

Tips for Filemaker Pro 8

by Michael Paine

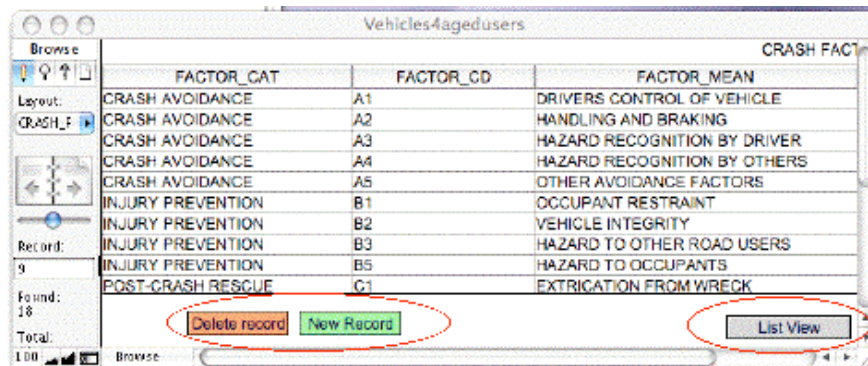
http://www4.tpg.com.au/users/aoaug/mac_osx.html

Updated 2 April 2006

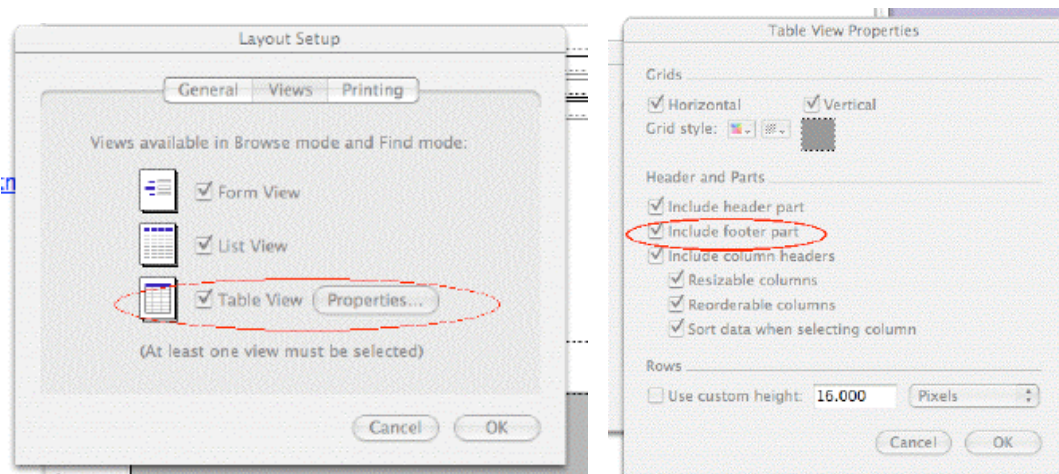
Contents

To include buttons in View as Table.....	1
To create a Browse records layout (aka Open Access 4).....	2
To create a report for specific records	3
To retrieve records based on a related field	3
To export records to an Excel file named after a field value.....	4
To create multiple relationship records	5
To open a Word Document from an FM8 layout.....	6
To copy a selected path and filename to a Filemaker container field.....	6
To change the default format for inserted fields	7
To change field contents to the default format.....	7
To print alternate backgrounds (stripes) in a report	7
To assign colour depending on the field contents	8
To create new records from a string of names	9

