

深入

Visual C++ 编程



蔡宝忠 彭吉梅 编著



中国电力出版社

www.infopower.com.cn

深入 Visual C++ 编程

蔡宝忠 彭吉梅 编著

中国电力出版社

内 容 提 要

本书详细地讲述如何使用 Visual C++ 开发应用软件，内容涉及从入门到精通的各个阶段。书中介绍了基于客户/服务器结构开发实时数据库管理系统的方法，这些方法和技巧为作者首次提出，具有一定的借鉴意义。书中的实例均经过调试，可以直接使用。

本书适合有一定 C 或 C++ 基础的读者。

图书在版编目 (CIP) 数据

深入 Visual C++ 编程 / 蔡宝忠, 彭吉梅编著. - 北京: 中国电力出版社, 2001

ISBN 7-5083-0564-7

I. 深… II. ①蔡…②彭… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 12487 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

三河市实验小学印刷厂印刷

各地新华书店经售

*

2001 年 5 月第一版 2001 年 5 月北京第一次印刷

787 毫米×1092 毫米 16 开本 27 印张 615 千字

定价 39.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

目 录

第一章	学习使用 AppWizard	1
1.1	应用程序框架	1
1.2	使用 AppWizard 创建应用程序	3
1.3	应用程序简单分析	11
第二章	对话框应用程序	23
2.1	对话框设置	23
2.2	对话框类	25
2.3	创建基于对话框的应用程序	31
第三章	文档—视应用	36
3.1	文档—视结构	36
3.2	单文档接口应用 (SDI)	53
3.3	多文档接口应用 (MDI)	74
第四章	控件的使用	102
4.1	传统控件	103
4.2	标准控件	137
第五章	图形设备接口	175
5.1	利用图形设备接口控制图形显示	175
5.2	图形设备描述表	176
5.3	CDC 类体系	176
5.4	图形绘制工具	183
5.5	具体应用	190
第六章	用 MFC 进行多线程编程	211
6.1	进程与线程	211
6.2	Windows 应用中的多线程	212

6.3	创建工作线程	212
6.4	创建用户界面线程	216
6.5	线程的同步化	218
6.6	一个示例程序	221
第七章	ActiveX 技术	246
7.1	基本知识	246
7.2	COM 模型	250
7.3	ActiveX 自动化	251
7.4	ActiveX 控件	253
7.5	创建一个 ActiveX 服务器应用	260
7.6	创建一个 ActiveX 容器应用	274
7.7	ActiveX 控件使用	284
第八章	数据库编程	289
8.1	数据库概念	289
8.2	数据存取对象 (DAO)	291
8.3	DAO 实例	302
8.4	ODBC	312
8.5	一个 ODBC 数据库应用程序	324
第九章	ActiveX 数据对象	345
9.1	为什么使用 ADO	345
9.2	ActiveX 数据对象概述	345
9.3	使用 VC++ 控制 ADO 访问数据库	354
第十章	网络编程	368
10.1	套接字	368
10.2	MFC Winsock 类	369
10.3	WinSock 编程实例	377
10.4	消息 API (MAPI)	395
10.5	实现 Internet 应用	411
10.6	小结	426

第一章 学习使用 AppWizard

创建 MFC 应用一般都是从 AppWizard 开始的。每个 AppWizard 会在你创建整个项目的过程中给以指导，并提供不同的选择。当完成了这些选择后，MFC 就会为你创建一个工程，这会节省大量的时间。在 Visual C++ 中，AppWizard 的作用是巨大的。

AppWizard 可以根据用户的设定自动建立一个应用项目中全部源程序和资源，那么应用中这些源程序和资源是如何组织、如何联系的呢？本章将要讨论这个问题。

本章包括以下内容：

- ☞ 应用程序框架。
- ☞ 使用 AppWizard 创建应用程序。
- ☞ 应用程序简单分析。

1.1 应用程序框架

程序运行时，应用程序框架是信息流程的控制核心。应用程序框架所起的作用就像建筑设计图纸一样，好的设计可以将软件中的各个元素巧妙、灵活地集成在一起，否则各元素就像一堆杂乱的建筑材料，让施工人员无从下手。在软件设计中应用程序框架主要体现在消息处理和流程处理两个方面；在程序运行时，程序框架负责在适当的时候（如响应某用户事件或系统事件）调用相应的程序代码，从而完成特定的功能。从某种意义上说，应用程序框架就是整个程序的指挥员，控制着系统的运行。

