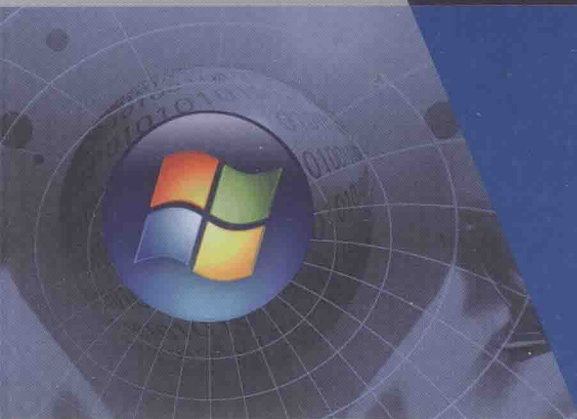


计算机类主干课程系列教材

高级 Windows 应用程序设计

◎ 李 媛 孙阳光 李 祥 主编



科学出版社

计算机类主干课程系列教材

高级 Windows 应用程序设计

李 媛 孙阳光 李 祥 主 编

科学出版社

北 京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书降低了学习 Windows 程序设计的门槛,以 Visual C++ 6.0 中文版为平台,根据微软基础类库所提供的应用程序框架,综合介绍用于 Windows 开发的各种组件,深入浅出地介绍 Windows 的图形图像处理、通用数据类、界面设计、鼠标键盘、资源设计、动态链接库、对话框、数据库应用、高级应用等知识。

本书可以作为大学本科、专科院校学生学习 Windows 程序设计(或 MFC Windows 程序设计)的教材,也可以供高职高专学生学习使用。

图书在版编目(CIP)数据

高级 Windows 应用程序设计/李媛,孙阳光,李祥主编. —北京:科学出版社,2014.6

计算机类——教材——系列教材

ISBN 9 53-7-05-04334-1

I.高… ①李… ②孙… ③李… III.① Windows 操作系统—高等学校—教材 IV.①TP367.7

中国版本图书馆 CIP 数据核字(2014)第 128964 号

责任编辑:张颖兵 杜 权/责任校对:肖 婷

责任印制:高 嵘/封面设计:苏 波

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉市科利德印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:787×1092 1/16

2014 年 6 月第 一 版 印张:15

2014 年 6 月第一次印刷 字数:325 000

定价:29.80 元

(如有印装质量问题,我社负责调换)

前 言

近年来,Windows 一直是一个主流操作系统。市场上开发 Windows 应用程序的需求与日俱增。随着 Windows 操作系统的发展,Windows 应用程序变得越来越复杂。

本书以实际应用为目的,以介绍 Windows 各个组件、对话框创建、数据库 Windows 创建为主线,系统介绍 Windows 的开发技术。全书共 11 章,主要内容包括 Windows 程序设计的基础知识、Windows 应用程序框架、图形图像处理、通用数据类、界面设计、鼠标键盘、资源设计、动态链接库、对话框、数据库应用、高级应用等知识。

为方便教学,本书每章都有配套的上机实验和习题(含答案)。这些实验内容和习题凸显了本章的知识要点,通过实践环节体现“做中学”的道理。与本书配套的程序代码,包括了例题和实验源代码,更是极大地方便了教与学。

本书结构合理,逻辑性强,写作特色鲜明。每个章节、每个知识点都有精心设计的典型例子说明其用法,各章节之间的联系紧凑、自然。

本书可以作为大学本、专科院校学生学习 Windows 程序设计(或 MFC Windows 程序设计)的教材,也可以作为高职高专学生学习使用。

本书由李媛、孙阳光、李祥主编,谢爱萍任副主编。李媛完成教材的第一章至第十一章,包括各章实验,执笔约 30 万字;孙阳光负责编写习题和答案,并对全书进行了统稿;由李祥负责审核例题代码,审核代码字数 1.5 万字;谢爱萍负责审核习题答案。

