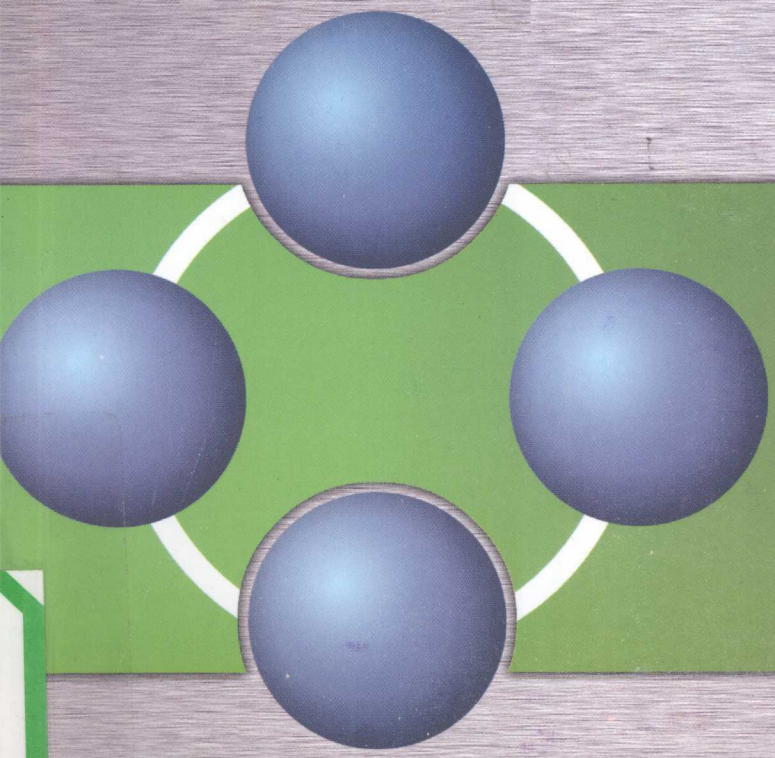
VFP编程实例及提高
——通向编程实战之路

VFP BIANCHENG SHILI JI TIGAO
TONGXIANG BIANCHENG SHIZHAN ZHILU

曾晓红　主编

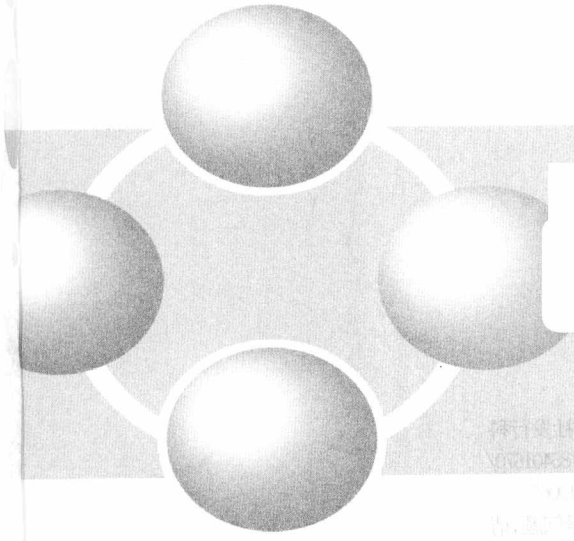


四川大学出版社

VFP编程实例及提高
——通向编程实战之路

VFP BIANCHENG SHILI JI TIGAO
TONGXIANG BIANCHENG SHIZHAN ZHILU

主　　编　曾晓红
副 主 编　冉　婕　秦　萌　邱广文
编委会名单　曾晓红　冉　婕　秦　萌　邱广文
石玉萍　李　坤　包龙翔　王健鹰



四川大学出版社

责任编辑：曾　鑫
责任校对：廖庆扬
封面设计：墨创文化
责任印制：李　平

图书在版编目(CIP)数据

VFP 编程实例及提高：通向编程实战之路 / 曾晓红主编. —成都：四川大学出版社，2010.7
ISBN 978-7-5614-4968-4

Ⅰ.①V…　Ⅱ.①曾…　Ⅲ.①关系数据库—数据库管理系统，Visual Foxpro—程序设计　Ⅳ.①TP311.138

中国版本图书馆 CIP 数据核字（2010）第 154164 号

书名　**VFP 编程实例及提高**
——通向编程实战之路

主　　编　曾晓红
出　　版　四川大学出版社
地　　址　成都市一环路南一段 24 号 (610065)
发　　行　四川大学出版社
书　　号　ISBN 978-7-5614-4968-4
印　　刷　郫县犀浦印刷厂
成品尺寸　185 mm×260 mm
印　　张　18
字　　数　443 千字
版　　次　2010 年 9 月第 1 版
印　　次　2010 年 9 月第 1 次印刷
定　　价　30.00 元

◆读者邮购本书，请与本社发行科联系。电 话：85408408/85401670/85408023　邮政编码：610065
◆本社图书如有印装质量问题，请寄回出版社调换。
◆网址：www.scupress.com.cn

前　言

由于 Visual FoxPro 教材一般都是讲解 Visual FoxPro 开发平台的基础知识，但是，从基础知识到编程实战仍然还有一段相当漫长的路，一般学习者很难把 VFP 基础知识应用到中、小型信息系统的开发实践中，导致学习者动手能力没有得到很大提高的现象发生。本书旨在帮助 Visual FoxPro 学习者提高把基础知识转化为能够进行系统开发的技能。

为填补 Visual FoxPro 编程基础和编程实战之间的空白，充分展现 Visual FoxPro 的强大功能，并满足不同层次读者的学习需要，我们在内容设计方面注重强化基础、突出重点、侧重应用；并对不同的实例予以恰当的分类，在文字编排方面力求语言精练、通俗易懂，方便读者的学习。

本书共三章，第一章面向对象编程基础，介绍面向对象程序设计的基本理念和基本概念，包括类、对象、对象的属性、对象的事件和事件发生的顺序等内容。第二章介绍了 VFP 实训案例，在回顾结构化程序设计的基础上，详细描述了 Visual FoxPro 中各类基本控件的属性和根据不同目的在基本控件的事件中编写相应的代码以实现特定的功能、充分利用函数降低编程难度、数据查询方法、创建可视化类的方法、菜单和报表的制作、多媒体功能的实现等内容；其中不乏 Hanoi 塔问题动画、产品加锁(时间掣)技术、通用查询、把汉字转换为拼音、利用摄像头采集照片信息、自动生成错误日志的主程序、音视频播放、编排考场、制作和调用 CHM 帮助文件等实用程序。所有实例均详细叙述了编程的具体步骤、相应的对象及其属性值、事件或方法的全部代码。第三章介绍了全国计算机等级考试二级 VFP 实例，旨在掌握 Visual FoxPro 实用编程的基础上，帮助读者通过全国计算机等级考试二级 Visual FoxPro。

配套光盘包含本书的所有实例，实例均在 Visual FoxPro 9.0 下调试通过。

本书在编著过程中借鉴了同行专家的教学科研成果。在出版的过程中，得到了昭通师范高等专科学校领导的大力支持和四川大学出版社的鼎力相助，在此一并感谢。

由于编者水平有限，本书难免有不足之处，光盘中的一些程序还可以作进一步改进，诚恳盼望广大读者提出宝贵意见和建议予以批评指正。

编　者

ZTZXH@163.COM

2010 年 3 月

目 录

第 1 章 VFP 面向对象编程基础 …… (1)
第 1 节 基类及其引用 …… (1)
第 2 节 VFP 常用事件和方法 …… (3)
第 3 节 事件发生的顺序 …… (6)
第 4 节 得到和失去焦点 …… (8)
第 5 节 何时更新数据 …… (9)

第 2 章 VFP 实训案例 …… (11)
第 1 节 程序设计基础 …… (11)
第 1 部分 结构化程序设计及应用 …… (11)
实验 1 求一元二次方程 …… (11)
实验 2 百钱百鸡问题(穷举法) …… (15)
实验 3 Fibonacci 问题(迭代法) …… (17)
实验 4 牛顿切线法(迭代法) …… (19)
实验 5 欧几里得算法(求最大公约数,迭代) …… (23)
实验 6 打印九九表 …… (25)
实验 7 冒泡法排序 …… (27)
实验 8 汉诺塔问题 …… (30)
实验 9 选择法排序动画 …… (35)
实验 10 折半查找 …… (37)
实验 11 阿拉伯数字转换成中文大写 …… (40)
第 2 部分 函数的使用 …… (44)
实验 12 从身份证中获取出生日期 …… (44)
实验 13 显示系统信息 …… (46)
实验 14 汉字转拼音 …… (47)
第 3 部分 数据查询 …… (50)
实验 15 父子表联动的表单 …… (50)
实验 16 模糊查询 …… (53)
实验 17 多表关联查询实例 …… (54)
实验 18 通用查询 …… (55)
实验 19 查询表中的任意字段 …… (62)
实验 20 设置比较符条件查询 …… (65)
实验 21 SQL 查询实例 …… (68)

第 2 节 面向对象程序设计……（70）
第 1 部分 常用控件的使用……（70）
实验 22 Hello World! ……（70）
实验 23 选项按钮组设计 ……（71）
实验 24 组合框设计 ……（74）
实验 25 自动换页 ……（77）
实验 26 页框标题的颜色控制 ……（80）
实验 27 动态添加列表框项目 ……（81）
实验 28 动态改变列表框的内容 ……（82）
实验 29 微调设计 ……（84）
实验 30 表格中添加微调和复选框控件 ……（85）
实验 31 日历控件应用实例 ……（89）
实验 32 容器类控件设计 ……（92）
实验 33 形状控制 ……（97）
实验 34 计算器设计 ……（99）
实验 35 动态增加列表框中的列 ……（113）
实验 36 表格中的颜色控制 ……（115）
实验 37 表格录入数据时光标的控制 ……（116）
实验 38 图形浏览器设计 ……（120）
实验 39 自动打开组合框的设计实例 ……（122）
实验 40 单击表格标题栏对表排序 ……（123）
第 2 部分 类和事件……（124）
实验 41 工具栏设计 ……（124）
实验 42 移动鼠标改变图形大小 ……（129）
实验 43 显示事件列表的表单 ……（130）
第 3 部分 报表及菜单 ……（131）
实验 44 主菜单设计 ……（131）
实验 45 向导生成、手动建立报表 ……（137）
实验 46 快捷菜单设计和调用 ……（144）
实验 47 报表中图片透明 ……（146）
第 3 节 VFP 中的多媒体应用 ……（148）
实验 48 图像、声音信息保存到数据库 ……（148）
实验 49 背景音乐的实现 ……（154）
实验 50 多功能音视频播放器 ……（159）
实验 51 给程序加入扫描功能 ……（160）
实验 52 利用摄像头采集图像 ……（162）
第 4 节 VFP 的项目管理及后期制作 ……（171）
实验 53 VFP 项目的建立和管理 ……（171）
实验 54 安装盘的生成 ……（175）
实验 55 制作和调用 CHM 帮助文件 ……（183）

实验 56 主程序设计 …… (195)
实验 57 利用 InstallShield Express 制作安装盘 …… (199)
第 5 节 实用编程 …… (216)
第 1 部分 其他使用技巧 …… (216)
实验 58 嵌入 EXCEL 和 WORD …… (216)
实验 59 闪烁立体字幕效果 …… (218)
实验 60 超链接设计 …… (219)
实验 61 异形表单设计 …… (220)
实验 62 防止表单重复运行 …… (222)
实验 63 隐藏和显示任务栏 …… (224)
实验 64 使用键盘操作表 …… (225)
第 2 部分 产品加密与数据备份 …… (228)
实验 65 低级文件函数加密数据表 …… (228)
实验 66 产品加锁(时间掣)技术 …… (231)
实验 67 数据库备份与恢复程序 …… (235)
第 3 部分 考务管理应用 …… (238)
实验 68 产生随机考试号 …… (238)
实验 69 自动编排考场 …… (240)
实验 70 手动编排考场 …… (243)

第 3 章 全国计算机等级考试二级 VFP 实例 …… (248)
第 1 节 全国计算机等级考试二级 VFP 上机概要 …… (248)
第 2 节 全国计算机等级考试二级 VFP 实例解析 …… (250)
第 1 部分 基本操作题 …… (250)
实验 1 基本操作题(一) …… (250)
实验 2 基本操作题(二) …… (255)
实验 3 基本操作题(三) …… (256)
实验 4 基本操作题(四) …… (258)
第 2 部分 简单应用题 …… (260)
实验 5 简单应用(一) …… (261)
实验 6 简单应用(二) …… (262)
实验 7 简单应用(三) …… (264)
实验 8 简单应用(四) …… (266)
第 3 部分 综合应用题 …… (268)
实验 9 综合应用(一) …… (269)
实验 10 综合应用(二) …… (271)
实验 11 综合应用(三) …… (273)

附录 2009 年全国计算机等级考试二级 VFP 考试大纲 …… (276)
参考文献 …… (279)

第 1 章　VFP 面向对象编程基础

Visual FoxPro(以下简称 VFP)是 Microsoft 公司从 Fox 公司的 FoxBase 数据库软件经过数次改良,并且移植到 Windows 之后形成的应用程序开发软件,主要适用于数据库的开发与管理。

VFP 是 Microsoft 公司推出的最新可视化数据库管理系统平台,是功能强大的 32 位数据库管理系统。它提供了功能完备的工具、极其友好的用户界面、简单的数据存取方式、独一无二的跨平台技术,具有良好的兼容性、真正的可编译性和较强的安全性,是目前最快捷、最实用的数据库管理系统软件之一。

VFP 的历史可追溯到 20 世纪 80 年代初,其前身是 dBASE 数据库管理系统,经过 30 年的发展,VFP 从最初的 DataBase 数据库软件,到后来的 FoxBase、FoxPro 2.5、FoxPro 3.0……直到今天的最新版本 FoxPro 9.0。本书以 VFP 9.0 为开发平台。VFP 不仅提供面向对象的程序设计方式,同时还保留面向过程的程序设计方式。面向对象的程序设计方式是以对象为中心,将数据和程序绑定在一起,封装在对象中,从而淡化解决问题的过程。面向对象的程序设计的内容包括类、设置对象、确定对象的数据(属性)、确定数据操作(方法)和对象的响应(事件)。以下详细介绍 VFP 面向对象程序设计方式的基本概念。

第 1 节　基类及其引用

要想利用面向过程的语言(如 C 语言)开发软件,除了要深入学习该语言的各种命令,熟练命令的语法外,还必须熟悉并能灵活应用该语言提供的系统库函数。而利用面向对象语言开发软件,除了要掌握命令的语法和系统库函数外,还必须熟悉系统提供的各种基类,以及各种基类的属性、事件和方法。VFP 为开发者提供了很多实用的基类,大大方便了开发者编程。

所谓类(Class),就是一组对象的属性和行为特征的抽象描述。或者说,类是具有共同属性、共同操作性质的对象的集合。在 VFP 中,类就像是一个模板,对象都是由类生成的,类定义了对象所有的属性、事件和方法,从而决定了对象的属性及其行为。使用类的目的主要是为了提高编程效率。

系统本身提供的类称为基类。VFP 的基类是系统内嵌的、并不存放在某个类库中。VFP 的基类共分成两个大类:容器类和控件类。

一、容器类

包含其他对象的类称为容器类,容器对象可作父类,其包含的对象称为子对象,并且允

许访问这些子对象。例如，表单（即窗口）对象作为容器，可以包含命令按钮、文本框、复选框等子对象，那么无论在设计时还是在运行时，都可以对该表单中任何一个对象进行单独操作。在设计阶段不仅可以改变上述子对象的属性，还可以给各子对象的事件编写代码；在运行时，可以向文本框内输入字符，可以通过单击命令按钮激发命令按钮的 click 事件发生，从而运行 click 事件中编写的代码。当然，容器内还可以包含容器类对象，例如，在表单中通过表格容器反映数据表和视图的数据内容是输入数据的一种重要手段。

容器类包括：容器（Container）、表单集（FormSet）、表单（Form）、表格（Grid）、页框（PageFrame）、页面（Page）、工具栏（Toolbar）、命令按钮组（CommandGroup）和选项按钮组（OptionGroup）等。

二、控件类

控件类的封装比容器类更为严密，不允许包含其他对象，换句话说，控件不能作为父对象，也因此丧失了一些灵活性。控件类没有 AddObject 方法程序。在该类中不能包含其他类，最典型的控件类就是命令按钮。容器类虽然在引用时可以视为一个整体，但无论是在设计阶段还是在运行阶段，其所包含的对象都是可以识别并可以单独操作的。

控件类包括：复选框（CheckBox）、下拉组合框（ComboBox）、命令按钮（CommandButton）、控件（Control）、自定义（Custom）、编辑框（EditBox）、图像（Image）、标签（Label）、线条（Line）、列表框（ListBox）、OLE 绑定型控件（OLE Bound control）、OLE 容器控件（OLE Container Control）、形状（Shape）、微调（Spinner）、文本框（TextBox）和计时器（Timer）等。

对象是客观世界中独立存在的、能够区分的实体。在进行容器类“子类”或“对象”的设计时，往往要引用容器中某一特定对象，这就要掌握 OOP（Object Oriented Programming，面向对象程序设计）方法中对象的标识方法。首先要明确下面几个问题：

1. 容器类中对象的层次

VFP 的对象分为容器对象和控件对象。容器对象中可包含容器对象和控件对象，这样就形成了容器对象嵌套层次关系，一般把一个“对象”的直接容器称为“父容器”，在引用特定的对象时，搞清该对象的“父容器”是至关重要的，不能把一个对象的间接容器错误地视为其“父容器”。

2. 对象局部范围的名称

每个对象都有一个名称。在给对象命名时，只要保证同一个“父容器”下的各对象不重名即可。换句话说，对象使用的是局部范围内的名称，因此不能单独使用对象名来引用对象。对象引用的一般格式是：

```
引用地址.对象名称
```

引用地址又分为绝对引用地址和相对引用地址，所以对象的引用又分为绝对引用和相对引用。

（1）绝对引用

如果引用的地址是从最外层容器算起直到目标对象，那就是绝对引用地址。用绝对引用地址引用对象称为绝对引用。例如，名为 frm_student 的表单中，有一个 cmd_quit 命令按钮，若设置其是否可用的属性为不可用，那么采用绝对引用的语法如下：

```
frm_student.cmd_quit.Enabled=.F.
```

(2)相对引用

如果引用地址从指定参照算起到目标地对象,那就是相对引用地址,用相对引用地址引用对象称相对引用。VFP 规定常用的相对引用关键字见表 1-1-1:

表 1-1-1　相对引用关键字

参照关键字	参照对象
This	当前该对象
ThisForm	包含当前对象的表单
ThisFormSet	包含当前对象的表单集
Parent	包含当前对象的父对象

例如,在上例中,如果 frm_student 表单的某个对象的事件中,要想设置 cmd_quit 命令按钮为不可用,那么采用相对引用的语法如下:

```
Thisform.cmd_quit.Enabled=.F.
```

或

```
This.Parent.cmd_quit.Enabled=.F.
```

需要注意的是,采用相对引用,一定要注意对象之间的层次关系,否则就会出错。所以上例中特别强调了"如果 frm_student 表单的某个对象的事件中,要想设置 cmd_quit 命令按钮为不可用"这个假设条件。

第 2 节　VFP 常用事件和方法

事件(Event)是每个对象进行识别和响应的行为和动作。我们对对象所做的操作(或者系统对某个对象的操作),如按钮被按动(单击)、对象被拖动、被改变大小、被鼠标左键双击等都可以看作对象的事件。在 VFP 中,对象可以响应多种事件。多数情况下,事件是通过用户的操作行为引发的,当事件发生时,将执行包含在事件过程中的全部代码。方法(Method)是对象能够执行的操作,是内置的通用过程,换句话说,是附属于对象的行为和动作。VFP 9.0 中的主要事件和方法包括:

Activate 事件:当激活表单、表单集或页对象,或者显示工具栏对象时,将发生 Activate 事件。

AddObject 方法:运行时,在容器对象中添加对象。

Addproperty 方法:向一个对象添加一个新属性。

Box 方法:在表单对象上画矩形。

Circle 方法:在表单上画一个圆或椭圆。

Click 事件:当在程序中包含触发此事件的代码,或者将鼠标指针放在一个控件上按下并释放鼠标左键,或者更改特定控件的值,或在表单空白区单击时,此事件发生。

Cls 方法:清除表单中的图形和文本。

Dblclick 事件:当连续两次快速按下鼠标左按钮(主按钮)并释放时,此事件发生。

Deactivate 事件：对于一个容器对象（例如一个表单），当所包含的对象没有焦点而不再处于活动状态时，该事件发生。

对于一个工具栏来说，当使用：

Hide 方法：隐藏工具栏时，该事件发生。

Destroy 事件：当释放一个对象的实例时发生。

DragDrop 事件：当完成拖放操作时，此事件发生。

DragOver 事件：在控件拖过目标对象时，此事件发生。

Draw 方法：重画表单对象。

Error 事件：当某方法中在运行出错时，此事件发生。

GotFocus 事件：当通过用户操作或执行程序代码使对象接收到焦点时，此事件发生。

Hide 方法：通过把 Visible 属性设置为“假”(.F.)，隐藏表单、表单集或工具栏。

Init 事件：在创建对象时发生。

Interavtive Change：当用户用键盘或鼠标改变对象的值时发生。

KeyPress 事件：当用户按下并释放某个键时，此事件发生。

Line 方法：在表单对象中画一条线。

Load 事件：在创建对象前发生。

LostFocus 事件：当某个对象失去焦点时发生。

MiddleClick 事件：当用户在一个控件上单击 3 键鼠标的中央键时发生。

MouseDown 事件：当用户按下某个鼠标键时发生。

MouseMove 事件：当用户在一个对象上移动鼠标时发生。

MouseUp 事件：当用户释放某个鼠标键时发生。

MouseWheel 事件：当用户在一个具有滚动轮的鼠标设备上滚动鼠标轮时发生。

Moved 事件：当对象移动到新位置时，或者以编程方式更改容器对象的 Top 或 Left 属性设置时发生。

NewObject 方法：直接从一个“.vcx”可视类库或程序将一个新类或对象添加到一个对象中。

OLECompleteDrag 事件：当数据被拖放到目标上，或取消了 OLE 拖放操作时发生。

OLEDrag 方法：开始一次 OLE 拖放操作。

OLEDragDrop 事件：当数据被拖放到目标上，并且被拖放目标的 OLEDropMode 属性设置为“1－Enabled”时发生。

OLEDragover 事件：当数据拖动到一个拖放目标上，并且拖放目标的 OLEDropMode 属性设置为“1－启用时发生该事件”。

OLEGiveFeedback：在每次 OLEDragOver 事件之后发生。允许拖动源指定 OLE 拖放操作的种类以及可视反馈。

OLESetData 事件：当拖放目标调用 GetData 方法，并且在没有指定格式的数据时，拖动源产生该事件。

OLEStartDrag 事件：当调用 OLEDrag 方法时发生。

Paint 事件：当表单或工具栏重画时发生。

Point 方法：返回一个表单上特定点的红－绿－蓝(RGB)颜色。

Print 方法：在表单对象上打印一个字符串。

PSet 方法：把一个表单或 VFP 主窗口中的一个点设置成前景色。

QueryUnload 事件：在卸载一个表单之前发生此事件。

ReadExpression 方法：返回属性窗口中某属性的表达式。仅在设计时可用。

ReadMethod：Refresh 方法：重画表单或控件，并刷新所有值，或者刷新显示某个项目。

Release 方法：从内存中释放表单集或表单。

RemoveObject 方法：运行时从容器对象中删除一个指定的对象。

ResetToDefault 方法：将属性还原成 VFP 默认的设置值，在运行和设计时可使用该方法。

Resize 事件：当调整对象大小时发生。

RightClick 事件：当用户按下并释放鼠标右键时此事件发生。

SaveAs 方法：把一个对象作为“. scx”文件保存起来。

SaveAsClass 方法：把对象的实例保存为类库中的类定义。

Scrolled 事件：在表格控件或表单中，单击水平或垂直滚动条，或移动滚动条中的滚动块时，此事件发生。

Setall 方法：为容器对象中的所有控件或某类控件指定一个属性设置。

SetViewPort 方法：设置表单的 ViewportLeft 和 ViewportTop 属性的值。

Show 方法：显示某个表单，并且确定是模式表单还是无模式表单。

ShowWhatsThis 方法：对于具有 WhatsThisHelpID 属性的对象，显示它的“这是什么”帮助主题。

TextHeight 方法：返回以当前字体显示的文本字符串高度。

TextWidth 方法：返回以当前字体显示的文本字符串宽度。

Unload 事件：在对象被释放时发生。

Valid：在对象失去焦点之前发生。

WhatsThisMode 方法：显示“这是什么”帮助的问号鼠标指针，并且启用“这是什么”帮助模式。

When：在对象得到焦点之前发生。

WriteExpression 方法：写一个表达式到一个属性。

ZOrder 方法：把指定的表单对象或控件放在其图形层内 z－order 的前面或后面；对于包含在工具栏对象中的控件，则将其放置在控件数组的前面或后面，该数组确定控件在工具栏中出现的顺序。

事件代码表示事件发生时所要执行的命令或程序，被放在相应的对象事件中，当该对象事件触发时就会执行该事件内的代码，来完成预定的功能。

编写事件代码时，需要注意两条规则：

(1)每个对象的事件触发是独立的，容器对象(如窗体、选项组等)不能处理它所包含的对象的事件。例如，在窗体上放置一个命令按钮。当点击命令按钮时，不会执行窗体的 Click 事件，而仅执行命令按钮的 Click 事件。

(2)如果某个对象事件没有相应的事件代码，则系统会逐层向上检查其父类是否有与此事件相关的事件代码，若有则执行，而该层以上的与此事件相关的代码不会被执行。若该对象有事件代码，则系统只执行它的代码，而不会再向它的上层去寻找相应的事件代码，即不会再执行其父类的事件代码。可以在该对象的事件代码中使用 Dodefault()函数，强制执行其父类的事件代码。

第 3 节 事件发生的顺序

基于事件驱动的编程技术不同于传统的过程化程序设计，程序的执行顺序不是在设计时确定的，而是取决于事件的顺序。有时一个用户动作只触发一个事件，有时在某些情况下会有多个事件接连发生，例如，在包含多个对象的表单被启动运行时，了解事件发生的顺序是很重要的，否则就无法确定该往哪个事件中写入代码。当一个对象事件发生时，可能会引起其他事件的发生，因此，理解事件发生的顺序也是非常重要的。观察事件执行顺序的简单方法是在 Debuger 工具中设置事件跟踪开关，这样，程序执行过程中的事件执行顺序就会显示在 Debugger 的事件跟踪窗口中。下面我们用一个例子来说明事件执行的顺序。在 VisualFoxpro 中新建一个窗体 Form1，加入两个文本框 Text1、Text2，两个命令按钮 Cmd1、Cmd2，TAB 键次序为 Text1、Text2、Cmd1、Cmd2，如图 1-1-1 所示，并以“test. scx”为名保存。

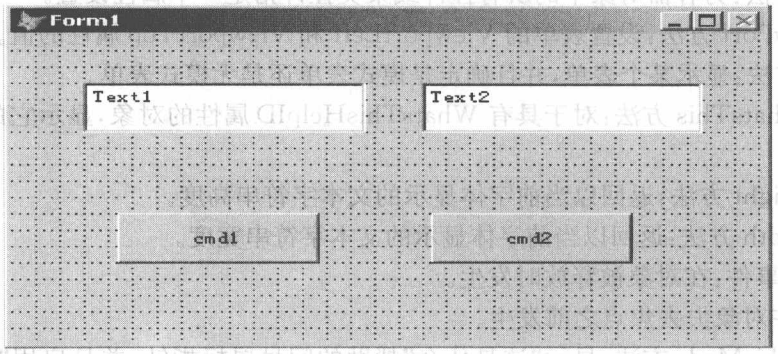


图 1-1-1 表单界面设计

单击 VFP 菜单“工具”→“调试器”命令，打开 VFP 调试器。在调试器窗口中，单击“工具”→“事件跟踪”命令，弹出“事件跟踪”对话框。如图 1-1-2 所示：

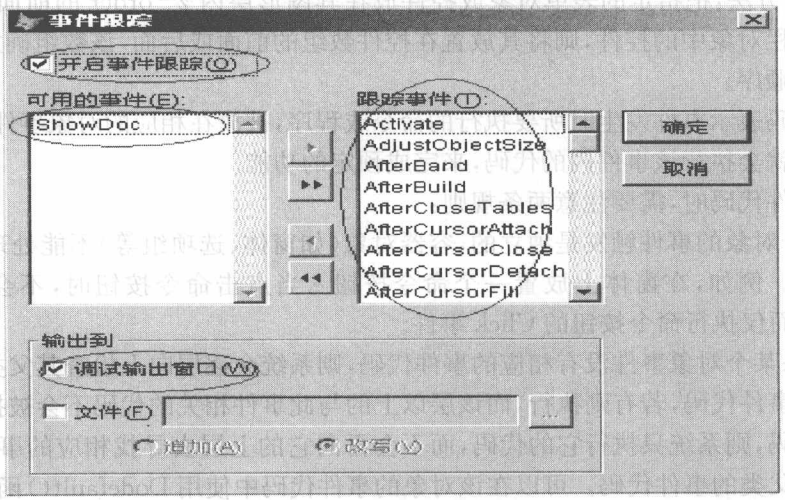


图 1-1-2 事件跟踪对话框

如图 1-1-2 所示，选择“开启事件跟踪”和“调试输出窗口”。单击“确定”按钮。如图1-1-3 所示，单击“继续”按钮。

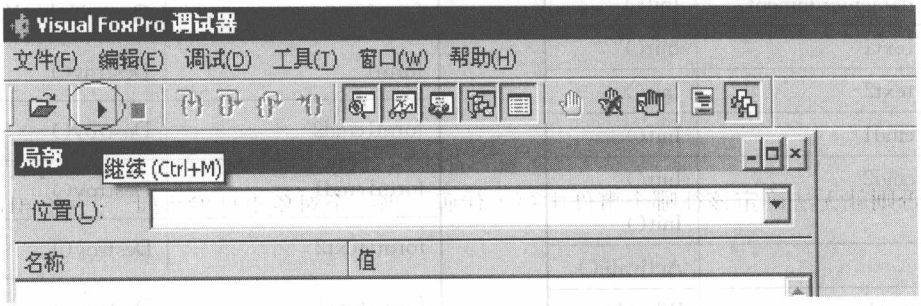


图 1-1-3　VFP 调试器对话框

弹出“运行”对话框。文件类型选择为“表单”，执行文件选择上面创建的表单 test. scx，如图 1-1-4 所示。

图 1-1-4　运行对话框

单击“运行”的按钮。查看 VFP 调试器窗口中的“调试输出”窗口，从中可以看出 test 窗体运行时，事件发生的先后顺序(见表 1-1-2)。窗体运行以后，进一步作如下测试：如果在 Text1 中输入数据，那么会交替执行 KeyPress 和 InteractiveChange 事件；如果输入完毕，按 Enter 或 TAB 键，那么执行 Text1 的 KeyPress 事件，然后执行 Valid 事件，最后执行 LostFocus 事件。

用同样的方法查看关闭窗体时事件执行的顺序。假设单击 Cmd1 关闭窗体，则事件执行的顺序见表 1-1-3：

表 1-1-2 启动窗体时事件发生顺序

对象	事件
form1. dataenvironment	Init()
form1. text1	Init()
form1. text2	Init()
form1. cmd1	Init()
form1. cmd2	Init()
form1	Init()
form1	Activate()
form1. text1	When()
form1	GotFocus()
form1. text1	GotFocus()
form1. text1	Message()
screen	Paint()
form1	Paint()

表 1-1-3 关闭窗体时事件发生顺序

对象	事件
form1	QueryUnload()
form1	Destroy()
form1cmd2	Destroy()
form1cmd1	Destroy()
form1text2	Destroy()
form1text1	Destroy()
form1	Unload()
form1dataenvironment	AfterCloseTables()
form1dataenvironment	Destroy()

第 4 节 得到和失去焦点

在应用程序开发中，窗体通常是用户对数据进行操作的界面。通常利用文本框、选择框、列表框等控件对象来显示、输入或修改数据库中的数据。有效地控制数据编辑的流程、数据对象焦点的转移，是建立用户友好的应用程序的重要部分。在窗体中，默认对象之间焦点的转移是按照所设定的 TAB 次序进行的，但多数情况下，这种默认的次序往往满足不了应用的需要。比如说，我们可能要根据用户的选择来决定焦点要转移到哪个对象，而不是按照 TAB 的次序来转移焦点。如果不理解 VFP 如何来控制对象的焦点，那么就会难以把握好处理流程。

涉及焦点的事件有四个：When、GotFocus、Valid、LostFocus；方法有一个：SetFocus。

(1)在控件接收焦点之前 When 事件发生。如果 When 事件返回“真”(. T.)，那么默认控件接收到焦点；如果返回“假”(. F.)，那么表示控件未接收到焦点。在 When 事件中编写事件代码，以控制是否让该对象得到焦点。但是，ListBox 和 ComboBox 两个控件对象对 When 事件的响应不同。在这两个控件中，每当用户单击列表中的项或用箭头键移动，使焦点在项之间移动时，都会执行 When 事件。因此，如果需要编写两个控件的 When 事件代码，需要特别注意它们的差别。

(2)失去焦点的过程与得到焦点的过程类似，当一个对象要失去焦点时会先执行 Valid 事件，如果 Valid 事件返回真(. T.)，该对象就会失去焦点，并执行 LostFocus 事件，否则该对象不会失去焦点。Valid 事件还可以返回整数值，如果返回 0，那么对象不会失去焦点；如果返回正整数值，那么焦点会转移到整数值所指定的往下第几个对象；如果返回负整数值，

那么焦点会转移到整数值所指定的往上第几个对象。因此，可以在 Valid 事件中编写事件代码来判断输入值是否有效，设定返回值“真”“假”以决定是否允许控件对象失去焦点。(3)当通过用户操作或执行程序代码使对象接收到焦点时，GotFocus 事件发生。在对象接收到焦点时，GotFocus 事件用来指定要发生的动作。例如，通过为表单中的每个控件附加 GotFocus 事件，可以显示说明或状态栏信息以指导用户；也可以通过激活、废止或显示依赖于拥有焦点控件的其他控件，提供可视化的提示；可根据用户的操作(例如单击鼠标)或在程序代码中调用 SetFocus 方法使控件接收焦点。值得注意的是，只有当对象的 Enabled 属性和 Visible 属性均设置为“真”(.T.)时，此对象才能接收焦点。要为焦点的移动定制键盘操作方式，可以为表单上的控件设置 TAB 键次序或指定访问键。在控件所在的容器 Activate 事件后，发生 GotFocus 事件。

(4)当一个对象失去焦点时，LostFocus 事件发生。这一事件发生的时间取决于对象的类型：控件由于用户的操作而失去焦点，这类操作包括选中另一个控件或在另一个控件上单击，或在代码中用 SetFocus 方法更改焦点。只有当表单不包含任何控件，或者所有控件的 Enabled 和 Visible 属性的设置均为“假”(.F.)，或者在其他表单得到焦点时，该表单失去焦点。对于表单，LostFocus 事件在 Deactivate 事件之前发生。

(5)NoDefault 命令可用于控件对象焦点的转移，这条命令可阻止 VFP 执行事件默认的行为。例如，在用文本框输入数据时，如果按 Enter 键，文本框会失去焦点，焦点会转移到下一个对象。如果不想让文本框失去焦点，那么可在文本框的 KeyPress 事件中插入一条 NoDefault 命令，按 Enter 键文本框也不会失去焦点。如果在 NoDefault 后面再加一条 SetFocus 命令，那么焦点就会转移到你需要的对象上去。

第 5 节　何时更新数据

VFP 是一个数据库应用开发环境，它数据绑定能力强大、易于使用。多数控件对象都可以通过其 ControlSource 属性与数据表的特定字段绑在一起，这样当控件对象的值发生改变时，会自动更新 ControlSource 所指定的数据字段的值。

值的更新发生在某个特定时刻。明确理解数据值更新何时发生，对于灵活运用控件对象实现预定的功能很有帮助的。在 VFP 中，控件对象更新 ControlSource 值是在 Valid 事件触发之前发生的；如果 Valid 事件没有触发，ControlSource 的值不会更新。本书前面已经解释过，Valid 事件的触发是在控件对象试图失去焦点时发生。通常在 Valid 事件中检查输入值的有效性，即使判断值无效，但由于触发了 Valid 事件，ControlSource 的值也改变了。

下面我们做一个简单的测试，观察一下数据值变化的情况。建立一个窗体，添加一个文本框。将文本框的 ControlSource 属性设置为数据表的某一个字段。假设该字段原来的值是 OLD，我们要把它改为 NEW，用 Debugger 工具跟踪数据变化过程。在文本框的 Gotfocus 事件中加入下面的代码：

```
DebugOut "Event Name: Gotfocus ControlSource: "; + Evaluate(This.ControlSource) + " value: "
+ This.value
```

同样，在文本框的 InteractiveChange、KeyPress、Valid 和 LostFocus 事件中都加上上面的代码，只是要把事件名 GotFocus 换成相应的事件名称。

打开 Debug Output 窗口，运行建立的 Form，在文本框中将显示 OLD，输入 NEW，然后按 TAB 键使文本框失去焦点，查看 Debugger 的 Output 窗口，将显示如下结果（见表 1-1-4）：

表 1-1-4 Debugger 的 Output 窗口

事件名	ControlSource 值	Value 值	说明
GotFocus	OLD	OLD	当文本框得到焦点时，ControlSource 的值和 Value 的值是相同的。
KeyPress	OLD	OLD	输入“N"。
InteractiveChange	OLD	N	执行 InteractiveChange 事件，Value 的值改变了，但 ControlSource 的值保持不变。
KeyPress	OLD	N	输入“E"。
InteractiveChange	OLD	NE	执行 InteractiveChange 事件。
KeyPress	OLD	NE	输入“W"。
InteractiveChange	OLD	NEW	执行 InteractiveChange 事件。
KeyPress	OLD	NEW	按 TAB 键。
Valid	NEW	NEW	执行 Valid 事件，ControlSource 的值变为新的输入值，数据表的值也变为新值。
LostFocus	NEW	NEW	执行 LostFocus 事件。

从表 1-1-4 可以看出在某个时间周期内 ControlSource 的值和文本框的 Value 值是不相同的。如果在这个时间内刷新文本框（即执行 Refresh 方法），则文本框的值就会恢复为 ControlSource 的当前值，这是在编写应用程序代码时需要注意的。