

.com Solutions Inc.



# How to Import FileMaker Pro Databases into FmPro Migrator

# How to Import FileMaker Pro Databases into FmPro Migrator

## 1 Installation

1.1 Installation Tips 4

## 2 Step 1 - Import Table Info

2.1 Step 1 - Import Table Definitions 10

## 3 Step 2 - Import Layouts

3.1 Step 2 - Import Layouts 14

## 4 Step 3 - Import Scripts

4.1 Step 3 - Import Scripts 22

## 5 Step 4 - Import Relationships, TOs, Value Lists & Custom Functions

5.1 Step 4 - Import Relationships, TOs, Value Lists and Custom Functions 27

## 6 Troubleshooting

6.1 Troubleshooting AppleScript GUI Automation [MacOSX] 35

# Installation

## Installation Tips

---

This document provides an explanation of the steps required to import info from FileMaker Pro database files using FmPro Migrator Developer Edition and FileMaker Pro Advanced.

Revision 14

2/5/2014

Changes with this version:

Replaced the use of QuickKeys with AppleScript code embedded within FmPro Migrator.

Added support for FileMaker 12/13.

### Installation & Pre-Migration Tips

Tip1: Don't rename the FmPro Migrator or FileMaker Advanced executable files on your hard disk. If you want to maintain different versions of these applications, put them into folders containing different names. The AppleScript/WinAutomation scripts look for these two applications by their executable filenames, and if they are renamed the shortcuts won't be able to find the applications even if the applications are currently running.

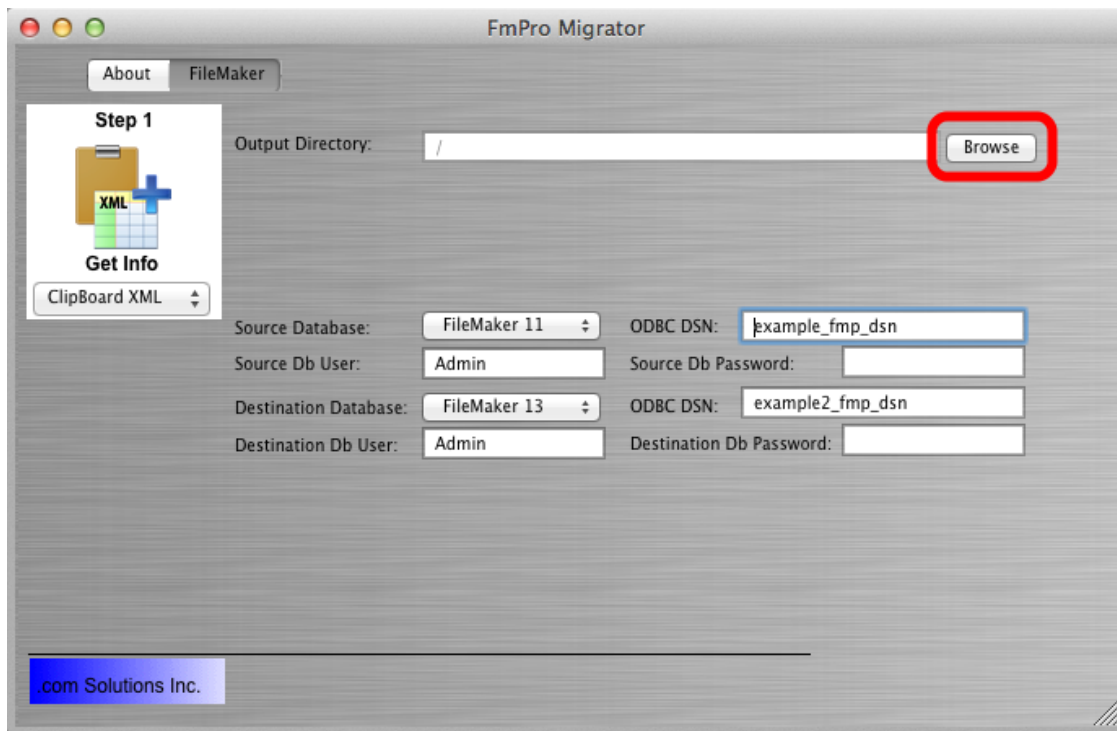
Tip2: On Mac OS X, make sure that you only have one copy of FileMaker Pro Advanced running when performing the GUI automation tasks (importing layouts into FmPro Migrator, creating value lists, custom functions, scripts, layouts in FileMaker Pro Advanced). FmPro Migrator will verify that the version of FileMaker you have selected in the Source Database and Destination Database menus matches what is currently running. But if 2 copies of FileMaker are running, this verification process will not necessarily work properly. It is likely that the first opened copy of the FileMaker application will receive the AppleScript commands.

Tip3: On Windows, in order for the WinAutomation shortcuts to find elements within the FileMaker user interface, the Manage/Define Database window needs to be resized to its smallest horizontal and vertical sizes. Otherwise, the shortcuts won't be able to find the correct locations for clicking on button or icons during the automated creation of TOs and Relationships.

**Note:** This is not an issue on MacOSX because the AppleScript code is not dependent upon the size or location of objects in FileMaker windows. The AppleScript code references objects by their underlying object name, not their location or their text label (i.e. "click button 4 of window 1" is used instead of "click OK button of window Manage Value Lists")

Tip4: Duplicate value lists and script names need to be considered prior to performing a Table Consolidation or Conversion project. If duplicated value lists and custom functions contain the same functionality, you don't have to do anything because FmPro Migrator will skip them automatically. Otherwise, these duplicated objects should be renamed within FileMaker prior to starting the Table Consolidation project.

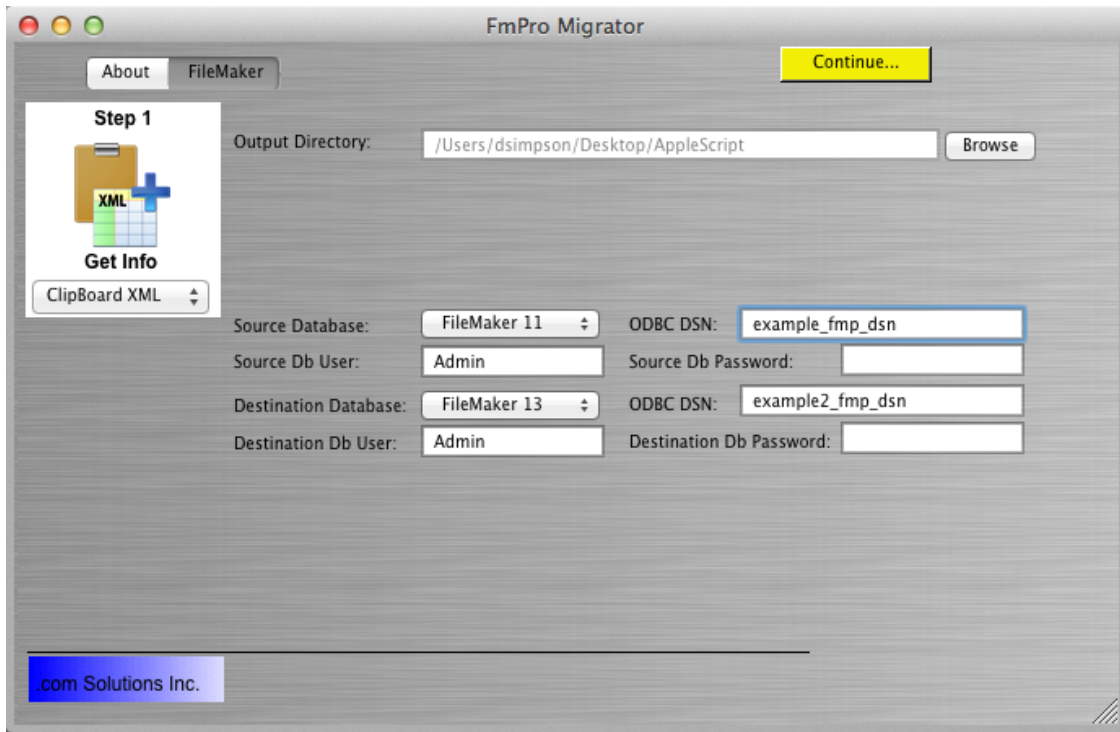
## Select Output Directory



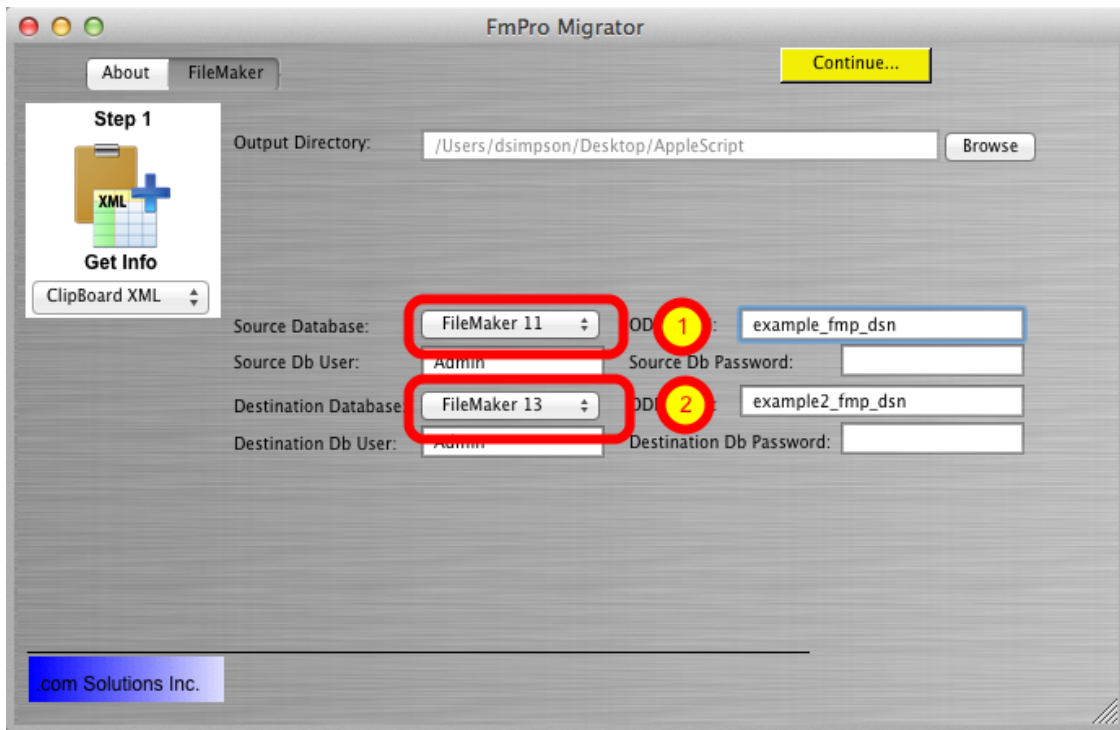
FmPro Migrator stores migration information for each project within a SQLite database file named MigrationProcess.db3. FmPro Migrator needs to know where this file should be stored, prior to starting your migration project. You can have as many MigrationProcess.db3 files as needed (for instance a separate file for each project), as long as these files are stored within different directories on your hard disk. This output directory is also used for writing automation scripts, instructions and the FileMakerDB.fp7/FileMakerDB.fmp12 template database file.

On the FileMaker tab within FmPro Migrator, click the Browse button, then select an output directory

[If you select an output directory which already contains a MigrationProcess.db3 file, the yellow Continue... button will be enabled so that you can continue working on a previously created project. Otherwise, the MigrationProcess.db3 file will be created automatically when you add the first table to your migration project.]



