



Visual C++ 程序员之路 2

Visual C++ 程序设计

北京希望电子出版社 总策划
徐晓梅 高 丽 编 著



科学出版社

www.sciencep.com



TP312
X824:1

Visual C++ 程序员之路 2

Visual C++ 程序设计

北京希望电子出版社 总策划
徐晓梅 高丽 编 著

内 容 简 介

这是一本介绍 Visual C++ 例程的实用编程指导书，它是 Visual C++ 程序员之路系列的第二本。

本书共由 13 章内容构成。主要内容如下：Win32 应用程序设计，Visual C++ 概述，对话框和控件编程，图形设备接口，文档编程，控件编程，动态链接库，数据库编程，多媒体编程，网络编程，Windows 系统编程，COM 技术，MFC 通用类。

本书实用性和可读性较强，适合了解 Windows 编程基本概念的初、中级软件开发人员参考使用。

需要本书或需要得到技术支持的读者，请与北京中关村 083 信箱（邮编 100080）发行部联系。电话：010-62528991, 62524940, 62521921, 62521724, 82610344, 82675588（总机） 传真：010-62520573, E-mail: yanmc@bhp.com.cn

图书在版编目 (CIP) 数据

Visual C++ 程序设计 / 徐晓梅 高丽编著. —北京：科学出版社，2003

(Visual C++ 程序员之路：2)

ISBN 7-03-012561-4

I. V… II. ① 徐…② 高… III. C++ 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 109517 号

责任编辑：周艳 / 责任校对：但明天
责任印刷：东升 / 封面设计：王 翼

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>

北京市东升印刷厂印刷

科学出版社发行 各地新华书店经销

*

2003 年 12 月第 一 版 开本：787×1092 1/16
2003 年 12 月第一次印刷 印张：27 1/2
印数：1—5 000 字数：631 332

定价：39.00 元

前言

Visual C++一直都是 Microsoft 公司综合性最高、最复杂的软件开发平台。在把各种新技术应用在 Windows 平台的同时,微软公司也把各种工具加到了 Visual C++中,使其能够适合各种编程风格,同时编程人员也能更容易地将这些新技术运用到应用程序。

然而,学习使用 Visual C++编程却并不是一件简单的任务。Visual C++编程的重点在于掌握基本的代码结构和程序设计方法。本书首先对 Windows 程序设计以及 Visual C++作概述性介绍,并对其中涉及的相关基本概念进行了阐述,然后分不同的主题结合具体的程序示例讲解如何使用 Visual C++开发 Windows 应用程序。当你结束了本书的学习时,可以用自己的编程知识和技巧开发出目前可用的大多数 Windows 应用程序。

本书第 1 章“Win32 应用程序设计”介绍了 Win32 编程的一些基本概念和基本原理,这对于学习和掌握 Visual C++编程非常必要,读者将会发现这些基本概念和基本原理始终贯穿于 Visual C++编程的过程中。

第 2 章“Visual C++概述”首先介绍了 VC++的编译环节以及编写 VC++程序的两个基本工具,即 AppWizard 和 ClassWizard。用户通过 AppWizard 可以建立一个基本的 VC++程序框架,而通过 ClassWizard 可以对这个基本的 VC++程序框架进行添加类和成员函数以及对控件进行消息映射的操作,然后编写用户代码,进而实现用户所要求的功能。最后介绍了 Visual C++的帮助,这对于程序员来说是非常重要的。因为 Visual C++的内容非常庞大,通过 Visual C++的帮助可以很容易地找到所需要的东西。

第 3 章“对话框和控件编程”首先介绍了对话框(Dialog)和控件的基本概念:对话框实际上是用户通过其中的控件与程序进行交互的一个窗口,在 MFC 中,对话框的功能被封装在了 CDialog 类中,CDialog 类是 CWnd 类的派生类;然后详细介绍了如何设计对话框类、如何为控件添加代码;最后详细介绍了基类 CWnd 以及 Windows 的标准控件,包括:按钮控件、静态控件、文本编辑控件、列表框控件、组合框控件、滚动条控件。

第 4 章“图形设备接口”首先介绍了图形设备接口及其紧密关联的设备上下文概念;然后详细介绍了画笔对象、刷子对象、字体对象以及如何在程序中运用这些对象的方法、步骤;最后还介绍了在设备上下文中绘制时所使用的映射模式。

第 5 章“文档编程”首先介绍了 Windows 环境下的文档/视结构基本概念,并分别给出了文档/视结构应用的例子和多文档例子;然后介绍了串行化数据和视类;最后还介绍了如何创建一个文档的多个视。

