多媒体篇

Visual C++ 6.0

高级编程技术

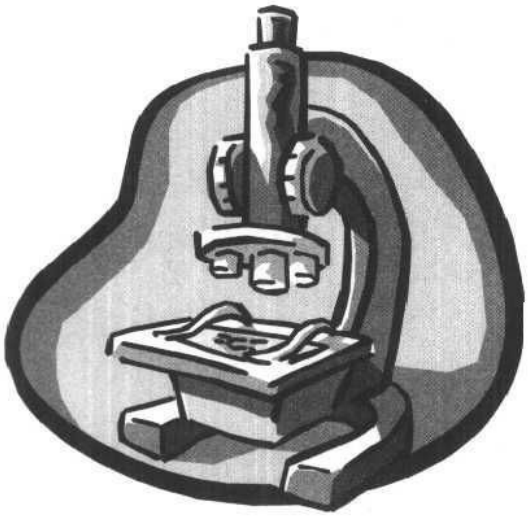
乔林 杨志刚 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Visual C++ 6.0高级编程技术

——多媒体篇

乔林　杨志刚等　编著



中国铁道出版社

2000·北京

（京）新登字 063 号

内 容 简 介

本书讨论如何用 Visual c++6 0 进行多媒体编程。以使用类 CJuneGlyph 和扩展至类 CJuneGlyphWorkex 进行图像基本变换、图像的点群运算，以观图出发讨论高级图像处理技术。本书创建的程序可以处理大多数图像文件格式，进行多种标准图像处理和播放多媒体文件，如 CD 音频、MDI 序列、WAV 和 AVI 文件等。

全书结合实例进行讨论，有助于读者能尽快掌握实践的方法。

图书在版编目（CIP）数据

Visual C++6.0 高级编程技术. 多媒体篇/乔林等编. －北京：中国铁道出版社，2000.1

ISBN 7－113－03603－1

Ⅰ.V… Ⅱ 乔… Ⅲ.C 语言-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(1999)第 55248 号

书　　名：Visual c++6 0 高级编程技术——多媒体篇
作　　者：乔林　杨志刚
出版发行：中国铁道出版社（100054,北京市宣武区右安门西街 8 号）
策划编辑：严晓舟
责任编辑：周长青
特邀编辑：郭晓霞
封面设计：新创工作室　冯龙彬
印　　刷：北京市兴顺印刷厂
开　　本：787×1092　1/16　印张：25 75　字数：623 千
版　　本：2000 年 1 月第 1 版　2000 年 1 月第 1 次印刷
印　　数：1～5000 册
书　　号：ISBN 7－113－03603－1/TP·418
定　　价：41 00 元

前　言

Visual C++ 6.0 是一个功能强大的可视化编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的可重用的面向对象 C++ 程序语言。

一个 Visual C++ 6.0 的程序首先是应用程序框架，而这一框架正是应用程序的“骨架”。应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。Visual C++ 6.0 已经为读者做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，我们所需要做的，只是在程序中加入完成所需功能的代码。

本丛书共分五个专题，涵盖了 Visual C++ 6.0 编程的方方面面：

《Visual C++ 6.0 高级编程技术——MFC 与多线程篇》：本书讨论如何使用 MFC 类库和 Visual C++ 6.0 的多线程技术，与泛泛地讨论如何使用 MFC 类库相比，我们将着眼点集中在如何扩展 MFC 类库、如何使用 MFC 的高级技术开发专业化的应用程序上。本书创建的实例可以极大地改进应用程序的外观。

《Visual C++ 6.0 高级编程技术——多媒体篇》：本书使用一个实例讨论如何使用 Visual C++ 6.0 进行多媒体编程。本书创建的程序可以处理大多数图像文件格式，进行多种标准图像处理，播放多媒体文件（CD 音频、MIDI 序列、WAV 和 AVI 文件等）。

《Visual C++ 6.0 高级编程技术——DirectX 与 OpenGL 篇》：本书讨论如何使用 Visual C++ 6.0 的 DirectX 函数和 OpenGL 库进行动画编程。通过学习本书，读者将能够对 DirectX 与 OpenGL 编程的基本问题和关键技巧有个透彻的了解。

《Visual C++ 6.0 高级编程技术——ActiveX 与 COM 篇》：本书讨论如何使用 Visual C++ 6.0 进行 ActiveX 与 COM 编程。ActiveX 与 COM 技术作为 Microsoft 的关键技术之一，在 Windows 操作系统中使用广泛。通过学习本书，读者将能够对 ActiveX 与 COM 编程的基本问题和关键技巧有个透彻的了解。

