

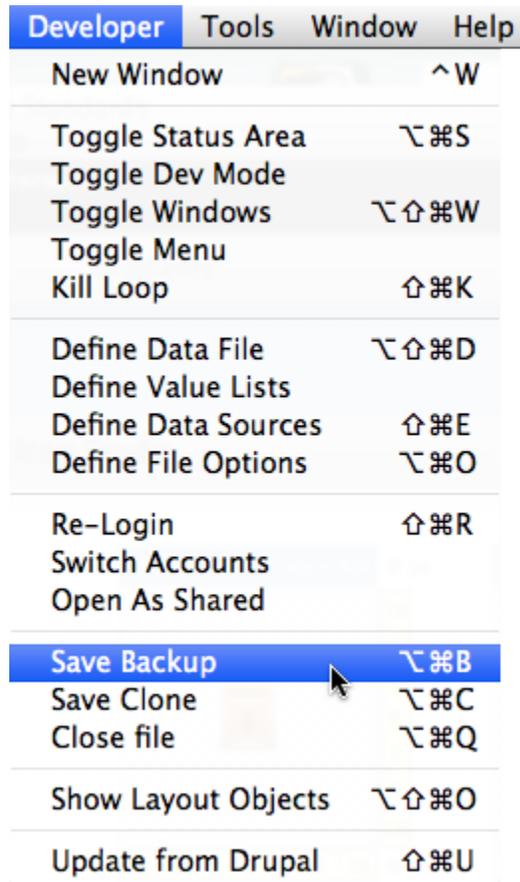
Developer menu

Overview

Developing within FileMaker is unlike working within an [Integrated development environment](#). In order to facilitate developer features beyond those provided by FileMaker itself, it's helpful to add specific developer features to aid in the development process.

Example of a Developer menu

The following screen shot presents a Developer menu which only appears when the file is opened with developmental access (typically by using the Developer custom function).