由于作者水平有限,书上不当之处在所难免,恳请读者批评指正。对本书的意见和建议请发送到电子信箱 liyuanlx@126.com。

目 录

第1章 Windows 应用程序概述	1
1.1 Windows 应用程序基本概念	1
1.1.1 Windows 应用程序界面	1
1.1.2 Windows 应用程序与 API 函数	2
1.1.3 Windows 应用程序的句柄	2
1.1.4 Windows 应用程序的数据类型	2
1.2 Windows 程序设计平台开发环境	3
1.2.1 Visual C++ 简介	3
1.2.2 创建 Win32 Console 应用程序	3
1.2.3 创建 Win32 应用程序	5
1.2.4 创建 MFC Windows 应用程序	6
1.3 消息映射	11
1.3.1 事件与消息	11
1.3.2 消息映射	12
1.3.3 消息处理	12
1.4 消息处理实例	13
1.4.1 鼠标消息处理的实例	13
1.4.2 键盘消息处理的实例	15
习题 1	17
实验 1 高级 Windows 应用程序开发基础	18
第2章 Windows 图形绘制	20
2.1 设备描述环境和图形设备接口	20
2.1.1 设备描述环境	20
2.1.2 图形设备接口	20
2.2 CDC 类	21
2.3 CPen 类	24
2.4 CBrush 类	26
2.5 文本和字体信息	27
2.5.1 文本	27
2.5.2 获取字体的信息	29
2.6 CDC 类的其他派生类	30
2.6.1 窗口用户区设备描述环境 CClientDC 类	30
2.6.2 图元文件设备描述环境 CMetaFileDC 类	31
习题 2	33

实验 2 图形绘制	34
第 3 章 通用数据类	36
3.1 简单数据类	36
3.1.1 点类 CPoint	36
3.1.2 矩形类 CRect	38
3.1.3 尺寸类 CSize	39
3.1.4 字符串类 CString	40
3.2 群体数据类	41
3.2.1 数组类 CArray	41
3.2.2 链表类 CList	44
3.3 文档类与视图类之间的数据传递	45
习题 3	49
实验 3 数据类	50
第 4 章 窗口界面的设计	55
4.1 单文档界面	55
4.2 多文档界面	55
4.3 单文档界面窗口样式	56
4.3.1 在 MFC AppWizard 中确定窗口样式	57
4.3.2 在函数 PreCreateWindow 中修改窗口的样式	58
4.3.3 可拆分窗口的单文档界面	61
4.4 窗口的同步更新	62
4.4.1 可拆分窗口的同步更新	62
4.4.2 提高同步更新的效率	65
4.5 创建有滚动条的窗口	68
习题 4	71
实验 4 界面设计	72
第 5 章 鼠标和键盘消息	73
5.1 鼠标消息	73
5.1.1 用户区鼠标消息	74
5.1.2 非用户区鼠标消息	77
5.1.3 捕获鼠标消息	78
5.2 键盘消息	80
5.2.1 虚拟码	80
5.2.2 按键消息	82
5.3 窗口的焦点	84
习题 5	86
实验 5 鼠标和键盘	87
第 6 章 资源	88
6.1 资源概述	88

6.1.1	资源的含义	88
6.1.2	资源头文件	88
6.1.3	资源描述文件	90
6.2	菜单资源	90
6.2.1	菜单资源的定义	91
6.2.2	编辑菜单	92
6.2.3	自定义菜单	94
6.2.4	菜单编辑器	98
6.3	快捷键表	100
6.3.1	快捷键表资源定义	100
6.3.2	快捷键编辑器	100
6.4	图标	102
6.4.1	图标编辑器	102
6.4.2	图标资源定义	103
6.5	位图	104
6.5.1	位图资源定义	104
6.5.2	使用位图	105
习题 6		108
实验 6	资源	109
第 7 章	控件	112
7.1	标准控件	112
7.1.1	控件的含义	112
7.1.2	标准控件	112
7.1.3	静态文本控件	113
7.1.4	按钮控件	113
7.1.5	编辑控件	116
7.2	控件的背景颜色	118
习题 7		120
第 8 章	对话框	122
8.1	对话框模板	122
8.1.1	资源描述文件对话框模板脚本	122
8.1.2	对话框编辑器设计对话框	124
8.2	对话框类	124
8.2.1	对话框类的定义	124
8.2.2	对话框的类型	126
8.3	对话框的数据交换	129
8.3.1	数据交换函数 DDX	130
8.3.2	进行数据交换函数 DoDataExchange	130
8.3.3	更新数据函数 DoDataExchange	130

