

第一章 Visual Basic 基本概念

本章是一个入门章节,主要是介绍操作 Visual Basic 所需具备的基本条件,同时也介绍启动与退出 Visual Basic 的方法。读者读完本章,相信必定可以了解下列基本知识:

1. 启动及退出 Visual Basic 的方法。
2. 认识 Visual Basic 的环境。
3. 了解工具箱的使用方法。
4. 了解在 DOS 环境及在 Windows 环境下建立应用程序的差别。
5. 了解 Visual Basic 的度量单位像素(Twip)。

自从 1990 年 5 月 22 日 Microsoft 公司在美国一场壮观的 Microsoft Windows 3.0 发表会后,几乎所有的 PC 软件产品开发即以适用 Windows 环境为目标。轻敲按钮,弹指之间即可协助用户完成工作,的确给了用户很大的便利。

Windows 环境尽管好用,不过如何设计可以在 Windows 环境下执行的 Windows 应用程序,却一直是程序设计人员的恶梦。基本上各位可以使用下列三种方式设计有关的 Windows 应用程序:

1. 使用 C 语言,配合 Windows API(Application Programming Interface)函数,例如,Visual C++ 或 Borland C++ 均提供以这种方式设计 Windows 应用程序。
2. 使用 C++ 配合 MFC,所谓的 MFC 是 Microsoft Foundation Class Library 的缩写,这是一个专门提供设计 Windows 应用程序的类型函数库,在 C++ 语言中,配合此类函数库的调用,可以很方便地建立 Windows 应用程序,例如,Visual C++ 提供这种方式设计 Windows 应用程序。
3. 使用 Visual Basic for Windows 软件,这也是本书的主题。事实上,这就是说,可以很容易地使用该软件设计 Windows 应用程序。

1.1 不一样的工作环境

在 DOS 环境下设计 DOS 应用程序时,程序设计人员可以独占性地使用系统资源(例如 CPU、内存、屏幕、键盘等等),执行数据的输入/输出和一般运算。不过在 Windows 环境下,则并不如此单纯,下面将介绍 Windows 应用程序和 DOS 应用程序的工作环境的差异。

1. 多任务(Multitasking)的特性

Windows 是个多任务(Multitasking)的作业环境,也就是在同一时间可能会有好几个应用在执行,在此情况下,所设计的应用程序无法独占性地使用所有系统资源(系统资源是指 CPU、内存、键盘等等)。而 DOS 则只是一个单用户环境,程序设计时,你可以充分使用系统可用资源(available resource)。

例如,对 DOS 应用程序而言,数据一定可在屏幕上执行输出,不可能有其它程序共同分享屏幕资源。但是在 Windows 环境下,一个屏幕可以同时出现好几个应用程序的输出窗口,因此,对于一个 Windows 应用程序而言,在数据输出时,必须考虑与其它正在执行的应用程

序共同分享屏幕资源。所以,所有的 Windows 应用程序必须遵照 Windows 操作系统特有的界面(Interface)执行运算,以确保 Windows 操作系统能有效地管理系统资源,供应用程序使用。

2. 操作界面

在 DOS 环境下,程序设计人员可以独占性使用屏幕,只要执行几个命令(Write 或 Print)调用,即可将数据输出,或是接收所输入的数据。

在 Windows 环境下,若想执行输入或输出动作,首先必须在屏幕上设计一个窗口,然后再通过此视图执行输入与输出。当然同一个应用程序是可能存在多个应用程序窗口的,这样,可形成多文档界面(multiple Document Interface)。

3. 数据的输入

在 DOS 环境下,若有输入动作发生,所输入的数据一定是供当前的正在执行的应用程序使用。在 Windows 环境下,由于有多个应用程序在执行,因此,究竟数据输入时,是由哪一个应用程序接收,将由 Windows 操作系统管理。

4. 数据输出模式

在 DOS 环境下,操作模式是假设在文本(Text)模式下工作,若想执行图形输出,必须先切换至图形(Graphics)模式,若不想显示图形输出,必须切换回文本模式。在 Windows 环境下,图形模式是基本工作模式,你可以很顺利地使用该模式执行文本、图形数据的输出。