## Select Source and Destination Database Types



Selecting both the Source Database and Destination Database types as FileMaker 7/8/9/10/11/12/13 triggers FmPro Migrator to enter Table Consolidation mode once the first table is copied from the Clipboard.

Note: It is important to accurately select the type of Destination Database in order for FmPro

Migrator to use the correct AppleScript or WinAutomation scripts when performing migration tasks. Different versions of FileMaker have used differing menu options, therefore customized automation scripts are used by FmPro Migrator based upon the database version.

**FileMaker 12+ Note1:** As of FileMaker 12, there have been changes made to the format of XML code placed onto the clipboard for layouts and scripts. These different formats are not fully compatible with different versions of the FileMaker database. For instance, if you capture layouts with FileMaker 11 and then try to paste them into FileMaker 13, the contents of the layout won't get pasted. So for Table Consolidation projects it is recommended that you use the same version of FileMaker for all phases of the project. Therefore, select the source and destination database versions to be the same version.

## Convert FileMaker 5/6 Files

When upgrading FileMaker 5/6 files, the files should be converted into .fp7 files using the FileMaker Pro 11 Trial Version from the FileMaker website. This tech note provides further details: [http://help.filemaker.com/app/answers/detail/a\\_id/10571](http://help.filemaker.com/app/answers/detail/a_id/10571)

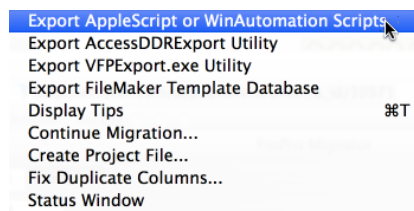
FmPro Migrator Developer Edition uses FileMaker Advanced 11-13 when performing a Table Consolidation or Conversion project. Open the original (older) files with the newer version of FileMaker Advanced and the files will be converted automatically by FileMaker. For the purposes of this documentation, these new files are considered to be the source data files for the project.

For example, if you want to convert FileMaker Pro 5.5 (.fp5) files to FileMaker Pro 13:

- 1) Download the FileMaker Pro 11 Trial version from the tech note provided above.
- 2) Convert the .fp5 files into .fp7 files using the FileMaker Pro 11 Trial version.
- 3) Convert the .fp7 files into .fmp12 files using FileMaker Pro Advanced 13.

Within FmPro Migrator, you must select FileMaker 13 as both the source and destination database versions.

## Install WinAutomation jobs [Windows]



Select the Export WinAutomation Scripts menu item from the File menu.

On Windows, the WinAutomation jobs will be written into the output directory into a file named:

FmPro Migrator WinAutomation Jobs.zip

A pdf installation manual named FmPro Migrator WinAutomation Installation.pdf will also be written into this directory.

Your web browser will then be directed to the appropriate company's website so that you can download the latest WinAutomation software.

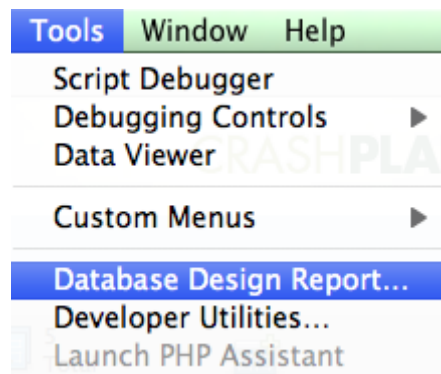
Follow the instructions listed in the WinAutomation pdf installation files to install the software and shortcuts as well as configure FileMaker Pro Advanced.

On MacOS X, the AppleScript code is embedded within FmPro Migrator therefore no installation is required.

Selecting this menu will save a text file of AppleScript code, a LiveCode AppleScript test stack and a LiveCode performance testing stack to the output directory.

These files have been released as open source for use by the FileMaker community.

### Export FileMaker Database Design Report (DDR XML)



FmPro Migrator requires information from the FileMaker Database Design Report (also known as DDR XML) in order to import information about FileMaker Pro TOs, Relationships, Value Lists, Custom Functions and Layouts.

For each FileMaker database file to be migrated, select Tools -> Database Design Report...

When prompted for the output directory, select XML as the output file type (do not select HTML).

DDR XML files can be exported for all open FileMaker database files at the same time.

**Note:** DDR XML files can only be exported by FileMaker Pro Advanced, not FileMaker Pro.



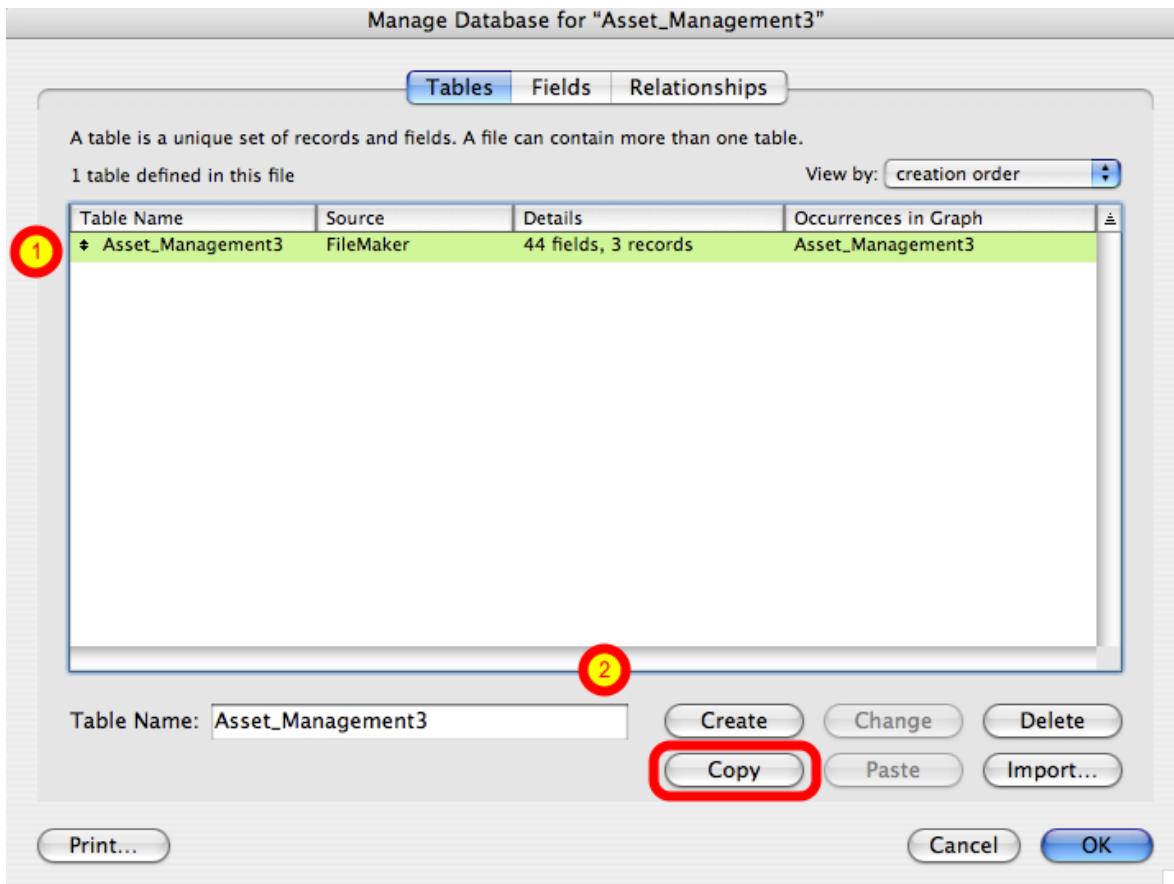
# Step 1 - Import Table Info

## Step 1 - Import Table Definitions

### Step 1 - Import Table Definitions

FmPro Migrator Edition uses FileMaker table definitions copied from the clipboard for table consolidation and conversion projects. These table definitions include all table/field options, validations and calculation formulas required to recreate the tables within a new database file. When performing SQL database conversions, this info is also used to create new tables in the SQL database.

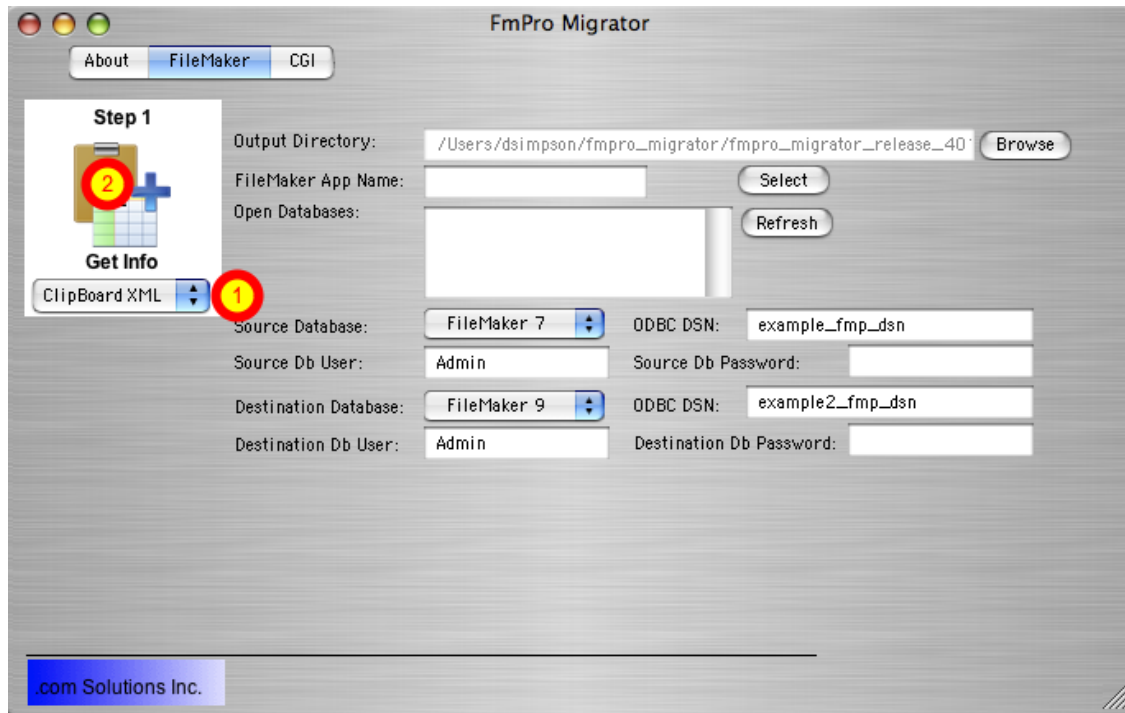
### Step 1 - Open Define/Manage Database Dialog within FileMaker Pro Advanced



Open each source database file within FileMaker Pro Advanced. Select the table within the Define/Manage Database dialog, click the Copy button.

**Note:** If you licensed FmPro Migrator Platinum Edition and need help with this step because you don't have FileMaker Pro Advanced, then please contact .com Solutions via the [Contact page on the www.fmpromigrator.com website](http://www.fmpromigrator.com).

