



.com Solutions Inc.

# **FmPro Migrator - FileMaker to MySQL Migration Procedure**

# FmPro Migrator - FileMaker to MySQL Migration Procedure

## 1

1.1	FmPro Migrator - FileMaker to MySQL Migration Procedure	4
1.2	Licensing FmPro Migrator	6
1.3	Pre-Migration Preparation	10
1.4	Pre-Migration Task - Delete Unstored Calc Fields	19
1.5	Step 1 - Get Info	25
1.6	Step 2 - Get Fieldsize	40
1.7	Step 3 - Create Table	43
1.8	Step 4 - Transfer Data - MySQL	46
1.9	Step 5 - Convert Data (Repeating Fields)	54
1.10	Step 6 - Create Table (Repeating Fields)	59
1.11	Step 7 - Transfer Data (Repeating Fields) - MySQL	60
1.12	Image Transfer (Optional Step)	62
1.13	MySQL Troubleshooting	66
1.14	Migrating Relationships - FileMaker to MySQL	67



# FmPro Migrator - FileMaker to MySQL Migration Procedure

---

This document provides an explanation of the steps required to perform a FileMaker to MySQL Migration project using FmPro Migrator Developer Edition or FmPro Migrator Platinum Edition. This manual also has enough details for converting FileMaker Pro (and other databases) to PostgreSQL databases. As of FmPro Migrator 11.62 - the latest versions of PostgreSQL are fully supported.

## Documentation Conventions:

FileMaker 7+ - refers to FileMaker Pro or FileMaker Pro Advanced version 7 or higher.

FileMaker 5/6 - refers to FileMaker Pro versions 5, 5.5 or 6.0

MySQL - refers to MySQL database server versions - All Versions

MariaDB - All Versions are fully supported.

PostgreSQL - All Versions are fully supported.

**Note:** This manual does not cover the steps required to create a [PHP web application](#), [.NET application](#), [LiveCode](#) stack or [Servoy application](#) from the FileMaker database. These tasks are covered in other manuals. This manual covers only the process involved with transferring the data from FileMaker to MySQL.

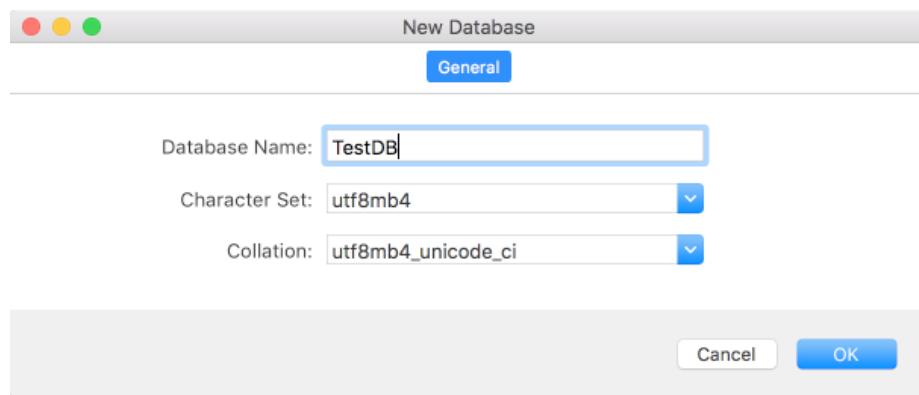
Document Version 9

1/26/2026

FmPro Migrator 11.73

Updated MySQL 8+ compatibility info, and notes about using this manual for PostgreSQL as well as MySQL.

## MySQL & Unicode



As of FmPro Migrator 8.32, FmPro Migrator always uses a the UTF8mb4 character set between the FileMaker and MySQL databases. This change insures that all non-latin characters for all languages will be transferred correctly to MySQL.

Your MySQL database should be configured with a default character set of UTF8mb4. If you are using Navicat ([www.navicat.com](http://www.navicat.com)) to manage your MySQL database, then select Database Properties from the Database menu to make this change.

**Note:** If you select a Default Collation of **utf8\_general\_ci** (shown here) instead of the default value of **utf8\_bin**, then MySQL will automatically perform case insensitive database searches.

## Licensing FmPro Migrator

---

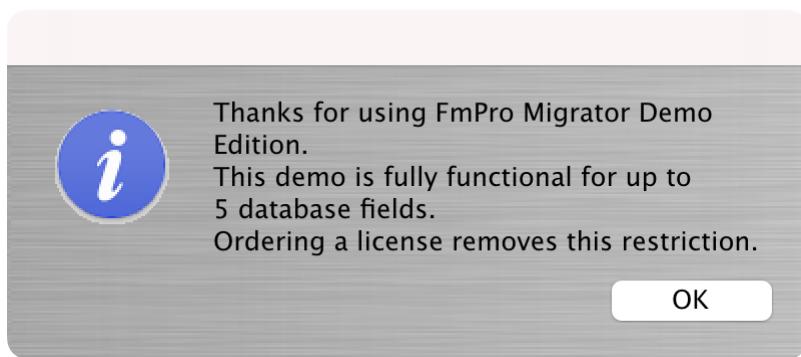
As of FmPro Migrator v11, all FmPro Migrator downloads are fully functional with features unlocked via a single license key.

This section of the manual shows how to enter the license key to unlock the features within FmPro Migrator.

FmPro Migrator 11.51

9/16/2025

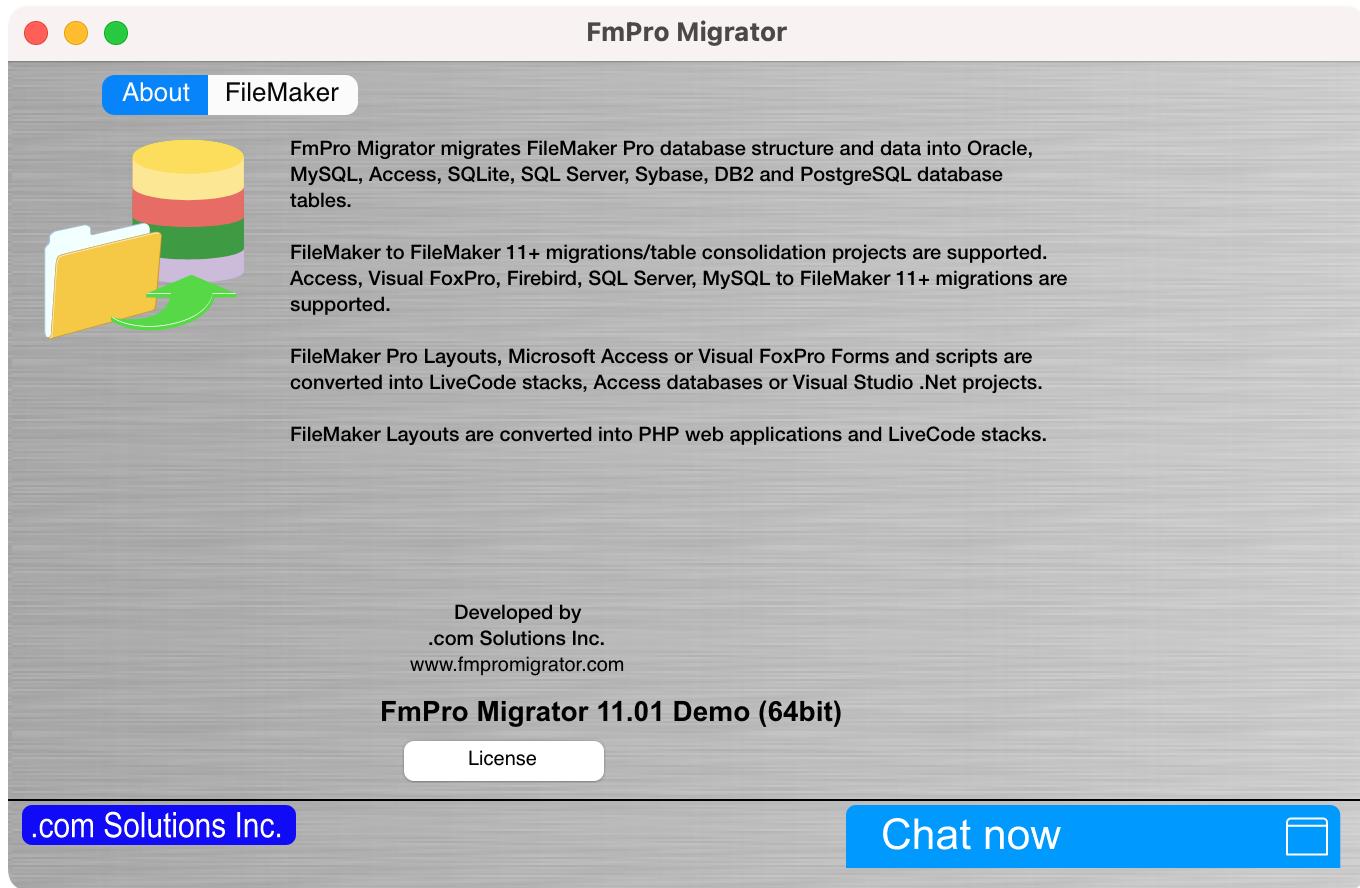
### Demo Edition Dialog



When launched the first time, FmPro Migrator will be running in Demo mode as shown in this screenshot. Clicking the Ok button opens the order page of the website.

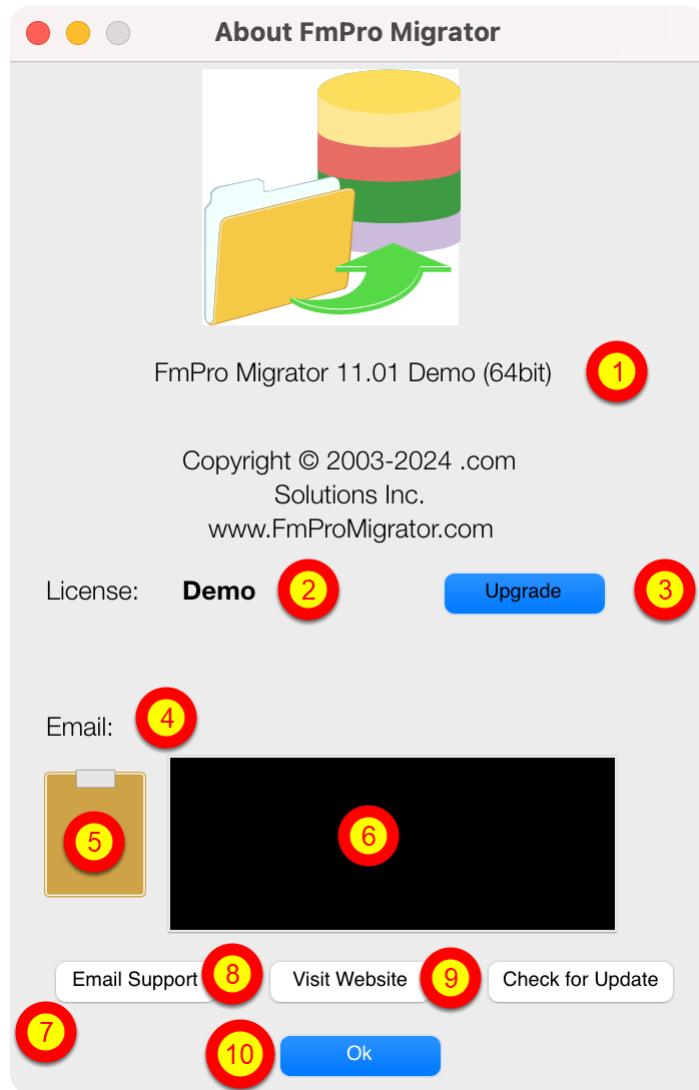
In addition to transferring data for 5 database fields, the conversion features shown on the GUI tab of the Migration Process window will convert 5 layouts or forms/reports & scripts.

## Demo Mode - About Tab



Clicking the "License" button on the About tab, or selecting the Help -> License/About menu items will open the About window where you can enter the license key.

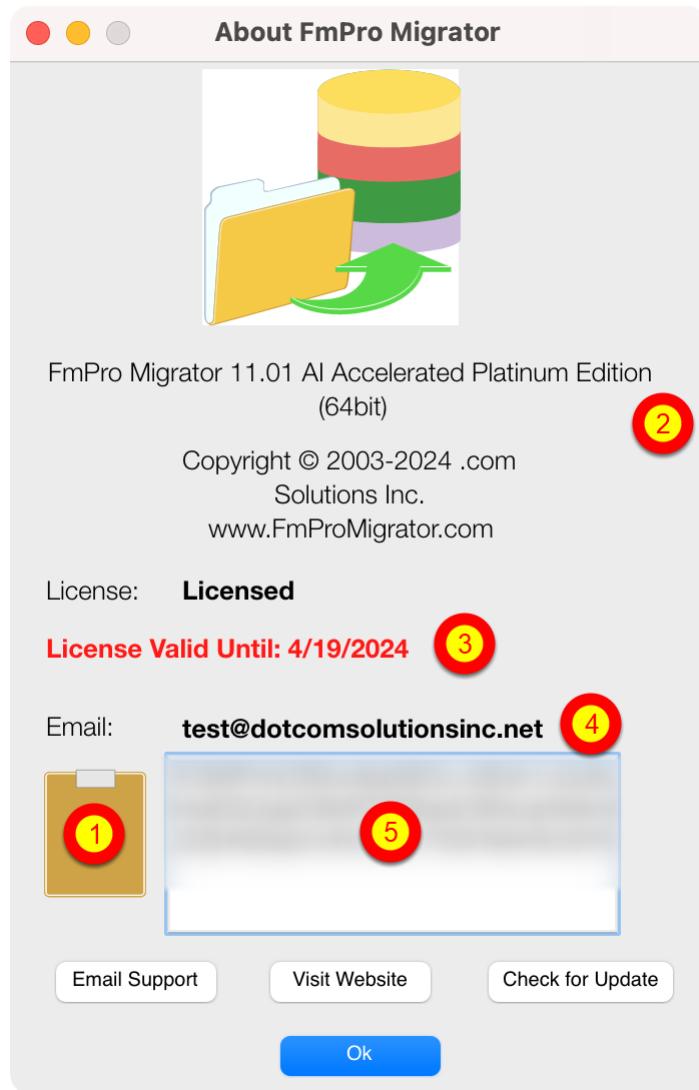
## About/License Window



On the About FmPro Migrator window:

(1) Product name and version licensed, (2) Demo or Licensed will be displayed, (3) The Upgrade button opens the website order page, (4) Email address associated with your license key, (5) Clipboard button reads the license key from the clipboard, (6) License key field, (7) Email Support button opens the Contact form on the website, (8) Visit Website opens the order page in Demo mode or the product page in Licensed mode, (9) Check for Update opens the downloads page with the latest software version, (10) Ok button closes the About window.

