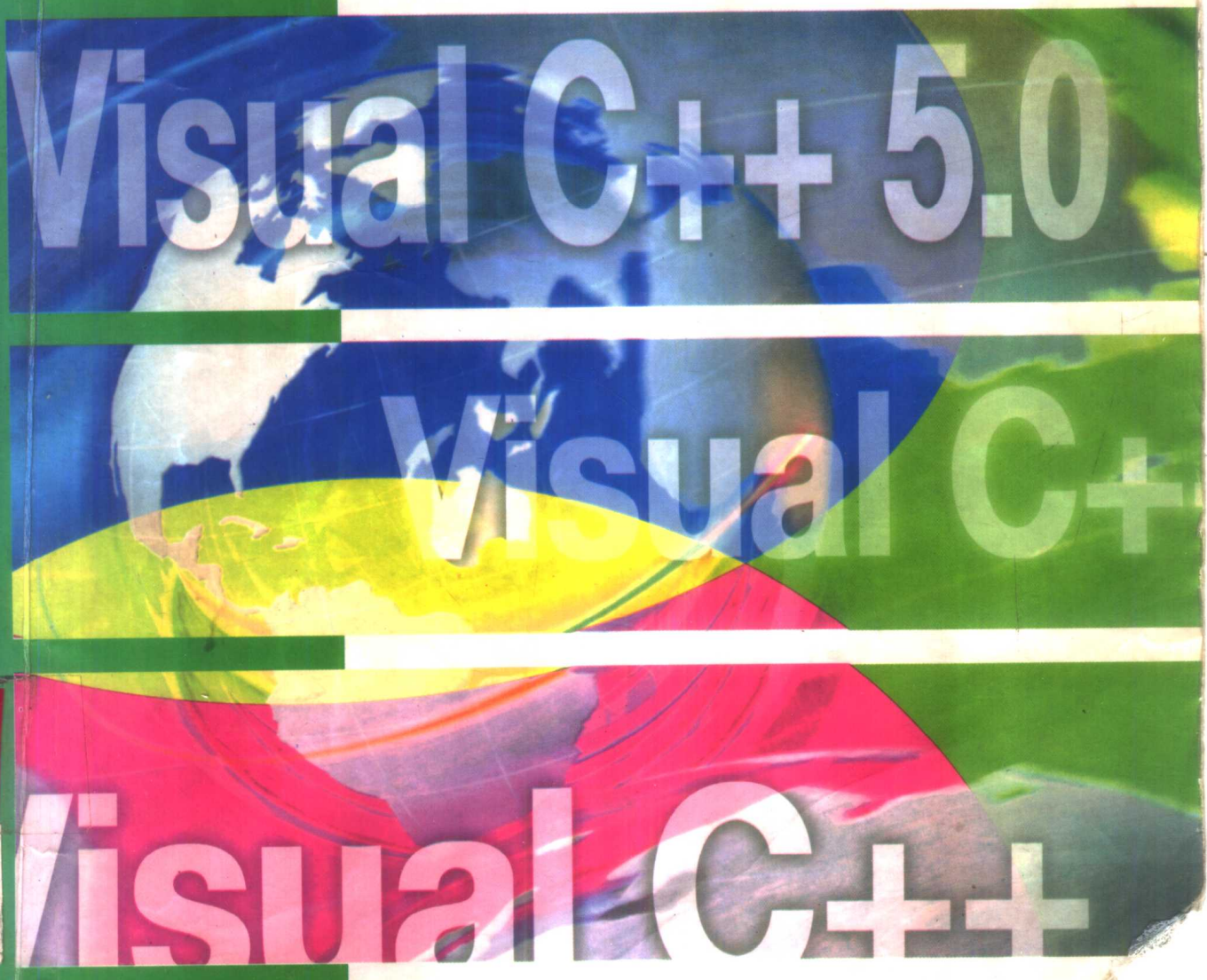
Visual C++ 5.0
程序设计教程



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL:http://www.phei.com.cn

Visual C++5.0 程序设计教程

程耀　宋守许　胡立　编著

電子工業出版社
Publishing House of Electronics Industry

内 容 简 介

Visual C++历经版本 1.0、1.5、2.0、4.0、4.2，于 1997 年 3 月， Microsoft 公司发布 VC++ 5.0。历次版本更新过程中，VC++保持了“应用程序框架”(Application Frame)的编程方法，并不断扩充 MFC 的内容，逐步面向 32 位内存方式，进一步扩展网络编程功能(特别是 Internet)，不断改善各工具的集成性能。

