

Hello guys

I'm sorry but I do not speak English all is translated with Google.

Long ago in Visual Object I had developed a library to create dynamic controls. It worked so well that I wanted to repeat in HMG.

After three years of fine-tuning and adjustments I can be said to be satisfied.

I am sure that someone will be useful and again through I decided to share it.

It states that the library does not contain all the HMG controls but only the most common ones.

I was used for supporting a remarkable program.

advantages:

-is all HMG

-you can add in your programs or you can add it to your programs

-you just have to connect **KForm3.prg** into your **.hbp**

-try moving a control line change unless you do not have to recalculate the position

for more complex applications can continue to write code with HMG

The windows are considered **Main** and **Modal**.

You can use the memory variable or field names not together. (see *Dbf tests*)

Let me illustrate:

```
FV3_OpenMain('This a Main Window')
```

```
FV3_Show()
```

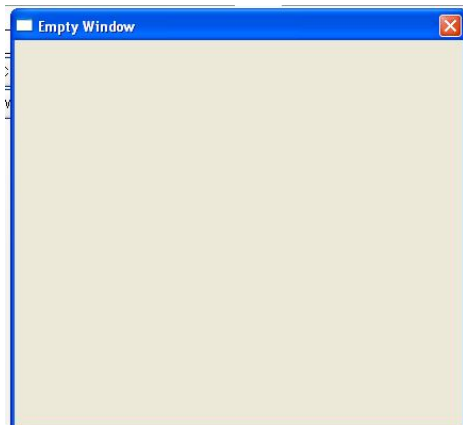
without controls window is resized by default to 400w and 300h

```
FV3_OpenModal('Empty Window')
```

```
FV3_Show()
```

```
FV3_Result() / * always indicated except for main window * /
```

Does this:



this:

```
FV3_OpenModal('Empty Window')
FV3_Show(1)
FV3_Result() / * always indicated except for main window * /
```

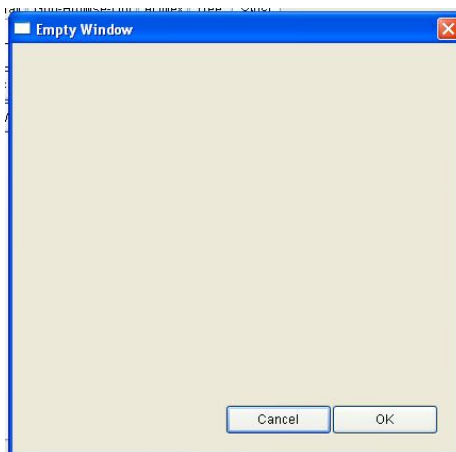
Does this:



this:

```
FV3_OpenModal('Empty Window')
FV3_Show(2)
IF FV3_Result() / * always indicated except for main window * /
    // OK button pressed
Endif
```

Does this:



Now I'll do some more examples:

```
Procedure MyInputBox(cDesc, cCaption, cText)
Local cRet: = ''
*
M -> _STRING_ : = cText
*
FV3_OpenModal(cCaption)
FV3_Label(cDesc, 250)
FV3_SkipRow(25)
FV3_TextBox('_STRING_', 250)
*
FV3_Show(2)
```

```

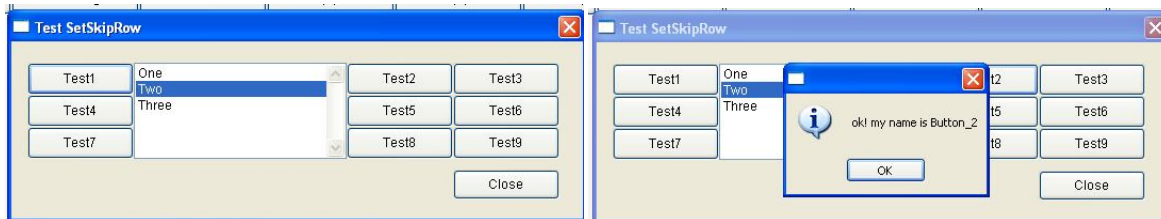
*
If FV3_Result()
    // OK button pressed
    cRet: = Trim (M -> _STRING_)
Endif
*
Return cRet