## Pasting License Key



Your product license key will be displayed in your web browser when your order has been completed. It will also be sent via email at the same time, please check your SPAM folder if the email doesn't arrive in a few minutes.

(1) Copy the license key to the clipboard and click the clipboard icon. Once validated (2) the product name will change, (3) the License validation date will be updated, (4) your email address will be displayed, and (5) the license key will be displayed in the field under the Email address when clicking inside the field. In this screenshot the license key has been obscured.

**Note:** The license key field also allows directly pasting the license key instead of clicking the clipboard icon.

And that is all you need to do to license all of the features of FmPro Migrator.

## Pre-Migration Preparation

---

The following steps are recommended prior to starting a FileMaker Pro to SQL database migration project with FmPro Migrator.

The first part of this process generally involves transferring the data into a SQL database. And for that part of the process you may need to delete the Unstored Calc, Global and Summary fields to make that part of the process more manageable.

But if you plan to convert the graphical interface and scripts later on, you would want to try to preserve as many of those fields as possible. Some databases only handle a limited number of columns (Microsoft Access is limited to 255 fields), so some deletion of fields may still be required.

You could transfer the data as one part of the process using a limited number of records, then use that smaller copy of the database having all of the fields in order to perform the GUI conversion (including layouts and scripts).

### **FileMaker Pro 2,3,4 - File Preparation**

There is no ODBC driver for FileMaker Pro 2,3,4 files, and the ODBC driver is required for transferring data from FileMaker Pro into other databases.

These older FileMaker Pro database files should be upgraded to FileMaker Pro 6 prior to performing the migration.

The software download page for the licensed version of FmPro Migrator includes a FREE download link for FileMaker Pro 6 in order to upgrade older files.

1) Select the Help -> Check for Updates... menu item within FmPro Migrator.

The FmPro Migrator licensed software download page will be displayed in your web browser.

2) Download and install the FileMaker Pro 6 Trial Software for your computer operating system.

3) Launch the FileMaker Pro 6 software.

4) Drag & Drop the FileMaker Pro 2,3,4 files onto the FileMaker Pro 6 software window.

Each file will be converted to a new file having the .fp5 file extension used by FileMaker Pro 6.

5) Proceed to the next step - FileMaker Pro 5,6 File Preparation.

### **FileMaker Pro 5,6 - File Preparation**

1) Make a copy of each of your FileMaker Pro files. Changes will need to be made within each file and you always want to be able to return to your original copy in case a problem occurs.

Select File -> Save a CopyAs... -> compacted copy (smaller)

2) Rename each saved filename so that it contains no spaces or special characters. The filename gets used as a table name when it is used with the ODBC driver, and spaces and

special characters are not permitted. Therefore a filename which was originally:

Asset Management.fp5

should be changed to:

Asset\_Management.fp5

If you allowed FileMaker Pro to name the file when you saved a copy, you might have ended up with a filename like this:

Asset Management Copy.fp5

which should be changed to:

Asset\_Management.fp5

It is confusing to be dealing with lots of files named ??? Copy.fp5, so just remove the word "Copy" which was appended by FileMaker Pro when you saved the file.

3) Copy all of your files into a convenient location, like a new folder so that you can keep them together. You might name this folder: Modified\_Files in order to keep them separate from the original files and the empty copies of the files which will be created later. In fact, you will probably want to create a top-level folder to keep track of all of the files/folders which will be created during the migration project.

So you could create a folder named: FM\_Migration\_Project

This top-level folder can then serve as the Output Directory which you will select within FmPro Migrator.

Notice that each of the folders doesn't contain spaces. Under some (rare) circumstances FmPro Migrator may need to run migration scripts via the command line and this process only works if there are no spaces within any of the the folder names leading to the Output Directory.

Open each .fp5 file saved into the Modified\_Files folder and make the following changes.

4) Delete all ScriptMaker Scripts.

Select the menu:

Scripts -> ScriptMaker...

5) Delete all Relationships.

Select the menu:

File -> Define Relationships...

**Note:** Relationships are only converted to SQL databases for FileMaker Pro 7+ file versions. You can potentially upgrade each file into an .fp7 file if you need this feature.

6) Delete all Passwords.

Select the menu:

File -> Access Privileges -> Passwords

Make sure that the remaining un-password protected access provides Full Access to the entire

file, with all file options checked within the dialog box.

7) Delete all fields having the following types:

Global

Summary

Unstored Calculation

Select the menu:

File -> Define Fields...

It is necessary to delete these fields from large database files having more than a few thousand records because the amount of time required for FileMaker Pro to perform the calculations may exceed the ODBC driver timeout interval. If this problem occurs, no records will be transferred from FileMaker Pro to the destination database.

**Note:** You may leave the Stored Calculation fields, as the data may be valuable within the destination database.

It will take some work to delete recursive field dependencies. You could make one pass thru the fields from top to bottom to delete all of the fields which can be deleted, and then make another pass thru the fields list from the bottom up to the top of the list.

8) Once all of the unneeded objects have been deleted from each file - make an empty copy of each file. You could store the empty copy of each file within a folder named *Empty\_Files*.

This empty copy of the database files will be used for Step 1 of the migration process where you will Drag & Drop the empty files onto the Step 1 icon within FmPro Migrator. This will cause FmPro Migrator to read the structure of each file to start the migration process.

Select the menu:

File -> Save a CopyAs... -> clone (no records)

This empty copy of each file needs to have exactly the same name as the compacted copy you saved previously within the *Original\_Files* folder.

This empty copy of the file will be used by FmPro Migrator for gathering the field info from each file. Saving an empty copy cleans up the internal structure of the file and reduces the size of the file which needs to be read by FmPro Migrator and speeds up the process compared to reading a full copy of the file.

1) Use FileMaker Pro to export a DDR XML file from each .fp7 or .fmp12 file.

Starting with FileMaker Pro 17, the Advanced tools are included at no extra cost and are enabled by checking the "Use Advanced Tools" checkbox in the Preferences dialog. Previous FileMaker versions required a separate purchase of FileMaker Pro Advanced. For the purposes of brevity, this manual will use assume that you are using FileMaker Pro 17+ versions.

This step is only required if you want FmPro Migrator to export relationships, value lists, layouts from FileMaker Pro into the destination SQL database.

2) Make a copy of each of your FileMaker Pro files. Changes will need to be made within each file and you always want to be able to return to your original copy in case a problem occurs.

Select File -> Save a CopyAs... -> compacted copy (smaller)

Saving a compacted copy of the file cleans up the internal structure of the file and reduces the size of the file by removing the indexes within the file.

If you allowed FileMaker Pro to name the file when you saved a copy, you might have ended up with a filename like this:

Asset Management Copy.fp7/.fmp12

which should be changed to:

Asset Management.fp7/.fmp12

It can be confusing to deal with lots of files named ??? Copy.fp7/.fmp12, so just remove the word "Copy" which was appended by FileMaker Pro when you saved the file.

3) Copy all of your files into a convenient location, like a new folder so that you can keep them together. You might name this folder: Modified\_Files in order to keep them separate from the original copy of the files. In fact, you will probably want to create a top-level folder to keep track of all of the files/folders which will be created during the migration project.

So you could create a folder named: FM\_Migration\_Project

This top-level folder can then serve as the Output Directory which you will select within FmPro Migrator.

Notice that each of the folders doesn't contain spaces. Under some (rare) circumstances FmPro Migrator may need to run migration scripts via the command line and this process only works if there are no spaces within any of the the folder names leading to the Output Directory.

Open each .fp7/.fmp12 file saved into the Modified\_Files folder and make the following changes.

4) Delete all Script Workspace Scripts.

Select the menu:

Scripts -> Script Workspace...

5) Delete all Relationships and non-base table TOs on the RelationshipGraph

Select the menu:

File -> Define/Manage Database...

Click the Relationships tab.

**Note:** If it is necessary to change the names of TOs to match base table names, while maintaining the relationship structure of the file, then the DDR XML file should be re-exported after making these changes, but before actually deleting the relationships.

Since the DDR XML file has already been exported, none of the relationships are actually needed for the actual data transfer part of the migration process. The FileMaker ODBC driver needs to see TOs on the RelationshipGraph which exactly match the name of each base table listed on the Tables tab.

If there are a lot of TOs and relationships on the RelationshipGraph it may be easier to delete all of the objects on the RelationshipGraph and then simply recreate the TOs from the existing base tables within the file. This can be done by clicking the Add Table button and selecting the base table name for each base table.

6) Define a Primary Key for each table.

Each table must have a primary key if you want to create relationships for the table in the SQL database or if you want to migrate repeating fields data.

FmPro Migrator identifies the Primary Key for each table by looking for the Unique and Not Empty field validation options being set for any field within the table. If there are primary key fields which don't have these attributes set, they should be set now.

**Note:** If you are using FileMaker Pro 17+ or FileMaker Pro Advanced, you will be able to copy and paste the table definitions from FileMaker Pro into FmPro Migrator via the Clipboard. FmPro Migrator will then be able to gather all of the attributes for each field. These attributes are also obtained from the file when using AppleScript on macOS.

If the migrated tables don't actually have any field which can serve as a primary key, then a new field should be added as an Auto-Enter Serial Number field within the table. Then each record can be renumbered in Browse mode starting with 1 by clicking in the new field and selecting the Records -> Replace field Contents... menu. Before renumbering the records, select the Records -> Show All Records menu.

Once again, if it is necessary to re-define primary keys and the associated relationships within the database, the DDR XML file should be exported again after making these changes, but before deleting the relationships from within the file.

## 7) Verify Primary Key and Foreign Key Data Types

It is possible (but not recommended) to create relationships within FileMaker Pro databases between Text and Numeric field types. SQL databases will usually not accept differing data types when creating relationships. The field types can be corrected within the Fields tab of the Define/Manage Database dialog.

## 8) Verify Admin Account Access to File

Select the menu:

File -> Manage Accounts & Privileges

In order to transfer data from FileMaker Pro using the ODBC driver, it is necessary for FmPro Migrator to log into the FileMaker Pro database using a password having [Full Access] including ODBC access privileges within the file. The built-in Admin account already has [Full Access] privileges within the file, but these privileges could have been changed after the file was originally created. FileMaker 2,3,4,5,6 files may not even have an Admin account with [Full Access] privileges by default, depending upon how the user accounts were created within the original file.

## 9) Delete all fields having the following types:

Global

Summary

Unstored Calculation

Select the menu:

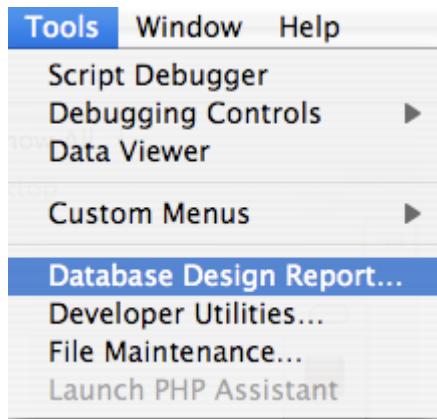
File -> Define/Manage Database...

It is necessary to delete these fields from large database files having more than a few thousand records because the amount of time required for FileMaker Pro to perform the calculations may exceed the ODBC driver timeout interval. If this problem occurs, no records will be transferred from FileMaker Pro to the destination database. The deletion of Global fields is not mandatory, but since Global fields don't exist within SQL databases they should be removed in order to reduce the field count.

**Note:** You may leave the Stored Calculation fields, as the data may be valuable within the destination database.

It will take some work to delete recursive field dependencies. You could make one pass thru the fields from top to bottom to delete all of the fields which can be deleted, and then make another pass thru the fields list from the bottom up to the top of the list.