《Visual C++ 6.0 高级编程技术——Internet 与 Web 篇》：本书讨论如何使用 Visual C++ 6.0 进行 Internet 编程。本书分成两个部分。第一部分着重介绍如何使用 Visual C++ 6.0 进行客户端编程，这部分内容是我们在 Internet 上冲浪时会频繁遇到的。而第二部分则详细讨论如何使用 Visual C++ 6.0 开发 Web 服务器应用程序。

本套丛书是集体劳动的结晶，参阅本套丛书编写工作的有乔林、杨志刚、魏志强、王仲华、何隽、林杜、费广正、金传恩、王诚铭、陈爱华、汪巨涛等。

本丛书的对象是 Visual C++ 6.0 的中级和高级读者。我们希望读者在阅读本书的过程中能够上机实践。每学完一个例子，尝试着改变一点点，或者添加一点东西，并改变一些代码将帮助读者体验进步和成功的乐趣。

编　者

1999 年 10 月

目　录

第 1 章

MFC 程序体系结构

本书之所以能够成文是因为笔者在网上冲浪时偶然发现了一个极其有用的 DLL——“ImageLoad.dll”，该动态链接库提供了 20 来个函数访问六种主要格式的图像文件。正巧笔者正在编写一个多媒体应用程序，且正为阅读 JPEG 源码搞得睡不安寝，我想读者肯定能够了解笔者当时发现“ImageLoad.dll”的心情，正是“漫卷诗书喜欲狂”也！

与市面上几乎全部的介绍 Visual C++ 6.0 编程的书不同，本书将以真正唯一的一个的多媒体应用程序的开发过程为主线，这意味着读者将跟随笔者一步一步地实现整个应用程序。

本章将讨论 MFC 应用程序的基本体系结构。我们首先讨论如何使用 MFC AppWizard 创建应用程序的基本框架；然后讨论应用程序基本框架是如何构成的，以及基于 MFC 类库的应用程序又是如何执行的。

本来笔者没有计划将本章和第 2 章列入本书，但为照顾中级以下的读者，也为了揭示应用程序开发过程的完整性，我们还是专门辟出两个章节讨论之。除了本章第一节，高级读者可以跳过本章其他小节和第 2 章。

1.1 观图程序“PhotoSee”（一）：使用 MFC AppWizard

一般而言，要创建基于 MFC 类库的 Windows 应用程序，使用 AppWizard 是最方便的方法。AppWizard 是一种步进式的辅助工具，在使用过程中 AppWizard 将显示一系列对话框，用户可以在对话框中选择满足应用程序和项目需要的选项，AppWizard 自动生成创建应用程序所必须的与 ClassWizard 兼容的文件。

使用 AppWizard 生成新项目和应用程序的步骤如下：

- 启动 Visual C++ 6.0。
- 从“File”菜单选择“New”命令（或按 Ctrl+N 组合键），系统弹出“New”对话框。
- 选择“Project”选项卡，在项目类型列表框中选择要创建的项目类型。我们这里选择“MFC AppWizard(exe)”。
- 在“Project Name”文本框输入项目名称“PhotoSee”，如图 1-1 所示。MFC AppWizard 根据所输入的名称派生相应的初始文件和类的缺省名，并在根目录下创建以该名字命名的子目录。
- 在“Location”文本框中输入用于存放项目的根目录。
- 如果要添加新项目到已有的工作区中，请选择“Add to current workspace”选

项，否则系统将自动创建包含新项目的新工作区。如果要使新项目成为已有项目的子项目，请选中“Dependency of”复选框并指定项目名。

- 从“Platforms”列表框选择可用的目标平台，缺省平台是 Win32。在创建项目时，系统缺省自动创建两种项目配置，即 Win32 Debug 和 Win32 Release。Win32 Debug 配置包含调试信息和未优化的设置，win32 Release 配置不包含调试信息并可以选择优化设置。

