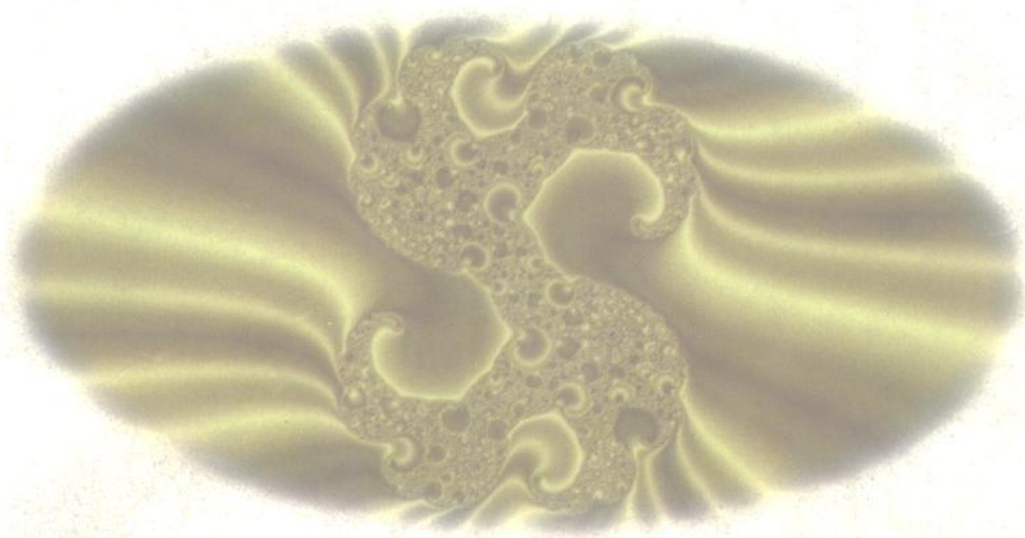




Visual C++ 4.2

编程实践指要

穆宗学 章 江 肖 蓉 编著

中国铁道出版社

Visual C++4.2 编程实践指要

穆宗学 章 江 肖 蓉 编著

中 国 铁 道 出 版 社

1997年·北京

(京)新登字 063 号

内 容 简 介

JS194/20

本书从实践的角度介绍了 Visual C++ 4.2 软件包的使用方法及其编程技巧。除基本概念外,本书重点论述了各种编程技术专题,如应用程序框架及交互界面开发、数据库应用开发、动态链接库的开发与调用、汇编语言使用、串口通信程序开发、位图格式及显示、多线程程序开发,并行数据采集和数字图象实例等。每个专题均给出了经过验证的实例,读者可直接或稍经修改后加以使用。值得指出的是,这些专题的技术及实例也可在 Visual C++ 的较低或更高版本中使用。

本书适于各类 Visual C++ 的使用及编程开发人员使用,也可作为各大专院校计算机及其相关专业师生的参考用书。

图书在版编目(CIP)数据

Visual C++ 4.2 编程实践指要/穆宗学等著. —北京:中国铁道出版社,1997.8
ISBN 7-113-02747-4

I. V… I. 穆… II. C 语言-程序设计 N. TP. TP312C

中国版本图书馆 CIP 数据核字(97)第 16364 号

Visual C++ 4.2 编程实践指要

穆宗学 章 江 肖 蓉 编著

中国铁道出版社出版发行

(100054,北京市宣武区右安门西街 8 号)

北京市燕山联营印刷厂印 各地新华书店经售

1997 年 8 月第 1 版 第 1 次印刷

开本:787×1092 1/16 印张:15 插页:1 字数:358 千字

印数:1—5000 定价:28.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社发行部调换。

前 言

VC++的若干特点及发展概要

Microsoft Visual C++简称VC++,是由Microsoft C发展演变而来的。国内见到的Microsoft C的早期版本有C 5.0、C 6.0和C/C++7.0。从C/C++ 8.0开始,改称Microsoft Visual C++。国内见到的版本有VC++1.0、VC++1.5、VC++1.52、VC++2.0、VC++4.0、VC++4.1和VC++4.2。VC++1.0、VC++1.5和VC++1.52有两个版本,一个版本支持16位的操作系统(DOS、WINDOWS3.2及WINDOWS3.2以下版本),另一个版本是为WINDOWS NT而写的,支持32位的操作系统(WINDOWS NT和WINDOWS 95)。

有人说,自VC++2.0开始,以后的版本放弃了对16位操作系统的支持。这样说并不全面,为此有必要提一下WIN32s,它是16位操作系统的32位扩展,它允许16位的操作系统通过一定的接口关系调用WIN32 API的一个子集。这就允许增加32位扩展的16位的操作系统可以执行32位的应用程序。VC++2.0、VC++4.0、VC++4.1都支持这种应用程序的开发。但值得注意的是,VC++4.2放弃了对WIN32s的支持。

正象Microsoft宣布的那样,它正在放弃对16位操作系统的支持,这就意味着开发16位的应用程序不再能使用新版的Microsoft的开发工具。

可以说VC++有两大特点:一个是可视化编程,另一个是面向对象。可视化通过一系列的编程工具及其丰富的功能来实现,它使WINDOWS应用程序的编程变得非常直观;而基类库又使面向对象的程序的编程变的非常简单。简言之,VC++使面向对象的WINDOWS应用程序的编程变得容易和迅速。