## Exporting DDR XML File

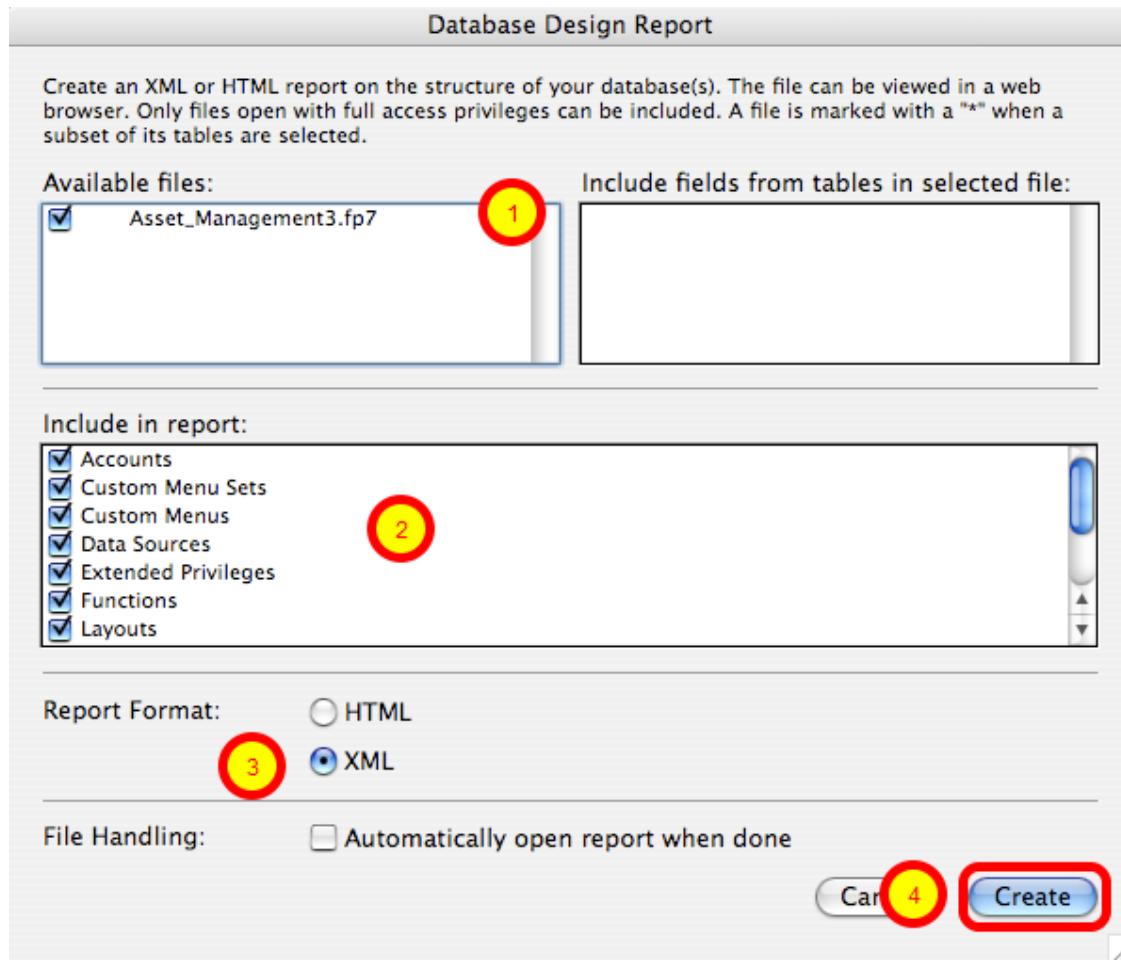


Using FileMaker Pro, select Database Design Report... from the Tools menu.

**Note:** Export the DDR file for the database to be migrated before removing Relationships and Table Occurrences from the Relationship Graph.

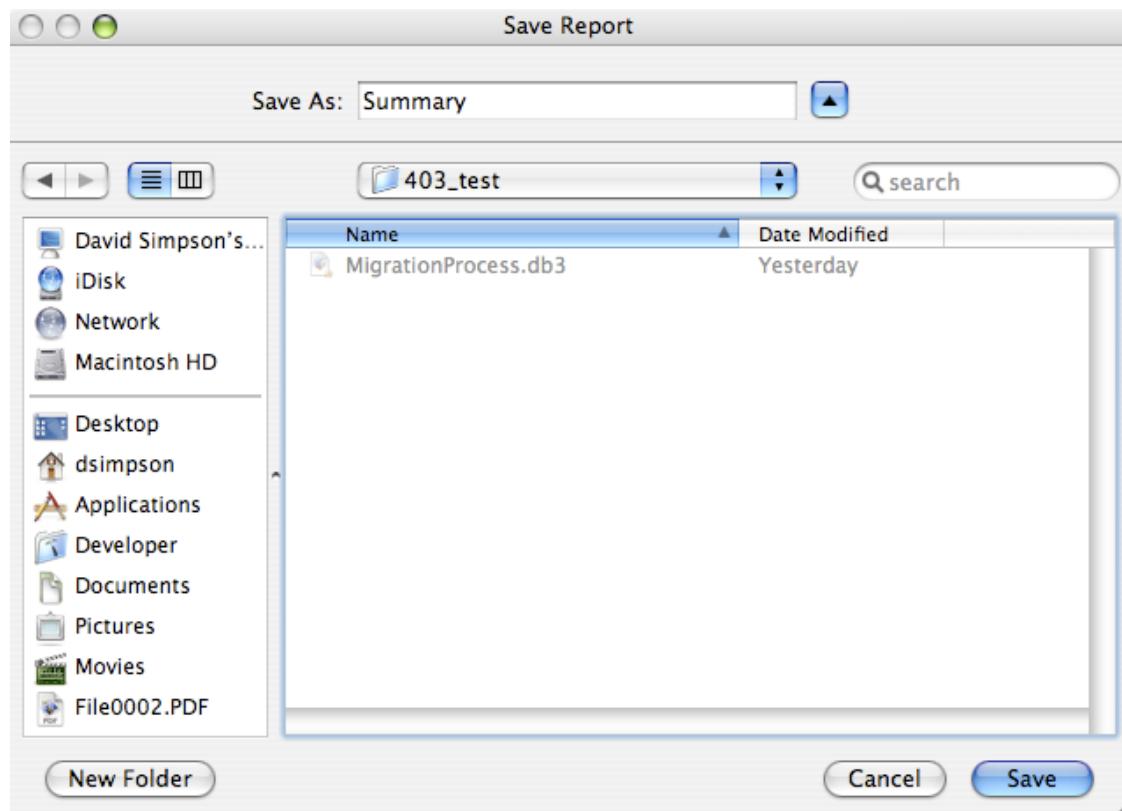
This DDR file will be used for importing Relationships, TOs, Value Lists, Custom Functions and Layouts into FmPro Migrator.

## Exporting DDR XML File - Selecting Output Options



Within the DDR Export dialog, make sure that the database file is (1) checked for export, (2) along with all objects, (3) XML instead of HTML report format, then click the (4) Create button.

## Exporting DDR XML File - Selecting Output Directory



Select the output directory, then click the Save button.

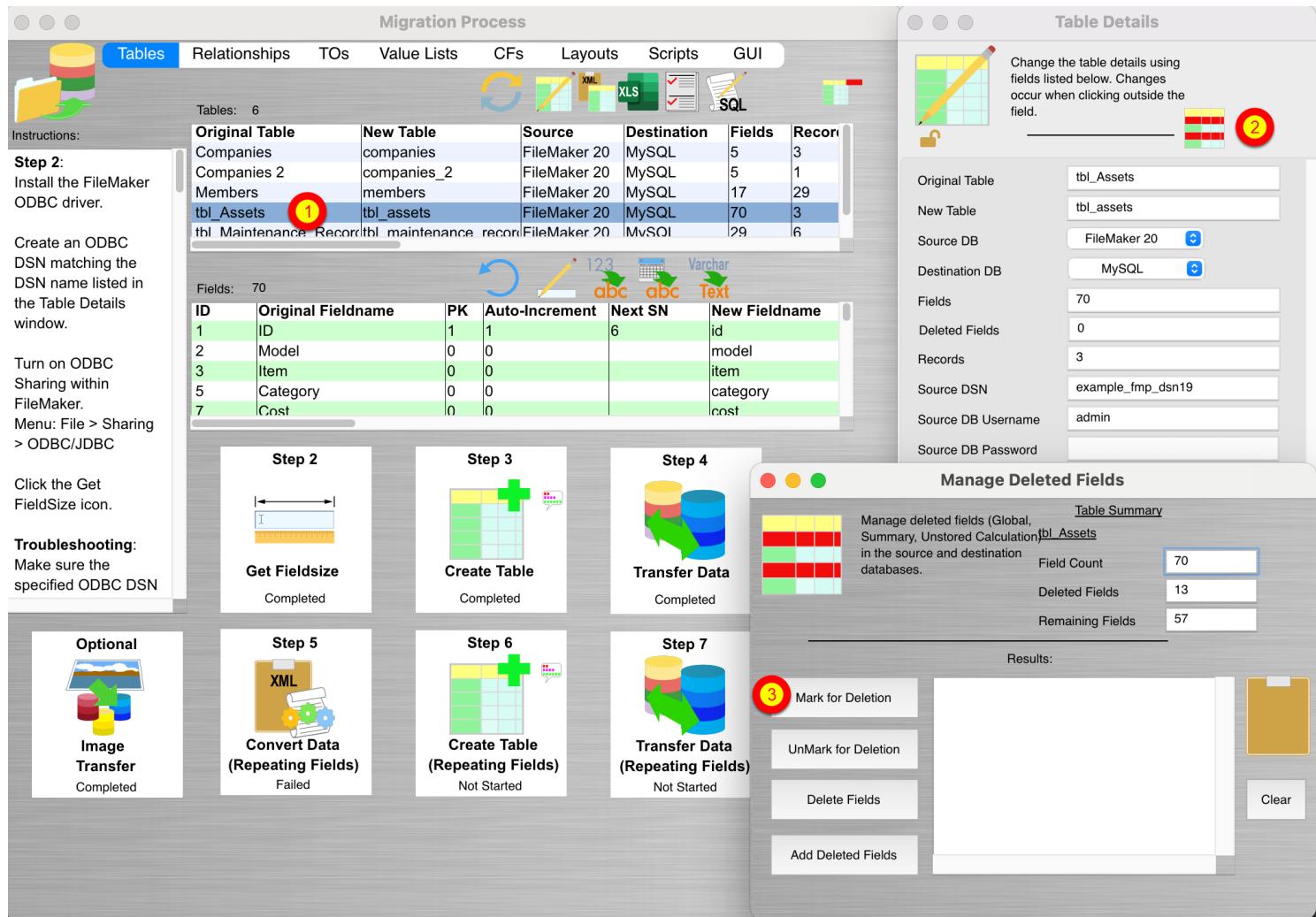
## Pre-Migration Task - Delete Unstored Calc Fields

---

Prior to transferring data from a FileMaker Pro database into a SQL database is is usually necessary to delete all Unstored Calculation and Summary fields from the FileMaker Pro database tables. This step is usually necessary when transferring more than a few thousand records because the amount of time required for FileMaker Pro to calculate these values may cause an ODBC timeout, thus preventing the data from being transferred at all. This step should be completed before Step 2 - Get Fieldsize is run.

FmPro Migrator includes an automated field deletion feature which can help with this task.

## Open Manage Deleted Fields Window

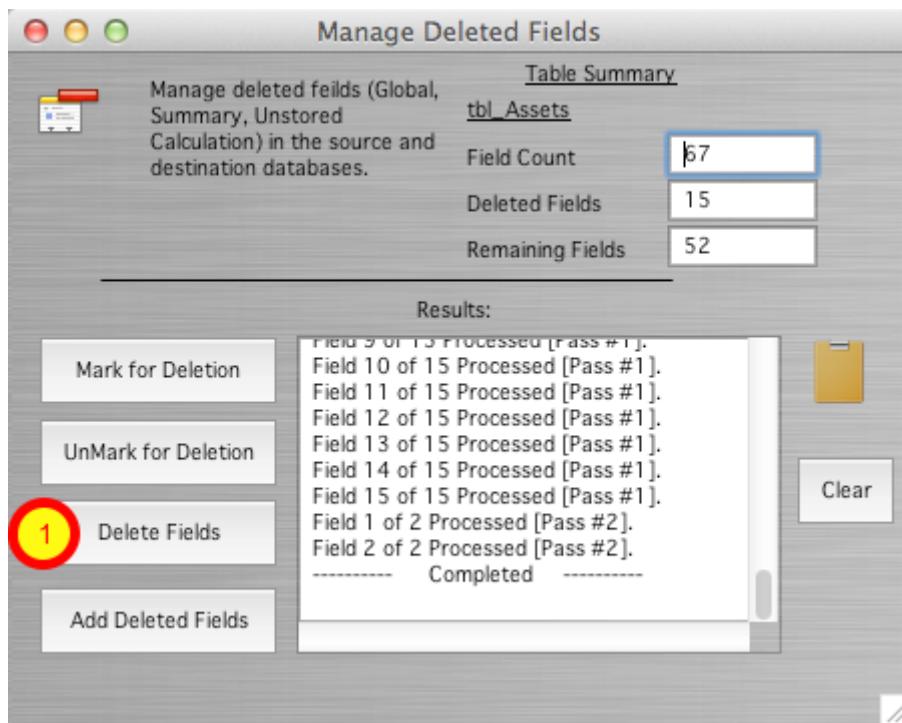


Once the FileMaker table definitions have been imported via the clipboard, the list of tables will be displayed within the Migration Process window.

- 1) Double-click on the table name, the Table Details window will be displayed. Click on the Manage Deleted Fields button near the top of the Table Details window, the Manage Deleted Fields window will open.
- 3) Click on the Mark for Deletion Button, all of the Unstored Calculation, Summary and Global fields in the selected table will be marked for deletion.

