

```
#ifdef _AFXDLL
```

```
Enable3dControls();
```

附源代码光盘

```
#else
```

```
Enable3dControlsStat
```

```
#endif
```

想学VC吗?

in a shared DLL

读这本书吧.....

Visual C++ 6.0

程序设计

从入门到精通

```
CSi
```

```
pDoc
```

```
IDR_MAINFR
```

```
RUNTIME_CL
```

```
RUNTIME_CL
```

```
RUNTIME_CL
```

```
AddDocTemplate(pD
```

```
Parse command lin
```

```
CCommandLineInfo
```

```
ParseCommandLine
```

```
Dispatch command
```

```
if (!ProcessShellCon
```

```
return FALSE;
```

The one, and only

求是科技 王正军 编著

- 学习 VC 应建立在充分了解 C++ 的基础上, 因此, 阅读本书需要读者具备 C++ 的基础知识
- 学 VC 难一些没错, 但最重要的是要有毅力坚持。学 VC 能深刻领会面向对象编程的精髓
- 作为精通 VC 的软件工程师, 作者在回顾自己的学习历程, 总结自己的开发经验, 吸收现行教材优点的基础上, 为读者奉献了本书

in SDI frame window

v and update it.

人民邮电出版社
POSTS & TELECOM PRESS

Visual C++ 6.0

程序设计

从入门到精通

求是科技 王正军 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Visual C++6.0 程序设计从入门到精通 / 求是科技编著. —北京: 人民邮电出版社, 2006.1

ISBN 7-115-14137-1

I. V... II. 求... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 130000 号

内 容 提 要

本书首先从 Visual C++ 6.0 的基础知识入手, 详细介绍了 Visual C++6.0 的操作环境和应用程序的基本框架, 然后按照由易到难、由浅入深地顺序介绍了用 Visual C++6.0 开发 Windows 程序的方法, 包括菜单和加速键、对话框、常用控件、文本和字体类、图形图像、文件操作、多线程、动态连接库、网络编程和数据库开发等内容。书中在每一个对应的知识点后面, 给出了一个或多个示例。

本书适合想按部就班地学习 Visual C++ 6.0 编程并很快应用到实践中的初学者阅读。

Visual C++ 6.0 程序设计从入门到精通

- ◆ 编 著 求是科技 王正军
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 23.75
字数: 579 千字 2006 年 1 月第 1 版
印数: 1-6 000 册 2006 年 1 月北京第 1 次印刷

ISBN 7-115-14137-1/TP · 5056

定价: 39.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223



求是科技

主 编：张宏林

编 委：周至喜、李玉国、姚金福、王 开、张永吉
张益贞、李 珏、王作远、曹广鑫、杜学峰
于忠秋、翟大中、颜学庆、闫海容、陈晓琳
张 伟、蒋啸宇、赵 夏、魏 雪、王 珏
邹宇斌、朱 昆、马亦飞、王伟滨、窦 亮
侯晓勇、虞福军、李玉周、张志颖、李 梅
王 军、王晓勇、赵文意、仁 宇、李铁柱
刘 静、朱 静、魏伟夫、李冰洋、石国明

前 言

随着计算机技术的发展,可视编程技术占有了越来越重要的作用。Visual C++ 6.0 是微软公司的重要产品——Visual Studio 工具集的组成部分。它用来在 Windows (包括 Windows 95、Windows 98、Windows NT、Windows 2000 等) 环境下开发应用程序,是一种功能强大、行之有效的可视化编程工具,因此成为广大软件开发人员的首选。

本书为希望学习 Visual C++的初学者而编写,按照由浅入深、循序渐进的顺序介绍了 Visual C++编程的相关方法和技巧。在介绍过程中,还在相应的知识点后给出了 1 个或多个相关的示例,以帮助读者尽快将所学知识应用到实践中。

全书的安排如下。

第 1 章 主要介绍了 Visual C++ 6.0 的集成开发环境 (IDE),让读者对 Visual C++先有一个初步印象和了解。

第 2 章 主要介绍在 AppWizard 工具中创建应用程序框架的步骤,然后介绍生成的工程文件中各部分所包含的类及相互间的关系,并着重阐述应用程序框架及整个应用程序的运行机理。

第 3 章 主要介绍了 Windows 界面设计的两个基本元素——菜单和加速键的设计和使用,同时还介绍了 Windows 消息驱动机制的概念和遵循这一机制的程序设计方法。

