

第0章 引言

这是一本组织严谨的 Access Basic 教程。Access Basic 作为一种强有力的程序设计语言，使最为畅销的 Microsoft Access 数据库管理系统如虎添翼。《Access Basic 教程》是激动人心的 Que 系列书籍之一，它通过简明指导、代码示例及快速提示等各种形式的标题设计，使读者能够快速获取知识。在很短的时间内，读者将学会编写代码，这些代码不仅能在 Access 中运行，而且只需很少的改动便能在用 Visual Basic 程序语言编写的应用程序中运行。

0.1 本书使用对象

对于具有程序设计语言——特别是 Basic 或 Visual Basic 使用经验的人来说，本书将帮助他们在其诸多的能力中再增加一种使用 Access Basic 的能力。Access 是新一代的开发工具，它能够使用 Windows 的图形用户界面(GUI)快速建立应用程序，并且能够设计出各种控制和窗口，而不必实际地编写代码。于是，便腾出了更多的时间让用户增加为数不多的程序执行代码。《Access Basic 教程》配合 Access 2 和 Windows 将会使用户快速掌握这一新的程序设计环境。

对于程序设计或者 Windows 编程的新手来说，本书仍是非常合适的。有关使用 Access 程序设计环境的循序渐进的步骤将使读者获得学习程序设计基本概念的基础。

尽管在使用本书时不必知道 Windows 程序的工作步骤，但对 Windows 界面应该熟悉。本书假定读者熟悉鼠标、窗口、下拉菜单、对话框及 Windows Help 系统的使用。

本书同时还假定读者对 Access 2 的环境(从用户的角度)已有所了解，并正准备使用 Access Basic。尽管许多主要的编程概念仍适用于较早的 Access Basic 版本，但是最好还是使用最新的版本，以便从本书中获取更大的收益。

0.2 语法一瞥

在本书中，有一种标有“语法”字样的框。在这些框中说明了 Access Basic 语言的重要成份，并简要描述了使用这些函数、语句、方法及事件的格式，同时还附有一到二段简短的示例代码。在 Access Basic 程序设计的进程中，读者可以不断地回头看看这些说明。语法一瞥框如下：

语法**Dim 语句**

Dim 有几种格式,最简单的是:

Dim 变量 1 As 类型 1[, 变量 2 As 类型 2][... 变量 x As 类型 y]

Dim 用于说明变量名为某种类型,并为之分配内存。

除语法一瞥框之外,还有其他三个显明的帮助格式——提示、注释和警告。

提示:提示为 Access Basic 程序设计提供了可能的捷径和线索,使得程序设计更容易和更高效。

注释:注释框提供的是附加的信息。这些信息有助于加速学习进程,并提醒读者经常要重复的重要信息。

警告:警告框用于警告问题的范围,包括损害计算机或者使程序不能运行的各种可能。

0.3 本书内容

本书分为十章和一个附录,每一章(及附录)都覆盖了一个程序员非常感兴趣的课题。

第一章说明 Access 实际上提供了许多种编程的方法,还介绍了宏、程序模块以及窗体代码。

第二章介绍了 Access Basic 程序设计环境。介绍了在进行实际的程序设计时 Access 所提供的一些机制。

第三章介绍了变量和运算符以及程序的构件块。

第四章的内容是如何使用变量和运算符来控制程序的执行。

第五章简要介绍了对象及其在程序中的使用(从中看出 Access Basic 可以真正地认为是面向对象的程序设计语言)。

第六章描述了如何使用记录集(数据库的局部或视图)。

第七章向 Access Basic 程序员介绍了一系列有效的函数,这些函数可以控制数据,还可对数据进行格式转换。通过阅读本章,可以获得足够的信息用于自己创建应用程序。

第八章讨论的是如何估计、防止、发现,并消除程序中的错误。同时还给出了错误处理例程和调试的有关概念。

第九章介绍了 Access Basic 程序设计的扩充技术,如 DDE 和 OLE。采用这些技术,可使自己的程序具有与其他应用程序,如 Excel 或 Microsoft Word 进行通信的能力。最后,在第十章中介绍了有关进一步提高 Access Basic 程序设计能力的内容,包括 Access 开发者工具包、宏转换及如何创建库等。附录 A 提供了一份便于使用的参考资料,其中列出了 Access Basic 中可用的属性和方法。详细内容见第五章和第六章。请不要迟疑,翻开第一页,开始精通 Access Basic 的愉快旅行吧。

第一章 Access 程序设计

当 Microsoft 推出 Access 的第一个版本时,成千上万的新用户便受益于这个功能强大且易于使用的数据库开发环境。今天 Access 又得到了进一步的发展,众多的工具简化了应用程序的开发,同时还增加了一项新的意义重大的性能——Access Basic 程序设计语言。