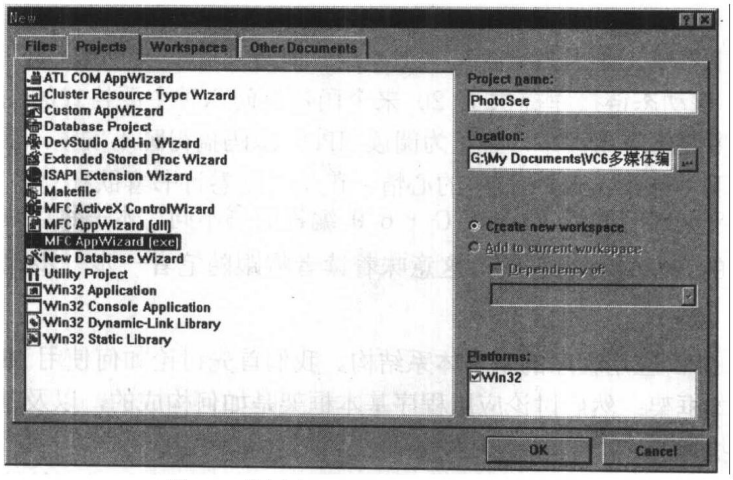


图 1-1 使用“MFC AppWizard”创建新项目

- 单击“OK”按钮，弹出如图 1-2 所示的对话框，从中确定应用程序的结构并为资源文本选择一种语言。一般而言，MFC AppWizard 可以创建以下三种类型的应用程序：
  - Single document（单文档界面，SDI）：一次只允许打开一个文档边框窗口；
  - Multiple documents（多文档界面，MDI）：允许在应用程序的同一实例中打开多个文档边框窗口；
  - Dialog-Based（基于对话框）：显示一个简单的对话框让用户输入数据。

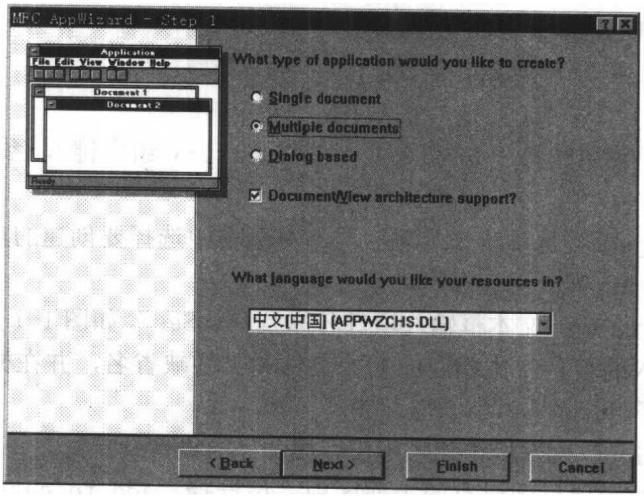


图 1-2 选择应用程序的结构和资源语言

☞ 我们这里选择“Multiple documents”选项，然后单击“Next”按钮，系统弹出如图 1-3 所示的对话框，从中可以选择数据库支持功能。如果应用程序为单文档或多文档界面，则可以选择以下数据库支持选项：

- None：该选项不支持任何 ODBC 库。如果应用程序不使用数据库，则选择此选项将创建较小的应用程序。
- Header files only：此选项提供最低限度的数据库支持，包含所有的数据库头文件和链接库，但 MFC AppWizard 不创建任何与数据库相关的类，用户必须手工创建。

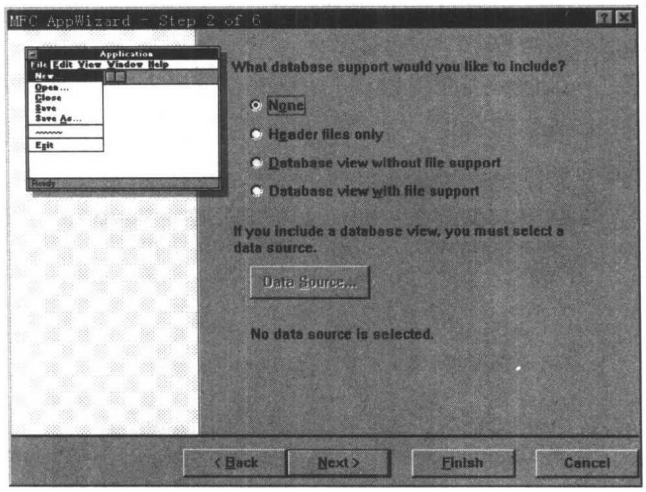


图 1-3　选择数据库支持选项

- Database view without file support：此选项包含所有的数据库头文件和链接库，并自动创建记录视图（record view）和记录集（recordset）。使用此选项，应用程序具有文档支持但不支持串行化（serialization）。
- Database view with file support：与前一选项的不同之处也支持串行化。