8.3.4 为对话框控件添加成员变量	131
8.3.5 数据检验	133
8.4 基于对话框的应用程序	134
8.4.1 对话框应用程序框架的设计	134
8.4.2 对话框应用程序的消息响应函数	134
8.4.3 属性页	137
习题 8	140
实验 7 对话框	141
第 9 章 动态链接库	144
9.1 链接库	144
9.1.1 静态链接库	144
9.1.2 动态链接库	144
9.2 静态链接库的创建	144
9.3 动态链接库的创建	147
9.3.1 导出函数的声明	147
9.3.2 MFC 动态链接库的创建	151
9.4 动态链接库的应用	154
9.4.1 隐式链接	154
9.4.2 显式链接	154
习题 9	156
实验 8 动态链接库	157
第 10 章 数据库编程	158
10.1 数据库概述	158
10.1.1 数据模型	158
10.1.2 数据库和数据表的创建	159
10.2 MFC ODBC 操作	161
10.2.1 MFC ODBC 向导过程	161
10.2.2 ODBC 数据表绑定更新	167
10.2.3 MFC 的 ODBC 类	168
10.3 MFC ODBC 常用编程	172
10.3.1 显示记录总数和当前记录号	172
10.3.2 编辑记录	173
10.3.3 多表处理	177
习题 10	182
第 11 章 高级应用	184
11.1 多媒体	184
11.2 ActiveX 控件	184
11.2.1 创建一个 ActiveX 控件程序	184
11.2.2 使用 ActiveX 控件	189

11.3 网络应用	190
11.3.1 MFC 对网络的支持	190
11.3.2 Windows Sockets 编程	190
11.3.3 WinInet 应用	195
习题 11	198
实验 9 Windows 高级应用	199
参考答案	206
参考文献	227

第 1 章 Windows 应用程序概述

Windows 是一种应用于计算机的操作系统,它为应用程序提供的多任务环境具有一致的图形化窗口和菜单。在 Windows 操作系统上运行的应用程序称为 Windows 应用程序。本章主要介绍如下内容:

(1) Windows 应用程序的基本概念;

(2) Windows 应用程序的建立;

(3) 消息的概念;

(4) 消息处理函数。

1.1 Windows 应用程序基本概念

1.1.1 Windows 应用程序界面

Windows 是微软公司开发的一种操作系统。以 Windows 作为平台运行的应用程序称为 Windows 应用程序。Windows 应用程序的一个显著特点是它有一个美观的图形用户界面 (graphical user interface, GUI), 参见图 1.1.1。通过鼠标、键盘配合操作图形用户界面, 大大方便了用户对 Windows 应用程序的控制。仔细观察这个图形用户界面, 它是由许多不同的图形元素组成的, 如图标、标题栏、菜单栏、工具栏、状态栏、窗口边界、滚动条、控制按钮等。其中某些图形元素在接受用户的某个动作后, 可以使 Windows 程序执行某种相应的操作。例如, 在图 1.1.1 中, 用户通过单击菜单栏中的某个菜单项, Windows 应用程序会产生一个下拉菜单; 用户通过单击工具栏中的文件打开图标的工具按钮时, Windows 应用程序会弹出一个打开对话框。

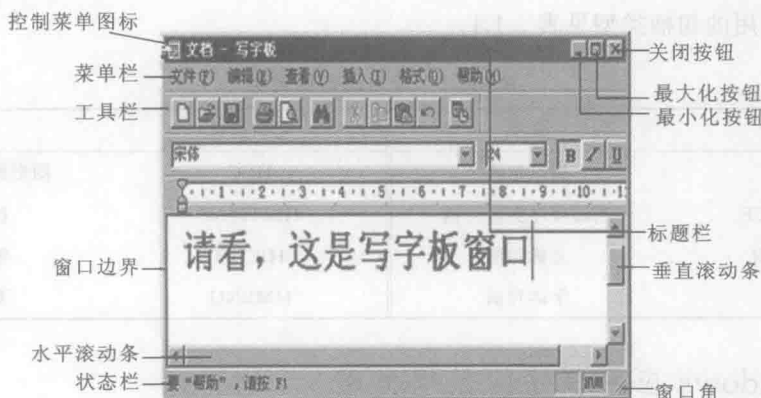


图 1.1.1 Windows 应用程序的图形用户界面

显然, 这个图形用户界面是 Windows 应用程序与用户之间交换信息的一个“窗口”, 简单

的 Windows 应用程序只有一个窗口,复杂的 Windows 应用程序有多个窗口。设计一个或一组满足用户要求的窗口,是 Windows 应用程序设计中一项必不可少的任务。