下面分别就 MFC 应用程序框架的特点和其模型来介绍应用程序框架。

1.1.1 MFC 应用程序框架特点

MFC 应用程序框架有以下特点：

1. 标准性

程序员在设计实现大的项目时需要先考虑合适的程序结构，显然，不同的程序员在构建项目时采用的思路可能大相径庭，从而使设计出的程序结构也是截然不同的。这样如果项目实现过程中需要临时增加人手，对于后来者来说，首先需要花费很大精力去研究分析原有的体系结构，只有在掌握了基本流程后才能真正投入开发工作。另外，对系统进行维护也同样面临这样的问题，其难度也是相当大的。

解决这一问题的唯一途径就是尽可能地实现系统结构上的统一。采用 Visual C++ 作

为编程工具，它所提供的 MFC 应用程序框架就能够满足这一需要。目前在许多软件环境中都采用了 MFC 应用程序框架，经实践证实，完全可以利用这种体系结构完成项目开发。

2. 高效性

Win32 程序存储和操作数据时都以 32 位为单位，这相对于 16 位数据处理而言，优势很明显。如果一个机器周期处理的是 16 位数据，那么 32 位数据的处理则至少需要两个机器周期，因此 Win32 程序的执行速度比 Win16 程序要快得多。

可以充分利用 MFC 应用程序框架带来的方便，通过动态地建立与 MFC 代码的连接，或者灵活地应用 DLL 动态连接库的功能，使一个功能强大的应用程序的执行代码非常简短，提高其执行速度。

3. 简便性

Visual C++ 为我们提供了一个功能强大的工具 AppWizard，可以利用 AppWizard 自动生成程序代码框架，具体的消息处理函数还可以在 ClassWizard 工具的帮助下完成，从而大大地简化了程序的工作。

4. 功能性

MFC 应用程序框架所提供的功能是非常丰富的，特别是 Visual 6.0 版本，除了对多文档接口应用程序、动态连接库、开放式数据库互连技术、TCP/IP 数据通讯技术的支持以外，还提供了对 OLE 控件、ActiveX 控件及模板库的支持。

1.1.2 MFC 应用程序框架模型

MFC 提供了应用级的类 CWinApp，它将程序启动后的所有初始化工作都进行了封装，包括注册窗口类、创建窗口、装入快捷键等，这些工作可以由成员函数 InitInstance 完成。另外还要对消息循环进行控制，这包括对消息的检索、转换和分发处理，消息循环的处理由成员函数 Run 实现。最后当接收到退出程序的消息时，还要通过成员函数 ExitInstance 实现程序退出前的环境清理工作。当然对于不同的系统在程序启动时和程序清理时所需完成的工作会有所不同，可以通过重载 InitInstance 和 ExitInstance 来实现。

由以上的简单介绍可以看出，在整个程序执行过程中消息处理过程是最关键的部分。MFC 提供了 CCmdTarget 类来支持消息处理，实际程序中都是由 CCmdTarget 的派生类来处理应用中由 CWinApp 分发下来的命令消息，如文档、文档模板、视等。如何处理消息还要根据消息映射表的内容决定，消息映射表（Message Map）是 CCmdTarget 派生类特有的属性，其中每个表项表示一条消息，包括消息内容及消息处理函数的入口地址。

