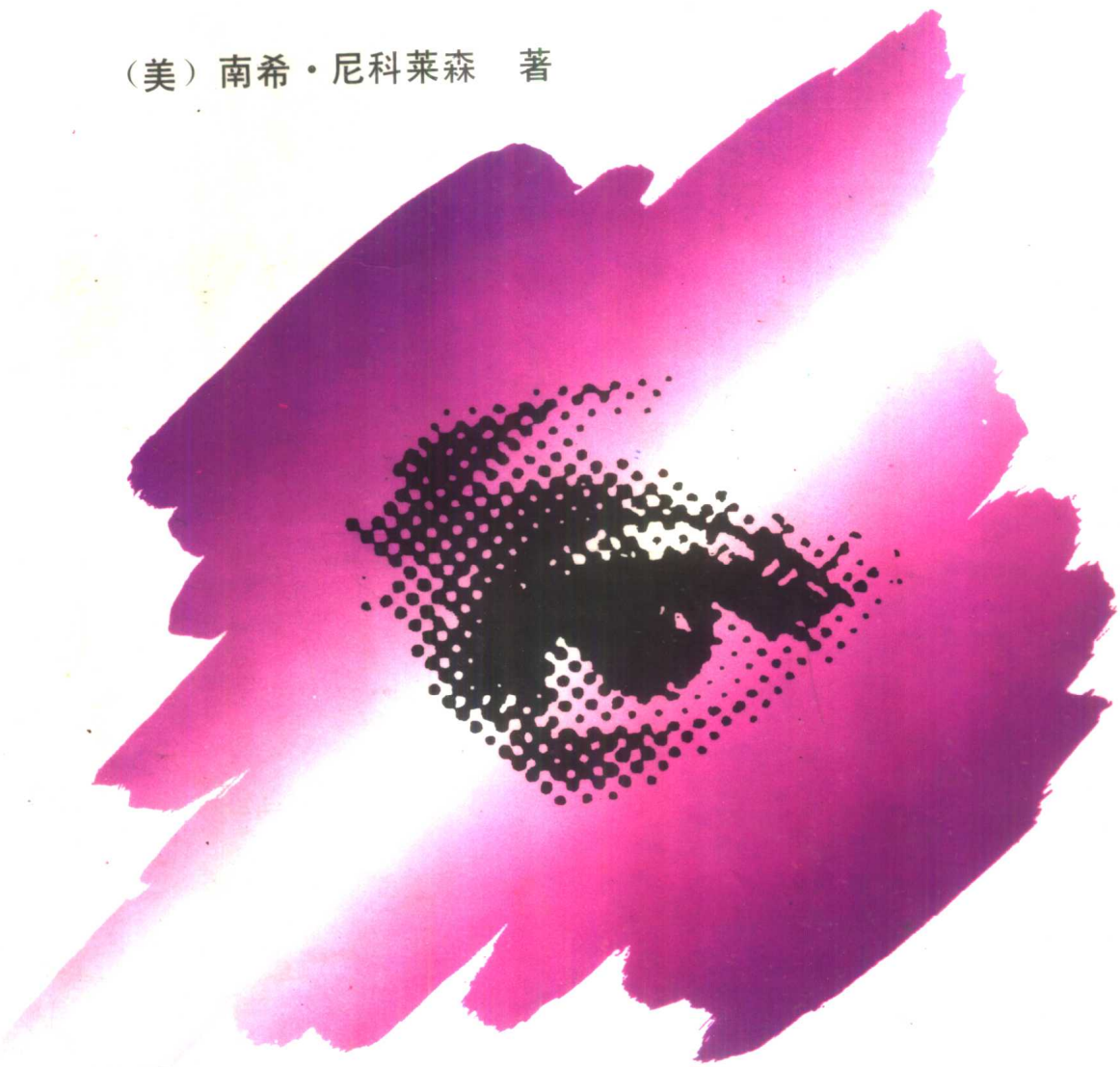
Visual C++ 2.0
可视化使用指南

（美）南希·尼科莱森　著



机械工业出版社

Visual C++2.0 可视化使用指南

（美） 南希·尼科莱森 著

焦宗夏 方永宏 袁 梅 译

机 械 工 业 出 版 社

内 容 简 介

本书共分4章，第1章为使用AppWizard的方法，介绍了AppWizard所能完成的基本功能，在应用类型及其特性的选择，数据库支持（ODBC），对象的连接与嵌入（OLE）等方面有较强的支持。可以根据用户的选择自动生成应用的架构代码，在整个生成过程中始终是可视的，在交互操作的同时便可看到所产生的效果；第2章为集成开发环境。详细地介绍了各种资源编辑器的使用方法，以及如何使用ClassWizard进行类的创建，这是Windows应用界面快速生成的有效方法；第3章为创建Windows应用指南，详细地讲解了各种Windows界面的创建过程并给出了相应的范例；第4章为MFC基类库，给出了MFC3.0版的所有类的详细说明及函数原型和使用范例。

本书深入浅出，既是初学者的很好的入门教程，又是编程人员必不可少的需要随时翻阅的参考书。

*　　*　　*

图书在版编目（CIP）数据

Visual C++2.0可视化使用指南＝THE VISUAL GUIDE TO VISUAL C++/（美）南希·尼科莱森著；焦宗夏等译．－北京：机械工业出版社，1995.12

ISBN 7-111-04861-X

Ⅰ.V… Ⅱ.①尼… ②焦… Ⅲ.C语言-程序设计-指南 Ⅳ.TP312-62

中国版本图书馆CIP数据核字（95）第14268号

出 版 人：马九荣（北京市百万庄南街1号　邮政编码100037）

责任编辑：温莉芳等　版式设计：冉晓华　责任校对：刘志文

封面设计：姚　毅　　责任印制：卢子祥

三河永和印刷有限公司印刷 · 新华书店北京发行所发行

1995年12月第1版第1次印刷

787mm×1092mm 1/16 · 38.75印张 · 2插页 · 952千字

0 001—4 000册

定价：72.00元

译 者 序