如果应用程序包含数据库视图，则应定义数据源。单击“Data Source”按钮，系统将弹出“Database Options”对话框，从中可以确定应用程序的数据源（如 ODBC 或 DAO）和记录集类型（如 Snapshot，Dynaset 或 Table）。

☞ 我们这里选择“None”选项。然后单击“Next”按钮，系统弹出如图 1-4 所示的对话框，从中可以选择是否支持复合文档（Compound document）。MFC AppWizard 可以为不同的 ActiveX 控件容器生成相应的支持代码。选择任一选项都会使标准 ActiveX 资源有效并将额外的自动化命令添加到菜单栏中。如果应用程序为单文档或多文档界面，则可以选择以下复合文档支持选项：

- None：缺省设置，应用程序不带 ActiveX 支持。
- Container：应用程序包含被链接和嵌入的对象。
- Mini-server：应用程序只可以创建嵌入对象。
- Full-server：应用程序既可独立运行又能支持 ActiveX，而且所创建的对象还可以包含在复合文档（Compound Documents）中。

- Both container and server：应用程序既可作容器又可作服务器。
- Yes，please：使用复合文档格式来串行化容器应用程序的文档。可以将包含 ActiveX 对象的文档保存为一个文件，但该文件可以被单独的 ActiveX 对象文件访问。
- No，thank you：不使用复合文档格式来串行化容器应用程序的文档。该选项必须一次将包含 ActiveX 对象的整个文件装入到内存中，不允许增量式保存单独的 ActiveX 对象，如果一个 ActiveX 对象被更改并加以保存，则文件中所有的 ActiveX 对象都必须保存。
- Automation：应用程序支持自动化，这样应用程序就可以被其他自动化客户（如 Microsoft Word）访问。
- ActiveX Controls：应用程序使用 ActiveX 控件。如果不选择此项，要向项目插入 ActiveX 控件时，就必须在应用程序的 InitInstance 成员函数中添加对 AfxEnableControlContainer 函数的调用。

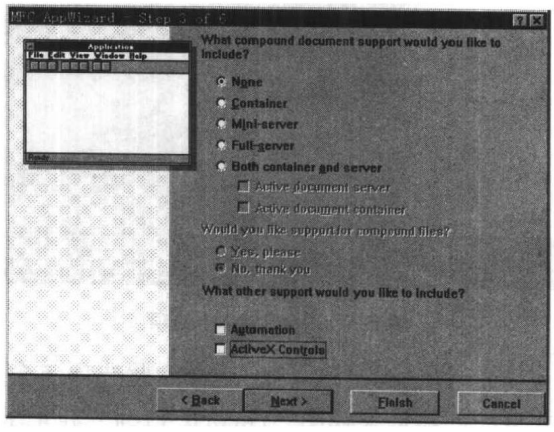


图 1-4 选择复合文档支持选项

☛ 我们这里选择“None”选项且不支持 ActiveX 控件。然后，单击“Next”按钮，系统弹出如图 1-5 所示的对话框，从中可以选择不同的用户界面选项。我们这里使用缺省设置。

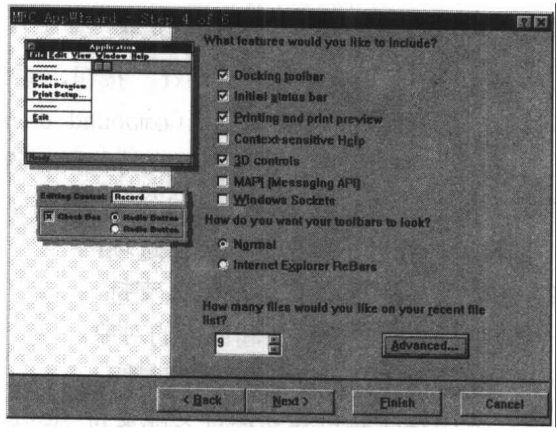


图 1-5 指定应用程序的基本特征

- 单击“Advanced...”按钮，系统弹出如图 1-6 所示的对话框，在“Document Template Strings”选项卡中设置文档类型名和缺省新建文件名。
- 单击“Window Styles”选项卡，系统弹出如图 1-7 所示的对话框，设置窗口样式。