**Note:** Global fields are also marked for deletion in order to reduce the number of fields of data to speed up the data transfer process. For the example table shown in this screenshot, there were 15 fields marked for deletion. At this point, the fields have not yet been deleted, they have just been marked for deletion.

## Delete Fields



- 1) Click the Delete Fields button. FmPro Migrator will first print a list of the ALTER TABLE SQL commands in the Results window, then the ALTER TABLE commands will be sent directly to the FileMaker database thru the ODBC connection. An example of these commands is listed below.

**Note:** It is common for some DROP COLUMN commands to fail as a result of Outstanding References by other fields. In order to resolve this issue, FmPro Migrator logs the failed DROP COLUMN commands and then re-sends those commands to the FileMaker Pro database again, in reverse order during Pass #2 processing. If PASS #2 still results in DROP COLUMN command failures, the remaining list of undeleted fields will be listed at the end of the processing.

Click on the Clipboard icon to copy the commands any other software for review purposes. It will be helpful to have these field names available when deleting the fields manually. It is necessary to manually delete any fields which were not deleted automatically in order to proceed with Step #2 - Get Fieldsizes.

```
ALTER TABLE "tbl_Assets" DROP COLUMN "Total Cost"  
ALTER TABLE "tbl_Assets" DROP COLUMN "Total Book Value"  
ALTER TABLE "tbl_Assets" DROP COLUMN "Total Depreciation"  
ALTER TABLE "tbl_Assets" DROP COLUMN "Template Information Global"  
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteLibrary"  
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteSortedBy"  
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteCategory"  
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteAssignedTo"
```

```
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteItem"
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteModel"
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteSerialNumber"
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteLocation"
ALTER TABLE "tbl_Assets" DROP COLUMN "SC_Full_URL_ReadWrite"
ALTER TABLE "tbl_Assets" DROP COLUMN "SC_Full_URL_ReadOnly"
ALTER TABLE "tbl_Assets" DROP COLUMN "SC_URL_Prefix"
```

Field 1 of 15 Processed [Pass #1].

Field 2 of 15 Processed [Pass #1].

Field 3 of 15 Processed [Pass #1].

Field 4 of 15 Processed [Pass #1].

Field 5 of 15 Processed [Pass #1].

```
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteLibrary"
```

[FileMaker][FileMaker] (13): Outstanding references

Field 6 of 15 Processed [Pass #1].

```
ALTER TABLE "tbl_Assets" DROP COLUMN "HiliteSortedBy"
```

[FileMaker][FileMaker] (13): Outstanding references

Field 7 of 15 Processed [Pass #1].

Field 8 of 15 Processed [Pass #1].

Field 9 of 15 Processed [Pass #1].

Field 10 of 15 Processed [Pass #1].

Field 11 of 15 Processed [Pass #1].

Field 12 of 15 Processed [Pass #1].

Field 13 of 15 Processed [Pass #1].

Field 14 of 15 Processed [Pass #1].

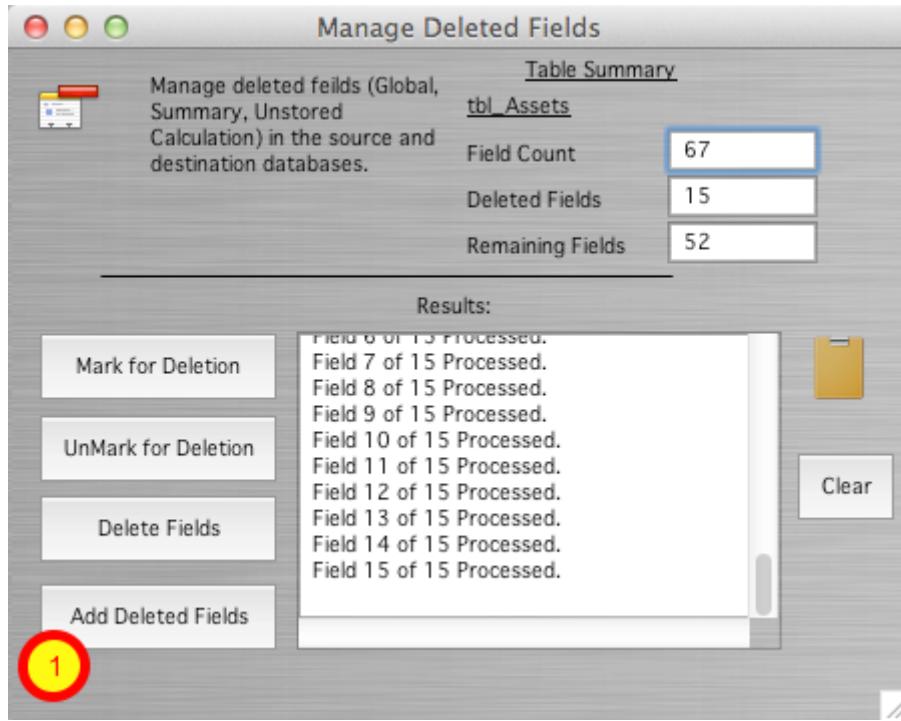
Field 15 of 15 Processed [Pass #1].

Field 1 of 2 Processed [Pass #2].

Field 2 of 2 Processed [Pass #2].

----- Completed -----

## Adding Deleted Fields to the SQL Database [PHP & LiveCode Conversions]



After the table has been created in the SQL database, and the data transferred to the SQL database, the deleted columns can then be added to the SQL database table.

**Note:** This step is only needed if a PHP Conversion project is being generated, in order to insure that the database columns referenced in the CakePHP Model files exist within the database table.

```
ALTER TABLE tbl_assets ADD total_cost DOUBLE NULL;
ALTER TABLE tbl_assets ADD total_book_value DOUBLE NULL;
ALTER TABLE tbl_assets ADD total_depreciation DOUBLE NULL;
ALTER TABLE tbl_assets ADD template_information_global TEXT NULL;
ALTER TABLE tbl_assets ADD hilitelibrary BLOB NULL;
ALTER TABLE tbl_assets ADD hilitesortedby TEXT NULL;
ALTER TABLE tbl_assets ADD hilitecategory BLOB NULL;
ALTER TABLE tbl_assets ADD hiliteassignedto BLOB NULL;
ALTER TABLE tbl_assets ADD hiliteitem BLOB NULL;
ALTER TABLE tbl_assets ADD hilitemodel BLOB NULL;
ALTER TABLE tbl_assets ADD hiliteserialnumber BLOB NULL;
ALTER TABLE tbl_assets ADD hilitelocation BLOB NULL;
ALTER TABLE tbl_assets ADD sc_full_url_readwrite TEXT NULL;
ALTER TABLE tbl_assets ADD sc_full_url_READONLY TEXT NULL;
ALTER TABLE tbl_assets ADD sc_url_prefix TEXT NULL;
Field 1 of 15 Processed.
Field 2 of 15 Processed.
Field 3 of 15 Processed.
```

Field 4 of 15 Processed.

Field 5 of 15 Processed.

Field 6 of 15 Processed.

Field 7 of 15 Processed.

Field 8 of 15 Processed.

Field 9 of 15 Processed.

Field 10 of 15 Processed.

Field 11 of 15 Processed.

Field 12 of 15 Processed.

Field 13 of 15 Processed.

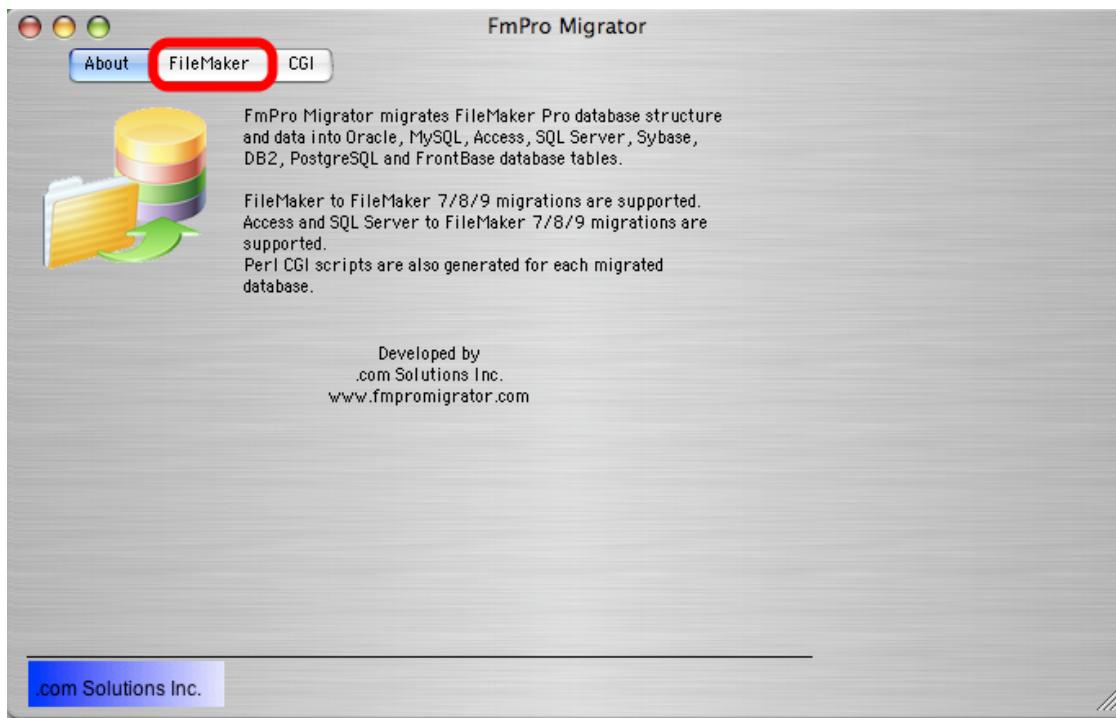
Field 14 of 15 Processed.

Field 15 of 15 Processed.

## Step 1 - Get Info

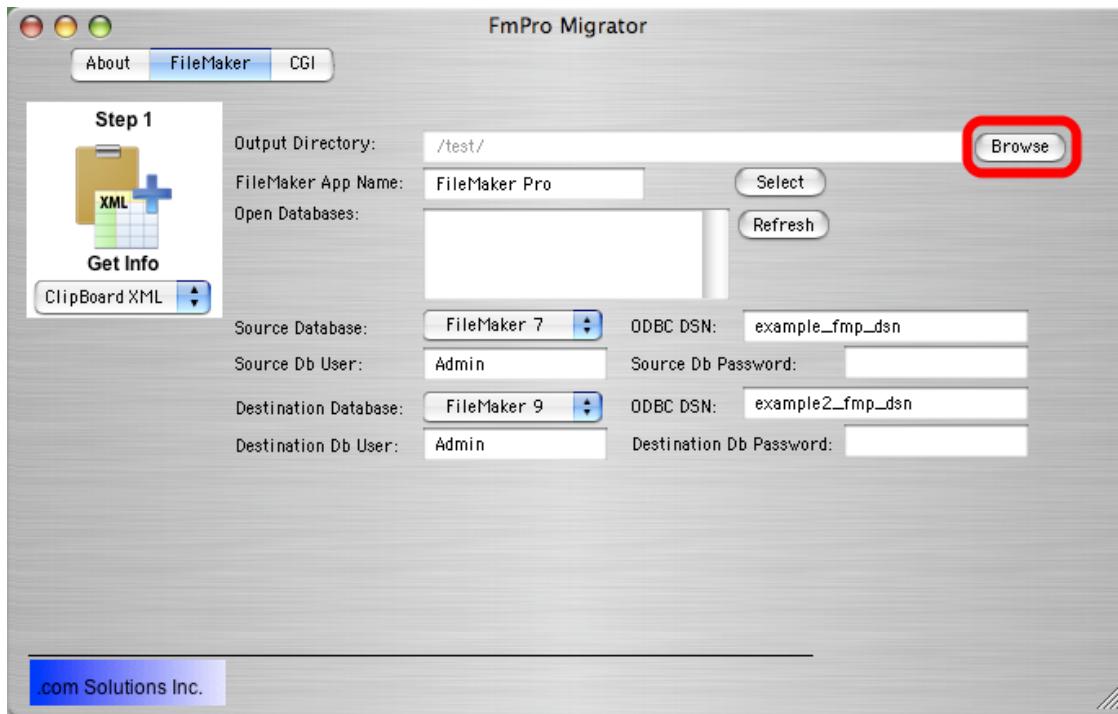
At this point in the process, the FileMaker ODBC Driver should already be installed and a system DSN should have already been created. Please see the info on the [FmPro Migrator support page](#) for ODBC Driver installation and setup details.

### Step 1 - Get Info - Select FileMaker tab



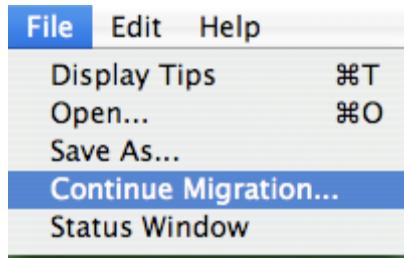
Launch FmPro Migrator, then click on the FileMaker tab at the top of the window.

## Step 1 - Get Info - Select Output Directory



FmPro Migrator stores migration process information within a SQLite database file named `MigrationProcess.db3`. This file contains the metadata for the source database file(s), including tables, fields, table creation SQL code and status info. Therefore the first step in the migration process is to select the **Browse** button to select the output folder FmPro Migrator will use when creating the `MigrationProcess.db3` file.

