有个初步的了解。在接下来的第 2 章，我们以棋牌类中相对简单的五子棋为例，初步展示如何进行位图粘贴、窗体设计等。在第 3 章的飞行棋制作中，我们除了让读者进一步熟悉之前所学习到的内容外，还将尝试实现同 Flash 的交互。在第 4 章、第 5 章的斗地主和象棋游戏的制作中，我们将初步接触到网络通信的问题和声音、动画问题。在第 6 章麻将的制作中，不仅综合使用了之前学习到的知识，同时更有升华。当然，随着难度的逐渐增加，涉及的算法也会越来越复杂，其中既有各种游戏的基本算法，也有各种响应的判断。

本书的特点如下：

- (1) 实例与原理并重，知识点全面；
- (2) 配合 Windows XP 程序来讲述 Visual C++，使书中的内容更加容易理解和实践；
- (3) 基本上覆盖了 Visual C++ 在小游戏设计中所要运用到的知识点，读者学完这本书后可以完成不同类型的小游戏制作；
- (4) 既适合初学者入门，也适合有一定基础的人进一步提高；
- (5) 以常见的游戏作为游戏范例，包括五子棋、斗地主等小游戏，并利用简单明了的程序代码，让读者可以轻松地学习游戏设计的技巧。

笔者深信这是一本非常实用的游戏设计宝典，最后，希望本书能够成为您进入游戏设计领域的敲门砖，帮助您实现游戏设计的梦想。

本书配套光盘的游戏试玩环节需要读者点击相应的游戏，并将程序下载到计算机硬盘才能运行。如果因为系统问题不能自动打开下载提示框的，请直接到光盘根目录下的 Games 文件夹中复制。在游戏互动环节中，光盘介绍了各种互动知识。由于互动是需要修改程序的，所以必须把互动程序复制到硬盘才能执行。方法是先选中需互动的游戏，然后点击页面右下角的“拷贝文件”按钮。程序会自动把需要的文件全部复制到“D:\VC 棋牌游戏开发实例”下，修改后的程序读者可以自己复制到其他地方。在点击“退出”按钮时会提示用户是否删除 D 盘下互动时复制的文件。如果读者想把文件复制到自定义位置，也可以把光盘根目录下的 downloads\code 文件夹中的对应源代码复制到需要的地方，然后手动运行文件即可。读者遇到疑难可以同我们一起进行探讨，联系邮箱为：chenqiok@163.com。

编 者



# 目 录

|                                  |    |
|----------------------------------|----|
| 第 1 章 Visual C++ 基础知识介绍 .....    | 1  |
| 1.1 Visual C++ 可视化集成开发环境 .....   | 1  |
| 1.1.1 Workspace .....            | 3  |
| 1.1.2 AppWizard .....            | 5  |
| 1.1.3 ClassWizard .....          | 11 |
| 1.1.4 WizardBar .....            | 15 |
| 1.1.5 Visual C++ 6.0 的视窗界面 ..... | 15 |
| 1.2 GDI 绘图 .....                 | 16 |
| 1.2.1 什么是 GDI .....              | 16 |
| 1.2.2 GDI 是如何实现输出的 .....         | 17 |
| 1.2.3 什么是有效矩形, 什么是无效矩形 .....     | 17 |
| 1.2.4 获取或释放设备描述表句柄以及相关操作 .....   | 17 |
| 1.3 对话框 .....                    | 26 |
| 1.3.1 对话框的基本概念 .....             | 26 |
| 1.3.2 控件的基本概念 .....              | 27 |
| 1.3.3 对话框模板的设计 .....             | 28 |
| 1.4 消息响应 .....                   | 32 |
| 1.4.1 Windows 消息概述 .....         | 32 |
| 1.4.2 MFC 消息处理 .....             | 34 |
| 1.4.3 MFC 窗口过程 .....             | 43 |
| 1.5 Visual C++ 游戏编程思想概述 .....    | 49 |
| 第 2 章 五子棋的制作与实现 .....            | 51 |
| 2.1 游戏规则与设计思路 .....              | 51 |
| 2.1.1 游戏的功能描述 .....              | 51 |
| 2.1.2 游戏的操作特性 .....              | 52 |
| 2.1.3 游戏设计思路 .....               | 52 |

