

1. Introduction

The purpose of this study is to investigate the effects of the proposed system on the performance of the participants. The study was conducted in a controlled environment, and the results are presented in the following sections. The study was designed to evaluate the effectiveness of the proposed system in improving the performance of the participants. The study was conducted in a controlled environment, and the results are presented in the following sections.

2. Method

The study was conducted in a controlled environment, and the results are presented in the following sections. The study was designed to evaluate the effectiveness of the proposed system in improving the performance of the participants.

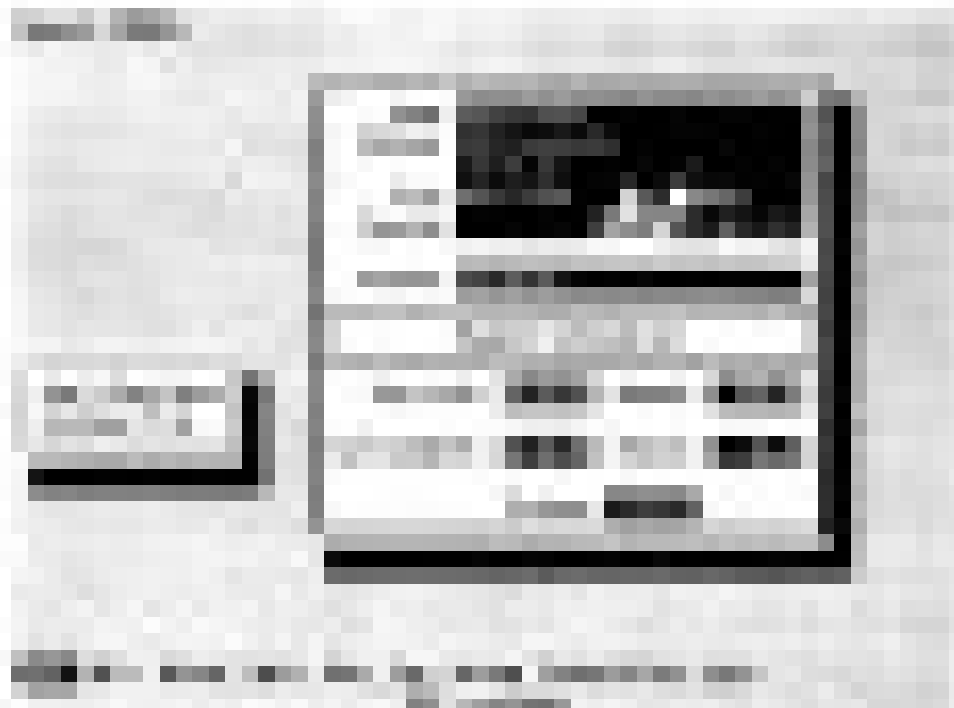


Figure 1: Screenshot of the web application interface.

第一章 应用程序设计方法

FoxPro 提供了一种全新的程序的设计方法,它可用来设计含有数百万条记录的数据库的屏幕菜单。

FoxPro 的 Screen Builder 是一个非常直观的屏幕菜单设计工具,但不是 FoxPro 中唯一的设计屏幕的方法。使用 Screen Builder 这一屏幕设计工具,比使用命令集编制屏幕菜单要简便得多。

本章中,我想向大家介绍多种设计屏幕菜单的方法。通过对各种设计方法的比较,你会明白没有一种方法是绝对最佳的。

1.1 简单模式

最简单的数据库设计模式是设计一个 Lotus 型菜单的扁平文件。我将从最简单的开始讲解,然后逐渐增加文件和菜单的复杂程度。

Record: 000001

Name	Harry Smith		
Address	2014 Annabelle Suite 22		
City	Menlo Park	CA	94025
Country		Phone	415-441-3313
Contact	Joe Smith		
Payment Information:			
Last sale:	03/12/91	Amount:	250.00
Last payment:	02/15/90	Amount:	35.00
Balance	815.25		

Joe's Furniture
Customer File

Add Edit Delete Next Prev Top Bottom Screen:Plain Quit
Add a customer

图 1-1 带简单菜单的扁平文件

带有简单菜单的扁平文件

图 1-1 所示的是一个带简单菜单的小型扁平文件,其程序代码见程序 1-1。它含有对一些色彩的设计。

程序中我使用了老版本中常用的 10 行 GET 变量 choice 程序,在 2.0 以上版本中,只要使用语句:

```
@ 24,0 GET nChoice PICTURE " * Accept; Ignore"
```