Visual C++的出现给使用C语言编程带来了革命性的变化。其最大的特点就是可视化编程，所见即所得，通过集成环境直接构筑程序界面构架和用户定制的新类。而Visual C++2.0则将这种新的特点推向了更加完美的境地，AppWizard、ClassWizard、AppStudio等工具更加紧密地结合，其衔接是如此的天一无缝，它们总是在您最需要的时候出现。这对一个想在Windows下编程而苦于难以下手的用户来说一定会喜出往外，因为整个编程环境提供了构造Windows界面的强大功能，用户只需将精力集中在与自身专业有关的编程方面即可。这种可视化的编程方法在以往的程序设计中是没有的，这不能不说是一个里程碑式的变革。

MFC3.0类库是Visual C++各种功能实现的基础，利用它几乎可以实现我们在Windows上可以看到的各种技术。另外一个很重要的功能是Visual C++的文档——视结构，文档与视分离使应用程序中的对象直接与文档结合，用户无需关心数据在文档中的存储形式，而直接对数据结构进行操作即可；另一个特别就是视，可以用不同的视结构实现同一文档的不同形式的显示。

本书的编写特点就是本着可视化编程这样一个概念，沿着程序构建的思路进行可视化讲解，浅显易懂，另外所提供的MFC3.0类库参考包含有类的原型及其使用范例，可以非常方便地查阅和引用。

由于本书的篇幅较大，另外还有以下几位同志参加了本书初稿的翻译工作：北京航空航天大学的洪冠跃、杨波、郝红卫、王萍、高晓宇及北京财贸管理干部学院的李大军，在此表示感谢，全书由焦宗夏和方永宏审校。

由于时间仓促，水平有限，不当之处请予斧正。

译 者

1995年10月于北京

目　录

1 AppWizard 的使用方法

什么是 Wizard？在 Microsoft 出版的《Computer Dictionary》中，Wizard 一词被定义为“在操作计算机方面有奇才的人（深通计算机技巧的人）”。每个人都有其自己的专长领域，并集中精力努力工作，但同时又要花费大量时间用于与专业领域无关的一些方面的工作。那么为什么不让一个好的软件来帮助您，帮您写出更完美、更高效的软件呢？

当您编程时，Wizard 几乎可以毫不费力地构造代码和与用户界面的标准结构。当系统环境和应用越来越复杂时，这就显得更加重要了。AppWizard 确定的结构标准使程序更加合理，正如前面所提到的，是符合广大用户的愿望的，也必将给程序员们带来巨大的利益；而且，这种自动创建软件的方法不受任何限制。同时，我们还将看到在 Visual C++中 Wizard 和应用框架（apllication frameworks）将提供给我们任何修改程序的灵活手段。