在考虑 Access 中的程序设计时,许多人只想到 Access Basic,其实,Access 提供了不止一种在数据库环境中进行程序设计的方法,或者使任务的实现自动化的方法。作为一个程序员,可以使用宏,也可以用 Access Basic 直接编程。另外,如果使用 Access Basic,还可创建自己的模块或者开发窗体代码(CBF)。本章中分别介绍了各种程序设计的方法。主要内容有:

- Access Basic 所用的特殊术语
- 宏的概念及宏的开发
- 如何执行宏
- 哪些工具是宏环境独有的
- Access Basic 的概念及如何编写 Access Basic 的过程
- 事件过程与全局过程有何不同
- 为什么说在有些情况下 Access Basic 更倾向于使用宏

1.1 概述

不论使用哪一种程序设计方法,要习惯 Access 中的程序设计,都有一个曲线式的学习过程。这个曲线是否合理,取决于用户的背景。如果熟悉其他应用程序中宏的开发,或者用其他版本的 Basic 编写过程序,那么可能很快就能适应 Access 程序设计环境。如果经验较少,那么这个曲线可能要陡峭些。

在使用 Access 时要确保有充足的时间,以便试试各种新技术,并让自己的想像力飞驰起来。从程序员的角度来看,这是一条真正学会 Access 并成为 Access 专家的道路。

像其他任何程序设计环境一样,Access 有它自己的术语和概念,明白了这些术语和概念,方能成功地使用它的环境。在使用宏或者 Access Basic 以前,最好弄清以下内容:

- **过程。**程序代码的一部分,用于完成特定的任务,它可以在应用程序的其他地方被调用。函数和子程序都是过程。
- **函数。**函数是一种当其执行完成后有返回值的过。当函数被调用时,典型的情况是它将出现在程序语句中等号的右边。
- **子程序。**子程序是一种当其执行完成后没有返回值的过。
- **模块。**模块是若干过程的集合体,并有一指定的名字。典型地,模块隶属于指定的窗体或者报表。
- **事件。**事件指触发过程的某一特定事件的发生。例如,不论何时当用户单击鼠标左键时,可以指示 Access 执行某一指定过程。单击鼠标按钮便是事件。

- **窗体。**用于采集或者显示信息的窗口。窗体包含一系列指定的对象。每个对象都有若干为之定义的事件。例如,按钮是一个对象,于是该按钮可能有若干已定义的事件,而其他对象则没有。窗体代码是隶属于窗体中对象的编码,作为整体,该代码是在与对象或者窗体相关联的指定事件发生时同时执行的。
- **报表。**报表是从应用程序中输出的打印或者显示内容。
- **属性。**属于某一特定对象,特别是与窗体相关联的对象的属性。属性定义了对对象在屏幕上的显示方式及其缺省情况。在设计窗体时,每个对象的属性都会显示在 Property(属性)窗口中。
- **排序。**指根据特定规则对数据库中的信息进行整理。
- **过滤。**指根据特定规则对数据库中的信息进行选择。
- **查询。**是一个被定义的数据库信息视图。查询本质上包括排序和过滤两个方面。

1.2 关于宏

宏是应用程序执行的一系列命令。宏语言的创建是为了帮助最终用户在不必学会,甚至不必理解编码基础的情况下设计其应用程序的。

注释: Access 中的宏编程细节不是本章或者本书的主要目的,这些内容是在 Access 中中级水平的资料中讲述的,但是明白在什么地方使用宏较为合适则是很重要的。另外,应该知道,许多用户已经熟悉了宏的编程,或许他们本来就希望跳过 Access Basic 程序设计。通过本章及第十章,读者将会对宏的使用有一个完整的认识。

例如,Microsoft Word 和 Excel 都有宏语言,它们能使一系列命令自动执行。同样,Access 也有宏语言。Access 有 47 种宏动作,每一种都有一类相关的变元,用以修改各种动作的工作方式。典型地,当宏被执行时,其执行过程就是用户手工使用某个程序的过程。

注释: 通过本章,可以学到有关动作和命令方面的内容。在大多数情况下,这两个术语是可互换的,从技术上讲,动作是应用程序执行某一指定行为的指令,例如从菜单中选定一选项。因此,动作最适合于宏语言。命令则超出了这层含义,因为它同时还是执行传统程序动作的指令,例如定义变量。

由于宏的作用类似于用户手工进行的动作,所以宏可用来使 Access 应用程序中的大部分重复工作自动进行,包括:

- 打开和关闭窗体和报表
- 使窗体和其他窗体与报表同步
- 设置窗体、报表以及控制的值和属性
- 执行单击按钮时的查找、过滤及查询操作
- 提高数据的合法性

■ 定制应用程序界面

1.2.1 开发宏

使用宏的时间越长,所学到的重要程序设计原则就越多。通常需要先确定要完成的任务,然后构造出完成这项任务必需的步骤轮廓。当这个轮廓被翻译成宏动作后,便形成宏程序代码。

当使用宏动作时,一定会发现大多数宏动作都需要变元。变元是简单的参数,用来影响动作的执行方式。如果手工执行这些动作(使用菜单选项),大多数变元与选项是一一对应的。

宏的创建有两种方式:

- 打开 Macro(宏)窗口,选定动作列表中的项
- 从 Database(数据库)窗口中拖动对象,然后放入动作栏中

Macro 窗口

要使用 Macro 窗口,必须先打开它。为此,可从 Database 窗口中单击 Macro 项。这时将出现一张用户数据库可用宏的列表。单击 New(新)按钮,Macro 窗口中将出现新的宏,即正在创建的宏,如图 1.1。

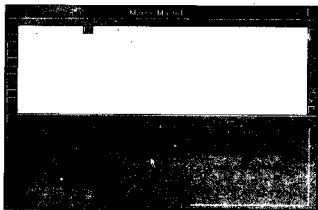


图 1.1 Macro 窗口

对宏动作的修改如同从下拉式列表中选择项目一样容易。单击 Action(动作)栏中某一单元右边的箭头,将出现一可用宏动作列表。选定动作后,则可修改该动作所用的变元。这些变元显示在 Macro 窗口底部。图 1.2 的例子表示 DoMenuItem 动作及其变元。

在其他时候,如果要选择动作或者指定变元,还可使用联机 Help(帮助)系统。按下 F1 键,将会出现上下文有关的帮助信息。如果需要更多的信息,它还能够提供有关的例子(Access 联机 Help 系统将在第二章中详细讨论)。

拖放宏

为了充分利用 Windows 环境的优越性,Access 允许用户在利用宏动作时使用鼠标。所需要做的仅是使用鼠标从 Database 窗口中拖动对象,并放置在某一动作栏中。当该动作放

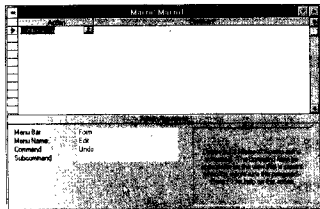


图 1.2 DoMenuItem 动作的可用变元

在动作栏中时, Access 对该动作使用缺省变元。缺省变元会基于正在使用的动作而准确地改变。

例如, 假设有一个数据库, 它包含一张名叫 Quote 的表。这时可以通过以下步骤, 自动创建一个使用该表的宏动作:

1. 确保 Macro 窗口和 Database 窗口都打开, 并同时可见。
2. 在 Database 窗口单击 Tables(表)。这时将显示用户数据库中可用的表。
3. 拖动 Quote 表至 Macro 窗口中 Action 栏中的某一单元。
4. 释放鼠标按钮。

这时, Access 假定要发生的动作是 OpenTable, 并自动地为该动作加入三个变元。图 1.3 显示了这时的屏幕布局。

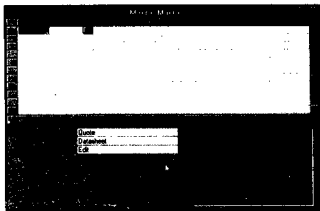


图 1.3 利用拖放技术, 创建宏动作

这时还可增加其他的动作或者修改变元。

注释:使用拖放的方法创建宏动作是有局限性的。其原因是这种方法仅当 Access 所假定的动作与用户真正要求宏所要完成的任务相匹配时才适用。因此,拖放的方法只能是偶尔使用。宏越复杂就越要直接来编写。

1.2.2 执行宏

宏的执行有这样几种方法:单击按钮、选择菜单项、按快捷键或者响应指定的事件。

单击一个按钮就使 Access 运行一个宏并不难。例如,假定要在一个窗体上创建一打开数据库表的按钮,只需完成以下步骤:

1. 完成前一部分中所描述的打开数据库表的宏动作的创建步骤。
2. 保存步骤 1 中创建的宏。
3. 确保 Database 窗口和 Form Design(窗体设计)窗口都打开,并同时可见。
4. 在 Database 窗口中,单击 Macros 项。这时会显示数据库中可用的宏,包括第 2 步中保存的宏。
5. 拖动新宏至 Form Design 窗口。
6. 释放鼠标按钮。