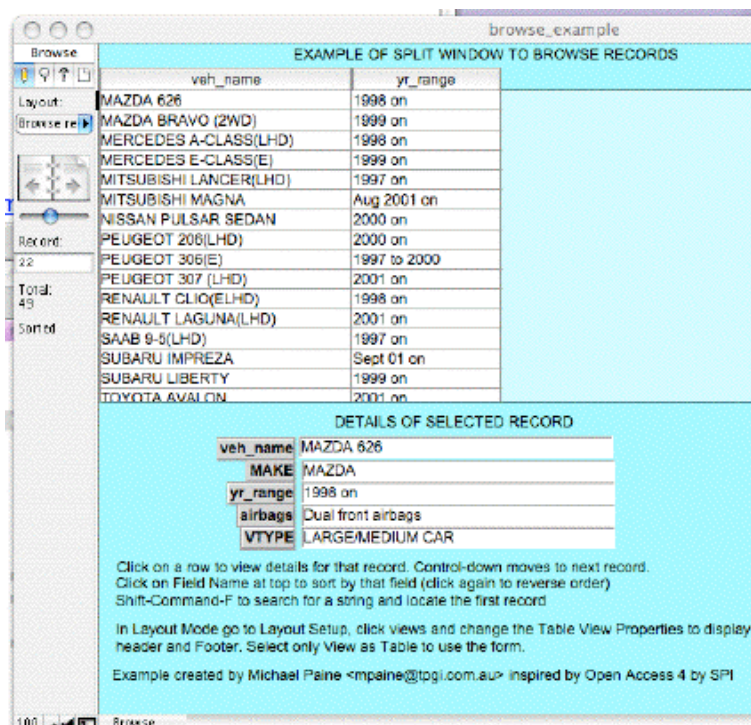
To include buttons in View as Table



Select the Layout and change to Layout Mode. Select **Layout Setup** from the Layouts menu. Click on the Views tab. Make sure the Table View checkbox is ticked. Click on **Properties**. The Table View Properties dialog is displayed. Tick **Include footer part**. The footer will now appear when the layout is opened as **View as Table** (Views menu). You can force the layout to only open as **View as Table** by unchecking the Form View and List View options in the Layout Setup Views dialog.



To create a Browse records layout (aka Open Access 4)



The above layout displays a list of all records in a table but also shows the details for the "current" record in the footer section.

Enable the footer section in the Layout Setup (see previous item). In the body section of the layout just add the fields that are required in the list. In the footer section add all the fields that are to be displayed.

To create a report for specific records

Unfortunately Filemaker Pro 8 does not assign a filter or sort to a report layout. This must be done either manually (each time the report is used) or via a script. The following script shows how this can be done.

```
Set Error Capture [ On ]
Enter Browse Mode
Go to Layout [ "CURRENT CRASH TESTS" (CRASH) ]
Enter Find Mode [ Specified Find Requests: Find Records; Criteria: VEHICLE_MODEL::PUB_STATUS_AU:
"TESTS PLANNED" ]
[ Restore ]
Sort Records [ Specified Sort Order: VEHICLE_MODEL::veh_name; ascending
CRASH::TEST_TYPE; ascending ]
[ Restore ]
Show Custom Dialog [ Title: "Report Preview"; Message: "Press Command-P to Print¶Press Command-1 to
return"; Buttons: "OK" ]
Enter Preview Mode
```

This script is created using Scriptmaker (**Scripts/Scriptmaker**) using the step-by-step process (there does not seem to be a way to enter scripts as text).

After the script is saved you will need to create a button on the Main Menu layout to activate the script. With the Layout selected go to **Layout Mode**. Click on the button toolbox item, click on the layout where the button is to appear and drag to the desired shape. Now the Button Setup dialog is displayed. Choose **Perform Script**. Click on the Specify and choose the report's script.

Your users will need an easy way to retrun to the Main Menu after previewing or printing a report. Filemaker assigns keyboard shortcuts to the top ten scripts in Scriptmaker. Create a script like the following:

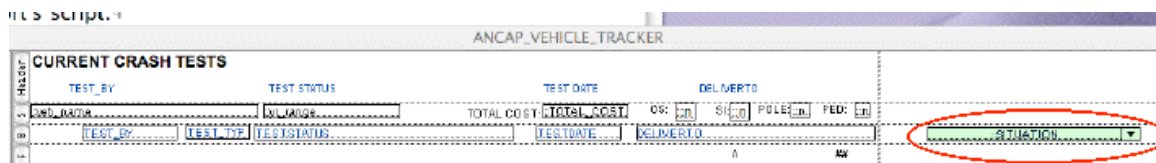
```
Enter Browse Mode
Go to Layout [ "TRACKER" (TRACKER) ]
Show All Records
```

Drag this script to the top of the Scriptmaker list. It can now be performed by pressing Command-1 (or Control-1 in Windows).

To retrieve records based on a related field

Table relationships can be created using **File/Define/Database/Relationships**. This displays a graph of all tables in the database. Drag a link between fields to create relationships.