1.1 AppWizard 能为您做些什么？

“好的开始等于成功的一半”。此话听起来有点儿陈旧，但却是一个事实。用 AppWizard 可进行初始化并创建程序工程的内核——您所需要的类、对象和程序，它甚至在构架代码中提供注释，以说明可以在哪里写您的代码。

关于 AppWizard，最有趣的是在哪儿找到它，因为它不出现在任何菜单或工具条里，然而，什么时候需要它，它就会什么时候出现。打开一个新的文件，从文件类型列表里选择 project 类型，它就出现了。（在电影《The Hobbit》中，小矮人告诉 Bilbo，男巫 Wizard 将由着自己的意愿来去。幸运的是，在 Visual C++中，它们似乎总是在最需要的时候出现。）

AppWizard 将根据您的项目的复杂程度弹出 4～6 个对话框供您选择，一但知道您要干什么，AppWizard 将为您创建初始代码和支持文件。在下面几页里，我们将一步步地讲解对话框的每一段。如果它非常明白易懂，就浏览一下；如果出现问题，一个直观的 AppWizard 对话框方法将帮您容易找到问题的答案。

1.2 AppWizard 的启动

起动一新的项目（project）：

1）在 File 菜单上，选择 New。

2）从图 1-1 中显示的 New 对话框中，选择 Project。AppWizard 将显示在 New Project 对话框中，如图 1-2 所示。

3）在 Project Name 文本框中键入一个名字，这个名字将做为该项目中所有由

AppWizard 产生的文件及类的一个基础。名字的转换规则将取决于您在磁盘上所建立的文件管理系统：FAT，NTFS 或 HPFS。

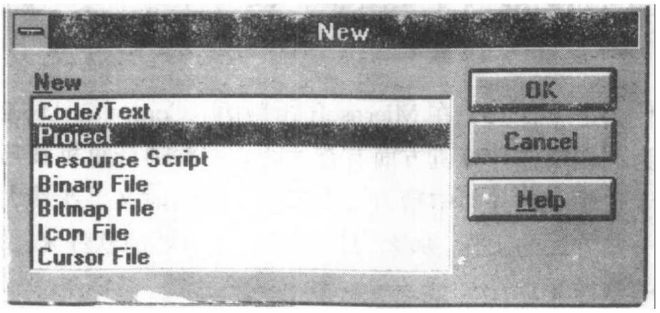


图 1-1　New 对话框

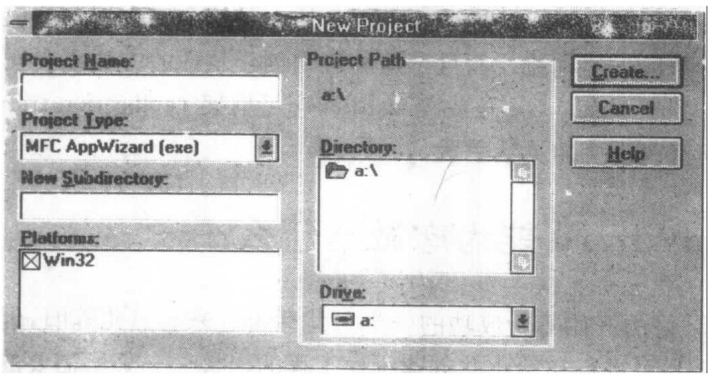


图 1-2　The New Project 对话框

文件系统

FAT（File Allocation Table）：这是一个 MS-DOS 系统并受传统的 DOS 约束。

- 名字可 8 个字母长并可有选择地带 3 个字符的扩展名，中间以“.”分开。
- 字母的大小写没有区别。
- 可以用从 A～Z，从 0～9 以及以下的特殊字符：

  — ^ $ ～ ! # % & — ( ) { } @ ’ ‘

NTFS（New Technology File System）：当您安装 Windows NT 时，您可以选 NTFS 选项。它将给您很大的灵活性。

- 名字可长达 255 个字符。
- “.”和空格可以以您认为合适的方式插入，只要总长度小于 255 字符。
- 名字不分大小写，尽管屏幕显示上有大小写，但名字本身并不区分大小写（警告：filename. txt 将覆盖 FILENAME. TXT。）
- 可以使用 A～Z 及 0～9 之间的字符。
- 可以使用除了？\ * " 〈〉 | /：以外的任意特殊字符。