|       |                             |     |
|-------|-----------------------------|-----|
| 2.1.4 | 热身运动 .....                  | 53  |
| 2.2   | 游戏界面制作 .....                | 54  |
| 2.2.1 | MFC 框架程序工程的创建以及内部结构简介 ..... | 54  |
| 2.2.2 | 显示器上的坐标系 .....              | 57  |
| 2.2.3 | 打造自己的窗体 .....               | 58  |
| 2.2.4 | 为窗体添砖加瓦 .....               | 60  |
| 2.2.5 | 消息响应 .....                  | 64  |
| 2.2.6 | 贴图与消息响应的实际运用 .....          | 66  |
| 2.3   | 核心算法的实现 .....               | 75  |
| 2.3.1 | 棋盘的数据结构分析 .....             | 75  |
| 2.3.2 | 棋盘的算法 .....                 | 76  |
| 2.4   | 游戏制作进阶 .....                | 92  |
| 2.4.1 | 鼠标跟踪的制作 .....               | 92  |
| 2.4.2 | “英雄榜”的制作 .....              | 94  |
| 2.5   | 互动与总结 .....                 | 100 |
| 第 3 章 | 飞行棋的制作与实现 .....             | 105 |
| 3.1   | 游戏规则与设计思路 .....             | 105 |
| 3.1.1 | 游戏的功能描述 .....               | 105 |
| 3.1.2 | 游戏的操作特性 .....               | 106 |
| 3.1.3 | 游戏设计思路 .....                | 106 |
| 3.1.4 | 热身运动 .....                  | 106 |
| 3.2   | 游戏界面制作 .....                | 107 |
| 3.2.1 | 界面制作的初期准备 .....             | 108 |
| 3.2.2 | 飞机的绘制 .....                 | 112 |
| 3.2.3 | 整体重绘和状态打印 .....             | 120 |
| 3.2.4 | LBUTTONDOWN 的串接 .....       | 122 |
| 3.3   | 核心算法的实现 .....               | 127 |
| 3.3.1 | 棋盘的数据结构分析 .....             | 127 |
| 3.3.2 | 棋盘的算法 .....                 | 128 |
| 3.4   | 游戏制作进阶 .....                | 144 |
| 3.4.1 | VC 与 Flash 的交互 .....        | 144 |
| 3.4.2 | 计算机 AI 制作 .....             | 150 |
| 3.4.3 | 走棋动画的实现 .....               | 156 |
| 3.5   | 网络连接与实现 .....               | 159 |
| 3.5.1 | 网络实现的方法 .....               | 159 |
| 3.5.2 | 简易聊天室的实现 .....              | 160 |
| 3.5.3 | 网络设置 .....                  | 171 |
| 3.6   | 互动与总结 .....                 | 172 |