看了以上的叙述可知 Windows 应用程序的设计的确较为困难,事实上以 C 语言设计 Windows 应用程序的确相当复杂,不过用 Visual Basic 设计 Windows 应用程序则不太困难。过去在 DOS 环境下,各位可能知道,汇编语言算是低级语言,以汇编语言设计 DOS 应用程序相当复杂。而 Basic 或 C 则称之为高级语言,它们设计 DOS 应用程序则简单多了。

若以 Windows 的观点来看,则我们可以将以 C 设计 Windows 应用程序想成低级语言,较复杂且难学。而以 Visual Basic 设计 Windows 应用程序则是高级语言,比较简单好学。

1.3 Visual Basic 的特色

Visual Basic 是 1991 年推出的,用它设计 Windows 应用程序较简单的原因是,一般 Windows 复杂的界面处理动作,例如,滚动条控制(scroll bar)、菜单建立(menu)、对话框(dialog box)……等复杂的动作,皆已被隐藏,读者可用几个步骤轻松地完成工作。

Visual Basic 是一种事件驱动(Event-Driven)的程序概念。你必须在程序内,设计各种事件(event)的处理方法(相当于程序码)。当此事件发生时,随即执行该事件的程序码。例如,你可以在程序内设计下列事件的处理方法。

1. 按关闭钮,可结束应用程序执行。
2. 拖动鼠标,可随着鼠标移动而画线。
3. 按清除钮,可清除窗口内容。

则你所设计的应用程序在执行时,只要有上述 3 个事件的任一事件发生,程序就执行所指定的任务,若是发生了未定义的事件,本应用程序则不予理会。

用 C 语言设计的 Windows 应用程序冗长,但在冗长的程序码中,大约 80%~90% 的程序码是用于建立用户界面,而只有约 10%~20% 左右的程序码是真正在处理应用程序本

身。Visual Basic for Windows 软件的诞生真是 Windows 应用程序设计人员的一个福音,因为有了此套软件后,你可以很轻松地使用 Visual Basic 所提供的工具建立用户界面,从而将精力花费在应用程序本身,如此将可以大大地提高设计程序的工作效率。

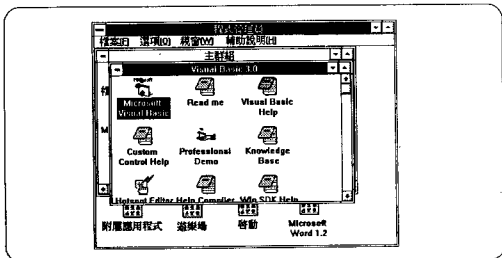
1.4 硬件需求

有关安装 Visual Basic for Windows 的步骤可参考附录 A,至于安装此软件所需的硬件配置则如下所示:

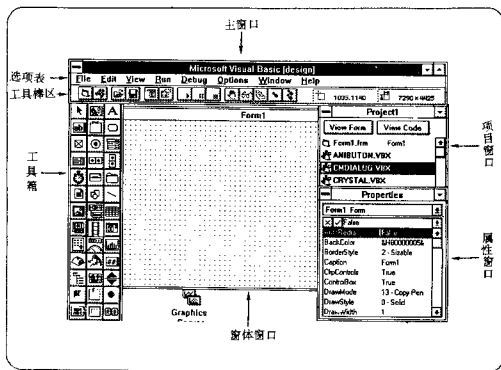
1. IBM PC 或兼容机,同时具有 80286 或更高档的微处理器。
2. 硬盘驱动器。
3. 5.5 英寸或 3.5 英寸的软盘驱动器。
4. 1M 内存(RAM)。
5. 鼠标。
6. MS-DOS 3.1。
7. 中文版或英文版的 Microsoft Windows 3.0 或更新版。

1.5 启动 Visual Basic for Windows

你可以将 Visual Basic 安装在中文版 Windows 下,也可以将 Visual Basic 安装在英文版 Windows 下,两者间最大差别在于,若将 Visual Basic 安装在中文 Windows 下,以后将可处理中文数据。由于国内的电脑大多安装有中文版的 Windows,因此,本书将以在中文版的 Windows 下讲解此软件。不过本书也适合于在英文版的 Windows 下,使用 Visual Basic 的读者首先请进入中文 Windows 且打开 Visual Basic 群组,如下所示:



连接两次 Microsoft Visual Basic 项目或其图标将可进入 Visual Basic 集成环境,如下所示:



上图是 Visual Basic 3.0 专业版 (Professional) 的窗口界面, 如果你是使用 Visual Basic 3.0 标准版 (standard), 则在工具箱内只能见到 21 个工具 (将在下一节说明)。

1.6 Visual Basic 环境说明

成功地进入 Visual Basic 集成环境后, 各位可见到 Visual Basic 窗口 (可参考前一小节, 一般可以将 Visual Basic 集成环境分成 5 个窗口, 如下所示):

1. 主窗口 Main Windows

主窗口内包含一个菜单区 (内含 8 个选择项) 和一个工具棒区 (Toolbar), 工具棒区的功能键主要是提供程序设计时常用的功能, 这样各位可以在此直接按一个键即可执行某个功能。

此外, 只要按一下菜单区的某个选择项, 即可打开此选择项底下的下拉式菜单 (pull down menu), 然后各位可按自己的需要执行所要的功能。

下面是工具棒区各钮的意义:

- : 建立一个新的窗体 (form)。你也可以执行 File 菜单的 New Form 命令, 执行此钮的功能。
- : 建立一个新的模块 (module)。你也可以执行 File 菜单的 New Module 命令, 执行此钮的功能。
- : 打开现存的项目文件 (project file)。你也可以执行 File 菜单的 Open Project 命令, 执行此钮的功能。
- : 存储当前的项目文件 (project file)。你也可以执行 File 菜单的 Save Project 命令, 执行此钮的功能。
- : 显示可设计菜单 (menu design) 的窗口, 以设计某个应用程序的菜单。你也可以

执行 Windows 菜单的 Menu Design 命令,执行此钮的功能。



: 显示属性窗口(Properties Window),若属性窗口被遮住而不在屏幕上显示,则可以按此钮,以便显示它。也可以执行 Windows 菜单的 Properties 指令,执行此钮的功能。



: 正式执行当前所设计的 Windows 应用程序。你也可以执行 Run 菜单的 Start 命令,执行此钮的功能。



: 当 Windows 应用程序在执行时,令程序暂停执行。你也可以执行 Run 菜单的 Break 命令,执行此钮的功能。



: 当 Windows 应用程序在执行时,令程序停止执行,同时返回 Visual Basic 屏幕,以便可以继续编辑或设计程序。你也可以执行 Run 菜单的 End 命令,执行此钮的功能。



: 在当前执行行内切换中断点,通常是用于调试(debug)程序阶段。你也可以执行 Debug 菜单的 Toggle Break 命令,执行此钮的功能。



: 在编码窗口(Code Window)内,显示当前选项的值。你也可以执行 Debug 菜单的 Instant Watch 命令,执行此钮的功能。



: 显示当前调用(active call)的结构,你也可以执行 Debug 菜单的 Calls 命令,执行此钮的功能。



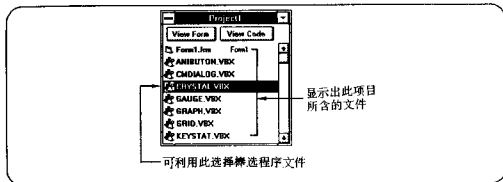
: 在编码窗口(Code Window)内,一次执行一条命令。你也可以执行 Debug 菜单的 Single Step 命令,执行此钮的功能。



: 在编码窗口(Code Window)内,一次执行一个过程(procedure)或一条命令。你也可以执行 Debug 菜单的 Procedure Step 命令,执行此钮的功能。

2. 项目窗口(Project Windows)

以前在 DOS 环境中,通常是一个程序文件即可完成所要的功能。不过在 Windows 环境中,若想设计一个功能,可借用许多程序文件完成,为了方便管理这些程序文件,可借用一个项目文件(Project file)管理它们。而项目窗口则是列出当前项目文件包含哪些程序文件。

