



### Stefanie Juchmes

(Jg. 1992) hat Informatik an der Universität Bonn studiert. Sie ist regelmäßig auf FileMaker Konferenzen anzutreffen und arbeitet seit Januar 2019 als Juniorentwicklerin bei Monkeybread Software.

[stefanie@monkeybreadsoftware.de](mailto:stefanie@monkeybreadsoftware.de)


[Link](#) [Shop](#) MBS-Plugin

# Unsere Menüempfehlung



## Kontextmenüs für Ihre Lösungen

Sie haben folgende Vorstellung: Sie drücken auf eine Taste, woraufhin Ihnen in Ihrer Lösung ein Kontextmenü angezeigt wird, aus dem Sie verschiedene Optionen wählen können. Außerdem soll das Menü individuell an den jeweiligen User angepasst sein. Dann sollten Sie einmal einen Blick auf die „Menü“-Funktionen des MBS FileMaker Plugins werfen.

In diesem Beitrag zeige ich Ihnen, wie Sie mit dem **MBS FileMaker Plugin** eigene Kontextmenüs erstellen und welche Möglichkeiten Ihnen diese Menüs bieten.

Hierfür lege ich zunächst ein Script an, das beim Drücken der -Taste das Menü aufruft. In einem weiteren Script wird das Menü zusammengebaut.

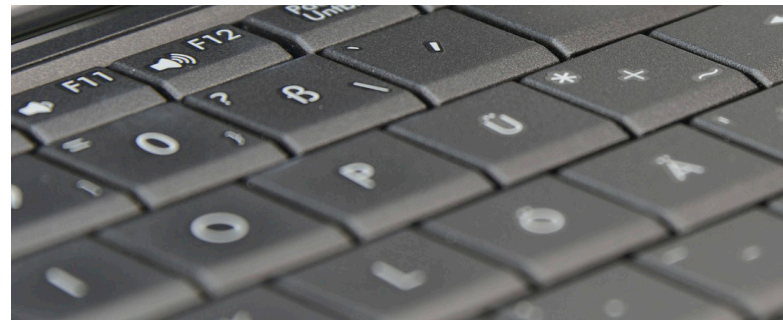
### Aufrufen eines Menüs

Da der Aufruf über die -Taste gestartet werden soll, wird der Scripttrigger „BeiLayoutTastendruck“ an das Script übergeben. Zuvor muss jedoch geprüft werden, ob tatsächlich die -Tast gedrückt wurde.

Diese Abfrage kann dann so aussehen:

```
◆ Wenn  
[ Hole ( TriggerTastendruck ) = Char ( 9 ) ]
```

Hat der Nutzer lediglich die Berechtigung zum Lesen, wäre es denkbar unpassend, wenn Sie z. B. in diesem Menü einen Eintrag haben, der einen neuen Datensatz hinzufügt. Der



Menüpunkt soll also nur bei einer Full-Access-Berechtigung angezeigt werden.

```
◆ Wenn  
[ Hole ( KontoBerechtigungenName ) = "[Full Access]" ]
```

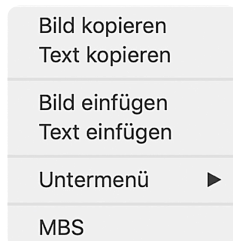
Mit dem Script „Menü erstellen“ wird das gewünschte Menü zusammengebaut. Wie dieses Script aussieht, schauen wir uns im Anschluss an.

```
◆ Script ausführen  
[ Angegeben: Aus Liste ; "Menü erstellen" ; Parameter: ]  
◆ Variable setzen  
[ $menu ; Wert: Hole ( ScriptErgebnis ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.PopUp" ; $menu ; "mouse" ) ]
```

Aufgerufen wird das Menü mit der **MBS-Funktion** „Menu.PopUp“, der die Referenz auf das Menü und die Position übergeben wird. In unserem Fall befindet sich die Position neben dem Mauszeiger, sie kann aber auch in Relation zum Fenster, zu Objekten und zum Bildschirm angegeben werden. In weiteren Parametern kann definiert werden, wie weit die Position des Menüs von dem im jeweiligen Anzeigemodus vorhandenen Ursprungspunkt entfernt sein soll. Wenn das

Menü also 20 Pixel rechts von der Position der Maus und 50 Pixel unterhalb dieser angezeigt werden soll, sieht der Schritt wie folgt aus:

```
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.PopUp" ; $menu ; "mouse" ; 20 ; 50 ) ]
```



Ein Kontextmenü

## Menü und Menüeinträge erstellen

Das Anlegen des Menüs geschieht mit der Funktion „Menu.CreateMenu“, die eine für den weiteren Ablauf notwendige Referenznummer zurück liefert. Es bietet sich an, eine minimale Breite für das Menü festzulegen – in unserem Fall sind das 400 Pixel.

```
◆ Variable setzen  
[ $menu ; Wert: MBS ( "Menu.CreateMenu" ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.SetMinimumWidth" ; $menu ; 400 ) ]
```

Als Nächstes soll ein erstes Element in das Menü eingefügt werden: das Kopieren eines Bildes. Dafür wird zunächst der Menüeintrag mit der Funktion „MenuItem.CreateMenuItem“ angelegt, wobei der Titel, der im Menü angezeigt werden soll, als Parameter übergeben wird. Zusätzlich kann ein Schlüssel angegeben werden, um den Eintrag weiter zu spezifizieren. Anschließend bekommt das Menü ein Script zugewiesen, das bei Auswahl des Eintrags ausgeführt wird. Wir nutzen dazu die Funktion „MenuItem.SetScriptAction“ und geben in den Parametern den Menüeintrag an, auf den die Funktion angewendet werden soll, den Namen der Datei, in der das Script liegt, und den Scriptnamen. Bei uns befindet sich das Script in derselben Datei, in der auch das Menü verwendet wird.

Sie können auch ein Script erstellen, das für das gesamte Menü zuständig ist. Mit der Funktion „Menu.SetScriptAction“ wird es immer dann aufgerufen, wenn ein Eintrag aus dem Menü gewählt wurde. Um abzufragen, welcher Menüeintrag gewählt wurde, stehen Ihnen die „Menu.SelectedItem“-Funktion zur Verfügung, die die Referenznummer eines Menüeintrags zurückgibt. Wenn Sie nach der Auswahl eines bestimmten Menüeintrags eine Berechnung durchführen möchten, verwenden Sie „MenuItem.SetEvaluate“ und geben in den Parametern die Berechnung in einem „Let“-Statement an.

Nachdem wir dem Eintrag gesagt haben, was er tun soll, fügen wir ihn z. B. mit „Menu.AddItem“ zu dem Menü hinzu.

```
◆ Variable setzen  
[ $itemBildKopieren ; Wert: MBS ( "MenuItem.CreateMenuItem" ;  
"Bild Kopieren" ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "MenuItem.SetScriptAction" ;  
$itemBildKopieren ; Hole ( Dateiname ) ; "KopiereBild" ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.AddItem" ; $menu ; $itemBildKopieren ) ]
```

Existieren bereits mehrere Einträge im Menü, wäre es schön, wenn man auswählen könnte, an welcher Stelle im Menü der Eintrag eingefügt werden soll. Das ist möglich, indem zusätzlich zu Menü und Menüeintrag der Index angegeben wird. Da der Index bei Null beginnt, sorgt die Eingabe einer „1“ dafür, dass das Element an die zweite Stelle im Menü rutscht.

```
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.InsertItem" ; $menu ;  
$itemTextKopieren ; 1 ) ]
```

Zur besseren Übersichtlichkeit können die Einträge im Menü durch Separatoren getrennt werden. Diese werden mit „MenuItem.CreateSeparator“ erzeugt und anschließend mit „Menu.AddItem“ in das Menü eingefügt.

```
◆ Variable setzen  
[ $Separator ; Wert: MBS ( "MenuItem.CreateSeparator" ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "Menu.AddItem" ; $menu ; $Separator ) ]
```

## Stylische Menüs

Standard-Menüeinträge können ziemlich langweilig aussehen, deswegen wollen wir ein bisschen Farbe ins Spiel bringen und die Einträge stylen. Dazu stehen uns einige Funktionen im **MBS FileMaker Plugin** zur Verfügung: Text kann z. B. in kursiver und/oder fetter Schrift gestaltet werden, Unterstreichungen sind mit der Funktion „MenuItem.SetUnderline“ möglich.

```
◆ Variable setzen  
[ $r ; Wert: MBS ( "MenuItem.SetItalic" ; $itemLogoEinfügen ;  
1 ) ]  
◆ Variable setzen  
[ $r ; Wert: MBS ( "MenuItem.SetBold" ; $itemLogoEinfügen ;  
1 ) ]
```

Wenn Sie eine andere Schriftart verwenden möchten, ist „MenuItem.SetFontName“ die Funktion Ihrer Wahl. Schriftfarbe und -größe sind ebenfalls anpassbar.

```
◆ Variable setzen  
[ $r ; Wert: MBS ( "MenuItem.SetFontName" ; $itemLogoEinfügen ;  
"Comic Sans MS" ) ]
```

◆ Variable setzen

```
[ $r ; Wert: MBS ( "MenuItem.SetFontColor" ;  
$itemLogoEinfügen ; "#0000FF" ) ]
```

◆ Variable setzen

```
[ $r ; Wert: MBS ( "MenuItem.SetFontSize" ;  
$itemLogoEinfügen ; 20 ) ]
```

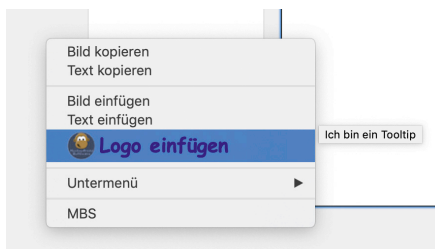
Mithilfe von „MenuItem.SetImage“ können Sie eine Abbildung wie z. B. ein Logo im Menüeintrag platzieren. In den Parametern muss dazu neben dem Menüeintrag auch der Container angegeben werden, in dem das anzuzeigende Logo liegt. Tooltips sorgen dafür, dass der Anwender einen Hinweis zu einem bestimmten Menüpunkt bekommt. Diese können Sie mit der Funktion „MenuItem.SetToolTip“ an einen Menüpunkt anhängen. Den Text des Tooltips geben Sie dann in den Parametern an.

◆ Variable setzen

```
[ $r ; Wert: MBS ( "MenuItem.SetToolTip" ; $itemLogoEinfügen ;  
"Ich bin ein Tooltip" ) ]
```

◆ Variable setzen

```
[ $r ; Wert: MBS ( "MenuItem.SetImage" ; $itemLogoEinfügen ;  
Menue::Logo ) ]
```



Ein Menüeintrag mit mehreren Styles und mit Tooltip

## Untermenüs

Wenn es viele Menüeinträge gibt, sind Untermenüs hilfreich, um ein bisschen für Ordnung und Übersichtlichkeit zu sorgen. Dazu muss zunächst ein Menüeintrag angelegt werden, an den das Untermenü angehängt werden soll. In den unten aufgeführten Scriptschritten wird der neue Menüeintrag „Untermenü“ erzeugt und dem Menü hinzugefügt.

◆ Variable setzen

```
[ $itemUnterMenue ; Wert: MBS( "MenuItem.CreateMenuItem" ;  
"Untermenü" ) ]
```

◆ Variable setzen

```
[ $r ; Wert: MBS( "Menu.AddItem" ; $menu ; $itemUnterMenue ) ]
```

Anschließend wird ein neues Menü mit dem Titel „Untermenü“ erstellt, dem wir Menüeinträge hinzufügen möchten – und das gleich mehrfach mit nur einem Funktionsaufruf. Ermöglicht wird das durch die Funktion „Menu.AddItem“, der wir das Untermenü als Referenz übergeben. Es folgt ein weiterer Parameter mit den Titeln der Untermenüeinträge, die mit einem Zeilenumbruch voneinander getrennt sind. Wahl-

weise könnten die Tags der Einträge in gleicher Weise angegeben werden.

◆ Variable setzen

```
[ $unterMenue ; Wert: MBS ( "Menu.CreateMenu" ; "Untermenü" ) ]
```

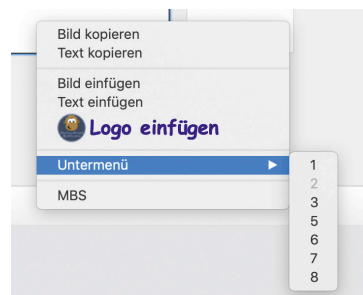
◆ Variable setzen

```
[ $r ; Wert: MBS ( "Menu.AddItem" ; $unterMenue ;  
"1¶2¶3¶4¶5" ) ]
```

Im Anschluss kann das Untermenü an den Eintrag „Untermenü“ des Hauptmenüs übergeben werden.

◆ Variable setzen

```
[ $r ; Wert: MBS ( "MenuItem.SetSubMenu" ; $itemUnterMenue ;  
$unterMenue ) ]
```



Ein Untermenü

Um eine Aktion für die Untermenüeinträge festzulegen, benötigen wir zunächst die Referenznummer zu dem entsprechenden Eintrag. Diese können mit „Menu.FindItemWithTitle“ gesucht werden, indem der Titel in den Parametern angegeben wird. Es besteht darüber hinaus die Möglichkeit, die Referenznummer mithilfe der Funktion „Menu.FindItemWithTag“ über das Tag zu erhalten.

◆ Variable setzen

```
[ $zwei ; Wert: MBS ( "Menu.FindItemWithTitle" ; $untermenue ;  
"2" ) ]
```

Soll der Titel oder das Tag geändert werden, stehen auch hierfür Funktionen bereit.

## Einträge deaktivieren, ausblenden oder löschen

An manchen Stellen Ihrer Lösung ist es sicherlich nicht gewünscht, dass der Nutzer bestimmte Aktionen ausführt. „MenuItem.SetEnabled“ bietet Ihnen die Möglichkeit, einen Menüpunkt zu deaktivieren, indem Sie in den Parametern eine „0“ angeben. Mit der Eingabe einer „1“ wird der Eintrag wieder aktiviert.

◆ Variable setzen





```
[ $r ; Wert: MBS ( "MenuItem.SetEnabled" ; $zwei ; 0 ) ]
```

Ähnliche arbeitet die „MenuItem.SetSelectable“-Funktion. Der Unterschied besteht darin, dass der Menüeintrag hier

ganz normal angezeigt wird, aber wenn man auf den Eintrag klickt, wird die Aktion, die normalerweise ausgeführt würde, nicht ausgeführt, sondern ignoriert.

1  
2  
3  
5  
6  
7  
8

Ein Menü, dessen zweiter Eintrag deaktiviert ist

Wenn ein bestimmter Menüpunkt gar nicht angezeigt werden soll, können Sie ihn mit der Funktion „MenuItem.SetHidden“ ausblenden. Sie können auch definieren, dass Menüeinträge erst dann angezeigt werden, wenn gleichzeitig eine Sondertaste gedrückt wird. Es können für folgende Sondertasten Funktionen gesetzt werden: /Alt, ,  und /Strg.

Um nicht mehr benötigte Menüeinträge aus einem Menü zu entfernen, haben Sie mehrere Möglichkeiten. Zunächst gibt es den Weg über die Referenz: Rufen Sie dazu die Funktion „Menu.RemoveItem“ mit der Referenznummer des zu löschenden Eintrages auf. „Menu.RemoveItemAt-Index“ wiederum arbeitet indexbezogen und entfernt z. B. das zweite Element in dem angegebenen Menü, wenn Sie als Index eine „1“ angeben. Mit der Funktion „Menu.RemoveAllItems“ werden alle Elemente aus einem Menü entfernt.

## Eine URL direkt aufrufen

Für Einträge im Menü, die eine URL direkt aufrufen, sodass z. B. das E-Mail-Programm des Anwenders geöffnet und als Empfänger eine bestimmte E-Mail-Adresse angegeben wird, verwenden Sie die „MenuItem.SetURL“-Funktion. In den Parametern können Sie die E-Mail-Adresse des Empfängers angeben.

```
◆ Variable setzen
[ $itemMBS ; Wert: MBS ( "MenuItem.CreateMenuItem" ; "MBS"
) ]

◆ Variable setzen
[ $r ; Wert: MBS ( "MenuItem.SetURL" ; $itemMBS ;
"mailto:stefanie@monkeybreadsoftware.de" ) ]

◆ Variable setzen
[ $r ; Wert: MBS ( "Menu.AddItem" ; $menu ; $itemMBS ) ]
```

## Menüs analysieren

Für den Fall, dass Sie wissen möchten, was genau in Ihren Menüs los ist, gibt es viele Funktionen, mit denen der Inhalt eines Menüs und der Status der einzelnen Menüeinträge abgefragt werden können. Einige davon möchte ich hier kurz vorstellen: „Menu.NumberOfItems“ ermittelt, wie viele Elemente ein Menü hat. Die IDs dieser Elemente können mit

„Menu.Items“ aufgelistet werden. Die Funktion „MenuItem.HasSubmenu“ gibt an, ob ein spezieller Menüeintrag ein Untermenü besitzt. Die Frage, ob es sich bei einem bestimmten Menüeintrag um einen Separator handelt, beantwortet „MenuItem.IsSeparatorItem“. Mit verschiedenen „Get-Funktionen“ können Details zu einem Menüeintrag abgefragt werden, z. B. erfahren Sie mit „MenuItem.GetFontSize“ die Schriftgröße eines Menüeintrags und „MenuItem.GetScriptName“ liefert den Namen des Scripts, das aufgerufen wird, wenn der Menüeintrag angeklickt wurde.

## Aufräumen

Da wir bei den Menüs und Einträgen mit Referenzen arbeiten, die im Speicher liegen, müssen wir diese wieder freigeben. Die Freigabe einzelner Referenznummern geschieht über die Funktionen „MenuItem.Release“ und „Menu.Release“, indem die Referenz in den Parametern angegeben wird. „Menu.ReleaseAll“ und „MenuItem.ReleaseAll“ geben alle Menüs und/oder Menüeinträge auf einmal frei. In unserem Beispiel werden diese beiden Funktionen ganz am Anfang des Menü-Aufbau-Scripts aufgerufen, um alte Referenzen zu beseitigen.

Wenn Sie die vorgestellten und viele weitere Funktionen in Ihrer Lösung nutzen möchten, benötigen Sie eine Lizenz des **MBS FileMaker Plugin**. Diese erhalten Sie beim K&K Verlag und auf unserer Webseite. Ich wünsche Ihnen viel Spaß mit den Kontextmenüs. ■

# FileMaker Magazin



## Das FileMaker Magazin

- Einzige deutschsprachige Fachzeitschrift zu FileMaker
- Wissen aus erster Hand von anerkannten FileMaker Fachautoren
- Große Themenvielfalt für Anwender und Entwickler

## Exklusiv für Premium-Abonnenten

- Sechs FMM Ausgaben pro Jahr
- Kostenlose Nutzung des Abonnentenbereichs auf [www.filemaker-magazin.de](http://www.filemaker-magazin.de)
- PDF-Archiv mit allen bisher veröffentlichten Ausgaben
- Jede Ausgabe mit kostenlosen Beispieldateien und Zusatzinfos zum Download

## Unser Service

- Aktuelle Neuheiten, Tipps und Infos, Kleinanzeigen und vieles mehr jederzeit auf unseren Webseiten
- Hilfe bei allen Fragen zu FileMaker im FMM Forum
- Kompetente Beratung zum Kauf von FileMaker Lizenzen: Einfach anrufen +49 (0)40 589 65 79 70.

Wenn Sie sich für ein FileMaker Magazin  
**Abo** interessieren, klicken Sie bitte hier!

Eine kostenlose **Leseprobe** des FileMaker Magazins erhalten Sie, wenn Sie hier klicken.

Hier finden Sie Aktuelles zu **FileMaker Lizenzen**, egal ob Sie kaufen, mieten oder sich einfach informieren möchten.