NTFS 也建立一个 FAT 文件名，以使其可以运行于 MS-DOS 环境下。这些名字的转换是通过如下方法实现的：去掉空格及句点（除了有字符跟随的最后一个句点）；转换不允许的字符为下横线(_)；截断其名字为6个字母，再加一个“～”和

数字。最后，截取扩展名为 3 字符。

HPFS（High Performance File System）：这是 OS/2 文件系统，其规则与 NTFS 相同。

4）按动 Project Type 下拉式列表框中的下拉按钮，做如下选择（参见图 1-3）：

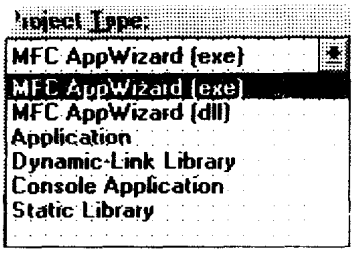


图 1-3 Project Type 下拉式列表框

（注：在此处，您将决定是否使用 AppWizard。如果您选中了头两个选项：MFC AppWizard（exe）或 MFC AppWizard(dll)，正如名字所暗示的那样，AppWizard 将启动。其余 4 个选项允许您不用 AppWizard 建立工程。为保证完整性，我们将它们列在这里，但以后将不再继续讨论它们。）

- MFC AppWizard(exe) AppWizard 询问您将选择什么样的特性，随后产生符合您所希望的代码并将其放入适合于编辑和连接的文件里，最终将完成 Windows 应用的基本构架，其中包括工作窗口、菜单、工具条等。这里有大量的标准功能：下拉式菜单，响应 File 的公共对话框、New 命令等。这一完整的功能框架允许您直接进入自己的应用领域。如果您愿意，AppWizard 将包含有注释，说明在哪里编制您自己的代码。
- MFC AppWizard（dll） 选择此命令，AppWizard 使您起动动态连接库。这是一个在运行时才连入的类库。使用动态连接库的优势是可以使几个应用同时共享同一段内存驻留程序，如果该类被多于一个的应用所使用，或被一个应用的几个实例所使用，将节省大量内存。
- Application 假设您有自己的类库并且不使用 MFC 库，选择该项可以建立一个独立的应用。
- Dynamic-Link Library 这是一个共享库，但它不依赖于 MFC。
- Console Application 建立运行于 MS-DOS 或 Windows 的 DOS 窗口中的字符型的应用。
- Static Library 静态库是在 build 时被连接的，属于非动态的。因此，每个应用在运行时必须有自己的拷贝。

5）当您键入项目名（project name）后，基于此名的子目录被建立起来了。并且，该名自动显示在 New Subdirectory 文本框中，参见图 1-4。只有当您已经有了该子目录下的源文件时，您才可以改变这个名字。若要指定不同的子目录，在 New Swbdirectory 文本框中键入名字和路径，或用 Directory 和 Drive 框来转换磁盘目录结构。

6）接下来，选择您的计算机环境。在平台的选择框中选择机型和操作环境。Win32，指的是 32 位 Windows 环境。若在其它平台上，如 Macintosh，您将需要一个交叉平台开发工具。

7）最后，按下 Creat 按钮可以起动 AppWizard。这里有两种可能性：一是您正在创建（building）一个 MFC AppWinzard 可执行应用；二是创建一个 MFC AppWizard 动态连接库。由于后者很简单，我们首先来叙述它。