To use these relationships in a layout, simply ensure that a field from the related table is included in the layout. An example is a report where a condition from a related table is used to select records for the report. To do this you will need to include the field of interest in the layout. If it is only to be used in a Find and is not to be printed then put the field in the Body section but over to the right, past the page divider.



Now this field can be used in a Find action within a Script (but try the Find manually first, to make sure everything is working as expected).

To export records to an Excel file named after a field value

Create script to do the export:

Set Error Capture [On]

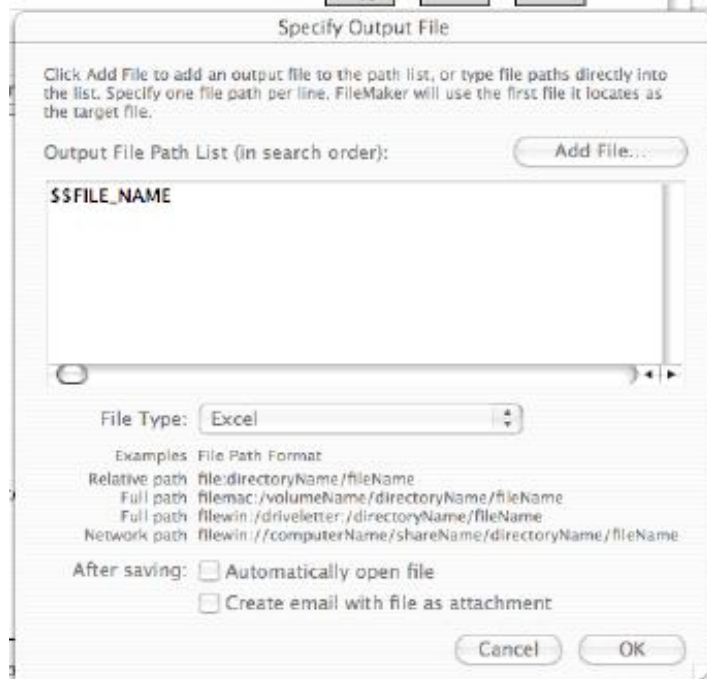
Set Variable [\$\$FILE_NAME; Value:"file:DATA/" & VARIANT::model_id & ".XLS"]

Export Records [File Name: "\$\$FILE_NAME"; Use field names as column names; Worksheet:

VARIANT::model_id]

[No dialog]

This creates a variable \$\$FILE_NAME that has the path and filename derived from the contents of the **model_id** field. This variable can be used in the Export dialog, where a path is to be specified:



Now include a button to perform this script on the layout that has the desired records. Note that all current records are exported to a folder called DATA under the database's folder.

To create multiple relationship records

Two tables are linked by a combined table to allow one to many joins on both sides: A-<AB>-B .

Below is some script I wrote to;

- a) check that the B (Brochure) record exists and create a new one if it does not
- b) check whether an AB (Broc_veh) record exists and create a new one if it does not. This table contains a unique field BROC_ID that is a combination of the A and B link fields.

The script is called from a button on the A (Vehicle_model) layout. The variables are used to automatically populate the fields in the newly created records.

```
Set Error Capture [ On ]
Set Variable [ $$MODEL_CD; Value:VEHICLE_MODEL::MODEL_ID ]
Set Variable [ $$BROC_CD; Value:VEHICLE_MODEL::AUST_PUB1 ]
Set Variable [ $$BV_CD; Value:VEHICLE_MODEL::AUST_PUB1 & "/" &
VEHICLE_MODEL::MODEL_ID ]
Enter Browse Mode
Go to Layout [ "BROCHURE" (BROCHURE) ]
Enter Find Mode [ ]
Set Field [ BROCHURE::broc_id; $$broc_cd ]
Perform Find [ ]
If [ Get(FoundCount)=0 ]
    Show Custom Dialog [ Title: "Confirm New BROCHURE Record"; Message: "A
    BROCHURE record does not exist for this
    model & brochure. Create new record?" & $$broc_cd & ""]; Buttons: "Yes", "No" ]
    If [ Get(LastMessageChoice)=1 ]
        New Record/Request
        Show All Records
        Set Field [ BROCHURE::broc_id; $$broc_CD ]
        Set Field [ BROCHURE::ncap_org; "ANCAP" ]
        Exit Script [ ]
    End If
End If
Enter Browse Mode
Go to Layout [ "BROC_VEH" (BROC_VEH) ]
Enter Find Mode [ ]
Set Field [ BROC_VEH::broc_mod; $$BV_cd ]
Perform Find [ ]
If [ Get(FoundCount)=0 ]
    Show Custom Dialog [ Title: "Confirm New BROCHURE/VEHICLE Record";
    Message: "A link record does not exist for this
    model & brochure. Create new record?" & $$bv_cd & ""]; Buttons: "Yes", "No" ]
    If [ Get(LastMessageChoice)=1 ]
        New Record/Request
```

```

        Show All Records
        Set Field [ BROC_VEH::model_id; $$MODEL_CD ]
        Set Field [ BROC_VEH::broc_id; $$broc_CD ]
        Set Field [ BROC_VEH::broc_mod; $$bv_cd ]
        Go to Layout [ "Model Master" (VEHICLE_MODEL) ]
        Exit Script [ ]
    End If
Else
    Go to Layout [ "Model Master" (VEHICLE_MODEL) ]
    Exit Script [ ]
End If

```

To open a Word Document from an FM8 layout

- Create a textfield <MyDocField>
- Enter complete path C:\MyFolder\MyFolder2\Mydocument.doc
- Create a script <OpenDoc>
- Enter ScriptStep Open URL. Specify MyDocField.
- Enable the Without Dialog option
- Close script editor

(Thanks Ursus, comp.databases.filemaker)

To copy a selected path and filename to a Filemaker container field

- 1) Create a container field, "c"
- 2) Create a script with the step:
Insert File[Reference; YourTable::c]
- 3) Create a field,
"MyFilePath" (calculation, text result) =

```

Let ( [FullURL = RightValues ( c ; 1 ) ;OSPrefix="filewin:";dDelimA = "/"; dDelimB =
"\\"];

```

```

Substitute(Right(FullURL;Length(FullURL) - Length(OSPrefix));
dDelimA ; dDelimB )
)

```

Now, when you run this script step, the user will be prompted to locate a file. They navigate to the file on your server. After they select the file and click "open," a reference is stored in container field c.

The reference looks like this, internal to "c":

file:MyServerFile.xyz

filewin://Server/Volume/Directory/MyServerFile.xyz

(FileMaker shows the document icon, or image, or media file in Browse mode, but has the text version of it stored as well,

accessible to formulas. The full path is only available when the file is stores as a "reference.")

The calculation in MyFilePath just strips out the beginning "filewin:" business and converts the "/" to "\" so it's appropriate for pasting into the address bar of any Windows Explorer or IE window. You'll want to adjust that formula if you are not running in an all-Windows environment. The result of the MyFilePath is:

\\Server\Volume\Directory\MyServerFile.xyz

(Thanks Bill Marriott)

To change the default format for inserted fields

You can click anywhere that's empty (ie. not a field or other layout object) and then set the Number format, which is then the default for the entire database and FileMaker, not just that one layout.

(Thanks Helpful Harry)

To change field contents to the default format

- Browse/find the records and select Records/Replace field contents/
- Replace with calculated result
- Enter the expression TextFormatRemove(<fieldname>)
- Click on REPLACE to apply this expression to all the records.

Bill Marriott suggests using an auto-enter calculation with this formula so that formatting is already the default when data is pasted.

To print alternate backgrounds (stripes) in a report

In Layout Mode select Format Part. Tick the box to use alternate backgrounds and select a colour for this purpose.

Script should include View as List before Preview (see below). View as Form does not print stripes. View as Table does not format correctly.

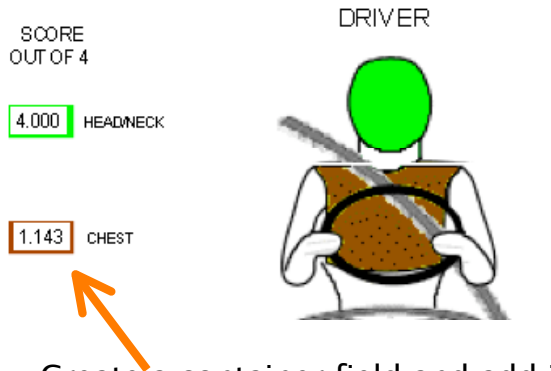
View As [View as List]

Enter Preview Mode

Print Setup [Orientation: Portrait; P[Restore; No dialog]

To assign colour depending on the field contents

There is no easy way to dynamically allocate colours to fields, depending on the field contents. Here is one trick.