## Step 1 - Paste Table into FmPro Migrator

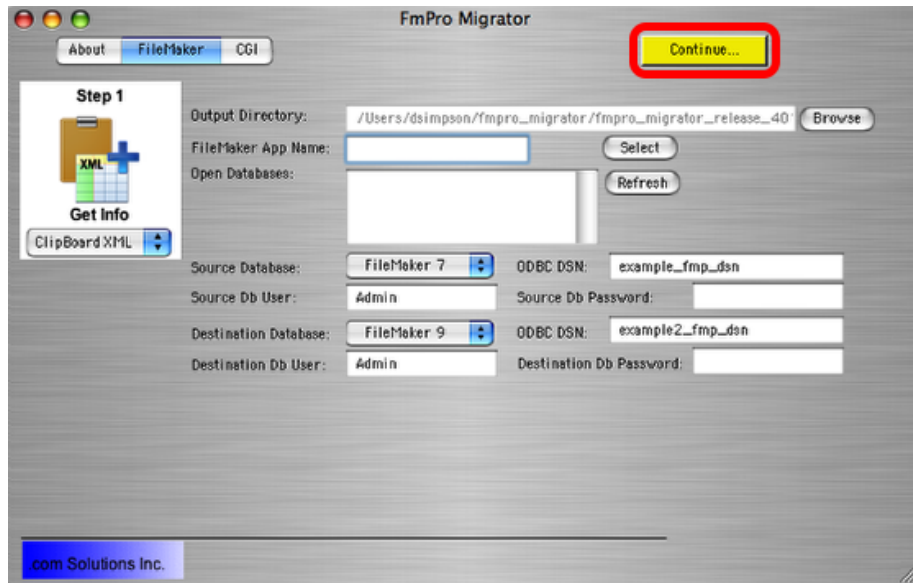


On the FileMaker tab of the main FmPro Migrator window, select the "Clipboard XML" menu item, then click the Get Info button.

## Export DDR File

After copying the table definition via the Clipboard, export a DDR XML file from each source database file. These files will be used to import Relationships into FmPro Migrator later in the migration process.

## Click Continue... Button

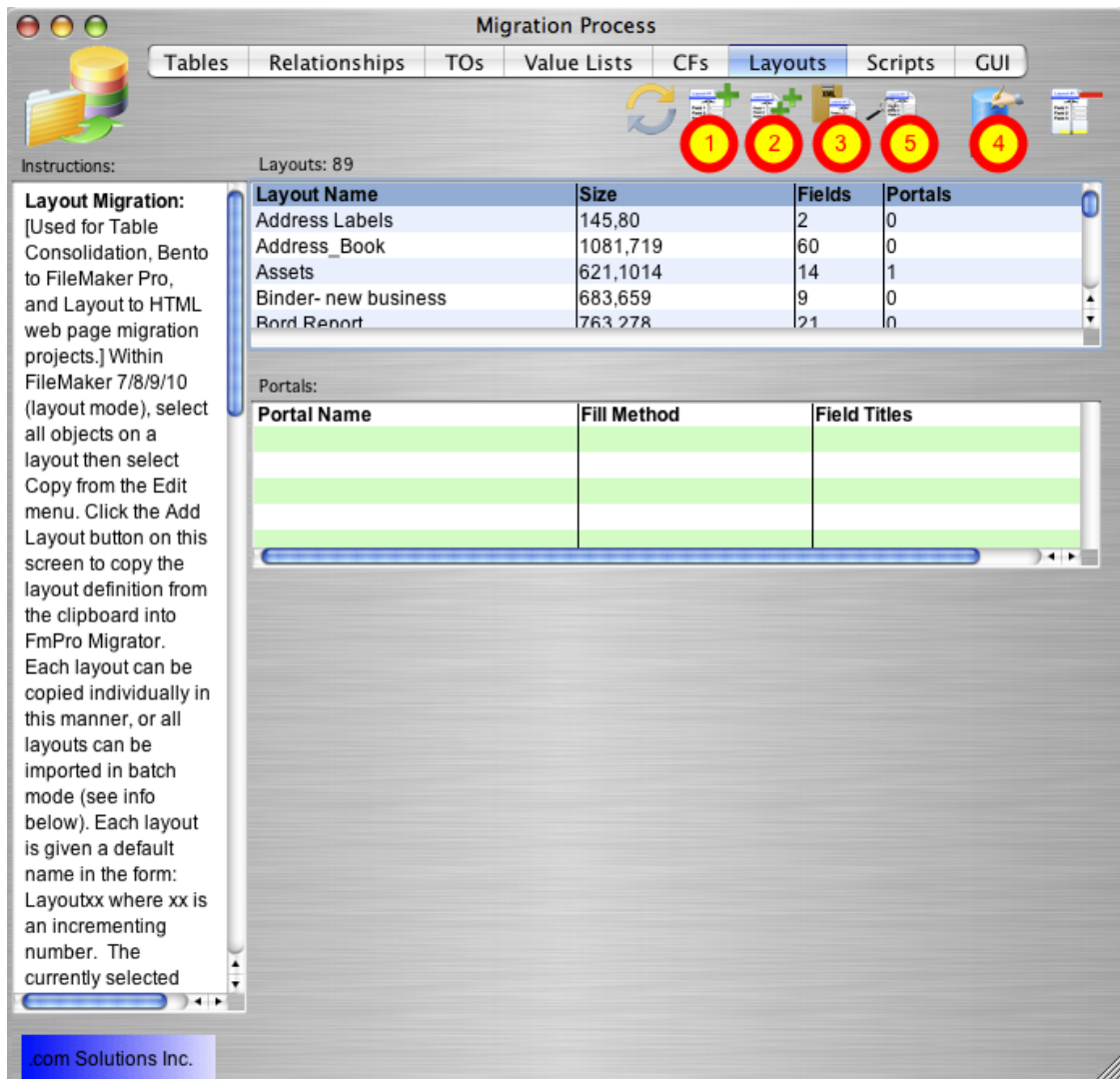


Once the first table has been copied into FmPro Migrator, the yellow Continue... button will become visible. Click the Continue... button to open the Migration Process window.

# Step 2 - Import Layouts

## Step 2 - Import Layouts

### Step 2 - Layouts Tab

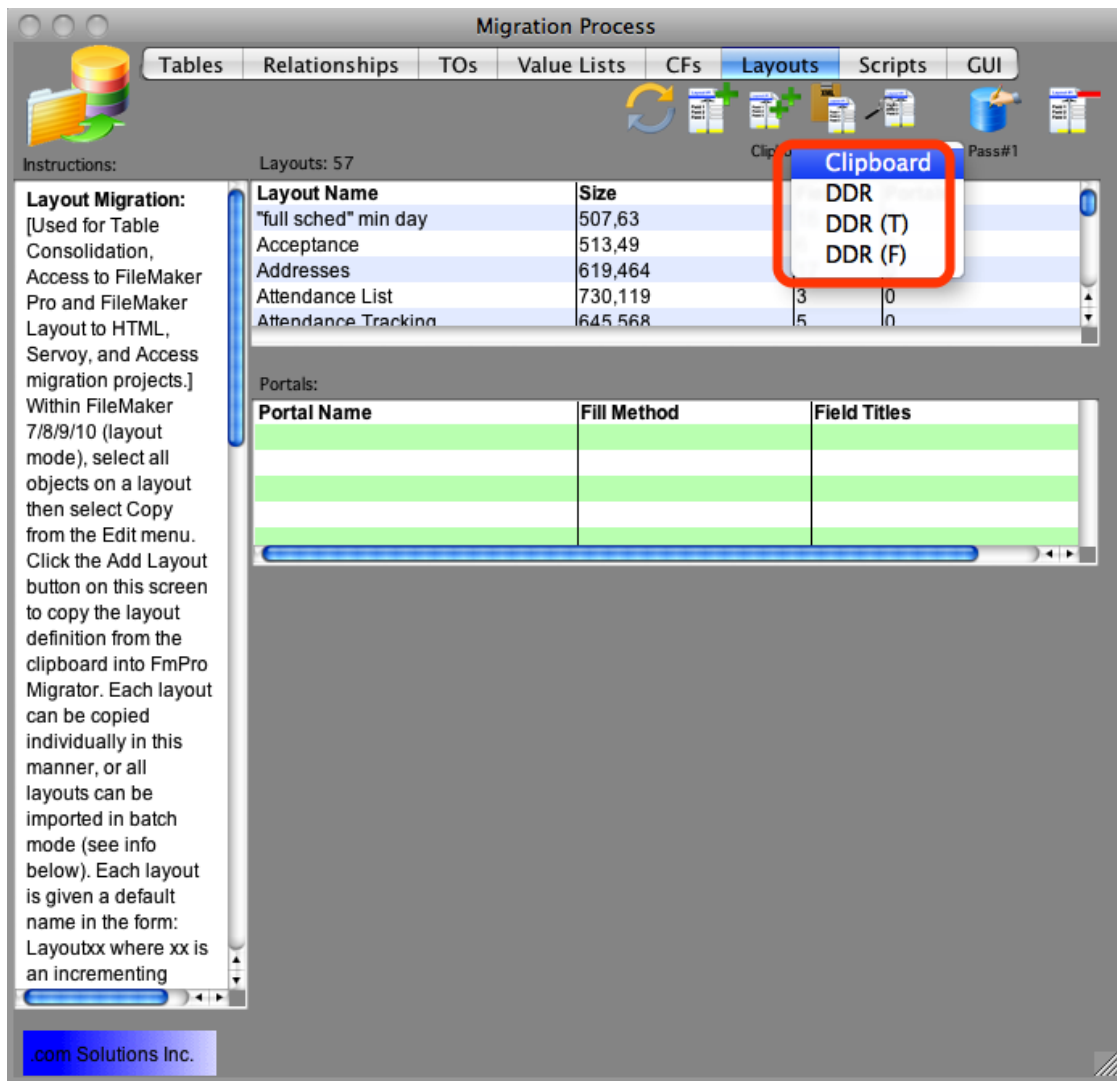


The Layouts tab provides an interface for managing layouts during a migration project. Layouts can be captured from FileMaker either one at a time [1] or in batch mode [2].

Clicking on a layout in the list displays details for the layout, including a thumbnail version of the layout image.

Clicking on a portal displays the portal rectangle on the thumbnail image.

## Select Layout Import Method



Select the import method pop up menu under the Layout Batch Import button. By default, clicking the Layout Batch Import button will import layouts from the Clipboard. The options within this menu are:

Clipboard - Uses AppleScript/WinAutomation to control the FileMaker Pro graphic interface and import each layout using the clipboard.

**Note:** This is the **ONLY** option which can be used for Table Consolidation/Database Rebuild projects and single file FileMaker Pro solutions, in order to insure that scripts and buttons continue to work correctly.

**Note:** The DDR Importing methods are now deprecated. These methods are only useful for project estimating purposes in order to quickly import and count the number of layouts in a multi-file project.