1.1.2 Windows 应用程序与 API 函数

Windows 提供了大量可以在应用程序中调用的、用 C 语言编写的函数。这些由操作系统提供,程序员可以在程序中直接调用的函数称为应用编程接口(application programming interface,API)函数。根据 Windows API 函数的功能,可以把它们分为以下三大类。

- (1) 窗口管理函数。实现窗口创建、移动和修改等功能。
- (2) 图形设备函数。实现图形的绘制和操作功能,这类函数的集合又称为图形设备接口。
- (3) 系统服务函数。实现与操作系统有关的一些功能。

早期用来设计 Windows 应用程序的开发工具是软件开发工具包。用软件开发工具包来设计 Windows 应用程序,实际上就是直接使用 API 函数来开发应用程序。由于用这种方法来设计 Windows 应用程序需要编程人员记忆大量的 API 函数,所以用这种方法设计 Windows 应用程序极其困难,从而阻碍了 Windows 应用程序的推广。

自从出现了面向对象的程序设计方法后,出现“类”这种数据类型并对 Windows API 函数进行了封装,从而使 Windows 应用程序结构和开发工具发生了巨大的变化。目前,用来设计 Windows 应用程序的开发工具大多是面向对象且可视的,如 Visual C++、Visual Basic、Visual Java、C++ Builder 等。由于这些可视化的开发工具大幅度地提高了 Windows 程序的开发效率,所以这些可视化开发工具得到了广泛的应用。例如,使用 Visual C++ 的微软基础类库(Microsoft foundation class,MFC)来设计 Windows 应用程序。作为 Windows 应用程序的设计开发人员,了解不同的 Windows 应用程序设计的方法是必要的。

1.1.3 Windows 应用程序的句柄

在 Windows 应用程序的窗口中,存在很多如窗口、按钮、滚动条、菜单、图标等复杂的对象和实例。为了在 Windows 应用程序中区分和使用它们,必须对它们进行标识。为了区别于普通变量的标识,Windows 把这种复杂对象的标识称为句柄。不同 Windows 对象的句柄具有不同的类型,常用的句柄类型见表 1.1.1。

表 1.1.1 常用的句柄类型

句柄类型	说明	句柄类型	说明
HWND	窗口句柄	HDC	图形设备环境句柄
HINSTANCE	当前程序实例句柄	HBITMAP	位图句柄
HCURSOR	光标句柄	HICON	图标句柄
HFONT	字体句柄	HMENU	菜单句柄

1.1.4 Windows 应用程序的数据类型

Windows 应用程序中的数据量非常大,为了提高程序的可读性,Windows 应用程序给许多基本数据类型定义了别名。例如:

```
typedef unsigned long    DWORD;
```

```
typedef int          BOOL;
typedef unsigned char BYTE;
typedef unsigned short WORD;
typedef float        FLOAT;
typedef unsigned int  UNIT;
```

除了这些基本数据类型的别名外,Windows 应用程序还使用了一些特定结构类型的数据。例如,下列的结构描述了一个点在窗口的位置坐标。

```
typedef struct tag_POINT
{
    LONG x;
    LONG y;
} POINT;
```

又例如,下列的结构描述了一个矩形在窗口中的大小和位置。

```
typedef struct tag_RECT
{
    LONG left;
    LONG top;
    LONG right;
    LONG bottom;
} RECT;
```

在 Windows 应用程序设计中,既可以使用 C 语言的基本数据类型,也可以使用 Windows 自定义的数据类型。所有 Windows 自定义的数据类型的关键字都要大写。

1.2 Windows 程序设计平台开发环境

1.2.1 Visual C++ 简介

Visual C++ 是 Windows 环境最优秀的 C++ 编程环境之一,它是微软公司开发的 Visual Studio 系列产品的一部分,具有集成开发环境,可以编辑 C 语言、C++ 等编程语言。VC++ 整合了方便的调试工具,特别是整合了 Win32 API 应用程序、MFC、Microsoft.NET 等框架。Visual C++ 支持三种类型的应用程序框架的创建。下面分别介绍如何创建控制台应用程序、Win32 应用程序和 MFC Windows 应用程序。

1.2.2 创建 Win32 Console 应用程序

打开 VC++ 6.0,在选择“文件→新建”菜单后,在“新建”对话框的“项目”选项卡中选择“Win32 Console Application”,输入项目名称 First 并选择好项目位置后,单击“确定”按钮。在如图 1.2.1 所示的应用程序向导中,选择其中第三项 A “Hello World!” application,单击“Finish”按钮,直接运行程序以后出现如图 1.2.2 所示的窗口。字符串“Hello World!”出现在窗口中的第一行,按下任意键后窗口将会消失。