|                        |     |
|------------------------|-----|
| <b>第4章 扑克类游戏的制作与实现</b> | 175 |
| 4.1 游戏规则与设计思路          | 175 |
| 4.1.1 游戏的功能描述          | 175 |
| 4.1.2 游戏的操作特性          | 176 |
| 4.1.3 游戏设计思路           | 176 |
| 4.1.4 热身运动             | 177 |
| 4.2 核心算法的实现            | 179 |
| 4.2.1 Managers 的实现     | 179 |
| 4.2.2 连网打牌的实现          | 194 |
| 4.3 游戏界面制作             | 210 |
| 4.3.1 界面制作的初期准备        | 210 |
| 4.3.2 Button 控件的添加     | 215 |
| 4.3.3 相关的绘图            | 220 |
| 4.3.4 LBUTTONDOWN 事件响应 | 227 |
| 4.4 游戏制作进阶——发牌动画的制作    | 229 |
| 4.5 互动与总结              | 235 |
| <b>第5章 中国象棋的制作与实现</b>  | 241 |
| 5.1 游戏规则与设计思路          | 241 |
| 5.1.1 游戏的功能描述          | 241 |
| 5.1.2 游戏的操作特性          | 242 |
| 5.1.3 游戏设计思路           | 242 |
| 5.1.4 热身运动             | 243 |
| 5.2 核心算法的实现            | 245 |
| 5.2.1 棋盘的数据结构分析        | 245 |
| 5.2.2 棋盘的算法            | 246 |
| 5.2.3 网络               | 260 |
| 5.3 游戏界面制作             | 268 |
| 5.3.1 界面制作的初期准备        | 268 |
| 5.3.2 棋盘的绘制            | 272 |
| 5.3.3 游戏模式选择与按钮功能实现    | 277 |
| 5.3.4 整体重画             | 282 |
| 5.4 游戏制作进阶——为游戏添加声音    | 287 |
| 5.5 互动与总结              | 295 |
| <b>第6章 麻将的制作与实现</b>    | 299 |
| 6.1 游戏规则与设计思路          | 299 |
| 6.1.1 游戏的功能描述          | 299 |

|       |                         |     |
|-------|-------------------------|-----|
| 6.1.2 | 游戏的操作特性 .....           | 300 |
| 6.1.3 | 游戏设计思路 .....            | 300 |
| 6.1.4 | 热身运动 .....              | 301 |
| 6.2   | 核心算法的实现 .....           | 303 |
| 6.2.1 | 麻将的数据结构分析 .....         | 303 |
| 6.2.2 | 麻将 Manager 类设计 .....    | 304 |
| 6.2.3 | 麻将 NetControl 类设计 ..... | 329 |
| 6.3   | 游戏界面制作 .....            | 345 |
| 6.3.1 | 界面制作的初期准备 .....         | 346 |
| 6.3.2 | 麻将的绘制 .....             | 351 |
| 6.3.3 | 麻将的选中和出牌 .....          | 361 |
| 6.3.4 | 整体重画 .....              | 364 |
| 6.4   | 声音的制作 .....             | 368 |
| 6.4.1 | 播放 MP3 格式的背景音乐 .....    | 369 |
| 6.4.2 | 为游戏添加综合音效 .....         | 374 |
| 6.5   | 对话框深入 .....             | 383 |
| 6.5.1 | 对话框和对话框类 CDialog .....  | 383 |
| 6.5.2 | 对话框的 MFC 实现 .....       | 384 |
| 6.5.3 | 数据交换 .....              | 400 |
| 6.6   | 互动与总结 .....             | 406 |

## 第 1 章 Visual C++ 基础知识介绍

### 1.1 Visual C++ 可视化集成开发环境

Visual C++ 是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后, 随着其新版本的不断问世, Visual C++ 已成为专业程序员进行软件开发的首选工具。虽然 Microsoft 公司推出了 Visual C++.NET (Visual C++ 7.0), 但它的应用有很大的局限性, 它只适用于 Windows 2000, Windows XP 和 Windows NT 4.0 平台。所以实际上, 更多还是以 Visual C++ 6.0 为平台。

Visual C++ 6.0 不仅是一个 C++ 编译器, 而且是一个基于 Windows 操作系统的可视化集成开发环境 (Integrated Development Environment, IDE)。Visual C++ 6.0 由许多组件组成, 包括编辑器、调试器以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。Developer Studio 是一个通用的应用程序集成开发环境, 它不仅支持 Visual C++, 还支持 Visual Basic、Visual J++ 和 Visual InterDev 等 Microsoft 系列开发工具。Developer Studio 包含了文本编辑器、资源编辑器、工程编译工具、增量连接器、源代码浏览器、集成调试工具, 以及一套联机文档。使用 Developer Studio, 可以完成创建、调试、修改应用程序等各种操作。