MFC 应用程序框架模型如图 1-1 所示。

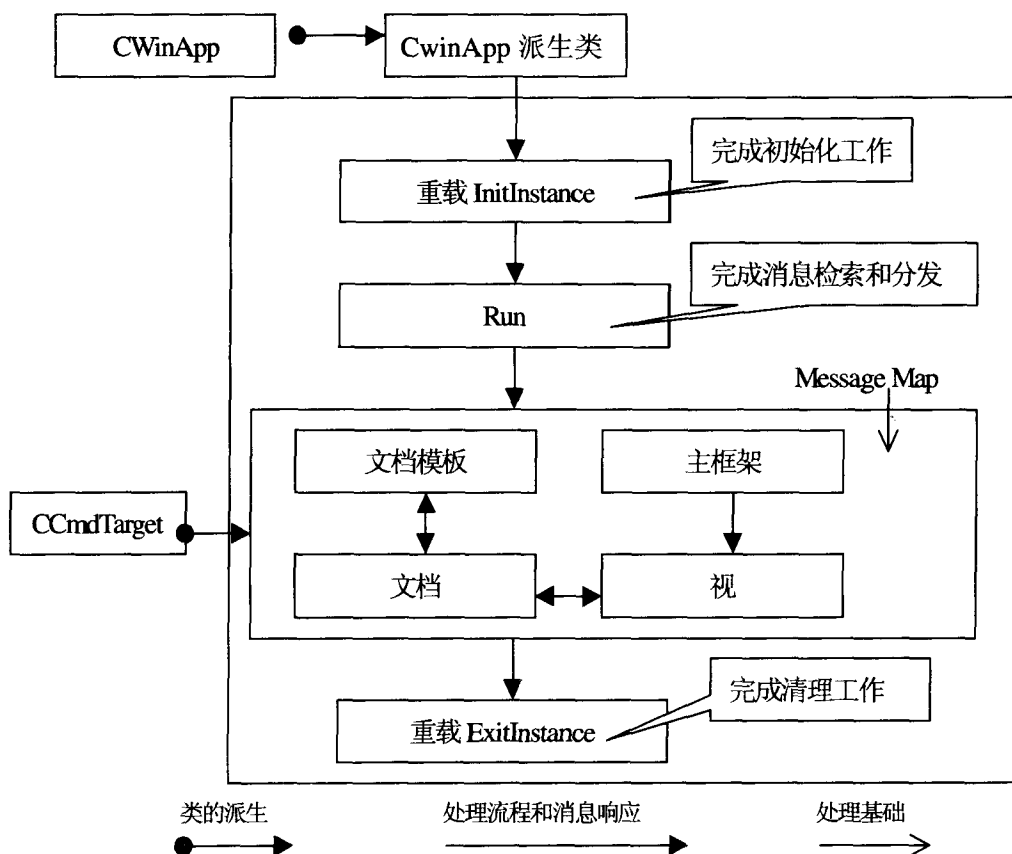


图 1-1 MFC 应用程序框架模型

1.2 使用 AppWizard 创建应用程序

借助于 Visual C++ 所提供的 AppWizard 代码生成器，可以非常方便地生成 MFC 应用程序。如果不增加任何代码，该应用程序可以显示一个菜单以及与之相关联的空窗口。下面我们就从这个最简单的应用程序开始学习 MFC 程序的编写。

1.2.1 启动 AppWizard

从 Developer Studio 的 File 文件菜单中选择 New... 选项，系统将弹出如图 1-2 所示对话框，要求选择新建工程或文件类型。

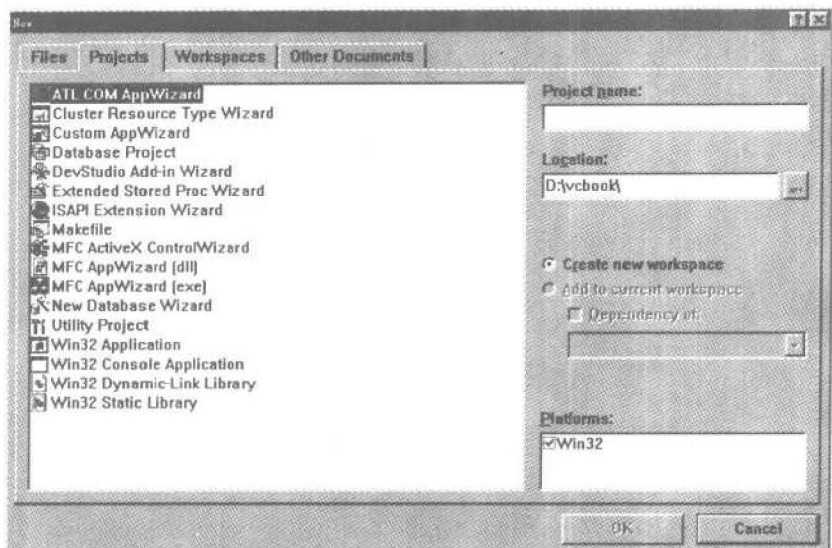


图 1-2 New 对话框——AppWizard 的启动

1.2.2 创建工程

在工程 (Projects) 页中, 选择 MFC AppWizard(exe), 表示我们需要生成一个 MFC 应用程序, 应用程序名需要在对话框右上方的 Project name 一栏中填写, 其路径可以在 Location 一栏中设置。由于这是一个新创建的独立的工程, 因此仍选用默认的 Create new workspace 单选设置, 系统平台 Win32 也不作变动。如在 d:\vcbook\ 路径下创建一个名为 first 的应用, 有关的设置如图 1-3 所示。