本书基于 VC++ 5.0，通过实例，循序渐进地介绍 VC++ 5.0 的编程方法和实现各种功能的具体措施，如文档-视结构的实现、各种消息的处理、GDI 绘图、对话框编程、工具条、DLL 编程、文档打印、初始化文件、VC++ 5.0 组件库的使用等。

初学者可以通过本书迅速掌握 VC++ 5.0 的编程方法，有一定经验的读者也可以通过本书掌握 VC++ 5.0 的新特点和相应的编程方法。

本书适合于有一定 *C/C++*和 Windows 知识的编程人员参阅，是否有 VC++编程经验并不重要。

书　　名：Visual C++5.0 程序设计教程
编　　著：程耀　宋守许　胡立
责任编辑：赵　平
印 刷 者：北京大中印刷厂印刷
出版发行：电子工业出版社出版、发行
北京市海淀区万寿路 173 信箱　邮编：100036　发行部电话　68214070
URL: http: //www. phei. com.
经　　销：各地新华书店经销
开　　本：787×1092　1/16　印张：36.5　字数：934.4 千字
版　　次：1998 年 9 月第 1 版　　1998 年 9 月第 1 次印刷
书　　号：ISBN 7-5053-4810-8
TP・2335
定　　价：44.00　元
凡购买电子工业出版社的图书，如有缺页、倒页、脱页者，本社发行部负责调换

引　言

Microsoft 公司的 Visual C++是世界上最优秀的面向对象编程环境之一，自推出以来，一直受到人们的关注和广大程序员的欢迎。它的以 MFC 应用框架为基础的编程方法将编程环境提供的代码和资源编辑器、编译器、连接、调试器、AppWizard、ClassWizard、Browser 等不同编程阶段使用的工具天衣无缝地结合在一体，大大提高了代码编制的自动化程序，使得编程工作简洁、高效。

Visual C++ 5.0 是当前最新的 VC++版本，它与 VB、VJ 等成为 MS Develop Studio 的一部分，除了秉承以前版本的各项优点以外，还不断扩充 MFC 的内容，逐步面向 32 位内存方式，进一步扩展网络编程功能(特别是 Internet)，不断改善各工具的集成性能。

因此，尽管 VB、VJ 日益受到人们的关注，VC++ 5.0 仍以最能体现编程者的风格、编码效率高等特性，为广大程序员所钟爱。

本书从初学者的角度出发，认真剖析用 VC++ 5.0 编程时的步骤和方法，并通过大量的实例展示编程效果。如 MFC 应用框架的核心——文档-视结构，应用程序中的消息处理、应用程序中的绘图机制 GDI、各种对话框的编程、工具条、状态条和对话条的实现、DLL 实现、应用程序的参数存储、VC++ 5.0 提供的组件的使用等。有了这些知识，我们就可实现一个功能相对完备的应用程序了。

同时，本书并没有停留在初学者的水平上，几乎在编程的每个方面，都给出了更高、更深层次的方法。如在第 7 章中，除了给出基本对话框的编程方法外，还举例说明了其它各类变型的对话框(卡片式对话框、常用对话框的扩展)的实现方法，并在最后介绍了在对话框中使用各种控制的方法。又如，在第 8 章中讲述菜单时，略去了一般菜单的实现方法(太简单了)，而是举例说明动态地改变菜单的方法；在说明工具条的编程方法时，不但给出了一般工具条的编程方法，还举例实现两种方式的工具条：检查框式和圆按钮式工具条，并在工具条中插入一个列表框。其它例子还可举出若干，如第 10 章 DLL 编程，第 11 章初始化文件等等，读者在阅读本书时会有更深的体会。

本书并不试图涉及 Visual C++ 5.0 的所有方面，如 ActiveX、Internet 编程。一是因为它们的内容很多，本身就可独立成书；二是因为目前这些技术正处于不断发展变化之中，很可能出现其它实现方法。

本书的大部分实例是环环相扣的，是同一个应用程序的不同功能模块。在第 3 章的实例中建立了一个应用程序 Step，以后各章的实例人都从这个应用程序出发，逐步丰富程序的功能，直到最后一章，形成一个完整的应用程序。也有一些实例是新建的应用程序。

