

VISUAL C++4.0

易学活用

王强 王利强 编写
廖果 周兰邻

WINDOWS
95 版

Windows 实用技术

Visual C++ 4.0 易学活用

(用Visual C++ 4.0 编写Windows 95 程序)

王 强 王利强 廖 果 周兰邻 编著

Windows 实用技术

成都出版社

编 者
1995 年

(川)新登字011号
CHENGDUCHUBANSHE

Windows 实用技术 Visual C++ 4.0 易学活用

作者: 王强 王利强 廖果 周兰邻
责任编辑: 乔文 春晓(特邀)
封面设计: 唐利明
技术设计: 晓毅
责任校对: 谢辉

出版发行: 成都出版社
地址: 中国·成都市西一路路北一段182号
邮政编码: 610031
电话号码: (028) 7765071 7783841
经销: 各地新华书店
排版: 四川中外科技文化交流中心
印刷: 四川大学印刷厂
版次: 1996年5月第1版
印次: 1996年5月第1次印刷
开本: 787×1092mm 1/16
印张: 23.4
字数: 392.7千
印数: 1—3000册
书号: ISBN 7-80575-022-x/0·1
定价: 36.00元

(版权所有, 翻印必究)

目 录

第一章 编写第一个Visual C++程序	(1)
§ 1.1 HELLO.EXE 程序	(1)
§ 1.2 Hello.EXE 程序的可视化设计	(10)
§ 1.3 为Hello.EXE 程序编写代码	(17)
§ 1.4 定制Hello.EXE 程序的图标	(23)
§ 1.5 打开已有的工程	(24)
§ 1.6 使用属性表(Properties Table)	(25)
第二章 VC 程序中的基本元素	(27)
§ 2.1 控制、属性、事件	(27)
§ 2.1.1 使用控制的例程	(27)
§ 2.1.2 创建工程	(29)
§ 2.1.3 可视化设计	(31)
§ 2.1.4 在编辑框及检查框中引入变量	(34)
§ 2.1.5 初始化对话框	(35)
§ 2.1.6 程序的退出	(37)
§ 2.1.7 程序中的文本显示	(38)
§ 2.1.8 设置检查框	(39)
§ 2.1.9 改动编辑框的内容	(41)
§ 2.2 OLE 控制与部件库	(45)
§ 2.2.1 OLE 控制的概念及例程	(45)
§ 2.2.2 创建工程	(46)
§ 2.2.3 注册Grid OLE 控制	(47)
§ 2.2.4 Table 程序的可视化设计	(49)
§ 2.2.5 Grid 控制中引入变量	(53)
§ 2.2.6 初始化Grid 控制	(54)
§ 2.2.7 在按钮事件中加入控制代码	(56)
§ 2.2.8 利用部件库增加OLE 控制	(59)
第三章 菜 单	(61)
§ 3.1 程序预览	(61)
§ 3.2 程序设计	(62)
第四章 工具条和状态条	(73)
§ 4.1 可视化设计	(73)
§ 4.1.1 创建项目文件	(73)
§ 4.1.2 功能预览	(74)
§ 4.1.3 添加画圆工具	(78)

§ 4.2	详细程序设计	(82)
§ 4.2.1	设计程序Shp 的选单	(82)
§ 4.2.2	将ID 与新工具连接	(82)
§ 4.2.3	设计Circle 选单	(83)
§ 4.2.4	声明视图类的数据成员	(84)
§ 4.2.5	声明文档类的数据成员	(84)
§ 4.2.6	设计Circle 选单项的UPDAFE_COMMAND_UI 消息	(85)
§ 4.2.7	设计Rectangle 选单的UPDATE_COMMAND_UI 消息	(86)
§ 4.2.8	初始化数据成员m_SelectedShapeOnToolbar	(86)
§ 4.2.9	往Circle 和Rectangle 选单项中添加代码	(87)
§ 4.2.10	设计View 类的WM_LBUTTONDOWN 事件	(88)
§ 4.2.11	绘制圆和矩形	(89)
§ 4.2.12	初始文档类的数据成员	(89)
§ 4.2.13	初始化视图类的数据成员	(90)
§ 4.2.14	设置文档修改标志	(91)
§ 4.2.15	向文件中写入或读出数据	(92)
第五章	对话框	(94)
§ 5.1	程序预览	(94)
§ 5.2	程序设计	(95)
§ 5.3	用户自定义对话框	(101)
第六章	鼠标和键盘	(109)
§ 6.1	鼠标程序	(109)
§ 6.2	程序设计	(109)
§ 6.3	程序设计	(110)
§ 6.4	键盘程序	(115)
第七章	绘制图形	(119)
§ 7.1	Graph(图形)程序	(119)
§ 7.1.1	创建Graph 程序的工程	(120)
§ 7.1.2	Graph Program 的可视化设计	(120)
§ 7.1.3	为Exit 按钮的BN_CLICKED 事件加上代码	(121)
§ 7.1.4	使用定时器(timer)	(122)
§ 7.1.5	WM_PAINT 事件	(125)
§ 7.1.6	IDD_CUSTOM_DIALOG 对话框的可视化设计	(131)
§ 7.1.7	修改Draw Graphics 按钮的BN_CLICKED 事件的代码	(134)
§ 7.1.8	初始化单选按钮	(135)
§ 7.1.9	给CGraphDlg 类添加成员变量并初始化	(136)
§ 7.1.10	给定时器事件附上代码	(138)
§ 7.1.11	修改与WM_PAINT 相连的代码	(139)
§ 7.1.12	修改定时器的安装设置	(141)

