



# Arbeiten mit Dateidialogen

## Komfortablere Dateieporte per MBS Plugin

In diesem Artikel widme ich mich einem Allerweltsproblem: dem Im- und Export mehrerer Dateien. Genauer gesagt geht es darum, dass der Benutzer in einem Dialog mehrere Bilder für den Import auswählen kann bzw. beim Export einen Ordner angibt, in den alle Bilder exportiert werden. Beides lässt sich mit den Dateidialog-Funktionen des „MBS Plugin“ realisieren. Mit den 29 Funktionen des „MBS Plugin 2.7“ (und neuer) können Dialoge für die Datei- und Ordnerauswahl sowie für das Speichern von Dateien gesteuert werden.

Legen wir zunächst eine Beispieldatei an: Für unsere Zwecke genügt eine einfache Tabelle mit einem Medienfeld namens **Bild** und einem Textfeld namens **Dateiname**, sodass Bilder samt ihrem Pfad gespeichert werden können. Wir fügen drei Tasten hinzu: Die erste Taste ruft das Script „Importieren“ für den Import von Bildern auf. Mit der zweiten Taste wird der Befehl „Exportiere alle Feldwerte“ für das Medienfeld mit dem Namen **Bild** und ohne Pfadparameter aufgerufen. Die dritte Taste startet das Script „Exportieren“, mit dem alle Bilder der aktuellen Auswahl in einen Ordner geschrieben werden.

### Importieren

Kommen wir zum Script „Importieren“. Wie bei vielen Funktionen des **MBS Plugin** rufen wir die Funktion einfach über den Script-Schritt „Variable setzen“ auf. Die erste Funktion ist „FileDialog.Reset“, um alle Dialogparameter auf ihren jeweiligen Defaultwert zurückzusetzen. Anschließend rufen wir die Funktion „FileDialog.SetAllowMulti“ auf und übergeben „1“ als Parameter. Dadurch erlauben wir dem Benutzer, mehrere Dateien auszuwählen.

Im nächsten Schritt wird die Funktion „FileDialog.OpenFileDialog“ aufgerufen. Damit zeigen wir den Dialog zum Auswählen von Dateien an. Diese Funktion liefert entweder „OK“ oder „Cancel“ zurück – je nachdem, was ausgewählt wurde. Erhalten wir als Ergebnis „OK“, können wir weiter-

machen und abfragen, wie viele Pfade zurückgegeben wurden. In einer Schleife gehen wir dann von Datei zu Datei und ermitteln mit „FileDialog.GetPath“ den jeweiligen Pfad. Dabei wird die Variable **\$Index** hochgezählt, bis alle Dateien bearbeitet wurden.

Anschließend müssen wir mithilfe der **MBS Funktion** „Path.NativePathToFilemakerPath“ den Pfad jeder Bilddatei von dem nativen Mac-OS-X- oder Windows-Format in das Format umwandeln, das FileMaker verwendet. Diesen Pfad können wir über die Script-Schritte „Gehe zu Feld“ und „Bild einfügen“ als Bild einfügen. Außerdem speichern wir den Pfad im Textfeld, damit wir eine Möglichkeit haben, nach Namen zu suchen.

Das fertige Script sieht folgendermaßen aus:

### Importieren

```

◆ Kommentar
  Text: Alle Optionen für den Dialog zurücksetzen
-----
◆ Variable setzen
  Name: $r
  Wert: MBS ("FileDialog.Reset")
-----
◆ Kommentar
  Text: Mehrfachauswahl erlauben
-----
◆ Variable setzen
  Name: $r
  Wert: MBS ("FileDialog.SetAllowMulti"; 1)
-----
◆ Kommentar
  Text: Dialog anzeigen
-----
◆ Variable setzen
  Name: $r
  Wert: MBS ("FileDialog.OpenFileDialog")
-----
◆ Kommentar
  Text: Wenn OK, dann gehen wir durch die Liste der Dateien
-----
◆ Wenn
  Formel: $r = "OK"
-----
◆ Variable setzen
  Name: $index
  Wert: 0
-----
◆ Variable setzen
  Name: $count
  Wert: MBS ("FileDialog.GetPathCount")

```

◆ Schleife (Anfang)
◆ Neuer Datensatz/Abfrage
◆ Kommentar Text: Pfad abfragen beim Plugin und ins FileMaker Pfad Format übertragen
◆ Variable setzen Name: Spfad Wert: MBS ("FileDialog.GetPath"; \$index)
◆ Variable setzen Name: Spfad Wert: MBS ("Path.NativePathToFilemakerPath"; Spfad)
◆ Kommentar Text: Bild importieren
◆ Gehe zu Feld Feld: Bilder::Bild
◆ Bild einfügen Pfadliste: imagemac:Spfad imagewin:Spfad
◆ Kommentar Text: Noch den Pfad merken
◆ Feldwert setzen Feld: Bilder::Dateiname Formel: Spfad
◆ Kommentar Text: Änderungen schreiben
◆ Schreibe Änderung Datens./Abfrage
◆ Kommentar Text: Und die Schleife geht weiter
◆ Variable setzen Name: \$index Wert: \$index + 1
◆ Verlasse Schleife wenn Formel: \$index = \$count
◆ Schleife (Ende)
◆ Ende (wenn)

## Exportieren

Als Nächstes sollen Bilder exportiert werden. Dafür wird der Benutzer nach einem Zielordner gefragt, in den alle Bilder hineingeschrieben werden. Unser „Exportieren“-Script beginnen wir ebenfalls mit der **MBS Funktion** „FileDialog.Reset“. Anschließend rufen wir die **MBS Funktion** „FileDialog.SelectFolderDialog“ auf, um den Dialog anzuzeigen. Der Benutzer kann einen Ordner wählen, dessen Pfad mit „FileDialog.GetPath“ abgefragt wird. Da man nur einen Ordner auswählen kann, brauchen wir nur den Pfad mit dem Index „0“ abzufragen.

Für den Export konvertieren wir den Pfad in das FileMaker Format. Dann gehen wir in einer Schleife durch die Bilder in der Tabelle und erzeugen aus dem Pfad und dem Bildnamen einen Pfad für den Export. Hierfür nutzen wir die Funktion „LiesAlsText“, die bei einem Medienfeld den Namen der Datei zurückgibt. Mit dem Pfad können wir den Befehl „Exportiere alle Feldwerte“ ausführen und das Bild auf die Festplatte schreiben.

Das komplette Script sieht so aus:

## Exportieren

◆ Kommentar Text: Alle Optionen zurücksetzen
◆ Variable setzen Name: \$r Wert: MBS ("FileDialog.Reset")
◆ Kommentar Text: Dialog für Auswahl eines Ordners
◆ Variable setzen Name: \$r Wert: MBS ("FileDialog.SelectFolderDialog")
◆ Kommentar Text: Bei Erfolg, fragen wir den Pfad ab
◆ Wenn Formel: \$r = "OK"
◆ Variable setzen Name: Spfad Wert: MBS ("FileDialog.GetPath"; 0)
◆ Variable setzen Name: Spfad Wert: MBS ("Path.NativePathToFilemakerPath"; Spfad)
◆ Kommentar Text: Schleife für alle Datensätze
◆ Gehe zu Datens./Abfrage/Seite Option: Erste(r)
◆ Schleife (Anfang)
◆ Kommentar Text: Pfad für das Bild zusammenbauen
◆ Variable setzen Name: \$bildname Wert: LiesAlsText(Bilder::Bild)
◆ Variable setzen Name: \$bildpfad Wert: Spfad & "/" & \$bildname ]
◆ Kommentar Text: Bild rausschreiben
◆ Exportiere alle Feldwerte Feld: Bilder::Bild Pfadliste: \$bildpfad
◆ Gehe zu Datens./Abfrage/Seite Option: Nächste(r); Nach letztem beenden
◆ Schleife (Ende)
◆ Ende (wenn)

Ich hoffe, dass dieser Artikel einen kleinen Einblick in die neuen „Datei-Dialog-Funktionen“ des **MBS Plugins** geben konnte. Außerdem lade ich Sie ein, sich die 100 Beispieldateien zu den 1000 anderen Funktionen anzuschauen.

Weitere Informationen zum **MBS FileMaker Plugin** finden Sie auf der Webseite des Entwicklers:

[www.monkeybreadsoftware.de/filemaker](http://www.monkeybreadsoftware.de/filemaker)

Das Plugin läuft auf FileMaker ab Version 7 auf Mac (ab Mac OS X 10.4) und Windows (ab Windows XP). Bestellen können Sie es **Im Webshop** des K&K Verlags. ◆



## Das FileMaker Magazin

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

## Exklusiv für Premium-Abonnenten

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

## Unser Service

- Aktuelle Neuheiten, Tipps und Infos, Kleinanzeigen und vieles mehr jederzeit online 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.

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

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

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