VC++4.2的新特点

VC++4.2的新特点主要包括下述几点:

- Internet 编程支持 MFC 打包了新的 Win32 Internet (WinInet)和 ActiveX 技术,使 Internet 的编程更容易;
- 标准 C++库 标准 C++库已被加入到 Visual C++ 4.2 中;
- ODBC 特征的增强 对 MFC ODBC 类作了重大改进,这些改进包括多线程支持;
- 更新的 Win32 SDK 文档 在线书籍包含 Windows NT 4.0 Beta 2 版发行的 Win32 SDK 文档;
- 彻底放弃对 16 位操作系统的支持 Visual C++ 4.2 不支持 Win32s 应用程序的开发。

VB 与 VC++ 的关系

VISUAL BASIC 以其可编译性、编程的简单性、编译的迅速性在近几年风靡全球。前几年流行 16 位的 VB,即 VB3.0;目前流行 32 位 VB,也就是 VB4.0、VB4.5 和 VB5.0。那么,VB 与 VC++ 之间是一个什么样的关系呢?

可以打一个不十分恰当的比喻,如果将编制成的应用程序比作一栋大楼,那么 VB 是用大块预制板建楼,而 VC++ 则是使用沙子、水泥和小块预制板建楼。

VB 是一种高级语言,是从 BASIC 语言演变而来的,它改变了 BASIC 语言解释性的弊端,成为编译 BASIC,提高了程序的执行速度。BASIC 语言是一种及其简单易学的语言,用起来非常简单方便。VB 对数据库的操作非常方便而有效,比较适合开发小型管理软件及数据库管理软件。

VB 并非完全具有面向对象的特点,其最重要的缺陷是没有可继承性。VB 对 WINDOWS 的接口、对硬件的低级访问以及计算能力都很简单、笨拙和低效。据有关资料统计,其代码效率只有较好的 C 语言(包括嵌入汇编语言)代码的十分之一。

VC++ 是一种广谱性编程语言,它覆盖从低级到高级的整个编程语言范围。可以只在低级,也可只在高级,或在整个编程语言范围内编程。它具有完整的面向对象的特点。与 WINDOWS 系统有着非常完美的接口关系。易于进行硬件低级访问。非常适合开发大型应用程序。可以毫不夸张地讲,VC++ 是一种万能编程语言。

正是由于 VB、VC 的不同特点,决定了其不同的用途。对于初涉计算机编程,开发小型应用程序的非专业人员,以使用 VB 为宜;对于开发大型应用程序的人员或专业人员来讲,VC 当然是恰当的选择。

使用 VB 编程的人员,需要使用、有时甚至是不得不使用 VC++ 语言(如需对硬件直接、低级访问时,编制设备驱动程序时)。这时只要将用 C++ 语言编写的程序写成动态链接库,VB 直接调用动态链接库就可以了。

本书的编写宗旨、读者对象

作者多年从事 16 位应用程序的开发,最初转到开发 32 位应用程序时,遇到很多困难。书店里虽然有很多关于 32 位应用程序编程方面的书,但多是介绍如何使用 MFC 的,也就是介绍如何使用 VC++ 的高级编程语言部分的。几年来,作者先是寻找,后是酝酿写一本有关 VC++ 中高级语言与低级语言的结合部分以及低级语言部分的书。本书力图贯彻这样一个思想:高级语言部分给读者一个完整、清晰而概括的一个概念,重点讲述高级和低级语言的结合部分及低级部分,侧重于实用性,可操作性。书中附有大量完整的例子,而这些例子都是用调试无误的程序清单嵌入到书中去的。

书中将作者编制动态链接库、硬件直接访问、设备控制、数据采集、图象处理、数据库操作及多线程编程等方面的一些体会写出来,力图使读者能迅速编写出实用的 32 位应用程序。本书适合下述读者:

- 用 VB 编程,但须调用 C、C++ 模块或直接硬件访问;

- 直接硬件访问、设备控制和数据采集；
- 用 VC++ 进行数据库操作；
- 用 VC++ 编制图象的显示、编辑和处理应用程序；
- 用 VC++ 开发串行通信应用程序；
- 使用 VC++ 但又使用汇编语言的读者；
- 广义上讲，适用于利用 VC++ 编制 32 位应用程序的读者。

本书不适合下述读者：

- 开发 16 位 Windows 应用程序；
- 开发 Dos 应用程序；
- 开发 16 位 VB(VB3.0)使用的动态链接库。

各章节内容概述

全书共分十一章。前四章对 VC++ 编程进行整体的介绍。第五到第十章分专项进行介绍。最后一章介绍几个应用专项的综合应用实例。

第一章介绍如何用 AppWizard、AppStudio、ClassWizard 快速生成 Windows 应用程序。

第二章介绍 VC++4.2 开发平台的使用方法。

第三章概述微软基类库(MFC)及其分类，使读者对基类库有一个整体的印象。

第四章介绍如何使用 MFC 中窗口类、控制类、菜单类、对话框类、以及文档和视类编写应用程序。

第五章介绍如何利用 ODBC 对数据库进行操作。

第六章介绍各种动态链接库的编程方法，VB 对动态链接库的调用，VC 对动态链接库的动态调用和静态调用，并给出完整的样例程序。

第七章介绍 VC++4.2 与汇编语言的混合编程问题。分两种情况介绍：一种是单独模块汇编语言的使用，另一种是嵌入式汇编语言的使用。每一种情况都给出大量完整的例子。