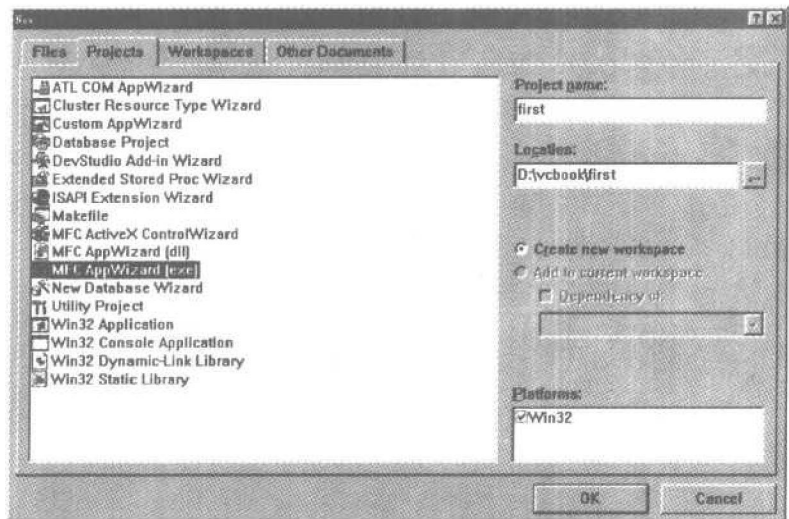


图 1-3 New 对话框——创建工程

1.2.3 设置应用程序类型

单击 OK 按钮后，系统将弹出对话框要求选择应用程序类型，如图 1-4 所示。

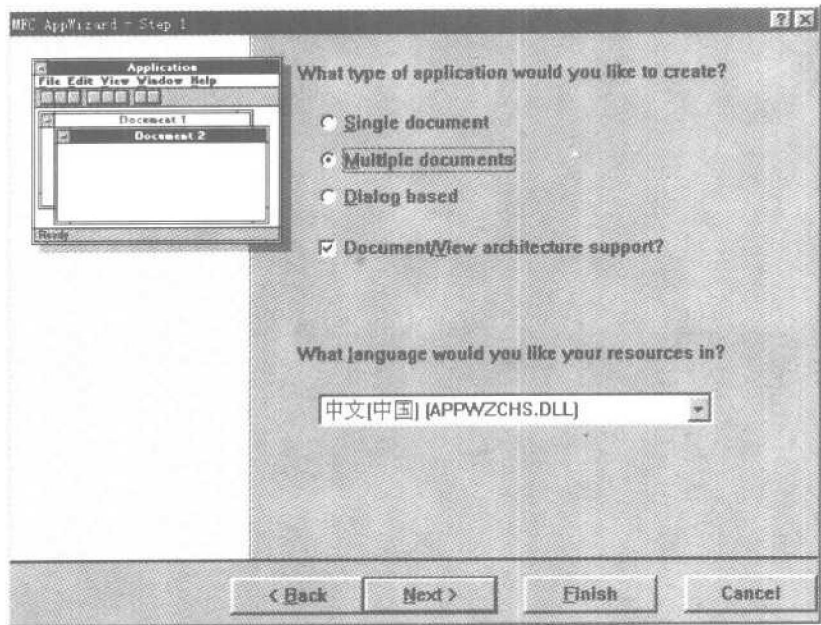


图 1-4 Step1——设置应用程序类型

MFC 最常用的结构是基于文档—视结构的，文档—视结构又包括单文档（SDI）和多文档（MDI）两种类型，此外还可以创建基于对话框的应用程序。这里仍选择系统的默认值 Multiple documents，即支持多文档的“文档—视”结构。语言采用中文，这取决于是否存在相应的语言动态连接库，如果没有安装 APPWZCHS.DLL，那么最可靠的做法是选用 English(United States)，因为系统在安装时必然安装有 APPWZENU.DLL，以提供对英文的支持。

1.2.4 数据库设置

单击 Next 按钮后，系统提示进行数据库选项设置，如图 1-5 所示。

Visual C++ 提供对 ODBC(开放数据库互连)的支持，可以在这里选择相应的设置，如果选择了数据库视图，则无论是不包含数据库视图文件支持（Database view without file support）还是包含文件支持（Database view with file support），下面的 Data Source... 按钮将处于启用状态，即要求设置相应的 ODBC 数据源。这里我们仍采用系统默认值，选择不支持数据库方式。