Developer Studio 采用标准的多窗口 Windows 用户界面, 并增加了一些新特性, 使得开发环境更易于使用, 用户很容易掌握它的使用方法。图 1-1-1 是一个典型的 Developer Studio 用户界面。

Visual C++ 的集成开发环境 Developer Studio 提供了大量的实用工具以支持可视化编程特性, 包括 Workspace、AppWizard、ClassWizard 和 WizardBar 等。下面几节将详细介绍 Developer Studio 的各个特性。



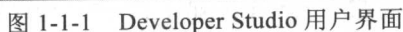
#### 小资料

Visual C++ 可以分成 3 个主要的部分。

1. Developer Studio。这是一个集成开发环境, 我们日常工作的 99% 都是在它上面完成的, 再加上窗口的标题栏上赫然写着 “Microsoft Visual C++”, 所以很多人理所当然地认为它就是 Visual C++ 了。其实不然, 虽然 Developer Studio 提供了一个很好的编辑器和很多 Wizard, 但实

2. MFC。从理论上讲, MFC 也不是专用于 Visual C++, Borland C++, C++Builder 和 Symantec C++ 同样可以处理 MFC。同时, 用 Visual C++ 编写代码也并不意味着一定要用 MFC, 只要愿意, 用 Visual C++ 来编写 SDK 程序, 或者使用 STL 和 ATL, 一样没有限制。不过, Visual C++ 本来就是为 MFC 打造的, Visual C++ 中的许多特征和语言扩展也是为 MFC 而设计的, 所以用 Visual C++ 而不用 MFC 就等于抛弃了 Visual C++ 中很大的一部分功能。但是, Visual C++ 也不等于 MFC。

2013年12月15日 星期一 12:00



### 1.1.1 Workspace

Workspace（项目工作区）是 Developer Studio 最重要的一个组成部分，程序员的大部分工作都在 Developer Studio 中完成。Developer Studio 使用项目工作区来组织项目、元素以及项目信息在屏幕上出现的方式。在一个项目工作区中，可以处理一个工程和它所包含的文件、一个工程的子工程、多个相互独立的工程以及多个相互依赖的工程。

Workspace 以窗口方式组织项目、文件和项目设置。Workspace 窗口一般位于屏幕左侧，如图 1-1-2。

Workspace 可以使用 View->Workspace(Alt+0)打开，如图 1-1-3 所示。

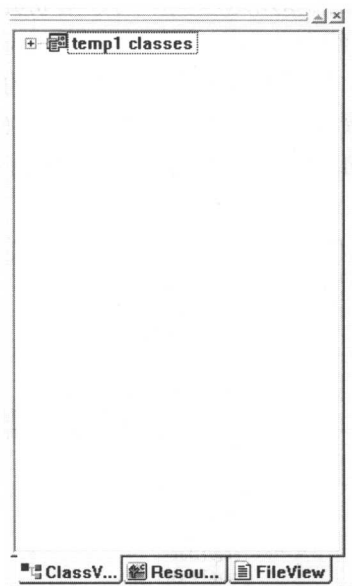


图 1-1-2

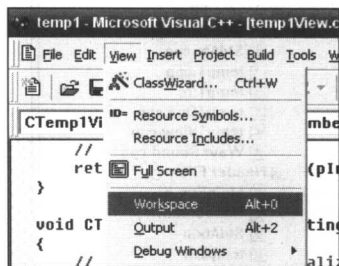


图 1-1-3

每个项目视图都有一个相应的文件夹，包含了关于该项目的各种元素。展开该文件夹可以显示该视图方式下工作区的详细信息。Workspace 包含 3 种视图。

- FileView（文件视图）：显示所创建的工程，展开文件夹可以查看工程中所包含的文件。
- ClassView（类视图）：显示项目中定义的 C++类，展开文件夹显示工程中所定义的所有类，展开类可察看类的数据成员和成员函数以及全局变量、函数和类型定义。
- ResourceView（资源视图）：显示项目中所包含的资源文件，展开文件夹可显示所有的资源类型。