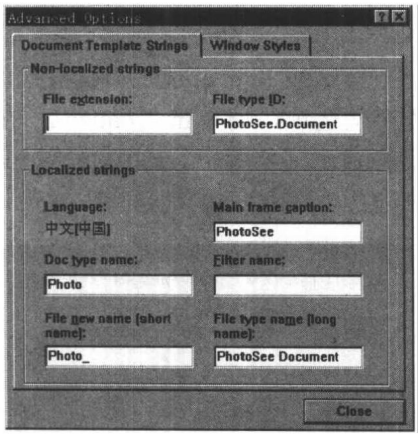


图 1-6　高级选项设置之一

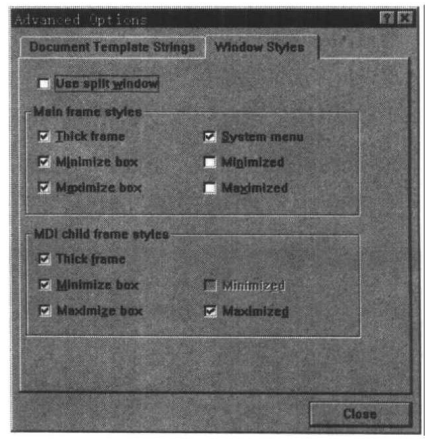


图 1-7　高级选项设置之二

- 单击“Close”按钮和“Next”按钮，系统弹出如图 1-8 所示的对话框，从中可以选择以下项目选项设置：
  - What style project would you like? 需要什么样式的工程。我们可以选择“MFC Standard”（标准 MFC 样式的工程）或“Windows Explorer”（Windows 资源管理器样式的工程）。我们选择“MFC Standard”。
  - Would you like to generate source file Comments? 是否需要 MFC AppWizard 在源文件中插入注释。所插入的注释将指示用户在何处添加自己的代码。
  - How would you like to use the MFC library? MFC 类库既可以作为静态链接库也可以作为动态链接库。如果应用程序既有 MFC 代码也有非 MFC 代码，则应该使用 MFC 静态链接库。如果所有模块都只使用 MFC 类库，则应该使用动态链接库以减少磁盘和内存空间开销。我们使用静态链接库。

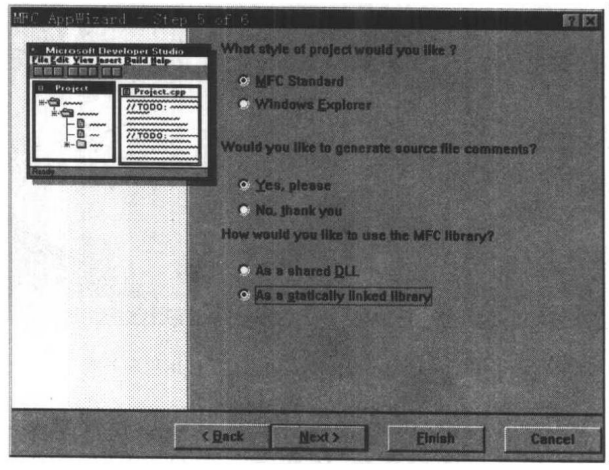


图 1-8　选择项目选项设置

- 单击“Next”按钮，系统弹出如图 1-9 所示的对话框，显示 MFC AppWizard 将创建的类、头文件和实现文件的名称。我们将类 CPhotoSeeView 的基类改变为 CScrollView（这将使得窗口可以滚动），接受其他缺省设置。
- 单击“Finish”按钮，系统弹出如图 1-10 所示的“New Project Information”对话框，显示应用程序所具有的特征。
- 单击“OK”按钮，确认前面所做的选择都是正确的，MFC AppWizard 根据这些选择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”按钮以关闭“New Project Information”对话框。完成上述步骤后，MFC AppWizard 自动生成应用程序的初始文件，并在项目工作区窗口打开生成的项目。