第 4 章 主要介绍了对话框的一般创建流程,模态和非模态的对话框、属性页对话框、各种通用对话框以及消息对话框的使用方法。

第 5 章 主要介绍了 MFC 编程中常用控件的设计方法,包括传统的 Windows 控件和新类型的 Win32 控件。

第 6 章 主要介绍了 Windows GDI 输出文本和字体的方法。

第 7 章 主要介绍了 Windows 基本的绘图函数和主要的 GDI 对象。

第 8 章 主要通过示例介绍在 MFC 中对文件进行操作的方法,包括串行化、CFile 文件操作和文件对话框 3 部分内容。

第 9 章 主要介绍 Windows 操作系统下多线程的基本概念、创建管理线程的方法,以及线程的同步问题。

第 10 章 主要介绍了在 Visual C++平台下进行 Win32 动态链接库的设计和应用。

第 11 章 主要介绍了网络编程的基本知识及基本步骤,并且详细介绍了 WinSock 类。

第 12 章 主要介绍了数据库的基本概念和数据库开发的几种主要技术、Access 数据库的创建和维护以及 SQL 语句的基本知识。

在本书的编写过程中,借鉴了许多现行教材的宝贵经验,在此,谨向这些作者表示诚挚的感谢。由于时间仓促,加之编者水平有限,书中难免有谬误或不足之处,敬请广大读者批评指正。

编者

目 录

第 1 章 Visual C++ 6.0 开发环境介绍	1
1.1 Visual C++ 6.0 概述	1
1.2 Visual C++ 6.0 的主要特点	1
1.3 Visual C++ 6.0 主框架窗口	3
1.3.1 工作区和客户区	3
1.3.2 菜单栏	4
1.3.3 工具栏	4
1.3.4 输出窗口	4
1.4 资源编辑器	4
1.5 本章小结	5
第 2 章 应用程序基本框架	6
2.1 利用 MFC AppWizard 自动生成应用程序	6
2.1.1 关于工程及工作区的概念	6
2.1.2 利用向导生成应用程序框架	6
2.1.3 Visual C++工程文件的组成	12
2.2 程序的应用框架和运行过程	14
2.2.1 应用框架	14
2.2.2 程序的运行过程	15
2.3 应用程序类及其主要成员函数	18
2.3.1 InitInstance()函数	19
2.3.2 Run()函数	21
2.3.3 ExitInstance()函数	22
2.3.4 OnIdle()函数	22
2.4 文档类和视图类	23
2.4.1 文档类	23
2.4.2 视图类	25
2.4.3 文档类与视图类的相互关系	26
2.4.4 单文档 (SDI) 与多文档 (MDI) 应用程序	28
2.5 窗口类及其控件栏	30
2.5.1 主窗口类	30
2.5.2 子窗口类	31
2.5.3 工具栏	33
2.5.4 状态栏	36
2.6 应用程序框架中各类对象间的关系	40
2.7 本章小结	41

第3章 菜单与加速键	42
3.1 消息与事件	42
3.1.1 事件驱动机制	42
3.1.2 消息和消息映射	42
3.1.3 消息处理过程	46
3.2 菜单和加速键的创建	47
3.2.1 菜单的创建	47
3.2.2 加速键的创建	49
3.3 添加菜单所需处理函数	51
3.3.1 添加普通菜单的处理函数	52
3.3.2 添加弹出式菜单的处理函数	57
3.4 创建加速键	59
3.5 本章小结	60
第4章 对话框	61
4.1 对话框的创建	61
4.1.1 对话框的创建流程	61
4.1.2 创建对话框资源	62
4.1.3 创建对话框类	67
4.2 模态对话框	73
4.3 非模态对话框	74
4.4 属性页对话框	77
4.4.1 主要相关类介绍	78
4.4.2 向导对话框的创建步骤	82
4.4.3 一般属性页对话框的创建步骤	88
4.5 通用对话框	88
4.5.1 打印设置对话框	89
4.5.2 页面设置对话框	90
4.5.3 查找对话框	91
4.5.4 文件对话框	94
4.5.5 字体对话框	95
4.5.6 颜色对话框	96
4.6 消息对话框	97
4.7 本章小结	98
第5章 常用控件	99
5.1 静态文本 (Static text) 控件	99
5.1.1 传统控件通知消息	100
5.1.2 静态控件的使用	101
5.1.3 Cstatic 类的主要成员函数	101
5.2 文本编辑 (Edit Box) 控件	103

