



Erstellen einer einfachen Anwendung mit FileMaker

Teil 4

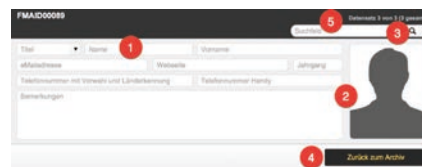
Ein Volltextsucharchiv für das FileMaker Magazin und Zeitschriften im PDF-Format

Der letzte Teil dieser Serie endete mit Überlegungen zum Autorenarchiv, das zur Auswahl von Autoren benötigt wird. Damit das Ganze nicht zu umfangreich und pflegeintensiv wird, sollten wir uns auf die notwendigsten Daten beschränken: Name, Vorname, Kontaktdaten, Bild und Titel sind allgemeine Informationen, die für unser Vorhaben vollkommen ausreichen.

Als Erstes kümmern wir uns um das Layout „FMM_Autor“, das von FileMaker beim Erstellen der gleichnamigen Tabelle angelegt wurde. Dieses Layout wollen wir ein bisschen umgestalten.



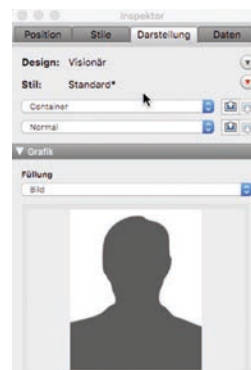
Dabei gibt es einiges zu beachten, worauf ich nachfolgend etwas näher eingehen möchte. Die Nummerierung im Text entspricht den Markierungen in der nachfolgenden Abbildung.



1. Seit Version 12 kann man Felder so definieren, dass ein bestimmter Text eingeblendet wird, wenn das Feld leer ist. Diese platzsparende Funktion mache ich mir zunutze, indem ich die Feldnamen in den Feldern selbst anzeigen lasse. Bitte gehen Sie ebenfalls so vor und fügen Sie für jedes Feld einen selbsterklärenden Platzhaltertext hinzu.

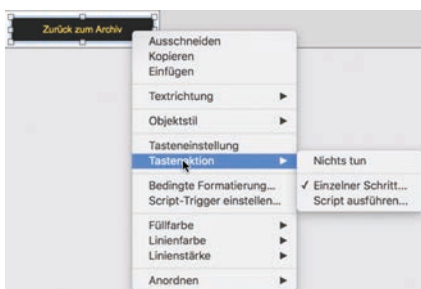


2. Bei einem Bild könnten Sie natürlich auch einfach einen Text verwenden, z. B. „Bild einfügen“. Aber es gibt einen weitaus eleganteren Weg: Gehen Sie im Inspektor auf das Register „Darstellung“ und fügen Sie als Hintergrund für den Container ein selbst gestaltetes Bild ein.

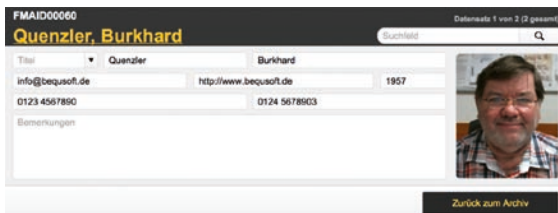


- Richten Sie ein Suchfeld nach demselben Prinzip ein, wie wir es in Teil 2 gemacht haben. Der einzige Unterschied besteht darin, dass die Suche nicht sofort bei einer Änderung des Feldinhaltes ausgelöst wird, sondern erst beim Anklicken der Lupen-Taste. Wenn Ihnen die automatische Suche beim Verlassen des Feldes besser gefällt, ist es Ihnen freigestellt, diese hier zu verwenden.
- Die Taste „Zurück zum Archiv“ dient lediglich dazu, das Fenster wieder zu schließen. Dabei ist es irrelevant, ob der Benutzer in einem Bezugsdatensatz steht oder nicht, so dass die folgende, einfache Tastenaktion genügt:

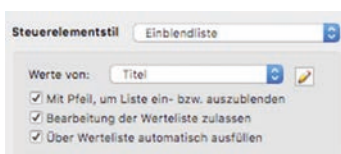
◆ Fenster schließen (Aktuelles Fenster)



- Zur Anzeige der vorhandenen Datensätze muss ich nicht mehr viel sagen, da wir diese Funktion schon im ersten Teil dieser Serie verwendet haben.



Im Blätternmodus sollte das Ganze nach Anlegen eines Datensatzes so aussehen, wie in der obigen Abbildung. Die Eingabe eines Titels erfolgt über eine Werteliste mit Spezialwerten, die jederzeit angepasst werden kann.