单击项目工作区底部的标签可以从一个视图切换到另一个视图。每个视图都是按层次方式组织的。可以展开文件夹和其中的项察看其内容，或折叠起来察看其组织结构。在项目视图中，如果某一项不可以再展开，那么它就是可编辑的。双击这一项便可以打开相应的文档编辑器进行编辑。对类和源程序文件来说，是打开文本编辑器；对于对话框来说，是打开对话框编辑器。每个视图还支持右键快捷菜单。下面将介绍每种视图的功能和使用方法。

## 1. FileView (文件视图)

FileView 窗格如图 1-1-4 所示, 显示了工程文件和项目工作区中所包含的文件的逻辑关系。一个工作区可以包含多个工程, 其中活动工程以黑体显示。活动配置决定了编译活动工程时的编译选项。活动工程是使用 Build 或 Rebuild All 时要编译的那一个工程。可以用 Build 菜单上的 Set Active Configuration 选择不同的活动配置, 也可以在 Project 菜单上用 Set Active Project 选择不同的活动工程。

使用 FileView 可以察看文件, 双击选中的文件名字即可打开对应的文件, 还可以往工程中添加文件。要想从 Workspace 中移除某个工程或者某个工程中的文件, 可以直接选中该工程或文件, 然后按 Delete 键即可。注意, 这样并不会删除磁盘上的文件。

## 2. ClassView (类视图)

ClassView 窗格如图 1-1-5 所示, 显示所有已定义的类以及这些类中的数据成员和成员变量。Visual C++ 自动从项目工作区中所包含的源程序文件中分离出类。在 ClassView 中, 文件夹代表工程文件名。展开 ClassView 顶层的文件夹后, 显示工程中所包含的所有的类。双击一个类的图标 (或单击图标旁的+号) 时, ClassView 展开该类并显示其类成员。双击类的名字将打开定义类的头文件, 双击函数名字将打开对应的 CPP 文件并直接跳到函数定义处, 双击变量名字可跳到该变量的声明处。

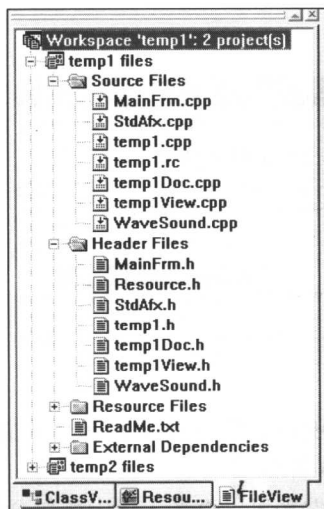


图 1-1-4 FileView

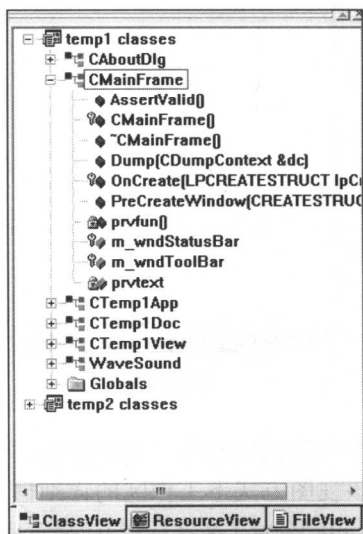


图 1-1-5 ClassView

ClassView 使用图标来标识类、类成员和工程中的其他项。表 1-1 显示了所有的图标及其含义。

表 1-1 类视图中的图标

| 图 标 | 含 义     |
|-----|---------|
|     | 工程      |
|     | 类       |
|     | 公有类成员函数 |

续表

| 图 标                                                                               | 含 义     |
|-----------------------------------------------------------------------------------|---------|
|  | 保护类成员函数 |
|  | 私有类成员函数 |
|  | 公有类成员变量 |
|  | 保护类成员变量 |
|  | 私有类成员变量 |
|  | 文件夹     |