5.2.1	文本编辑控件的使用	103
5.2.2	Cedit 类的主要成员函数	104
5.3	滚动条 (Scroll Bar) 控件	108
5.3.1	滚动条控件的使用	108
5.3.2	CscrollBar 类的主要成员函数	109
5.4	按钮 (Button、Radio Box、Check Box、Group Box) 控件	111
5.4.1	按钮控件的使用	111
5.4.2	Cbutton 类的主要成员函数	112
5.5	列表框 (List Box) 控件	117
5.5.1	列表框控件的使用	117
5.5.2	ClistBox 类的主要成员函数	119
5.6	组合框 (Combo box) 控件	122
5.6.1	组合框控件的使用	122
5.6.2	CcomboBox 类的主要成员函数	124
5.7	图片 (Picture) 控件	125
5.8	列表 (List Ctrl) 控件	128
5.8.1	Win32 新型控件概述	128
5.8.2	列表控件的使用	129
5.8.3	ClistCtrl 类的主要成员函数	133
5.9	树型 (Tree Ctrl) 控件	135
5.9.1	树型控件的使用	135
5.9.2	CtreeCtrl 类的主要成员函数	139
5.9.3	树型控件的通知消息	141
5.10	选项卡 (Tab Ctrl) 控件	141
5.10.1	选项卡控件的使用	141
5.10.2	选项卡控件通知消息	142
5.11	本章小结	145
第 6 章	文本和字体类	146
6.1	基础知识介绍	146
6.2	CFont 类	146
6.2.1	概述	146
6.2.2	主要成员函数	147
6.3	文本输出过程	151
6.4	字体和文本输出实例	153
6.5	本章小结	161
第 7 章	图形图像	162
7.1	绘图的基础知识	162
7.2	屏幕绘图的主要函数	163
7.3	图形设备接口 (GDI)	165

7.3.1	设备环境类 (CDC)	165
7.3.2	GDI 对象	166
7.4	画笔和画刷	166
7.4.1	画笔	167
7.4.2	画刷	171
7.5	本章小结	173
第 8 章	文件操作	174
8.1	串行化基础知识	174
8.1.1	文档类	174
8.1.2	Serialize()函数	177
8.1.3	CArchive 类	180
8.1.4	加入串行化代码	182
8.2	文件 CFile 类	186
8.2.1	打开操作	188
8.2.2	读写操作	190
8.2.3	定位操作	192
8.2.4	关闭操作	194
8.2.5	异常操作	195
8.2.6	管理操作	197
8.3	文件对话框	203
8.3.1	多选文件对话框	203
8.3.2	重载文件对话框	204
8.4	本章小结	206
第 9 章	多线程	207
9.1	多线程的基本概念	207
9.2	两种重要的线程	207
9.2.1	用户界面线程	207
9.2.2	工作者线程	208
9.3	线程的操作	208
9.3.1	线程的创建	208
9.3.2	线程的终止	211
9.3.3	线程的管理和操作	212
9.3.4	线程间的通信	229
9.3.5	线程的同步	234
9.3.6	示例: 串口通信	243
9.4	本章小结	250
第 10 章	动态链接库	251
10.1	动态链接库的基础知识	251

10.2 DLL 的出入口函数	252
10.2.1 DllMain()函数	253
10.2.2 DLL 的导出函数	255
10.3 两种链接 DLL 的方式	257
10.3.1 隐式链接	257
10.3.2 显式链接	258
10.4 开发 DLL	259
10.4.1 创建 Non-MFC DLL 动态链接库	259
10.4.2 MFC AppWizard[dll]方式生成常规/扩展 DLL	261
10.4.3 导出函数调用约定	261
10.4.4 模块定义文件 (DEF 文件)	264
10.5 动态链接库中的资源	264
10.6 界面汉化示例	267
10.7 使用钩子 (Hook) 函数	271
10.7.1 钩子函数的类型	271
10.7.2 使用钩子函数	273
10.7.3 鼠标钩子应用实例	274
10.7.4 键盘钩子应用实例	278
10.8 本章小结	279
第 11 章 网络编程	280
11.1 网络编程的基础知识	280
11.1.1 Socket 简介	280
11.1.2 Windows 套接字规范简介	280
11.2 网络应用程序	281
11.2.1 网络应用程序的基本模型	281
11.2.2 建立套接字对象	282
11.2.3 绑定地址	283
11.2.4 建立连接	283
11.2.5 监听 socket	284
11.2.6 数据传输	284
11.3 WinSock 类	285
11.3.1 CAsyncSocket 类	285
11.3.2 CSocket 类	286
11.4 网络编程的实现	288
11.4.1 程序设计的基本任务	288
11.4.2 服务器端程序设计	288
11.4.3 客户端程序设计	309
11.5 利用 MSComm 控件进行串口编程	322
11.5.1 MSComm 控件介绍	322