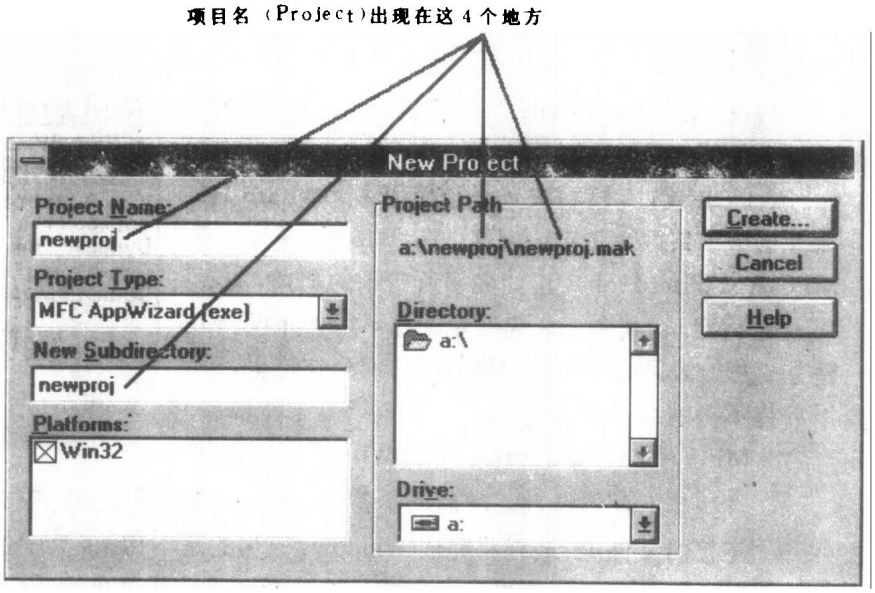


图 1-4 New Project 对话框，四处出现工程名

1.3 建立 MFC AppWizard dll

您可从用 AppWizard 按照如下步骤建立一个动态连接库（dll）

1）AppWizard 询问是否包含注释用于说明您将在哪里插入自己的程序代码。点

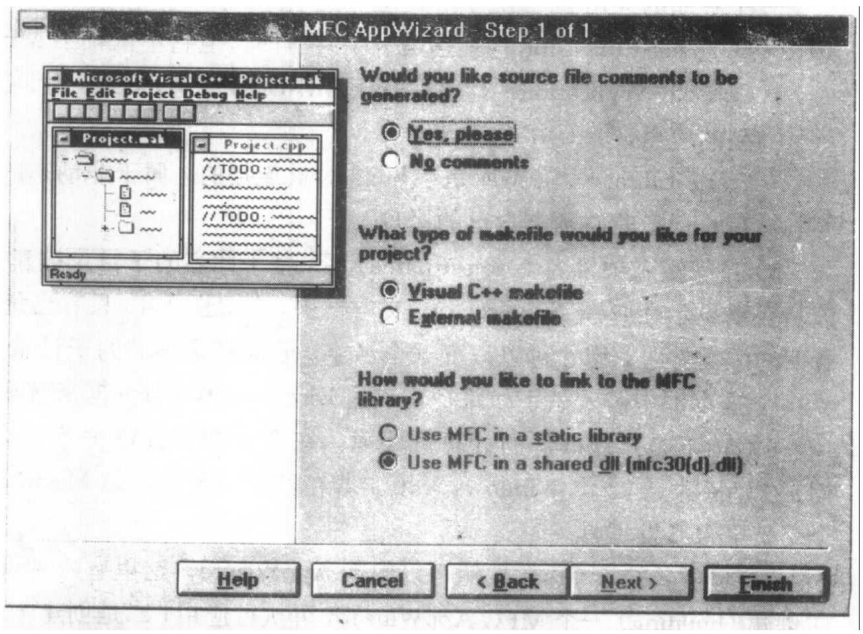


图 1-5 在此对话框中选择选项，以生成 MFC AppWizard dll

取 Yes, please 按钮。AppWizard 产生了一个 README. TXT 文件作为注释的一个补充来描述所生成的文件。参见图 1-5。

2）选择按钮，以表示您是想生成 Visual C++ makefile 还是外部的 makefile。

以上是如何使用 MFC AppWizard dll 的全部过程简介，这样您可以显示所有得到的文件。

1.4 建立 MFC AppWizard exe

在建立一个可执行文件（exe）之前，AppWizard 需要了解您的想法。它提供了几个对话框用来询问有关您的应用，其工作方式是由整体到局部，这是一个很好的自上而下的思维方式。

1.4.1 选择应用类型和语言支持——Step 1

AppWizard 首先询问您想建立什么类型的应用。您有 3 个互锁（单选）按钮可以选择（参见图 1-6）：单文档（Single Document），多文档（Multiple Document）或基于对话框（Dialog-based）。左边的应用窗口将如实反应您的每个选择结果。