即可完成同样的功能。在程序中我保留了原方法,只是想告诉大家有多种程序编制法。

程序 1-1

```
* Program-ID.....: CUST1.PRG

SET BLINK OFF
SET TALK OFF
SET BELL OFF
SET SCOREBOARD OFF

USE Customer IN A
SCATTER MEMVAR

Version1 = .F.  && set to .F. to see improved screen

ColorTest = IsColor()
AtrBackGr = IIF ( ColorTest , [N/W*] , [W+/N,N/W] )
AtrScreen = IIF ( ColorTest , [GR+/B,N/W] , [W+/N,N/W] )
AtrData = IIF ( ColorTest , [W+/BR,W+/BR] , [NU/W,W/N] )
AtrMenu = IIF ( ColorTest , [W+/B,W+/BR] , [W+/N,N/W] )
AtrMenu2 = IIF ( ColorTest , [N/W*,W+/R] , [W+/N,N/W] )

DO DispForm

DO WHILE .T.

    DO DispRec
    choice = " "

    IF Version1
        SET COLOR TO &AtrMenu
    ELSE
        SET COLOR TO &AtrMenu2
    ENDIF

    @ 23, 0 PROMPT " Add " MESSAGE [ Add a customer ]
    @ 23, COL() PROMPT " Edit " MESSAGE [ Edit this customer ]
    @ 23, COL() PROMPT " Delete " MESSAGE [ Delete this customer ]
    @ 23, COL() PROMPT " Next " MESSAGE [ Skip to next customer ]
    @ 23, COL() PROMPT " Prev " MESSAGE [ Skip to previous customer ]
    @ 23, COL() PROMPT " Top " MESSAGE [ Skip to first customer ]
    @ 23, COL() PROMPT " Bottom " MESSAGE [ Skip to last customer ]
    @ 23, COL() PROMPT [ Screen:]+ IIF ( Version1,"Fancy " , "Plain " ) ;
    MESSAGE [ Toggle screen display ]
    @ 23, COL() PROMPT " Quit " MESSAGE [ Exit]

    MENU TO cchoice
    choice = IIF ( cchoice = 0 , [Q] , SUBSTR ( [AEDNPTBSQ] , cchoice , 1 ) )
    @ 23, 0 CLEAR
```



```

@ 3,32 SAY 'Name '
@ 4,29 SAY 'Address '
@ 6,32 SAY 'City '
@ 7,29 SAY 'Country '
@ 7,49 SAY 'Phone '
@ 9,29 SAY 'Contact '
@ 11,27 SAY PADC('Payment Information:',40)
@ 13,30 SAY 'Last sale: '
@ 13,51 SAY 'Amount: '
@ 15,27 SAY 'Last payment: '
@ 15,51 SAY 'Amount: '
@ 17,42 SAY 'Balance '

```

RETURN

PROCEDURE Setup1

```

SET COLOR TO &AtrScreen
CLEAR

```

RETURN

PROCEDURE Setup2

```

SET COLOR TO &AtrBackGr
CLEAR
SET COLOR TO N/N
@ 13, 19, 16, 20 BOX REPL(CHR(32),9)
@ 16, 1, 16, 20 BOX REPL(CHR(32),9)
@ 3, 69, 19, 70 BOX REPL(CHR(32),9)
@ 19, 26, 19, 70 BOX REPL(CHR(32),9)

```

RETURN

PROCEDURE DispRec

```

IF Version1
  SET COLOR TO &AtrScreen
ELSE
  SET COLOR TO &AtrMenu2
ENDIF

```

```

@ 0, 0 SAY "Record: "+SUBSTR(STR(RECNO()+1000000,7),2)
IF DELETED()
  @ 0,50 SAY "*DELETED*"
ELSE
  @ 0,50 SAY " "
ENDIF

```

```

SET COLOR TO &AtrData
@ 3,37 SAY CUSTOMER.Name
@ 4,37 SAY CUSTOMER.Address1
@ 5,37 SAY CUSTOMER.Address2
@ 6,37 SAY CUSTOMER.City
@ 6,52 SAY CUSTOMER.State
@ 6,57 SAY CUSTOMER.Zip
@ 7,37 SAY CUSTOMER.Country
@ 7,55 SAY CUSTOMER.Phone
@ 9,37 SAY CUSTOMER.Contact
@ 13,41 SAY CUSTOMER.Last_sale
@ 13,59 SAY CUSTOMER.Lastsalamt PICTURE '99999.99'
@ 15,41 SAY CUSTOMER.Last_pmt
@ 15,59 SAY CUSTOMER.Lastpmtamt PICTURE '99999.99'
@ 17,50 SAY CUSTOMER.Balance PICTURE '99999.99'