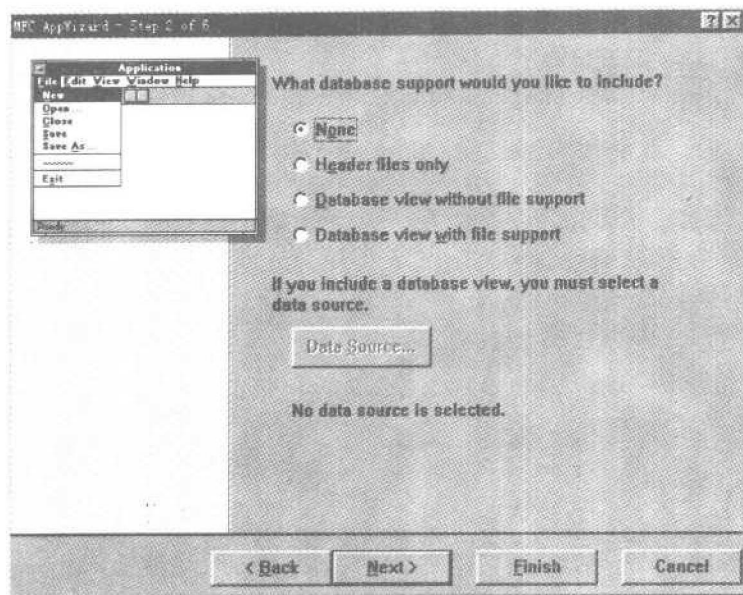


图 1-5 Step2——数据库设置

1.2.5 组件设置

单击 Next 按钮后，系统提示进行组件选项设置，如图 1-6 所示。

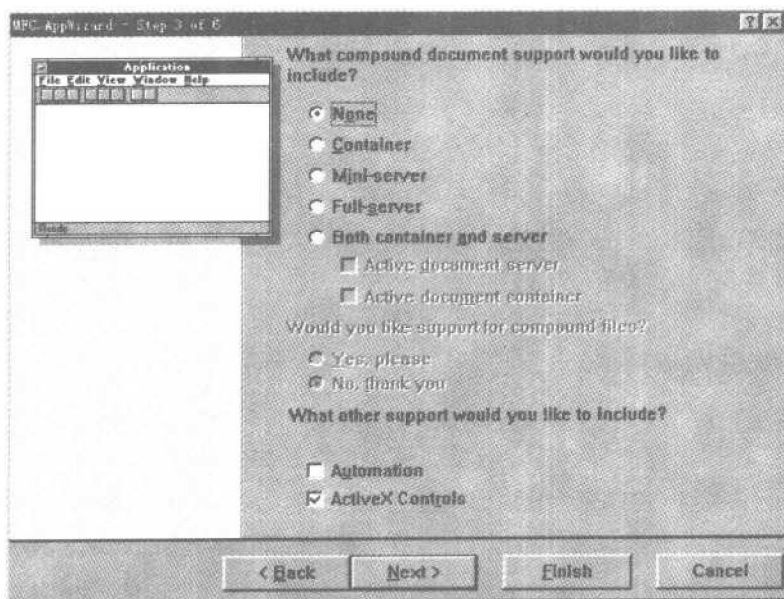


图 1-6 Step3——组件设置

这里提供了 5 个单选的设置选项，通过设置可以产生相应的组件程序。其中 Container 表示应用程序将作为组件容器；选择 Mini_server 和 Full_server 可以分别生成 MFC 最小服务器和完全服务器；Both container and server 则表示应用程序既可作为组件容器，又可以作为服务器。这里选择 None，即不生成组件程序。

另外系统还要求设置是否对自动化和 ActiveX 控件的支持，这里取系统默认值，即支持 ActiveX 控件。

1.2.6 系统设置

单击 Next 按钮后，系统提示进行系统设置。

系统提供了 7 个复选框以提供不同的设置风格，即浮动的工具条、初始化状态条、打印和打印预览功能设置、联机帮助、3D 控件、MAPI 和 Socket 联接。另外还可以选择工具条样式，如普通样式或类似于 Internet 浏览器的“酷”样式。

最近打开的文件可以在文件菜单下列出，一般可列 4 个新近打开的文件，也可以通过设置对话框下面的下拉表调整文件个数。

通过单击 Advanced...，还可以对界面属性进行进一步设置。

这里我们选用浮动的工具条、初始化状态条、打印与打印预览功能设置和 3D 控件设置，如图 1-7 所示。

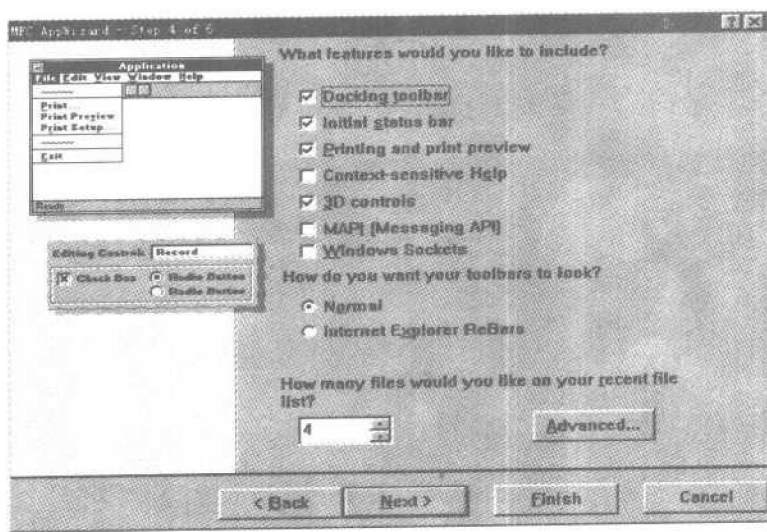


图 1-7 Step4——系统设置

1.2.7 工程设置

工程设置包括对工程类型的设置、文件评注的设置和 MFC 类库应用方式。这里设置

工程为 MFC 标准应用程序，并要求自动生成文件评注。特别要强调的是：在一般情况下 MFC 类库的连接方式均采用共享的动态连接形式，即选择 As shared DLL，否则将使执行代码大大增加，如图 1-8 所示。