这时,Access 便在窗体中创建了一个按钮。这个按钮用宏的名字作为标签,在随后执行该窗体并单击该按钮时,宏便会被自动执行。不需要编写任何代码,甚至不需要画命令按钮,便实现了这样一些常用的程序设计任务。

执行宏(菜单项、快捷键或者使用事件)的方法几乎像“排队”一样容易,但这些内容超出了本章甚至本书的范围。重要的是要记住 Access 提供了一种快速方便的方法,让用户编制并运行宏。

1.2.3 使用其他宏工具

为了帮助用户开发更复杂更有效的宏,Access 提供了许多特殊的工具和性能。这些工具若能正确使用,则将发挥出在 Access 环境中用其他方法难以做到的更强大的性能和灵活性。

这些工具和性能有:

- 使用 Menu Builder(菜单生成器)
- 自动运行宏
- 开发自己的工具条

菜单生成器

随着 Access 窗体和报表变得越来越复杂,就会发现跟踪这些界面的工作方式也越来越困难。因为每个窗体上放置的空间非常有限。于是有些开发者利用转换板来进行设计,这样用户便可单击若干个连接按钮中的某一个,从而引出该应用不同的部分。依次地,这些子窗体或者报表中的每一个都可以有与其他屏幕连接的按钮。图 1.4 示出了一个转换板的例子。这是 Access 提供的 Northwind 示例中的一个转换板。

使用转换板的缺点是用户常常会在一堆迷宫般的窗体、按钮以及选项中迷失方向。另一



图 1.4 转换板设计示例

种很有效的办法是定制菜单。用户可以设计自己的菜单系统,增加或者取代 Access 本身的菜单。菜单能很容易地让用户同时看到多个选项,而且这些选项是有关联的,同时它所采用的是人们熟悉的、美观的 Windows 图形用户界面,使应用程序更显得专业化。

定制菜单最容易的方法是使用 Access 提供的菜单生成器。要启动菜单生成器,可在设计窗体或者报表的同时,选择 File(文件)菜单中的 Add-ins(增加)选项,这时会显示出一列可用的增加项,单击 Menu Builder 选项,菜单生成器便出现在屏幕上。

注释:如果 Add-ins 菜单中的 Menu Builder 选项无效,则说明第一次安装 Access 时没有安装它。因此要使用它,就必须重装 Access Add-ins。

菜单生成器允许用户对已有菜单进行重新设计,如增加新的命令以及取消不需要的命令,或者从头开始,建立新的菜单系统。这两种方法都是比较简单的,方法是使用菜单生成器对话框来选择、编辑,并保存菜单设计结果。这时所进行的是无宏程序设计,菜单生成器将把用户的各种选择转换为一系列特殊的宏。然后,当执行该窗体或报表时,这些宏便被执行。

AutoExec 宏

如果想要控制应用程序的启动方式,包括如何打开多窗体、指定定制的菜单以及初始化重要的设置,则必须用宏来完成。对此有两种方法:

- 启动 Access 时使用“/x 宏”命令行选项(“宏”是欲运行的宏的名字)。
- 使用特殊的 AutoExec 宏。

第一种方法允许用户在命令行上指定宏来执行他的设置动作。可以在 Properties(属性)对话框中修改,用户 Access 程序项的图标,这样双击该 Access 图标时就会执行该启动。

例如,假定在启动 Access 时,常常要运行一个叫 CompanySetup 的宏,这可以通过使用 Windows Program Manager(程序管理器)以命令行方式完成。简单的方法是在 Access 程序组中选择 Access 图标,然后选择 File 菜单中的 Properties 选项。这时将出现一个 Program Item Properties(程序项属性)对话框,如图 1.5。

所要做的仅是在 Command Line(命令行)对话框中修改命令。例如,若该框中有这样一个命令:

```
C:\WINAPPS\ACCESS2\MSACCESS.EXE
```

这时仅需做如下修改即可运行 Company Setup 宏:



图 1.5 Program Item Properties 对话框

C:\WINAPPS\ACCESS2\MSACCESS.EXE /x CompaySetup

AutoExec 宏则更易于实现。当数据库第一次打开时, Access 会找名为 AutoExec 的宏, 如果找到了便自动执行该宏。

提示: 如果不想在启动数据库时, 运行 AutoExec 宏, 只需在 Open Database (打开数据库) 对话框中选择数据库时简单地按下 Shift 键。