```

RETURN

PROCEDURE EditRec

SCATTER MEMVAR

@ 3,37 GET m.Name
@ 4,37 GET m.Address1
@ 5,37 GET m.Address2
@ 6,37 GET m.City
@ 6,52 GET m.State
@ 6,57 GET m.Zip
@ 7,37 GET m.Country
@ 7,55 GET m.Phone
@ 9,37 GET m.Contact
@ 13,41 GET m.Last_sale

@ 13,59 GET m.Lastsalamt PICTURE "99999.99"
@ 15,41 GET m.Last_pmt
@ 15,59 GET m.Lastpmtamt PICTURE "99999.99"
@ 17,50 GET m.Balance PICTURE "99999.99"

READ

IF .NOT. MOD(READKEY(),256) = 12

GATHER MEMVAR

ENDIF

RETURN

PROCEDURE BlankRec

SCATTER MEMVAR BLANK

RETURN

PROCEDURE ReplRec

GATHER MEMVAR

RETURN

PROCEDURE AddRec

DO BlankRec

@ 23,0

@ 23,0 SAY "Press (Ctrl-W) to Exit"

SET COLOR TO &AtrData

@ 3,37 GET m.Name
@ 4,37 GET m.Address1
@ 5,37 GET m.Address2
@ 6,37 GET m.City
@ 6,52 GET m.State
@ 6,57 GET m.Zip
@ 7,37 GET m.Country
@ 7,55 GET m.Phone
@ 9,37 GET m.Contact
@ 13,41 GET m.Last_sale

@ 13,59 GET m.Lastsalamt PICTURE "99999.99"
@ 15,41 GET m.Last_pmt
@ 15,59 GET m.Lastpmtamt PICTURE "99999.99"
@ 17,50 GET m.Balance PICTURE "99999.99"

READ

choice = " "

SET COLOR TO &AtrScreen

@ 24,0

@ 24,0 SAY "SELECT: (A)ccept (I)gnore";
GET choice PICTURE "!" VALID(choice \$ "AI")

```

READ
@ 24,0 CLEAR
IF choice = "A"
    APPEND BLANK
    DO ReplRec
ENDIF
RETURN

```

带下拉式、阴影菜单的扁平文件

图 1-2 所示的是一个含有特殊选项的下拉式菜单。LOCATE 对话框使用 FoxPro 中的 GETEXPR 函数,需要注意的是只有选择了 LOCATE 选项后,才能激活 CONTINUE 菜单选项,而不能单独使用 CONTINUE 选项。其代码见程序 1-2。