11.5.2	利用 MSComm 控件编程的基本流程	323
11.5.3	利用 MSComm 进行串口通信	324
11.6	本章小结	328
第 12 章	数据库开发	329
12.1	概述	329
12.1.1	数据库的基本概念	329
12.1.2	数据库相关技术介绍	330
12.1.3	数据库应用程序的开发步骤	331
12.1.4	选择数据库管理系统的原则	332
12.2	设计创建和维护 Access 数据库	333
12.2.1	创建 Access 数据库	334
12.2.2	创建数据表	334
12.2.3	设计和修改数据表的结构	335
12.2.4	设置表之间的关系	336
12.2.5	编辑管理数据库记录	338
12.3	标准的 SQL 语句	340
12.3.1	SQL 的基本知识	340
12.3.2	基本的 SQL 语句	341
12.3.3	关系数据库的联合查询技术	342
12.4	ADO 基础知识	343
12.4.1	ADO 的技术特点	344
12.4.2	ADO 的结构	344
12.5	ADO 的对象和集合	345
12.5.1	连接对象	345
12.5.2	命令对象	346
12.5.3	记录集对象	346
12.5.4	ADO 的其他对象	347
12.5.5	集合	349
12.6	数据库操作	350
12.6.1	动态链接库的引入和 OLE/COM 库的初始化	351
12.6.2	ADO 与数据库的连接	351
12.6.3	获得和遍历记录集	352
12.6.4	对记录的操作	354
12.6.5	关闭记录集	355
12.7	利用 ADO 查询并操作数据库实例	355
12.8	本章小结	370

第 1 章 Visual C++ 6.0 开发环境介绍

1.1 Visual C++ 6.0 概述

Visual C++是微软公司的重要产品之一——Visual Studio 工具集的重要组成部分。它用来在 Windows（包括 Windows 95、Windows 98、Windows NT、Windows 2000 等）环境下开发应用程序，是一种功能强大、行之有效的可视化编程工具。Visual C++以可视化技术为基础，以 C++语言为蓝本，以众多的集成工具为骨架，在计算机领域的诸多方面都发挥着重要的作用。其以实用的开发环境和集成的工具集让用户高效率地开发应用程序。

Visual C++提供了 MFC 类库，使用户可以很方便地开发自己想实现的功能。与 Turbo C 等工具相比，Visual C++完成目标（尤其对于图形界面的程序）所花费的时间要少得多。

Visual C++ 6.0 是微软公司推出的 Visual C++开发工具的 6.0 版本，发行于 Microsoft 的 Visual Studio 6.0 套装软件中。该工具在速度和代码量要求较高的场合中是首选工具。Visual C++ 6.0 在以前版本的基础上又增加了许多特性。除了支持 Internet 特性外，ADO 数据绑定、ATL 复合控件、Auto Completion、编辑和继续特性、OLE-DB 提供者模块、延迟加载移入以及新的调试特性等都给 Visual C++6.0 增色不少。开发者可以充分利用 Visual C++ 6.0 的这些特性，构建出自己的 Windows 应用程序。

1.2 Visual C++ 6.0 的主要特点

Visual C++ 6.0 中的 MFC 类库和以前的版本中相比没有做太大的改进，只是增加了对微软公司的 IntelliMouse（智能鼠标）的支持，但 Visual C++集成开发环境中做了一些改进，增加了一些新特性，使其更易于使用。这些新特性包括以下几个方面。