### 3. ResourceView (资源视图)

ResourceView 窗格如图 1-1-6 所示, 包含一个工程中所遇到的各种资源, 如对话框、工具条、图片、声音等资源。双击对应项可以打开对应资源的编辑窗口。

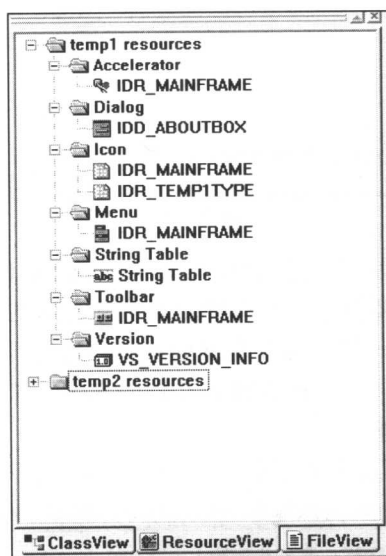


图 1-1-6 ResourceView

## 1.1.2 AppWizard

AppWizard (应用程序向导) 是一个标准的 C++ 源代码生成器。它通过一系列的对话框来提示用户输入所需创建的程序的信息, 如它的名称和位置。用户还可以指定它是否具有某些特性, 如多文档接口或工具条、对数据库和 OLE 的支持等。然后 AppWizard 生成一些文件, 这些文件构成了程序的框架。由 AppWizard 生成的程序是一个基本的 Windows 程序, 用户可以编译并运行——它实际上什么也不做, 只是准备好增加那些为程序提供功能性的资源和代码。这样就节省了用户设计应用程序框架的时间和精力, 用户所要做的工作只是直接向框架中添加自己的处理代码。

AppWizard 是用类别名称和源程序名产生 Windows 应用程序的一个工作向导。当想要制作一些程序的时候, 便会用到 AppWizard 并能了解到它用途的广泛性。不要把 AppWizard 与一个应用

程序所产生的所有程序源码与产生器所生成的源程序弄混。AppWizard 生成的源码是在功能性应用程序框架里基础类别的最低限度要求。AppWizard 将帮助你很快地开始一个新的应用程序。

AppWizard 是 Visual C++ 提供的一个高级编程工具，它可以产生应用的 C++ 源代码框架。通过与另一个工具 ClassWizard 一起配合使用，可大大节省开发应用程序的时间和精力。下面以使用 AppWizard 生成一个 MFC 单文本工程为例来介绍 AppWizard 的使用。

首先，选择菜单 File->New，打开文件属性页（如图 1-1-7 所示）。

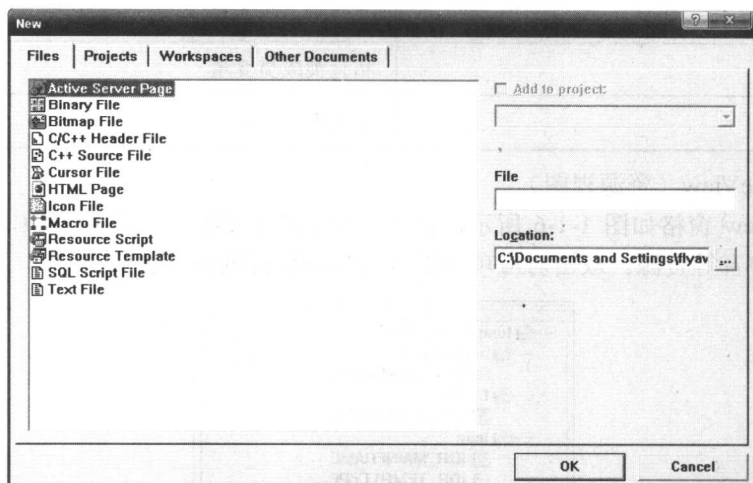


图 1-1-7

文件类型虽然很多，但常用的只有两种：C/C++ Header File（C/C++头文件）和 C/C++ Source File（C/C++源文件）。要新建一个文件先要选择合适的文件类型，如果想把新文件加入到工程内，选中 Add to project 复选框并在下面选中合适的工程。如果当前没有打开任何工程，则 Add to project 将不可用。在 File 栏输入文件的名字，在 Location 栏输入文件的存放地址，单击 OK 即完成文件的创建。