§ 7.2	使用WinG 库来避免闪烁	(142)
§ 7.3	使用不同的颜色来画圆	(143)
第八章 文本显示 (144)		
§ 8.1	MyFnt 程序	(144)
§ 8.2	创建MyFnt 程序的工程	(145)
§ 8.3	MyFnt 程序的可视设计。	(145)
§ 8.3.1	添加一个变量到编辑盒	(146)
§ 8.3.2	添加代码到Exit 按钮的BN_CLICKED 事件。	(146)
§ 8.3.3	添加代码到编辑盒的EN_CHANGE 事件	(147)
§ 8.4	WM_PAINT 事件的代码:	(147)
§ 8.5	显示文本。	(148)
§ 8.6	在运行过程中时改变字体	(150)
§ 8.7	添加一个变量到无线按钮	(151)
§ 8.7.1	无线按钮的初始化	(152)
§ 8.7.2	用Size Radio 钮改变字体	(153)
§ 8.8	显示带有阴影的文本	(155)
§ 8.8.1	添加一个阴影检查框	(156)
§ 8.8.2	连接代码到阴影检查框	(156)
§ 8.8.3	在透明背景下显示文本	(159)
第九章 位图 (163)		
§ 9.1	将位图嵌入EXE 文件中。	(163)
§ 9.1.1	MyBMP 程序。	(163)
§ 9.1.2	创建MyBMP 程序的项目	(163)
§ 9.1.3	MyBMP 主窗口的设计	(164)
§ 9.1.4	菜单设计	(164)
§ 9.1.5	将菜单移到对话框并连接相应代码	(165)
§ 9.1.6	显示About Dialog 框	(166)
§ 9.1.7	在用户项目中插入位图	(166)
§ 9.1.8	变量m_hlInstance	(168)
§ 9.1.9	显示位图	(169)
§ 9.1.10	扩大位图	(171)
§ 9.1.11	在CMyBMPDlg 类中添加两个数据成员	(172)
§ 9.1.12	提取窗口的宽度和高度	(172)
§ 9.1.13	改变位图大小	(173)
§ 9.2	动态装入位图	(174)
§ 9.2.1	简介	(174)
§ 9.2.2	创建项目文件	(176)
§ 9.2.3	利用可视化工具设计主窗口	(176)
§ 9.2.4	菜单设计	(177)

§ 9.2.5	装入位图	(178)
§ 9.2.6	Open 菜单	(178)
§ 9.2.7	设计函数DibOpenFile()	(180)
§ 9.2.8	显示位图	(184)
§ 9.2.9	调整显示窗口	(187)
§ 9.2.10	动态装入和显示位图	(189)
第十章	单文档界面与多文档界面	(193)
§ 10.1	单文档界面程序	(193)
§ 10.1.1	CIRCLE 例程	(193)
§ 10.1.2	创建工程	(194)
§ 10.1.3	文档(Document)	(196)
§ 10.1.4	视图View 类的数据成员	(197)
§ 10.1.5	初始化文档类的数据成员	(198)
§ 10.1.6	初始化视图类的数据成员	(198)
§ 10.1.7	画圆	(199)
§ 10.1.8	屏幕的任意位置画圆	(200)
§ 10.1.9	文档的存储和装入	(202)
§ 10.1.10	改进程序Circle	(205)
§ 10.2	多文档界面	(208)
§ 10.2.1	概述	(208)
§ 10.2.2	MCIRCLE.EXE 例程	(208)
§ 10.2.3	创建MCIRCLE 程序的工程	(211)
§ 10.2.4	文档(DOCUMENT)类的数据成员	(213)
§ 10.2.5	视图(VIEW)类的数据成员	(214)
§ 10.2.6	显示圆环	(215)
§ 10.2.7	MCircle 程序的MDI 特色	(217)
§ 10.2.8	在任意位置画圆	(219)
§ 10.2.9	文档的存贮和调入	(220)
§ 10.2.10	同一文档的多重显示	(224)
§ 10.2.11	改进MCircle 程序	(227)
第十一章	多任务管理	(231)
§ 11.1	Taks 例程简介	(231)
§ 11.2	创建Tasks 例程的工程	(231)
§ 11.3	定制Tasks 例程的菜单	(233)
§ 11.4	Tasks 程序的可视化设计	(234)
§ 11.5	Tasks 例程的初始化	(236)
§ 11.6	执行后台任务	(237)
第十二章	文件操作	(241)
§ 12.1	档案(CAriche)类	(241)