1. 自动化和宏功能

自动化（Automation）功能用于实现一些重复性过程和工作。宏记录功能可以根据用户的操作自动生成宏操作序列。Visual Studio 及其组件都可以看作对象来处理，这意味着可以进行自动化和诸如打开、编辑、关闭文档和调整窗口等操作。

2. 可定制的工具栏和菜单

可以灵活地定制菜单和工具栏，使其更适合工作需要。如可以创建新的工具条和菜单（增加、删除菜单命令和工具条按钮等）。

3. 调试器

可以直接运行和调试程序，还可以使用宏语言来自动操作调试器。

4. 支持 Internet 连接

可以直接在集成开发环境（IDE）中查看网页页面，可以使用全新的 InfoViewer 或注册的 Web 浏览器查看 Web 上的页面。该特性可以让 Visual Studio 用户了解最新信息、获取更新的文档以及完成产品的升级和修正工作。

5. 项目工作区和文件

一个新的便捷的项目系统应该可以允许一个工作区内包含多个不同的项目类型。比如说，可以创建一个包含 Visual C++ 工程和 J++ Applet 的工作区。在 Visual C++ 6.0 中，工作区文件以 dsw 为后缀名（以前为 mdp），项目文件以 dsp（以前为 mak）为后缀名。Build 文件现在分为两种：内部文件（dsp 后缀）和外部文件（mak 后缀）。在 IDE 中创建一个新的工程或从以前版本的工程进行转换时创建，内部 Build 文件与 NMAKE 外部编译工具不兼容，可以通过选择“Project|Export Makefile”菜单命令创建一个与 NMAKE 兼容的外部 Build 文件。

6. 在工作区内可以包含多个并列的工程文件

要在当前工作区中增加一个工程，可以打开该工作区，然后选择“Project|Insert Project into Workspace”菜单命令。通过选择“Project|Set Active Project”菜单命令，可以设置当前活动工程，也就是执行 Build 操作时编译的那一个工程，这可能是令许多程序员喜爱的一个特性，因为这一特性使得可以在不同工程之间复制代码和资源。在工作区中还支持工程中鼠标直接拖放对象，这样用户可以直接从一个工程的资源文件中拖动一个对话框资源，然后放到另一个工程的资源文件中，而在以前版本中要完成类似的操作，必需先打开一个工程，然后打开另一个工程的资源文件，再进行资源复制操作。

7. 增强的资源编辑器

可以在对话框中使用 WizardBar 将程序同可视化元素联系起来。在使用加速键、对话框、菜单、字符串时，如果需要对多个项作同一修改，可以选择所有要改的项目，然后选择“View|Properties”菜单命令，在 Properties 对话框中一次完成多个项的值的更改。

技巧：要选择多个项，可以先用鼠标选中某一项，然后按住 Ctrl 键，再用鼠标选中其他项。或用鼠标左键拉框选择一个区域的多个项。要向从多个项中去掉一个选择，可以按 Ctrl 键，单击不选的那一项，从而去掉那一项的选择。

8. 文本编辑器

Find in Files 命令支持两个独立的输出窗格，可以保存上一次搜索结果。

这些新特性可以更好地利用 Visual C++ 开发工具进行 Windows 应用程序开发。下面将带领读者进入 Visual C++ 6.0 的集成开发环境。

1.3 Visual C++ 6.0 主框架窗口

Visual C++ 6.0 是一种集成开发环境 (Integrated Development Environment, IDE), 它拥有友好的可视化界面。在 Visual C++ 中能够进行多种操作, 包括建立、打开、浏览、编辑、保存、编译等, 下面将一一介绍各窗口、菜单和工具栏。由于 Visual C++ 是一个可视化的开发工具, 在介绍 Visual C++ 6.0 的各个组成部分之前, 首先了解一下可视化编程的概念。

可视化技术是当前发展迅速并引人注目的技术之一, 它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图象的形式表现出来。可视化编程是它的重要应用之一。所谓可视化编程, 就是指在软件开发过程中, 用直观的具有一定含义的图标按钮、图形化的对象取代原来手工的抽象的编辑、运行和浏览操作, 软件开发过程表现为鼠标点击按钮和拖放图形化的对象以及指定对象的属性、行为过程。这种可视化的编程方法易学易用, 而且大大提高了工作效率。

从开始菜单中启动 Microsoft Visual C++ 6.0, 将进入 IDE 的主窗口。默认的 IDE 主窗口由标题栏、菜单栏、工具栏、工作区窗口、代码编辑客户区、输出窗口和状态栏等几个部分组成, 下面将分别加以介绍。