在 Win32 Console 应用程序运行后产生的窗口中,有标题栏和垂直滚动条,没有菜单栏、工具栏和状态栏,标题栏中的内容是可执行文件在磁盘中的存放位置。当用户单击窗口右上

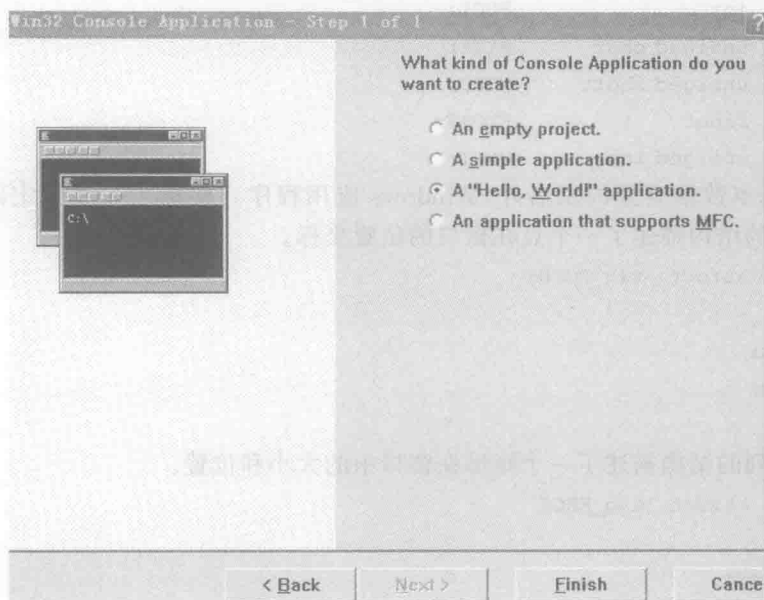


图 1.2.1 Win32 Console 应用程序向导



图 1.2.2 Win32 Console 应用程序运行窗口

角的关闭按钮时,窗口将会消失。这是一个功能比较简单的窗口。

由于控制台应用程序运行后的窗口功能比较简单,计算机用户经常把控制台应用程序看成一种“遗物”。但是,实际上很多系统管理员和一些编程高手还是非常喜欢控制台应用程序的。对于那些使用 Visual Studio 学习 C/C++ 语言的初学者,编写控制台应用程序是必须经历的学习阶段。

在 Windows 程序设计的学习过程中,部分算法的设计、开发工具的使用,仍然可以使用简单、直观的控制台应用程序。当准备创建控制台应用程序时,在“新建项目”对话框中直接选择“Win32 Console Application”。在图 1.2.1 的 Win32 Console 应用程序向导中,选择第一项

An empty project, 创建一个空的工程。接着, 选择“文件→新建”菜单, 在“新建”对话框的“文件”选项卡中选择 C++ Source File。在源程序的入口函数 main 中需要使用大量的 API 函数, 来编写程序员自定义的控制台应用程序。

1.2.3 创建 Win32 应用程序

打开 VC++ 6.0, 在选择“文件→新建”菜单后, 在“项目”选项卡中选择“Win32 Application”, 输入项目名称 Second 并选择好项目位置后, 单击“确定”按钮。在如图 1.2.3 所示的对话框中选中“一个典型的‘Hello World!’程序”单选按钮, 单击“完成”按钮, 直接运行程序以后出现如图 1.2.4 所示的窗口。