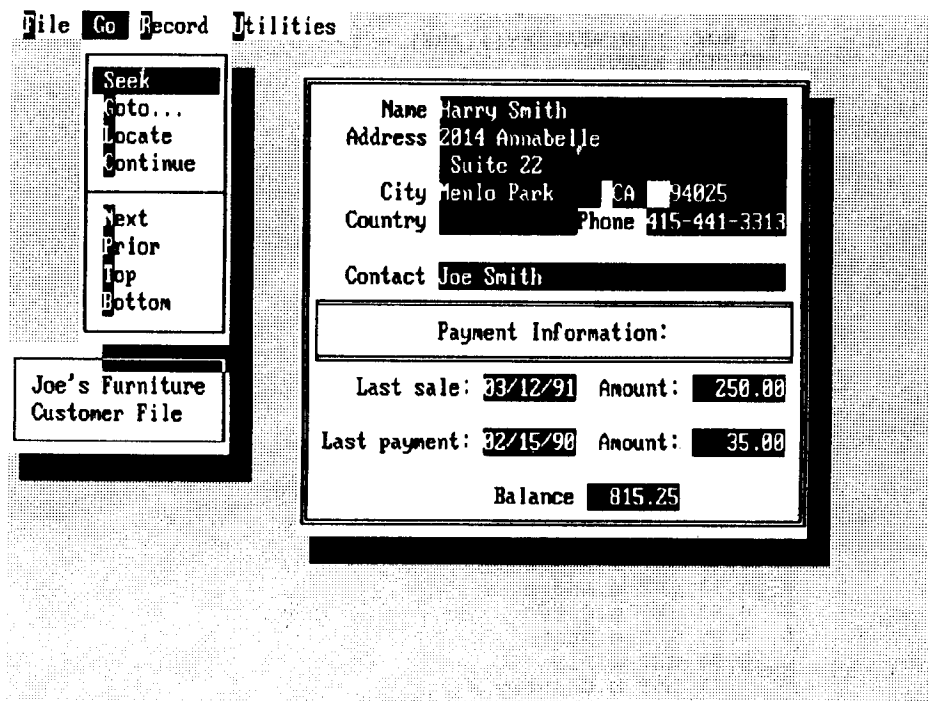


图 1-2 带 LOCATE/CONTINUE 选项的扁平文件屏幕

· 程序 1-2

```

* Program.: CUST.PRG

SET BLINK OFF
SET TALK OFF
SET BELL OFF
SET SCOREBOARD OFF

```

```
Plain      = .T.          && set to .F. to see improved screen
ColorTest = IsColor()    && set to .NOT. IsColor() to test monochrome colors
```

```
AtrBackGr = IIF ( ColorTest , [N/W*]      , [W+/N,N/W] )
AtrScreen = IIF ( ColorTest , [GR+/B,N/W] , [W+/N,N/W] )
AtrData   = IIF ( ColorTest , [W+/BR,W+/BR] , [WU/N,N/W] )
AtrMenu    = IIF ( ColorTest , [W+/B,GR+/BG] , [W+/N,N/W] )
AtrMenu2   = IIF ( ColorTest , [N/W*,W+/R]  , [W+/N,N/W] )
```

```
DEFINE WINDOW mbrowse FROM 4,15 TO 20,65 ;
    PANEL SHADOW ;
    CLOSE FLOAT GROW ZOOM ;
    COLOR &AtrData
```

```
DEFINE WINDOW alert FROM 15,15 TO 21,65 ;
    DOUBLE SHADOW ;
    COLOR &AtrMenu
```

```
DEFINE WINDOW menter FROM 8,20 to 18,60 ;
    DOUBLE SHADOW ;
    NOCLOSE NOFLOAT NOGROW NOZOOM ;
    COLOR &AtrScreen
```

```
SELECT 1
USE Customer
SCATTER MEMVAR
```

```
mdelete = ' \<Delete '
medit    = .F.
mexit    = .F.
```

```
IF Plain
    SET COLOR TO &AtrMenu
ELSE
    SET COLOR TO &AtrMenu2
ENDIF
```

```
DEFINE MENU mbartop1
DEFINE PAD mbar0 OF mbartop1 AT 0,0 PROMPT '\<File'
DEFINE PAD mbar1 OF mbartop1 AT 0,6 PROMPT '\<Go'
DEFINE PAD mbar2 OF mbartop1 AT 0,10 PROMPT '\<Record'
DEFINE PAD mbar3 OF mbartop1 AT 0,18 PROMPT '\<Utilities'
DEFINE POPUP mbar0 FROM 1,0 SHADOW
DEFINE BAR 1 OF mbar0 PROMPT '\<Help...' SKIP
DEFINE BAR 2 OF mbar0 PROMPT '\<Database...' SKIP
DEFINE BAR 3 OF mbar0 PROMPT '\<-'
DEFINE BAR 4 OF mbar0 PROMPT '\<Quit'
DEFINE POPUP mbar1 FROM 1,6 SHADOW
DEFINE BAR 1 OF mbar1 PROMPT '\<Seek' SKIP FOR RECCOUNT() = 0
DEFINE BAR 2 OF mbar1 PROMPT '\<Goto...' SKIP FOR RECCOUNT() = 0
DEFINE BAR 3 OF mbar1 PROMPT '\<Locate' SKIP FOR RECCOUNT() = 0
DEFINE BAR 4 OF mbar1 PROMPT '\<Continue' SKIP
DEFINE BAR 5 OF mbar1 PROMPT '\<-'
DEFINE BAR 6 OF mbar1 PROMPT '\<Next' SKIP FOR RECCOUNT() = 0
DEFINE BAR 7 OF mbar1 PROMPT '\<Prior' SKIP FOR RECCOUNT() = 0
DEFINE BAR 8 OF mbar1 PROMPT '\<Top' SKIP FOR RECCOUNT() = 0
DEFINE BAR 9 OF mbar1 PROMPT '\<Bottom' SKIP FOR RECCOUNT() = 0
DEFINE POPUP mbar2 FROM 1,10 SHADOW
DEFINE BAR 1 OF mbar2 PROMPT '\<Edit' SKIP FOR RECCOUNT() = 0
DEFINE BAR 2 OF mbar2 PROMPT '\<Add'
DEFINE BAR 3 OF mbar2 PROMPT '\<Browse' SKIP FOR RECCOUNT() = 0
DEFINE BAR 4 OF mbar2 PROMPT '\<-'
DEFINE BAR 5 OF mbar2 PROMPT mdelete
```

```

DEFINE POPUP mbar3 FROM 1,18 SHADOW
DEFINE BAR 1 OF mbar3 PROMPT ' \<Report... ' SKIP FOR RECCOUNT() = 0
DEFINE BAR 2 OF mbar3 PROMPT ' \<Label... ' SKIP FOR RECCOUNT() = 0
DEFINE BAR 3 OF mbar3 PROMPT ' \<Pack '
DEFINE BAR 4 OF mbar3 PROMPT ' Re\<index ' SKIP
DEFINE BAR 5 OF mbar3 PROMPT ' Pl\<ain '
DEFINE BAR 6 OF mbar3 PROMPT ' \<Fancy '

ON PAD mbar0 OF mbartop1 ACTIVATE POPUP mbar0
ON PAD mbar1 OF mbartop1 ACTIVATE POPUP mbar1
ON PAD mbar2 OF mbartop1 ACTIVATE POPUP mbar2
ON PAD mbar3 OF mbartop1 ACTIVATE POPUP mbar3
ON SELECTION POPUP mbar0 DO cu2_mnct WITH 0,BAR()
ON SELECTION POPUP mbar1 DO cu2_mnct WITH 1,BAR()
ON SELECTION POPUP mbar2 DO cu2_mnct WITH 2,BAR()
ON SELECTION POPUP mbar3 DO cu2_mnct WITH 3,BAR()
CLEAR

optpad = 'mbar0'

DO DispForm

DO WHILE .NOT. mexit

    DO DispRec

    IF Plain
        SET COLOR TO &AtrScreen
    ELSE
        SET COLOR TO &AtrBackGr
    ENDIF

    IF DELETED()
        @ 0,0 SAY ' File Go Record Utilities
Deleted **
        mdelete = ' \<Recall '
    ELSE
        @ 0,0 SAY ' File Go Record Utilities
        mdelete = ' \<Delete '
    ENDIF
    DEFINE BAR 5 OF mbar2 PROMPT mdelete

    IF Plain
        SET COLOR TO &AtrMenu
    ELSE
        SET COLOR TO &AtrMenu2
    ENDIF

    ACTIVATE MENU mbartop1 PAD &optpad

ENDDO

SET BELL ON
SET TALK ON

RETURN

PROCEDURE cu2_mnct
PARAMETER row,col

    optpad = PAD()
    mpop = 'mbar'+ LTRIM(STR(row))

    *---Hide the menu popup