1.3.1 工作区和客户区

工作区是 Visual C++ 的一个最重要的组成部分。程序员的大部分工作都在 IDE 中完成, IDE 使用项目工作区来组织项目、元素以及项目信息在屏幕上出现的方式。在一个项目工作区中, 可以处理一个工程和它所包含的文件、一个工程的子工程、多个相互独立的工程和多个相互依赖的工程。

一个项目工作区可包含由不同的开发工具包生成的工程, 如 Visual C++ 和 Visual J++。在桌面上, 项目工作区以窗口方式组织项目、文件和项目设置。项目工作区窗口一般位于屏幕左侧, 如图 1-1 中“工作区窗口”所示。项目工作区窗口底部有一组标签, 用于从不同的角度 (视图) 查看项目中包含的工程文件信息。

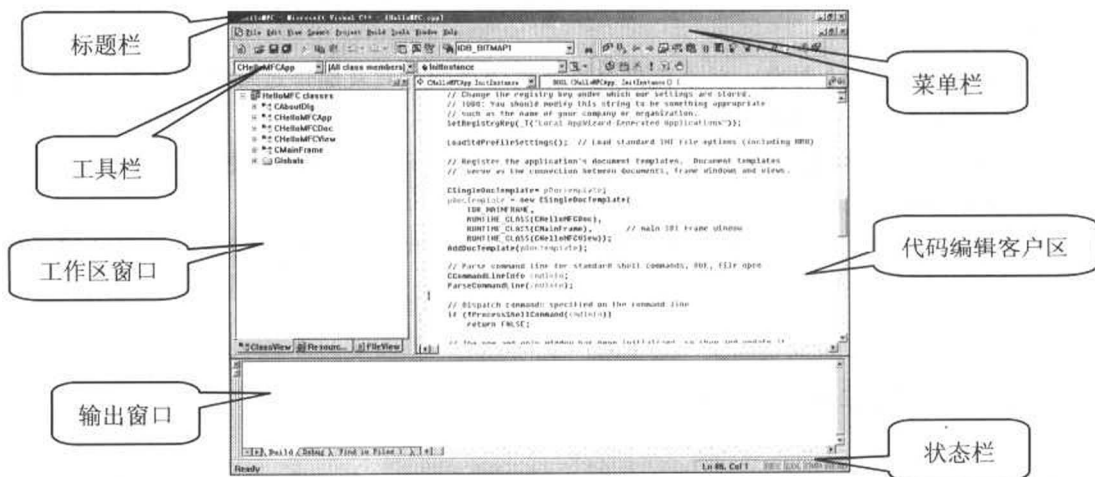


图 1-1 Visual C++ 主窗口

每个项目视图都有一个相应的文件夹，包含了关于该项目的各种元素。展开该文件夹可以显示该视图方式下工作区的详细信息。项目工作区包含 3 种视图：

- **FileView** (文件视图)：显示所创建的工程。展开文件夹可以察看工程中所包含的文件。
- **ClassView** (类视图)：显示项目中定义的 C++ 类，展开文件夹显示工程中所定义的所有类，展开类可查看类的数据成员和成员函数以及全局变量、函数和类型定义。
- **ResourceView** (资源视图)：显示项目中所包含的资源文件。展开文件夹可显示所有的资源类型。

项目工作区底部有 3 个标签项，分别为类视图 (ClassView)、资源视图 (Resource View) 和文件视图 (FileView)。单击某个具体的标签项可以切换到对应的项目工作区视图中。

客户区位于整个主窗口的中部，如图 1-1 中的“代码编辑客户区”所示。客户区是程序员进行代码开发的场所，能直接提供给程序员编写代码、改写代码和调试代码的空间。

1.3.2 菜单栏

菜单栏是 Visual C++ 集成开发环境中的重要组成部分，几乎包含了常用的所有操作功能。菜单所涉及到的操作主要包括文件控制、文本编辑、视图查看、工程设置、编译调试和工具选项等功能。

1.3.3 工具栏

工具栏也是 IDE 的重要组成部分，主要列出了在代码开发过程中经常用到的一些功能。显示和隐藏工具栏有下面两种方法：

- 选择“Tools|Customize|Toolbars”菜单命令，将显示所有工具栏名称，根据需要进行选择。
- 在工具栏上单击鼠标右键，弹出所有工具栏名称菜单，根据需要进行选择。