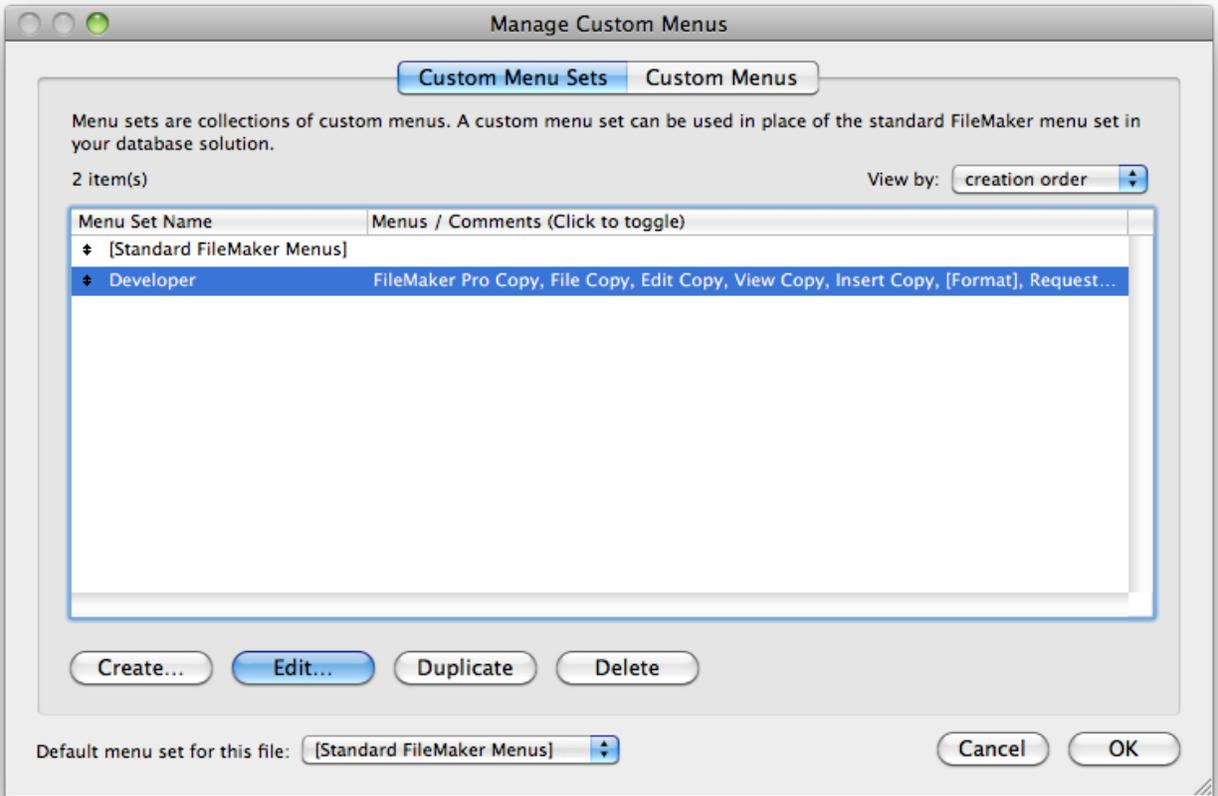


Implementation

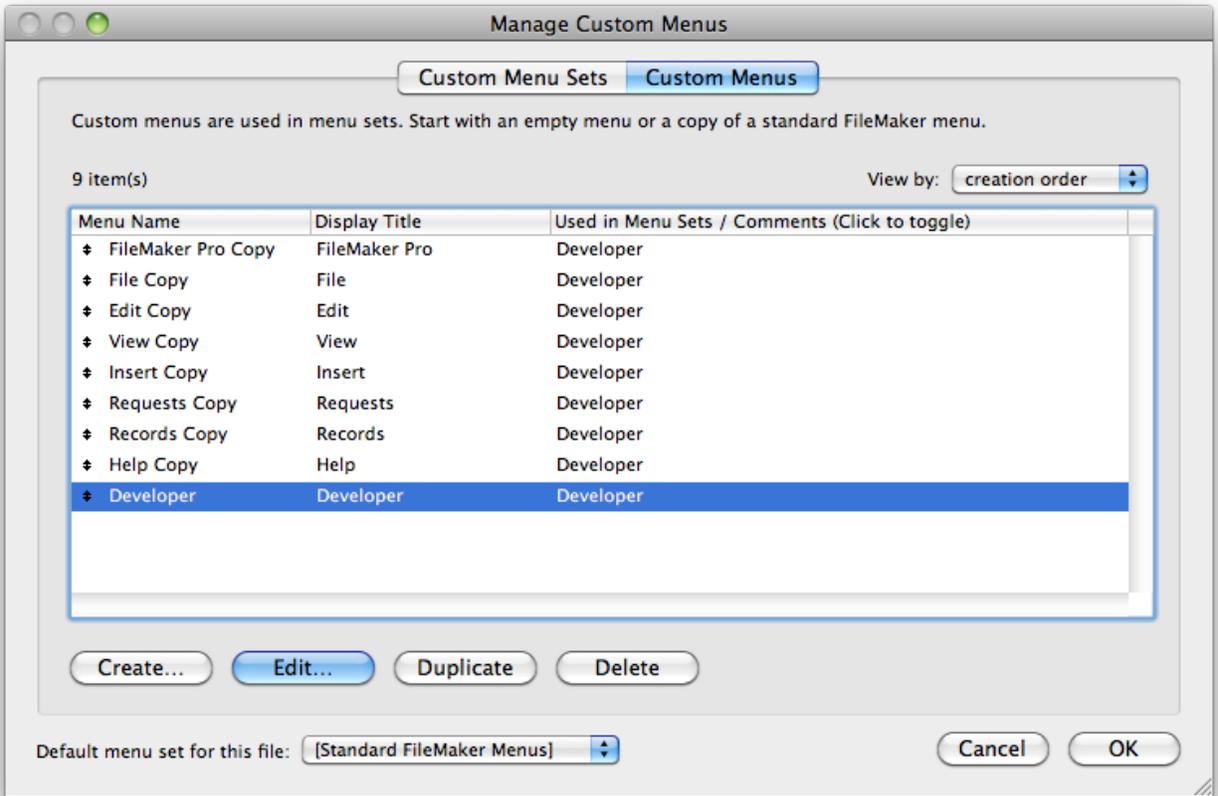
While the [Standards.fp7](#) file does include the Developer menu set and menu, it is documented here for reference.

The process of adding a Developer menu is very straight forward. Follow these steps.

1) Rename the secondary menu created by FileMaker from **Custom Menu Set 1** to **Developer**. If you've already renamed this menu and deleted items, then you'll have to start from scratch or duplicate any other menu as a starting point.



2) Create a new Custom Menu named Developer and add this to the Custom Menu Set named Developer (note: A **Custom Menu Set** and **Custom Menu** are two different things).

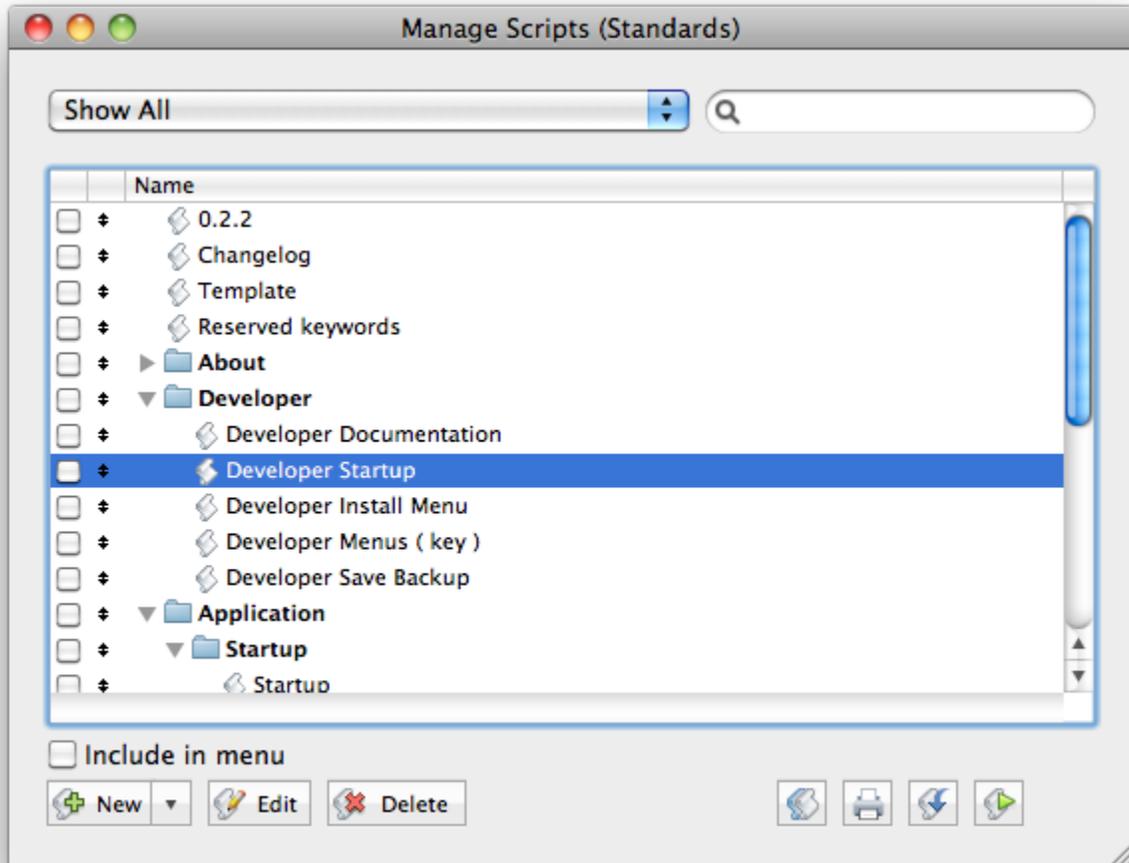


3) Make sure to add the [Developer custom function](#) to your solution. This is used to limit access to running developer related scripts.

4) Create a dedicated **Developer Startup** script which is called via a Perform Script step within the actual Startup script set in the File options. Provided the logged-in account is a Developer account, the Developer menu should be installed.

! Relationship Graph Warning

As of FileMaker 11, there is an issue (bug) with the alignment features for Table Occurrences when attempting to align using the alignment icons at the bottom of the graph. These items will be grayed out if the default [Standard FileMaker Menus] is not currently set. To regain these alignment features in the Graph, you will need to set the current menu set to the standard menus temporarily and then back to the Developer menu set.



Edit Script "Developer Startup" (Standards)

View: **all by category** Script Name: **Developer Startup**

Control
Perform Script
Pause/Resume Script
Exit Script
Halt Script
If
Else If
Else
End If
Loop
Exit Loop If
End Loop
Allow User Abort
Set Error Capture
Set Variable
Install OnTimer Script

Navigation
Go to Layout
Go to Record/Request/Page
Go to Related Record
Go to Portal Row
Go to Object
Go to Field
Go to Next Field
Go to Previous Field
Enter Browse Mode
Enter Find Mode

Show Compatibility
Client

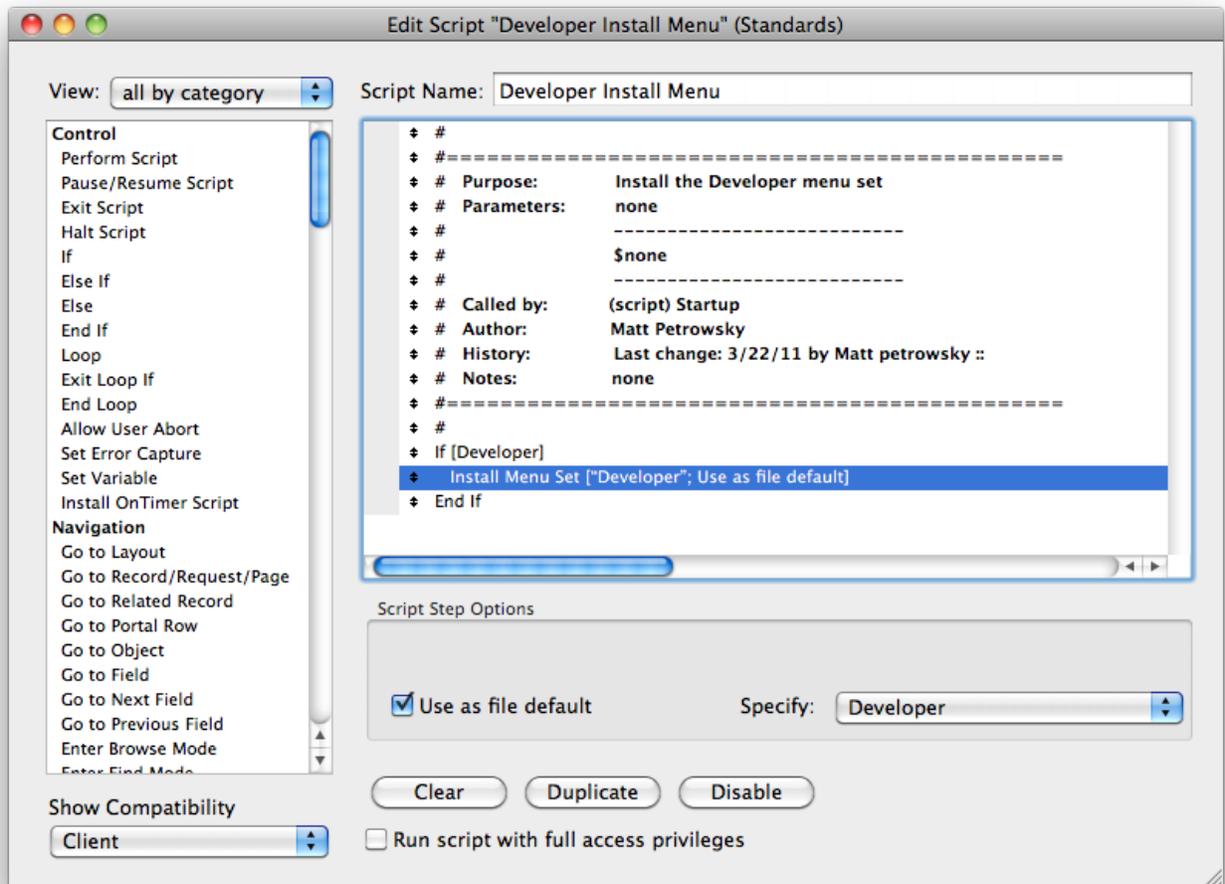
Script Name: **Developer Startup**

```
◆ #  
◆ #=====  
◆ # Purpose:      Run scripts specific to the development environment  
◆ # Parameters:   none  
◆ #  
◆ #              $none  
◆ #  
◆ #=====  
◆ # Called by:    (script) Startup  
◆ # Author:       Matt Petrowsky  
◆ # History:      Last change: 3/22/11 by Matt petrowsky ::  
◆ # Notes:       Make sure all of your developer specific scripts are protected by  
◆ #=====  
◆ #  
◆ If [Developer]  
◆   Perform Script ("Developer Install Menu")  
◆ End If
```