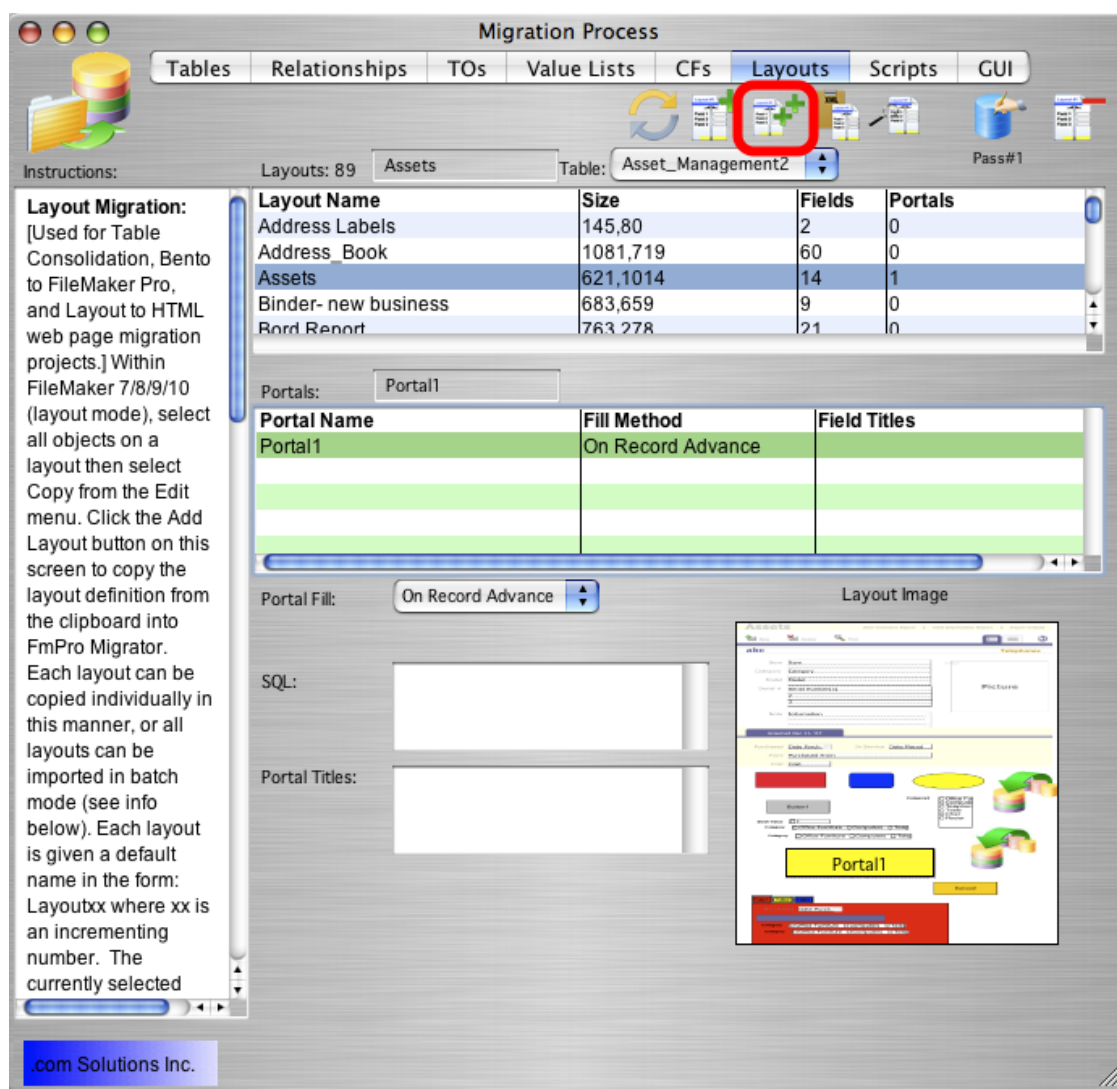
DDR - Imports all layouts from a DDR XML file selected by the user. Layouts are given the exact same name found within the DDR XML file.

**Note:** Using the DDR XML file to import layouts is no longer recommended for Table Consolidation or Database Rebuild projects.

DDR (T) - Imports all layouts from a DDR XML file selected by the user. The layout base table name is added as a suffix to the original layout name. The Import Relationships from DDR step needs to occur prior to importing layouts in order for the base table names to be available.

DDR (F) - Imports all layouts from a DDR XML file selected by the user. The FileMaker Pro filename used to create the DDR XML file is added as a suffix to the original layout name.

## Step 2 - Click Import All Layouts Button

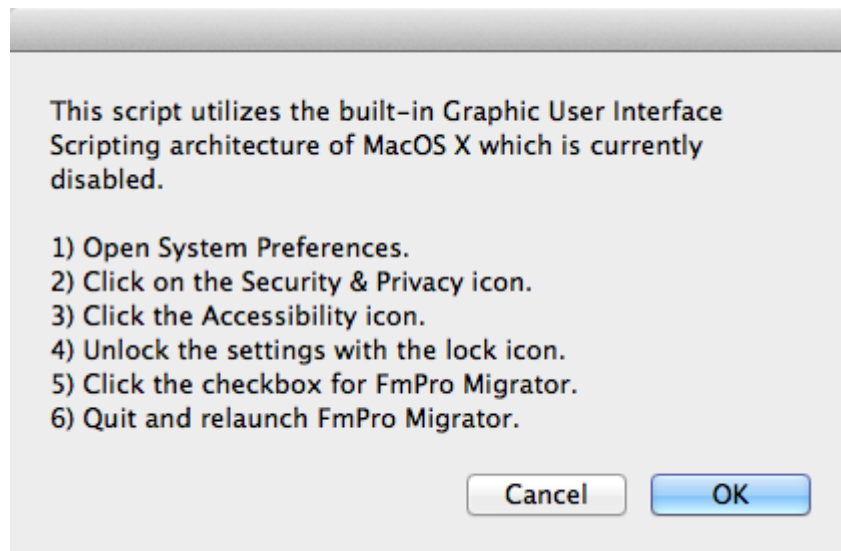


The Import All Layouts button starts the process of importing all of the layouts.

**Note:** The Import Type menu is covered up by the TO Name menu when a layout has been selected in the list - as shown in this screenshot. Also shown in this screenshot is the location of Portal1 on the thumbnail image of the layout, since Portal1 has been selected from the list of portals.

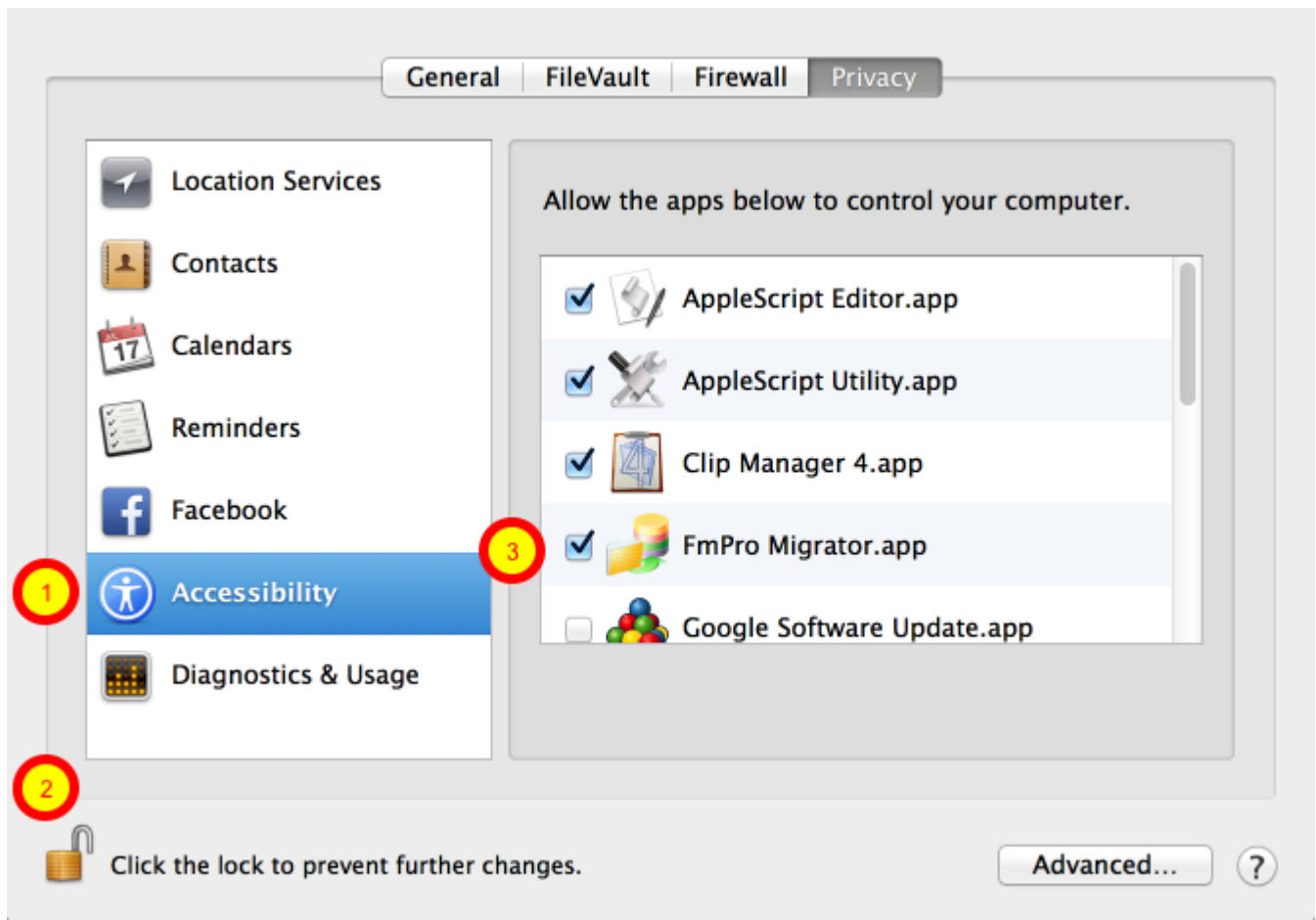


## Step 2 - MacOSX 10.9 Security Dialog



On MacOSX, FmPro Migrator uses AppleScript to perform batch processing tasks with FileMaker database files. Starting with MacOSX 10.9 (Mavericks), each application which uses the Accessibility API to control the computer needs to be authorized. FmPro Migrator performs a test to verify that it has been authorized to control the computer via AppleScript GUI scripting, and presents this dialog if it has not yet been enabled.

## Step 2 - MacOSX 10.9 Security Preferences

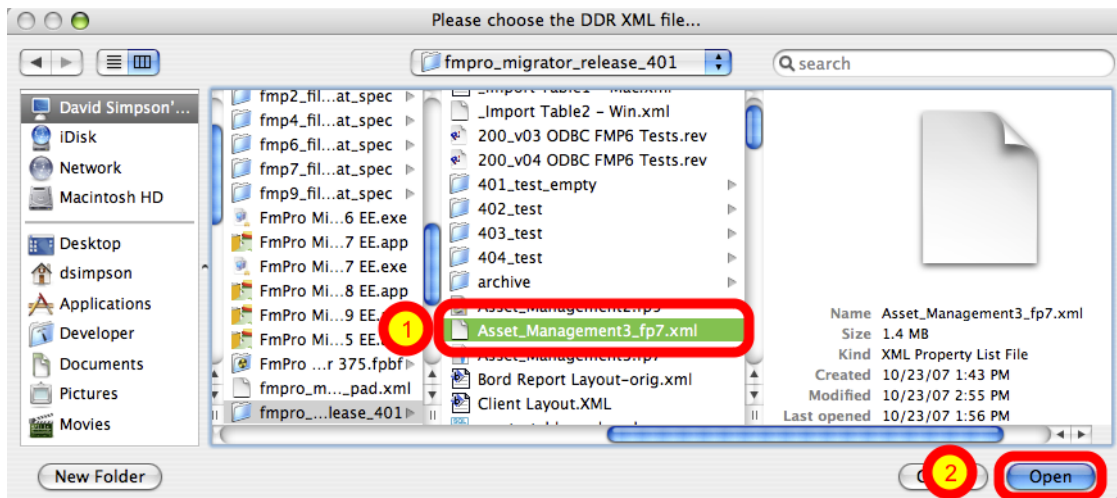


In the MacOSX System Preferences, click the Security & Privacy icon, (1) click the Accessibility item in the list, (2) click the lock button to allow changes, (3) click the checkmark next to the FmPro Migrator application.

**Note:** You will need to perform this unlocking process once, for each new version of FmPro Migrator is installed on your computer.

After allowing Accessibility access for FmPro Migrator, it will often be possible to immediately click on the same GUI batch processing button in FmPro Migrator to proceed with the migration. Sometimes you need to re-launch FmPro Migrator after making this change, and it can be helpful to close the System Preferences window to insure that MacOSX has recognized the change.

## Step 2 - Select the DDR XML File



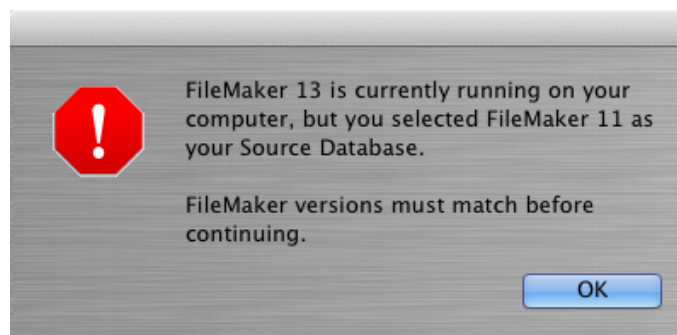
(1) Select a DDR XML file, (2) click the Open button.

**Note:** FmPro Migrator Developer Edition requires a Database Design Report as a DDR XML file created with FileMaker Developer/Advanced 7+. This DDR XML file is used by FmPro Migrator to count the number of layouts which will be copied via the clipboard, as well as the order in which the layouts are found within the FileMaker database file. The DDR XML file should be re-created if any layouts have been deleted or moved within the file by using the Manage Layouts dialog.

This DDR XML file needs to be created with the same version of FileMaker Pro Advanced which is being used for the conversion project. The version of FileMaker Pro Advanced should remain the same during the entire project.

FmPro Migrator will loop thru all of the layouts and capture them via the clipboard by using AppleScript or WinAutomation to drive the FileMaker graphical interface.

## Step 2 - Troubleshooting - Mis-Matched FileMaker Versions Error [MacOSX]



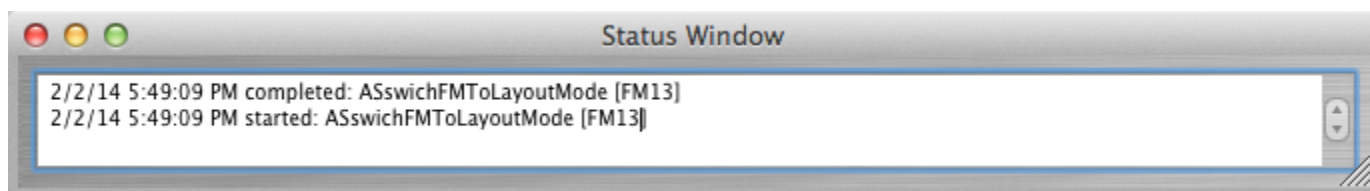
FmPro Migrator needs to have an accurate idea of which version of FileMaker Pro Advanced is currently running on your computer in order to use the correct AppleScript code to perform each task. This means that it is important to select the correct Source Database and Destination

Database on the main FmPro Migrator window.

If you are capturing layouts from FileMaker Pro Advanced, FmPro Migrator will verify that the source database version you selected is the actual version which is running on your computer [see dialog above].

If you are building objects in FileMaker Pro Advanced (i.e. value lists, custom functions, scripts, layouts), then FmPro Migrator will verify that the destination database version you selected is the actual version which is running on your computer.

## Step 2 - Troubleshooting - Locks Up with No Processing Completed [MacOSX]



If a batch processing task starts, but appears to lock up with no activity, then force quit and re-launch FmPro Migrator from the Finder and open the Status Window by selecting the File -> Status Window menu item. Click the batch processing button again, and review the results in the Status Window.

FmPro Migrator will continue to log the results of its AppleScript processing handlers in the Status Window as long as the window is open.

As shown above, the ASswitchFMTtoLayoutMode handler has run, but nothing else has completed. This type of problem can be due to having 2 copies of FileMaker Pro Advanced open at the same time. Some of the commands might get sent to one copy of FileMaker and other commands could get sent to the 2nd copy of FileMaker.

If a single copy of FileMaker Pro Advanced is running, but the FileMakerDB.fp7/FileMakerDB.fmp12 file is not open, a similar problem might also occur.

For more details about the tasks performed by the AppleScript handlers, please see the Troubleshooting AppleScript GUI Automation at the end of this document.

# Step 3 - Import Scripts

## Step 3 - Import Scripts

### Step 3 - Scripts Tab

Migration Process

Tables Relationships TOs Layouts **Scripts** GUI

Scripts: 4

Script Name	Steps
----- Address Book	0
Address Book - Import from AB2FM File (Admin)	17
Address Book - Add Group	9
Perform Script - File Reference Example	3

Script Text:

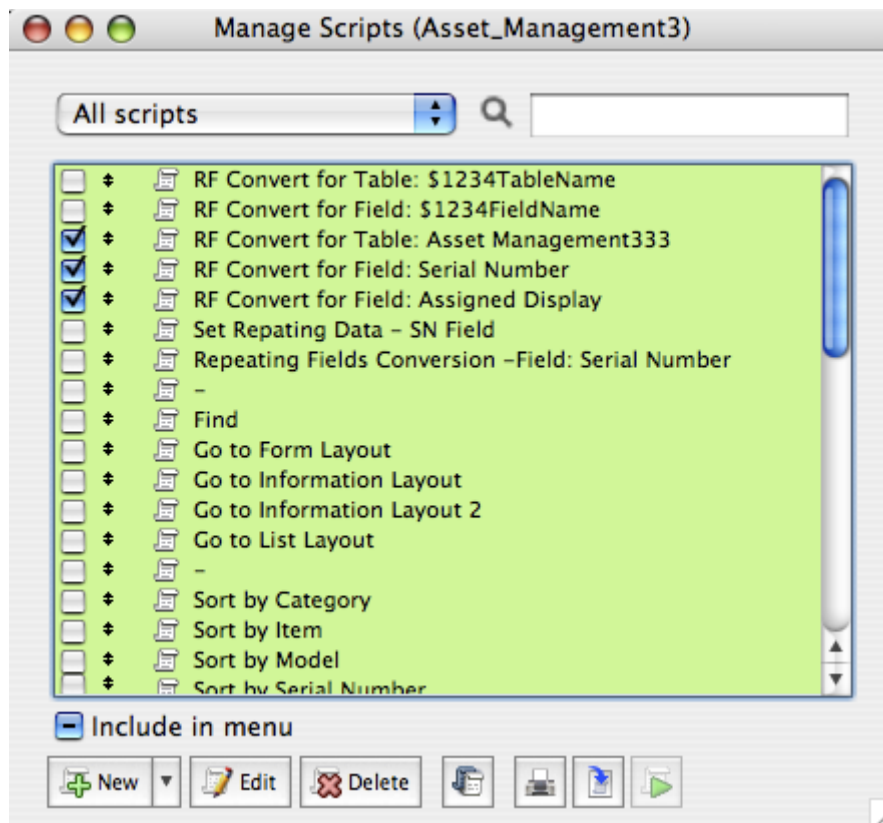
```
3 This script imports records from the AB2FM FileMaker file which was used to import records into the database. It requires: AB2FM.fp7 file in same directory as this INS database file, for importing purposes.  
# Access - Admin only  
# 11/4/2006 - David Simpson - .com Solutions Inc. - www.fmpromigrator.com - Initial Release  
#  
#  
Go to Layout [ "Address_Book" ]  
# Import Data - Look for AB2FM File in same folder (or prompt?)  
Import Records [ Source: "file:AB2FM.fp7"; Target: "Address_Book"; Method: Update matching records ]  
Source field 2 import to Address_Book::phonetic last name  
Source field 3 import to Address_Book::name  
Source field 4 import to Address_Book::organization  
Source field 5 import to Address_Book::department  
Source field 6 import to Address_Book::suffix  
Source field 7 import to Address_Book::note  
Source field 8 import to Address_Book::last name  
Source field 9 import to Address_Book::phonetic first name  
Source field 10 import to Address_Book::company  
Source field 11 import to Address_Book::birth date month  
Source field 12 import to Address_Book::birth date day  
Source field 13 import to Address_Book::birth date year  
Source field 14 import to Address_Book::title  
Source field 15 import to Address_Book::job title  
Source field 16 import to Address_Book::nickname
```

.com Solutions Inc.

The Scripts tab provides an interface for managing ScriptMaker Scripts during a migration project. Scripts can be captured [1] from FileMaker by copying them to the Clipboard within the ScriptMaker window within FileMaker Advanced. Once the scripts have been updated by FmPro Migrator, they can then be copied to the Clipboard [2] and pasted back into FileMaker Advanced.

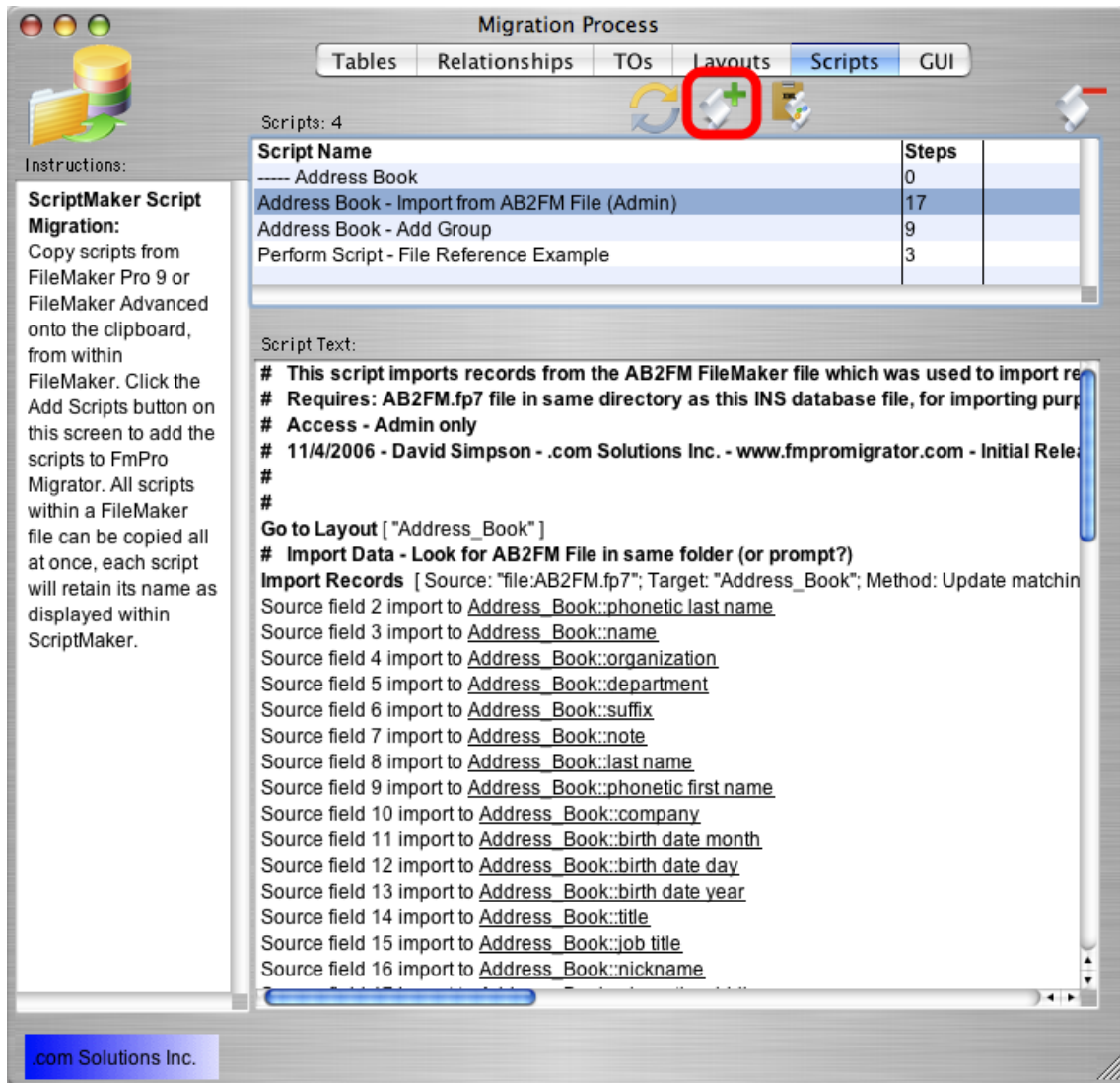
Scripts can also be viewed within FmPro Migrator by clicking each script and viewing the text version of the script within the Script Text field [3]. Duplicate or unneeded scripts can also be deleted by selecting the scripts and clicking the Delete button [4].