第八章介绍开发串行通信应用程序及应注意的事项。

第九章介绍位图文件的格式及显示方法。

第十章介绍多线程应用程序的开发方法。

第十一章介绍两个综合应用实例：一个是并行数据采集，另一个是数字图象处理。在这两个实例中用到在线汇编语言、动态链接库、位图、多线程编程、VB 调用动态链接库和 VC 调用动态链接库的综合知识。

作者

1997 年 4 月

目 录

1 创建第一个应用程序	1
1.1 AppWizard 的应用	2
1.1.1 AppWizard 简介	2
1.1.2 用 AppWizard 创建程序	3
1.1.3 编译、运行程序	5
1.2 AppStudio 的应用	8
1.2.1 AppStudio 简介	8
1.2.2 运用 AppStudio 编辑主对话框	8
1.2.3 运用 AppStudio 编辑图标	10
1.2.4 编译、运行程序	12
1.3 ClassWizard 的应用	12
1.3.1 ClassWizard 简介	12
1.3.2 为 Display 按钮编写代码	13
1.3.3 为 Close 按钮编写代码	14
1.3.4 编译、运行程序	16
2 Visual C++4.2 开发工作台简述	17
2.1 File 菜单	17
2.2 Edit 菜单	20
2.3 View 菜单	22
2.4 Insert 菜单	25
2.5 Build 菜单	25
2.6 Debug 菜单	27
2.7 Tools 菜单	28
2.8 Windows 菜单	29
2.9 Help 菜单	30
3 MFC(微软基类库)概述	31
3.1 什么是 MFC	31
3.2 MFC 的结构层次	31
3.3 MFC 分层概述	32
3.3.1 根 类	32

3.3.2	应用程序体系结构类	32
3.3.3	窗口、对话框、控制类	33
3.3.4	绘图和打印类	34
3.3.5	简单数据类型类	35
3.3.6	数组、列表和映象类	35
3.3.7	文件和数据库类	36
3.3.8	Internet 和网络类	37
3.3.9	OLE 类	37
3.3.10	调试和异常类	38
3.3.11	宏和全局函数	39
3.3.12	MFC 与 C 语言 API 的关系	39
4	应用程序框架及交互界面的开发	41
4.1	控制类及其编程的实现	41
4.1.1	创建工程	41
4.1.2	控件的设置	41
4.1.3	引入变量	43
4.1.4	编写代码	44
4.2	实现菜单和对话框控制	49
4.2.1	创建工程	49
4.2.2	窗口的可视化设计	49
4.2.3	引入变量	53
4.2.4	编写代码	53
4.2.5	对话框类的定制	57
4.3	文档和视简介	60
4.3.1	单文档界面和多文档界面	60
4.3.2	文档和视	60
4.4	单文档界面	61
4.4.1	创建工程	61
4.4.2	对主窗口进行可视化设计	61
4.4.3	引入数据成员	63
4.4.4	编写代码	65
4.5	多文档界面	70
4.5.1	创建工程	70
4.5.2	进行可视化设计	72
4.5.3	引入数据成员	73
4.5.4	编写代码	77
4.6	滚动视	84
4.6.1	创建工程	84

4.6.2	修改视窗滚动尺寸	85
4.6.3	编写代码检测滚动情况	86
5	对数据库进行操作	88
5.1	ODBC 简介	88
5.2	SQL 基础	89
5.2.1	关系对象	90
5.2.2	SQL 常用语句介绍	90
5.3	MFC ODBC 数据库类	93
5.4	编程前的准备工作	94
5.5	用 AppWizard 生成数据库应用程序框架	96
5.6	理解 AppWizard 生成的应用程序框架	99
5.6.1	记录集合类	99
5.6.2	记录视类	102
5.6.3	文档类	104
5.6.4	查看 AppWizard 生成的资源	104
5.7	设计对话框表单	108
5.8	连接表单控件和记录数据成员	113
5.9	使用组合框控件代替编辑控件	115
5.10	增加、删除和恢复记录	116
5.10.1	修改用户界面	116
5.10.2	数据库更新的基本方法	117
5.10.3	编制 Add 命令处理函数	118
5.10.4	编制 Delete 命令处理函数	120
5.10.5	编制 Refresh 命令处理函数	121
6	动态链接库的开发与调用	123
6.1	动态链接库的编程要点与分类	123
6.1.1	有关动态链接库的一些概念	123
6.1.2	编制动态链接库需特别注意的四大要素	124
6.1.3	DLL 的分类	125
6.2	Win32DLL 的编程	125
6.2.1	用模块定义文件进行输出说明的 DLL	126
6.2.2	用 dllexport 属性进行输出说明的 DLL	128
6.3	VC++ 对 DLL 的调用	130
6.3.1	对用模块定义文件进行输出说明的 DLL 的静态调用	130
6.3.2	对用 dllexport 属性进行输出说明的 DLL 的静态调用	132
6.3.3	动态调用	134
6.4	VB 对 DLL 的调用	136

6.4.1	定制窗口	137
6.4.2	说 明	137
6.4.3	DLL 调用过程的编写	138
6.4.4	产生 EXE 文件	138
7	汇编语言的使用	139
7.1	在线汇编语言	139
7.1.1	___asm 关键词的使用	139
7.1.2	在___asm 块使用汇编语言	142
7.1.3	在___asm 块中使用 C、C++	143
7.2	单独模块汇编语言的使用	150
7.2.1	编译方法及调用过程介绍	150
7.2.2	C 语言对独立模块汇编语言调用的分类介绍	153
8	串口通信程序的开发	130
8.1	串行通信的内部机制	160
8.1.1	通信机制	160
8.1.2	从接收队列中获取数据的几种方式	160
8.2	有关通信的介绍	161
8.2.1	通信资源句柄	161
8.2.2	修改通信资源设置	161
8.2.3	通信资源配置	162
8.2.4	通信事件	162
8.2.5	扩展函数	164
8.3	读写通信资源	164
8.3.1	读写操作	164
8.3.2	重叠操作	164
8.3.3	超 时	165
8.3.4	通信错误	166
8.4	通信函数的使用	166
8.4.1	配置一个通信资源	166
8.4.2	监视通信事件	167
9	位图的格式及显示	170
9.1	BMP 文件的结构	170
9.1.1	文件头	170
9.1.2	信息头	171
9.1.3	色彩表	172
9.1.4	位图阵列	172

9.2 位图的显示	174
10 多线程程序的开发	189
10.1 使用 C 运行时库和 Win32 API 编制多线程应用程序	189
10.1.1 使用 C 运行时库	190
10.1.2 使用 WIN32 API	190
10.1.3 编写多线程应用程序应注意的问题	190
10.1.4 线程本地存储(TLS)	190
10.2 利用 MFC4.2 编制多线程应用程序	193
10.2.1 建立主线程	194
10.2.2 用 AfxBeginThread()函数创建辅助线程	195
10.2.3 用 CreateThread()函数创建辅助线程	197
10.2.4 完整的清单	201
11 并行数据采集和数字图象处理实例	207
11.1 一种 32 位 WINDOWS 的并行数据采集技术	207
11.1.1 前言	207
11.1.2 并行数据采集的原理、方法及程序实现结构	207
11.1.3 动态链接库的编写	208
11.1.4 VC 应用程序对动态链接库的调用	215
11.1.5 VB 应用程序对动态链接库的调用	219
11.2 数字图象处理	220
11.2.1 编制动态链接库	221
11.2.2 编制调用程序	223
11.2.3 程序的运行结果	225
参考文献	226
附录图	227

1

创建第一个应用程序

本章将简要介绍如何用 VC++4.2 的开发工具,建立自己的 32 位 WINDOWS 应用程序。目的是给读者一个用 VC++ 编程的整体的轮廓性的概念,并使读者体会到编程过程的简单方便性。在创建新的应用程序之前,首先在 Microsoft Visual C++4.2 程序组中找到 Microsoft Developer Studio(微软开发工作室)图标,双击该图标,使其启动。这时将会看到如图 1.1 所示的画面。



图 1.1 Microsoft Developer Studio

Microsoft Developer Studio 是集成性的多语言开发环境,它为编程人员开发、调试应用程序的整个过程提供了有效的管理手段。作为一种框架结构,它不仅适用于 Visual C++,而且适用于 Microsoft 公司的 Fortran、Visual Test、Development Library 及 Visual J++ 语言,使这些语言开发工具具有统一的交互界面。与过去低版本的 Visual C++ 交互界面相比,Microsoft Developer Studio 便于实现多语言开发及混合语言编程,使软件开发工具在通用性上迈进了一大步。

在安装 Visual C++ 4.2 的过程中,安装程序将提示用户选择开发环境的形式:一种是类似于 Visual C++ 2.0 或更低版本的 C 开发环境;另一种则是我们现在所选择的 Microsoft Developer Studio(适用于 Visual C++ 4.0 或更高版本 C 语言的开发环境)。如果在此选定的是 Microsoft Developer Studio 的形式,则安装完软件后,启动 Visual C++ 4.2,首先将出现

如图 1.1 的桌面,在此桌面中可发现 Visual C++ 图标被高亮度显示。然后再进入到如图 1.2 的主窗口当中。如果在安装过程中选择了不同的开发环境的形式,那么 Visual C++4.2 的主窗口外观将与图 1.2 略有不同,但其功能和用法是完全一致的。