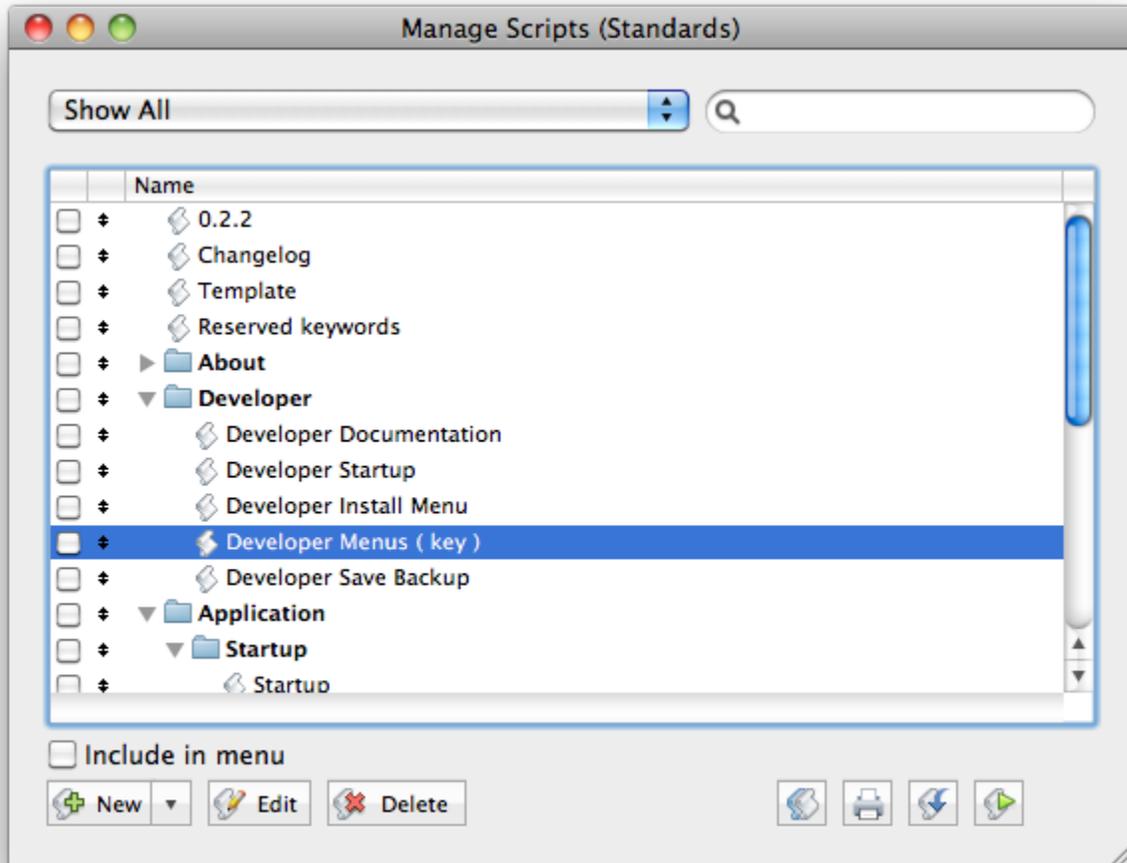
Script Step Options

Move **Duplicate** **Disable**

Run script with full access privileges



5) Use one single script as a dispatcher for all developer related actions (menu items within the Developer menu). This simplifies the process of managing individual menus. Passing a named parameter of \$key tells the dispatcher script which sub-script to run.



6) Add as many relevant Developer related menu items to this menu. The following screen shot shows two common developer actions of saving a backup (only works on a locally developed file) and re-login. Make note of the Perform Script action for the menu where making a new menu item is simply a matter of clicking the **Duplicate** button and changing the inbound parameter of \$key.