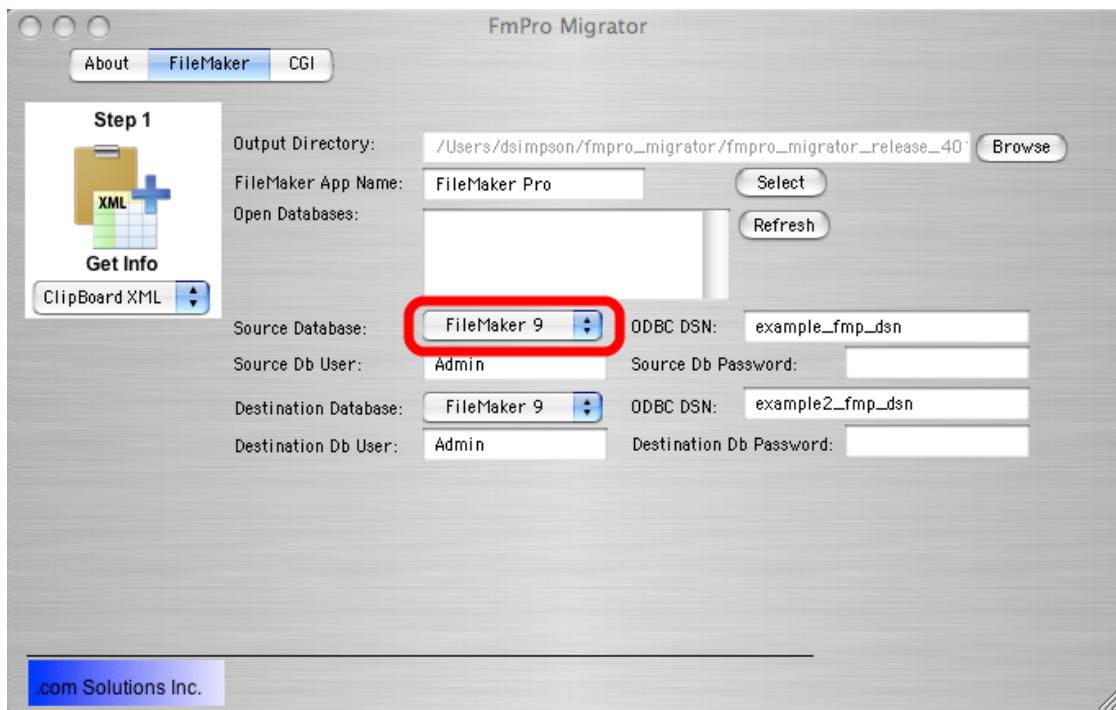
## Continue Migration...



Tip: To restart an existing migration project, select the output directory, then select **Continue Migration...** from the **File** menu.

Click the yellow Continue button to open the Migration Process window.

## Step 1 - Get Info - Enter Source Database Info



Select the type of source database from the Source Database menu.

Note: Since there isn't a macOS compatible FileMaker ODBC driver for FileMaker 5/6, these older database files should be converted to FileMaker 7 or higher in order to perform the migration on macOS.

Either use the existing ODBC DSN name "example\_fmp\_dsn" or change this name to reflect the name of a System ODBC DSN you have created on your computer.

Enter the Username and Password required to access the source database file. If a password has not been configured within a FileMaker 7+ file, set the username to "Admin" with the password field empty. The user account entered here needs to have full access to the database file and needs to have ODBC access privileges within the file. This is the default access for the Admin account.

## Step 1 - Install FileMaker ODBC Driver

FmPro Migrator uses the FileMaker ODBC driver provided by FileMaker Inc. to transfer data from FileMaker databases.

### For FileMaker 5/6 Databases on Windows:

The FileMaker ODBC driver is installed automatically when the FileMaker application is installed. This driver is installed with the Demo and Full versions of the FileMaker application. If you are converting FileMaker 2, 3 or 4 database files, upgrade these files into FileMaker 5/6 files in order

to perform the migration.

**For FileMaker 7+ Databases on Windows:**

Install the DataDirect SequeLink driver using the installer located in the xDBC folder on the FileMaker application CD. Do not download and install the DataDirect SequeLink driver from the Datadirect website. The FileMaker ODBC driver is only supplied on the FileMaker installation CD or the FileMaker ESD (downloadable CD image) with the fully licensed versions of FileMaker Pro, FileMaker Developer 7, FileMaker Pro and FileMaker Server.

For FileMaker 11 databases on Windows use the ODBC Client Driver Installer within the xDBC folder.

**For FileMaker 2, 3, 4, 5, 6 Databases on macOS:**

Upgrade these database files to FileMaker 7+, use the FileMaker 7+ ODBC driver on macOS.

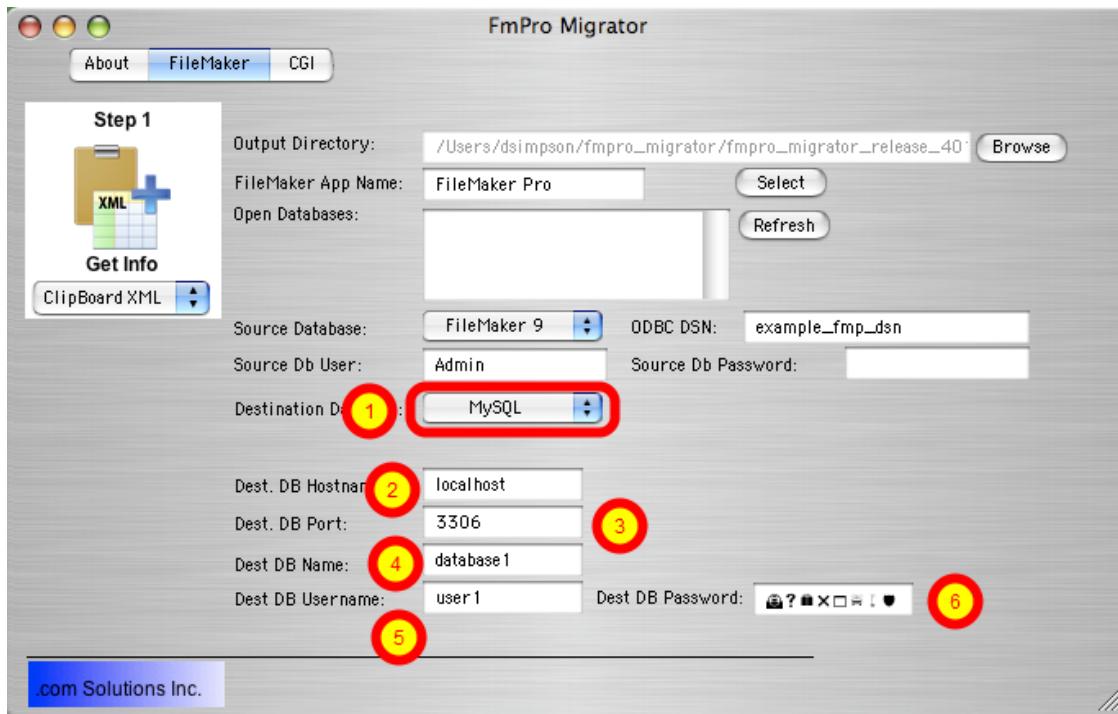
**For FileMaker 7+ Databases on macOS:**

Manually copy the SequeLink.bundle file from the ODBC Client Driver folder on the FileMaker application CD or ESD dmg to the /Library/ODBC folder within macOS.

For FileMaker 11 databases on macOS double-click the file named FileMaker ODBC.mpkg within the xDBC/ODBC Client Driver Installer folder. Install the ODBC Manager app from the [www.odbcmanger.net](http://www.odbcmanger.net) website.

**Note:** Using the FileMaker ODBC driver with FileMaker Pro 7+ requires the FileMaker database file and ODBC driver to be open locally on the same computer where FmPro Migrator is running. The exception to this guideline is when the database file is opened with FileMaker Server.

## Step 1 - Get Info - Enter Destination Database Info



(1) Select MySQL as the destination database type. Once the destination database has been selected, a new set of fields will become visible for entering the connection parameters for the destination database.

The destination database may located either locally or remotely. (2) Enter the hostname or IP address for the database server, (3) TCP/IP Port number, (4) database name, (5) username, (6) password.

Note1: If your MySQL database is located at an ISP, verify that you have remote access to log in directly to the MySQL database. Some ISPs don't allow remote database logins for security reasons. You might need to install a copy of MySQL on your local machine in order to complete the migration, then export, upload and import the data to your ISP's server if direct remote access is not possible.

Note2: Make sure that the MySQL database account entered in field (5) has full access to the MySQL database. This account should have all privileges granted in order for FmPro Migrator to be able to create/drop tables and insert data. You may also need to update the MySQL access privileges to allow remote access for this user account because the default privileges for new MySQL accounts provides no remote login access.

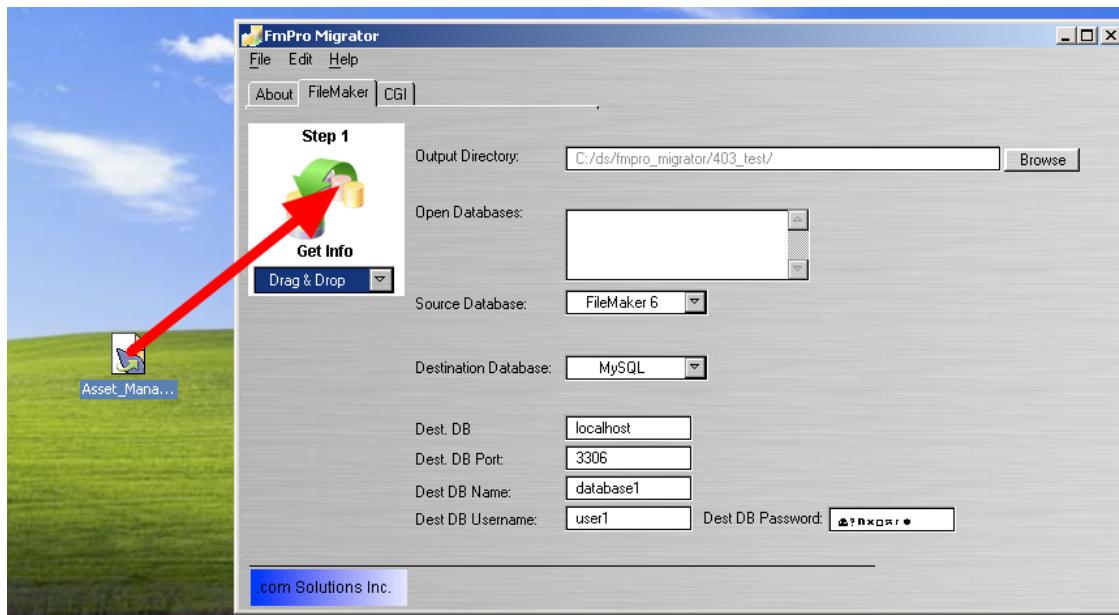
## Step 1 - Get Info

FmPro Migrator needs to get info about tables, fields, field types and repeating fields status within the source FileMaker database file(s). This info is used to create the tables/fields within the destination database and is used to copy data between the databases.

There are multiple ways which FmPro Migrator can use to get info about the source database file, as listed below. Use the method which works best for your computer OS and the version of the FileMaker database file being migrated.

Select the pop-up menu below the Step 1 Get Info icon to select the method you want to use to get info from the source database file(s). This pop-up menu is context-sensitive, based upon the selected source database type and the platform where FmPro Migrator is running (macOS or Windows).

### Step 1 - Get Info - Using Drag & Drop - macOS & Windows



FmPro Migrator reads the metadata directly from older FileMaker database files, including FileMaker 2, 3, 4, 5 and 6 on macOS and Windows. This method of getting info from FileMaker provides a very accurate representation of the source database file, including calculation formula definitions and repeating field counts for each field.

1) Rename the original source file(s) to remove any spaces or special characters. This filename will be used as a table name within SQL statements used by FmPro Migrator to retrieve data, and the FileMaker 5/6 ODBC driver does not allow spaces within tablenames. Spaces can be replaced with underscore characters to make the name easier to read. It is not necessary to replace characters within most fieldnames, as FmPro Migrator handles this task automatically

when creating the SQL code which creates the destination table.

2) Delete all scripts, relationships and file references from the source file(s). Deleting the scripts, relationships and file references reduces the number of dependencies within the file and makes it easier to delete any fields which need removed from the table.

Select File > Define Fields... menu to open the Define Fields window within FileMaker. Delete all fields which are defined as Unstored Calculation, Summary or Global type fields. If these types of fields remain within the database file during the migration, it is likely that the ODBC driver will time out before retrieving the data from the source database.