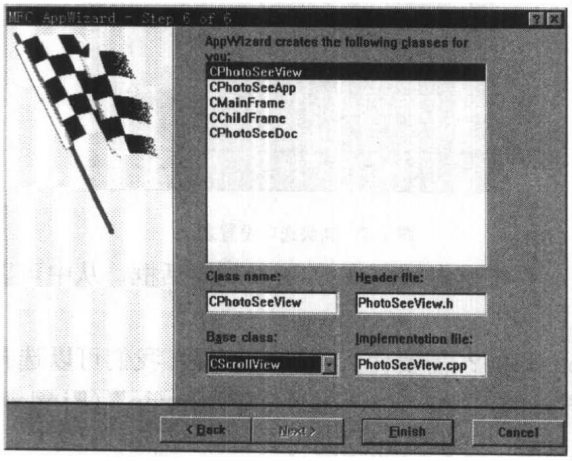


图 1-9 显示 MFC AppWizard 将要创建的类名

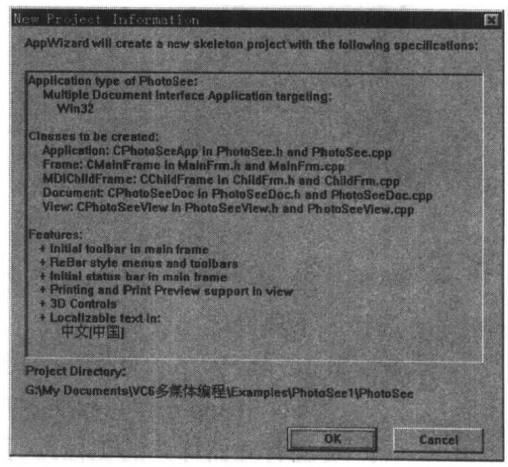


图 1-10 显示 MFC AppWizard 将要创建的文件名称

MFC AppWizard 生成的初始项目本身就是一个完整的应用程序，只不过该程序并不具备我们所需要的功能而已。从“Build”菜单中选择“Execute PhotoSee.exe”菜单选项，程序的运行结果如图 1-11 所示。

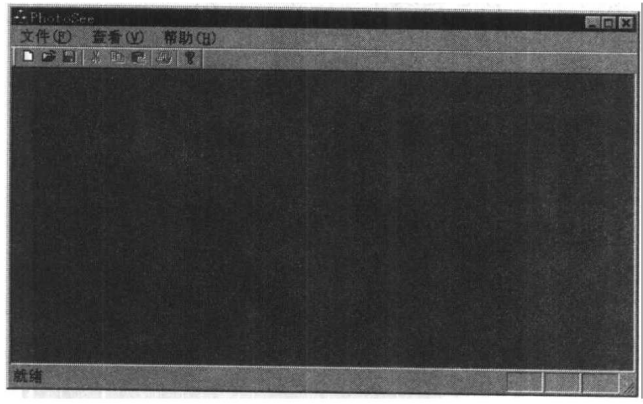


图 1-11 MFC AppWizard 自动创建的应用程序外观

从上图可以看出，MFC AppWizard 创建的 PhotoSee 包含标题栏、工具栏、状态栏和菜单栏。作为一个 MDI 应用程序，PhotoSee 还包含一个打开的文档子窗口。如果实际操作一下上述程序，我们还可以发现：

- 可以用“File”菜单的“New”命令打开新的文档子窗口。可以使用“Window”菜单上的命令重排打开的多个窗口。
- “Edit”菜单命令的代码未实现。
- “File”菜单上的“Open”、“Save”和“Save As”命令已经部分实现。选择“Open”命令将显示“Open”对话框让用户选择文件，被选择的文件名将显示在文档子窗口的标题栏中，但不读进任何内容。同样，选择“Save”或“Save As”命令将显示“Save As”对话框，但不真正向文件写任何内容。
- “File”菜单的“Close”、“Print”、“Print Preview”、“Print Setup”和“Exit”命令已经实现。
- “File”菜单的“Recent File”命令是位置句柄，用于列出最近使用过的文件。当保存和打开文件时，位置句柄由最近使用过的文件列表所取代。
- “View”菜单上的命令用于显示或隐藏工具栏或状态栏，这些命令已经实现。
- “Help”菜单的“About”命令已经实现，该命令显示缺省“About”对话框。
- Visual C++ 6.0 已经自动实现了部分工具栏按钮的功能。
- 状态栏显示工具栏按钮提示和菜单命令的功能描述，Visual C++ 6.0 已经自动实现了部分状态栏信息。

1.2　MFC AppWizard 生成的类

生成 MDI 应用程序时，MFC AppWizard 将派生五个主要的类——文档类、视图类、主边框窗口类、子边框窗口类和应用程序类，主要的程序任务均分布在这些类中。MFC AppWizard 为每个类生成单独的源文件，并从程序名中获取类名和类源文件名作为缺省设置。类之间通过调用公有成员函数和消息传递进行通信和数据交换。