接着单击 Projects 项，出现工程页属性（如图 1-1-8 所示）。

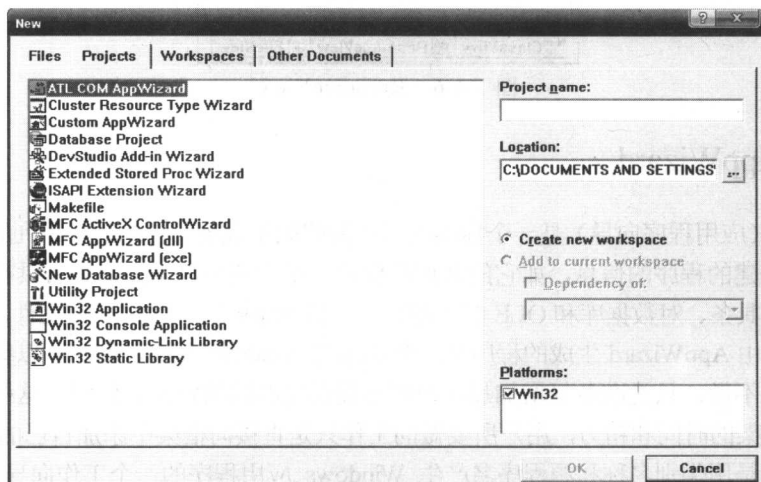


图 1-1-8

常见工程 (Project) 类型如下。

- ATL COM AppWizard: ATL COM 应用程序向导。
- MFC ActiveX ControlWizard: MFC ActiveX 控件向导。
- MFC AppWizard (dll): MFC 应用程序向导 (动态库)。
- MFC AppWizard (exe): MFC 应用程序向导 (可执行文件)。
- Win32 Application: Win32 应用程序。
- Win32 Console Application: Win32 控制台程序

选择 MFC AppWizard(exe), 在 Project name 输入工程名, Location 是工程存放的地址, 向导将自动生成一个以工程名为名字的文件夹用于存放整个工程的文件。单击 OK 进入下一项 (如图 1-1-9 所示)。

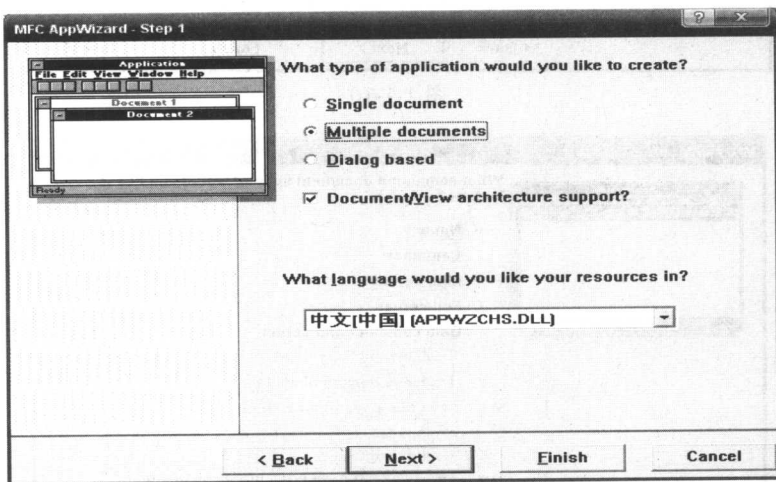


图 1-1-9

这一步是选择生成何种类型的应用程序。

- Single document: 单文档。
- Multiple documents: 多文档。
- Dialog base: 基于对话框。
- Document/View architecture support: 生成文档视图结构程序。
- What language would you like your resources in: 资源中使用的语言。