```

```
HIDE MENU mbartop1
HIDE POPUP &mpop
```

```
DO CASE
```

```
    CASE row = 0 .AND. col = 4
        mexit = .T.
```

```
    CASE .NOT. (row=2 .AND. col=1) .AND. (RECCOUNT() = 0)
        DO alert WITH [ Unrecoverable error - exiting ]
        mexit = .T.
```

```
    CASE row=1
        DO Movement WITH col
```

```
    CASE row=2 .AND. col= 1
        DO EditRec
```

```
    CASE row=2 .AND. col=2
        DO AddRec
```

```
    CASE row=2 .AND. col=3
        BROWSE WINDOW mbrowse COLOR SCHEME 8
```

```
    CASE row=2 .AND. col=5
        IF DELETED()
            RECALL NEXT 1
        ELSE
            DELETE NEXT 1
        ENDIF
```

```
    CASE row=3 .AND. col=3
        choice = 1
        DO alert2 WITH choice, 'Not undoable - go ahead with pack?'
        IF choice = 1
            PACK
        ENDIF
```

```
    CASE row=3 .AND. col=5
        Plain = .T.
        DO DispForm
```

```
    CASE row=3 .AND. col=6
        Plain = .F.
        DO DispForm
```

```
ENDCASE
```

```
DEACTIVATE MENU
```

```
RETURN
```

```
PROCEDURE DispForm
```

```
    IF Plain
        DO Setup1
    ELSE
        DO Setup2
    ENDIF
```

```
    SET COLOR TO &AtrScreen
```

```
    @ 12, 0,15,18 BOX ' ' ' ' ' '
    @ 2,25,18,68 BOX ' ' ' ' ' '
    @ 10,26,12,67 BOX ' ' ' ' ' '
```

```

@ 13, 2 SAY [Joe's Furniture]
@ 14, 2 SAY 'Customer File'

@ 3,32 SAY 'Name '
@ 4,29 SAY 'Address '
@ 6,32 SAY 'City '
@ 7,29 SAY 'Country '
@ 7,49 SAY 'Phone '
@ 9,29 SAY 'Contact '
@ 11,27 SAY PADC('Payment Information:',40)
@ 13,30 SAY 'Last sale: '
@ 13,51 SAY 'Amount: '
@ 15,27 SAY 'Last payment: '
@ 15,51 SAY 'Amount: '
@ 17,42 SAY 'Balance '