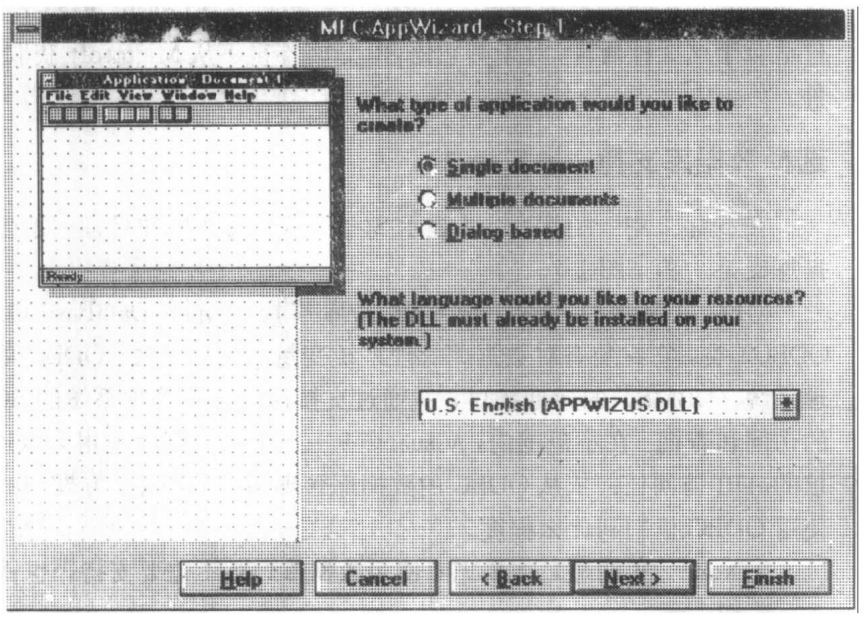


图 1-6 Step 1 对话框

文档

在 MFC 世界中，一个核心的组织原则是文档/视（Document/View）结构概念。文档对象用来存储应用数据，它也可能封装一些特殊数据类型，如电子表格（spreadsheet）或位图（bitmap），或是反映应用运行状态的数据。关于文档的一个非常重要的概念是它是一个分开的对象，尽管它是通过公共数据成员或公共成员函数来提供

数据给其它对象的，但对用户来说，它却是与显示它的应用部分（视）相分离的。

视（View）是应用的一个组成部分，可以将数据画在屏幕上（以图形形式）或打印在纸上，与文档进行交互，获得数据并进行显示。一个文档可以有多种“视”形式，如电子表格可以有数表、曲线图和饼图几种形式。与之相反，一种“视”一次只能与一个文档相关。

- 单文档（Single document） 单文档应用一次只有一个激活的文档。一个 SDI 应用可以通过断开或连接而在各文档之间进行切换，此过程可以不断重复，但一次只能有一个文档，而一个文档同时则可有多个激活的“视”。
- 多文档（Multiple document） 一个多文档应用可以有多个文档。例如，它可以有一个封装字处理文本的文档，同时也可以有一个封装实况电视输入的文档。在每一个单文档应用情况下，每个文档同时可以有多个“视”被激活。
- 基于对话框（Dialog-based）：Dialog-Based 应用通常用于收集最终用户的格式化的键盘输入数据。选择此选项自动建立对话框数据交换和对话框数据确认（Dialog Data Exchange and Dialog Data Validation，简称 DDX/DDV）支持代码。

Step1 对话框询问您的资源中使用什么语言。如果您的应用是面向国外市场的，这正是可以做出选择的地方。标准的选择是 U. S. English，French 或 German。dll 文件适合于这 3 种语言，对于其它语言，您将不得不从 Microsoft 获取适当的 dll 支持文件。

1.4.2 选择数据库支持——Step 2

在 Step 2 中，AppWizard 显示 Step 2 of 6 对话框并且询问您想用什么类型的数据库支持，参见图 1-7。

MFC 框架提供一个开放的数据库连接（Open DataBase Connectivity，简称 ODBC）类，它封装有对数据库管理系统变量的访问方法。ODBC 可以建立数据库管理系统（DBMS）的标准接口，并通过支持该数据库管理系统的 SQL 语言进行数据的访问和查询。在小范围内，Visual C++产生的数据库应用可以通过一致的、可移植的 API 访问任何依从 ODBC 的数据库管理系统的数据。方框中的选项可以决定您使用 ODBC 机制到什么程度。

