

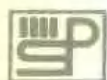
软件开发文集

第六辑

《软件开发文集》编委会 编



- 用 MFC 编写 Windows 95 程序(1)
—— 使用 Visual C++ 4.0 类和应用程序框架
- Visual FoxPro 中的 10 个最常见的错误
- Windows NT Server 3.51 评估指南(上)



科学出版社

软件开发文集

第六辑

《软件开发文集》编委会 编



科学出版社

1996

JS199/28

内 容 简 介

本书是为软件开发人员和计算机用户编写的《软件开发文集》系列书第六辑。本书向读者提供了开发技术规范、软件开发管理、开发平台、应用设计策略方面的技术资料。本辑的重点文章是:用 MFC 编写 Windows 95 程序(I)——使用 Visual C++ 4.0 类和应用程序框架, Visual FoxPro 中的 10 个最常见错误, Windows NT Server 3.51 评估指南(上)等。

本书适用于软件开发人员、计算机用户学习和参考。

图书在版编目(CIP)数据

软件开发文集 第六辑/《软件开发文集》编委会编.

-北京:科学出版社,1996.10

ISBN 7-03-005505-5

I. 软… II. 软… III. 软件开发-文集 IV. TP311.52-53

中国版本图书馆 CIP 数据核字(96)第 15668 号



科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

北京管庄永胜印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1996 年 10 月第 一 版 开本:787×1092 1/16

1996 年 10 月第 一 次印刷 印张:5

印数:1-3500 字数:120 000

定价:9.00 元

编委会名单

顾问

杜家滨 冯玉琳 秦人华 林资山

主 编

郑茂松

副主编

(按姓氏笔画为序)

刘晓融 李 浩 廖恒毅

编 委

(按姓氏笔画为序)

王淑兰 巴建芬 华京梅 查良钊

编者的话

近年来,我国的计算机产业发展迅速,PC机销量已居世界第六位。同时,我国有一批人数众多、技术水平相当不错的软件开发人员,他们分布在全国各地、各个行业、各个领域。然而,由于计算机产业,特别是软件产业充满了变化,极富动态性,加之,我国幅员辽阔,通信技术又落后于较发达的国家,致使,软件开发人员面临着这样的问题:了解最新技术动态不及时、不方便,获取最新技术信息比较困难,所得到的信息往往不够全面、完整和深入,相互之间缺乏交流。因此,重复开发、重复别人走过的弯路的情况屡见不鲜。另一方面,全世界积累的资料浩如烟海,价格亦非常昂贵,鉴于国内经济承受能力有限,许多技术信息不能及时获得。有鉴于此,我们组织有关人员翻译、编写了以技术专集为特色的《软件开发文集》系列书,该套书将陆续出版、发行,及时向用户提供最新技术资料,力图为解决上述问题做一点尝试。

《软件开发文集》系列书的宗旨是为软件开发人员开辟一个相互交流经验和体会的园地,提供一条了解新技术、新工具、新产品的渠道,设立一个探讨软件开发理论和开发技术的论坛,帮助软件开发人员更快、更直接地获得有关软件开发的信息,提高软件开发人员的技术水平和工作效率,充分发挥软件的作用,提高软件的使用价值,促进我国软件产业与世界同步发展。

《软件开发文集》是面向软件开发人员的中、高档次的技术专集,所收集的文章向软件开发人员提供了最新科技动态,以及开发技术规范、开发经验和技巧、软件开发管理、应用设计策略、开发平台方面的内容等,同时,根据用户在软件使用 and 开发过程中遇到的难点、疑点给予一定的解答和帮助。《软件开发文集》资料主要来源于微软公司的“Microsoft System Journal”,“Developer News”,“Microsoft Development Library”,“TechNet”等,并收入国内软件开发人员撰写的有关文章。欢迎国内广大软件开发人员为《软件开发文集》撰稿,分绍自己的经验、心得、体会。特别要注重文章内容适合我国国情。

目前,在国内面向计算机最终用户的图书、杂志丰富多彩,但面向软件开发人员的图书、杂志却寥寥无几。我们期望《软件开发文集》能够弥补这方面的空白,并且不辜负广大用户的期望,把提好学术方向,确定好信息服务和技术支持的层次和深度,把《软件开发文集》办成深受广大用户喜爱的丛书。

• ■ •

由于时间仓促,《软件开发文集》难免存在这样或那样的问题,敬请广大的用户及读者提出宝贵意见和建议,以利改进,为我国软件产业的发展作出应有的贡献。

《软件开发文集》编委会



和润于心 兼烛天下

和光集团是一家集计算机软硬件经营与开发,系统集成,通信技术,实业投资等多领域、多行业、多种经营模式为一体的现代化高科技企业集团。

和光集团自 1991 年创立以来,努力为员工创造发挥才能的机会和氛围,汇集了各方面的众多优秀人才,在信息产业经营方面已形成了一个具有雄厚技术实力和优质服务的开放的大市场网络体系,在全国各地拥有 20 余家分支机构,并在日本、香港等国家和地区设立了全资子公司,业务范围遍及海内外。和光集团同世界著名的企业结成了亲密的伙伴关系,作为美国微软公司(Microsoft)中国总代理和 IBM 中国总代理,和光不仅能为用户提供软硬兼备的计算机精品,同时拥有一系列完备的服务体系,已与铁路、公安、银行、邮电、石化、新闻、商业、学校等行业系统及诸多电脑公司建立起长期的合作关系。

和光集团自成立迄今,凭藉着前瞻性策略及“和谐、诚志、创意”的和光精神,始终把为用户服务作为首要任务,取得了良好的经济效益和社会效益。和光人不满足于今天的成就,作为辽宁省高科技优秀企业的和光集团,在未来的岁月里,愿与社会各界朋友携手合作,一同贡献于迅猛发展的中国计算机事业,为把和光发展成为国际型企业而努力奋斗。

和光精神:和谐、诚志、创意

和光宗旨:以和光精神培育和光企业,承担时代建设者使命,谋求社会的改善与发展,并贡献于世界文化。

和光集团各地合作伙伴:

东方飞鸿科技发展中心 (010)62549508 徐 新	杭州方欣电子技术开发公司 (0571)8835833 李 彬	广州方舟科技开发有限公司 (020)7514371 熊 健
连邦软件产业发展公司 (010)62568648 赵立奎	苏州新世纪软件开发中心 (0512)5296929 张未名	深圳曙光信息产业有限公司 (0755)3253778 申少军
北京清华网络系统技术公司 (010)62594746 詹 睿	南通范思软件公司 (0513)3567212 范 煜	珠海市万腾电脑有限公司 (0756)2211104 蔡 宇
金瑞通新技术发展有限责任公司 (010)62355892 罗晓农	南京培斯计算机软件公司 (025)4417907 夏振达	沈阳鸿合科技公司 (024)3908893 侯力朝
上海南洋微电子联营公司 (021)62828760 赵守国	武汉追梦信息技术有限公司 (027)7877509 张炜力	辽宁华储资讯科技有限责任公司 (024)3895234 王明义
上海电子计算机厂东海软件中心 (021)62476535 郭玉英	西南交大通用技术实业公司 (028)7784160 许慧民	大连铁路科技开发公司 (0411)2807730 马立辉
上海三爱司房屋科技实业公司 (021)64715865 黄海丽	成都科技大学科一实业公司 (028)5554364 邱 勇	大连电台传媒中心 (0411)4635492 张 文
杭州智星科技服务有限公司 (0571)8077618 金 渊	沈阳希望软件专卖店 (024)3909650 孙祖林	长春施乐贸易公司 (0431)5670145 温远忠

目 录

编者的话

1. 用 MFC 编写 Windows 95 程序 (I) 1
——使用 Visual C++ 4.0 类和应用程序框架
2. Microsoft MAIL 中文版 28
3. 关于 Access 的问题解答 33
4. 关于 Access Developer's Toolkit 的问题解答 37
5. Visual Basic 4.0 企业版的技术特性 38
6. 分而治之的设计方法 58
7. Visual FoxPro 中的 10 个最常见错误 63
8. 远程自动化的奥秘 69
9. Windows NT Server 3.51 评估指南 (上) 75
10. 用 GryptoAPI 保卫你的应用程序 84

用 MFC 编写 Windows 95 程序(1)

——使用 Visual C++ 4.0 类库进行开发

编者按:从本辑起,我们将陆续连载 Jeff Prosise 撰写的《用 MFC 编写 Windows 95 程序》一书,希望对我国广大软件开发人员学习 MFC 有所帮助。