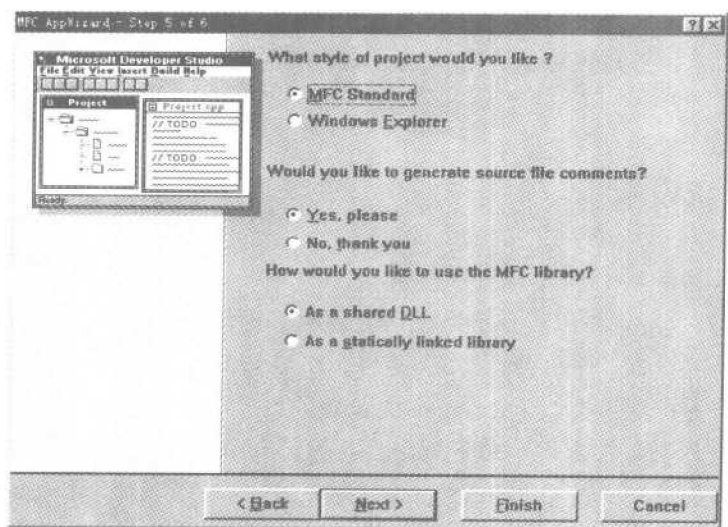


图 1-8 Step5——工程设置

1.2.8 设置类与文件名

单击 Next 按钮，进入工程创建的最后一步，如图 1-9 所示。

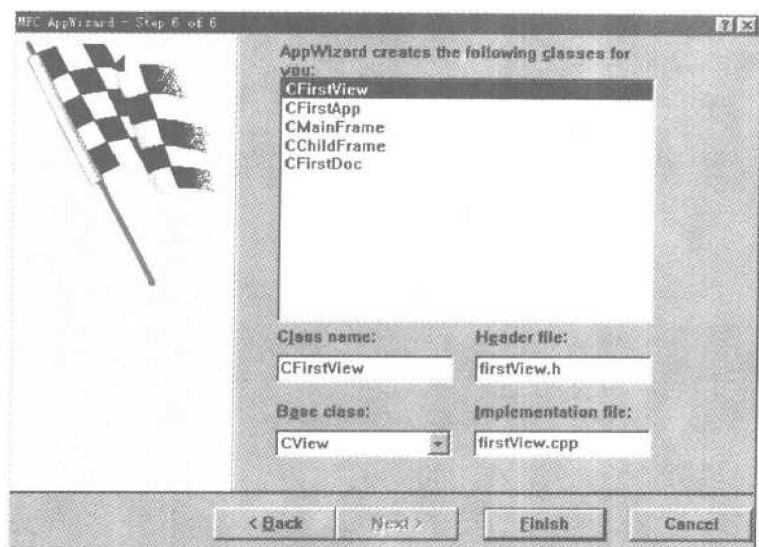


图 1-9 Step6——类与文件名的设置

如果需要改变类的设置，如类的基类、类名等可以在此对话框内完成。注意一旦单击 **Finish** 按钮，系统将自动创建工程，而且以此对话框中类的设置为基础生成相应的文件；因此在单击 **Finish** 按钮之前，需要对相应设置进行检查，否则如果发现设置有误，则必须在已经生成的文件中进行修改；由于修改的内容往往是互相关联的，所以修改工作相对要麻烦一些，为了避免不必要的修改工作，建议在创建工程前要明确需要设置的属性。