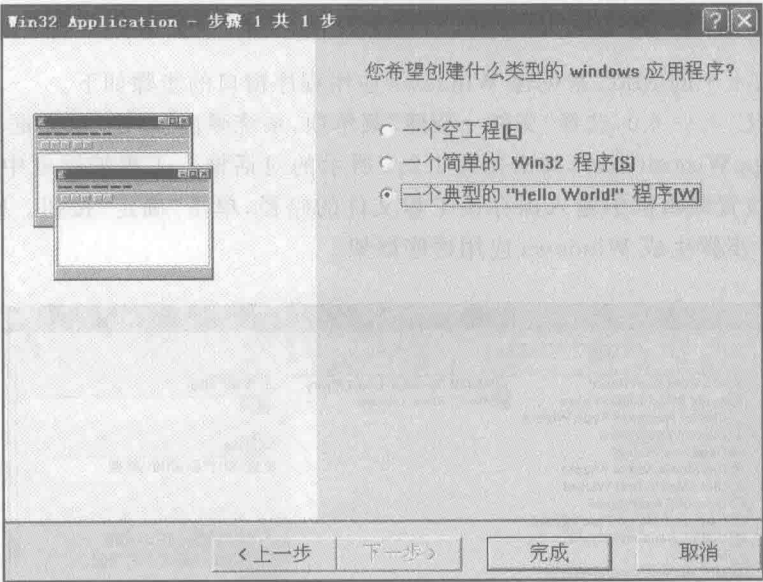


图 1.2.3 Win32 应用程序向导

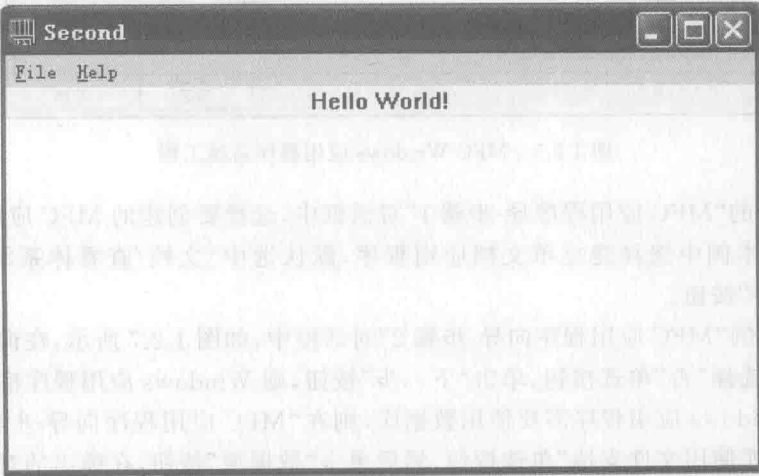


图 1.2.4 Win32 应用程序运行窗口

在如图 1.2.4 所示的窗口中,以项目名 Second 作为标题栏中的标题内容,并且标题内容的左边有 Win32 应用程序的图标。窗口的菜单栏中有 File 和 Help 两个菜单选项。用户单击任何一个菜单选项以后,出现一个下拉子菜单栏。窗口中没有工具栏、状态栏和滚动条,当用户单击窗口右上角的关闭按钮时,窗口将会消失。相比于控制台应用程序运行后的窗口,Win32 应用程序运行后的窗口功能要多一些,这是因为多了一个菜单栏。

当程序员准备创建自定义的 Win32 应用程序时,在“项目”选项卡中选择“Win32 Application”,选择创建一个空的工程。选择“文件→新建”菜单后,在“新建”对话框的“文件”选项卡中选择 C++ Source File。在源程序的入口函数 WinMain 中需要使用大量的 API 函数,来编写程序员自定义的 Win32 应用程序。

1.2.4 创建 MFC Windows 应用程序

用 Visual C++ 的 MFC 来创建 Windows 应用程序窗口的步骤如下。

(1) 打开 VC++ 6.0,选择“文件→新建”菜单项,系统弹出“新建”对话框,单击“工程”选项卡→MFC AppWizard(exe),弹出如图 1.2.5 所示的对话框。工程编辑框中输入工程的名称,如“框架”,位置编辑框中输入保存该工程文件的路径,单击“确定”按钮。MFC 应用程序向导将通过 6 个步骤生成 Windows 应用程序框架。

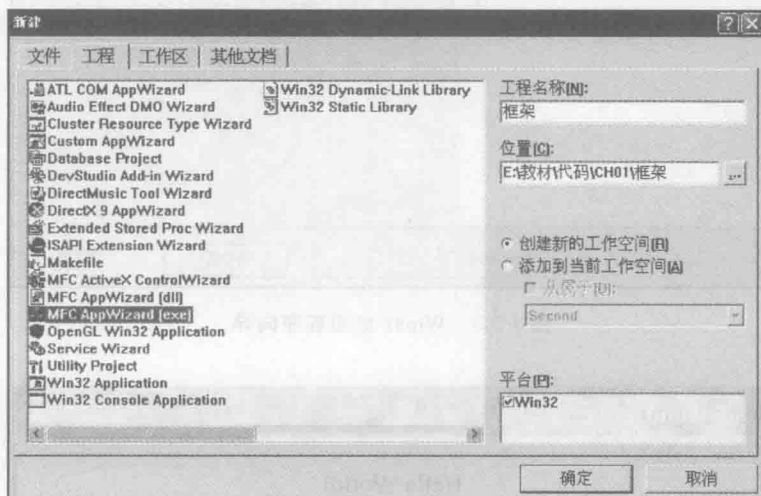


图 1.2.5 MFC Windows 应用程序新建工程

(2) 在弹出的“MFC 应用程序向导-步骤 1”对话框中,选择要创建的 MFC 应用程序类型,如图 1.2.6 所示。本例中选择建立单文档应用程序,默认选中“文档/查看体系结构支持”复选框,单击“下一步”按钮。