Kehren wir nun zum Layout „FMMA“ zurück, wo eine Verknüpfung zu „FMM_Autor“ hergestellt werden soll. Bei einem Klick auf das Autorenfoto soll der dazugehörige Autoren datensatz in einem neuen Fenster angezeigt werden. Ist noch kein Autor vorhanden, soll ein neuer Datensatz angelegt werden. Um zu dem gewünschten Bezugsdatensatz zu springen, benötigen wir das Script „FMM_Autoren“:

```

◆ # -----
◆ # © 2013 by Burkhard Quenzler, BeQuSoft
◆ # Script-Parameter: keine
◆ # Allgemeines Startscript ohne Parameter
◆ # Version: 1.01
◆ # -----
◆ Wenn
  FMM_INHALT::FS_FMAID = ""
◆ Neues Fenster
  Stil: Schwebendes Dokument ; Name: "Autorenverzeichnis" ;
  Höhe: 400 ; Breite: 745 ; Oben: 250 ; Links: 500
◆ Gehe zu Layout
  "FMM_AUTOR" (FMM_AUTOR)
◆ Neuer Datensatz/Abfrage
◆ Gehe zu Feld
  FMM_AUTOR::Titel
◆ Sonst
◆ Gehe zu Bezugsdatensatz
  Aus Tabelle: "FMM_AUTOR" ; Mit Layout:
  "FMM_AUTOR" (FMM_AUTOR) ; Neues Fenster
◆ Ende (wenn)
  
```

Klicken Sie das Bild mit der rechten Maustaste an, wählen Sie den Menüeintrag „Tasteneinstellung“ aus und weisen Sie der Aktion „Script ausführen“ das Script zu.

Damit bei einem leeren Autoren bild ein neuer Datensatz angelegt wird, habe ich eine Wenn-Bedingung eingefügt:

```

◆ Wenn
  FMM_INHALT::FS_FMAID = ""
◆ Neues Fenster
  Stil: Schwebendes Dokument ; Name: "Autorenverzeichnis" ;
  Höhe: 400 ; Breite: 745 ; Oben: 250 ; Links: 500
  
```

Wenn das Feld **FMM_INHALT::FS_FMAID** leer ist, wird ein Fenster mit den angegebenen Informationen (Name, Größe) geöffnet.

```

◆ Gehe zu Layout
  "FMM_AUTOR" (FMM_AUTOR)
◆ Neuer Datensatz/Abfrage
◆ Gehe zu Feld
  FMM_AUTOR::Titel
  
```

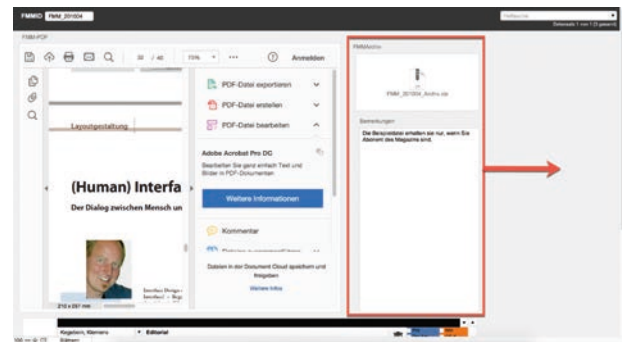
In diesem neuen Fenster wird das Layout „FMM_Autor“ angezeigt und ein leerer Datensatz angelegt. Der Cursor wird ins Feld **Titel** gesetzt.

```

◆ Sonst
◆ Gehe zu Bezugsdatensatz
  Aus Tabelle: "FMM_AUTOR" ; Mit Layout: "FMM_AUTOR";
  FMM_AUTOR) ; Neues Fenster
  
```

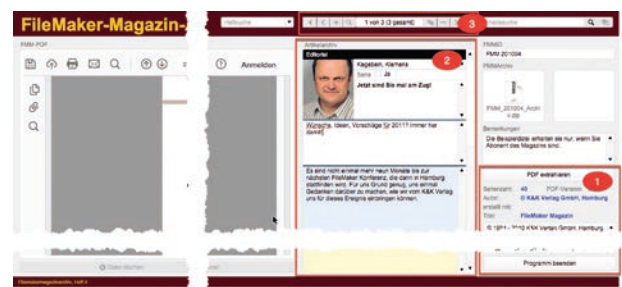
Ist das Feld `FMM_INHALT::FS_FMAID` nicht leer, soll der zugehörige Datensatz zum Editieren aufgerufen werden.