1.2.9 工程生成

单击 **Finish** 按钮后，系统将弹出如图 1-10 所示对话框，以通知系统将会自动生成的文件。

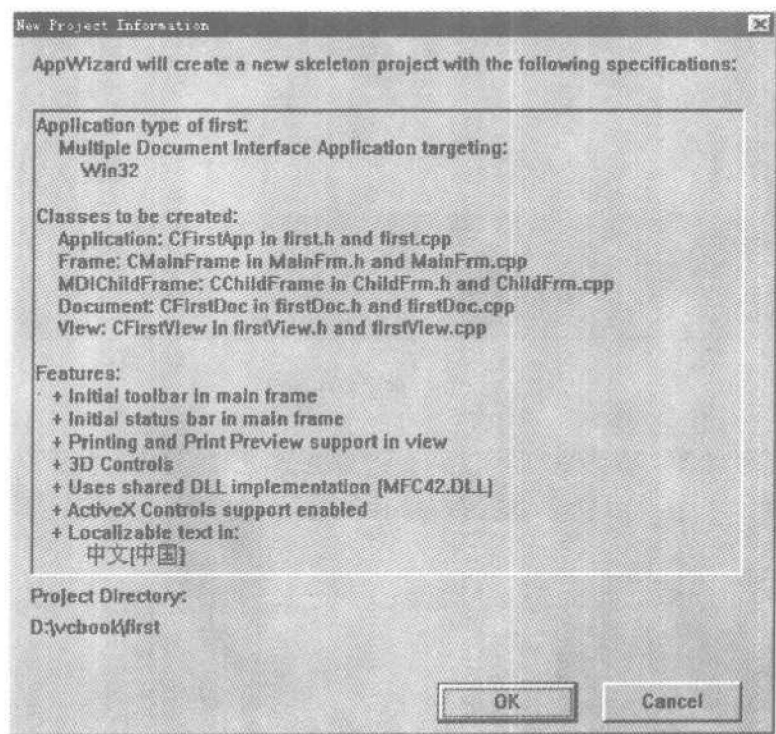


图 1-10 New Project Information (新工程信息)

这些内容仅供查看，不能编辑。但如果程序员对设置不满意，也可以单击 **Cancel**，撤销此次创建。否则系统将在 D:\vcbook 路径下生成 first.dsw 工程及相应文件。

1.2.10 应用程序功能

到此为止，我们没有增加一句代码，但一个应用程序确实在已经诞生了。可以单击

execute program 按钮或选择相应的菜单来执行程序，系统在编译、连接程序前会提示无当前工程的可执行程序，若要求生成，选择 OK 即可。

下面看一看应用程序的运行效果如何，如图 1-11 所示。

显然，这个程序已经具备了 Windows 应用程序的所有特征了，仅从界面上，可以看到这是一个多窗口、包含工具条和状态条的应用程序，而且当选择 file 菜单下的打印预览时，将弹出如图 1-12 所示窗口。