在给出各例的实现代码时，为节省篇幅，并尽可能保持代码的可读性，本书遵循以下规则：

1．只给出有变化的文件的代码。其它文件的代码参阅前例；
2．在有变化的文件中，.h 文件全部列出，.cpp 文件只列出有变化的函数的代码，其它函数体长度若大于一行，则用省略号“.....”代替，具体代码参见前例；
3．个别有变化的函数体中，用了“见 xx 节步骤 x”的描述，请参见指定章节的代实现步骤。

本书的几位作者都是具有VC++编程经验的软件开发人员，在本书中注入了各自的编程体会，力争在有限的篇幅中给予读者更多。冷文浩、李陆、王涛、刘士健在本书编写过程中对本书内容的编排提出了许多有益的建议。汪诚、李晓阳、高石参与完成了本书的部分实例。张玉玲、张凡一、贾烨在录入文稿、编辑图片、打印和校对方面做了大量工作。陈睿、宋韬、赵华书阅读了书稿并从使用者的角度提出了修改意见。在此一并致谢。

书中不当之处，欢迎指正，以期共同提高。

作者　　1998年3月

目　录

第 1 章　Windows 编程和 Visual C++ 5.0

Windows 下的应用程序与 DOS 程序有着显著不同。从表面上看，Windows 应用程序中一般有对话框、窗口、菜单等资源；用户需要使用鼠标或键盘等交互手段向程序发送消息，通知程序完成一定的动作；Windows 程序中可以打开一个或多个窗口；除了一般的 EXE 可执行文件外，DLL 也是 Windows 应用程序的一种特殊表现形式。从更深的角度来讲，32 位 Windows 下的应用程序已经根本打破了 DOS 系统 640KB 界限的障碍，在程序风格、界面、编程风格等各方面有了巨大的差异。在当今 Wintel(Windows 和 Intel)体系主宰 PC 机的时代，熟练掌握 Windows 编程技巧是大多数程序员的必备技能。事实上，在国外，Windows 程序员也比其它程序员更具有竞争力。

Visual C++是软件巨人 Microsoft 公司的重要产品，是一个集成度很高的软件开发工具，适合多种形式的软件开发。历经多次版本更新后，Microsoft 于 1997 年 3 月推出了 Visual C++ 5.0。

本章主要向读者强调一下 Windows 程序的几个重要方面，如消息驱动(Message-driven)，SDI 和 MDI、资源(Resource)、DLL(Dynamic Linked Library)和 EXE 等，以便在纷杂的编程工作中不至于迷失方向；另外再介绍一下使用 Visual C++ 5.0 编程中的几个基本问题，如应用程序框架、类库 MFC、编程流程等，以希读者能从整体把握 Visual C++ 5.0 的精神。在随后的章节中，读者逐步了解使用 VC++ 5.0 编程的各个细节、如怎样建立一个新程序、如何处理消息、如何处理对话框等等。随着内容的递进，读者驾驭 VC++ 5.0 的能力、Windows 实战编程的水平将大大提高。

1.1　Windows 程序一般是有窗口的

一般情况下，Windows 程序都有窗口(包括对话框)。窗口首先接收来自用户的消息，再把消息分类处理，传递给其它对象，最后完成用户的要求。

对编程者而言，窗口管理也是不可忽视的一面。VC++ 5.0 类库(MFC)中对窗口也有强大的功能描述。

1.2　Windows 程序是由消息驱动(Message-driven)的

与 DOS 程序不同，Windows 应用程序大都是由消息驱动的。所谓消息，是指事件之间相互传送的具有一定意义的信号。如在 Windows 应用程序中，用户的信息和操作目的一般是通过消息传递给最终执行者，以完成相应的动作。消息的发送、传递、接收和处理是 Windows 程序的一个重要方面。实质上，Windows 程序员总是在与各种消息打交道。例如，我们想通过鼠标在 Windows 的 PaintBrush 窗口中画一条直线，首先单击工具箱中的画线工具，工具接收到单击鼠标的消息，工具条经过处理消息，允许用户画线；用户在窗口中按下鼠标左健，发出 WM_LBUTTONDOWN 消息；窗口处理消息后，记下鼠标按下时的位置，

并把该位置作为画线的起点；在窗口内拖动鼠标，有消息 WM_MOUSEMOVE 发送给窗口；放开鼠标，有消息 WM_LBUTTONUP 发送给窗口，窗口记下鼠标放开时的位置，将它作为画线的终点；最后，连接起点和终点，画出直线。