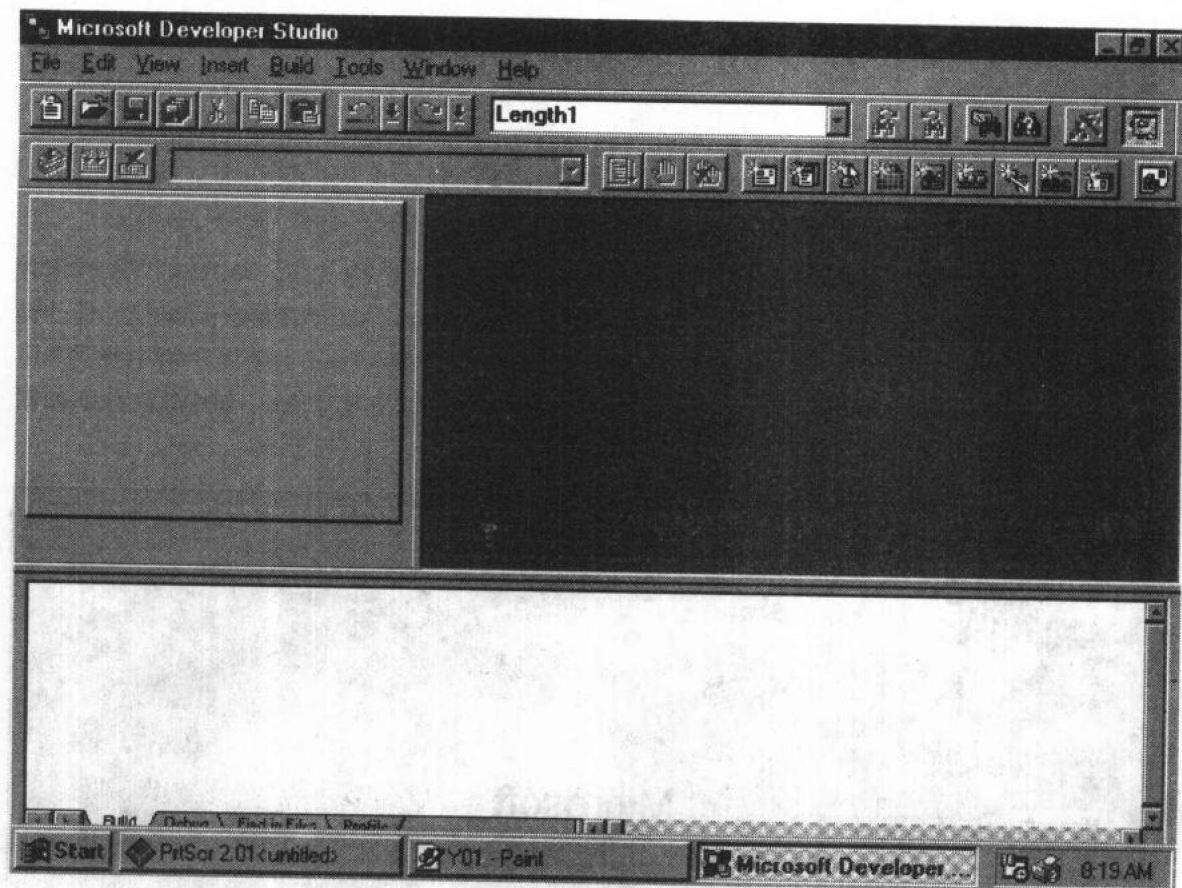


图 1.2 Microsoft Developer Studio 的主窗口

Visual C++ 4.2 为编程人员提供了一整套方便快捷的编程工具,其中最为基本的是以下三个导航系统:AppWizard(应用程序框架生成器)、AppStudio(应用程序资源编辑器)、ClassWizard(类编辑管理器)。几乎编写每一个应用程序都将用到这三大导航系统,利用它们可以迅速编写 Windows 应用程序。本章将主要介绍怎样利用导航系统编写 Visual C++4.2 应用程序。

1.1 AppWizard 的应用

1.1.1 AppWizard 简介

AppWizard 是 Visual C++4.2 为编程人员编写 Windows 应用程序而提供的最便捷、最直观、最迅速的编程工具之一。应用 AppWizard 可生成应用程序的基本框架而不需编写一句代码,从而把程序员们从大量繁琐的重复性编程劳动中解脱出来。

1.1.2 用 AppWizard 创建程序

有了 AppWizard 作为我们的向导,就可以开始以下的步骤:

1. 从主窗口的 File 菜单项中选择 New 命令,这时 Visual C++ 将显示如图 1.3 所示的 New 对话框。

2. 在 New 对话框中,选择 Project Workspace (工程工作区) 项,再单击 OK 按钮。此时,将会显示如图 1.4 所示的 New Project Workspace 对话框。

3. 在 New Project Workspace 对话框的左边 Type 列表框中列出了可创建的工程文件类型:其中 MFC AppWizard (exe) 表示创建基于 MFC 的可执行程序;MFC

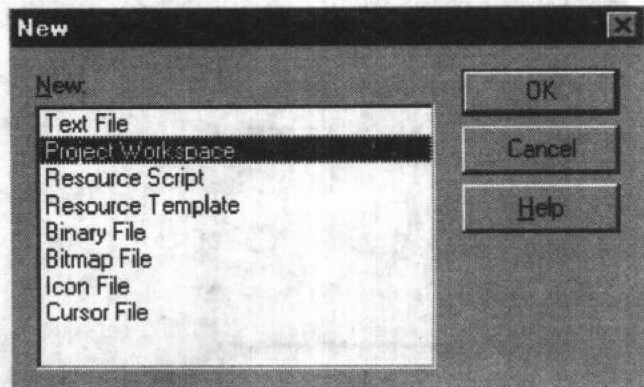


图 1.3 New 对话框

AppWizard [dll] 表示创建基于 MFC 的动态链接库;OLE ControlWizard 表示创建 OLE 控制;Application 表示创建通用性工程文件;Dynamic-Link Library 表示创建基于 C 的动态链接库;Console Application 表示创建字符型窗口工程文件。最常用的工程类型是可执行程序 (exe) 和动态链接库 (dll)。本章将以创建可执行程序为例,说明 AppWizard 的用法,因此在此列表框中选定 MFC AppWizard (exe) 项,即通知 Visual C++ 将要创建的是一个基于 MFC 的 exe 程序。