§ 12.2	Arch 例程	(241)
§ 12.3	创建工程及可视化设计	(242)
§ 12.4	Arch 程序中的文件操作	(244)
第十三章	库文件	(247)
§ 13.1	软件模块	(247)
§ 13.2	创建项目文件	(247)
§ 13.3	详细设计	(248)
§ 13.3.1	生成Circle.cpp 和Circle.h 文件。	(248)
§ 13.3.2	定义CCircle 类。	(249)
§ 13.3.3	文件Circle.cpp	(249)
§ 13.3.4	建立库文件Circle.LIB 和测试程序Test.EXE。	(250)
§ 13.3.5	各控件的详细设计	(252)
第十四章	动态链接库	(256)
§ 14.1	何为DLL?	(256)
§ 14.2	生成DLL	(256)
§ 14.2.1	创建MyDLL.DLL 项目	(256)
§ 14.2.2	DLL 的头代码	(256)
§ 14.2.3	测试程序Test2。	(259)
第十五章	创建用户OLE 控制	(263)
§ 15.1	创建MyClock.ocx 控制的项目	(263)
§ 15.2	检测MyClock 控制。	(264)
§ 15.3	在MyClock 控制中显示文本	(266)
§ 15.4	在MyClock 控制中显示当前时间	(270)
§ 15.5	连续显示当前时间。	(271)
§ 15.6	使用BackColor 和ForeColor 特性	(277)
§ 15.7	设置MyClock 控制的初始大小	(279)
§ 15.8	添加用户特性。	(279)
§ 15.9	初始化UpdateInterval 特性	(281)
§ 15.9.1	初始化UpdateInterval 特性	(281)
§ 15.9.2	使用UpdateInterval 特性:	(282)
§ 15.9.3	UpdateInterval 特性值的合法性	(283)
§ 15.10	特性页	(284)
§ 15.11	添加栈事件	(289)
§ 15.12	添加用户事件。	(291)
§ 15.13	添加方法	(295)
§ 15.13.1	AboutBox()方法	(295)
§ 15.13.2	添加栈方法	(296)
§ 15.13.3	添加用户方法	(297)

§ 16.4	JukeW 程序的可视化设计	(304)
§ 16.5	引入变量并针对事件编程	(307)
第十七章	游戏与三维虚拟现实	(314)
§ 17.1	虚拟现实技术	(314)
§ 17.2	3-D Floor OLE 控制	(314)
§ 17.3	VR 程序	(315)
§ 17.4	注册OLE 控制	(319)
§ 17.5	可视化设计	(320)
§ 17.6	VR 程序中的Floor 文件	(321)
§ 17.7	编制VR 程序中的代码	(323)
	问题与练习答案	(340)

.....
.....
.....

第一章 编写第一个Visual C++程序

学习一门高级编程语言时,通常都以编写一个最简单的Hello. EXE.程序开始。本书中第一个VC++程序也是Hello. EXE,它是用C++所写的一个Windows 95程序,涉及到C++和Windows的很多特性。Hello. EXE程序与以往的程序相比,外观上有很大不同,功能上也有所改变,但更重要的是编写它非常简单、容易!

一般而言,编写VC++程序有两个步骤:一是可视化设计阶段,一是代码编写阶段。在可视化设计阶段,编程者设计程序的外观,这可以通过使用Visual C++自带的工具软件来实现,即用Visual C++的工具把各种不同的对象(诸如矩形按钮、滚动条和单选按钮等等)置放于程序窗口之中。在可视化设计阶段,编程者不用编写任何代码。在代码编写阶段,编程者使用Visual C++的文本编辑器和C++程序设计语言来为程序编写代码。

§ 1.1 HELLO. EXE 程序

前面提到过Hello. EXE程序与以前的Hello. EXE程序相比有很大的不同,其具体的外观及其所具有的功能如下:

1. 当运行Hello. EXE程序时,其窗口外观如图1.1所示。

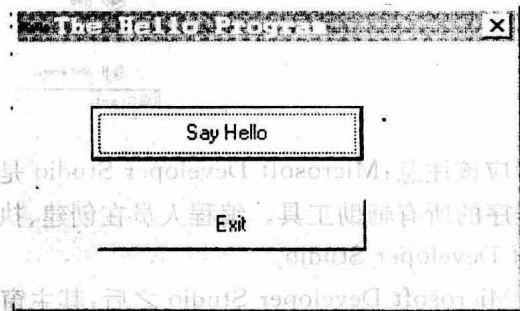


图1.1 Hello. EXE程序的窗口

2. Hello. EXE程序有两个矩形按钮:Say Hello按钮和Exit按钮。当用户点击Say Hello按钮时,将会出现一个HELLO消息框(如图1.2所示)。

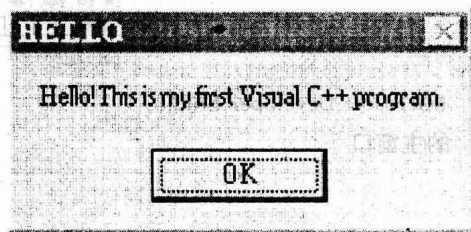


图1.2 HELLO消息框

3. 点击HELLO消息框中的OK按钮就可以关闭HELLO消息框。
4. 点击Hello. EXE程序窗口中的Exit按钮就可以结束Hello. EXE程序的运行。