```

It produces this result:



Another example:



It Is done with:

```

Static Procedure TestSkipRow()
Local bLock := {|cWin,cControlName| MsgInfo('ok! my name is '+cControlName) }
*
FV3_OpenModal('Test SetSkipRow')
*
M->LSTBOX := 'Two'
*
FV3_Button('Test1',100,30,bLock)
FV3_SetSkipRow(.F.)
FV3_ListBox('LSTBOX',200,90,{'One','Two','Three'},.F.,bLock)
FV3_SetSkipRow(.T.)
FV3_Button('Test2' ,100,30,bLock)
FV3_Button('Test3' ,100,30,bLock)
*
FV3_SkipRow30()
FV3_Button('Test4' ,100,30,bLock)
FV3_SKipCol(200)
FV3_Button('Test5' ,100,30,bLock)
FV3_Button('Test6' ,100,30,bLock)
*
FV3_SkipRow30()
FV3_Button('Test7' ,100,30,bLock)
FV3_SKipCol(200)
FV3_Button('Test8' ,100,30,bLock)
FV3_Button('Test9' ,100,30,bLock)
*
FV3_Show(1)

```

FV3_Result()

*

Return

KFORM3 WINDOWS:

Main and Modal

Parameters:

-cTitle

-bInit

-bPostInit

-bRelease

You can add menus and configure procedure Postinit

Example:

```
cTitle      := 'Window Test'
```

```
bInit       := nil
```

```
bPostInit  := {|cWinName,cControlName| PostInit(cWinName,cControlName)}
```

```
bRelease   := nil
```

*

```
FV3_OpenModal(cTitle,bInit,bPostInit,bRelease)
```

*

Example:

```
Static Procedure PostInit(cWinName,cControlName)
Local nHeight
...
DEFINE MAIN MENU OF &cWinName
POPUP 'File'
ITEM 'Version' ACTION {|cWinName| MsgInfo('This is KForm3-version '+FV3_Version() )}
SEPARATOR
ITEM 'Exit'      ACTION ThisWindow.Release
END POPUP
END MENU
...
SetProperty(cWinName,'TEXT12','FontName','Lucida Console')
...
// if there is the menu you need to fix height
nHeight := GetProperty(cWinName,'Height')
SetProperty(cWinName,'Height',nHeight+50)
...
Return
```

important to know that

FV3_SetProperty(cControlName,cProp, uValue)	As SetProperty except window name (Window name is added to KForm2 class)
FV3_GetProperty(cControlName,cProp,uValue)	As GetProperty except window name (Window name is added to KForm2 class)
FV3_DoMethod(cControlName,cProp,uValue)	As DoMethod except window name (Window name is added to KForm2 class)
FV3_Refresh(cControlName)	As DoMethod(cWinName,cControlName,'Refresh') (Window name and refresh is added to KForm3 class)