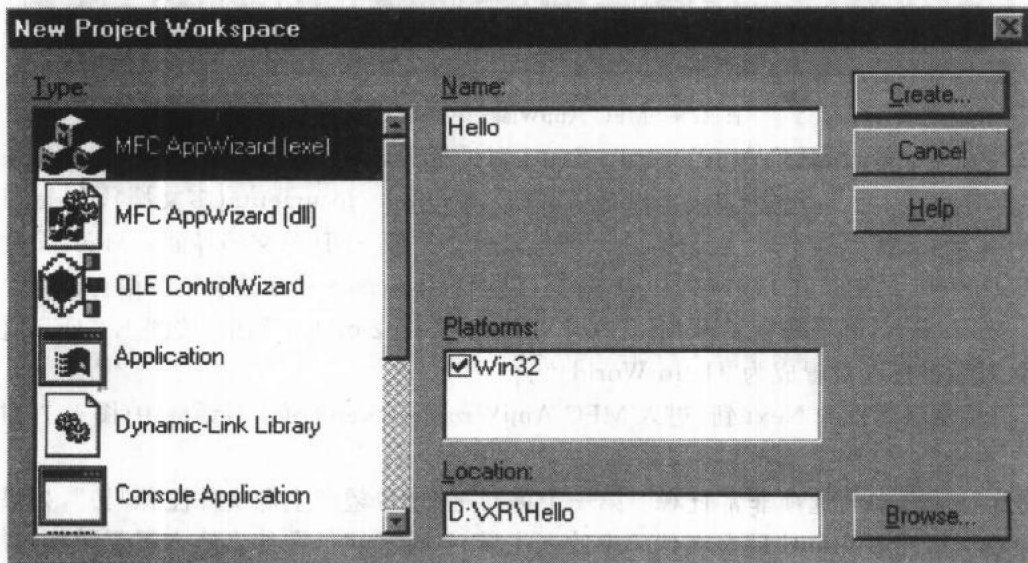


图 1.4 New Project Workspace 对话框

在 New Project Workspace 对话框的右上部 Name 框中键入所起的工程名,这里键入 Hello。在右下部 Location 框中所显示的是目录,可通过 Browse 按钮选择想存贮工程文件的目录。Visual C++ 将自动在所选目录下以 Hello 为名创建一个子目录。将工程中所有的文件

存入其中。Platforms 项是选择所支持的开发平台, Visual C++ 只安装了 Win32 库和工具, 因此无需改动此项。

4. 在选定了工程的类型、名称、存贮目录和支持平台后, 单击 Create 按钮, 此时, 系统进入到如图 1.5 所示的 MFC AppWizard—Step1 对话框中。

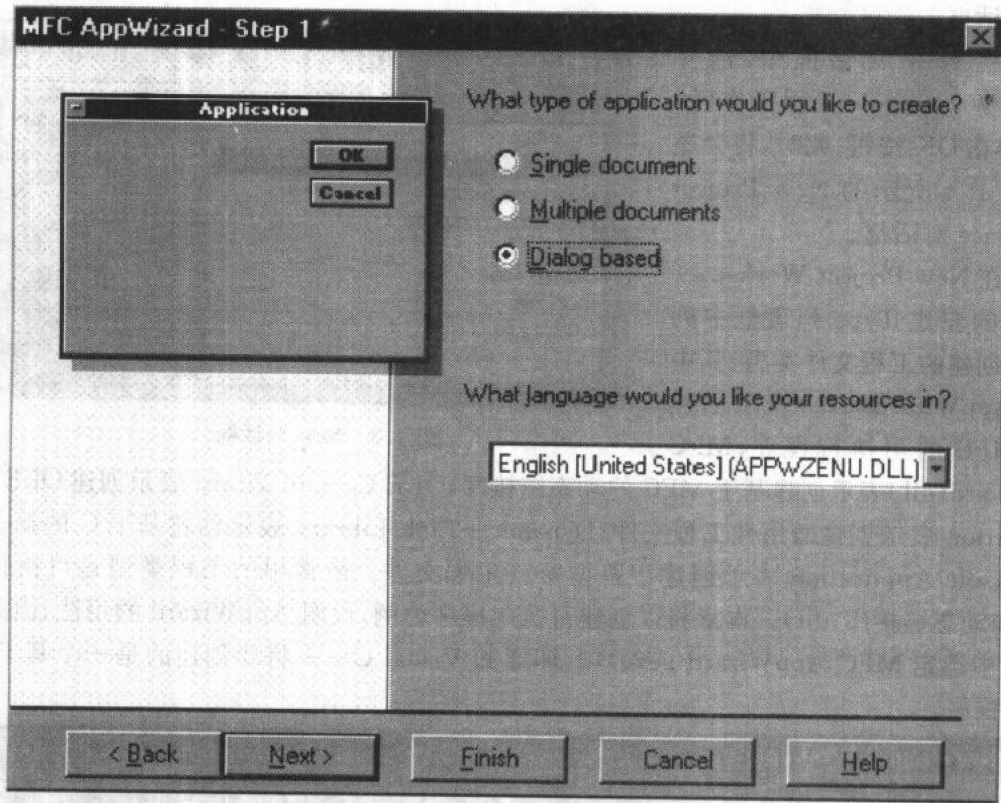


图 1.5 MFC AppWizard—Step1 对话框

在当前所显示的 MFC AppWizard—Step1 对话框中, 可以确定所要创建程序的体系结构类型和资源语言。可从 Single document (单文档), Multiple documents (多文档), Dialog based (基于对话框) 三种模式中任选一种, 这三种模式分别对应 SDI (单文档界面), MDI (多文档界面), 以及简单对话框。在 Hello 程序中, 我们选择 Dialogbased 项和缺省资源语言设置。