```

RETURN

PROCEDURE Setup1

```

SET COLOR TO &AtrScreen
CLEAR

```

RETURN

PROCEDURE Setup2

```

SET COLOR TO &AtrBackGr
CLEAR
SET COLOR TO N/N
@ 13, 19, 16, 20 BOX REPL(CHR(32),9)
@ 16, 1, 16, 20 BOX REPL(CHR(32),9)
@ 3, 69, 19, 70 BOX REPL(CHR(32),9)
@ 19, 26, 19, 70 BOX REPL(CHR(32),9)

```

RETURN

PROCEDURE DispRec

```

IF Plain
    SET COLOR TO &AtrScreen
ELSE
    SET COLOR TO &AtrMenu2
ENDIF

@ 0, 0 SAY "Record: "+SUBSTR(STR(RECNO()+1000000,7),2)
IF DELETED()
    @ 0,50 SAY "*DELETED*"
ELSE
    @ 0,50 SAY "      "
ENDIF

SET COLOR TO &AtrData
@ 3,37 SAY A->Name
@ 4,37 SAY A->Address1
@ 5,37 SAY A->Address2
@ 6,37 SAY A->City
@ 6,52 SAY A->State
@ 6,57 SAY A->Zip
@ 7,37 SAY A->Country
@ 7,55 SAY A->Phone
@ 9,37 SAY A->Contact
@ 13,41 SAY A->Last_sale
@ 13,59 SAY A->Lastsalamt PICTURE '99999.99'

```

```

@ 15,41 SAY A->Last_pmt
@ 15,59 SAY A->Lastpmtamt PICTURE '99999.99'
@ 17,50 SAY A->Balance PICTURE '99999.99'

```

RETURN

```

PROCEDURE EditRec
  SCATTER MEMVAR
  DO Gets
  READ
  IF .NOT. MOD(READKEY(),256) = 12
    GATHER MEMVAR
  ENDIF
RETURN

```

PROCEDURE Gets

```

SET COLOR TO &AtrData
@ 3,37 GET M->Name
@ 4,37 GET M->Address1
@ 5,37 GET M->Address2
@ 6,37 GET M->City
@ 6,52 GET M->State
@ 6,57 GET M->Zip
@ 7,37 GET M->Country
@ 7,55 GET M->Phone
@ 9,37 GET M->Contact
@ 13,41 GET M->Last_sale
@ 13,59 GET M->Lastsalamt PICTURE "99999.99"
@ 15,41 GET M->Last_pmt
@ 15,59 GET M->Lastpmtamt PICTURE "99999.99"
@ 17,50 GET M->Balance PICTURE "99999.99"

```

RETURN

```

PROCEDURE BlankRec
  SCATTER MEMVAR BLANK
RETURN

```

```

PROCEDURE ReplRec
  GATHER MEMVAR
RETURN

```

```

PROCEDURE AddRec
  DO BlankRec
  @ 24,0
  @ 24,0 SAY "Press {Ctrl-W} to Exit"
  SET COLOR TO &AtrData
  DO Gets
  READ
  choice = " "
  SET COLOR TO &AtrScreen
  @ 24,0
  @ 24,0 SAY "SELECT: {A}ccept {I}gnore";
    GET choice PICTURE "!" VALID( choice $ "AI" )
  READ
  @ 24,0 CLEAR
  IF choice = "A"
    APPEND BLANK
    DO ReplRec
  ENDIF
RETURN

```

```

PROCEDURE alert
PARAMETER mess
ACTIVATE WINDOW alert
CLEAR
@ 2,0 SAY PADC(mess,WCOLS())
WAIT ''
DEACTIVATE WINDOW alert
RETURN