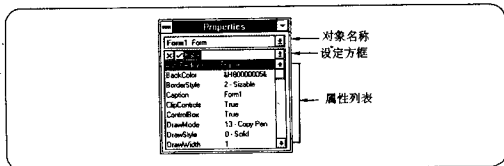


上述项目窗口内包含一系列的的文件,同时有一个菜单,你可以利用选择棒选择某个文件,然后执行下列两个功能钮。

① View Form 钮:若是以实线显示,则表示按此钮可促使在屏幕上显示选择棒所选窗体文件的窗体(form)。若是此钮以虚线显示,则表示当前选择棒所选的并不是窗体文件(注:以 frm 为扩展文件名文件称窗体文件),所以无法启用此钮。

② View Code 钮:若以实线显示,则表示按此钮可在屏幕上显示所需文件的程序码。

3. 属性窗口(Properties Window)



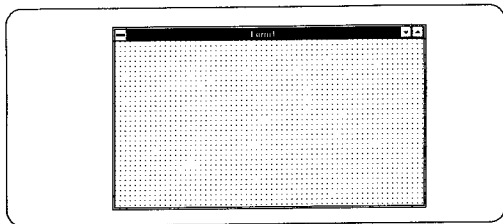
属性窗口包含三个部分:

①名称栏位:此栏位列出窗体(form)或是控件(control,control 的意义将在窗体(form)窗口说明)的名称。

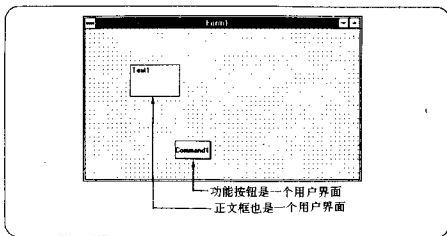
②设定方框:列出属性列表内当前所选的属性。

③属性列表:列出当前所选的窗体或控件包含的所有的属性。

4. 窗体窗口(Form)

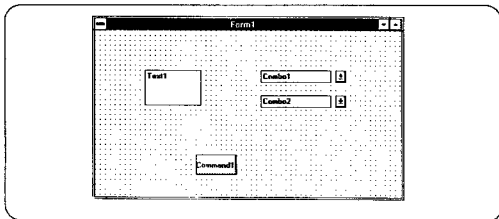


上述的窗体窗口(Form),简称为窗体,而标题栏的Form1 是此窗体的名称。在以 Visual Basic 设计 Windows 应用程序时,通常都是先在上述窗体内利用工具箱(下一章将说明)设计用户介面,如下所示:



在 Visual Basic 的术语中,又可将上述用户介面称为某个工具绘制的对象(object),不同的工具将可画出不同的对象,而我们又可将各个不同的对象称之为控件(control)。

下面是一个窗体内含 4 个控件的实例。



若是各位用 C 设计 Windows 应用程序,将可发现有相当多的时间是用于建立上述窗体(form)或各控件(control)的。但是若使用 Visual Basic 设计 Windows 应用程序,程序设计人员只要利用工具箱的工具,便可很轻松地画出窗体或控件,然后 Visual Basic 将自动产生它们的程序码。

5. 工具箱(Toolbox)



Visual Basic 3.0 标准版或
Visual Basic 2.0 的工具箱



Visual Basic 3.0 专业版
的工具箱

从上述图中各位应可看到专业版的 Visual Basic 较标准版有更多工具可以使用,当然你也可以另外增加工具箱内的工具,事实上在项目窗口(Project Window)内,凡是以 VBX 为扩展文件名的文件都提供一个工具箱的工具。例如,GRID.VBX 便是网格工具(Grid,有关此工具的按钮,下面会说明)的程序文件。事实上专业版和标准版的最大差别在于专业版的

项目窗口(Project Windows)提供更多以 VBX 为扩展文件名的文件,因此,使得专业版在工具箱内有较多的工具可以使用。

本书后面的程序实例,都可以在专业版或标准版内执行,下面是工具箱内各常用工具的使用说明。工具绘完控件之后,Visual Basic 将自动选取此工具。



: 指针(Pointer),在工具箱内这是唯一一个无法用于绘控制按钮(control)的工具,它主要是用来移动表格(form)或是控制按钮。或是用于更改表格或控制按钮的大小。当你使用其它工具绘完控制按钮之后,Visual Basic 将自动选取此工具。