第 6 章“控件编程”首先介绍了在 VC++ 6.0 中创建 ActiveX 控件的方法;然后详细介绍了如何使用 ActiveMovie 控件创建一个视频播放器,并给出了详细步骤和程序示例。

第 7 章“动态链接库”首先介绍了动态链接库的基本概念;然后以非 MFC DLL 为例来详细介绍了动态链接库的结构和导出方法,并介绍创建和使用动态链接库的方法和步骤。

第 8 章“数据库编程”首先介绍了数据库编程的基本原理,以及 Visual C++进行数据库编程的技术和方法,并分别简单介绍了这些技术的基本原理和方法;然后结合一些程序

实例介绍进行数据库编程的技巧；最后结合程序示例分别介绍了如何使用 Visual C++访问 Visual Foxpro, ACCESS, Foxbase, ORACLE 数据库。

第 9 章“多媒体编程”首先介绍了声音编程的方法,并介绍了 PlaySound 函数、多媒体控制接口 MCI 及其类、CWave 类,以及如何使用它们进行声音编程的方法;然后介绍了视频编程的方法,并分别结合程序示例介绍了如何使用 MCIWnd 类和 MediaPlayer ActiveX 控件创建一个多媒体播放器的过程;最后详细介绍了 OpenGL 图形库的基本概念、函数以及使用 OpenGL 图形库编程的一般步骤,并给出了一个应用 OpenGL 图形库编程的程序示例。

第 10 章“网络编程”主要介绍了网络编程的原理、通过 Winsock 进行网络编程的方法与技巧,并列举了多个用 Visual C++进行网络编程的实例以供参考。

第 11 章“Windows 系统编程”主要介绍了 Win32 API、多线程多任务、磁盘文件操作、钩子程序以及注册表编程,特别是列举的大量实例可供参考。

第 12 章“COM 技术”主要介绍了 ATL、ActiveX、COM 技术以及 IE 编程的知识,并列举了相应的编程实例可供参考,其中重点讲解了 COM 编程。

第 13 章“MFC 通用类”分别介绍了数组类、列表类、映射类、字符串类、日期和时间类,并给出了一些程序示例演示如何运用这些类的方法和技巧。

本书实用性和可读性较强,适合了解 Windows 编程基本概念的初、中级软件开发参考使用。

本书由徐晓梅、高丽组织编写,完成本书是一项工作量很大的工程,是集体合作的结果。其他参与本书策划、评审、编写、录入、审校和排版的还有:龚克、田军、田蕴哲、牛志奇、丁天、龚志翔、李红玲、金荣学、薛彪、叶哲、李林、张巧莉、龚丽娜、刘建成、杨志强、李海涛、李志、刘海深、赵军锁、姜南、陈培、汪建华、吴秋丽等。最后,龚波对全部书稿进行了审核和统稿,并提出了许多宝贵的意见。

由于时间仓促,且经验和水平有限,文中难免有不妥之处,我们殷切地期望你能给我们提出中肯、尖刻的意见,以便提高水平,把更好的图书呈现给大家!如有问题,请与 tony_hcb@sina.com 联系。

作 者

2003 年春

一个 VC 爱好者的入门之路

看到那些对 VC 不知从何下手而苦苦挣扎的朋友，希望我的经验能给他们一些借鉴。

学 VC 并不是传说的那么难，可不下些苦功夫是学不成的。学编程急不得，没有编程的基础知识上来就学 VC 肯定碰一头灰，说 VC 难就难在这点上了。如果硬上，意志坚强的话也许能挺过来，但最后还得回过头来补习基础知识。意志不坚强的话，很有可能就此放弃了，并留下一个 VC 难得不得了的印象。其实，只要踏踏实实一步一步来，学 VC 很简单。

对于没有编程基础的人首先要学习编程的基础知识，如变量、语句、基本的算法等，然后写一些小程序，实现些常用算法对自己的思维是很好的锻炼，对以后的学习大有好处。起码要能排排序，求个八皇后，算个汉诺塔，再知道一点深、广度优先搜索。如果你已经有一些编程的基础知识了，自然就可以跳过这步。

在对编程有一定的认识后，就应该开始进军 C++ 了。注意，C++ 而不是 VC，这两者不是一码事，C++ 是一门语言，而 VC 则是讲解如何在 VC 环境下使用 MFC 类库，学习 VC 应建立在充分了解 C++ 的基础上的。