5. 按下 Next, 屏幕将会显示 MFC AppWizard—Step2 of 4 对话框, 按图 1.6 所示进行设置, 并将对话框标题设置成为“Hello World!”。

6. 再单击对话框的 Next 钮, 进入 MFC AppWizard—Step3 of 4 对话框中(图 1.7)对话框中。

在此对话框中的选项非常直观。第一个提问是“您希望产生源文件注释吗?”, 如果选择“*Yes, please*”, AppWizard 就会在代码中放入注释, 告诉程序员在什么地方需要增加功能, 通常是“*TODO*”注释形式。同时, AppWizard 也用少量的注释语句来提示用户程序是如何运行的。选择“*No, Comment*”将不产生任何注释。

对第二个提问“您希望采用什么方式将 MFC 连入应用程序”的回答是“*As a shared DLL*”。即采用动态链接方式, 而非静态链接库, 从而使 EXE 文件的长度变得小一些。

7. 单击 Next 按钮进入到 AppWizard 的最后一个对话框: MFC AppWizard—Step4 of 4

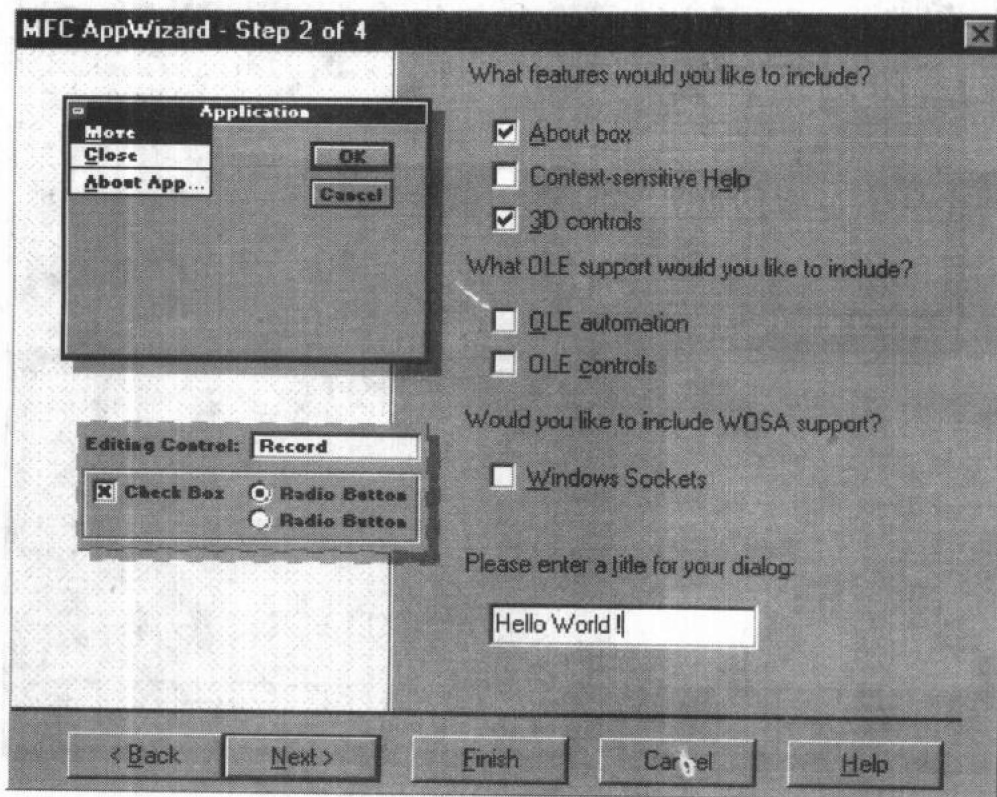


图 1.6 MFC AppWizard—Step2 of 4 对话框

对话框中。(见图 1.8)

此对话框列出了将要产生的新类和文件名,在此 AppWizard 为 Hello 应用程序建立了两个新类:CHelloApp(应用程序类)和 CHelloDlg(对话框类)。如果此时,你对于前面的设置想做一些修改,可以按 Back 键回到前面的对话框中去重新设置。同样,在前面的每一个对话框中都包含有 Back 和 Next 按钮,以便程序员随时回到前面的设置中进行修改。

8. 如果一切如你所愿,你就可以按下 Finish 钮。此时将出现如图 1.9 所示的 New Project Information 对话框。这个对话框中的信息是对你在前面各步中所做工作的总结。但是,到此为止,AppWizard 实际上还并没有为你创建任何东西。如果你此时想改变主意的话,按下 Cancel 按钮,将会回到前面的对话框中,你仍可以重新进行设置。

9. 在 New Project Information 对话框中,按下 OK 按钮,AppWizard 才真正开始生成应用程序。

注意:在运用 AppWizard 生成应用程序的过程中,如果你在 MFC AppWizard—Step 1 中选择的体系结构类型不是 Dialog based,而是 Single document 或 Multiple documents,AppWizard 将显示不同数量和不同选项的设置对话框,对于这两种情况,我们将在本书后面部分中提及。