(3) 在弹出的“MFC 应用程序向导-步骤 2”对话框中,如图 1.2.7 所示,在询问是否要包含数据库时,本例选择“否”单选按钮,单击“下一步”按钮,则 Windows 应用程序框架不能使用数据库。如果 Windows 应用程序需要使用数据库,则在“MFC 应用程序向导-步骤 2”对话框中选择“查看数据库使用文件支持”单选按钮,然后单击“数据源”按钮,在弹出的对话框中选择数据库连接的数据源名称,单击“确定”按钮后回到如图 1.2.7 所示的对话框。

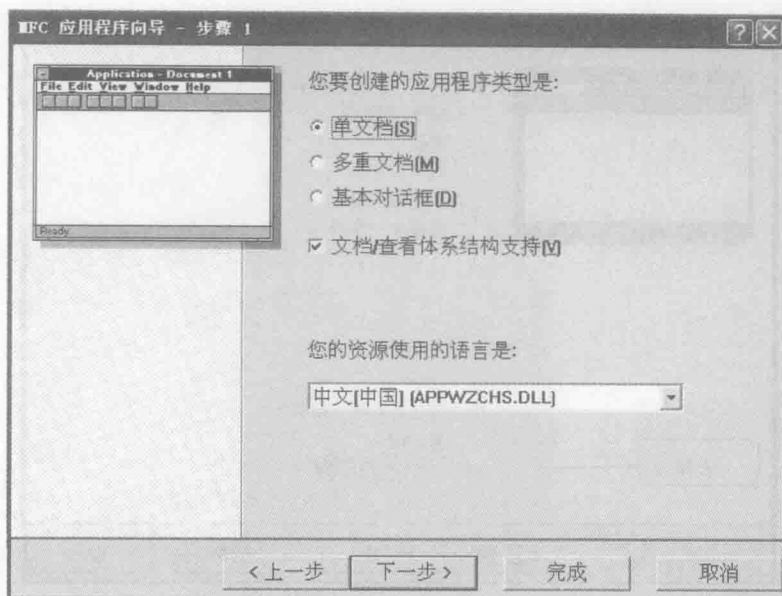


图 1.2.6 MFC 应用程序向导-步骤 1

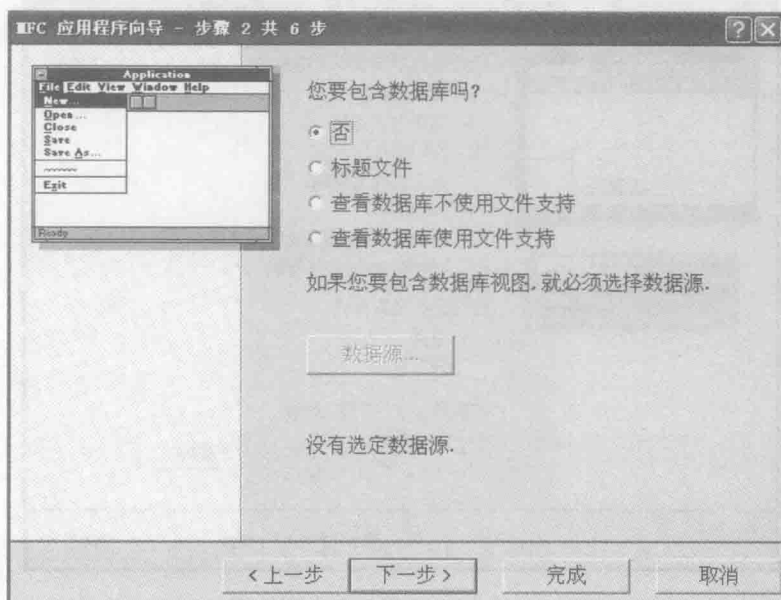


图 1.2.7 MFC 应用程序向导-步骤 2

(4) 在弹出的“MFC 应用程序向导-步骤 3”对话框中,取消选中“ActiveX 控件”复选框,如图 1.2.8 所示。确保默认选中“没有,不需要”复选框,单击“下一步”按钮。