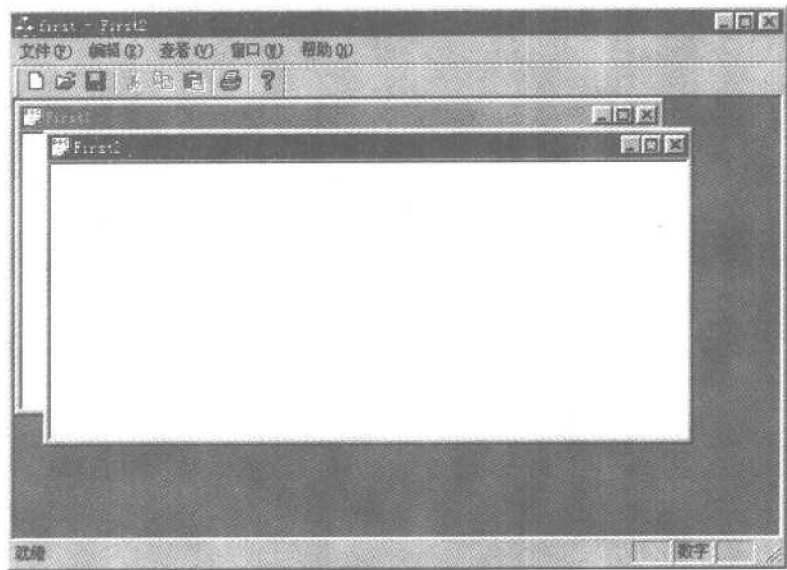


图 1-11 first 应用程序执行

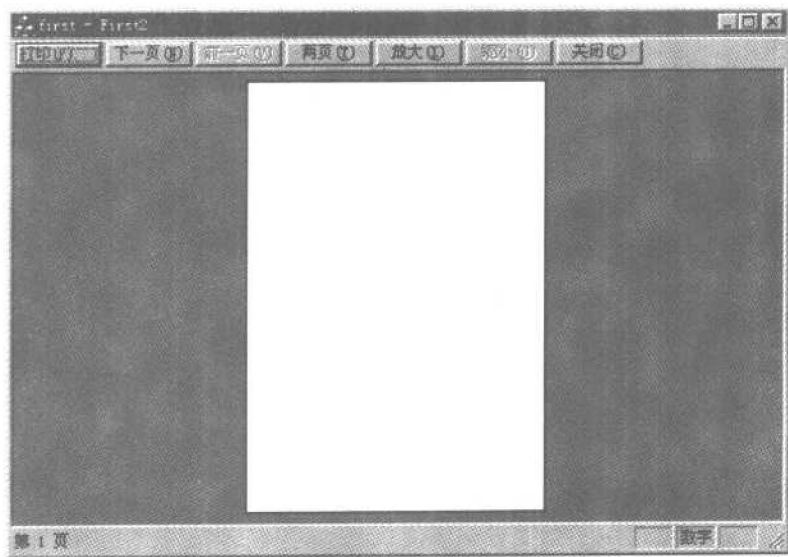


图 1-12 first 应用程序——打印预览窗口

这些功能是在前面的设置工作中规定让 Visual C++ 完成的，虽然它并没有什么具体用途，但在此基础上，就可以开发出功能相对复杂的应用程序了。

1.3 应用程序简单分析

在 AppWizard 使用过程中，我们如同通过一个“黑箱”，进入“黑箱”前，手中什么也没有，走出“黑箱”时，系统已经送给了我们一个小小的礼物——一个功能相当简单的应用程序。那么这个黑箱到底做了什么工作呢？可以通过对这份小礼物作一个简单的分析。

下面就按分类、资源和文件三个方面来对这个程序分析。

1.3.1 first 应用程序类的分析

在 Visual C++ 开发环境中，Project Workspace 窗口的 ClassView 页中显示了应用程序中所有的 C++ 类，其内容是由 Visual C++ 从项目文件源程序中自动搜集的，包括类的成员变量和成员函数，如图 1-13 所示。

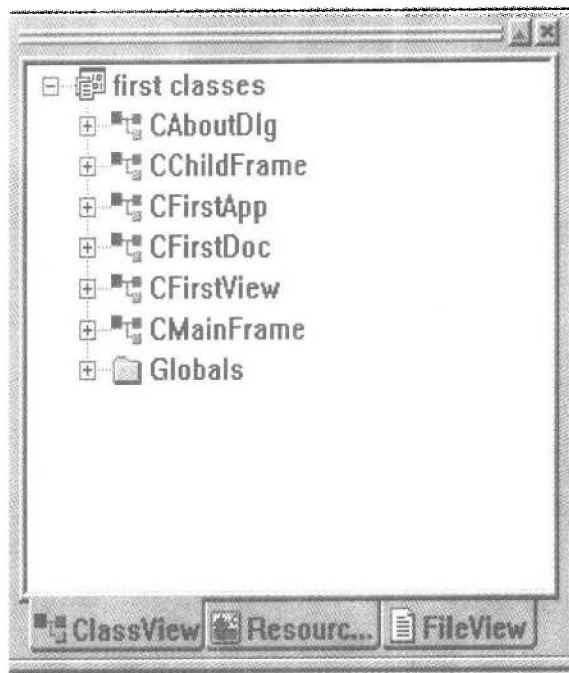


图 1-13 first 应用程序类

First 应用程序中共生成了 6 个类，其中 CFirstApp 是基类为 CWinApp 的应用程序类；由于这是一个多文档应用，因此其结构是建立在“文档一视”基础上的，CFirstView 类即

为基类为 CView 类的视类；为了实现多文档，这里生成了 CMainFrame 和 CChildFrame 类，分别管理主窗口和子窗口；文档类 CFirstDoc 是以 Cdocument 类为基类生成的。

下面来看一下，这些基类的派生类究竟完成了什么工作。

CFirstApp 应用程序类

双击 ClassView 中的 CFirstApp，将在程序区打开类的头文件，并将光标停在 CFirstApp 类的定义处，可以看到一个标准的类定义：

```
class CFirstApp : public CWinApp
{
public:
    CFirstApp();

    // Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CFirstApp)
    public:
    virtual BOOL InitInstance();
    //}}AFX_VIRTUAL

    // Implementation
    //{AFX_MSG(CFirstApp)
    afx_msg void OnAppAbout();
    // NOTE - the ClassWizard will add and remove member functions here.
    // DO NOT EDIT what you see in these blocks of generated code !
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
```

类定义的结构如图 1-14 所示。

CFirstApp 类的定义中主要包括三个部分：构造函数、重载方法和实现部分，其中实现部分主要是消息映射方法的声明。

这个定义说明了 CFirstApp 类为 CWinApp 类的派生类，除了公共的构造函数外，重载了 CWinApp 类的 InitInstance 函数以处理初始化操作，另外设置了对 AppAbout 消息的响应事件，AppAbout 消息是单击应用程序帮助菜单上的关于 first.... 产生的。

先来看 InitInstance 函数的工作；