you can use them everywhere except Procedures Init and Postinit

KFORM3 Controls: *(the parameter names are enough to understand the meaning)*

FV3_Label1 (cCaption,nWidth,nAlignment,aBackColor,cFont,nSize,aForeColor,lBold,lBorder,lClientEdge,lUnderline,lItalic)
FV3_Label2 (cField,nWidth,nAlignment,aBackColor,cFont,nSize,aForeColor,lBold,lBorder,lClientEdge,lUnderline,lItalic)
FV3_TextBox (cField,nWidth,cPict,xOpz,bEnter,bLostFocus,aBackColor,aForeColor,lBold,lItalic,bChange) xOpz := 0 normal - (1:=upper) - (2:=lower) - (3:=password)
FV3_ComboBox (cField,nWidth,nHeight,aRay,lEdit,bChange,lDisplayValue)
FV3_DatePicker (cField,nWidth)
FV3_EditBox (cField,nWidth,nHeight)
FV3_CheckBox (cField,nWidth,cCaption)
FV3_ListBox (cField,nWidth,nHeight,aRay,lMultiSelect,bEnter,bChange,lDisplayValue)
FV3_RadioGroup (cField,nWidth,aRay,lHorizontal,nSpace,aReadOnly)
FV3_RtfBox (cField,nWidth,nHeight)
FV3_Spinner (cField,nWidth,nMin,nMax,cAction)
FV3_Button (cText,nWidth,nHeight,cAction,lEnable)
FV3_Frame (cText,nWidth,nHeight)
FV3_Activex (cField,nWidth,nHeight,cProgId)
FV3_Grid (cField,nWidth,nHeight,aWidths,aHeaders,aItems,bDbClick,bChange,aJustify,aCtrl,aValid,cAlias,lCheckBox,; lEdit)
FV3_Browse (cField,nWidth,nHeight,aWidths,aHeaders,cAlias,a_Fields,bDbClick,bChange,aJustify,abHead,aBackColor,; lLines)
FV3_Image (cImage,nWidth,nHeight,lStretch)

the values returned are those of HMG with two exceptions:

ComboBox

- if the array is 2-dimensional return [n, 2]
- if lDisplayValue is .T. return DisplayValue

ListBox

- if the array is 2-dimensional back to [n, 2]
- if lDisplayValue is .T. return DisplayValue

if you have doubts after FV3_Result() checking with Valtype

Other functions:

FV3_DisableButton (cCaption)	disable Button
FV3_EnableButton (cCaption)	enable Button
FV3_Find_ListBox (cWin,cField,cSearch)	It does a search in the listbox
FV3_Get_ListBox (cWin,cField)	get array items ListBox
FV3_GetMaxCol ()	
FV3_GetMaxRow ()	
FV3_SkipRow (nSkip)	nRow += nSkip
FV3_SkipCol (nSkip)	nCol += nSkip
FV3_SetSkipRow (lValue)	Disable/Enable SkipRow
FV3_SetRow (nRow)	nRow := nRow
FV3_SetCol (nCol)	nCol := nCol
FV3_GetCol ()	
FV3_GetRow ()	
FV3_SetFont (cControl,cFont,nSize)	
FV3_GetLabelName (cCaption)	Get label name
FV3_GetButtonName (cCaption)	Get button name
FV3_Disable (cControl)	
FV3_Okay ()	as OK from codeblock
FV3_Recno (cControl,Recno)	
FV3_SetDbf (lValue)	Kform3 works with fieldname instead memvar
FV3_GetLastWindow ()	Get Name Last Window

-you do not know the name of your control and you can only find with caption (does not give the same caption in two controls)

In the demo are all examples to understand the operation and some perfect working procedures

Customizations:

in the demo there are some of my customizations

FramedText, **FramedText2** and **FramedText3** are as **Label2**

Also you can customize your functions.

Example:

```
Static Procedure GetFieldvalue(cWin,cOpz)  
Return GetProperty(cWin,cOpz, 'Item', GetProperty(cWin,cOpz, 'Value'))
```

Or:

```
Static Procedure GetFieldvalue(cField)  
Return FV3_GetProperty(cField, 'Item', FV3_GetProperty(cField, 'Value'))
```

Or:

```
Static Procedure UpdateCustomers(cWin)  
SetProperty(cWin, 'BROWSE1', 'Value', Customers->(Recno()))  
DoMethod(cWin, 'BROWSE1', 'Refresh')  
Return
```

Or:

```
Static Procedure UpdateCustomers()  
FV3_SetProperty('BROWSE1', 'Value', Customers->(Recno()))  
FV3_DoMethod('BROWSE1', 'Refresh') // or FV3_Refresh('BROWSE1')  
Return
```

Remember that you have to try to do tests with small steps until you understand how it works.