消息的发送、传递、接收和处理都有严格的步骤，编程者必须对此多加注意；同时，消息的种种特征往往比较类似，在多数情况下是一致的。VC++ 5.0 通过类库(MFC)描述了大部分消息的一般特征，这样，编程者只需从类库中派生出自己的类，便可以很好地处理消息，使得编程工作能着重放在自己的应用方面。

1.3 资源(Resource)

在 Windows 程序中，往往有菜单、工具条、对话框、位图、图标、字符串等，这些元素均被称为资源。资源可以预先编辑，独立编译，最后与可执行模块链接在一起，形成 EXE 文件或 DLL 文件。VC++ 5.0 集成环境提供了很好的资源编辑工具，编译和链接工具也都非常融洽地与整个程序的编译、链接工具融为一体。针对对话框的特点，VC++ 5.0 还通过 ClassWizard 提供了优秀的对话框资源的编辑接口，使得对话框的编辑和编程一气呵成。

在此需提请读者注意的是，编程过程中常常要用到字符串(如错误信息字符串)，应尽可能将字符串以字符串资源的方式存在，以方便程序的本地化(如汉化、日文化、韩文化等)。

1.4 SDI 和 MDI

Windows 应用程序是有窗口的。有的程序只允许打开一个文档，进行查看、编辑等操作，这种程序称为 SDI(Single Document Interface)应用程序，如 Windows 中的 NotePad 工具，在一个应用程序实例中只能打开一个 Text 文件，若再打开其它文件，当前文件即被关闭。另有一类应用程序，允许同时打开多个文档，甚至多个不同类型的文档，这种程序称为 MDI(Multiple Document Interface)应用程序。如 VC++ 5.0 开发环境本身就是一个很好的 MDI 的例子。在 VC++ 5.0 中，可以同时打开多个 Text 文件(如.h, .cpp 文件)、资源文件(如对话框、菜单、图标资源等)。在 MDI 应用程序中打开其它文档时，原文档不受影响。

在 VC++ 5.0 环境下编程时，应根据具体情况决定编制 SDI 或 MDI 应用程序。一般说来，如果处理的数据类型单一，操作简单，可以考虑使用 SDI，反之，则可采用 MDI，尤其是当数据类型复杂，需要分别处理数据(如文档中既有文本文字，又有图形，需要分别编辑)时，应考虑使用多种文档类型的 MDI。

SDI 和 MDI 除了上面所说的不同外，程序其它方面，如消息处理、文档和视的关系等各方面是相同的，在编程时并没有太大的区别。

1.5 DLL 和 EXE

在 Windows 程序中，有两类可执行程序模块文件：EXE 和 DLL。EXE 文件是独立可执行程序，DLL 是特殊形式的可执行模块，它被 EXE 模块或其它 DLL 模块调用执行。与 EXE 相比，DLL 具有运行开销较少，可被多种程序调用，可同时被多个程序调用等优点。

DLL 还是程序模块化的一个重要手段。一般来讲，如果一个开发小组要开发一个大型应用程序，先把应用程序分成若干功能模块，各小组成员独立开发自己的模块，将这些模块生成 DLL；开发管理人员开发出一个 EXE 来调用各个 DLL，完成整个应用程序的开发。可以看出，由于小组成员独立开发生成 DLL，他们之间的交叉依赖关系大大减少，“代码合并”等琐碎、繁杂又容易出错的工作量会大大降低，应用程序的维护也易于进行。

VC++ 5.0 不但可以生成正常的 EXE 应用程序，还可生成三种形式的 DLL。这使得 EXE 和 DLL 的开发都很容易，而且对 DLL 的优化使得 DLL 文件大小和调用开销都显著降低。三种形式的 DLL 可以兼容以前的 DLL 形式。

1.6 OLE 接口

OLE(对象的链接和嵌入)是 Windows 平台的一个重要特性。所谓对象，即指各种应用程序操作的数据对象，如 Windows 附件 PaintBrush 中处理的图、WordPad 中处理的文字。Windows 定义了一个公共接口，使得在一类应用程序中可以处理其它类程序的操作对象。如在 WordPad 中，用户可以激活 PaintBrush，在 PaintBrush 中创建或编辑的图在 WordPad 中显示。当要编辑该图时，PaintBrush 都被激活。在 WordPad 中看，图与文字一样，也是 WordPad 的处理对象。