: 图片方框(Picture Box),用于显示图形,当然也可用于显示文件,或是可将其它控件(control)放在此图片方框内。



: 标签(Label),用于显示字符串,而此字符串内容将无法更改。例如,你可以将它放在图片的下方,用于标明图片的名称。



: 正文框(Text Box),供输入和显示文本数据。



: 框架(Frame),供建立框架,以后你可以将其它控件放在此框架内。



: 功能钮(Command Button),供建立功能钮,当用户按功能钮时,将可执行所建的功能。



: 复选框(Check Box),供建立一方框,然后供选择以便显示是 TRUE 或 FALSE,或是 YES 或是 NO,值得注意的是在一个窗体(Form)内,你是可以同时设定多个复选框的。



: 选项钮(Option Button),供建立一选项,值得注意的是在一系列选项钮组内,你一次只能选一个选项钮。



: 组合框(Combo Box),可用于建立包含列表(list)性质和文本(text)性质的方框,用户可从列表方框内执行选取某项目,或是直接在文本方框内输入某项目。



: 列表方框(List Box),可用于显示列表性质的数据项,用户可从此列表内选择某项目。若是列表无法一次显示时,你将可以滚动此列表。



: 水平滚动条(Horizontal Scroll Bar),提供水平滚动条供你选择某范围内的值。



: 垂直滚动条(Vertical Scroll Bar),提供垂直滚动条供你选择某范围内的值。



: 定时装置(Timer),可用于在指定的时间内执行指定的工作。



: 磁盘机列表框(Drive List Box),显示系列磁盘编号及供选择某磁盘之用。



: 目录列表框(Directory List Box),显示系列目录,同时供选择某目录之用。



: 文件列表框(file List Box),显示系列文件,同时供选择某文件之用。



: 外形工具(Shape),可以利用此工具在窗体(Form)内建立矩形或圆或椭圆。



: 直线工具(Line),可以利用此工具在窗体(Form)内画直线。



: 影像工具(Image),用于显示图片(Picture)或是图标(icon)或是位元文件(Metafile)。当你按此影像时,将具有类似功能钮(Command Button)的效果。



: 网格工具(Grid), 可用于建立具有试算表(spreadsheet)性质的方格, 然后你可以自行处理各存储格内的数据。



: OLE, 可用于在 Visual Basic 内执行有关 OLE (英文全名是 Object Linking and Embedding) 方面的应用。



: 数据控件(Data Control), 可供你连接至现存的数据库内, 同时在窗体(Form)内显示数据库内的信息。



: 通用对话框(Common Dialog), 使用建立常见的对话框, 例如, 打开文件, 存文件, 字型设定或颜色选取。

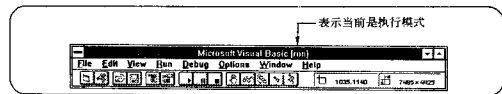
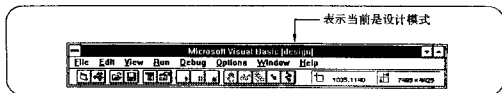
由于篇幅有限, 本书将不介绍以上所有工具, 取而代之的是介绍上述 Windows 应用程序中最常用的工具。至于未介绍的部分及专业版增加的部分, 则请读者参阅有关书籍。

1.7 Visual Basic 的工作模式

Visual Basic 共有 3 种工作模式, 分别如下所示:

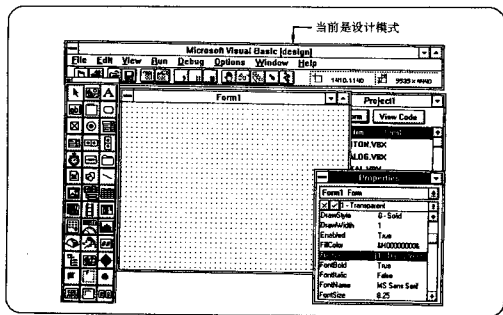
1. 设计(design)模式: 成功地启动 Visual Basic 后, 即可自动进入此模式内, 在此模式下, 用户可以自行设计 Visual Basic 程序。
2. 调试(debug)模式: 在此模式下屏幕将出现调试窗口(Debug Window), 通常我们可以利用此窗口测试一些 Visual Basic 命令, 每当执行一道命令, 此窗口将立即列出执行结果。
3. 执行(Run)模式: 程序设计好以后, 若执行此所设计的程序将可进入此执行模式内。