过去,大多数基于 Windows 的程序都是用 C 语言写的,今天,这种情况正在迅速发生变化。现有的各种各样的程序设计环境意味着你可用 C, C++, Pascal, BASIC 和许多其他语言来写商用性程序。其中, C++ 似乎注定要代替 C 来作为专业程序员选择的语言。C++ 和一个类库相结合,该类库用表示公用对象(例如窗口和菜单)的基类来抽象 Windows API,从而帮助学习 Windows 编程避免走弯路,并为缺乏经验的程序员消除一些障碍。

许多 C++ 程序员都确定在 Microsoft Foundation Class(Microsoft 基础类;MFC)库上,将其作为 Windows 下开发所选择的类库,MFC 的支持者并不相信它是什么灵丹妙药,但它确实可以使人免去花费更多时间开发程序的结构部件,而且不必过多担心 Windows API 过多的细节。MFC 简化了用于如 16 位 Windows, Win32, 甚至 Macintosh 之类的不同平台的代码开发过程。一个代码基很大程度上可服务于这三个平台,而产生一个本地可执行文件,你全部要做的就是使用 Visual C++, 或你选择的 MFC 编译器(例如 Symantec C++)及重新编译。

本文是关于用 MFC 去写基于 Windows 应用程序的系列文章中的第一篇。当基本焦点是围绕用 MFC 3. X 进行 Windows 95 和 32 位应用程序开发时,该系列文章中出现的绝大多数概念和样板代码,也适用于 MFC 1.5 X 的 16 位的 Windows 开发及 Windows NT。无论你对 Windows 是新手还是正在考虑转向 MFC 的富有经验的开发者,本文(以及它的后续文章)将为你提供关于 MFC 的第一手资料。为了在小的篇幅中提供尽可能多的信息,假定你已经熟悉基于消息的 Windows 程序设计模型及 C++ 的基本特性,如类和继承性。你不必一定是个解决 MFC 的权威。MFC 使用一个 C++ 合理的子集,它对从 C 和 SDK 转来的程序员降低了要求。

(一) MFC 导引

MFC 的最新版本是 3.1,它随同 Visual C++ 2.1 提供并支持 32 位的应用程序开发。一般说来,无论什么时候修改 Visual C++,同时也修改 MFC,MFC 版本号总是等于

Visual C++ 的版本号加 1。但 Visual C++ 3.1 也支持 MFC 1.5X 用于 16 位的应用程序的开发。将来有可能 16 位和 32 位版本的 MFC 要修正,并独立编号。32 位版本要做很大增强。

MFC 3.X 包含 130 多个类,其中某些你将直接使用;某些将基本用作你自己的那些类的基类。有些类非常简单,如封装一个点对象的 CPoint 类(位置由 X 和 Y 坐标表征)。其他类就复杂了,比如 CWnd 类,它封装你在屏幕上看见的窗口。

在 MFC 程序中,你很少会直接调用 Windows API,而是从 MFC 类创建对象并调用那些对象的成员函数。数百个 MFC 成员函数中许多都是环绕 API 的薄薄的一层包装,许多甚至有和对应的 API 函数相同的名。这就加速 C 程序员向 MFC 的过渡过程。想移动一个窗口吗? C 程序员或许会调用 SetWindowPos API 函数。在 MFC 参考手册中查找 SetWindowPos 你将会发现 MFC 也支持 SetWindowPos。它是 CWnd 类的一个成员,这就懂得了什么时候你把一个窗口作为一对象考虑,把 SetWindowPos 作为你可能想对该对象做的某事来考虑。

MFC 也是一个应用程序框架。它帮助定义应用程序的结构,以及为应用程序处理许多过程杂务。从 CWinApp 表示应用本身的类开始,MFC 实际上封装一个程序操作的每一方面。框架提供 WinMain 函数,WinMain 又调用应用对象的成员函数使程序执行。WinMain 调用的 CWinApp 成员函数之一封装消息循环并逐字运行程序。框架还提供超出 Windows API 的抽象概念。MFC 支持文档和视图(允许程序数据与该数据的图形表示分离的范例)。MFC 也引进它自己的几个 API 函数,这几个函数与它们自己的类无关并且全程有效。

大部分 MFC 类都归入这六个范畴之一:

- CObject
- 应用体系结构
- 可视对象
- OLE 2
- 数据库
- 通用

让我们看看 MFC 提供的这些类及 API 函数。

(二) CObject, 所有类之母

注视 MFC 分层图表你就会发现多数 MFC 类直接或间接地由 CObject 派生而来。有时你会把 CObject 作为你自己那些类的基类;但更多时候你将利用它增加到内置的 MFC 类上的特性。CObject 对继承它的那些类提供三种基本类型支持:

- 串行化
- 运行时的类信息
- 诊断输出

串行化是对象本身往返于一种永久性存储介质,如光盘驱动器,或非永久性存储介质,如 Clipboard 存储器,的存档过程。CObject 定义两个在串行化进程中起作用的成员函数: IsSerializable 和 Serialize。程序可调用一个由 CObject 派生的对象的 IsSerializable 函数,以

确定是否该对象支持串行化(串行化支持是任选项,不是要求)。包含在建立支持串行化的类的步骤之一是重载继承 CObject 的 Serialize 函数,并提供串行数据成员类专用代码。直接从 CObject 派生的 CBirthday 类型对象的 Serialize 函数看起来可能是这样:

```
void CBirthday::Serialize (CArchive& ar)
{
    CObject::Serialize (ar);
    if (ar.IsStoring)
        ar << m_day << m_month << m_year;
    else
        ar >> m_day >> m_month >> m_year;
}
```

对象首先调用基类的 Serialize 函数,这样基类就可存档它的数据成员。然后,它调用 CArchive 对象的 IsStoring 函数以确定数据是否正在流入或流出,并利用由 CArchive 提供的“或”运算符以保存“或”还原它自己的数据成员。CArchive 对象与 CFile 对象一起工作进行物理磁盘的 I/O,并缓冲进行出出的数据使性能优化。通过处理你的程序作为对象存储的数据,并从 CObject 派生用于它的类(或用 MFC 提供的几个 CObject 派生物中的一个),你可建立你的程序,以便把数据保存在磁盘中,且读起来相对轻松一些。

继承 CObject 的由 CObject 派生的类的另一特性是支持运行时的类型信息。如果一个通用的 CObject 指针指向由 CObject 派生的类时,运行时的类型信息机制就允许程序检索对象的类名及其他信息。CObject 提供两个成员函数支持这点: IsKindOf 和 GetRuntimeClass。后者返回一个指向 CRuntimeClass 结构的指针。该结构除了别的东西之外,还包含一个指向类名的指针和一个指向描述基类的 CRuntimeClass 结构的指针。MFC 的 RUNTIME_CLASS 的宏提供一种类似的功能。IsKindOf 返回 TRUE 或 FALSE 指示一个对象是否属于规定的类或是从规定的类派生出的类,例如如果 pObject 参考的对象由 CWnd 或它的一个派生物来说明,则语句

```
pObject->IsKindOf (RUNTIME_CLASS (CWnd))
```

就返回 TRUE。

CObject 加到它的派生类的第三个特性是诊断支持。虚拟成员函数 AssertValid 指示一个对象利用一系列断言对其本身进行有效性检查。虚拟成员函数 Dump 进行对象的数据成员的诊断转储,这常常对调试程序及它使用的对象有所帮助。由 CObject 派生的类一般用指定类代码重载这些函数以便进行有意义的有效性检查和诊断转储。CBirthday 类可能实现这些函数如下:

```
void CBirthday::AssertValid () const
{
    CObject::AssertValid ();
    ASSERT (m_day > 0);
    ASSERT (m_month > 0);
    ASSERT (m_year > 0);
    ASSERT (m_day <= 31);
    ASSERT (m_month <= 12);
}
```

```

;
void CBirthday::Dump (CDumpContext& dc) const
{
    CObject::Dump (dc);
    dc << "Birthday = " << m_month << " " << m_day \
        << " " << m_year;
}

```

dc 常参考预先定义的 afxDump 对象,后者用 Windows OutputDebugString 函数将输出转储给调试终端。

CObject 对它的派生的类也提供了其他好处。例如:它重载新的和删除的操作符,以防止内存漏泄。随着你对 MFC 更加熟悉,这个最基本的 MFC 类的重要性将变得越来越明显。

(三) 应用体系结构类

应用体系结构类影响一个 MFC 应用程序的表单和结构。CWinApp 表示应用程序本身。几乎所有的基于 MFC 的应用程序都从 CWinApp 派生出一个类,并例示派生的类以便创建一个应用程序对象。CWinApp 包括若干有用的公用数据成员,如 m_lpCmdLine。它使指针定在转向应用程序的命令行参数上。它也包括有用的成员函数,如 GetProfileString,该函数从系统登记处或私有 INI 文件阅读程序设置。以及 InitInstance,应用程序重载它来进行例程起动杂务。例如,建立一个窗口。这个范畴还包括 CDocument 类,即利用 MFC 的文档/视图体系结构的应用程序中文档的基类。文档就是程序数据的抽象表示。CDocument 允许应用程序保持数据独立,和包括屏幕上画出的那些数据的视图在内的程序的其他成分脱离。在文档/视图程序中,数据是一个对象。例如,C 程序可能含有一个全局标志,该标志指示一文档自它最后被保存以来是否修改过。MFC 程序可以把文档作为自身的实体处理,调用 CDocument 成员函数 SetModifiedFlag,设置(或清除)修改过的标志和调用 IsModified,以便确定文档是否是修改过的。它提供的许多好处,不只是框架知道什么时候——文档包含未存盘的数据,并提示用户在文档关闭之前保存它。

CWinApp,CDocument 及许多其他类都是从用于 MFC 的消息映像体系结构的基础类 CCmdTarget 派生来的。MFC 程序一般不用窗口过程处理消息,用成员函数处理消息;消息到达时,框架激活成员函数。一个类继承 CCmdTarget 的能力使它可以作为一消息的目标。只有 CCmdTarget 派生的类才能接收和处理消息。这就是为什么所有的 MFC 的应用程序、窗口、文档和视图类直接或间接地从 CCmdTarget 派生出的原因。

另一个重要的类是 CWinThread,它封装操作系统的线程化功能。CWinThread 对象表示一个执行的线程,成员函数如 CreateThread,SetThreadPriority 及 SuspendThread 提供 MFC 程序用来创建和操作线程的工具。CWinThread 还通过利用线程局部数据来维护个体线程特定信息,使 MFC 程序线程安全。CWinApp 本身是从 CWinThread 派生出来的,因为在多线程环境中,例如 Windows 95,每一应用程序均以执行一个单线程形式开始其生存的。

(四) 可视对象类

当一个基于 Windows 的程序运行时,可视对象类封装屏幕上可视的大部分对象,包括窗口、对话框、菜单及各种类型的控件。它们还封装 Windows 设备描述表及 GDI 图形对象,如位图、画刷和画笔。该范畴中更为抽象的类之一是 CView,它是 CDocument 的副本及你的数据的特定应用视图所用的基类。

所有可视对象类的“祖父”是用作所有其他窗口类的基类的 CWnd,CWnd 定义窗口的基本功能,并包括数十个用各种方式操纵窗口的成员函数。CWnd 还定义窗口对大多数消息的默认响应。CWnd 有时被直接使用,但更多时候将从 CWnd 派生的类中派生类,并或许例示 MFC 从 CWnd 派生的那些类。这些派生出的类包括在其中的 CFrameWnd,它模型化单文档界面(SDI)框架窗口(为应用程序服务,用来一次处理一个文档的顶层窗口)的行为;CControlBar,它作为工具栏、状态栏和其他 MFC 控件的基类;CDialog,封装对话框的功能;你可以猜 CButton、CListBox、CScrollBar 与什么相对应。扩展 Windows API 的类(与仅仅包装它相反)的一个例子是由 CWnd 派生出的 CSplitterWnd 类,它使你容易在你所见的应用程序,如 Visual C++ 中添加拆分窗口到你自己的程序中。

CView 类也是从 CWnd 派生的。CView 是用于所有的支持文档/视图应用体系结构的视图类的基类。其他视图类包括添加滚动能力到 CView 的 CScrollView;支持由对话框模板创建的基于表单视图的 CFormView;将 CFormView 型视图连到 CRecordset 数据库对象上的 CRecordView;以及借助打印、搜索/替代、剪切/复制/粘贴操作;利用一个 Windows 编辑控件提供一视图到文本数据中的 CEditView。

其他可视对象类不是从 CWnd 派生来的,而代之直接从 CObject 来。CDC 类,封装 Windows 的设备描述表的功能。CDC 成员函数,如 Draw Text 和 LineTo 包装大部分 GDI 函数,并提供在窗口中绘图的手段。CDC 派生物如 CPaintDC,CWindowDC,CClientDC 表示更为普通的 CDC 类的特例。例如:CPaintDC 从它的构造程序和析构程序自动调用 BeginPaint 和 EndPaint API 函数,简化对 WM_PAINT 消息的处理;其间 CMetaFileDC 类提供面向对象的 GDI 图元文件的包装。

CGdiObject 类用作 GDI 对象,如笔、刷和字型的 MFC 的表示。而由 CGdiObject 派生的类,如 CPen,CBrush,CFont 表示对象本身。在 MFC 中,指定一支 GDI 笔就涉及例示来自 Cpen 类的对象及调用定义笔特性的成员函数。一旦用 CDC(或由 CDC 派生的)类的 SelectObject 函数选择到一设备描述表时,笔就可以绘图了。该范例创建一对象,将其选入一个设备描述表中,然后用它来作图,准确模仿 GDI 绘图模型,对此,C 程序员已习惯去做了。由 CGdiObject 派生出的其他类包括:CBitmap;表示逻辑调色板的 CPalette,以及封装屏幕上定义为矩形、多边形和椭圆形组合的不规则区域 GDI 对象的 CRgn。

可视对象范畴的最后一类是 CMenu,它提供一个面向对象的菜单界面。CMenu 成为一个方便类,正像你经常想用菜单做的每件事一样。从加载到修改一菜单到复选菜单项,基本上是 API 中提供的各种及其他等无联系的菜单函数的混合。框架也在 CMenu 之外工作,以加强选单系统。例如无论何时你下拉一个菜单时,可登记一个 CCdUI 菜单信息处理程序,并用指向 CCmdUI 对象和指针去调用它。CCmdUI 对象的成员函数允许信息处理程序在菜单实际显示前更新对应的菜单项的状态;激活或失效、复选或非复选等等。用

MFC 保持菜单项和状态度量同步,这就比它在 Vanilla C 中麻烦少一些。

(五) OLE 类

MFC 的最大优点之一是它简化了在你的应用程序中添加 OLE 支持。OLE 是一个复杂的协议,它定义标准界面使对象和对象的用户相互通信。OLE 是原位激活及编辑(也称之可视编辑),拖放数据传送,结构比存储及复合文档,OCX 控件等的基础,是更为实用的一级。原来想使由一应用程序创建的文档能被链接或嵌入在由另一应用程序创建的文档中。OLE 后来被扩展成为 Windows 进化到一个面向对象的,以文档为中心的操作系统中心。它也是你再不能忽视的东西了。大多数应用程序将必须包括 OLE 支持,以便具有 Windows 95 徽标的资格。另外,OLE 正逐渐成为 Windows 本身编织的“纤维”。如果你不相信,就检查一下 Windows 95 外壳、用来和外壳扩展的 DLLs 文件浏览器及其他扩充的用户界面部件、通信的 OLE 界面。

MFC 提供一个 OLE 类的分类。这些类由 20 000 多行预先写入的代码组成,使所要求的写入和调试程序的 OLE 部分的时间减少一个数量级。简言之,它对写入大部分 MFC 之外的 OLE 应用程序是有些意义的。

MFC 的 OLE 类可分为 6 个范畴:

- 基类
- 可视编辑容器类
- 可视编辑服务器类
- 数据传送类
- OLE 对话类
- 杂项类

基类如 COleDocument 和 CDocItem 封装复合文档(包含由其他应用程序创建对象的文档)及那些文档中的对象,并用作更为专业化的 OLE 类的基类。可视编辑容器类。如 COleClientItem 及 COleLinkingDOC 提供用于 OLE 容器的基础结构去支持可视编辑。同时,有些类,如 COleServerDoc 和 COleServerItem 在服务器方起类似作用。数据传送类。COleDropSource,COleDropTarget,COleDataSource,及 COleDataObject 封装用来借助拖放及通过 Clipboard 进行 OLE 数据传送的界面。OLE 对话类。如 COleInsertDialog 显示标准的 OLE 对话框。杂项类。如 CRectTracker,它围绕一个插入在复合文档中的项建立边框,这样可使该项移动和调整大小,它们填充了其他类之间的某些裂缝。

(六) 数据库类

为了支持向带有 ODBC 驱动程序的各种数据库管理系统提供标准化界面的 Open Database Connectivity(ODBC,开放数据库互连)标准,MFC 提供 CDatabase 和 CRecordset 类。CDatabase 表示和数据源的一种 ODBC 连接,和 CRecordset 表示那个数据源中的一组记录。同时,这些类抽象 ODBC API,并允许你用继承自 CRecordset 的成员函数把存储在数据库中的数据作为被查询、更新和用其他方式操作的对象来处理。

一个典型的利用 ODBC 的 MFC 应用程序通过创建 CDatabase 对象并调用对象的 Open 函数来建立与数据库的连接。然后,它从一个由 CRecordset 派生的类创建记录集对

象(向它的构造程序传送一个指向 CDatabase 对象的指针);该派生的类包含模仿数据库记录中的域的数据成员,并通过调用继承 CRecordset 的 Open 函数打开该记录集。当应用程序利用记录集对象的 MoveNext 和 MovePrev 函数滚动记录时,数据成员被自动更新以反映存储在当前记录中的数值。当前记录可由调用 Delete 删除,或由调用 Edit 接着 Update 来修改。由于与数据库的物理交互作用要通过 ODBC 驱动程序进行,因此,不必担心记录存储的格式。完成时,应用程序调用记录集及数据库对象的 Close 函数来终止连接。

由 CFormView 派生的 CRecordView 类进一步通过连接记录集对象和显示当前记录中的域值的类似对话框视图来简化操作。CLongBinary 类提供大的二进制对象(如有时存储在数据库域中的位图)的抽象表示。当它碰到作为记录集对象中的数据成员实现的 CLongBinary 对象时,MFC 的记录域交换(RFX)机制分配内存以存储对象并从数据库装入它。这样就使程序员不必提供读写不是正好适合预定义数据库类型的数据库域的专门逻辑。

(七) 通用类

不是所有的 MFC 的类都设计成封装 Windows API 的。MFC 包括许多通用类用于各种各样任务,如管理字符串变量及进行文件 I/O。特别有用的 CString 类定义类似 Basic 的串数据类型,带有标准操作符,如 =、+=、<and> 用于进行赋值、级连和比较。CFile 类及它的派生物采用标准的文件 I/O 函数,如 Read 和 Write,并在文件对象的上下文中实现它们。CFile 对象常常通过 CArchive 对象被间接访问;后者表示二进制数据流,并利用过载插入和抽取操作符(即 >> 和 <<)来流动基本数据类型,甚至往复磁盘的全部对象。

CException 是一个抽象基类,供 C++ 的 try/throw/catch 异常处理机制使用。由 CException 派生的 CFileException 描述由文件操作产生的异常。公用数据成员包含显示引起异常的错误的确切特性的出错代码。经常使用的其他异常类包括用于内存不够错误的 CMemoryException,用于档案文件错误的 CArchiveException,用于 OLE 错误的 COleException,以及用于装入 Windows 资源错误的 CResourceException。MFC 从它自己的代码中抛出各种类型的异常,这样,应用程序就可利用捕获程序块检测错误情况和据此做出反应。MFC 还提供全局函数(如: AfxThrowFileException 和 AfxThrowOleException)来帮助应用程序抛出异常。

MFC 聚合类,提供对数组及链接到表之类的公用数据结构类库支持。数组类 CByteArray, CIntArray, CStringArray 及其他封装可动态调整大小的数据类型和对象的数组。不像 C 程序中使用的静态数组。数组类对存储应用程序的文档数据是理想的。例如,一个完整的文字处理文件,或许可作为一个 CByteArray 对象来实现,该对象随着用户类型文本发展成文档(或随文本删除缩小)。列表类,像 CStringList 和 CObList 实现双链接列表的 MFC 版本,它们的成员包含指向列表下个和前个元素的指针。成员函数,像 InsertBefore, InsertAfter 及 RemoveAt 增加对例程的插入和删除操作的支持。映象类支持对象的表格和用其他数据类型键入或映射的数据类型。例如, CMapWordToPtr 类键入每一个它存储的调用程序供给 16 位字值的指针变量。已经键入的值,则对应的指针就易于通过调用对象的 LookUp 函数检索。MFC 还以 C++ 模板的形式提供这些类的一般版

本。而这些类最值得人们注意的是,由于它们支持串行化,所以,一个完整的数组、列表或表格就可用一行代码存档到磁盘。当你开发 MFC 应用程序时,你可反复使用这些类。

其他通用类包括 CPoint, CSize, CRect, CTime 和 CTimeSpan。CPoint 和 CRect, 表示点和矩形,在 Windows 中特别有用,因为点和矩形常常作为参数传送到 MFC 成员函数。诊断类 CDumpContext 和 CMemoryState 进行诊断转储及检测由于故障引起的内存漏泄,以便删除用 new 操作符例示的对象。

(八) AfxXxx 函数

不是 MFC 提供的所有函数都是类的成员。MFC 以其名开始为 Afx 的全局函数形式提供属于自己的 API(Afx 中的 A 和 f 代表应用程序框架。符号中添加“X”是因为要产生未来音的缩写)。这些函数可从任何地方、在任何 MFC 应用程序中调用,不像成员函数,要求一个对象或对象指针。

表 1 列出你可能有用的几个 AfxXxx 函数。AfxBeginThread 函数常常用来代替 CWinThread::CreateThread,因为它用一个单步创建和执行一个线程。AfxMessageBox 等同 Windows MessageBox 函数,并且不像 CWnd::MessageBox 不用指向一个由 CWnd 派生的类的指针即可调用。AfxFormatString1 经常同 AfxMessageBox 一起使用完成像 printf 类似的字符串格式化。一个字符串代替另一个中的 %1 一个占位符,并将结果复制到 CString 对象上。一个相似的命名为 AfxFormatString2 的函数接受 2 个字符串及两个占位符(是的,%1 和 %2)。AfxGetApp 对于获得指向在另一源代码文件中说明的应用程序对象的指针是有用的,并且当你需要一个实例句柄传给 Windows API 函数时,AfxGetInstanceHandle 是便利的(即使 MFC 程序不时地调用 API 函数!)。最后,当 MFC 登记的默认窗口类恰好不想做,而你却需要登记属于你自己的窗口类时,AfxRegisterWndClass 是不可缺少的。你将会看到一个例子,即 AfxRegisterWndClass 以及表 1 中示出的其他 Afx 函数在本文的后面是如何使用的。

表 1 AfxXxx 函数样板

函数名	描 述
AfxAbort	无条件终止一个应用程序。通常在一个不可恢复的错误出现时调用
AfxBeginThread	创建一个新线程并开始执行它
AfxEndThread	终止正在当前执行的线程
AfxFormatString1	用一个指定的字符串代替一个串资源中的 %1 实例,并将结果字符串复制到 CString 对象上
AfxMessageBox	显示一个 Windows 消息框
AfxGetApp	将一指针返回到应用程序对象上
AfxGetInstanceHandle	返回标识当前应用程序实例的句柄
AfxRegisterWndClass	登记用于 MFC 应用程序的定制窗口类

(九) 第一个 MFC 应用程序

现在你准备建立第一个 MFC 应用程序了。什么地方开始比用窗口中显示消息 Hello,

MFC 的位置更好呢? HELLO 1 演示一些涉及使用 MFC 的基本原理。其中,你将要了解 CWinApp 和 CFrameWnd 类。还要了解消息映象,即 MFC 用来和消息发生关系的机制,消息则是你的应用程序接收给 MFC 和由 MFC 派生出的类中的成员函数的东西。最后,你将学到更多的关于最重要的 CPaintDC 类的内容。它起导管作用,通过它在一窗口中给出文本和图形以响应 WM-PAINT 消息。还有 CRect 类,它是 MFC 提供的帮助 Windows 适应 C++ 样板的几种通用类之一。

图 1 列出 HELLO1 的两个源码文件。HELLO1.H 包含 HELLO1 用的两个派生类的说明。HELLO1.CPP 包含实现这些类的 C++ 源码。C++ 程序员中,传统的是把类定义放在 H 文件中,源码放在 CPP 文件中。但不一定必须这样。你可把一个程序的源码分割成你喜欢的任何方式。今天,可视编译程序通过采用文件列表为你产生构造文件,使其容易了。对于大型应用程序,你也许会发现每类写入一个 H 文件和一个 CPP 文件是有利的。HELLO1 可用任何支持 MFC 的编译程序建立。

```
HELLO1.H
class CMyApp : public CWinApp
{
public:
    BOOL InitInstance ();
};

class CMainWindow : public CFrameWnd
{
Public:
    CMainWindow ();

Protected:
    afx_msg void OnPaint ();
    OECLARE_MESSAGE_MAP ()
}

HELLO1.CPP
#include <afxwin.h>
#include "hello1.h"

CMyApp myApp;

////////////////////////////////////
// CMyApp member functions

BOOL CMyApp::InitInstance ()
{
    m_pMainWnd = new CMainWindow ();
    m_pMainWnd->ShowWindow (m_nCmdShow);
    m_pMainWnd->UpdateWindow ();
    return TRUE;
}

////////////////////////////////////
// CMainWindow message map and member functions
```