```

```

PROCEDURE alert2

PARAMETER answer,mess1
ACTIVATE WINDOW alert
CLEAR
@ 2,0 SAY PADC(mess1,WCOLS())
DO WHILE .T.
    @ 3,3 PROMPT '< \<Yes >'
    @ 3,34 PROMPT CHR(174)+' \<No '+CHR(175)
    MENU TO answer
    IF answer <> 0
        EXIT
    ENDIF
ENDDO
DEACTIVATE WINDOW alert

RETURN

```

```

PROCEDURE Movement
PARAMETER col
PRIVATE oldrecnum

```

```

* --- Determine Go choice
DO CASE

```

```

    * --- Goto was chosen
    CASE col=2

```

```

        ACTIVATE WINDOW menter
        DO dogoto
        DEACTIVATE WINDOW menter

```

```

    * --- Locate was chosen
    CASE col=3

```

```

        expr = []
        GETEXPR 'Create Logical Expression:' TO expr TYPE 'L;Must be a Logical
Expression' DEFAULT expr
        IF '' <> TRIM(expr)
            SaveRec = RECNO()
            LOCATE FOR &expr
            IF FOUND()
                DEFINE BAR 4 OF mbar1 PROMPT ' \<Continue '
            ELSE
                DO Alert WITH [ No match ]
                GO SaveRec
            ENDIF
        ENDIF

```

```

    * --- Continue was chosen
    CASE col=4
        DO docont

```

```

* --- Next was chosen
CASE col=6
  IF .NOT. EOF()
    SKIP
  ENDIF
  IF EOF()
    DO alert WITH [ End of data ]
    SKIP -1
  ENDIF

* --- Prior was chosen
CASE col=7
  IF .NOT. BOF()
    SKIP -1
  ELSE
    DO alert WITH [ Beginning of data ]
  ENDIF

* --- Top of file was chosen
CASE col=8
  GO TOP

* --- Bottom of file was chosen
CASE col=9
  GO BOTTOM
ENDCASE
RETURN

```

PROCEDURE docont

```

PRIVATE oldrecnum
oldrecnum = RECNO()
CONTINUE
IF EOF()
  DO alert WITH [End of locate scope]
  DEFINE BAR 4 OF mbar1 PROMPT ' \<Continue ' SKIP
  GOTO oldrecnum
  continueon = .F.
ELSE
  continueon = .T.
ENDIF
RETURN

```

PROCEDURE DoGoTo

```

where = 0
recnum = 0
okcancel = 0
getnum = .F.

* --- Define and activate Goto options

@ 1,2 SAY 'Goto: '
@ 0,10 TO 9,26
@ 1,13 PROMPT ' \<Top '
@ 3,13 PROMPT ' \<Bottom '
@ 5,13 PROMPT ' \<Record '
@ 7,13 PROMPT ' \<Skip '
MENU TO where
DO CASE

```

```

*--- Goto first record
CASE where = 1
    GO TOP

*--- Goto last record
CASE where = 2
    GO BOTTOM

*--- Goto specified record number
CASE where = 3
    recnum = 0
    @ 5,29 GET recnum PICTURE '@Z 99999999'
    READ
    DEACTIVATE WINDOW menter
    IF recnum < 1
        RETURN
    ENDIF
    IF recnum > RECCOUNT()
        DO alert WITH [ Only ] + STR(RECCOUNT(),4) + [ records in file ]
    ELSE
        GO recnum
    ENDIF

*--- Skip specified number of records
CASE where = 4
    recnum = 0
    @ 7,29 GET recnum PICTURE '@Z 99999999'
    READ
    DEACTIVATE WINDOW menter
    mrecno = RECNO()

    SKIP recnum
    IF EOF() .OR. BOF()
        DO alert WITH msg_range
        GOTO mrecno
    ENDIF
ENDCASE
RETURN

PROCEDURE Alert
PARAMETERS msg
?? CHR(7)
WAIT msg WINDOW TIMEOUT 1
RETURN

```

一般说来,这儿介绍的应用程序容易编码和使用,当然,对用户来说,它们还有许多局限性。你不容易找到光标定位在哪一字段。在进行屏幕编辑时受各种限制。基于这一原因,dBASE 的程序设计人员引入了 READ LOOP 语句。