学完了 C++，最好先找本 Win32 编程的书，Win32 编程是指用 Windows API 编程，MFC 是将众多的 API 函数进行了封装的类库，用起来方便了不少，但如果没有一些 API 编程基础，对相关概念不了解，学起 MFC 来可能会很费解。API 函数有 2 千多，不需要都学。其实只要知道如何创建窗口、对话框、控件、消息循环，了解了句柄的含义就差不多了，最好再了解一点 GDI 编程。这个阶段的关键是要理解 Win32 程序是如何运作的，把握各种窗口间的关系。因为你以后应该很少直接用 Win32 SDK 编写程序，所以关键是理解整体结构，记住几个函数并没有什么意义。函数用的时候可以现查。

打好了基础，现在进军 VC 的障碍都已扫清了。不过还是应做好充分的思想准备，事情不会那么顺利的。刚开始你就要面对一个瓶颈，大量的紧密关联的知识，使你必须一下理解一个整体，而很难一点点来。可以说学 VC 最难的就是刚开始。情况有些类似于第一次创建窗口时面对的那长长的程序，很多人就是在此被它难住，放弃了学习。事实上，这几乎是 Windows 编程的一大特点，突破了这个瓶颈之后便是一片海阔天空，而有着扎实基础的你是一定能闯过去的。

建议不要等彻底弄明白一句才看下一句，等能做出个菜单、对话框时便回过头再看一遍，几遍之后应该就可以大概明白 MFC 程序的运作了。这里有个关键，一定要用心体会面向对象编程的精神，努力将眼前的程序与之前所学的知识联系起来。通过比较成员函数和对应的 API 函数间的区别，尝试去理解 MFC 是怎样封装相关的 API 的。例如，相当多的类将句柄封装了起来，在传递参数时加以省略，这就是一个很重要的特性。MFC 中有许多的宏，看起来很奇怪，但你不必理会其实现方法，只要知道如何使用便可，这点也很重要。还有，对于其生成的代码要自己进行分析，偷懒就不能真正理解 MFC。

这是一个人的学习经验，不过经验要因人而异，所以还要根据个人的情况决定。开始时，教程让怎么办就怎么办，比猫画虎的作出个菜单、对话框，然后试着添加控件。等到

不看教程便做到这些时,开始逐字逐句的试着理解 MFC 程序的组织。若看不懂就跳过看下面的。就这样一遍一遍来。如果觉着头大就先歇上 2 天,等清醒了再接着来,最好在这期间多找些介绍 MFC 框架的文章。然后便可试着写个定时器、名片夹之类的小东西,熟悉一下各种控件的使用。

突破了开始的瓶颈之后再学下去便会越学越容易,因为你可以一点一点的积累知识,而不用担心被噎着了。再过不久,你就可以使用 API 函数完成一些功能,如在系统托盘加图标,自画列表框等。这时,你也会体会到学习 Win32 编程的好处了。

最后,建议有时间的话多研究一下各种算法。数据结构是一定要看的。不一定要记住,但至少要知道有那么回事,万一碰到问题了可以知道问题出在哪里。其实,常见的链表等结构以及排序等算法在 C++ 标准类模板中已经实现了,不用你去费劲,但有些了解总是有好处的。至于离散数学、线性代数什么的,对于复杂些的算法是必需的,其中离散数学是数据结构的基础。

目 录

第 1 章 Win32 应用程序设计	1
1.1 Win32 编程介绍	1
1.1.1 消息驱动机制	1
1.1.2 Win32 API	1
1.1.3 句柄	2
1.1.4 程序入口点 WinMain	2
1.2 MFC 机制	4
1.2.1 MFC 是一个编程框架	4
1.2.2 MFC 应用程序的构成实例	5
1.2.3 消息映射	7
1.3 本章小结	10
第 2 章 Visual C++ 概述	11
2.1 Visual C++ 和 MFC 的历史	11
2.2 Visual C++ 6.0 开发环境	11
2.3 用 AppWizard 生成 Win32 程序框架	16
2.3.1 AppWizard 生成 Win32 程序 框架步骤	16
2.3.2 AppWizard 生成的对话框应 用程序	20
2.4 ClassWizard 类管理器	21
2.4.1 ClassWizard 简介	21
2.4.2 使用 ClassWizard 类管理器	23
2.5 使用资源管理器	24
2.6 编译与运行	25
2.7 获取帮助	26
2.8 第一个 VC++ 程序	29
2.9 本章小结	30
第 3 章 对话框和控件编程	31
3.1 对话框概述	31
3.1.1 对话框的基本概念	31
3.1.2 控件概述	32
3.1.3 使用对话框编辑器	33
3.2 对话框的编程	35
3.2.1 对话框的属性	35
3.2.2 对话框类的设计	36

3.2.3 为控件添加代码	37
3.3 Windows 标准控件	39
3.3.1 窗口类的基类 CWnd	39
3.3.2 Visual C++ 控件的分类	41
3.3.3 按钮控件	42
3.3.4 静态控件	46
3.3.5 文本编辑控件	47
3.3.6 列表框控件	49
3.3.7 组合框控件	51
3.3.8 滚动条控件	53
3.4 本章小结	54
第 4 章 图形设备接口	55
4.1 设备上下文	56
4.1.1 与图形绘制有关的简单数 据类型	57
4.1.2 显示设备上下文	59
4.2 画笔对象	66
4.2.1 创建画笔	66
4.2.2 使用画笔进行输出	67
4.2.3 图形输出函数	70
4.3 刷子对象	75
4.4 字体对象	80
4.4.1 创建字体对象	80
4.4.2 LOGFONT 结构	81
4.4.3 使用字体对象和枚举系统 中的所有字体	83
4.4.4 创建特殊的字体效果	91
4.5 映射模式	98
4.5.1 预定义的映射模式	98
4.5.2 自定义的坐标系统	106
4.6 本章小结	110
第 5 章 文档编程	111
5.1 文档/视结构	111
5.1.1 Windows 环境下的文档机制	111
5.1.2 使用 AppWizard 创建框	

架应用程序	112	8.2.6 从数据库中读大于 32KByte 的 内容	201
5.1.3 一个简单的例子	119	8.2.7 基于 ActiveX 的多通道数据 曲线编辑控件的实现	202
5.1.4 多文档的例子	122	8.2.8 用 MFC Dao 查找	204
5.2 串行化数据	125	8.2.9 ODBC 的查找函数	205
5.3 视类	129	8.2.10 在无绑定模式下使用 DBGrid	206
5.4 同一文档的多个视	130	8.3 数据库编程实例	212
5.5 本章小结	134	8.3.1 用 VC++ 和 Visual Foxpro 进行 ActiveX 数据通讯	212
第 6 章 控件编程	135	8.3.2 用 VC 处理 Access 中的位 图数据	215
6.1 ActiveX 控件的建立	135	8.3.3 用 VC 开发基于 ORACLE 数 据库应用程序的两种方法	216
6.2 使用 ActiveXMovie 控件的视频播 放器	140	8.3.4 用 VC++ 给 Foxbase 数据库 文件加密	220
6.3 本章小结	160	8.4 本章小结	221
第 7 章 动态链接库	161	第 9 章 多媒体编程	223
7.1 概述	161	9.1 声音编程	223
7.2 创建和使用动态链接库	163	9.1.1 PlaySound	223
7.2.1 DLL 的结构和导出方式	163	9.1.2 多媒体控制接口 MCI	225
7.2.2 链接应用程序到 DLL	167	9.1.3 MCIWnd 窗口类	227
7.2.3 使用动态链接连库扩展 MFC	170	9.1.4 CWave 类	229
7.3 本章小结	175	9.2 视频编程	237
第 8 章 数据库编程	176	9.2.1 用 MCIWnd 类生成一个 多媒体播放器	238
8.1 数据库编程基本原理	176	9.2.2 用 ActiveX 控件生成一个 多媒体播放器	240
8.1.1 开发数据库应用程序	177	9.3 OpenGL 三维图形设计	244
8.1.2 Visual C++ 中的 ODBC 编程	178	9.3.1 OpenGL 特点	244
8.1.3 调用 DAO	180	9.3.2 OpenGL 图形库	244
8.1.4 使用 ADO 开发数据库应用 程序	182	9.3.3 OpenGL 编程步骤	244
8.1.5 在 VC++ 中建立自定义数据 库类	187	9.3.4 OpenGL 函数	245
8.2 数据库编程技巧	190	9.3.5 OpenGL 实例	248
8.2.1 在动态集中使用虚拟 ClistView	190	9.4 本章小结	251
8.2.2 在 Access DB 中存储画笔 图像	195	第 10 章 网络编程	252
8.2.3 在程序中动态创建并修改 BDE 别名	196	10.1 网络编程原理简介	252
8.2.4 在 ODBC 中应用 DDX 和 RFX	197	10.1.1 Winsock 应用程序设计	252
8.2.5 定位到最新的记录	201		

10.1.2	利用 WinInet 类进行 TCP/IP 网络通信程序设计	267
10.1.3	Win32 下的几种串口通信编程	271
10.1.4	SMTP 协议类库	272
10.1.5	POP3 协议类库	272
10.2	网络编程技巧和方法	280
10.2.1	如何得到拨号网络传输速度	280
10.2.2	如何在应用程序中映射网络驱动器	282
10.2.3	如何在 NT 下调试 ISAPI 程序	283
10.2.4	如何应用 TCP/IP 的套接字开发网络通信应用程序	285
10.2.5	在浏览器中改变字体	289
10.2.6	在对话框中如何打印网页	290
10.3	网络编程实例	291
10.3.1	用 VC++ 和 Winsock 实现 HTTP 服务器通话	291
10.3.2	用 VC++ 编写 CGI 程序	292
10.3.3	用 VC++ 编写 FTP 客户端应用程序	295
10.3.4	用 VC++ 实现上网拨号功能	298
10.3.5	用 VC++ 编写服务器与客户机互相传送消息的程序	300
10.3.6	用 WINSOCK 实现聊天室的 VC++ 程序设计	307
10.3.7	用 ActiveX 实现串口通信	307
10.3.8	用 VC++ 编写电子邮件程序	308
10.3.9	如何用 VC++ 开发读取网卡 MAC 地址的程序	312
10.4	本章小结	314
第 11 章	Windows 系统编程	315
11.1	Win32 API	316
11.1.1	API 函数简介	316
11.1.2	API 函数的调用	318

11.1.3	API 函数实用技巧	322
11.1.4	调用 API 函数实例	325
11.2	多线程多任务	328
11.2.1	多线程编程概述	328
11.2.2	多线程使用技巧	329
11.2.3	用 MFC 开发多线程应用程序的实例	332
11.3	磁盘文件操作系统	341
11.3.1	理解 Windows 的文件 I/O	344
11.3.2	写、读和关闭及打开文件	344
11.4	钩子程序	357
11.4.1	钩子原理	357
11.4.2	钩子应用技巧	358
11.4.3	钩子实用实例	358
11.5	注册表	362
11.6	本章小结	372
第 12 章	COM 技术	373
12.1	ATL 技术	373
12.1.1	ATL 基本原理	373
12.1.2	ATL 实现方法	374
12.1.3	ATL 与 COM	376
12.1.4	ATL 的应用示例	376
12.2	ActiveX 控件技术	377
12.2.1	OLE 技术	378
12.2.2	ActiveX 控件的定义和内容	378
12.2.3	创建 ActiveX 控件	379
12.2.4	使用 ActiveX 控件	382
12.2.5	创建 ActiveX 控件的技巧	384
12.2.6	如何分发 ActiveX 控件	384
12.3	COM 技术	386
12.3.1	COM 组件概述	386
12.3.2	COM 接口技术	387
12.3.3	DCOM 简介	392
12.3.4	COM 编程技巧	395
12.3.5	COM 编程实例	396
12.4	IE 编程	402
12.4.1	IE 基本原理	402
12.4.2	IE 编程技巧	402

12.4.3 IE 编程实例.....	403
12.5 本章小结.....	404
第 13 章 MFC 通用类.....	405
13.1 数组类.....	405
13.2 列表类.....	411
13.3 映射类.....	416

13.4 字符串类.....	419
13.5 日期和时间类.....	422
13.5.1 CTime 类.....	422
13.5.2 CTimeSpan 类.....	425
13.6 本章小结.....	426

第 1 章 Win32 应用程序设计

对于具有一定基础的程序员而言,知道 Windows 应用程序的运行机制,以及它们之间以何种方式来进行通信是非常重要的,然而,明确底层所发生的事件的过程,不是一件容易的事。只有理解了 MFC 应用程序框架的运行机制,才可能用好和用活这一工具,最终达到熟练运用 Visual C++ 的境界。

Windows 应用程序的运行机制大部分功能是通过调用最基本的 Win32 应用程序接口 (Win32 API) 来实现的,通过消息来和操作系统之间进行交互。MFC 对这一切进行了比较彻底的封装,即使对底层一点也不懂,仍可以在 Visual C++ 中使用 MFC 来进行程序设计。

1.1 Win32 编程介绍

1.1.1 消息驱动机制

Windows 应用程序是消息驱动的应用程序。Windows 是一个多任务的操作系统,也就是说,在同一时刻,在 Windows 中有着多个应用程序的实例正在运行。Windows 时刻监视着用户的一举一动,并分析用户的动作与哪一个应用程序相关,然后,将用户的动作以消息的形式发送给该应用程序,而应用程序则时刻等待着消息的到来,一旦发现它的消息队列中有未处理的消息,就获取并分析该消息,最后,应用程序根据消息所包含的内容采取适当的动作来响应用户所作的操作。