上面介绍了Hello. EXE程序的外观及其所要完成的功能,下面就开始一步一步地设计

Hello. EXE 程序。

一、创建Hello. EXE 程序文件的目录

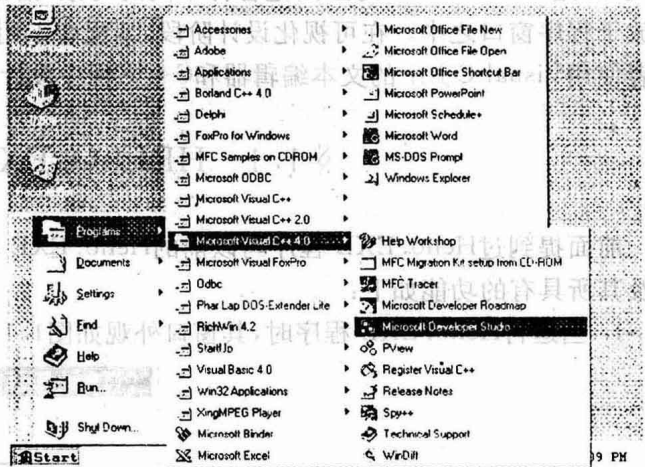
在生成Hello. EXE 程序的工程文件和其它文件之前,首先应在硬盘上创建一个新目录,之后,为Hello. EXE 程序所生成的各个文件就可置放在此目录之中。具体步骤如下:

使用Windows 95 的Windows Explorer 创建一个新目录(其步骤可参考Windows 95 相关手册):C:\TYVCProg\Programs\CH01。

二、创建Hello. EXE 程序的工程

在开始编写一个新的Visual C++ 程序之前,第一件事情就是要为程序创建一个工而在创建一个程序的工程之前,首先必须启动Microsoft Developer Studio。Microsoft Developer Studio 的图标位于Microsoft Visual C++ 4.0 程序组之中(见图1.3)。

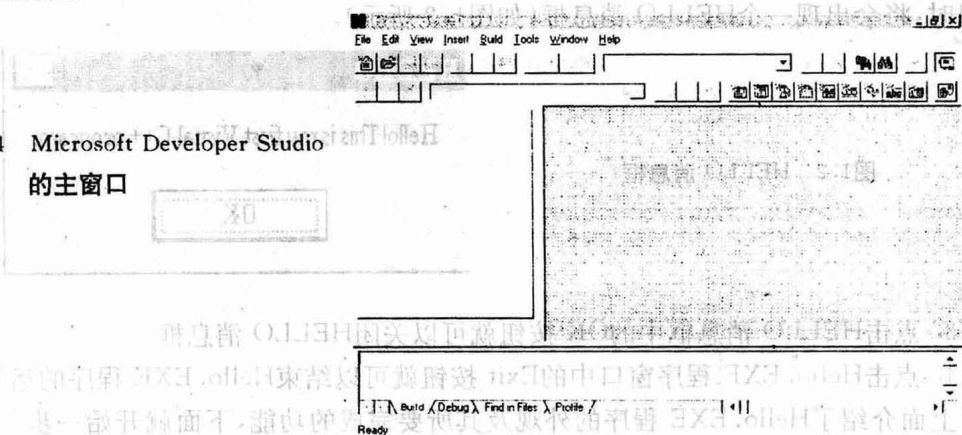
图1.3 运行Microsoft Developer Studio



读者应该注意:Microsoft Developer Studio 是Visual C++ 软件包的精髓,它包括了用于开发程序的所有辅助工具。编程人员在创建、执行和维护Visual C++ 程序时,都要用到Microsoft Developer Studio。

执行Microsoft Developer Studio 之后,其主窗口就出现在屏幕上,如图1.4 所示。(Microsoft Developer Studio 主窗口的实际外观可能与图1.4 有所不同,这主要视Visual C++ 的桌面设置而定)。

图1.4 Microsoft Developer Studio 的主窗口



注意:编写和维护Visual C++程序是通过Microsoft Developer Studio 来进行的,但是,名字“Developer Studio”和“Visual C++”是可以互相交换的。从现在起,本书用“Visual C++”来代替“Developer Studio”。

创建Hello. EXE 程序的工程的步骤如下:

①从Visual C++的File 菜单中选择New。此时,Visual C++将显示一个如图1.5 所示的New 对话框。

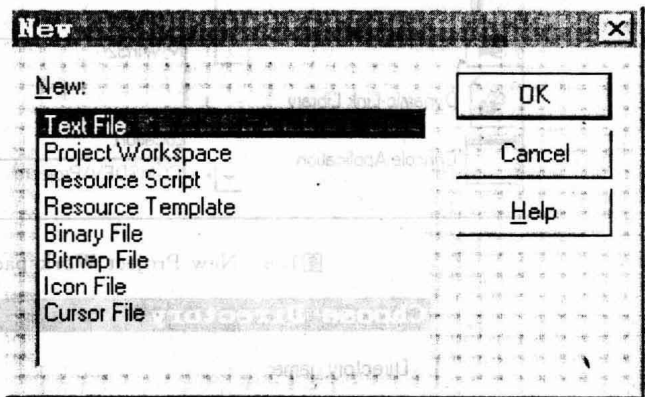


图1.5 New 对话框

②在New 对话框中选择Project Workspace,再点击OK 按钮。此时,Visual C++将显示一个如图1.6 所示的New Project Workspace 对话框。

在New Project Workspace 对话框中,选择工程的类型,并键入新的工程名。

③在New Project Workspace 的Type 列表框中选择类型MFC AppWizard (exe)。

选择MFC AppWizard (exe)的意图是告诉Visual C++即将创建一个EXE 程序。

④点击New Project Workspace 对话框右下角的Browse 按钮。此时,将出现一个如图1.7 所示的Choose Directory 对话框。

⑤在Choose Directory 对话框中选择C:\TYVCProg\Programs\CH01 目录,再点击OK 按钮。(不要把工程文件和其它程序文件存放在Visual C++推荐的缺省目录中,这儿的的结果是将文件存放在C:\TYVCProg\Programs\CH01 目录中。)

⑥在New Project Workspace 对话框中的Name 文本框中输入Hello,此即工程的名字。

至此,已经指定了工程的类型、工程的位置和工程的名字。因此,New Project Workspace 对话框应与图1.8 所示相同。(如有不同,请加以更正。)

⑦点击New Project Workspace 对话框中的Create 按钮。此时Visual C++将显示如图1.9 所示的MFC AppWizard - Step 1 对话框。

MFC AppWizard 是一个功能强大的“向导”,它可以为程序编写出框架代码。以前程序员开始编写一个新程序时,每一次都要重复编写一些相同的代码,现在可以让MFC AppWizard 来完成这些工作。