◆ Ende (wenn)



Dazu habe ich das „FMMArchiv“ nach rechts verschoben und das Artikel- und Autorenarchiv zwischen dem PDF und dem Archiv eingefügt.

Prüfen Sie nun Ihre Einstellungen, indem Sie auf das Bildfeld klicken: Je nachdem, ob ein Bild vorhanden ist oder nicht, sollte sich ein Fenster mit dem entsprechenden Autor oder mit einem leeren Datensatz öffnen. Damit haben wir schon einen Großteil unseres Projekts realisiert.



Im nächsten Schritt soll der Text aus dem PDF extrahiert werden. Da das mit FileMaker Bordmitteln nicht realisierbar ist, benötigen Sie ein Plugin. Darunter versteht Wikipedia ein „Softwaremodul, das von einer Softwareanwendung während seiner Laufzeit entdeckt und eingebunden werden kann, um dessen Funktionalität zu erweitern“.

Neu zu erstellen sind die in der Abbildung mit den Nummern 1-3 gekennzeichneten Bereiche.

Ich habe für mich das **MBS-Plugin** von *Monkeybreadsoftware* entschieden. Mit seinen mehr als 4000 Funktionen ist es eine Bereicherung für jeden FileMaker Entwickler und seine ca. 120 Euro mehr als wert. Sollte doch mal eine Funktion fehlen, genügt ein Hinweis an den Entwickler Christian Schmitz: Sie können fast sicher sein, die gewünschte Funktion in der nächsten Version zu finden. Wie ein Plugin implementiert wird, kann auf mehreren Internetseiten und in der FileMaker Hilfe nachgelesen werden, sodass ich auf eine Anleitung verzichte.

1. Hier befindet sich der Textteil der PDF-Dateien. Dazu erstellen Sie in der Tabelle *FMMA* folgende Felder mit den dargestellten Optionen:

Wenn Sie Lust haben, können Sie immer mal einen Blick in die Datei *FMMA Heft 3.FMP12* werfen und sie mit der aktuellen Version unseres Archivs vergleichen. Im Layout „FMMA“ werden Ihnen einige Veränderungen auffallen: Neben dem Fensterteil mit dem PDF ist nun auch der Teil mit dem Artikel- und Autorenarchiv implementiert worden. Da alles auf einen Blick zu sehen sein soll, habe ich die Aufteilung der Seite etwas umstrukturiert.

Feldname	Typ	Optionen
PDFText	Text	Indiziertes Feld
Lizenzkey	Text	Global
Seitenzahl	Text	
Autor	Text	
Creator	Text	
Titel	Text	
PDFVersion	Text	
Zähler	Formel	Nicht gespeichert, =Current_record & " von "& Current_count & " ("& Current_total & " gesamt")

Sind die Felder angelegt, benötigen wir ein Script, das ich „PDF einlesen“ genannt habe:

```

◆ Wenn
  MBS ("DynaPDF.IsInitialized") ≠ 1
◆ Script ausführen
  "InitDynaPDF"
◆ Ende (wenn)
◆ Variable setzen
  $pdf ; Wert: MBS("DynaPDF.New")
◆ Wenn
  MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
  "DynaPDF Fehler" ; $pdf
    
```

```

◆ Sonst
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.OpenPDFFromContainer";$pdf;
FMMA::FMM)
◆ Wenn
MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
"DynaPDF Fehler" ; $r
◆ Sonst
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.ImportPDFFile";$PDF)
◆ Wenn
MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
"DynaPDF Fehler" ; $r
◆ Sonst
◆ Feldwert setzen
FMMA::PDFText ; Austauschen ( MBS("DynaPDF.
ExtractText";$pdf) ;
[" " ; " "]; ["¶"; " "]; ["¶"; ""]; ["-¶"; ""])
◆ Feldwert setzen
FMMA::Seitenzahl ; MBS("DynaPDF.GetImportPageCount"; $pdf)
◆ Feldwert setzen
FMMA::Autor ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Author")
◆ Feldwert setzen
FMMA::Creator ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Creator")
◆ Feldwert setzen
FMMA::Titel ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Title")
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.GetImportPageBounds"; $pdf; 1;
"Mediabox")
◆ Feldwert setzen
FMMA::PDFVersion ; MBS("DynaPDF.GetPDFVersion"; $pdf)
◆ Schreibe Änderung Datens./Abfrage
Ohne Dialogfeld
◆ Ende (wenn)
◆ Ende (wenn)
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.Release";$PDF)
◆ Ende (wenn)

```

Zum besseren Verständnis werde ich Teile des Scripts kommentieren.

```

◆ Wenn
MBS ("DynaPDF.IsInitialized") ≠ 1
◆ Script ausführen
"InitDynaPDF"
◆ Ende (wenn)

```

Hier wird geprüft, ob das erforderliche Plugin (**DynaPDF**) initialisiert ist. Fehlt das Plugin, wird das Script „InitDynaPDF“ ausgeführt.

```

◆ Variable setzen
$pdf ; Wert: MBS("DynaPDF.New")

```

Es wird eine lokale Variable **\$PDF** mit dem Wert „MBS("DynaPDF.New")“ angelegt.

```

◆ Wenn
MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
"DynaPDF Fehler" ; $pdf
◆ Sonst
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.OpenPDFFromContainer";$pdf;
FMMA::FMM)

```

Gibt es bei dieser Aktion einen Fehler, wird ein Dialogfeld angezeigt. Ansonsten wird die Variable **\$PDF** angewiesen, den Ort des Containers mit dem PDF zu ermitteln.

```

◆ Wenn
MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
"DynaPDF Fehler" ; $r
◆ Sonst
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.ImportPDFFile";$PDF)

```

Auch hier informiert ein Dialogfeld über einen eventuellen Fehler. Ist alles glatt gelaufen, wird das PDF eingeleesen!

```

◆ Wenn
MBS("IsError")
◆ Eigenes Dialogfeld anzeigen
"DynaPDF Fehler" ; $r
◆ Sonst
◆ Feldwert setzen
FMMA::PDFText ; Austauschen ( MBS("DynaPDF.
ExtractText";$pdf) ;
[" " ; " "]; ["¶"; " "]; ["¶"; ""]; ["-¶"; ""])

```

Wenn es beim Einlesen keinen Fehler gab, wird der Text des PDFs extrahiert, die Bilder werden dabei ignoriert. Zur besseren Lesbarkeit der Textdatei werden doppelte Leerzeichen usw. entfernt oder ersetzt.

```

◆ Feldwert setzen
FMMA::Seitenzahl ; MBS("DynaPDF.GetImportPageCount"; $pdf)
◆ Feldwert setzen
FMMA::Autor ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Author")
◆ Feldwert setzen
FMMA::Creator ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Creator")
◆ Feldwert setzen
FMMA::Titel ; MBS("DynaPDF.GetImportDocInfo"; $pdf;
"Title")
◆ Variable setzen
$r ; Wert: MBS("DynaPDF.GetImportPageBounds"; $pdf; 1;
"Mediabox")
◆ Feldwert setzen
FMMA::PDFVersion ; MBS("DynaPDF.GetPDFVersion"; $pdf)
◆ Schreibe Änderung Datens./Abfrage
◆ Ohne Dialogfeld

```

Hier werden die relevanten Informationen ermittelt und in die angegebenen Felder geschrieben: die Anzahl der Seiten, der Autor des PDFs, das Programm, mit dem das PDF erstellt wurde, der Titel des PDFs und die Version.

- ◆ Ende (wenn)
- ◆ Ende (wenn)
- ◆ Variable setzen
\$r ; Wert: MBS("DynaPDF.Release";\$PDF)

Wie bereits weiter oben angedeutet, wird im Verlauf dieses Scripts ein weiteres Script namens „InitDynaPDF“ aufgerufen.

- ◆ # Locate DynaPDF library
- ◆ Variable setzen
\$databasePath ; Wert: Hole(DateiPfad)
- ◆ Variable setzen
\$databasePath ; Wert: MBS("Path.FileMakerPathToNativePath"; \$databasePath)
- ◆ Variable setzen
\$databasePath ; Wert: Austauschen (\$databasePath ; ".fp7" ; ".fmp12")
- ◆ Variable setzen
\$databaseName ; Wert: Hole(Dateiname) & ".fmp12"
- ◆ Wenn
Hole(SystemPlattform) = -2
- ◆ Variable setzen
\$path ; Wert: Austauschen (\$databasePath ; \$databaseName ; "dynapdf.dll")
- ◆ Sonst
- ◆ Variable setzen
\$path; Wert: Austauschen (\$databasePath; \$databaseName ; "dynapdf.dylib")
- ◆ Ende (wenn)
Initialize DynaPDF
- ◆ Variable setzen
\$r ; Wert: MBS("DynaPDF.Initialize"; \$path; FMMA::Lizenzkey)
- ◆ Wenn
\$r ≠ "OK"
- ◆ Eigenes Dialogfeld anzeigen
"DynaPDF failed to initialize" ; \$r
- ◆ Sonst
- ◆ Variable setzen
\$\$DynaPDFInited ; Wert: 1
- ◆ Ende (wenn)

Diese Datei wurde vom Hersteller des Plugins vorgegeben und ist meiner Ansicht nach fast selbsterklärend.

Erstellen Sie anschließend zwei Tasten, die Sie „PDF auslesen/extrahieren“ und „Programm beenden“ nennen. Letztere benötigen wir, falls die Werkzeugleiste oben ausgeblendet wird. Zum Zuweisen der Aktion klicken Sie mit der rechten Maustaste auf Ihre Taste, wählen den Menüeintrag „Tastenaktion“ und „Einzelner Schritt“ aus und geben dann den Script-Schritt „Programm beenden“ an.

Der Taste „PDF auslesen...“ weisen Sie das Script „PDF einlesen“ zu. Haben Sie das Feld **PDFText** platziert, können Sie den ersten Einleseversuch wagen.

Erstellen Sie noch die Felder **Seitenzahl**, **PDF-Version**, **Autor**, **erstellt mit** und **Titel**. Im Layout können Sie über den Menübefehl „Einfügen → Platzhalter ...“ das entsprechende Feld auswählen und einfügen.

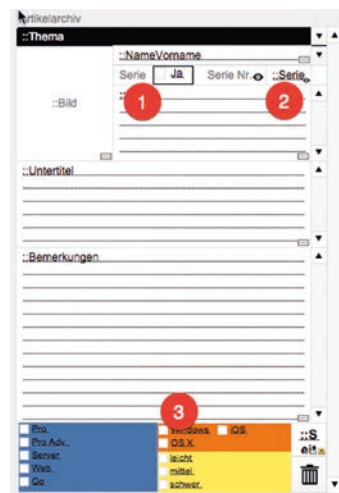


Der Bereich für das Extrahieren des PDF-Textes ist damit fertiggestellt.

2. Für das Artikelarchiv benötigen wir folgende Felder:

Feldname	Typ	Optionen
Inhaltssuchfeld	Text	Global
Suche erfolgreich	Text	Global

Fügen Sie die beiden neuen Felder so ein, wie es in der Musterdatei zu sehen ist.



Als dritten Schritt platzieren Sie die Auswahlfelder wie oben dargestellt. Beachten Sie dabei, dass Sie mit diesen Feldern innerhalb des Ausschnitts bleiben, sonst funktioniert Ihre Anzeige nicht so, wie sie soll.

3. In diesem Bereich gibt es einige Änderungen. Als Erstes legen Sie eine Tabelle namens **System** an, die weder eine ID noch ein Layout benötigt. In dieser Tabelle werden lediglich systeminterne Daten gespeichert, sodass es auch nur eines Datensatzes bedarf, der global angezeigt wird. Da wir später auch andere Daten aus dieser Tabelle benötigen, erstellen wir diese gleich mit. Die Erklärung dazu folgt im weiteren Verlauf des Textes.

Feldname	Typ	Optionen
----------	-----	----------

Systemstart	Text	Global
Programmname	Text	Global
Haupt	Text	Global
Neben	Text	Global
Version	Text	Global
Autor	Text	Global
FHöhe	Text	Global
FBreite	Text	Global
PosX	Text	Global
PosY	Text	Global
MHöhe	Text	Global
MBreite	Text	Global
Versionsnummer	Formel	Nicht gespeichert, = "V." & ZeichenRechts ("00" & Haupt; 2) & "." & ZeichenRechts ("000" & Neben; 3) & "-" & ZeichenRechts ("0000" & Version; 4)
Systembezug	Text	Global, Autom. Wert



Widmen wir uns nun dem Kopfbereich. Dort soll zunächst der Programmname angezeigt werden, was wir mit dem entsprechenden Feld **<<SYSTEM::Programmname>>** realisieren. Alles, was innerhalb einer doppelten Spitzklammer steht, wird von FileMaker als Platzhalter identifiziert, der im Blätternmodus mit unserem Programmnamen ersetzt wird.

Das **FMMSuchfeld** brauchen wir nur richtig zu positionieren, erstellt hatten wir es schon vor längerer Zeit.

Der Bereich daneben besteht aus Tasten, die der Navigation dienen („Gehe zum nächsten Datensatz“), bzw. dem Löschen oder Neuanlegen eines Datensatzes. Das Ganze lässt sich ganz leicht nachbauen – probieren Sie es einfach aus und wenn Sie nicht weiterkommen, schauen Sie in die Datenbank.

Das Feld **Zähler** ist nichts weiter als der Inhalt des Formelfeldes **Zähler** aus der Tabelle **FMMA**.

Wenden wir uns nun dem Inhaltssuchfeld zu. Dieses Feld ist der Grund, weshalb wir das Programm überhaupt geschrieben haben: die Volltextsuche in einem PDF. Wenn Sie das Inhaltssuchfeld zerlegen, sehen Sie, dass es aus drei Teilen besteht: dem Eingabefeld, der Suchtaste und der „AlleDatensätzeAnzeigen“-Taste.

Eine ähnliche Suche haben wir zu Anfang unseres Projekts schon einmal durchgeführt. Deshalb folgt nur eine kurze Erklärung. Dazu erstellen wir in der Tabelle **FMMA** zwei neue Felder.

Feldname	Typ	Optionen
Inhaltssuchfeld	Text	Global
Suche erfolgreich	Text	Global

Das Inhaltssuchfeld platzieren wir im Kopfbereich und erstellen zwei Tasten direkt daneben. Dazu kommt noch das erforderliche Script.

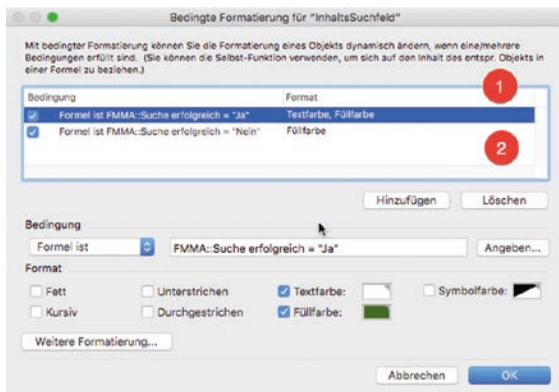
```

◆ # -----
◆ # © 2016 by Burkhard Quenzler, BeQuSoft
◆ # Script-Parameter: keine
◆ # SucheIndex
◆ # Suche erfordert in der zu Durchsuchenden DB
    ein Suchfeld.
◆ # Version: 1.01
◆ # -----
◆ Fehleraufzeichnung setzen
◆ Ein
◆ Gehe zu Feld
    FMMA::InhaltsSuchfeld
◆ Variable setzen
    $Suchbegriff ; Wert: FMMA::InhaltsSuchfeld
◆ Suchenmodus aktivieren
◆ Feldwert setzen
    FMMA::PDFText ; $Suchbegriff
◆ Ergebnismenge suchen
◆ Wenn
◆ Hole (AnzahlGefundeneDatensätze) = 0
    // Eigenes Dialogfeld anzeigen [ "Keine Datensätze" ;
    "Keine Datensätze mit dem Wort vorhanden!"
◆ Feldwert setzen
    FMMA::Suche erfolgreich ; "Nein"
◆ Blätternmodus aktivieren
◆ Alle Datensätze anzeigen
◆ Ende (wenn)
◆ Feldwert setzen
    FMMA::Suche erfolgreich ; "Ja"

```

Im Grunde entspricht dieses Script genau dem **FMM-ID-Suchscript**. Doch abweichend dazu wird das Suchwort jetzt Buchstabe für Buchstabe eingegeben. Aus diesem Grund benutzen wir keinen Trigger, sondern eine Taste zum Auslösen des Scripts. (Die Fortgeschrittenen unter Ihnen könnten diese Aktion auch per Trigger beim „Verlassen des Feldes“ realisieren.)

Erstellen Sie also eine Taste und weisen Sie ihr das Script „Inhaltssuche“ zu. Damit nach erfolgter Suche wieder alle Datensätze sichtbar gemacht werden können, benötigen wir noch eine Taste mit der Aktion „alle Datensätze anzeigen“. Um eine erfolgreiche Suche sichtbar zu machen, weisen wir dem Inhaltssuchfeld eine bedingte Formatierung zu. Klicken Sie dazu mit der rechten Maustaste auf das Feld, wählen Sie „bedingte Formatierung“ aus und geben Sie folgende Texte ein.



Jetzt ist der Moment gekommen, die Suche einmal auszuprobieren. Klappt alles? Vergessen Sie nicht, anschließend wieder alle Datensätze anzeigen zu lassen, denn sonst arbeiten Sie mit eingeschränktem Datenbestand weiter.

Als Letztes benötigen wir noch zwei Tasten zum Importieren und Exportieren von Dateien. Wählen Sie dazu das Tastenleistenwerkzeug aus, erstellen Sie zwei Tasten und belegen Sie diese mit folgenden Aktionen:

- 1. Aktion:** Datei importieren
Datei einfügen [Filter ; Einfügen ; Inhalt anzeigen ; Nie komprimieren ; FMMA::FMM]
- 2. Aktion:** Datei exportieren
Exportiere alle Feldwerte [FMMA::FMM]

Die Hilfefunktion, das Einbinden eines Videos und das Ausschalten der Werkzeugleisten sind in FileMaker selbst sehr gut beschrieben, sodass ich auf eine Erläuterung verzichte. Davon abgesehen können Sie jederzeit einen Blick in die Beispieldatei werfen, wo Sie sich den einen oder anderen Hinweis anschauen können.

Was aber noch einer Erklärung bedarf, ist das Startscript. In den Dateioptionen können Sie einstellen, welches Layout als Erstes aufgerufen werden soll. Sie können aber auch ein Script erstellen, das diese Aufgabe übernimmt. Mein Script habe ich „SYSTEM“ genannt und folgendermaßen aufgebaut.

```

◆ # -----
◆ # © 2016 by Burkhard Quenzler, BeQuSoft
◆ # Script-Parameter: keine
◆ # Startscript mit gemerkter Fenstergröße
◆ # Dazu ist eine Tabelle System erforderlich
◆ # Version: 1.01
◆ # -----
◆ Variable setzen
$Fensterbreite ; Wert: Hole(FensterSchreibtischBreite)
◆ Variable setzen
$Fensterhöhe ; Wert: Hole(FensterSchreibtischHöhe)
◆ Wenn
($Fensterbreite = System::MBreite UND $Fensterhöhe =
System::MHöhe) ODER (System::MHöhe = "" ODER
System::MBreite = "")
Fensterposition/-größe ändern [ Name: System::Programmname

```

```

& " " & System::Versionsnummer ; Höhe: System::FHöhe ;
Breite: System::FBreite ; Oben: System::MHöhe ; Links:
System::MBreite
◆ Sonst
◆ Fenster anpassen
◆ Alles anzeigen
◆ Ende (wenn)
◆ Fenstertitel setzen
Aktuelles Fenster ; Neuer Titel: System::Programmname & " " &
System::Versionsnummer
◆ Feldwert setzen
System::Systemstart ; System::Systemstart +1
◆ Gehe zu Layout
"FMMA" (FMMA)

```

Zum besseren Verständnis möchte ich auch hier einzelne Schritte erläutern:

```

◆ Variable setzen
$Fensterbreite ; Wert: Hole(FensterSchreibtischBreite)
◆ Variable setzen
$Fensterhöhe ; Wert: Hole(FensterSchreibtischHöhe)

```

In diesen beiden Script-Schritten wird die aktuelle Fensterbreite und die Fensterhöhe in jeweils einer Variablen hinterlegt.

```

Wenn
($Fensterbreite = System::MBreite UND $Fensterhöhe =
System::MHöhe) ODER (System::MHöhe = "" ODER
System::MBreite = "")
◆ Fensterposition/-größe ändern
Name: System::Programmname & " " & System::Versionsnummer
; Höhe: System::FHöhe ; Breite: System::FBreite ; Oben:
System::MHöhe ; Links: System::MBreite

```

Wenn diese zwei Werte identisch mit den vorher gespeicherten sind und die Position des Fensters nicht verändert wurde, muss nichts gemacht zu werden.

```

◆ Sonst
◆ Fenster anpassen
Alles anzeigen
◆ Ende (wenn)

```

Ansonsten wird das Fenster an die Werte angepasst und die Abfrage beendet.

```

◆ Fenstertitel setzen
Aktuelles Fenster ; Neuer Titel: System::Programmname & " " &
System::Versionsnummer

```

Hier wird der aktuelle Fenstertitel gesetzt, ergänzt mit der aktuellen Versionsnummer.

```

◆ Feldwert setzen
System::Systemstart ; System::Systemstart +1
◆ Gehe zu Layout
"FMMA" (FMMA)

```

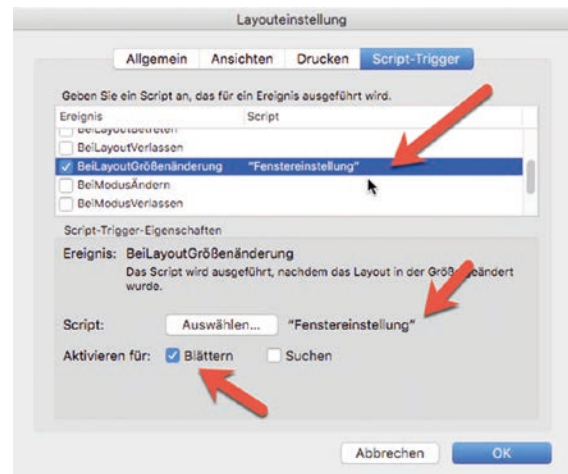
Diese Script-Schritte sorgen dafür, dass die Anzahl der Systemstarts um eins erhöht und das Layout „FMMA“ geöffnet wird. Ersteres kann später dazu benutzt werden, ein Backup o. ä. auszulösen.

Jetzt kommen auch die Felder zum Einsatz, die wir in dieser Folge erstellt haben und für die wir bisher noch keine Verwendung hatten. In diesen Feldern speichern Sie Ihre bevorzugte Fenstergröße und -position und sorgen mithilfe des Scripts „SYSTEM“ dafür, dass das Fenster bei jedem Neustart wie gewünscht geöffnet wird. In einem weiteren Script, das ich „Fenstereinstellung“ genannt habe, werden Änderungen an der Fenstergröße gespeichert:

Fenstereinstellung

- ◆ **Feldwert setzen**
System::FHöhe ; Hole(Fensterhöhe)
- ◆ **Feldwert setzen**
System::FBreite ; Hole(Fensterbreite)
- ◆ **Feldwert setzen**
System::PosX ; Hole(FensterLinks)
- ◆ **Feldwert setzen**
System::PosY ; Hole(FensterOben)
- ◆ **Feldwert setzen**
System::MHöhe ; Hole(FensterSchreibtischHöhe)
- ◆ **Feldwert setzen**
System::MBreite ; Hole(FensterSchreibtischBreite)

Eigentlich muss diese Einstellung nur nach einer Änderung des Layoutfensters „FMMA“ vorgenommen werden. Klicken Sie dazu im Layoutmodus mit der rechten Maustaste auf einen leeren Hintergrund (das heißt, in einen Bereich, wo es keine Felder oder dergleichen gibt), wählen den Menüeintrag „Layouteinstellung“ und anschließend „ScriptTrigger“ aus. Stellen Sie alles so ein, wie es in der Abbildung zu sehen ist.



Jetzt speichert FileMaker bei jeder Größenänderung dieser Datei die Werte in unseren Feldern. Damit das funktioniert, müssen Sie zum Schluss noch das Startscript „SYSTEM“ einbinden. Unter dem Menüpunkt **Ablage** → **Dateioptionen** → **Script-Trigger** weisen Sie dem Ereignis „Beim ersten Fenster öffnen“ das Script „SYSTEM“ zu. An dieser Stelle können Sie auch die Werkzeugleisten dauerhaft ausschalten.

Nun haben wir es geschafft: In vier Teilen haben wir unser Vorhaben umgesetzt und das gewünschte Programm erstellt.

Aber es geht noch weiter: In den nächsten beiden Folgen soll die Datei so umgebaut werden, dass sie nicht nur für das FileMaker Magazin, sondern für alle unsere PDF-Magazine als Archiv dienen kann. Auch etwas mehr Komfort kann unsere Datei vertragen, zum Beispiel, indem die Dateien automatisch nach dem Einlesevorgang in Text umgewandelt werden. ◆

Fußnote

¹ www.monkeybreadsoftware.de/FileMaker oder im FMM Webshop

Adressen

Deutschland, Österreich, Schweiz

FileMaker GmbH · Arnulfstraße 19
80335 München · Deutschland
Telefon +49 89 317759 -0
Telefax +49 89 317759 -20
URL www.filemaker.com/de

FileMaker Info Line

Telefon (D, A) +49 1802 000534
Telefax (D, A) +49 1802 000535
Telefon (CH) +49 89 317759 -70
Telefax (CH) +49 89 317755 -21

FileMaker Support Line

Montag bis Freitag von 9 bis 17 Uhr (außer an Feiertagen)
Deutschland 0800 724 3514
Österreich 0800 070 6144
Schweiz 043 55 79 035 ¹
¹ 12 Rappen/Min. aus Schweiz, Festnetz,
max. 60 Rappen/Min. Mobilnetz

Support Online: Fragen stellen

<http://filemaker-de.custhelp.com/app/ask>
(Eingangsbestätigung mit Vorgangsnummer und Antwort innerhalb von zwei Tagen per E-Mail)

FileMaker, Inc. Zentrale

5201 Patrick Henry Drive · P.O. Box 58168
Santa Clara; CA 95052-8168 USA
E-Mail info@filemaker.com
URL www.filemaker.com

FileMaker, Inc. Europa, Mittlerer Osten & Afrika

2 Furzeground Way / Stockley Park East
Uxbridge / Middlesex UB11 1BB
United Kingdom

Stand: 23.12.2016