PhotoSee 的文档类从 CDocument 类派生而来，类名为 CPhotoSeeDoc。CPhotoSeeDoc 的类声明在头文件 PhotoSeeDoc.h 中，类实现在文件 PhotoSeeDoc.cpp 中。文档类用于存放应用程序的数据并实现文件保存和装入功能。

PhotoSee 的视图类从 CScrollView 类派生而来，类名为 CPhotoSeeView。CPhotoSeeView 的类声明在头文件为 PhotoSeeView.h 中，类实现在文件 PhotoSeeView.cpp 中。视图类确定以什么方式显示文档数据，以及用户如何与程序进行交互。

PhotoSee 的主边框窗口类从 CMDIFrameWnd 类派生而来，类名为 CMainFrame。CMainFrame 的类声明在头文件 MainFrm.h 中，类实现在文件 MainFrm.cpp。主边框窗口类用于管理应用程序窗口，显示标题栏、菜单栏、工具栏、状态栏、控制菜单和控制按钮。主边框窗口是所有 MDI 子窗口的容器。

PhotoSee 的子边框窗口类从 CMDIChildWnd 类派生而来，类名为 CChildFrame，CchildFrame 类声明在头文件 ChildFrm.h 中，类实现在文件 ChildFrm.cpp 中。子边框窗口类用于管打开的文档。每个文档及其视图都有一个单独的 MDI 子窗口。

PhotoSee 的应用程序类从 CWinApp 类派生而来，类名为 CPhotoSeeApp。CPhotoSeeApp 的类声明在头文件 PhotoSeeApp.h 中，类实现在文件 PhotoSeeApp.cpp 中。应用程序类控制应用程序的所有对象（文档、视图和边框窗口）并完成应用程序的初始化工作和最后的清除

工作。每个基于 MFC 类库的应用程序都必须有一个从 CWinApp 类派生的对象。全局唯一的应用程序对象 theApp 将创建并管理应用程序所支持的文档模板。文档模板使得所创建的文档、视图和边框窗口有机地结合起来。

除了生成主要类的源文件外，MFC AppWizard 还生成立应用程序所必须的其他文件：

- Resource.h：包含所有资源符号定义的标准头文件。
- StdAfx.h 和 StdAfx.cpp：用于生成预编译的头文件 PhotoSee.pch 和预编译类型文件 StdAfx.obj。
- PhotoSee.clw：ClassWizard 生成的数据库文件，存放由 ClassWizard 使用的信息。ClassWizard 使用这些信息来编辑已有的类和添加新类。
- PhotoSee.rc：包含资源描述信息的资源文件。资源文件包括所有的应用程序资源——存储在子目录“\res”中的图标、位图和光标等。这个文件可以使用 Developer Studio 直接编辑。
- res\PhotoSee.rc2：包含非 Developer Studio 编辑的资源。我们可以将所有不能由资源编辑器编辑的资源放置到这个文件中。
- res\PhotoSeeDoc.ico：包含 MDI 子窗口图标的图标文件，这个图标包含在资源文件 PhotoSee.rc2 中。
- res\PhotoSee.ico：包含应用程序图标的图标文件。应用程序图标包含在资源文件 PhotoSee.rc 中。
- res\Toolbar.bmp：用于创建工具栏按钮的位图文件。初始工具栏和状态栏是在主边框窗口类中构造。
- 此外，MFC AppWizard 还生成 ReadMe.txt 文件，用于描述为应用程序生成的所有源文件。

1.3 程序运行过程分析

Windows 应用程序的初始化、运行和结束工作都是由应用程序类完成的。应用程序类构成了应用程序的主执行过程。每个基于 MFC 类库创建的应用程序都有一个唯一的从 CWinApp 类派生的全局对象 theApp，该对象在窗口创建之前构造。

1.3.1 程序入口函数 WinMain

与所有 Windows 应用程序一样，基于 MFC 类库创建的应用程序也有一个 WinMain 函数。但是，我们不需要在应用程序中手工编写 WinMain 代码，它是由 MFC 类库自动提供的。应用程序启动时将自动调用 WinMain 函数，该函数执行注册窗口类的标准服务，并调用应用程序对象中的成员函数来初始化和运行应用程序。

在 PhotoSee 中，声明的 CPhotoSeeApp 类的对象变量 theApp 在源文件 PhotoSee.cpp 中：

```
CPhotoSeeApp theApp;
```

全局对象 theApp 的定义是全局的，这意味着在程序入口函数 WinMain 接收控制之前就