1.3.4 输出窗口

输出窗口位于整个主窗口的下方，主要用于显示代码调试和运行中的相关信息，包括下面几个方面。

- **编译 (Compile) 信息**：列出代码和资源编译详细过程及编译过程中的警告 (Warning) 和错误 (Error) 信息。
- **连接 (Link) 信息**：列出工程对目标模块 (Obj) 连接过程中的警告 (Warning) 和错误 (Error) 信息。
- **调试 (Debug) 信息**：在调试 (Debug) 状态下输出相关的调试信息 (如 TRACE 宏输出调试信息等)。

1.4 资源编辑器

资源编辑器是 Visual C++ 6.0 开发环境中的一个重要可视化开发工具，绝大多数 Windows

应用程序的可视化界面元素都可以利用资源编辑器生成。Windows 资源包括加速键、图标、光标、位图、对话框、工具栏等，通过资源编辑器，可以打开资源文件（res 文件），并能对资源文件里面的资源进行编辑。在 Visual C++ 6.0 中，源代码和资源编辑器为了应用资源和用户自定义的其他对象，引入了一种与资源一一对应的符号，从而使资源能够以字符串的方式被引用。

资源符号是映像到某个整数值上的一字符串，即标识符。每当创建一个新的资源时，系统就会为其提供一个默认的资源符号名称并赋予一个整数值。

1.5 本章小结

本章介绍了 Visual C++ 6.0 的集成开发环境（IDE），旨在让读者对 Visual C++ 先有一个初步印象和了解，至于更详细的使用方法，将在后面章节随着开发技术的具体讲解进行仔细地介绍。

第 2 章 应用程序基本框架

在熟悉了 Visual C++6.0 的开发环境后, 就可以在该开发环境下进行程序设计了。Visual C++6.0 提供了应用程序框架生成向导 (AppWizard), 以方便用户自动生成一个简单的应用程序。当然, 利用 AppWizard 只是得到一个程序框架而已, 说它是框架, 是因为程序的执行结果仅仅是显示一个包括菜单、工具栏和状态栏的空窗口。

本章中, 首先介绍在 AppWizard 工具中创建应用程序框架的步骤, 然后介绍生成的工程文件中各部分所包含的类及相互间的关系。

本章着重阐述应用程序框架及整个应用程序的运行机理, 这部分内容可能读者暂时不能完全明白, 不过在后面各章的学习中回过头来阅读相关部分便能更为深刻的领悟。

2.1 利用 MFC AppWizard 自动生成应用程序

2.1.1 关于工程及工作区的概念

在启动 Visual C++6.0 之后, 将会进入它的集成开发环境 IDE, 用户可以在这个集成开发环境中进行程序设计。现在的应用程序, 尤其是 Windows 应用程序, 一般由很多相关联的文件共同组成, 包括程序源文件、头文件、资源文件等, 所以在利用 Visual C++进行 Windows 程序开发的时候, 有必要引入工程的概念。讲一个应用程序作为一个工程来处理, 通过工程化的管理, 使组成应用程序的所有文件形成一个有机的整体。

在 Visual C++6.0 开发环境中, 工程又是置于工作区的管理之下, 所以工作区又称为工程工作区, 一个工作区可以管理多个工程, 甚至是不同类型的工程。同一个工作区中的工程之间相互独立, 但共用一个工作区的设置环境。在 Visual C++6.0 的集成环境中专门设置了一个工作区窗口, 用以显示当前工作区的内容。在新建一个工程时, 可以选择是为该工程新建一个工作区还是加入当前工作区。

在利用应用程序向导 (AppWizard) 工具生成一个应用程序时, Visual C++会在用户选择的路径下以当前的工程名建立一个目录, 然后将自动生成的文件置于该目录下。

2.1.2 利用向导生成应用程序框架

实例 2-1: 利用向导生成的应用程序框架。源代码在光盘中 “\02\实例 2-1\HelloMFC” 目录下。

下面介绍使用应用程序向导工具生成 HelloMFC 应用程序的具体步骤。

(1) 选择 “File|New” 菜单命令, 将弹出 “New” 对话框, 如图 2-1 所示。此时可以选择