1.3 关于 Access Basic

细心的读者一定会想到, 既然宏如此强大, 为什么还必须学习 Access Basic 编程呢? 答案很简单, 不论宏有多么强大, 它们总会有局限性。尽管 Microsoft 所设计的 Access 宏技术能够完成大部分使应用程序自动化的工作, 但还有许多更高级的性能只有 Access Basic 才能提供。

初级时可使用 Access Basic 创建自己的过程。这些过程存放在下列两种类型之一的模块中:

- 事件过程(Event Procedures)是响应事件的代码段, 这些事件的发生和窗体的报表隶属的对象有关。
- 全局模块(Global Modules)是可以访问应用程序任何地方的代码段。

1.3.1 事件过程

Access Basic 提供了一种事件驱动的编程环境。这其实并不完全, 因为用它还能开发顺序执行的程序, 而不只是响应事件。

在 Access 中开发窗体和报表时, 很快便会发生窗体或者报表中的对象都有与其相关联的事件。例如, 如果要在报表中设计一按钮, 那么一个事件也许会通过用户单击鼠标而被引发。于是该事件发生时, 就可为之编写一运行过程。

为事件编写的过程仅当该事件与某一对象产生关系时才被使用。例如, 若一个窗体上有两个按钮, 则不会发生同样的过程, 而不论是哪一个按钮被单击。相反, 每一个对象都有它自己的事件组, 这些事件与其他事件, 即该窗体运行时, 可以发生的事件是相分离的。

1.3.2 全局模块

没有连接到特定事件上的过程就是所谓的全局模块。这些程序模块在单击 Database 窗

口中的 Modules 项时会出现。图 1.6 中显示了 Northwind 示例数据库(由 Access 提供)中有效的全局模块。

注意,图中只有一个模块。但是该模块能够(且确实)包含其他过程。模块名是一种能够方便地分开过程的简单方式;它是组织应用程序的一种方法。

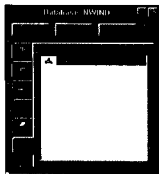


图 1.6 Northwind 数据库示例中的有效全局模块

任何给定模块中所包含的过程对整个应用程序都是有效的。这意味着它们能够在其他过程中被调用,能够分配给事件(前节中已做过描述),或者被宏调用。

为什么不顺理成章地把所有过程都作为全局过程呢?有两个原因:首先,开发仅在一处(如单击窗体上某些按钮)使用的全局过程是没有意义的。在这个例子中,如果代码作为一个事件过程来处理,程序会被更好地组织。第二个原因是全局模块在初始装载时要比事件过程花费更多的时间。这是因为事件过程仅当窗体或者报表显示时,才会被调入内存。另一方面,全局过程总是在内存中,并一直在消耗内存资源。基于这种原因,全局过程仅当绝对有理由时,才应使用。

1.3.3 不必永远使用宏

为什么要使用 Access Basic 而不用宏来完成某一特定任务呢?这个问题将在第十章中详细讨论,但在这里可以粗略地谈一下其原因:

- **错误处理。** Access Basic 允许用户预见并处理程序中不可避免的错误,宏则没有提供错误处理能力(关于错误处理的详细内容,参见第八章“错误处理与调试”)。
- **模块化。** 函数可用在不止一个程序或者不止一组环境中,宏则典型地只有一个设计目标。
- **文件编制。** Access Basic 模块能够被更容易、更迅速、更合法地文档化。这使维护变得更加容易,特别是当用户忘记了代码的用途时。
- **与其他程序的通信。** 由于 DDE 和 OLE 技术的应用,Access Basic 可以在 Windows 环境中与其他程序进行通信(在第五章“数据存取对象”中将对此作以介绍)。

但是,这仅是涉及到皮毛。随着对 Access Basic 的更加熟悉,用户无疑会发现 Access Basic 比宏更受欢迎的原因。

1.4 小 结

本章简要地介绍了在 Access 中进行程序设计的原由和背景。其中介绍了宏和 Access Basic——概念及使用方法。现在读者应能较好地理解 Access Basic 到底是什么,为什么要使用它。特别介绍了以下内容:

- 在 Access Basic 中进行程序设计时的特殊术语
- 什么是宏
- 如何开发宏(使用 Macro 窗口及拖放技术)
- 如何执行宏
- 宏环境中有哪些工具
- Access Basic 与宏有何区别
- 什么是事件过程
- 什么是全局过程
- 为什么在有些情况下 Access Basic 比宏更受欢迎

既然已经明白了 Access Basic 与宏之间的差别,就可以利用这些知识来开发自己的应用程序了。在下一章中,将介绍 Access Basic 编程环境中的一些规则。

第二章 Access Basic 编程环境

在第一章中介绍了两种 Access Basic 程序设计环境,即宏和 Access Basic。如果在数据库应用中使用了宏,则一定对方便的 Macro 窗口及其界面已非常熟悉。宏的创建与编辑比较容易,就如同从下拉式菜单中选择动作和变元一样,简单的调试工具允许用户跟踪宏中的单个动作,并且用户仅需几次鼠标单击即可把宏加到事件之中。

本章将引入 Access Basic 界面——Module(模块)窗口。要访问这个编辑器有许多种不同的途径,这依赖于所编写或编辑的函数或过程是存放在全局模块、窗体模块还是报表模块中。在这里将介绍如何利用 Immediate(跟踪)窗口检查代码,如何创建新的过程、编辑并运行它们以及如何从 Access Help 系统中获得帮助。

2.1 Module 窗口

在编写或编辑 Access Basic 代码时,将要用到 Module 窗口及其编辑器功能。已熟悉文本编辑器如 Windows Notepad 应用程序的用户,一定会发现 Module 编辑器更加友好和方便。Notepad 和 Access Basic 都是文本编辑器,但不像 Windows Write 或者 Word for Windows 是完全的字处理程序。它们之间重要的区别是像 Word for Windows 这样的应用程序所创建的文档包含格式化文本、字体、字体大小,及文本式样如粗体、斜体或者下划线加强体。而文本编辑器仅创建 ASCII 文本文件,并没有特殊的格式。

为了打开 Module 窗口创建或编辑一个全局模块,需从 Database 窗口开始。全局模块是这样一些数据库对象,即表、查询、窗体、报表及宏数据库对象。单击 Module 项,观察现成的模块,然后双击某一模块名(或者单击一次选定之,然后单击 Design 按钮)。如果要创建新模块,可单击 New 按钮。

注释:本书使用了一些 Access 2 提供的示例应用来作为 Access Basic 示例源代码。如果按典型的安装方法安装 Access,这些文件应在 Access 目录之下的 SAM-PAPPS 子目录中。

要想查看现成的全局模块,可打开 SAMPAPPS 目录中的 NWIND.MDB 数据库,单击 Modules 项,然后双击 Utility Functions(实用函数)模块名。Module 窗口的外观如图 2.1。

本图及任何全局模块的顶层部分是说明(Declarations)部分。在注释性描述正文后面,有两个语句。第一个 Option 语句以缺省形式,由程序加在任何一个新的模块中。否则,第二个语句会告诉 Access 任何新变量都是显式说明的,关于变量与常量的说明将在第三章中描述。

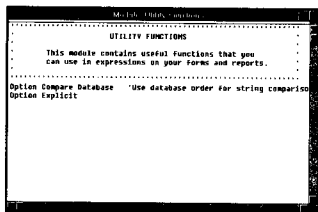


图 2.1 Module 窗口

2.1.1 Object 框

在屏幕顶部的工具条中,有几个活动按钮及两个列表框。用户可使用这些控制来访问全局模块中的其他过程。左列表框是 Object(对象)框,它允许用户在不同的窗体对象间移动。由于这是一个全局模块,而不是窗体或者报表模块,该框仅列出(常规)对象。

2.1.2 Procedure 框

Object 框的右边是 Procedure(过程)框,单击指向 Procedure 框右侧的箭头,将出现一张活动模块可用的 Utility Functions 列表——它们是为该模块定义的过程。这时可单击某一特定过程的名字以选定之,或者单击 Procedure 框右边的前一个及后一个过程按钮来移动过程。

由于 Access(及 Windows)众多功能,通常会有许多种方法来完成某一特定动作。除工具条外,还可以使用菜单或者快捷键来移动过程。即在 View 菜单中选定 Next Procedure(前一个过程)及 Next Procedure(后一个过程)项,或者按下 Ctrl+Up 箭头及 Ctrl+Down 箭头,快捷键命令紧挨着菜单项的名字,或者在菜单或菜单项的名字中有加强的热键。

还有一种移动过程的方法是使用 F2 键,它会引出 View Procedure 对话框。试着用一次,并双击列表下方的 IsLoaded 函数过程。

于是,IsLoaded 函数出现在 Module 窗口中,这时可以用鼠标及键盘来进行所有典型的文本处理操作,比如有许多组合式快捷键可以非常容易地进行查找和替换正文字串、插入空行、删除行、缩进或对齐单行或多行代码。

还可以在同一过程或者不同的过程之间,使用 Windows 鼠标或键盘技术,剪切、复制、粘贴代码。要在单个模块中观察两个过程,可以在 View 菜单中选定 Split Windows(分割窗口)项。这时 Module 窗口便被分为相同的两个部分,每一部分都有它自己的滚动条。顶部显示的是说明(Declarations)过程,底部显示的是前面已显示的过程(在这里为 IsLoaded 过程)。可以通过单击过程框的箭头以及单击下拉式列表中的另一个函数,来改变各个分窗口中显示的过程。