- Create a container field and add it to the layout so that is slightly larger than the working field. Arrange/Send to back
- Using a drawing program, create rectangle pictures for each colour
- Write some script to assign the appropriate coloured box to the container field:

#Coloured boxes

Go to Field [DUMMY_PICS::rate_head]

Set Variable [\$\$n1; Value:scoring::HEAD_SCORE]

Perform Script ["assign_colour_box"]

#Passed variable is score \$\$n1, calling script must be set to a container field

Perform Script ["calc_colour"]

If [\$\$t1="G"]

Insert Picture ["image:pics/box_g.gif"]

[Reference]

Else If [\$\$t1="Y"]

Insert Picture ["image:pics/box_y.gif"]

[Reference]

Else If [\$\$t1="O"]

Insert Picture ["image:pics/box_o.gif"]

[Reference]

Else If [\$\$t1="B"]

Insert Picture ["image:pics/box_b.gif"]

[Reference]

Else If [\$\$t1="R"]

Insert Picture ["image:pics/box_r.gif"]

[Reference]

End If

Similar methods can be used to draw coloured diagrams, as in the example above.

To create new records from a string of names

```
#BUILD LINK
Go to Layout [ "Family" (Family) ]
Sort Records [ Specified Sort Order: FAMILY_LINK::ENG_FAMILY; ascending ] [ Restore; No dialog ]
Go to Record/Request/Page [ First ]
Loop
  #Loop for each record
  Set Variable [ $$T; Value:Family::Name ] # This string contains the names
  Set Variable [ $$lc; Value:0 ]
  If [ not IsEmpty ( Family::Name ) ]
    Loop
      #Separator can be comma or semi-colon
      Set Variable [ $$P; Value:Position ( $$T ; "," ; 1 ; 1 ) ]
      Set Variable [ $$P2; Value:Position ( $$T ; ";" ; 1 ; 1 ) ]
      If [ $$P = 0 ]
        #Both need to be zero to ensure no more names
        Set Variable [ $$P; Value:$P2 ]
      End If
      If [ $$P=0 ]
        #Comma not found - just one model
        Set Variable [ $$EM; Value:Trim($$T) ]
        Perform Script [ "NEW ENG_FAM LINK" ]
        Exit Loop If [ 1 ]
      Else
        Set Variable [ $$EM; Value:Trim(Left($$T,$$P-1)) ]
        Perform Script [ "NEW ENG_FAM LINK" ]
        Set Variable [ $$T; Value:Trim(Right($$T;Length($$T) - $$P )) ]
      End If
      #A way out if logic fails!
      Set Variable [ $$lc; Value:$lc + 1 ]
      Exit Loop If [ $$lc > 15 ] # Allow a maximum of 15 loops
    End Loop
  End If
  Go to Record/Request/Page [ Next; Exit after last ]
End Loop

#NEW ENG_FAM LINK
#Called from BUILD LINK
Go to Layout [ "FAMILY_LINK" (FAMILY_LINK) ]
New Record/Request
Set Field [ FAMILY_LINK::ENG_MODEL; Trim($$EM) ]
Commit Records/Requests
Go to Layout [ original layout ]
```

Scraps:

Just in case this is news to anyone else, 24U Software (www.24usoftware.com) demonstrated how to dynamically go to a tab AND how to know which tab your at. The documentation file is called "ScriptedTabControl".

In short it involves having a UNSTORED repeating calculation field with formula:

```
Let($$TheRepetitionNumber = Get (CalculationRepetitionNumber),"")
```

The calculation will always evaluate to empty BUT it will set the variable "\$\$TheRepetitionNumber" to the repetition number that was used to evaluate the calculation.

On tab 1 you only put repetition 1 of the calc, on tab 2 only rep 2, 3 only rep 3, etc.

Every time the user switches to a different tab the unstored calc is reevaluated and the variable "\$\$TheRepetitionNumber" contains the tab the user is at.

(Blag)