

Hooks mit Visual Extend (VFX)

Stefan Zehner

Hooks mit VFX - Eine vorbereitete Nutzungsmöglichkeit in VFX erlaubt es Hooks Formularabhängig, wie auch global einzusetzen, und damit auf leichte und überschaubare Weise zusätzlich Funktionalität mit wenig Code in die Anwendung zu bringen.

Hier sind die grundsätzlichen Einstellung, ein Codeansatz und verschiedene Beispiele aufgezeigt.

Wo sind die Hooks versteckt?

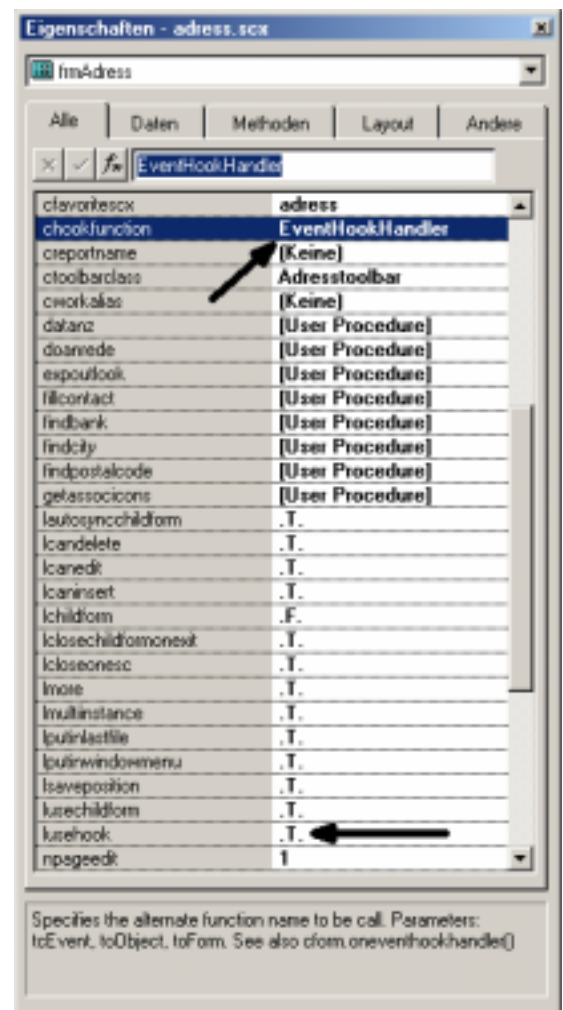
Umgesetzt – und versteckt – ist die Hook-Funktionalität in den mitgelieferten Klassen von VFX.

Die Globale Einstellung kann im Hauptprogramm (vfxmain.prg -> nEnableHook = 1) vorgenommen werden, aber es kann auch jede Form einzeln gesteuert werden:

```
ThisForm.chookfunction = "EventHookHandler"  
ThisForm.lusehook = .T.
```

...oder über das Eigenschaftsfenster.

Ebenfalls mitgeliefert wird die Programmdatei *vfxhook.prg*, welches eine schon vorbereitete Funktion *EventHookHandler* beinhaltet



Der „Kochtopf“ *EventHookHandler*

Die Funktion *Eventhookhandler* in *vfxhook.prg* dient dazu den Hook-Code aufzunehmen. Im Anfangsstadium sieht sie wie folgt aus:

[Schnipp...]

```
FUNCTION eventhookhandler(tcEvent, toObject, toForm)
LOCAL lContinue

lContinue = .T.

RETURN lContinue
ENDFUNC
```

[Schnapp...]

Und trotzdem reicht es aus, um von hieraus auf Objekte im ganzen Projekt Funktionalität zu projizieren. Dazu greifen die *Objekte* über den VFX-Klassencode und unsere Einstellungen zu. Wir brauchen lediglich zu unterscheiden, was für ein Objekt es ist, welche Aktion ausgeführt wurde, als die Hook-Funktion angesprochen wurde und zu welchem Formular das Objekt gehört. Diese Angaben bekommen wir auch schon über die Parameter mitgeliefert.