在下面的步骤中,将利用MFC AppWizard 对话框来为程序指定各种不同的特性,而每当指定一特性时,MFC AppWizard 就会为程序生成相应的代码。

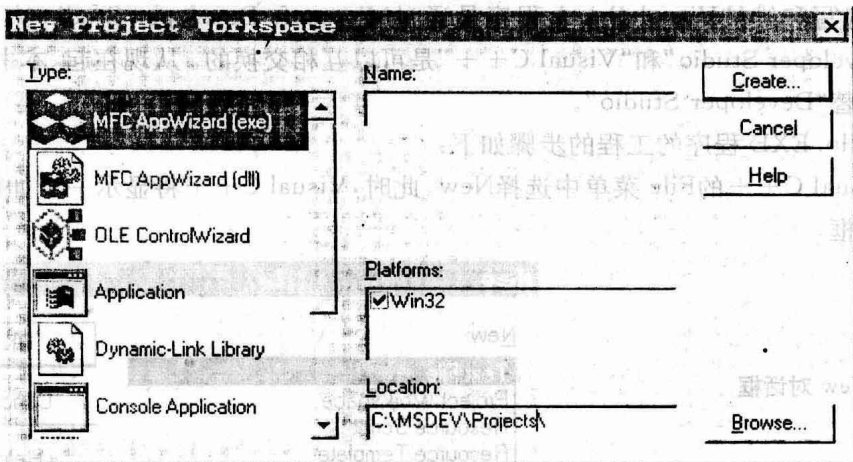


图1.6 New Project Workspace 对话框

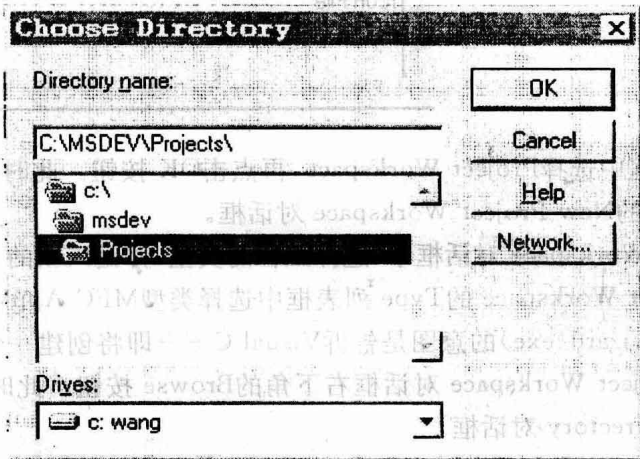


图1.7 Choose Directory 对话框

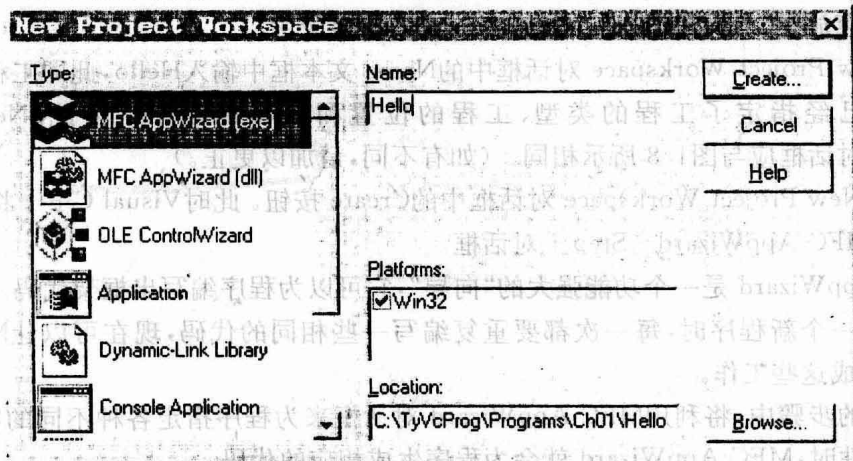
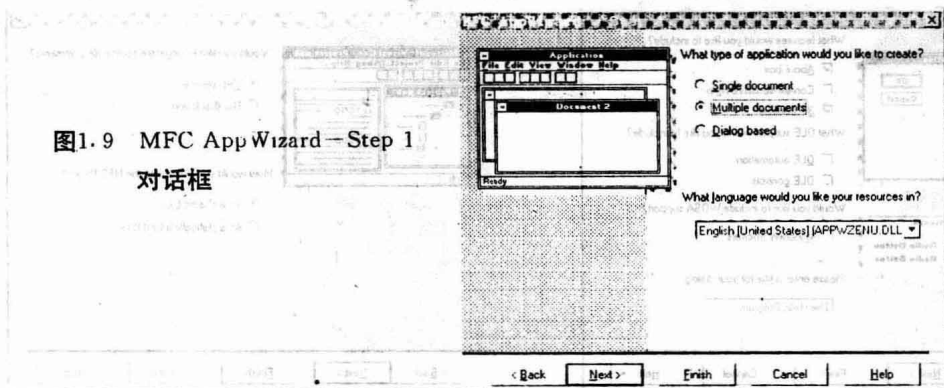


图1.8 已指定工程的类型、位置和名字的New Project Workspace 对话框

图1.9 MFC AppWizard—Step 1
对话框



MFC AppWizard 当前显示的是图1.9所示的MFC AppWizard—Step 1对话框。在此对话框中，可以指定所要创建的应用程序的类型以及程序员想为程序资源所使用的语言类型。

⑧点击MFC AppWizard—Step 1对话框中的“Dialog based”按钮。

选择Dialog based 单选按钮之后，MFC AppWizard 就会为用户生成一个“基于对话框”的程序，也即是说，Hello 程序的主窗口是一个对话框。在前面图1.1中可以看到，Hello.EXE 程序的主窗口是一个带有两个矩形按钮(Say Hello 和Exit)的对话框。

⑨使用Language 组合框的缺省设置:English [united states](APPMENU.DLL)。

现在，MFC AppWizard—Step 1对话框应该与图1.10中所示的一样。

⑩点击Next 按钮以进入MFC AppWizard 的下一个对话框。此时，将出现一个如图1.11所示的MFC AppWizard—Step 2 of 4对话框。

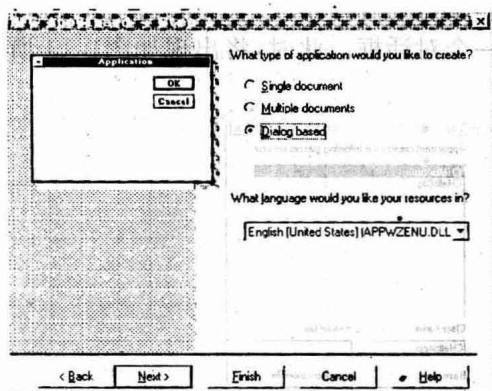


图1.10 MFC AppWizard—Step 1对话框

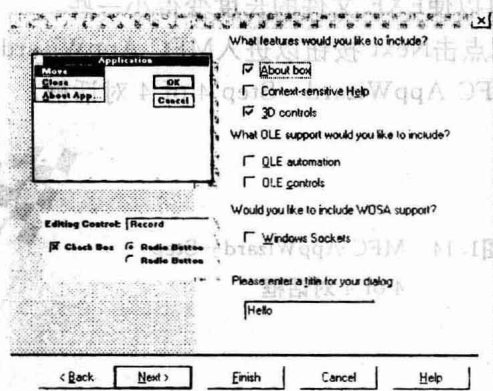


图1.11 MFC AppWizard—Step 2 of 4对话框

MFC AppWizard—Step 2 of 4对话框让用户指定程序中是否包括各种不同的其它特性。并且，它还让用户指定程序的对话框(主窗口)的标题。

⑪按照图1.12所示设置MFC AppWizard—Step 2 of 4对话框的各个项目，所有的复选框保留其初始设置，并把对话框的标题设置为The Hello Program。