选择 Single document, 单击 Next 按钮进入下一步 (如图 1-1-10 所示)。

这里是选择数据库的支持情况。

- None: 不支持数据库。
- Header files only: 只包含头文件。
- Database view without file support: 没有文件支持的数据库视图。
- Database view with file support: 带文件支持的数据库视图。
- If you include a database view, you must select a data source: 用户如果选择包含数据库视图, 则必须选择一个数据源。

选择默认属性, 单击 Next 按钮进入下一步 (如图 1-1-11 所示)。

选择默认属性, 单击 Next 按钮进入下一步 (如图 1-1-11 所示)。

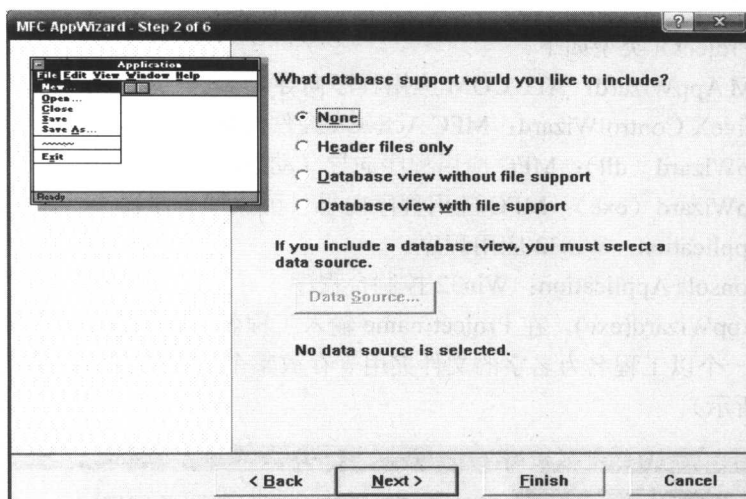


图 1-1-10

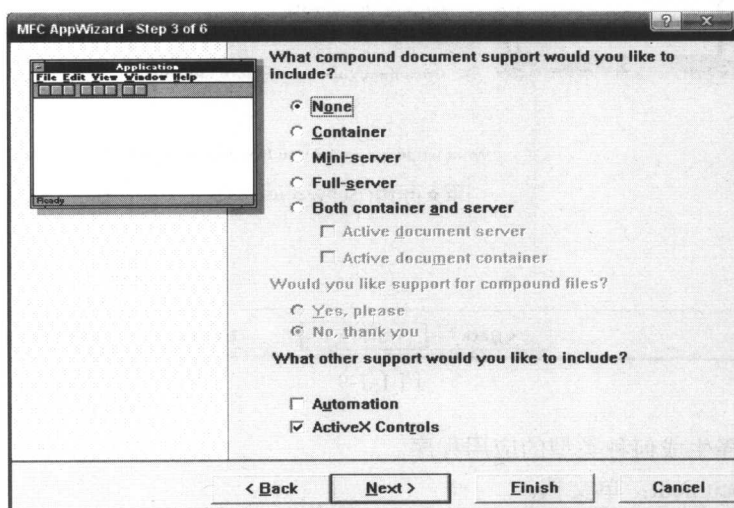


图 1-1-11

这一步是选择程序对复合文档的支持情况。

- None: 不支持 OLE 复合文档。
- Container: 容器。
- Mini-server: 应用程序能够创建和管理复合文档对象, Mini-server 程序不能单独运行, 只支持嵌入对象。
- Full-server: 应用程序能够创建和管理复合文档对象, Full-server 程序能够单独运行, 并支持链接和嵌入的对象。
- Both container and server: 容器和服务。
- Would you like support for compound files: 是否希望支持复合文档。
- What other support would you like to include: 希望包含的其他支持。
- Automation: 自动化。