OLE 是一种允许用户创建和编辑文档内含对象(即其它应用程序的对象)的机制。以前，OLE 是 Object Linking and Embedding 的缩写，表示对象的链接和嵌入，现在，它就被称为 OLE，OLE 中链接和嵌入以外的成分成为 ActiveX 的一部分。

OLE 文档(OLE document)，曾被称为 compound document，可将多种类型的数据(或称部件)无缝整合，声音、报表、位图等均可成为 OLE 文档中的部件。在应用程序中支持 OLE 可允许用户利用 OLE 文档处理多种数据而不需在多个应用程序中来回切换：OLE 承担了这一切。

OLE 结合了若干实现应用程序间无缝接口的关键概念，其中包括：

1. 链接和嵌入(Linking and Embedding)

链接和嵌入是两种在 OLE 文档中保存其它应用程序创建的数据对象的方法。

在 container 应用程序中激活“Paste”命令时，可以创建一个嵌入的对象。对象的源数据被保存在 OLE 文档(container 应用程序的文档)中。使用这种方式，一个文字处理应用程序的文档中可能不只包含文字，还可能包含位图、图形、公式等其它数据形式。

OLE 还提供了另外一种方法，包含来自其它应用程序中的数据，即创建链接对象(linked component, 或称 linked item, 或称 link)。创建链接对象的方法与创建嵌入对象的方法类似，只是，需用应用程序的命令“Paste Link”而非命令“Paste”。链接对象只保存指向原始数据(一般是一个独立的文件)的路径。

所有 OLE 对象，不管是嵌入对象，还是链接对象，都有一个与之相联系的基于创建它的应用程序的类型。如，一个 Microsoft 的 Paintbrush 对象有一个类型，一个 Microsoft 的 Excel 对象有另一个类型。有的应用程序能创建两种以上的对象类型，如 Microsoft 的 Excel 可创建三种类型的对象(worksheet, chart, macrosheet)，那么，每种对象可被系统根据类识别码(Class Identifier，即 CLSID)唯一识别。

2. Container 和 Server

创建 container 应用程序可以建立 OLE 文档，以包容其它应用程序的数据对象；创建 server(或称为 component)应用程序可提供被其它应用程序使用的数据对象；也可以创建既是 container，又是 server 的应用程序。

Container 应用程序是一种能够在自身文档中包含其它应用程序创建的嵌入对象和链接对象的应用程序。Container 应用程序必须能够存储和显示包含的 OLE 对象，允许用户在必要时激活 Server 应用程序插入新的 OLE 对象、编辑已有的 OLE 对象。

Server 应用程序(或称 Component)是能够创建 OLE 对象以被 Container 使用的应用程序。Server 应用程序通常支持拖放功能(drag and drop)和拷贝数据到裁剪板的功能，以便 Container 应用程序可以插入 OLE 对象。有的应用程序即是 Container，又是 Server。

大多数 Server 应用程序是独立的应用程序(即 full-server)，它们可独立运行，或由 Container 应用程序激活而运行。Mini-server 是一种特殊的 Server 应用程序，它们只能被 Container 激活而运行。Microsoft 的 Draw 和 Graph 即是 Mini-server 应用程序。

Container 和 Server 并不能直接通讯，它们通过 OLE 系统 DLL。这些 DLL 提供函数，供 Container 和 Server 调用，同时 Container 和 Server 提供回调函数，供系统 DLL 调用。

使用这种通讯方式，Container 不需知道 Server 的实现细节。它允许 Container 接受任意 Server 创建的 OLE 对象，而不必预先定义将与之联系的 Server。因此，Container 用户能够得益于未来的应用程序和数据形式，只要新应用程序能够创建 OLE 对象。

3. In-Place Activation (Visual Editing)

在 OLE 文档中激活嵌入(embedded)的对象被称为 in-place activation 或 visual editing。Container 应用程序的界面会改变，以反映创建嵌入对象的应用程序(Component)的特点。链接的对象不需要这种激活方式，因为对象的实际数据被保存在一个独立于 OLE 文档的文件中。

链接和嵌入、In-Place Activation 体现了 OLE 可视编辑的主要特点。

4. 自动化(Automation)

自动化允许一个应用程序驱动另一个应用程序，前者被称为自动化客户(automation client)或自动化控制器(automation controller)，后者被称为自动化侍者(automation server 或 automation component)。