⑫点击Next 按钮以进入MFC AppWizard 的下一个对话框。此时，将出现如图1.13所示的MFC AppWizard—Step 3 of 4对话框。

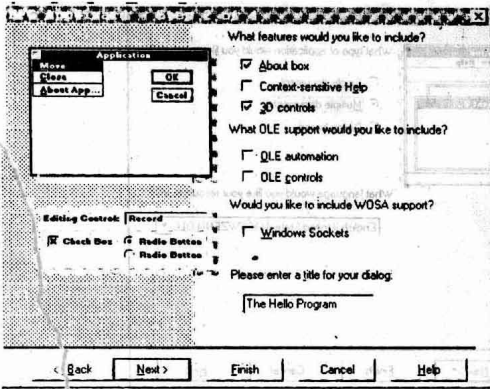


图1.12 MFC AppWizard—Step 2 of 4 对话框

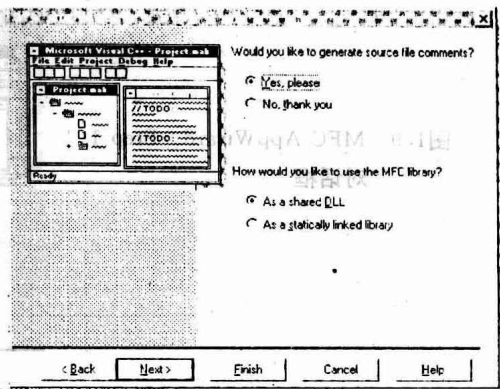


图1.13 MFC AppWizard—Step 3 of 4 对话框

MFC AppWizard—Step 3 of 4 对话框提出了两个供用户回答的问题：

- Would you like to generate source file comments?
- How would you like to use the MFC library?

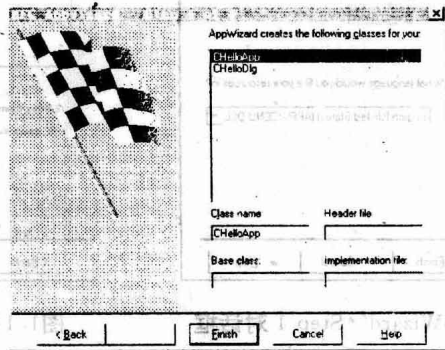
⑬按照图1.13 所示作出选择。

“Would you like to generate source file comments?”的回答是“Yes, please”，标志着MFC AppWizard 生成程序代码时将加上必要的注释说明。这些注释将有助于编程者了解MFC AppWizard 代码。

“How would you like to use the MFC library?”的回答是“As a shared DLL”，它标志着MFC AppWizard 创建的此程序不是使用静态连接库，而是使用动态连接库(DLL)。使用DLL 可以使EXE 文件的长度变得小一些。

⑭点击Next 按钮以进入MFC AppWizard 的下一个对话框。此时，将出现如图1.14 所示的MFC AppWizard—Step 4 of 4 对话框。

图1.14 MFC AppWizard—Step 4 of 4 对话框



MFC AppWizard—Step 4 of 4 对话框让用户选择MFC AppWizard 将要创建哪些类和文件。对于Hello. EXE 程序，直接使用对话框中的缺省设置。

⑮点击图1.14 中的Finish 按钮以让MFC AppWizard 去生成Hello 程序的工程和框架文件。此时，将出现如图1.15 所示的New Project Information 对话框。