### Step 3 - Copy ScriptMaker Scripts From FileMaker to ClipBoard



Within FileMaker Advanced, open the ScriptMaker window, select all of the scripts and select Copy from the Edit menu.

## Step 3 - Paste Scripts From ClipBoard Into FmPro Migrator



Click the Add Scripts from ClipBoard button to paste the scripts into FmPro Migrator.

You will perform this task for every source database file which contains scripts to be added to the destination database file.

### Troubleshooting - Duplicate Script Names

Within a multi-file solution, it is possible to have the same script located within several files. However duplicate scripts are not allowed within the new single file FileMaker database. When duplicate scripts are pasted into the new consolidated database file, the duplicated scripts will be renamed with an incrementing numeric suffix. There are a couple of options available to solve this situation.

#### Option #1:



If duplicated scripts are have the same functionality, just delete the duplicated scripts within the FmPro Migrator scripts window.

**Option #2:**

If the duplicated scripts contain different functionality, they should be renamed before copying either the scripts or layouts into FmPro Migrator. One technique could be to add a file referencing prefix to each duplicated script to indicate the source of the script. This will insure that references between the scripts and layout objects will be maintained throughout the migration process.

# **Step 4 - Import Relationships, TOs, Value Lists & Custom Functions**

## Step 4 - Import Relationships, TOs, Value Lists and Custom Functions

### Step 4 - Relationships Tab

The screenshot shows the 'Migration Process' window with the 'Relationships' tab selected. The window title is 'Migration Process' and it has tabs for 'Tables', 'Relationships', 'TOs', 'Value Lists', 'CFs', 'Layouts', 'Scripts', and 'GUI'. The 'Relationships' tab shows a list of 77 relationships. The list has columns for 'Left Table Name', 'Right Table Name', 'Count', and 'Relationship SQL'. The first row is highlighted. Below the list, there are configuration options for 'Left Table' and 'Right Table', including 'Table', 'Cascade Create', and 'Cascade Delete'. At the bottom, there is a 'Predicates' section with a table showing 'Left Table::Field', 'Join Type', and 'Right Table::Field'. Red circles highlight various UI elements: 1-6 in the relationship list, 7-12 in the configuration area, and 13-15 in the predicates list.

Left Table Name	Right Table Name	Count
Client	Client_Call_L	0
Client	Horse	2
Client	Policy	2
Underwriter	Policy	1
Underwriter_Self_Join	Underwriter	1

Left Table::Field	Join Type	Right Table::Field
Client::Client_ID	=	Horse::Client_ID
Client::Logic_1_field	<>	Horse::Inactive

FmPro Migrator imports Relationships, Table Occurrences, Value Lists and Custom Functions by reading the contents of the DDR XML file for each source database file.

Click the Relationships tab on the Migration Process window.

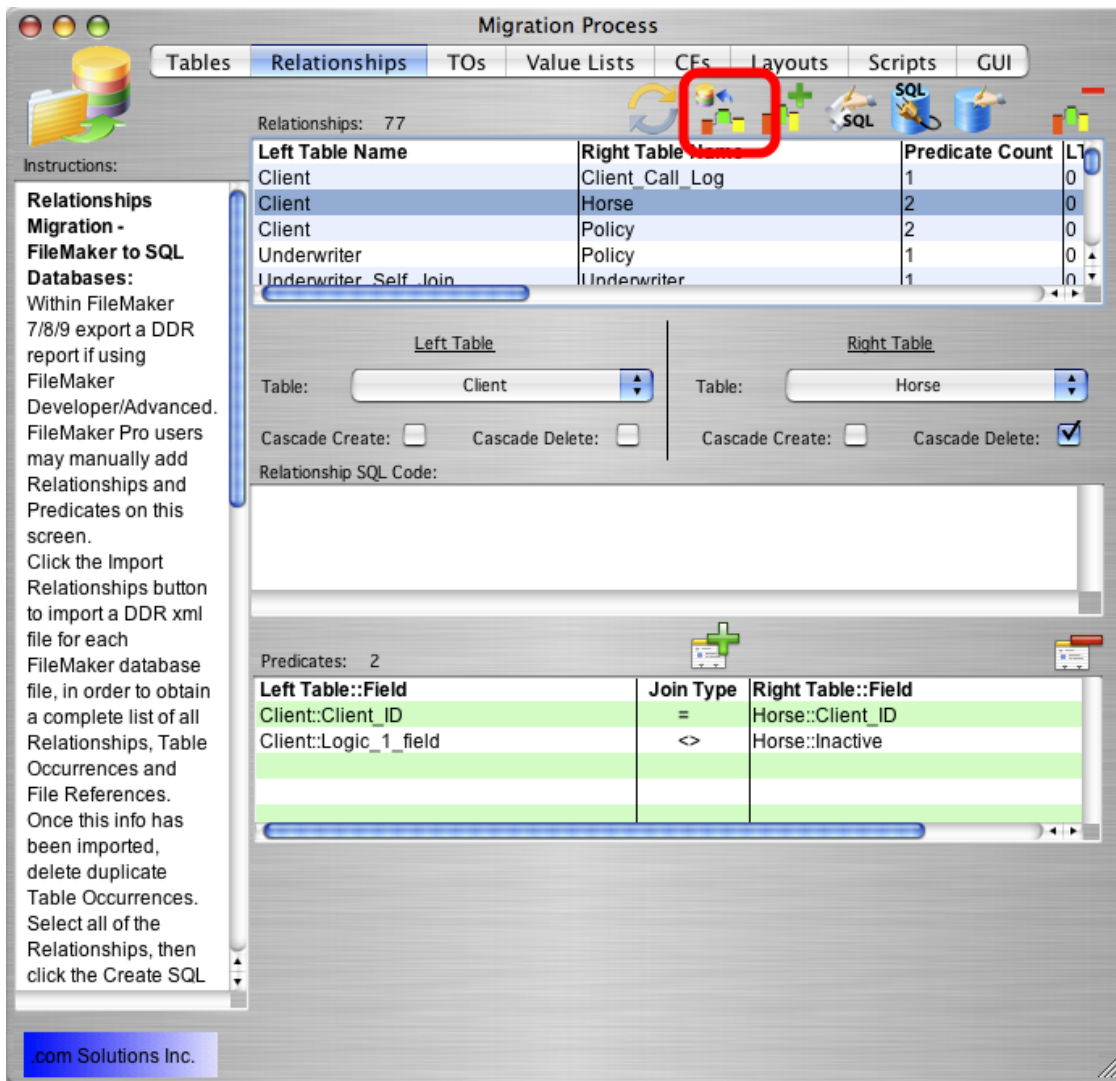
The Relationships tab provides an interface for managing the relationships which will be migrated into the consolidated database file.

It won't generally be necessary to make changes on either the Relationships or TOs tab when performing a Table Consolidation project. But changes made on both of these tabs and will be reflected within the destination database file.

Tasks which can be performed on the Relationships tab include:

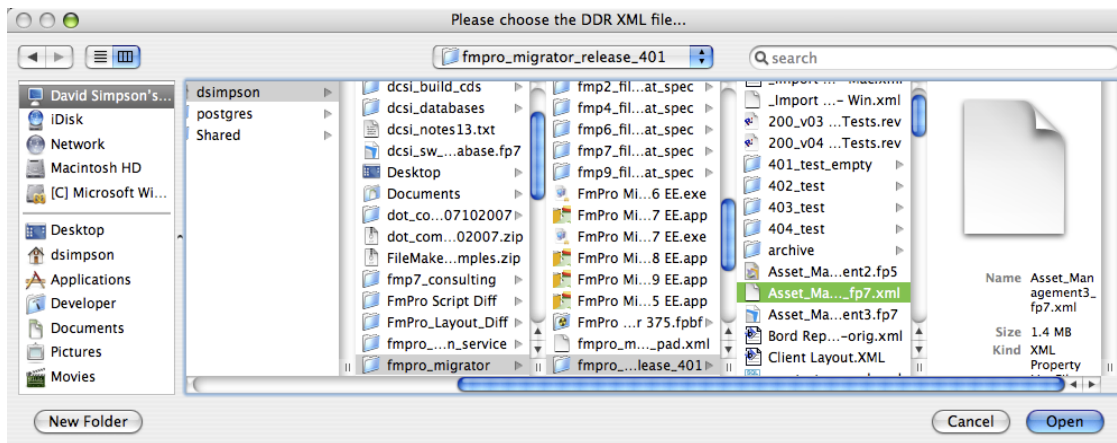
- [1] importing Relationships, TOs, Custom Functions and Value Lists from a DDR XML file,
- [2] manually adding one relationship at a time,
- [3] generating SQL code to represent a relationship in a SQL database (not used for Table Consolidation projects),
- [4] creating relationships within a SQL database (not used for Table Consolidation projects),
- [5] creating relationships within a FileMaker Advanced database,
- [6] deleting selected relationships,
- [7] changing Left Table name,
- [8] changing Left Table Cascade Create status,
- [9] changing Left Table Cascade Delete status,
- [10] changing Right Table name,
- [11] changing Right Table Cascade Create status,
- [12] changing Right Table Cascade Delete status,
- [13] manually changing relationship predicate parameters,
- [14] manually adding a relationship predicate,
- [15] manually deleting a selected relationship predicate.

## Step 4 - Click Import Relationships Button



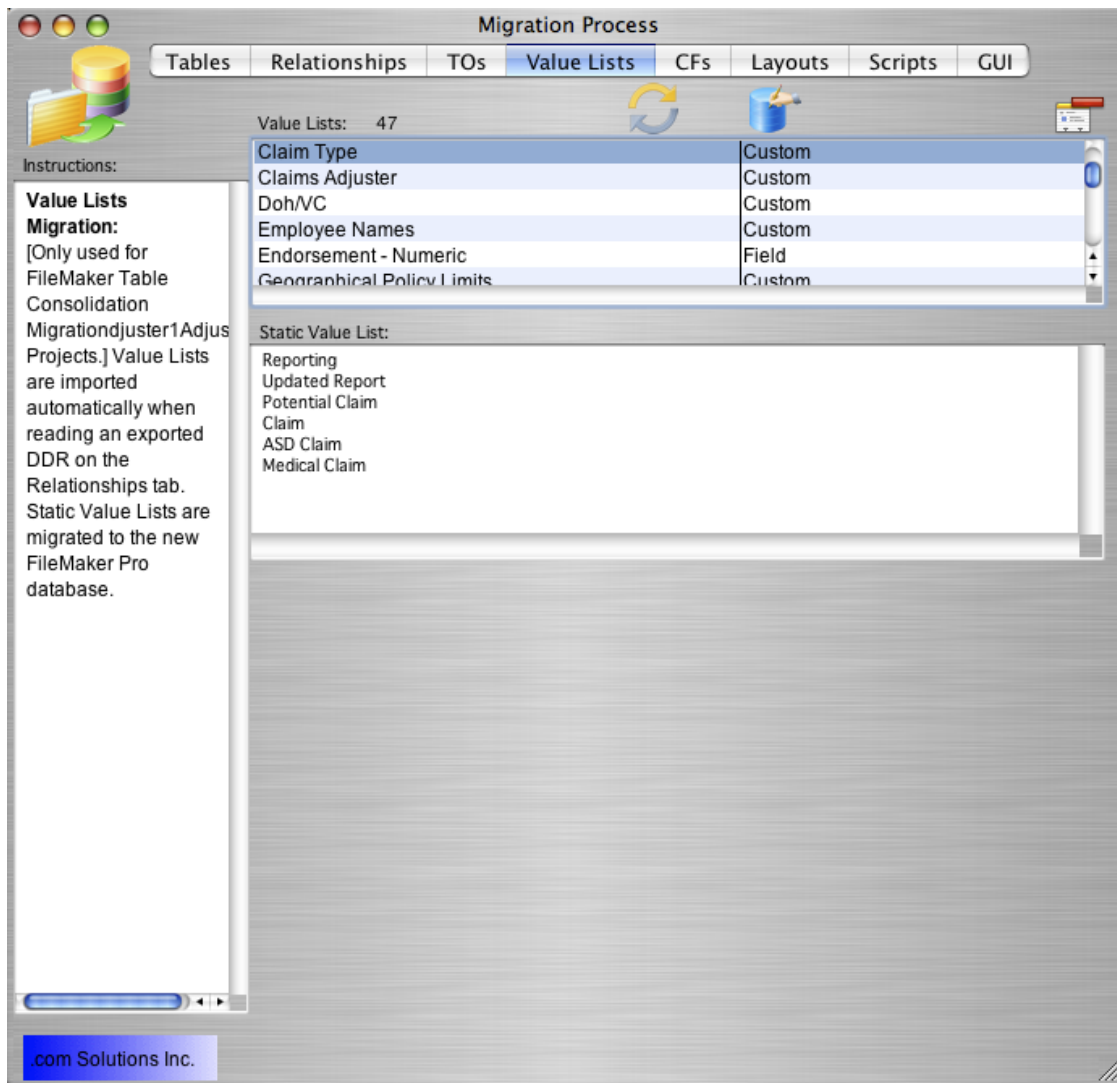
Click the Import Relationships button to import relationships from the DDR XML file for the source database file.

## Step 4 - Select DDR XML File



The relationships will be imported from the DDR file and added to the list of relationships on the Relationships tab. TOs, Value Lists and Custom Functions will also be imported from the DDR XML file at the same time, and displayed under the Value Lists and CFs tabs. A dialog will be displayed if there are no Custom Functions or Value Lists found within the DDR XML file.

## Step 4 - Value List - Details



Clicking on the Value Lists tab displays the Value Lists which were imported from the DDR XML file. Value Lists shown with a type of Custom, are static value lists having a manually edited list of values (as shown in this screenshot). Value Lists displayed with a type of Field, are dynamically generated value lists which get their value from a field within the database. Static value list items can be manually edited by clicking on the Value List and changing the values. Changes are saved automatically when clicking out of the Static Value List field.

## Troubleshooting - Duplicate Value List Names

Within a multi-file solution, it is possible to have the same value list located within several files. However duplicate value lists are not allowed within the new single file FileMaker database. When duplicate value lists are created in the new consolidated database file, the duplicated value list will be renamed with an incrementing numeric suffix. There are a couple of options available to solve this situation.

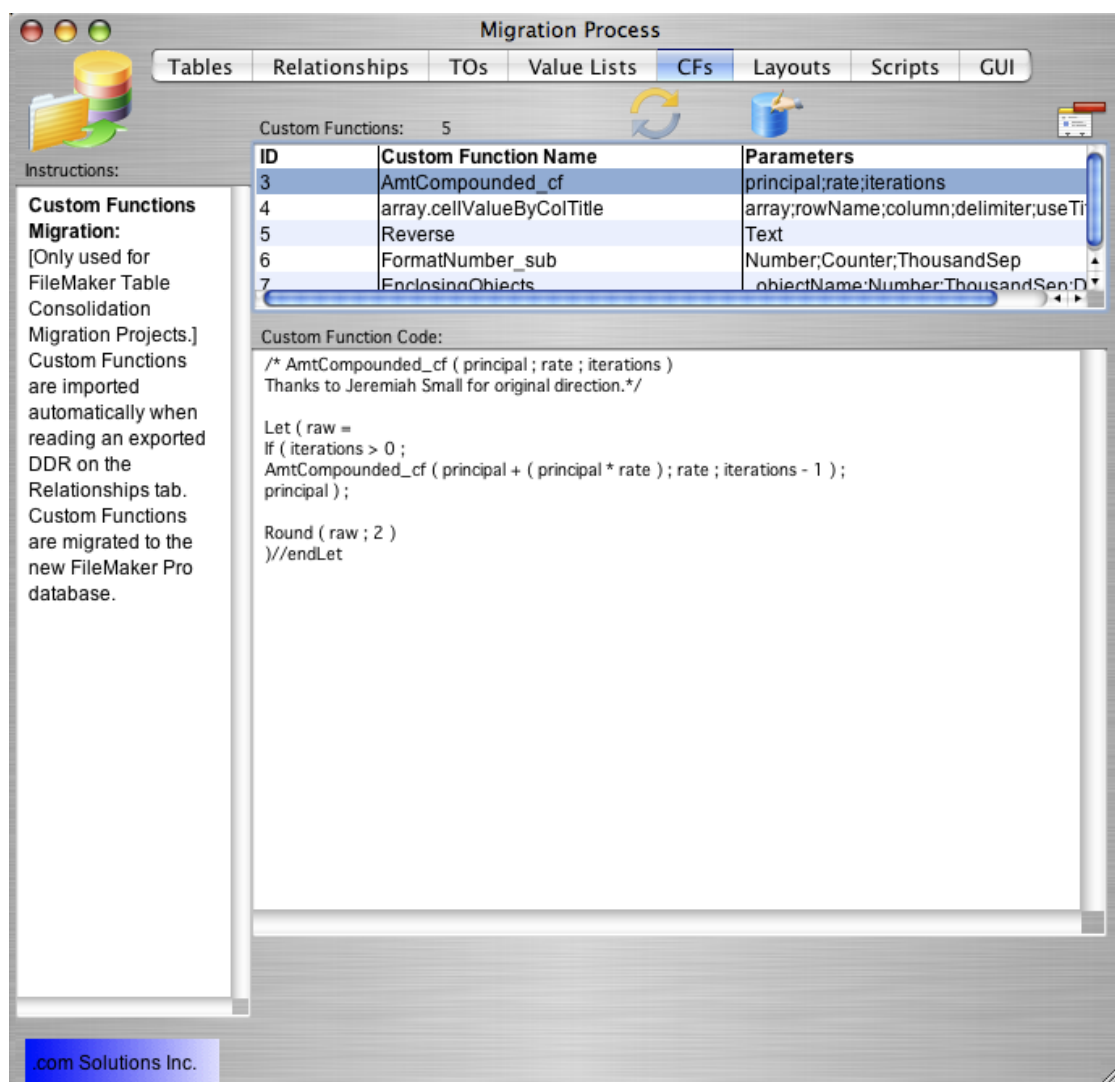
### Option #1:

If duplicated value lists contain the same values, just delete the duplicated value lists within the FmPro Migrator Value Lists window.

### Option #2:

If the duplicated value lists contain different values, they should be renamed before copying either the value lists or layouts into FmPro Migrator. One technique could be to add a file referencing prefix to each duplicated value list to indicate the source of the value list. This will insure that references between the value list and layout objects will be maintained throughout the migration process.

## Step 4 - Custom Function - Details



Clicking on the CFs tab displays the Custom Functions which were imported from the DDR XML file. Custom Functions and their parameter lists are displayed.

The Custom Function Code can be manually edited by clicking on the Custom Function and changing contents of the Custom Function Code field. Changes are saved automatically when



clicking out of the Custom Function Code field.

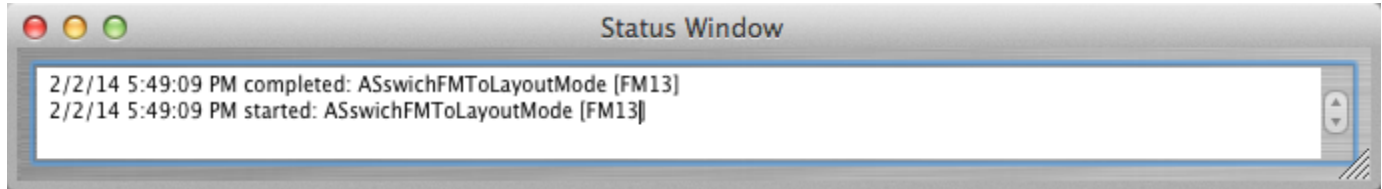
# Troubleshooting

## Troubleshooting AppleScript GUI Automation [MacOSX]

---

FmPro Migrator uses embedded AppleScript code to automate importing Layouts, and building Custom Functions, Value Lists, Table Occurrences, Relationships, Scripts and Layouts in FileMaker database files. Detailed info about this AppleScript code is included in this section of the manual.

### Troubleshooting - Locks Up with No Processing Completed [MacOSX]



If a batch processing task starts, but appears to lock up with no activity, then force quit and re-launch FmPro Migrator from the Finder and open the Status Window by selecting the File -> Status Window menu item. Click the batch processing button again, and review the results in the Status Window.

FmPro Migrator will continue to log the results of its AppleScript processing handlers in the Status Window as long as the window is open.

As shown above, the ASswitchFMToLayoutMode handler has run, but nothing else has completed. This type of problem can be due to having 2 copies of FileMaker Pro Advanced open at the same time. Some of the commands might get sent to one copy of FileMaker and other commands could get sent to the 2nd copy of FileMaker.

If a single copy of FileMaker Pro Advanced is running, but the FileMakerDB.fp7/FileMakerDB.fmp12 file is not open, a similar problem might also occur.

### Troubleshooting - List of AppleScript Handlers

FmPro Migrator uses the following handlers containing dynamically compiled AppleScript code:

**ASCheckFMVersion** - check the currently running FileMaker version to make sure it matches the selected version

**AScheckGUIScriptingMode** - Checks to determine if GUI Scripting is enabled for use by AppleScript. If false, it will issue the command to enable MacOSX Accessibility features on MacOSX 10.8 or below. For MacOSX 10.9, the user will be prompted to open System Preferences to enable Accessibility for FmPro Migrator.

**AScloseFMCustomFunctionsWindow** - Closes the "Manage Custom Functions..." window by

clicking the Ok button.

**AScloseFMManageScriptsWindow** - Closes the “Manage Scripts...” window by clicking on the close button, selects first script in list in preparation for creating the next script.

**AScloseFMRelationshipGraphWindow** - Sends a return keystroke to close the FileMaker Relationship Graph window.

**AScloseFMValueListsWindow** - Closes the “Manage Value Lists...” window by clicking the Ok button.

**AScopyFMLayout** - Copies the current FileMaker layout into FmPro Migrator.

**AScopyLayout2FM** - Creates a new empty layout in FileMaker (Pass #1), sets layout name, sets TO Name for new layout, deletes layout header, footer parts, pastes the layout XML already on the clipboard, clicks Yes to extend objects dialog #1, #2.

**AScreateFMCustomFunction** - Reads Custom Function creation parameters from the Clipboard. Creates a new FileMaker Custom Function, having the proper name, parameters and code, then saves it.

**AScreateFMCustomValueList** - Creates a FileMaker custom value list and saves it. Field based value lists are also created, but the properties need to be set manually.

**AScreateFMRelationship** - Reads relationship creation parameters from the Clipboard, storing info in 10 variables, creates a new relationship or a predicate for an existing relationship which is already being worked on.

**AScreateFMTO** - Gets the TOName and Base Table Name from the Clipboard then creates a new FileMaker TO. Note: Setting the TO color is not supported.

**ASdeleteFMLayoutParts** (function) - Deletes the Header and Footer layout parts from a new empty layout, leaving the Body part.

**ASgetTOListPosition** (function) - Returns the position number of a TO name within a list of TOs. The returned value is used for selecting a specific line (TO name) within a popup menu. This needs to be done with a numeric index to the TO name instead of passing in the actual TO name via keystrokes.

**ASglobalsAndHandlers** (function) - returns the AppleScript code having the global variables and commonly used handlers

**ASgotoNextFMLayout** (function) - Advances to the next FileMaker layout while in layout mode, by

sending the control down arrow keystroke.

**ASopenFMCustomFunctionsWindow** - Opens the "Manage Custom Functions" window using the File menu.

**ASopenFMEditRelationshipWindow** - Opens Edit Relationship window using command shift r keystroke, waits for window to open.

**ASopenFMEditRelationshipWindow** - Opens Edit Relationship window, waits for window to open.

**ASopenFMManageScriptsWindow** - Opens the "Manage Scripts..." window using the Command Shift S keystroke.

**ASopenFMManageTablesWindow** - Opens "Manage Database" window, clicks on the Tables tab.

**ASopenFMRelationshipGraphWindow** - Opens "Manage Database" window, clicks on the Relationship Graph tab. NOTE: Doesn't work with 8,9,10 because window doesn't refresh when Relationship Graph tab is clicked.

**ASopenFMValueListsWindow** - Opens the "Manage Value Lists..." window using the File menu."

**ASpasteFMScript** - Clicks the first row of the scripts displayed in the "Manage Scripts..." window. Pastes the current script XML on the clipboard using Command V keys.

**ASreplaceLayout2FMPPass2** - Selects the contents of the current layout in Layout mode, deletes the contents, pastes the layout contents already on the clipboard. Used during Pass #2 processing during Table Consolidation projects.

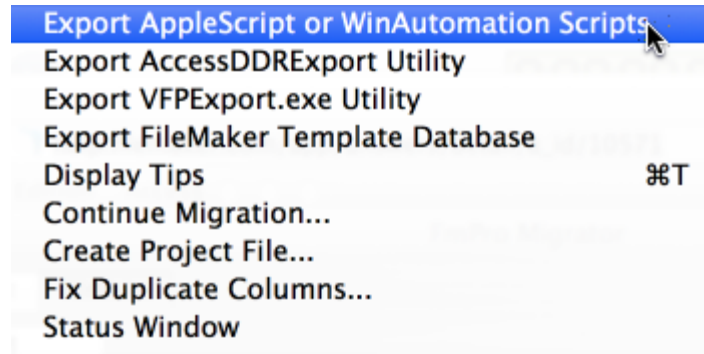
**ASsaveFMRelationship** - Sends keystroke return key to click the Ok button of the Edit Relationship window to save the current relationship.

**ASsaveFMRelationship** (function) - Sends return key keystroke to click the Ok button of the Edit Relationship window to save the current relationship. Sends another return key keystroke if the Rename TO dialog is displayed.

**ASswichFMTobrowseMode** - Switches to Browse Mode using Command b keys.

**ASswichFMToLayoutMode** - Switches to Layout Mode, using Command L keys, advances to layout #1.

## Export AppleScript Code



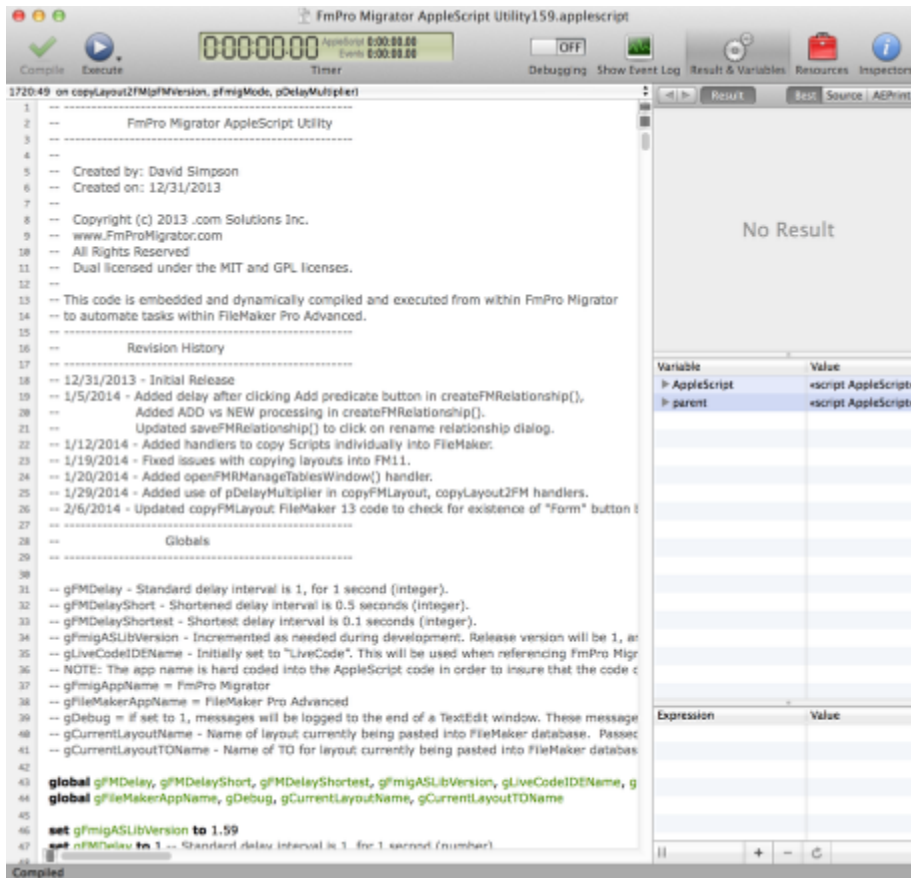
On MacOS X, the AppleScript code is embedded within FmPro Migrator therefore no installation is required.

It may be helpful to review the AppleScript source code for troubleshooting purposes, therefore it can be exported to the output directory by selecting the File -> Export AppleScript or WinAutomation Scripts menu.

Selecting this menu will save a text file of AppleScript code, a LiveCode AppleScript test stack and a LiveCode performance testing stack to the output directory.

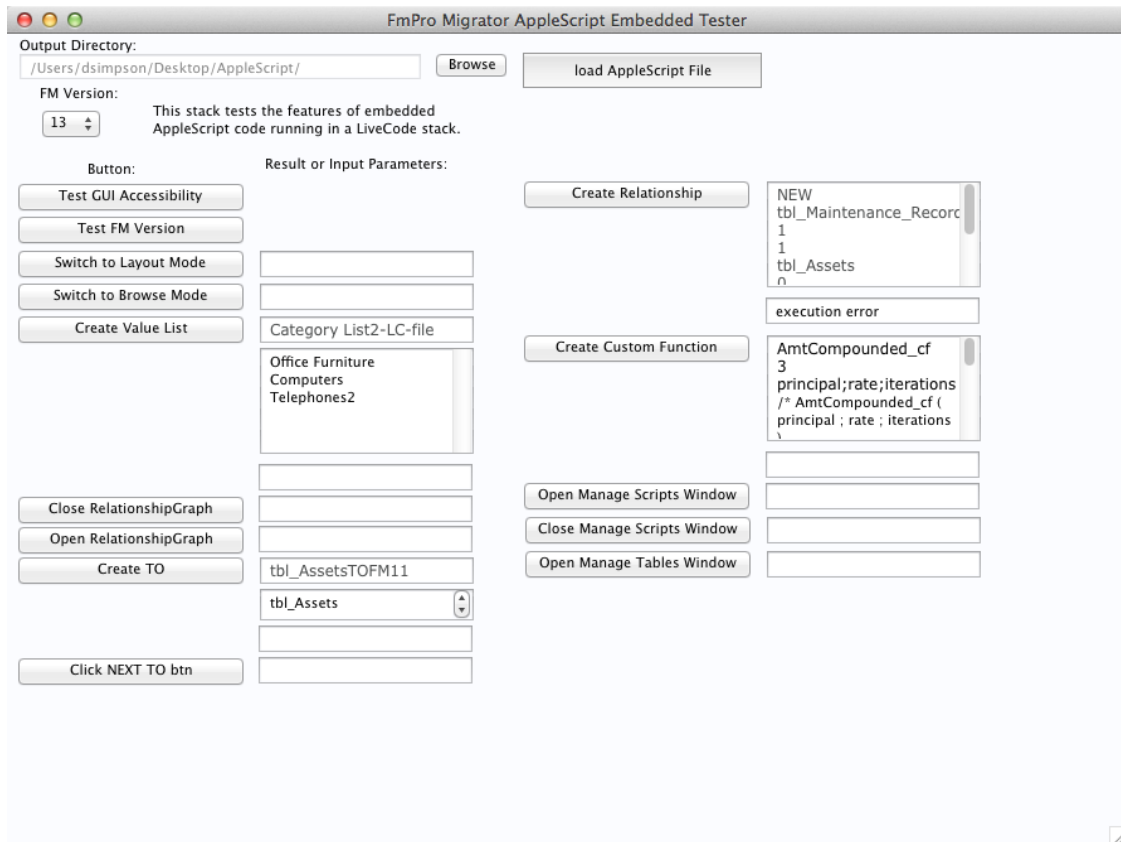
These files have been released as open source for use by the FileMaker community.

## FmPro Migrator AppleScript Utility???.applescript File



The AppleScript code embedded within FmPro Migrator follows the naming strategy used within the FmPro Migrator AppleScript Utility???.applescript text file.

## FmPro Migrator AppleScript Embedded Tester.livecode Stack



Many of the AppleScript handlers embedded within FmPro Migrator have been converted into LiveCode handlers within the FmPro Migrator AppleScript Embedded Tester stack file. This stack file serves as a test harness for testing the embedded AppleScript code, prior to implementing it within FmPro Migrator. This stack is also released as open source software.

Download a copy of LiveCode from the [www.livecode.com](http://www.livecode.com) website to open this stack file.

Right click the mouse on any of the buttons to see the code which is being called.

Select the LiveCode Object -> Stack Script menu item to see the stack level handlers.