例如,一个程序中有一个按钮,在运行该应用程序的时候,如果用户单击了按钮,这个动作将被 Windows 发现并分析,得知这个动作应该用这个应用程序去处理,Windows 就发送了一个叫做 WM_COMMAND 的消息给应用程序,应用程序得知这一消息之后,采取相应的动作来响应它,这个过程称为消息处理。Windows 应用程序是由一系列的消息处理代码来实现的。

Windows 程序在处理消息时使用了一种叫做回调函数(callback function)的特殊函数。回调函数由应用程序定义,但是,在应用程序中并没有调用回调函数的代码,回调函数是供操作系统或者其子系统调用的,这种调用通常在某一事件发生,或者在窗口或字体被枚举时发生。典型的回调函数有窗口过程、对话框过程和钩子函数。

1.1.2 Win32 API

Windows API 是 Windows Application Programming Interface 的缩写,即应用程序编程接口,它是所有 Windows 应用程序的根本之所在。

API 实际上就是一系列的例程,应用程序通过调用这些例程来请求操作系统完成一些底层服务。通常应用程序的窗口、图标、菜单和对话框等就是由 API 来管理和维护的。

Win32 API 依靠三个主要的核心组件提供 Windows 的大部分函数。

(1) USER32.DLL: 负责窗口的管理,包括消息、菜单、光标、通信、计时器和控制

窗口显示。

(2) GDI32.DLL：提供图形设备接口、管理用户界面和图形绘制，包括 Windows 元文件、位图、设备描述表和字体等。

(3) KERNEL32.DLL：处理存储器低层功能、任务和资源管理等 Windows 核心服务。

1.1.3 句柄

句柄概念在 Windows 编程中是一个很重要的概念，在 Windows 环境中，句柄是用来标识项目的，这些项目包括模块、任务、实例、文件、内存块、菜单、控制、字体、资源、GDI 对象等。

Windows 程序中并不是用物理地址来标识一个内存块、文件、任务或动态装入模块的，相反的，Windows API 给这些项目分配确定的句柄，并将句柄返回给应用程序，然后通过句柄来进行操作。

句柄是 Windows 用来标识被应用程序所建立或使用的对象的惟一整数，Windows 使用各种各样的句柄标识，如应用程序实例、窗口、控制、位图、GDI 对象等等。

Win32 的编程可以归结为一句话：“创建或接收句柄，然后操作句柄”。例如，画一条直线的过程如下，“接收画面句柄，然后创建画笔句柄，接着用画笔句柄在画面句柄上画一条直线。”

1.1.4 程序入口点 WinMain

DOS 程序以 main 函数作为进入程序的初始入口点，在 Windows 应用程序中，程序的初始入口点是 WinMain 函数，WinMain 函数的原型如下：

```
int WINAPI WinMain (HINSTANCE hInstance,    // 当前实例句柄
                    HINSTANCE hPrevInstance, // 前一实例句柄
                    LPSTR lpCmdLine,         // 指向命令行参数的指针
                    int nCmdShow)            // 窗口的显示状态
```

在上面的函数原型中，可以看到一些以前不熟悉的数据类型，如 HINSTANCE 和 LPSTR 等，其实，很多这样的数据类型只是一些基本数据类型的别名。表 1-1 列出了一些在 Windows 编程中常用的基本数据类型的别名。

表 1-1 Windows 基本数据类型

Windows 中所用的数据类型	对应的基本数据类型	说明
BOOL	int	布尔值
BSTR	unsigned short *	32 位字符指针
BYTE	unsigned char	8 位无符号整数
COLORREF	unsigned long	用作颜色值的 32 位值
DWORD	unsigned long	32 位无符号整数，段地址和相关的偏移地址
LONG	long	32 位带符号整数
LPARAM	long	作为参数传递给窗口过程或回调函数的 32 位值

续表

Windows 中所用 的数据类型	对应的基本数据类型	说明
LPCTSTR	const char *(注 1)	指向可移植的 Unicode 和 DBCS 字符串常量的 32 位指针
LPCSTR	const char *	指向字符串常量的 32 位指针
LPSTR	char *	指向字符串的 32 位指针
LPTSTR	char *(注 1)	指向可移植为 Unicode 和 DBCS 字符串的 32 位指针
LPVOID	void *	指向未定义类型的 32 位指针
LRESULT	long	来自窗口过程或回调函数的 32 位返回值
UINT	unsigned int	32 位无符号整数
WNDPROC	long (__stdcall *)(void *,unsigned int,unsigned int,long)(注 2)	指向窗口过程的 32 位指针
WORD	unsigned short	16 位无符号整数
LPARAM	unsigned int	当做参数传递给窗口过程或回调函数的 32 位值

表 1-2 列出了常用的预定义句柄，它们的类型均为 void *，即一个 32 位指针。熟悉 Windows 数据类型的别名和常用句柄对了解和掌握 VC++ 程序代码非常有用。

表 1-2 Windows 常用句柄类型

句柄类型	说明
HBITMAP	保存位图信息的内存域的句柄
HBRUSH	画刷句柄
HCTR	子窗口控件句柄
HCURSOR	鼠标光标句柄
HDC	设备描述表句柄
HDLG	对话框句柄
HFONT	字体句柄
HICON	图标句柄
HINSTANCE	应用程序的实例句柄
HMENU	菜单句柄
HMODULE	模块句柄
HPALETTE	颜色调色板句柄
HPEN	在设备上画图时用于指明线型的笔的句柄
HRGN	剪贴区域句柄
HTASK	独立于已执行任务的句柄
HWND	窗口句柄

1.2 MFC 机制

1.2.1 MFC 是一个编程框架

MFC (Microsoft Foundation Class Library)中的各种类结合起来构成了一个应用程序框架,它是一种比 SDK 更简单的编程方法。其目的是让程序员在此基础上更容易建立 Windows 下的应用程序。

MFC 框架定义了应用程序的框架,并提供了用户接口的标准实现方法,程序员只要通过预定义的接口把具体应用程序特有的东西填入这个框架即可。Microsoft Visual C++提供了相应的工具来完成这个工作。

- AppWizard 可以用来生成初步的框架文件——代码和资源;
- 资源编辑器用于帮助直观地设计用户界面;
- ClassWizard 用来协助添加代码到框架文件;
- 编译并连接类库实现了应用程序特定的逻辑。

封装

构成 MFC 框架的是 MFC 类库即——C++类库,这些类封装了 Win32 应用程序编程接口、应用程序的概念、OLE 特性、ODBC 和 DAO 数据访问的功能等。

(1) 对 Win32 应用程序编程接口的封装。

例如,窗口类 CWnd 把 Windows 有关的 API 函数封装在 C++ window object 的成员函数内,后者的成员变量 m_hWnd 就是前者的窗口句柄。

(2) 对应用程序概念的封装。

使用 SDK 编写 Windows 应用程序时,要定义窗口过程等一系列复杂过程。MFC 把许多类似的处理封装起来,自动完成这些工作。

(3) 对 COM/OLE 特性的封装。

支持 OLE 的应用程序必须实现一系列的接口因而相当繁琐。MFC 的 OLE 类封装了 OLE API 大量的复杂工作,这些类提供了实现 OLE 的更高级接口。

(4) 对 ODBC 功能的封装。

MFC 提供与 ODBC 之间接口的 C++类,封装了 ODBC API 的大量的复杂的工作,提供了一种更简洁的编程模式。

继承

首先,MFC 抽象出众多类的共同特性,设计出一些基类作为实现其他类的基础。这些类中,最重要的类是 CObject 和 CCmdTarget。常用的 MFC 层次关系如图 1-1 所示。

CObject 是 MFC 的根类,绝大多数 MFC 类是其派生的,包括处理消息的类 CCmdTarget。CObject 实现了一些重要的特性,包括动态类信息、动态创建、对象序列化、对程序调试的支持等。所有从 CObject 派生的类都将具备或者可以具备 CObject 所拥有的特性。

针对每种不同的对象,MFC 都设计了一组类对这些对象进行封装,每一组类都有一个基类,从基类派生出众多更具体的类。这些对象包括以下种类:

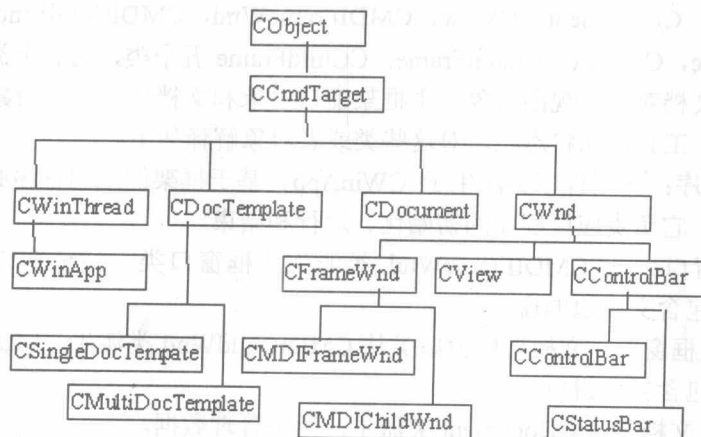


图 1-1 常用的 MFC 层次关系示意图

- 窗口对象，基类是 CWnd;
- 应用程序对象，基类是 CwinThread;
- 文档对象，基类是 Cdocument。

程序员将结合自己的实际，从适当的 MFC 类中派生出自己的类，实现特定的功能，达到自己的编程目的。

1.2.2 MFC 应用程序的构成实例

这里举一个典型的单文档应用程序的构成。

用 AppWizard 产生一个单文档应用程序工程 B，AppWizard 创建了一系列文件，构成了一个应用程序框架。这些文件分四类：头文件、实现文件、资源文件、模块说明文件。如图 1-2 所示。

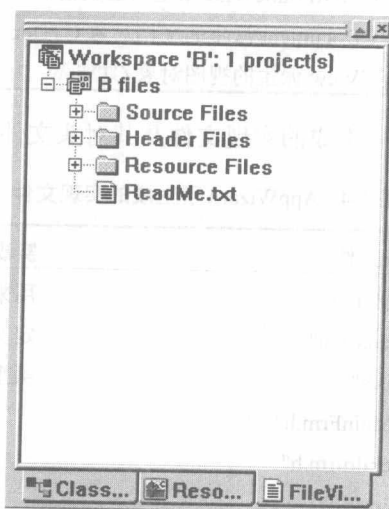


图 1-2 应用程序框架的文件分类

从 CWinApp, CDocument, CView, CMDIFrameWnd, CMDIChildWnd 类对应地派生出 CTApp, CTDoc, CTView, CMainFrame, CChildFrame 五个类, 这五个类的实例分别是应用程序对象、文档对象、视图对象、主框架窗口对象和文档边框窗口对象。主框架窗口包含了视图窗口、工具条和状态栏。对这些类或者对象解释如下。

(1) 应用程序: 应用程序类派生于 CWinApp。基于框架的应用程序必须有且只有一个应用程序对象, 它负责应用程序的初始化、运行和结束。

(2) 边框窗口: 从 CMDIFrameWnd 类派生边框窗口类, 边框窗口的客户子窗口 (MDIClient)直接包含文档边框窗口。

(3) 文档边框窗口: 文档边框窗口类从 CMDIChildWnd 类派生, MDI 应用程序使用文档边框窗口来包含视图窗口。

(4) 文档: 文档类从 CDocument 类派生, 用来管理数据。

(5) 视图: 视图从 CView 或它的派生类派生。视图和文档联系在一起, 在文档和用户之间起中介作用, 即视图在屏幕上显示文档的内容, 并把用户输入转换成对文档的操作。

(6) 文档模板: 文档模板类一般不需要派生。MDI 应用程序使用多文档模板类 CMultiDocTemplate; SDI 应用程序使用单文档模板类 CSingleDocTemplate。

表 1-3 列出了 AppWizard 所生成的头文件。

表 1-3 AppWizard 生成的头文件

头文件	用途
stdafx.h	标准 AFX 头文件
resource.h	定义了各种资源 ID
B.h	#include "resource.h" 定义了从 CWinApp 派生的应用程序对象 CTApp
childfrm.h	定义了从 CMDIChildWnd 派生的文档框架窗口对象 CTChildFrame
mainfrm.h	定义了从 CMDIFrameWnd 派生的框架窗口对象 CMainFrame
Bdoc.h	定义了从 CDocument 派生的文档对象 CTDoc
Bview.h	定义了从 CView 派生的视图对象 CTView

表 1-4 列出了 AppWizard 所生成的实现文件及其对头文件的包含关系。

表 1-4 AppWizard 所生成的实现文件

实现文件	所包含的头文件	实现的内容和功能
stdafx.cpp	#include "stdafx.h"	用来产生预编译的类型信息
B.cpp	# include "stdafx.h" # include "B.h" # include "MainFrm.h" # include "childfrm.h" #include "Bdoc.h" #include "Bview.h"	定义 CTApp 的实现, 并定义 CTApp 类型的全局变量 theApp