New Project Information 对话框中列出了前面步骤中选择的所有结果。

⑯点击New Project Information 对话框中的OK 按钮。于是，Visual C++ 就会在C:\

TYVCProg\Programs\CH01\Hello 目录中生成Hello 程序的所有框架文件。最后,显示如图1.16 所示的Visual C++窗口。

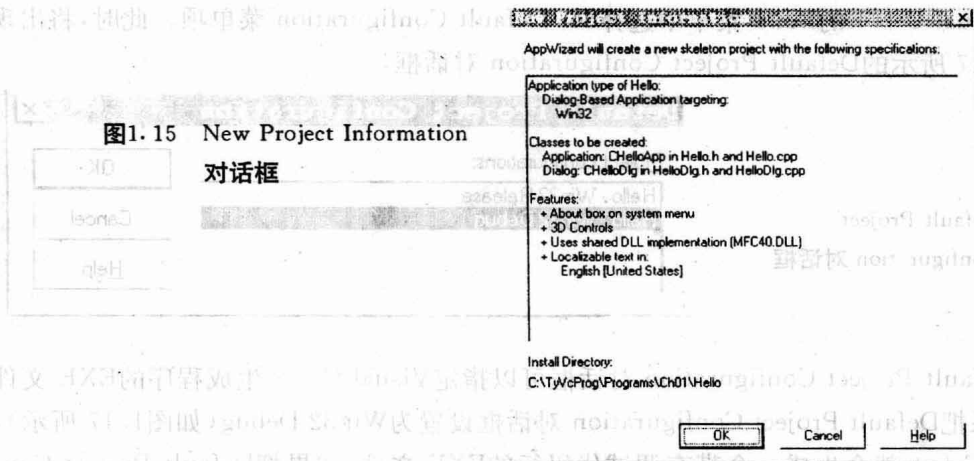


图1.15 New Project Information 对话框

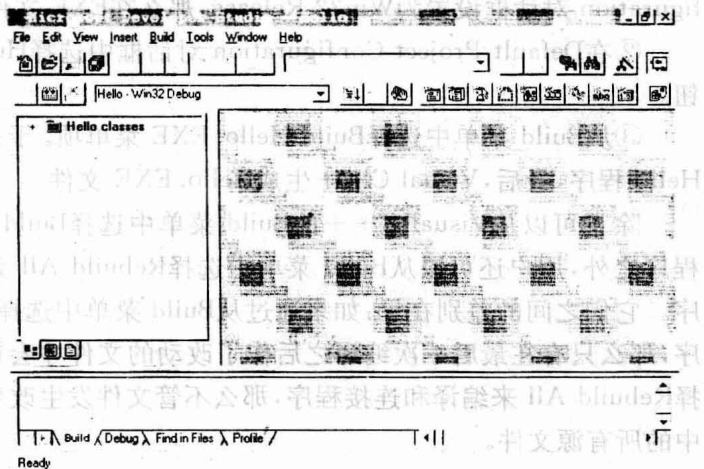


图1.16 创建Hello 程序之后的 Visual C++ 窗口

在图1.16 中,Visual C++窗口的标题是Microsoft Developer Studio - Hello, 其中的“Hello”说明了当前正处于Hello 工程之中。

注意:

1. 在实际创建Hello 程序时所显示的Visual C++桌面可能与图1.16 中所示的有点儿差别,这是由于Visual C++的桌面设置不同造成的。

2. Visual C++窗口的左边有一个对话框,它含有文本“Hello classes”,Microsoft 称之为“Project Workspace”对话框。如果在窗口左边没有显示出这一对话框,那么从Visual C++的View 菜单中选择Project Workspace 即可打开它。

3. 什么是Project Workspace 以及怎样使用Project Workspace 的内容将在本书后面部分讨论,在此暂且不谈。

三、编译和连接Hello 程序

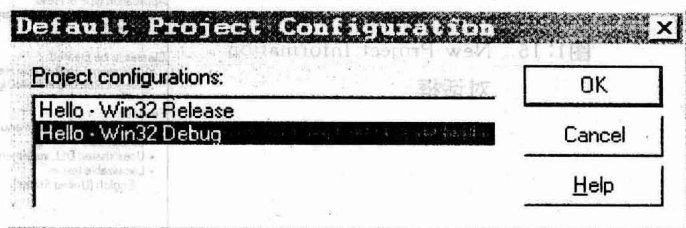
直到现在,我们还没有为Hello 程序写过任何代码,但是,我们已经有了一个能够工作的Windows 95 程序了,而所有这一切都得归功于MFC AppWizard。为了观察Hello 程序究

竟干了哪些工作,不妨先编译并连接该程序,然后再运行它,以便有一个直观的了解。

编译和连接Hello 程序的步骤如下:

①从Visual C++的Build 菜单中选择Set Default Configuration 菜单项。此时,将出现一个如图1.17 所示的Default Project Configuration 对话框。

图1.17 Default Project Configuration 对话框



使用Default Project Configuration 对话框可以指定Visual C++生成程序的EXE 文件的方式。如果把Default Project Configuration 对话框设置为Win32 Debug(如图1.17 所示),那么Visual C++就会生成一个带有调试代码行的EXE 文件;如果把Default Project Configuration 对话框设置为Win32 Release,那么在EXE 文件中就不会嵌入调试代码。

②在Default Project Configuration 对话框中选择Hello - Win32 Release,并单击OK 按钮。

③从Build 菜单中选择Build Hello. EXE 菜单项。于是,Visual C++就开始编译并连接Hello 程序,最后,Visual C++生成Hello. EXE 文件。

除了可以从Visual C++的Build 菜单中选择Build Hello. EXE 项来编译和连接Hello 程序之外,用户还可以从Build 菜单中选择Rebuild All 菜单项来编译和连接Hello. EXE 程序。它们之间的差别在于:如果通过从Build 菜单中选择Build Hello. EXE 来编译和连接程序,那么只有在最后一次编译之后作了改动的文件才会被编译;如果通过从Build 菜单中选择Rebuild All 来编译和连接程序,那么不管文件发生改变与否,Visual C++都会编译工程中的所有源文件。

四、运行和结束Hello. EXE 程序

执行文件Hello. EXE 生成之后,用户就可以启动运行它,以考察其具有的功能。运行和结束Hello. EXE 程序的步骤如下:

①选择Build 菜单中的Execute Hello. EXE 菜单项以执行Hello. EXE 程序。之后,Hello. EXE 程序就会被执行,并显示出如图1.18 所示的主窗口。

MFC AppWizard 生成的框架代码创建了一个简单的EXE 程序,这个程序的主窗口含有两个矩形按钮(OK 和Cancel),并且还显示了一段文本:“TODO: Place dialog controls here”。

程序窗口的标题是“The Hello Program”,这是在图1.12 中的MFC AppWizard - Step 2 of 4 对话框中给出的。

因为在MFC AppWizard 的第二个阶段中为程序引入了一个About 对话框(参看图1.12),所以Hello. EXE 程序也含有一个About 对话框。接下来的步骤将打开Hello. EXE 程序的About 对话框。

②点击程序窗口左上角的小图标以打开Hello. EXE 程序的系统菜单。此时,将弹出一