提示:如果在工具条的按钮或者框上,使鼠标指针暂停一秒,工具提示(ToolTip)消息将显示该工具的用途。

提示:要想显示 Module 窗口键盘命令的完整列表,可选定 Access Help 菜单中的 Search(查找)项,键入 Module Window,并单击 Show Topics(显示标题)按钮;然后在 Topic(标题)列表中双击 Module Window Keyboard Guide 标题(Access Basic 帮助系统将在本章的后面讨论)。

通过重选同样的菜单项,可以恢复分割开的窗口。另一种打开分割窗口的简单方法是找分割条(split bar)的位置,分割条是 Module 窗口顶部的一根水平细线——在函数代码之上,模块标题栏之下。当鼠标指针移到这个条上时,它会变成一个双向箭头,表示此时可拖动该线。这样做便可将窗口重新分为两个部分。要恢复为一个窗口时,可把分割条拖回到窗口顶部。

2.2 CBF 过程的创建与编辑

前面已介绍如何访问全局模块的 Module 窗口。另外还可以为窗体或报表创建局部过程,方法是使用 Access 2 的窗体代码(CBF——Code Behind Form)新技术。在创建一个窗体或者报表时,窗体或者报表模块会自动创建,这时事件过程被存放在该窗体或报表中,而不是像全局模块那样存放在独立的模块对象中。

以前介绍的所有关于同一 Module 窗口的编辑特性对事件过程都是适用的,而且 Access 还提供了另外几种工具以加速事件过程的开发。打开 CBF Module 窗口的最容易的办法是单击 Access 工具条上的 Code(代码)按钮。图 2.2 显示了 Database 窗口工具条中的 Code 按钮;单击它,首先打开 Design 视图中受选的窗体或报表,然后在前面打开其连通的 Module 窗口。如果该窗体或报表已在 Design 视图中打开,则只需单击 Code 代码便可打开 Module 窗口。同时还可使用 View、Code 项来完成同样的工作。

注释:读者可能已经注意到,在某些 NWIND.MDB 窗体或报表中并没有事件过程。在原来的 Access 1.x 版本中创建的 NWIND.MDB 数据库,不支持 CBF。因此,它的 Access Basic 代码示例被存放在实用函数(Utility Function)全局模块中。在另外两个示例数据库,ORDERS.MDB 和 SOLUTION.MDB 中还有许多 CBF 过程的例子,这将在本章的后面讨论。

在创建或编辑 CBF 过程前,应为 NWIND 数据库增加一新的窗体。用这种方法,如果在使用本书时想保存所做过的工作,则不必担心对 NWIND 数据库中窗体的修改。当然,也可以经常从 Access 安装盘中装载示例数据库。

要创建一新的窗体,需以下步骤:

1. 单击 Database 窗口中的 Form 项,然后单击 New 按钮,紧接着单击 Blank Form(空窗

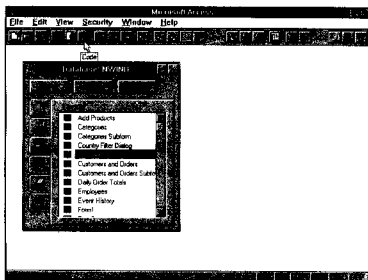


图 2.2 Database 窗口中的 Code 工具条按钮

体)。

2. 确保 Toolbox 工具条可见,Control Wizards 工具不被选中。
 3. 单击 Toolbox 中的命令按钮图标,然后再次单击,从而把大小缺省的命令按钮 Button0 放到新的 Form1 窗体中。
 4. 确保新按钮被选,然后选定 Properties(属性)窗口(如果需要,可单击 Access 工具条上的 Properties 按钮),并使用 Properties 窗口顶部的下拉式箭头,选择 All Properties 列表项。
 5. 选定 Name(名字)属性正文框中的 Button0 正文(如果它还没有受选的话),键入 Open Form,然后按 Enter 键。
 6. 在 Caption(标题)属性正文框中键入 Open Form(确保在两个单词间留一空格,这与前面 Name 属性设置中的正文不一样)。
 7. 单击工具条中的 Properties 按钮,关闭 Properties 窗口。
- 这样就可以开始编写事件过程了。

2.2.1 事件的建立及代码生成器

这里将使用 Access 2 新的快捷菜单的能力来进入新窗体的 Module 编辑器。为此,需进行以下步骤:

1. 由于将要创建的代码是重命名的 Open Form 命令按钮的代码,所以先右击命令按钮的显示快捷菜单。
2. 选择 Build Event... (建立事件...) 菜单项,用以打开 Choose Builder(选择生成器)对话框。
3. 选定该列表中的 Code Builder(代码生成器)选项,然后单击 OK,或者双击 Code Builder 选项。

这时模块窗口便形成了,其标题栏反映了当前的 Module 编辑器正是 Form.Form1 模块的模块编辑器。屏幕的外观应类似于图 2.3。

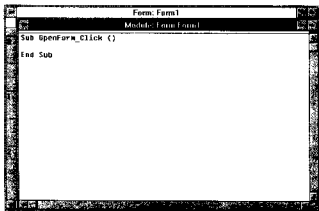


图 2.3 Form.Form1 的模块窗口

读者也许会注意到一些有趣的事情,即代码生成器自动地产生了代码模板。因为曾右击过命令按钮,所以代码生成器便为 OpenForm 按钮的缺省事件过程创建了代码模板,令人吃惊的是这个过程并不是 Click 事件过程。代码生成器用一下划线(_)把 Name 属性正文框中键入的命令按钮名与 Click 事件名连结在一起。窗体或报表所隶属的 Access 事件过程是子过程,而不是函数,所以模板被括在 Sub 和 End Sub 语句之间。当在运行时单击该命令按钮时,Access 便会执行 OpenForm_Click 事件过程中的代码。

代码的开始和结尾已经有了,剩下的事情就是写实际要用的代码。请键入以下代码行:

```
DoCmd OpenForm "Orders"
```

自然,这比 Command Button Wizard 代码要小得多,后者包含错误处理及其他语句,这些内容将在后面几章中介绍,但这并不意味着它不做工作,至少在现在。有关 Access Basic 编程环境最有意义的事情之一是用户能够增量式地建立代码,只要认为需要,便可在过程中不断地增加代码。

在继续下面的内容之前,先测试一下该过程,方法是离开 Design 视窗,并在 Form 视窗中显示该窗体。为此,需要以下步骤:

1. 选择 File,Save 以保存该窗体及模块(或者按 Ctrl+S 键,或者单击工具条上的 Save 按钮)。
 2. 关闭该模块。
 3. 单击 Form View(窗体视窗)工具条按钮,或者选择 View 菜单中的 Form 项。
 4. 单击 Open Form 按钮,当 NWIND 数据库的 Orders 窗体被装入并显示时,观察之。
- 在继续之前,单击工具条上的 Design 按钮,返回到 Design 视窗。

2.2.2 事件属性与 Build 按钮

在保存 Open Form_Click 事件过程时,Access 把 Open Form 命令按钮对象的 OnClick

属性设置为[Event Procedure]。为了证实这一点,可以关闭或者最小化刚打开的 Orders 窗体,然后选定 Form1 模块窗口,并观察其 Properties 窗口。选定该命令按钮,然后从 Properties 窗口的下拉式列表选定 Event Properties,以证实 OnClick 属性正文框的变化,如图 2.4 所示。如果在正文框的任何一个地方单击,或者左侧,或者右侧,不仅会出现一下拉式箭头,而且还会出现另一个带有三个句点的按钮。

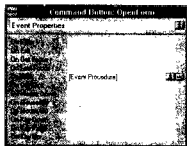


图 2.4 OpenForm Event Properties 窗口

用户对下拉式箭头与 Build (建立)按钮都应熟悉,下拉式箭头可引出可用宏及[Event Procedure]选项的列表,Build 按钮可显示 Choose Builder 对话框,同时还可右击属性框,并选择 Build 菜单项。在任何情况下,该事件过程都将出现在 Module 窗口中。

2.3 使用事件过程和函数

注意,现在 CBF 模块工具条中的 Object 框反映了当前 OpenForm 命令按钮对象。单击 Object 框上的下拉箭头可显示出该窗体的其他对象。在这种情况下,窗体、OpenForm、Detail0 以及缺省(常规)对象便是有效的选择。Procedure 框上的下拉式箭头可引出命令按钮对象能够触发的其他事件。具有事件过程的事件在该事件名的旁边有一检查标记,单击下拉式箭头,观察列表,就能证实 OpenForm Click 事件是活动的(如果事件分配有程序代码,则它是活动的)。

要想不仅能观察到当前窗体的对象和事件,而且还能观察到当前其他打开的窗体的对象和事件以及全局模块,请按 F2 键,显示 View Procedures(观察过程)对话框,如图 2.5 所示。



图 2.5 View Procedures 对话框