Es bietet sich also an zu erfragen, um welche Art von Objekt es sich handelt und ob wir die uns mitgeteilte Aktion unterstützen möchten. Für ganz spezielle Dinge können wir sogar hier noch wieder auf einzelne Formulare runterbrechen.

Das könnte dann ungefähr so aussehen (Beispiel):

[Schnipp...]

```
DO CASE
CASE UPPER(tcEvent)=="INIT"
    *!* Wir möchten etwas machen wenn die Init-Methode abläuft
CASE UPPER(tcEvent)=="RIGHTCLICK"
    *!* Wir möchten etwas machen, wenn ein Rechtsklick ausgeführt wurde
    IF UPPER(toObject.baseclass)=="TEXTBOX"
        *!* Wir möchten etwas machen, wenn dieser Rechtsklick
        *!* auf einer Editbox ausgeführt wurde
    ENDIF
CASE UPPER(toForm)=="FRMADRESS"
    *!* Dieser Fall gilt nur für das Formular „Adressen“
ENDCASE
```

[Schnapp...]

Sie sehen, es sind eigentlich alle erdenklichen Umstände abfangbar und alles mit allem kombinierbar, wenn es Sinn macht.

In diese Case-Struktur können wir jetzt unsere Beispiele einbauen.

Die Beispiele

Folgende Beispiele inkl. des oben verwendeten Case-Konstruktes können Sie hier finden:

- Schwarze Fordergrundfarbe in (fast) allen disabled controls
- Gridsizing verhindern (inkl. Zeilenhervorhebung)
- Ein Minirechner
- Rechtschreibprüfung

Schwarze Felder in (fast) allen "disabled" controls

Wer kennt das nicht - Im Anzeigemodus eines Formulars werden Textboxen u.s.w. mit mittelgrauer Schriftfarbe auf unbedeutend hellerer Hintergrundfarbe angezeigt. So richtig lesen kann da keiner was.

Durch folgende Hook-Prozedur werden alle Schriftfarben der Controls in sattes schwarz umgewandelt, ohne dass dieses Manuell in jedem Formular gemacht werden muss. Voraussetzung ist lediglich, dass die Anzeige des Controls die Eigenschaften für Hinter- und Vordegrundfarbe unterstützen:

[Schnipp...]


```
CASE UPPER(tcEvent)=="INIT"
```

```
IF PEMSTATUS(toObject,"disabledforecolor",5)
  !* Gibt es für das Objekt eine "disabledforecolor"?
  toObject.disabledforecolor=eval(left(rgbscheme(1,2),at(",","rgbscheme(1,2),3)-1)+")")
  !* Wir weisen der Eigenschaft unsere gewünschte Farbe (Windows Standard Text) zu
IF PEMSTATUS(toObject,"disabledbackcolor",5)
  !* Gibt es für das Objekt eine "disabledforecolor"?
  toObject.disabledbackcolor=eval("rgb("+substr(rgbscheme(1,2),at(",","rgbscheme(1,2),3)+1))")
  !* Wir weisen der Eigenschaft unsere gewünschte Farbe (Windows Standard Text) zu
ENDIF
ENDIF
```

...

[Schnapp...]

Aus der VFP-Hilfe:

 Microsoft Visual FoxPro-Sprachreferenz

PEMSTATUS()-Funktion

Gibt für eine Eigenschaft, ein Ereignis oder eine Methode ein Attribut zurück.

```
PEMSTATUS(oObjectName | oClassName, oProperty | oEvent | oMethod
| oObject, nAttribute)
```

Gridsizing verhindern:

Die Anwender, die voller Elan mit der Maus auf dem Bildschirm herumfuchteln haben schon so manch einen von uns zur Weisglut gebracht. Zum Beispiel dadurch, dass Zeilenbreiten - natürlich ungewollt auf Striche zusammengepresst wurden. Jetzt hat unser Anwender durch eine Mausektion unser schönes Grid unbrauchbar gemacht und weiß nicht mehr wie er das wieder rückgängig machen kann.

Wir sehen nur noch Striche!

Außerdem verrutscht man gerne mal in der Zeile eines Grids und gibt dem Kunden eine falsche Auskunft. Das muss nicht sein.

Damit der Anwender erst gar nichts verstecken kann und er immer genau sieht, in welcher Zeile er sich befindet helfen wir mit folgender Hook-Prozedur einfach und global, nach:

[Schnipp...]

```
CASE UPPER(tcEvent)=="INIT"  
  !* das machen wir inder INIT jeder Form.  
  IF UPPER(toObject.baseclass)=="GRID"  
    !* Bist du ein Grid?  
    toobject.allowrowsizing=.F.  
    toobject.allowheadersizing=.F.  
    toobject.highlightrowlinewidth=4  
    !* ... dann stellen wir dich mal bewegungsunfähig!  
  ENDIF
```

[Schnapp...]

Durch *toobject.allowrowsizing=.F.* wird die Höhe der Spalten und durch *toobject.allowheadersizing=.F.* wird die Spaltenkopfhöhe festgelegt, wie wir sie eingestellt haben. Um dem Anwender die Orientierung in welcher Zeile er sich befindet zu erleichtern, weisen wir *toobject.highlightrowlinewidth* den Wert 4 zu. Damit ist dann der Rahmen etwas breiter und der ausgewählte Datensatz damit mehr hervorgehoben.

Eine weitere Möglichkeit zur Hervorhebung der ausgewählten Zeile ist die farbliche Markierung, die wir mit folgendem Snipped aktivieren können:

[Schnipp...]

```
**      Spaltenmarkierung  
      FOR EACH loColumn IN toObject.columns  
        WITH loColumn  
          .dynamicbackcolor="IIF(RECNO()=this.nrecno,RGB(255,255,0),RGB(255,255,255))"  
          .setall("backcolor",RGB(255,255,0),"textbox")  
          .setall("selectedbackcolor",RGB(255,200,0),"textbox")  
          .setall("selectedforecolor",RGB(0,0,0),"textbox")  
        ENDWITH  
      ENDFOR  
**      Spaltenmarkierung – END
```

[Schnapp...]

Ein Minirechner:

Wie oft greifen wir zum Taschenrechner - oder starten den Windows eigenen Taschenrechner um 'schnell mal was zu überschlagen'.

Mit folgender Hook-Prozedur wird das Kontextmenü eines Controls mit dem Eintrag 'Berechne' erweitert. Diese Erweiterung erfolgt bei allen Controls, die eine numerische Datenbasis haben - damit wir auch Numerische Ergebnisse anzeigen können:

[Schnipp...]

```
CASE UPPER(tcEvent)=="RIGHTCLICK" AND ;
  !* War das etwa ein 'rightclick'?
  UPPER(toObject.baseclass)=="TEXTBOX" AND ;
  !* bist du eine Textbox?
  toForm.nformstatus>0
  !* und ist deine Form auch im Änderungs- bzw. Einfügemodus?
  IF VERSION(5)>=700 AND TYPE("toObject.value") $ "YN"
    !* Wir müssen sicherstellen, das VFP 7 läuft
    DEFINE BAR 4 OF shortcut PROMPT "Berechne"
    ON SELECTION BAR 4 OF shortcut berechne(_screen.ActiveForm.activecontrol)
    !* ... dann ran mit dem Menüpunkt und Funktion aufgerufen
  ENDIF
```

[Schnapp...]

Durch einen Rechtsklick auf das Control erscheint ein Kontext-Menü mit einem Eintrag 'Berechnen'. Mit diesem Eintrag wird eine Funktion gestartet, die z. B. in der Datei applfunc.prg hinterlegt sein kann:

[Schnipp...]

```
FUNCTION berechne
LPARAMETERS toObject
LOCAL lcError
lcError=ON("ERROR")
ON ERROR *
toObject.value = EVALUATE(STRTRAN(INPUTBOX("", "Berechnen", TRANSFORM(toObject.value)), ", ", ". "))
toObject.interactivechange()
ON ERROR &lcError
toObject=.NULL.
RELEASE toObject
RETURN .T.
```

[Schnapp...]

Diese Funktion öffnet die VFP-Inputbox und errechnet mit Evaluate() einfache Grundrechenaufgaben. Evaluate() gibt das Ergebnis an das mitgegebene Objekt (toObject) zurück. Dadurch gelangt der Wert dann in unser Control.

Rechtschreibprüfung:

Für Alle 'Editboxen' kann eine Rechtschreibprüfung eingebaut werden, nämlich die von MS-Word - Auch ein Grund seinem Kunden Word zu verkaufen :-).

Mit folgender Hook-Prozedur wird im Kontextmenü der Editbox der Eintrag 'Rechtschreibprüfung' angefügt:

[Schnipp...]

```
CASE UPPER(tcEvent)="RIGHTCLICK" AND
  UPPER(toObject.baseclass)="EDITBOX" AND toForm.nformstatus>0
  *!* Bist du eine Edit-Box?
  DEFINE BAR 4 OF shortcut PROMPT "Rechtschreibprüfung"
  *!* Definition des neuen Eintrags im Kontextmenue
  ON SELECTION BAR 4 OF shortcut spellchk(_screen.ActiveForm,_screen.ActiveForm.activecontrol)
  *!* Aufruf der Funktion Spellcheck
```

[Schnapp...]

Mit diesem Menüeintrag wird die Funktion 'spellchk()' aufgerufen, die ihrerseits die Word-Rechtschreibprüfung benutzt.

Diese Funktion kann z. B. im applfunc.prg untergebracht sein:

[Schnipp...]

```
FUNCTION spellchk
LPARAMETERS toForm,toObject
PUBLIC goSpellchk
toform.addobject("oWordlabel","wordlabel")
toform.oWordlabel.top=_screen.top
toform.oWordlabel.width=_screen.Width
toform.oWordlabel.left=_screen.left
toform.oWordlabel.height=_screen.height
toform.oWordlabel.visible=.t.
toForm.addobject('acargo[1,1]','olecontrol','word.document')
toForm.acargo[1,1].Height = toObject.height
toForm.acargo[1,1].Width = toObject.width
toForm.acargo[1,1].Top = toObject.top+25
toForm.acargo[1,1].Left = toObject.left
toForm.acargo[1,1].DoVerb(0)
toForm.acargo[1,1].object.content = toObject.Value
toForm.acargo[1,1].object.checkspelling()
goSpellchk=toObject
RETURN .T.
```

[Schnapp...]

Mit den Parametern 'toForm' und 'toObject' hält die Funktion zur Editbox Kontakt. Es wird auf unserer Form (toForm) ein Word-Object erstellt, (möglichst) genau auf unserer Editbox plaziert und die Größe angeglichen. Text aus unserer Editbox (toObject) wird in das Word-Objekt kopiert und die Funktion 'checkspelling()' aus Word gestartet.

Das Wordobjekt selbst wird aus der Klasse >Wordlabel< referenziert, welches wir zuvor auf Basis der Klasse >label< z. B. in der Bibliothek >appl< abgelegt haben. Hier ist in der Click-Methode folgender Code einzugeben:

[Schnipp...]

```
thisform.acargo[1,1].object.select()
thisform.acargo[1,1].object.content.copy()
goSpellchk.Value = _cliptext
goSpellchk.Refresh()
goSpellchk=.NULL.
RELEASE goSpellchk
thisform.acargo[1,1]=.NULL.
RELEASE this
```

[Schnapp...]

Alle Beispiele stammen aus der Feder von Uwe Habermann (Uwe@Habermann-leu.de) und wurden veröffentlicht auf www.my-vfx.de. Hier steht auch der Download von Beispielcode bereit.

Stefan Zehner ist Inhaber und Leiter der Firma Zehner Software & Beratung. Sie erreichen ihn unter zehner@web.de oder über www.my-vfx.de.