(5) 在弹出的“MFC 应用程序向导-步骤 4”对话框中,取消选中“打印和打印预览”复选框,如图 1.2.9 所示,单击“下一步”按钮。Windows 应用程序窗口中将没有“打印和打印预览”菜单命令子项。若不取消选中“打印和打印预览”,则会出现这个菜单命令子项。

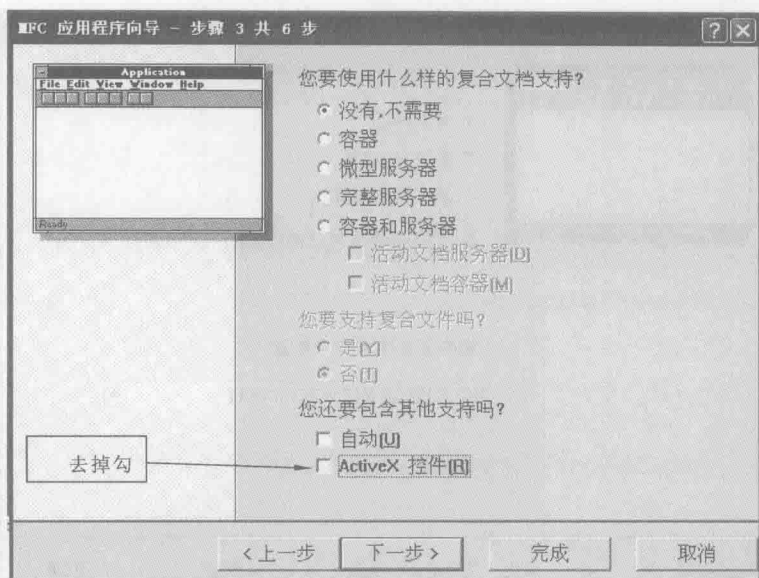


图 1.2.8 MFC 应用程序向导-步骤 3

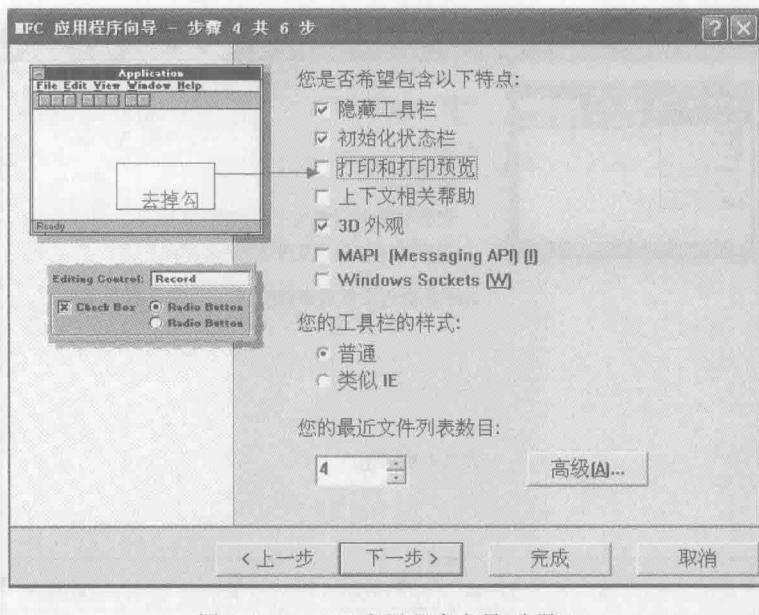


图 1.2.9 MFC 应用程序向导-步骤 4

(6) 在弹出的“MFC 应用程序向导-步骤 5”对话框中,接受所有的默认设置,如图 1.2.10 所示。也可以根据个人喜好选择不同的窗口风格,不生成源文件备注,使用 MFC 库作为静态的动态链接库(dynamic link library,DLL)。单击“下一步”按钮。

(7) 在弹出的“MFC 应用程序向导-步骤 6”对话框中,显示 MFC 应用程序向导创建的 4 个类,如图 1.2.11 所示,单击“完成”按钮。在弹出的如图 1.2.12 所示的对话框中,显示了创建工程的相关信息,在对话框中单击“确定”按钮。