然而我们怎样辨别当前是在哪一种模式内呢? 答案是屏幕 Visual Basic 的主窗口标题栏将列出目前的工作模式, 如下所示:



1.8 使用工具箱

假设当前屏幕内容如下所示：



由于当前窗体(Form)窗口尚未有其它控件,所以属性(Properties)窗口将列出当前窗体窗口的属性。

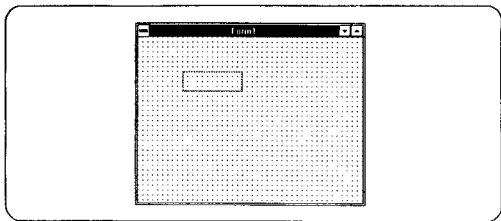
使用工具箱在窗体窗口内绘制控件的步骤如下：

1. 选取工具箱的某工具,选取的方式是在该工具钮上按一下(以后不特别注明都表示按鼠标左键),此时该工具钮将有被按下的现象。
2. 将鼠标在窗体窗口某位置按一下,然后拖动鼠标,此时你将看到有一浅灰色框随着拖动鼠标而改变大小。
3. 当框的大小固定后,若松开鼠标按钮,便算是利用工具建立对象成功了。

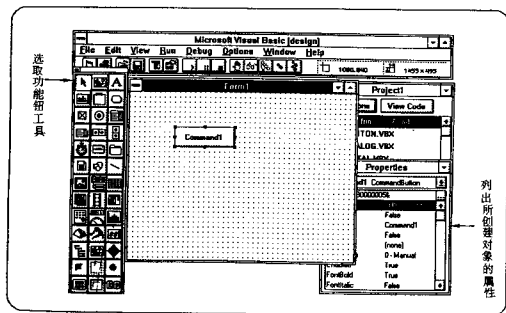
下而是利用工具箱的功能钮(Command Button)在窗体窗口内建立功能钮控制按钮的实例。下面是选取功能钮的实例。



下而是将鼠标箭头在窗体窗口内按一下,且拖动鼠标时,将看到浅灰色框,如下所示：

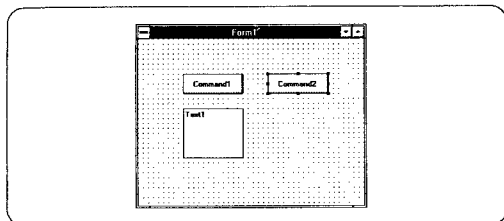


松开鼠标按键,便可看到利用功能钮(Command Button)所建的对象(object)(注意,此对象又可称为控件),如下所示:

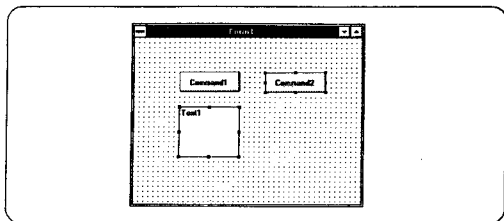


同时当你建立一个对象成功后,此对象将被外框(各角落及各边线中央有黑方框)框住,同时在工具箱内,将自动返回选取指针工具(Pointer)位置,如上图所示。此时若再观察属性(Properties)窗口,将可看到当前所建控件的属性(这些属性的预设值),如上图所示,此时你可在属性窗口更改所建控件的属性(详情可参考 3.1.1 节)。

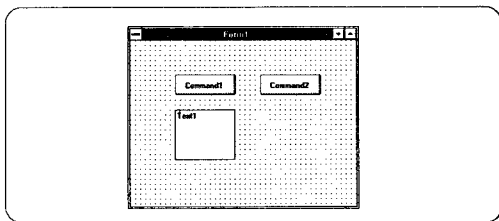
一个窗体窗口可能会同时有不同工具创建好的控件,如下所示:



在上图中窗体窗口内含 3 个控件, 分别是 Command1、Command2、Text1。而 Command2 外框被框住了, 这表示 Command2 是当前所选的控件。除了新建控件将被自动选取外, 若是你想选某一控件, 只要利用鼠标在该控件上按一下即可, 例如, 下面是用鼠标在 Text1 控件上按一下, 选取 Text1 控件的情形。



值得注意的是, 属性(Properties)窗口将显示当前所选控件的属性。因此, 若是你希望更改某控件的属性, 只要选该控件即可。若是你在上述 3 个控制元件外, 但在窗体窗口内按一下, 将可看到下列结果。



上述没有控件被选取,表示当前是选取窗体窗口本身,此时属性窗口将显示窗体窗口的属性。当然在此情况下,你可以利用属性窗口修改此窗体窗口的属性。

1.9 控件的编辑

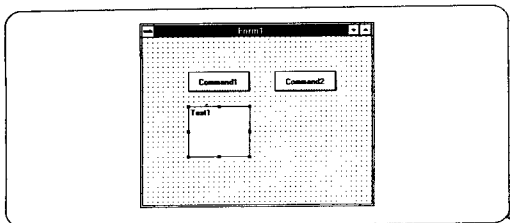
成功地在窗体窗口建立控件之后,各位可能想进行下列编辑操作:

移动控件

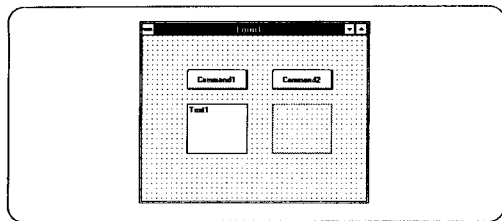
你可以利用更改属性窗口内容的方式移动控件,此方法将在 5.1 节说明。另一种简单的方法是利用拖动控件的方式移动控件的,其步骤如下:

1. 选取要移动的控件。
2. 拖动所选的控件。
3. 移至定位后,放松鼠标按键即可。

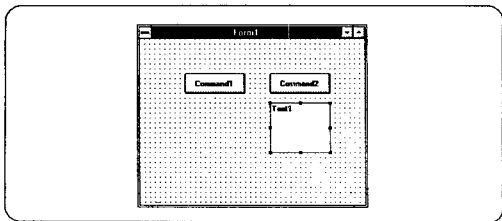
下面是选取某控件的实例。



下面是拖动控件的实例。



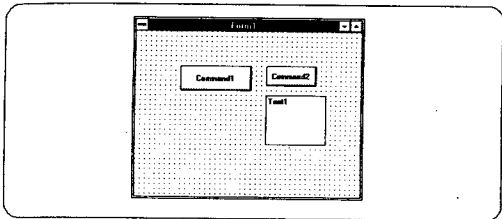
放松鼠标按键,将可看到下列执行结果。



放大或缩小控件

你可以利用更改属性窗口内容的方式修改控制按钮的大小的,此方法将在 5.1 节说明。另一种简单的方法是利用拖动所选取控件外框角落的黑方框或外框各中点黑方框的方式,也可修改控件的大小。

下面是将 Command1 控件放大,将 Command2 控件缩小的实例。



删除控件

在窗体窗口内成功地建立某控件后,若感觉此控件没有用,想将它删除,可按下列步骤操作。

1. 选取该控件。
2. 执行主窗口 Edit 菜单的 Delete 命令。

假如你想删除 Command2 控件,请先选取它,如下所示:

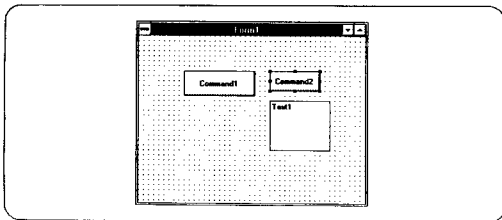


图 1-10 删除 Command2 按钮

执行 Edit 菜单的 Delete 命令后,结果如下:

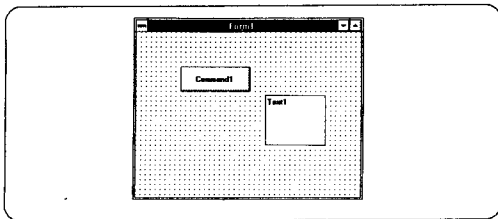


图 1-11 删除 Command2 按钮后的结果

复制控件

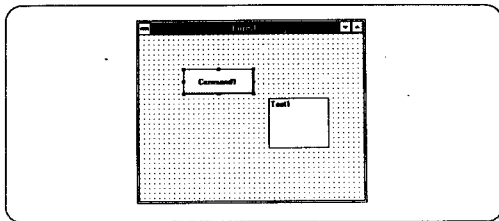
在程序设计时,有时我们可能会想设计大小相同、但位置不同的控件,此时可考虑先用拷贝方式在窗体窗口内另复制一份,然后再将所复制的控件移至定位。

复制控件步骤如下:

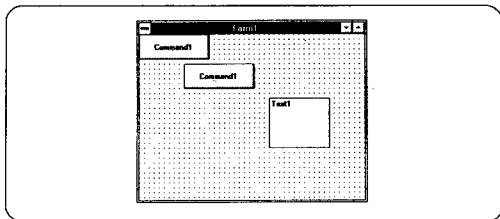
1. 选取要复制的控件。
2. 执行主窗口 Edit 菜单的 Copy 命令。
3. 此时所拷贝的控件将在剪贴簿内。
4. 执行主窗口 Edit 菜单的 Paste 命令,如果是初次复制控件,此时将出现对话框,询问是否要建立控件数组(control array),如果以后在程序设计时,你想要以数组的方式处理所复制的控件,则选择 Yes,否则选择 No。无论怎样,这些控件都会在窗体窗口左上角出现。

5. 将新复制元件拖至定位即可。

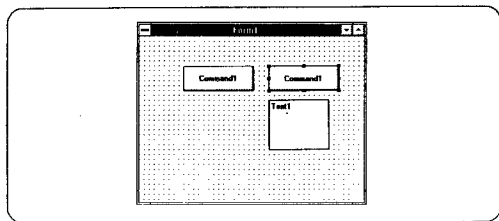
下面是要复制 Command1 控件的实例,首先请选取 Command1,如下所示:



请先执行主窗口 Edit 菜单的 Copy 命令,再执行主窗口 Edit 菜单的 Paste 命令。下面是成功复制 Command1 控件的实例。



下面是将所复制的控件拖动至定位的实例。



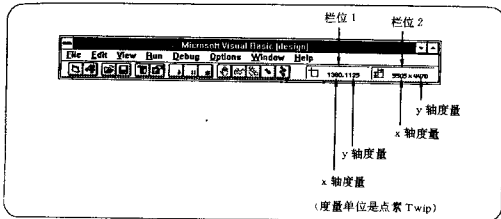
1.10 度量单位

Visual Basic 的度量单位叫 Twip, Twip 是 Microsoft 公司自建的度量单位。在本书中,笔者将 Twip 解释为“点素”。在常见的电脑概念中,在图形模式状态下,通常是以“像素”(pixel)做为度量的单位。例如,若是有一个 640×480 的屏幕,则 640 或 480 的度量单位是“像素”(pixel)。

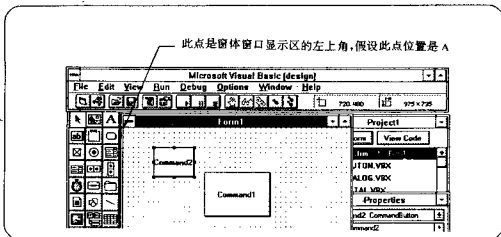
Twip 和 pixel 的关系如下:

1 个点素(Twip) = $1/20$ 像素(pixel)

由于在电脑界,72 个像素(pixel)等于 1 英寸,所以 1 个点素(Twip)等于 $1/1440$ 英寸。在 Visual Basic 的主窗口内,有两个栏位和度量有关,如下所示:



在上图“栏位 1”是指当前窗体窗口内所选控件的左上角距离窗体窗口的显示区域左上角的差距,而“栏位 2”则是指窗体内所选控件本身的宽度(x 轴)和高度(y 轴)。例如,若有一屏幕内容如下:



从上图可知: