

Für Einsteiger

Link



Burkhard Quenzler

(Jg. 1957) Fernstudium, 1990-2014 Administrator IT, seit 2014 Datenbankprogrammierer. Ab 1993 nebenbei für Datenbanken wie dBase, MS SQL und Access, seit FileMaker 7 auch als FileMaker Entwickler tätig, hat viele kleinere und größere Projekte mit FileMaker realisiert.

info@fmqu.de

Flexible FileMaker Vorlage

Teil 1

Oder: Aller Anfang ist schwer



Sind Sie es auch leid, jedes Mal alle notwendigen Start- und Ende- und sonstige Scripts, die für wiederkehrende Aufgaben entwickelt wurden, in eine neue FileMaker Datei zu integrieren?

Wenn man programmiert, muss man sich jedes Mal wieder die gleichen Gedanken machen: Was brauche ich in meiner FileMaker Datei, welches Ergebnis will ich damit erreichen und was soll die Datenbank alles beinhalten? Zu diesem Zweck habe ich mir eine FileMaker Datei erstellt, die den Anfang all meiner Projekte darstellt. Sie hat ein funktionierendes Startfenster sowie ein Logbuch, in dem die Änderungen protokolliert werden, die der Anwender in der Datei macht. Außerdem beinhaltet sie meine im vergangenen Jahr vorgestellte Codeversionierung, allerdings in einer überarbeiteten Form. Somit fange ich nicht jedes Mal bei Null an, sondern habe eine solide Basis für jede zu erstellende FileMaker Anwendung.

Im Folgenden möchte ich Ihnen meine FileMaker Vorlage etwas genauer vorstellen, sodass auch Sie von ihrem Einsatz profitieren können. Dafür gliedere ich das Ganze in drei Teile:

1. Die Datei und ihr Aufbau
2. Das Logbuch und die Scripts
3. Die Codeversionierung (neue Version)

Jeder dieser Teile bekommt seinen eigenen Beitrag in meiner kleinen Artikelserie zu diesem Thema. In dieser Ausgabe geht

es mit dem Aufbau der Datei los. Die FileMaker Vorlage liefere ich natürlich gleich mit, Sie können sie sich einfach von der Website des FileMaker Magazins downloaden.

Die Dateivorlage

In der Vorlage sind die Scripts für „Autostart“ und „Autoclose“ integriert, die alle notwendigen Befehle für den Start und das Schließen der Datei beinhalten. Das sind die Voraussetzungen dafür, die Codeversionierung ordnungsgemäß durchführen zu können.

Startfenster und Globaleinstellungen

Haben Sie die Datei gestartet, wird das Startbild eingeblendet. Die Position des mittleren Fensters ändert sich nicht – egal wie groß Sie das FileMaker Fenster aufziehen, das Bild bleibt immer genau in der Mitte. Farblich können Sie das Fenster nach eigenen Vorstellungen anpassen, das dürfte Ihnen als FileMaker Entwickler keinerlei Probleme bereiten.



Im Layout „Global“ können Sie auch alle anderen Eintragungen des Startfensters ändern, wie zum Beispiel das Logo. Achten Sie aber darauf, dass dieses Layout über die Einstellungen keinem Nutzer zugänglich gemacht werden sollte. Es steht also nur Ihnen als Admin bzw. Entwickler zur Verfügung.

Alle Felder, die mit „_g“ enden, sind globale Felder. Experimentieren Sie ruhig ein wenig mit den Eintragungen herum.

Die Endung „_kl_g“ kennzeichnet Felder in Kleinschrift, das wird für das von mir vorgegebene Layout so benötigt.

Haben Sie die Felder entsprechend Ihrer Vorstellung geändert, ist das erste Formular bereits fertig, ohne dass Sie eine Zeile programmieren mussten.

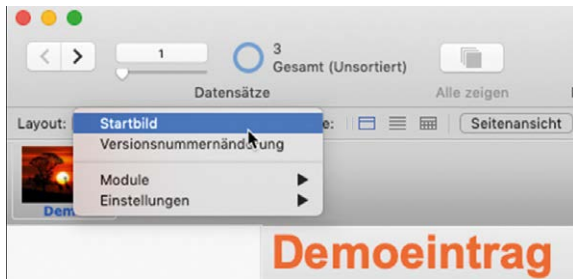
Menüstruktur

Im zweiten Schritt geht es um den Aufbau der Menüstruktur. Auch hier ist schon alles vorbereitet, wie Sie im Layout „Einstellungen“ sehen können. Das erste Menü habe ich schon mal ausgefüllt, die anderen Einträge können Sie nach Ihren Wünschen anpassen. Weisen Sie dem Menü ein Bild zu, indem Sie eine beliebige Bilddatei auf das erste Feld ziehen. In diesem Moment wird Ihre neues Menü in der darüberstehenden Symbolleiste angezeigt. Der Inhalt des Feldes „Menüeintrag“ erscheint unterhalb des Bildes. Mit einem Klick auf die Taste am Ende der Zeile (siehe Abbildung nächste Seite oben die Ziffer 4) wird das Formular geöffnet.

Copyright_g	© 2017 - 2019 by BeQuSoft	Hintergrundbild Startfenster. 480 x 280 Pixel	
Land_g	0		
Lizenzierte Firma_g	bequsoft		
LizenzUser_g	Burkhard Quenzler		
Lizenznummer_g	X579-3456-2954-1553-2V79		
Autor_g	Burkhard Quenzler		
Autorkl_g	burkhard quenzler		
Progname_g	FM-Vorlage		
Prognakl_g	fm-vorlage		
Prognamekurz_g	BTK		
Programmbezeichnung_g	BeQuSoft		
Sprache_g	0		
MJR	0		
MNR	00		
RVR	001		
BNR	0009		
VN	0.00.001.0009	0.00.001.0009	
VNK	0.00.001		
Aenderungsstempel	15.01.2019		
Aenderungsbeschreibung	Zurücksetzen der Versionsnummer		
Wunsch_g			
Ordnerpfad_g	file:/BQ HD 10 Ablage/0 Aktuelles/Aktuelle BeQuProjekte/Codeversionierung/Codeversionierung II/Druckvorlage/FM-Vorlage.fmp12		
Zaehler_g	118		
Verbinder_g	1		
BackupZaehler_g	6		
Backupanzahl_g	10		
BisBackup_b	4		
Abbruch_g			
Obenabstand_g	30		
Linksabstand_g	202		
Fensterhöhe_g	934		
Fensterbreite_g	1408		
Monitorbreite_g	1920		
Monitorhöhe_g	1080		
Autorsynonym_g	BeQuSoft		
Autorsykl_g	bequsoft		
ZaehlerInfo_G	5		
Logo_g			



Ist kein Bild für den Menüeintrag vorhanden, wird dieser ausgeblendet und kann im Programm nicht verwendet werden.



Layouts

Grundsätzlich können Sie jedes Layout Ihren Wünschen entsprechend anpassen, doch es gibt einige Ausnahmen, die für bestimmte Vorgänge benötigt werden. Die vorhandenen Layouts haben folgende Bedeutung:

Startbild – das Startbild des Programmes.

Versionsnummernänderung – dieses Layout wird für die Versionsnummernänderung benötigt, bitte so lassen (wird im 3. Teil erklärt).

Module 1 bis 10 – Layoutfenster der Menümodule. Diese können Sie vollkommen nach Ihren Wünschen gestalten und auch die Scripts anpassen.

Einstellungen – Layoutfenster für die Hilfe, Einstellungen und für das Logbuch, das im 2. Teil der Serie erklärt wird.

Scripts

Kommen wir nun zu den erforderlichen Scripts.

Autostart

- ◆ **Fenstertitel setzen**
[Aktuelles Fenster ; Neuer Titel: Global::Programe_g & " - Lizenznehmer: " & Global::Lizenzuser_g & " - " & Global::Lizenzfirma_g]
- ◆ **AnwenderAbbruchZulassen setzen**
[Aus]
- ◆ **Werkzeugleisten ein-/ausblenden**
[Ausblenden]
- ◆ **Script ausführen**
[Angegeben: Aus Liste ; "Variablen setzen" ; Parameter:]

- ◆ **Script ausführen**
[Angegeben: Aus Liste ; "Startfenster" ; Parameter:]
- ◆ **Variable setzen**
[\$\$FensterVar ; Wert: ""]
- ◆ **Wenn**
[Global::Zaehler_g > 50]
- ◆ **Scriptpause setzen**
[Dauer (Sekunden): 4]
- ◆ **Sonst, wenn**
[Global::Zaehler_g > 30]
- ◆ **Scriptpause setzen**
[Dauer (Sekunden): 6]
- ◆ **Sonst, wenn**
[Global::Zaehler_g > 20]
- ◆ **Scriptpause setzen**
[Dauer (Sekunden): 8]
- ◆ **Sonst, wenn**
[Global::Zaehler_g > 10]
- ◆ **Scriptpause setzen**
[Dauer (Sekunden): 10]
- ◆ **Ende (wenn)**
- ◆ **Script ausführen**
[Angegeben: Aus Liste ; "Autostart2" ; Parameter:]

Dieses Script setzt alle Variablen und Fensterbezeichnungen, die für den Start der Datei benötigt werden. Ist das geschehen, wird das Startfenster geöffnet. Hier habe ich eine kleine Besonderheit eingebaut: Wurde das Programm mehr als 50 mal aufgerufen, wird das Fenster nur noch vier Sekunden angezeigt, während Sie es bei einer Aufrufhäufigkeit bis zu 30 Mal sechs Sekunden lang sehen, bei den ersten Starts sind es sogar zehn Sekunden.

Ist dieses Script durchgelaufen, kommen wir zu dem zweiten Teil des Startscripts, dem Script „Autostart2“, in dem das Menü gesetzt und geprüft wird, ob die richtige FileMaker Version vorhanden ist:

Autostart2

- ◆ **Menüset installieren**
["[Dateistandard]"]
- ◆ **Variable setzen**
[\$fmVersion ; Wert: Hole (ProgrammVersion)]
- ◆ **Variable setzen**
[\$Auswertung ; Wert: LiesAlsZahl (ZeichenLinks (WörterRechts (\$fmVersion; 1) ; 2))]
- ◆ **# Fenster fixieren**
- ◆ **Wenn**
[\$Auswertung < 17 UND Global::ZaehlerInfo_G < 5]
- ◆ **Fehlerton**
- ◆ **Feldwert setzen**
[Global::ZaehlerInfo_G ; Global::ZaehlerInfo_G + 1]
- ◆ **Eigenes Dialogfeld anzeigen**

```
[ "Nicht passende Programmversion!" ; "Diese Datei wurde
für FileMaker ab Version 17 erstellt. Die Mehrzahl der
Programmteile wird nicht funktionieren? Trotzdem weiter?" ]
◆ Feldwert setzen
[ Global::ZaehlerInfo_G ; Global::ZaehlerInfo_G + 1 ]
◆ Wenn
[ Hole ( LetzteMeldungswahl ) = 1 ]
◆ Datei schließen
[ Aktuelle Datei ]
◆ Alle Scripts abbrechen
◆ Ende (wenn)
◆ Ende (wenn)
◆ #
◆ Wenn
[ Hole ( Kontoname ) = "FMAdmin" ODER Hole(Kontoname)
= "Developer" ]
◆ Gehe zu Layout
[ "Logbuch" (LOGBUCH) ; Animation: Ohne ]
◆ Gehe zu Datens./Abfrage/Seite
[ Erste(r) ]
◆ Variable setzen
[ $offen ; Wert: 0 ]
◆ Variable setzen
[ $i ; Wert: 0 ]
◆ Schleife (Anfang)
◆ Wenn
[ LOGBUCH::erledigt = "" ]
◆ Variable setzen
[ $offen ; Wert: $offen+1 ]
◆ Variable setzen
[ $i ; Wert: $i+1 ]
◆ Ende (wenn)
◆ Gehe zu Datens./Abfrage/Seite
[ Nächste(r) ; Nach letztem beenden: Ein ]
◆ Schleife (Ende)
◆ Wenn
[ $offen ≠ 0 ]
◆ # Wird erst im zweiten Teil benötigt!
◆ Ende (wenn)
◆ Feldwert setzen
[ Global::Wunsch_g ; "" ]
◆ Feldwert setzen
[ Global::Abbruch_g ; "" ]
◆ Sonst
◆ AnwenderAbbruchZulassen setzen
[ Aus ]
◆ Fehleraufzeichnung setzen
[ Ein ]
◆ Werkzeugleisten ein-/ausblenden
[ Ausblenden ]
◆ Formatierungsleiste zulassen
[ Aus ]
◆ Menüset installieren
[ "[Dateistandard]" ]
◆ Ende (wenn)
◆ Variable setzen
[ $$System ; Wert: Hole(SystemPlattform) ]
◆ Feldwert setzen
[ Global::Zaehler_g ; Global::Zaehler_g + 1 ]
◆ Feldwert setzen
[ Global::Ordnerpfad_g ; Hole(DateiPfad) ]
◆ Wenn
[ Hole(LetzteMeldungswahl)=2 ]
```

```
◆ Wenn
[ $i > 500 ]
◆ Eigenes Dialogfeld anzeigen
[ "Aufträge" ; "Es waren weit über 500 Aufträge in der Datei
vermerkt. Bitte löschen Sie die Aufträge, die Sie nicht
mehr benötigen." ]
◆ Ende (wenn)
◆ Aktuelles Script verlassen
[ Textergebnis: ]
◆ Ende (wenn)
◆ Fenster aktualisieren
◆ #
◆ Gehe zu Layout
[ "Module 1" (Meneul) ; Animation: Ohne ]
◆ Script ausführen
[ Angegeben: Aus Liste ; "Fenstergröße gesetzt" ;
Parameter: ]
◆ Gehe zu Datens./Abfrage/Seite
[ Erste(r) ]
```

Zudem schaut das Script, wer sich angemeldet hat. Handelt es sich um den FMAdmin bzw. Developer (Entwickler) müssen anschließend Prüfungen durchgeführt werden, die ebenfalls erst im zweiten Teil beschreiben werden.

Autoclose

```
◆ AnwenderAbbruchZulassen setzen
[ Aus ]
◆ Fehleraufzeichnung setzen
[ Ein ]
◆ #
◆ Variable setzen
[ $error ; Wert: 0 ]
◆ Variable setzen
[ $param ; Wert: Hole ( ScriptParameter ) ]
◆ #
◆ Script ausführen
[ Angegeben: Aus Liste ; "Backup nach Zähler" ;
Parameter: ]
◆ #
◆ Wenn
[ (Hole ( Kontoname ) = "FMAdmin" ODER Hole ( Kontoname )
= "Developer") UND $$Versionierung ≠ 1 ]
◆ Variable setzen
[ $fmVersion ; Wert: Hole ( ProgrammVersion ) ]
◆ Variable setzen
[ $Auswertung ; Wert: LiesAlsZahl ( ZeichenLinks (
WörterRechts ( $fmVersion; 1 ) ; 2 ) ) ]
◆ Eigenes Dialogfeld anzeigen
[ "Versionsverwaltung" ; "Haben Sie an der aktuell
vorhandenen Programmierung etwas verändert?" ]
◆ Wenn
[ Hole ( LetzteMeldungswahl ) = 2 ]
◆ Variable setzen
[ $$Versionierung ; Wert: 1 ]
◆ Wenn
[ $Auswertung > 15 ]
◆ Fenster fixieren
◆ Gehe zu Layout
[ "Einstellungen 12" (Global) ; Animation: Ohne ]
◆ Neues Fenster
[ Stil: Karte ; Name: "Änderung der Versionsnummer" ;
Mit Layout: "Versionsnummernänderung" (Global) ]
◆ Alle Scripts abbrechen
```

```

◆ Sonst
◆ Eigenes Dialogfeld anzeigen
  [ "Falsche Version von FileMaker" ; "Die Versionierung
    kann erst ab Version 16 erfolgen. " ]
◆ Ende (wenn)
◆ Ende (wenn)
◆ Ende (wenn)
◆ #
◆ Variable setzen
  [ $$FensterVar ; Wert: "Das Programm wird geschlossen ..." ]
◆ #
◆ Script ausführen
  [ Angegeben: Aus Liste ; "Startfenster" ; Parameter: ]
◆ Scriptpause setzen
  [ Dauer (Sekunden): 2 ]
◆ Programm beenden
  
```

Dieses Script wird vom Programm geladen, um alle Prozesse ordnungsgemäß abzuschließen, bevor das letzte Programmfenster geschlossen wird. Hier passiert das, was sich viele Programmierer wünschen: die Versionierung. Es wird kontrolliert, ob die angemeldete Person „FMADMIN“ oder „DEVELOPER“ ist. Ist das nicht der Fall, wird das Programm geschlossen. Handelt es sich aber um eine der oben genannten Personen, geht das Script in die Prüfschleifen. Mehr dazu erfahren Sie im zweiten und dritten Teil der Beitragsserie.

Passwörter setzen

Lassen Sie sich die Zeit nicht zu lang werden, fangen Sie ruhig schon mal an die Vorlage auszuprobieren. Passwörter sind noch keine gesetzt, diese müssen Sie aber zwingend vergeben, um alle Funktionen richtig testen zu können.

Folgende Passwörter sind erforderlich:

- „FMAdmin“ – Datenbankadministrator
- „Developer“ – Entwickler
- „Hauptuser“ – Hauptuser des Programmes mit besonderen Rechten
- „User...“ – User

Nun kann es losgehen. Bleiben Sie gespannt auf die Fortsetzung, es sind ja nur ein paar Wochen bis zur nächsten Ausgabe. ■

Anzeige

**Möglichkeiten entdecken.
Inspirationen finden.
Neue Wege gehen.**

 **Arnold Kegebein**
IT Consultant
+49 (0)40 8537 2010 arnold@kegebein.net