- None 很明显，如果您不使用数据库，则不需要支持，也就不产生额外的代码。
- Only include header files（只有头文件） AppWizard 将提供适合于您创建自己数据库类的头文件和连接库。它不提供用于数据输入、记录组织、数据查看与存储等用途的类。
- A database view, without file support 这个选项提供用于数据输入、记录组织，数据查看的类，但不提供做为永久文件存于磁盘上的功能。如果您只需要一观测器，用来查看但不修改别人的数据，如查看股市行情信息，那么，这个选项是合适的。

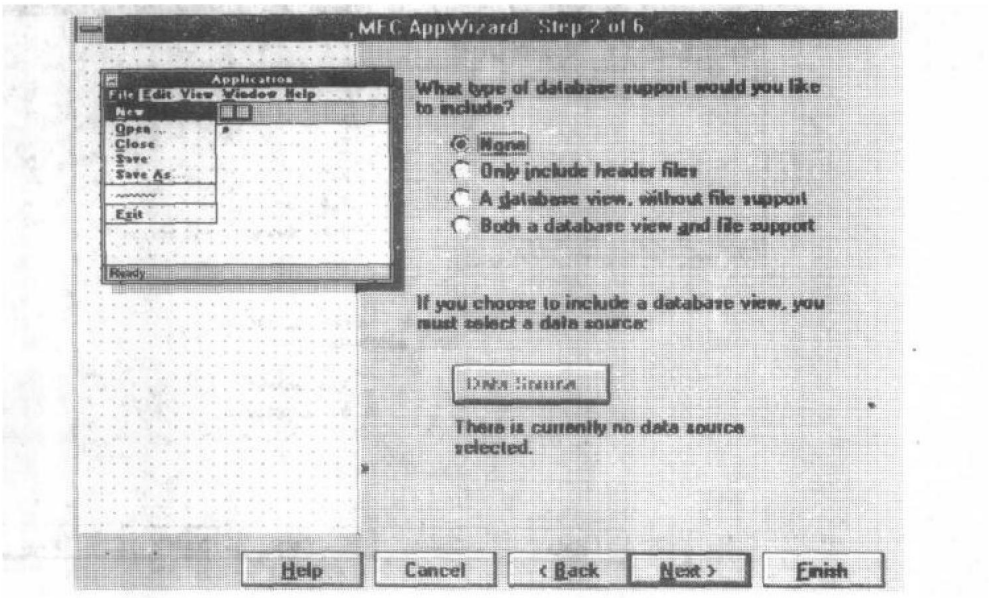


图 1-7 Step 2 of 6 对话框

• Both database view and file support 此选项提供数据输入、记录组织、数据查看和文件存储的全面的支持。

如果您不管选中后两项中的哪一项——包含有数据库视的项—Data Source 按钮将变暗，那您必须进入 SQL Data Source and Data Table 对话框，才能继续进行。

1.4.3 加入对象连接与嵌入（OLE）支持——Step 3

Step 3 将提供对象的连接与嵌入支持，见图 1-8。如果您是建立一个基于对话框的应用，则见不到此对话框。

对象连接与嵌入是目前 Windows 编程中最激动人心的，也是最复杂的技术。AppWizard 和 MFC 类使用户可以在较短时间内用 C 语言的 API 建立起强大而可靠的 OLE 应用。

因为 OLE 是一项复杂的技术，AppWizard 生成的 OLE 类支持，其结构可以使开发者逐渐完成其程序代码的编制。您可以从一般复杂的 OLE 功能开始，如拖放（drag and drop）支持，随着理解的深入，可以将更具权威性的 OLE 支持加进来。下列选项可供您选择。

• None AppWizard 不为 OLE 接口写任何代码。

• Container AppWizard 使您的应用接收嵌入数据，并且生成复合文档。换言之，它将能访问标准化的、由 OLE 服务且建立的数据对象。这里不支持数据发送，因此该应用不能成为服务程序（server）。

• Mini-Server 微型服务程序能够建立一个用于嵌入的对象，但不能做为文件存放。这当然禁止任何连接。这类服务程序可以在持有这类对象的包容器