1.1.3 编译、运行程序

虽然你还没有编写任何一句源代码,但 AppWizard 已经为你产生了一个真正的 32 位

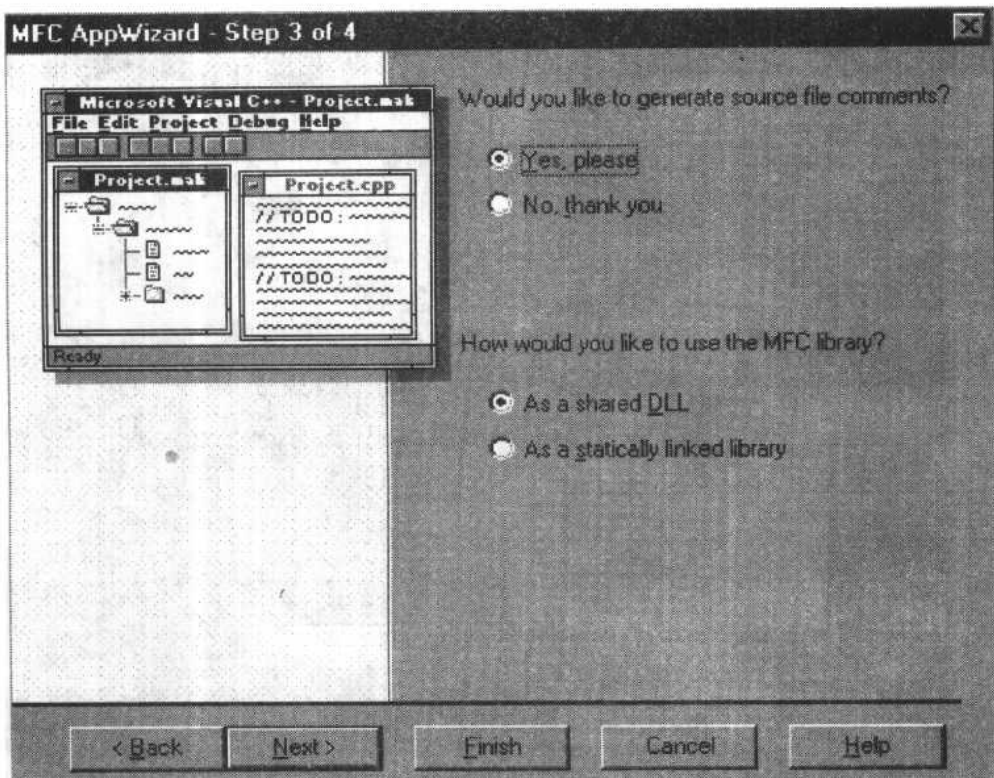


图 1.7 MFC AppWizard—Step3 of 4 对话框

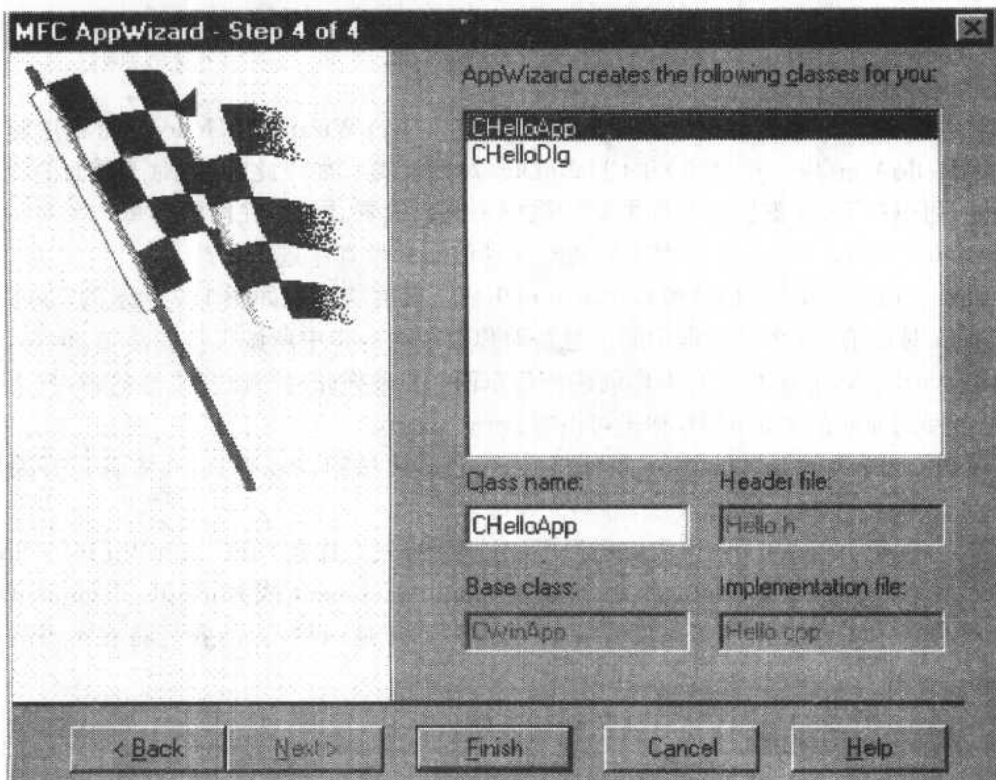


图 1.8 MFC AppWizard—Step4 of 4 对话框