特殊情况:读循环

在 FoxPro 2.0 版本出现以前,人们制作所喜欢的屏幕的唯一方法是使用一种叫作读循环的技术。之所以称为读循环,是因为程序不停地一个接一个地读记录。直到 FoxPro 2.0 问世,人们才越来越少地使用读循环,只是一些地方还需使用它。特别是在不使用鼠标的情况下,若要以任何方向从一个输入字段到另一字段,还需要使用读循环。你也可使用随 LASTKEY()值而变化的 _curobj,但这并非捷径。与使用 FoxPro 2.0 中的简单方法相比,比较复杂,但是,这么做是必须的。我认为这可了解历史及理解 READ CYCLE 的本质。

Superget

我们称最简单的读取循环为 Superget 循环,使用一个 DO CASE 语句来决定读什么,见程序 1-3 所包含的程序代码和图 1-3 所示的屏幕图画。只有激活的字段才以高亮度显示,这是比较合理的设计。图 1-4 所示的是弹出式情形。

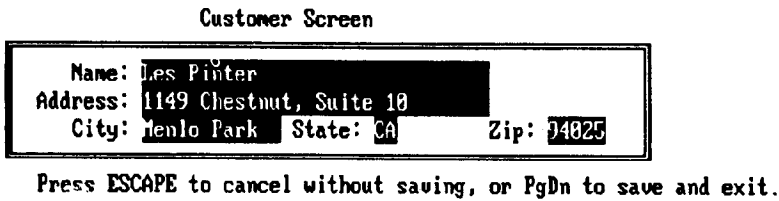


图 1-3 使用 SUPERGET 循环屏幕

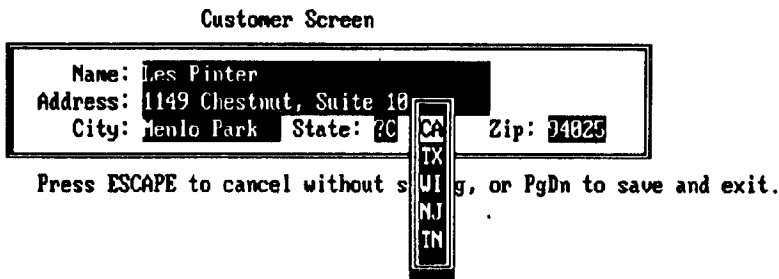


图 1-4 使用 SUPERGET 循环的弹出式情形

为了举例说明这一方法,我们建了 40 个字段,见屏幕所示。

程序 1-3

```
* Program-ID.....: SuperGet.PRG
* Purpose.....: Demonstrate CASE-based read loops for editing.

SET BLINK ON
SET TALK OFF

USE CUSTFILE

*   Structure for database: CUSTFILE.DBF
*   Field  Field Name  Type      Width
*       1  NAME        Character  30
*       2  ADDRESS      Character  30
*       3  CITY         Character  12
*       4  STATE        Character   2
*       5  ZIP          Character   5
*   ** Total **                80

SCATTER MEMVAR
```

```

TestColor = Iscolor()

AttrScreen = IIF ( TestColor , [GR+/B,W+/R] , [W+/N,N/W] )
AttrData   = IIF ( TestColor , [N/W,W+/R]   , [W/N,N/W] )

SET COLOR TO &AttrScreen
DO DispScrn

@ 11, 8 SAY 'Press ESCAPE to cancel without saving, or PgDn to save and exit.'

SET COLOR TO &AttrData
DO DispData

counter = 1

DO WHILE .T.
    DO GetVar
    READ
    DO DispMDat
    IF MOD(READKEY(),256) = 12 .OR. MOD(READKEY(),256) = 7  && ESC or PgDn
        EXIT
    ENDIF
    IF MOD(READKEY(),256) <> 4          && If not up arrow.....
        counter = counter + 1
        IF counter > 5
            counter = 1
        ENDIF
    ELSE
        counter = counter - 1
        IF counter < 1
            counter = 5
        ENDIF
    ENDIF
ENDDO

IF MOD(READKEY(),256) <> 12
    Wait [ Saving data ] WINDOW TIMEOUT 1
    GATHER MEMVAR
ELSE
    Wait [ Canceling data entry ] WINDOW TIMEOUT 1
ENDIF

DO DispData

RETURN

PROCEDURE DispScrn

CLEAR

@ 5, 22 SAY 'Customer Screen'
@ 6, 5 TO 10, 60 DOUBLE
@ 7, 7 SAY [   Name:]
@ 8, 7 SAY [ Address:]
@ 9, 7 SAY [   City:]
@ 9,27 SAY [   State:]
@ 9,42 SAY [   Zip:]

RETURN

PROCEDURE DispData

```