自动化在 OLE 和 ActiveX 中均有效。有时，可能会自动化某个基于 COM 的对象。

5. Compound File

Compound file 提供一种标准的文件格式，以简化 OLE 文档的结构。OLE 文档中的这种存储结构也被称为结构化存储(structured storage)。

6. Uniform Data Transfer(UDT)

UDT 是一套数据发送和接收机制的标准接口，它不管实际的数据传输方法。UDT 构成了拖放操作(drag and drop)的数据传输基础，UDT 目前还是已存在的 Windows 数据传输(如裁剪板、DDE)基础。

7. Drag and Drop

Drag and drop 是一种易于使用、操作直观、在不同应用程序之间、同一应用程序的不同窗口之间、或同一窗口中不同部分之间传输数据的技术。用户操作时，只需选择源数据，拖动该数据至目的处，再放开鼠标即可。Drag and drop 机制基于 UDT。

8. Component Object Model(COM)

COM 提供了 OLE 对象之间相互通讯的基础。MFC 中有关 OLE 的类可简化程序员处理 COM 的方法。

COM 也是 ActiveX 的成分，因为 COM 对象是 OLE 和 ActiveX 的共同基础。

OLE 的 API 是非常庞大而复杂的，直接调用它们完成 OLE 支持是十分浪费时间的。OLE 的 MFC 实现使得 OLE 应用程序的开发变得比较直观、简洁。当然 MFC 中的 OLE 实现并没有完全包容 API 中 OLE 的所有内容，有时还需调用 Win32 SDK 中的内容。

VC++ 5.0 对 OLE 接口的编程进行了优化，它已在类库 MFC 中加入了有关 OLE 的类(Server, Client)，编程者从这些类中可以派生出自己的类，以处理有关 OLE 的各种问题。总之，在 VC++ 5.0 中实现各种 OLE 功能已变得方便、快捷。

在 VC++ 5.0 中，可以直接创建支持 OLE 的应用程序，也可对现有的应用程序修改，以支持 OLE 功能。

1.7 数据库接口 DAO 和 ODBC

许多应用程序需要将数据存放在数据库中，同时许多应用程序能从使用数据库中得益。数据库可提供灵活的数据存放方式，还可被多个用户和应用程序访问，并具有容量大、查询和更新速度快的优势。

MFC 支持两种形式的数据库访问功能：

- 通过 Data Access Objects (DAO)和 Microsoft Jet 数据库驱动(database engine)；
- 通过 Open Database Connectivity (ODBC)和 ODBC 数据库驱动(driver)。

这两种方式均简化了数据库访问的方法，充分利用了 C++在速度、功能和灵活性方面的优势，并将数据库访问和 MFC 应用框架结合为一体。

DAO 和 ODBC 是两种编程接口，可使编程者独立于某种特定的数据库管理系统(database management system, 即 DBMS)。

使用过 Microsoft Access Basic 或 Microsoft Visual Basic 的用户可能对 DAO 比较熟悉。DAO 使用 Microsoft Jet 数据库驱动(database engine)来提供一套数据库访问对象，即：database object, tabledef 和 querydef object, recordset object 等。DAO 与 Microsoft Access 创建的.MDB 文件合作最为有效，但也可以使用 DAO 和 Microsoft Jet 数据库驱动(database engine)来访问 ODBC 数据。

ODBC 提供一个 API，不同的数据库厂商可通过特定的 ODBC 驱动实现特定的 DBMS。应用程序使用 API 调用 ODBC Driver Manager，并通过 ODBC Driver Manager 调用适当的驱动，而该驱动使用 SQL(Structured Query Language)与 DBMS 交互。

作为 WOSA(Microsoft Windows Open Standards Architecture)的一个主要部分，ODBC 已存在相当长的时间。DAO 是根据 Microsoft Jet 数据库驱动(database engine)优化而来的，我们可利用它访问 ODBC 和其它外部数据库，建立在它之上的独立的 ODBC API 和 MFC 类仍可使用，并作为一种访问数据库的工具。

MFC 实现 DAO 和 ODBC 的内部细节颇为不同，但它们在界面上非常类似，这使得二者的相互移植比较方便。

MFC 编程模式为每个开放的数据库提供一个数据库对象，该对象代表程序与数据库的联系。程序通过 recordset 对象查询或更新数据库。DAO 还提供了另外的对象，以便与表结