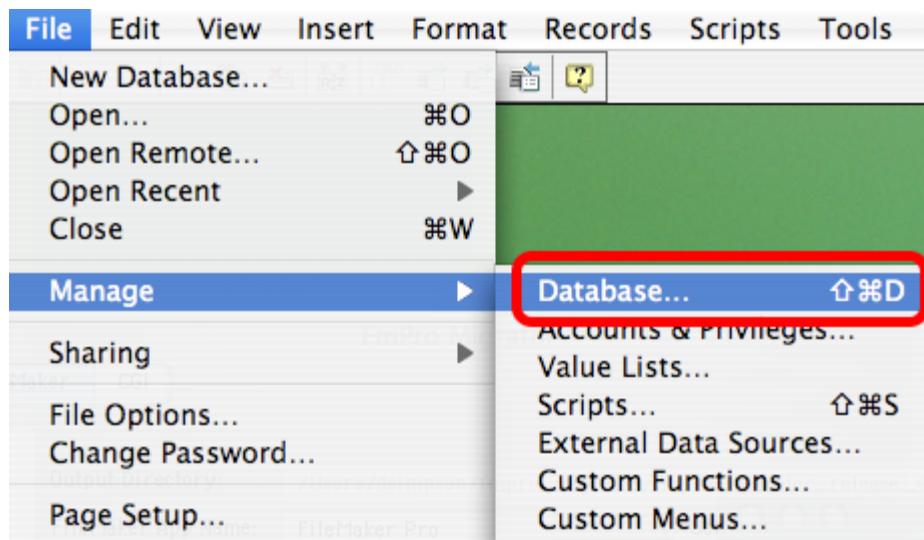
2) Create an empty copy of the source database file by selecting Save a Copy As... from the File menu, and selecting "clone (no records)" from the save dialog. Change the name of the saved file to exactly match the name of the original file.

3) Drag & Drop the empty copy of the source database file onto the Step 1 Get Info icon.

4) Repeat these steps for each source database file which needs migrated.

Note: Since there is no FileMaker ODBC driver available for older versions of FileMaker on macOS, if you use this method to get info from your source database file(s), you will then need to upgrade the file(s) to FileMaker 7+ prior to performing the migration. You will then need to change the source database type from FileMaker 5/6 to FileMaker 7+ within FmPro Migrator.

#### Step 1 - Get Info - Using ClipBoard XML - macOS & Windows

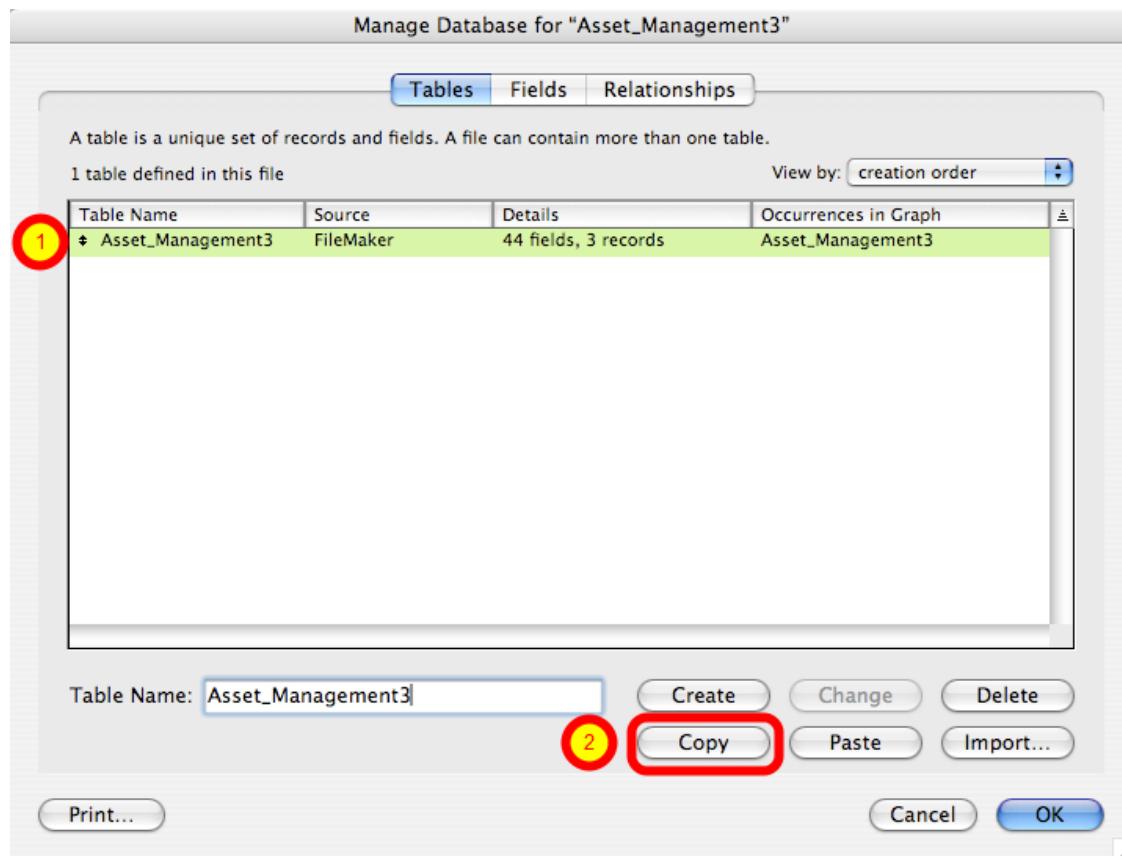


The most desirable method for obtaining info about FileMaker 7+ database files on macOS and Windows is thru copying the table structure using FileMaker Pro. This method copies the table creation XML code from the clipboard and includes all of the info required to create the tables,

fields, repeating field count and calculation formulas.

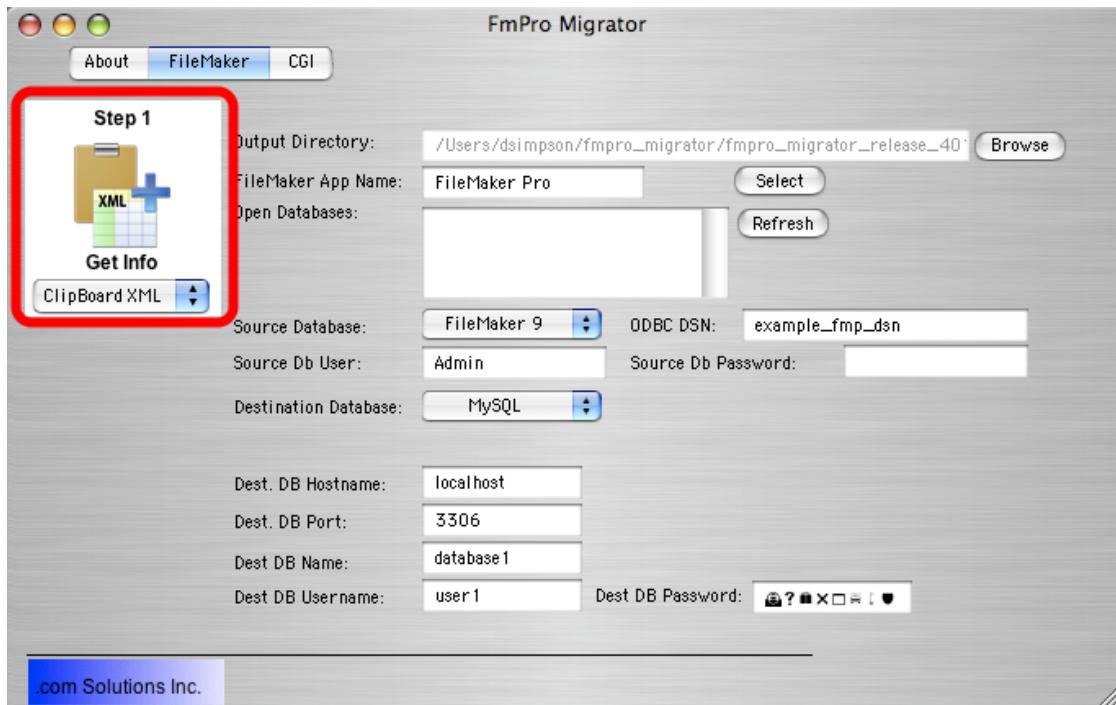
Open the FileMaker database file.

Select Define/Manage Database from the File menu within FileMaker.



(1) Select all of the tables on the Tables tab, (2) click the Copy button.

Once the Table XML info has been put onto the ClipBoard, the Paste button will become active. Click the Cancel or Ok buttons to close the Define dialog.



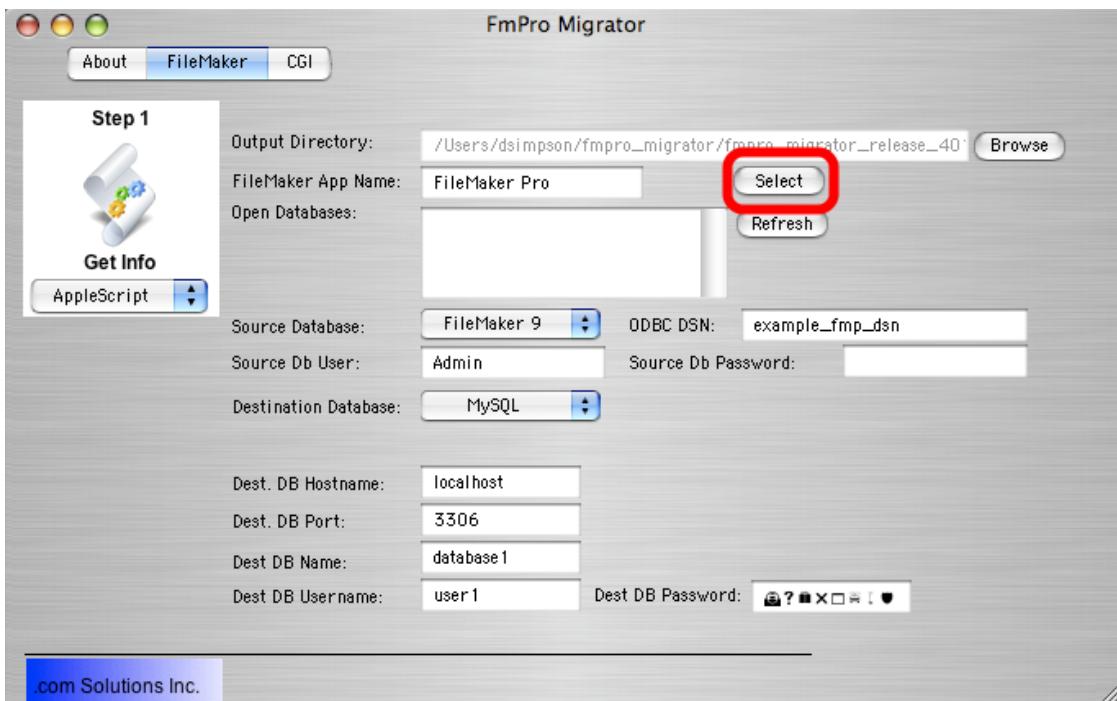
Within FmPro Migrator, click the Step 1 Get Info button (with the Clipboard XML menu option selected). The yellow Continue... button will become visible once FmPro Migrator has completed processing the Table XML from the Clipboard.

Note1: In order for the ODBC driver to find each table within the source database file, there must be a TO on the relationship graph which exactly matches the name of the base table within the FileMaker database file. It is generally easier to delete all of the TOs and relationships from the relationship graph. Then create new TOs consisting of only the base table name. Otherwise, the ODBC driver will not be able to find and transfer data from each table within the FileMaker database.

Note2: If you want to transfer relationships to the destination database, use FileMaker Pro to export a DDR from the original FileMaker 7+ file before removing the TOs. Then import the relationships from the DDR later in the migration process, so that SQL code can be created to represent these relationships within the destination database.

Note3: It is not necessary to replace spaces within FileMaker 7+ tablenames, as was necessary for FileMaker 5/6 files. It is necessary to replace high ASCII characters (greater than ASCII character code 127) and Unicode characters within table and field names, as these characters are not supported by the FileMaker ODBC driver.

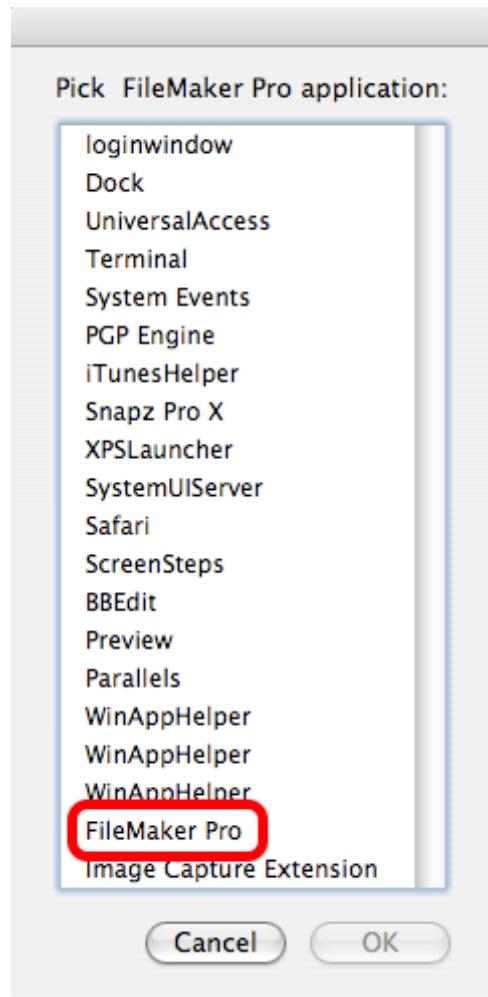
## Step 1 - Get Info - Using AppleScript - macOS



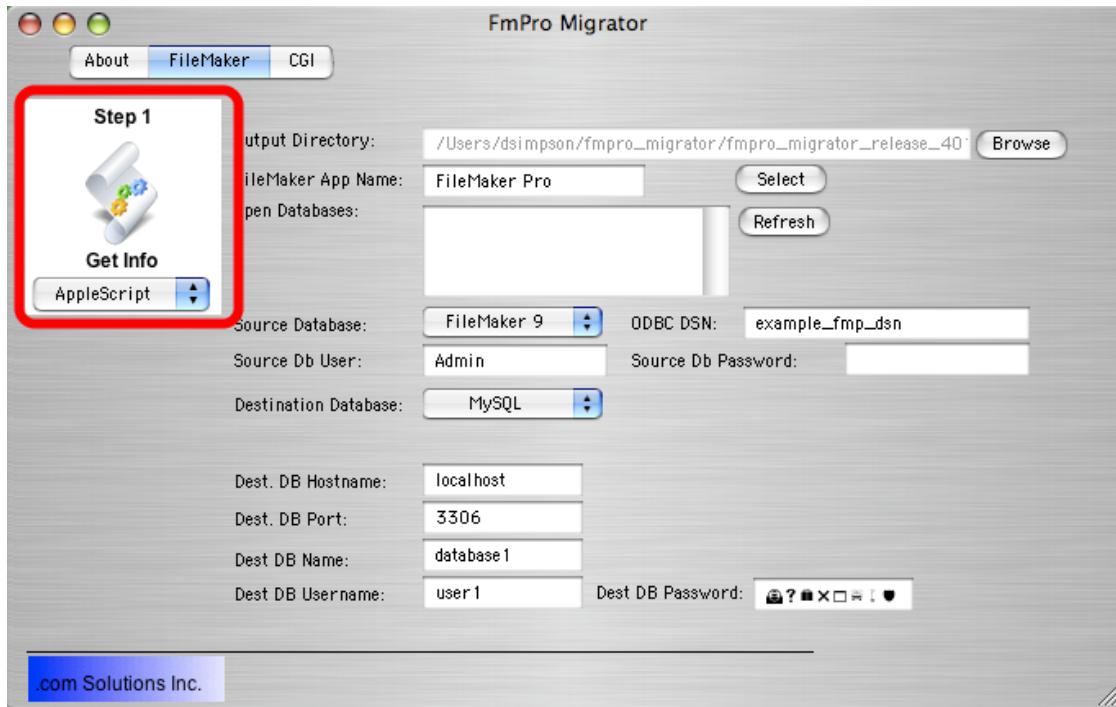
On macOS, AppleScript can be used to gather Table/Field info from FileMaker 7+ databases. AppleScript is slower than copying the Table XML via the ClipBoard, and may take several minutes to gather info for large databases containing many tables and fields. Using AppleScript does provide valuable info such as calculation formulas and repeating fields count info.

Open the FileMaker database file.

If necessary, click the Select button to select FileMaker Pro from the list of running applications.



It is not usually necessary to perform this step, and is generally only required if an error occurs. The FileMaker application name has changed thru the years (sometimes FileMaker Developer, FileMaker Pro Advanced, FileMaker Pro), but now seems stable as the name FileMaker Pro - unless you have renamed the application on your computer. Now, even FileMaker Pro Advanced shows up as FileMaker Pro according to AppleScript.



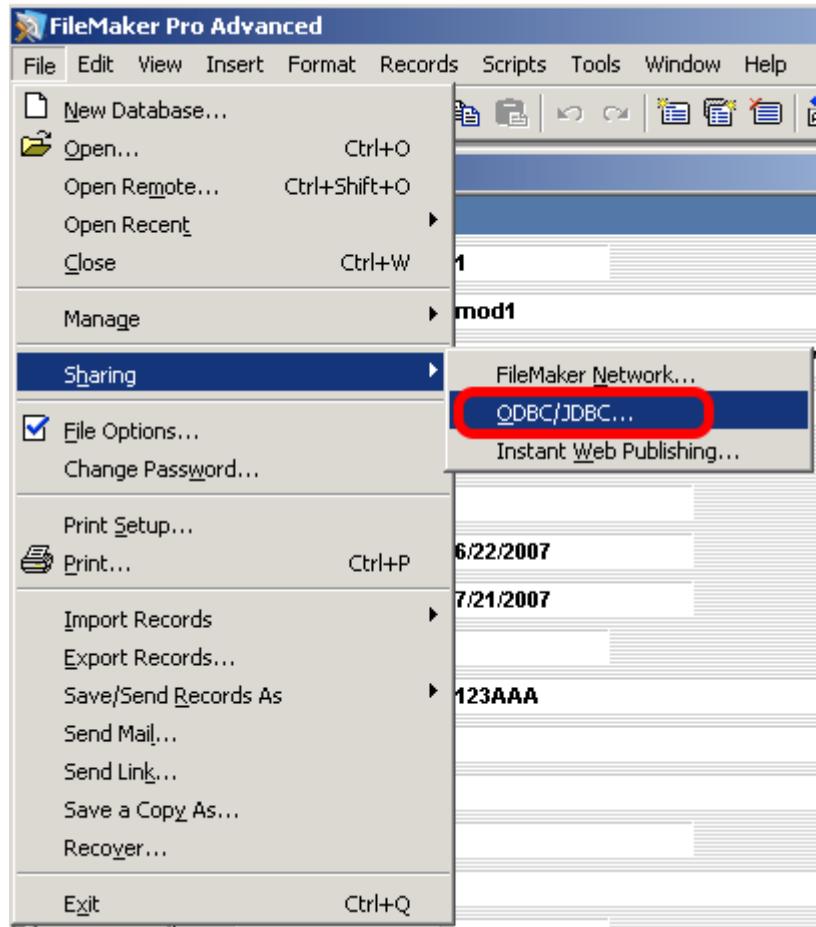
Click the Step 1 Get Info button, with AppleScript selected from the pop-up menu. FmPro Migrator will send AppleScript commands to the open FileMaker database file(s) and gather info about each file. The yellow Continue... button will be made visible once this process has completed.

Note1: In order for the ODBC driver to find each table within the source database file, there must be a TO on the relationship graph which exactly matches the name of the base table within the FileMaker database file. It is generally easier to delete all of the TOs and relationships from the relationship graph. Then create new TOs consisting of only the base table name. Otherwise, the ODBC driver will not be able to find and transfer data from each table within the FileMaker database.

Note2: If you want to transfer relationships to the destination database, use FileMaker Pro to export a DDR from the original FileMaker 7+ file before removing the TOs. Then import the relationships from the DDR later in the migration process, so that SQL code can be created to represent these relationships within the destination database.

Note3: It is not necessary to replace spaces within FileMaker 7+ tablenames, as was necessary for FileMaker 5/6 files. It is necessary to replace high ASCII characters (greater than ASCII character code 127) and Unicode characters within table and field names, as these characters are not supported by the FileMaker ODBC driver.

## Step 1 - Get Info - Using ODBC - Windows

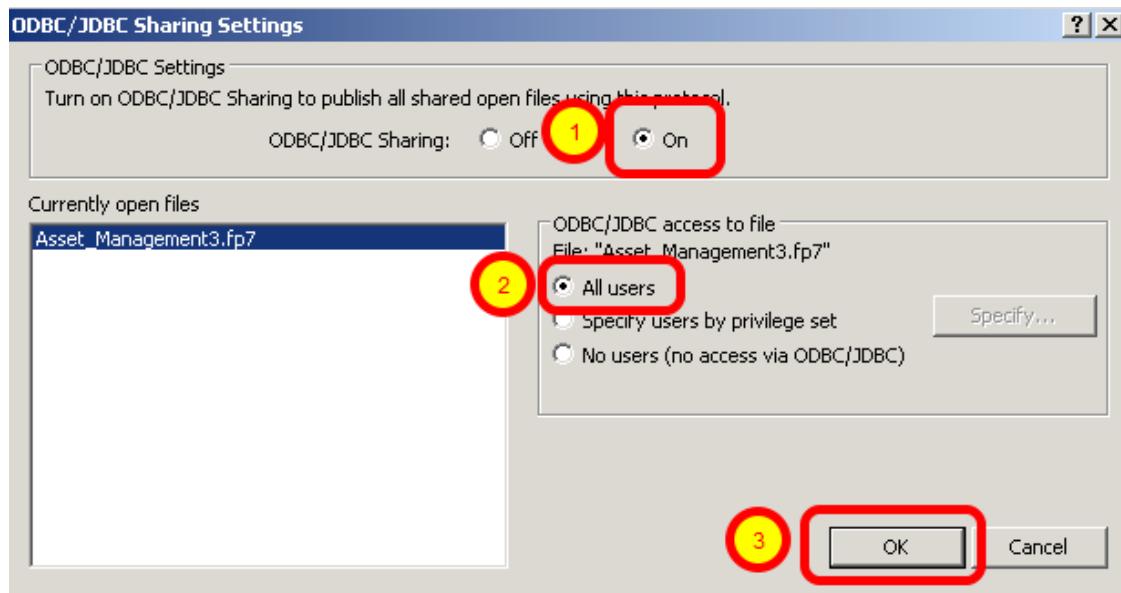


An alternative method for getting info from source FileMaker 7+ databases is by using ODBC on Windows. This method is less desirable compared to using the ClipBoard XML method, because repeating field count values are not obtained thru an ODBC connection. The repeating fields need to be defined manually within the Migration Process Field Details window.

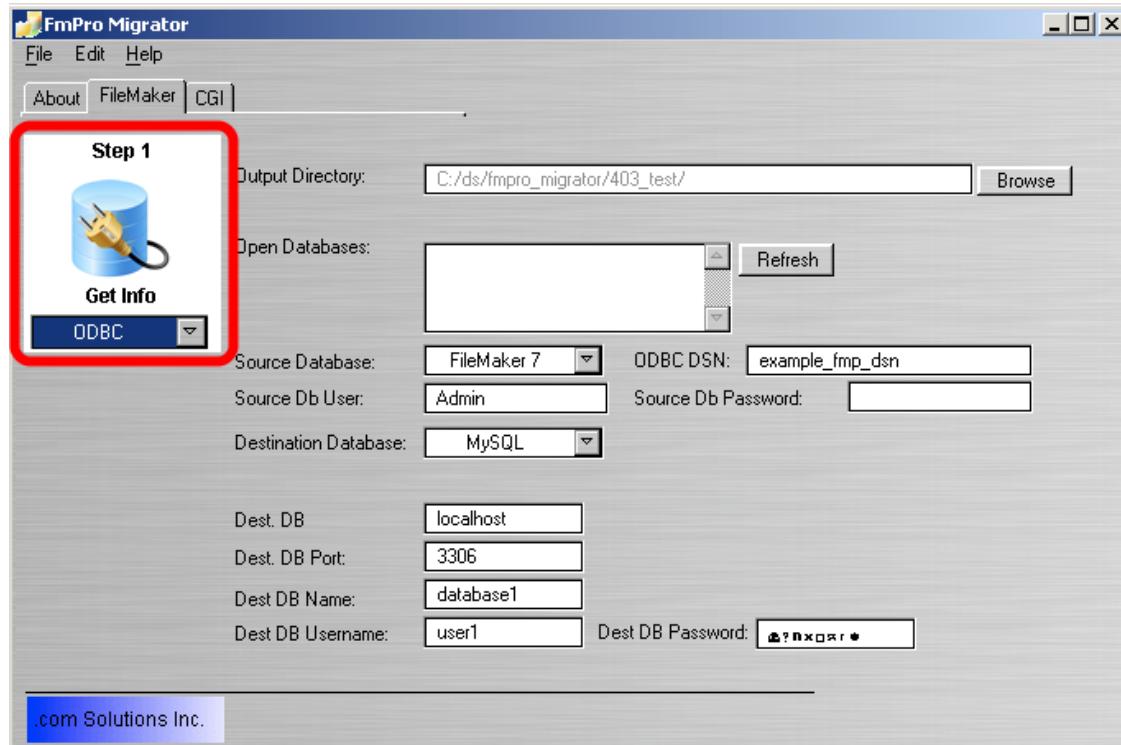
Open the FileMaker database file.

Create an ODBC System DSN matching the ODBC DSN name shown for the Source Database.

Enable ODBC sharing within FileMaker. Select the File > Sharing > ODBC/JDBC... menu.



(1) Turn on sharing for (2) All users, (3) click the Ok button.



Click the Step 1 Get Info button. The table structure info will be read from the source database file, and written into a newly created MigrationProcess.db3 file. The yellow continue button will then become visible.

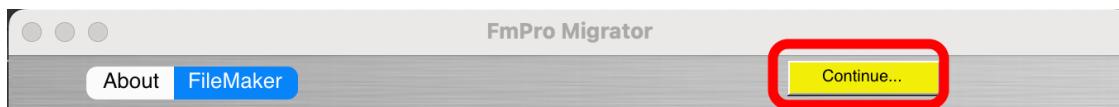
Note1: In order for the ODBC driver to find each table within the source database file, there must be a TO on the relationship graph which exactly matches the name of the base table within the FileMaker database file. It is generally easier to delete all of the TOs and relationships from the

relationship graph prior to gathering info via ODBC. Then create new TOs consisting of only the base table name. Otherwise, the ODBC driver will report each TO (including self-joins) as a separate base table within the FileMaker database. So you will end up trying to migrate the same data multiple times, just based upon the TOs on the relationships graph - which is not required. For migration purposes, you only need to migrate the data from the base tables.

Note2: If you want to transfer relationships to the destination database, use FileMaker Pro to export a DDR from the original FileMaker 7+ file before removing the TOs. Then import the relationships from the DDR later in the migration process, so that SQL code can be created to represent these relationships within the destination database.

Note3: It is not necessary to replace spaces within FileMaker 7+ tablenames, as was necessary for FileMaker 5/6 files. It is necessary to replace high ASCII characters (greater than ASCII character code 127) and Unicode characters within table and field names, as these characters are not supported by the FileMaker ODBC driver.

#### Step 1 - Get Info - Click Continue... Button



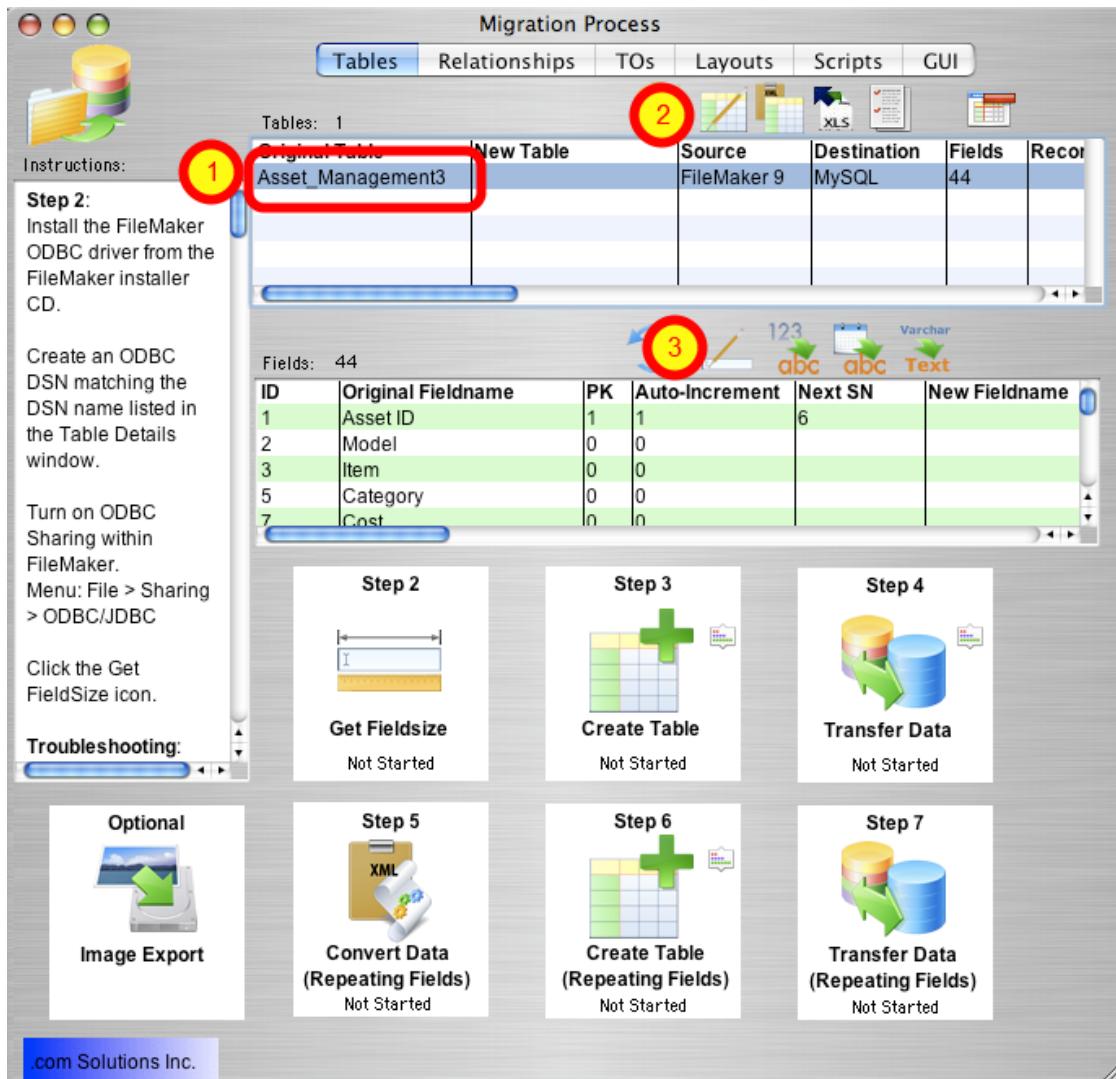
Regardless of the method used to gather info from the source FileMaker database or table, FmPro Migrator will make the yellow Continue... button visible once it has gathered info from the first FileMaker database. Click this button to open the Migration Process window and continue on to Step 2 of the migration process.

## Step 2 - Get Fieldsize

FmPro Migrator needs to determine the maximum amount of data stored within each field of the source database table. To accomplish this task, FmPro Migrator performs a SELECT query to read all of the records from the source FileMaker database table. Before FmPro Migrator can make a successful ODBC connection to the source database, a System ODBC DSN needs to be created.

At this point in the process, the FileMaker ODBC driver should already be installed on either MacOS X or Windows.

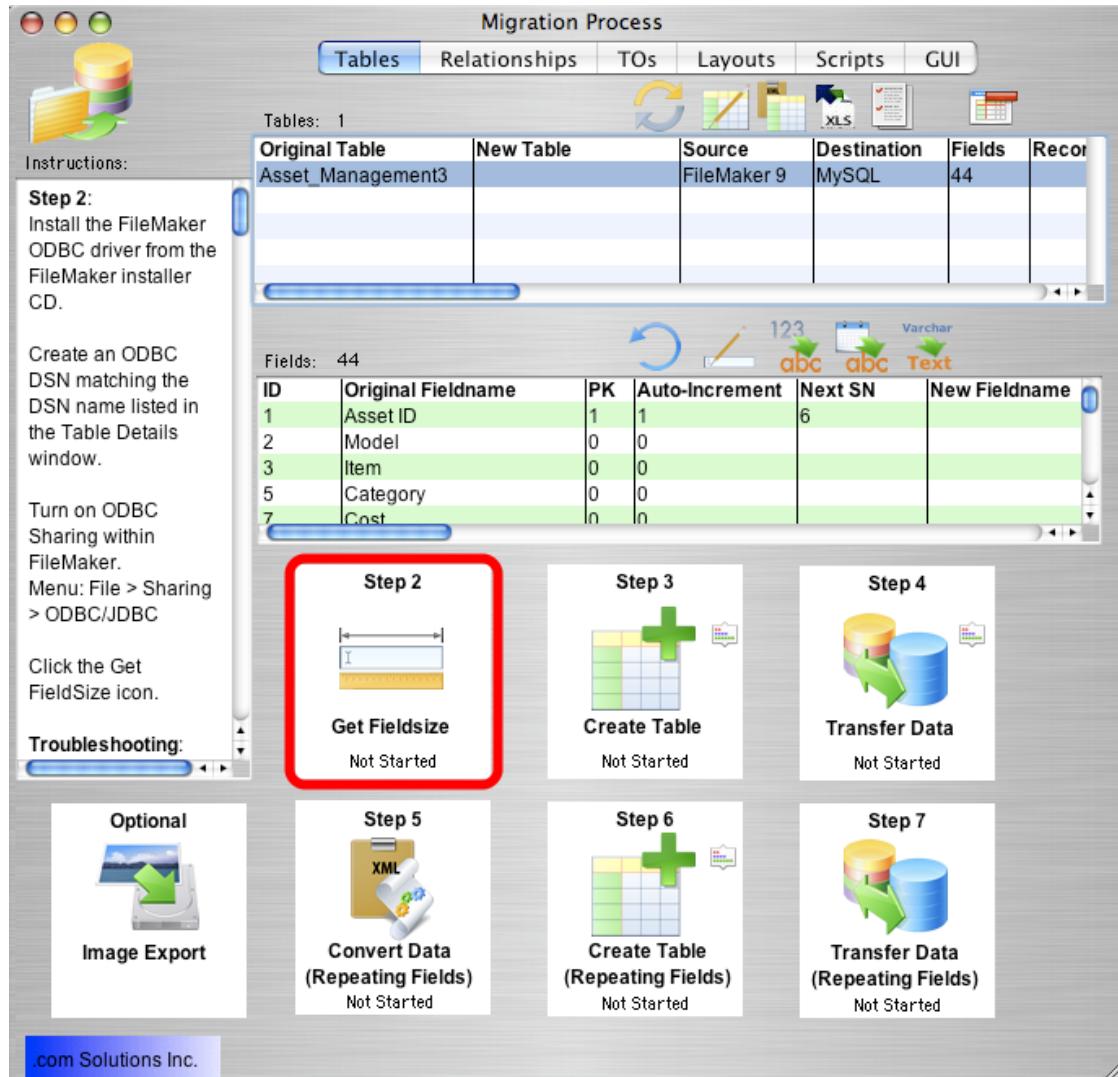
### Step 2 - Get Fieldsize



(1) Click on a table in the Tables list. Once a table is selected, a list of fields contained within the table will be displayed in the Fields list. These fields represent the info gathered from the source database by FmPro Migrator.

(2) Clicking the Table Details button opens the Table Details window, which enables you to view and make changes to the individual parameters for the table.

(3) The Field Details button provides you with the ability to view and make changes at the field level.



Click the Step 2 Get Fieldsize button. FmPro Migrator will make an ODBC connection to the source FileMaker database, and put up a progress dialog as it is reading thru the records. After the records have been successfully read from the table, the status menu at the bottom of the Step 2 button will change from "Not Started" to "Completed". The status will be set to "Failed" if the Get Fieldsize step fails for some reason.

If the Get Fieldsize step fails:

- 1) Check to make sure that the source database is open within FileMaker Pro on the local computer.

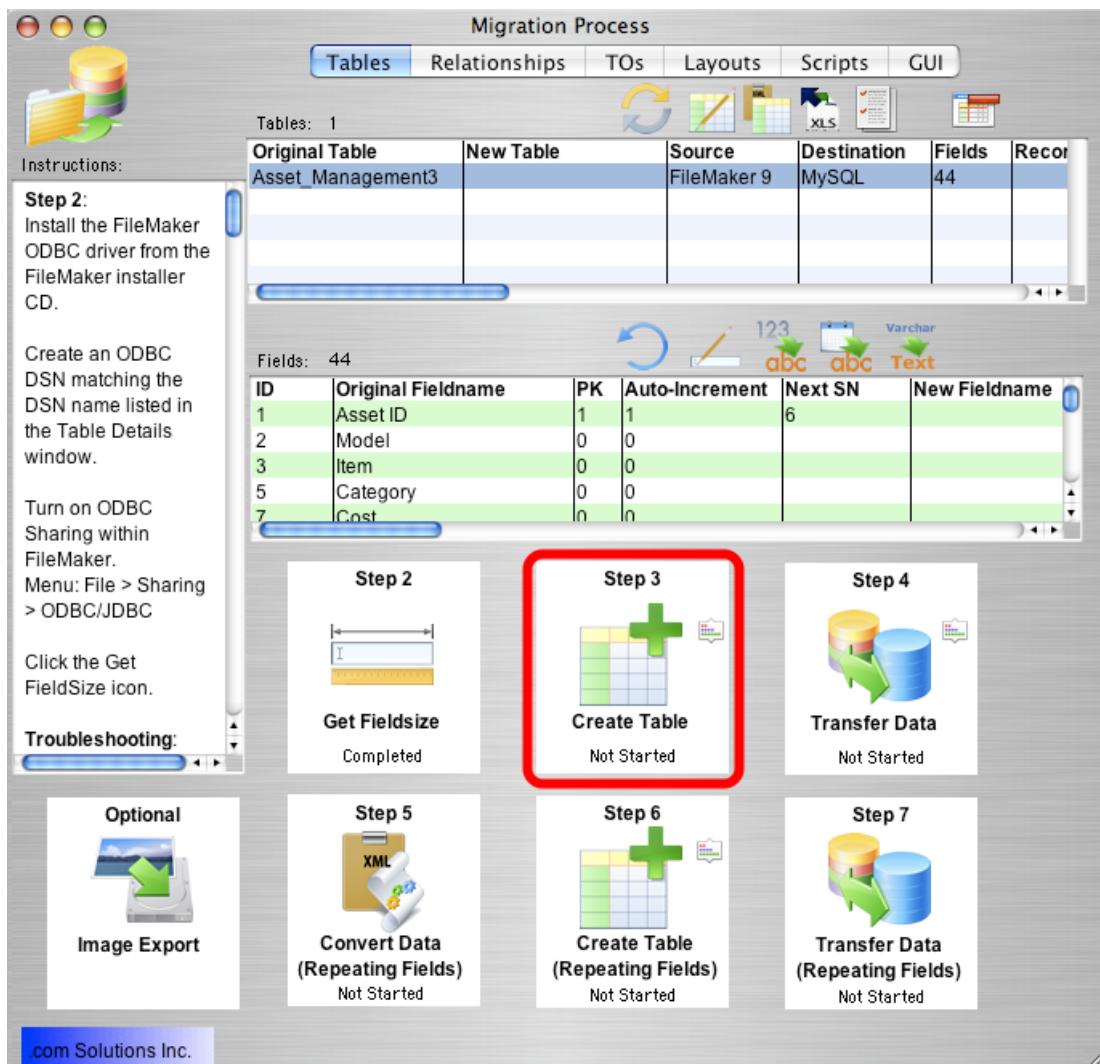
- 2) Make sure that ODBC sharing is enabled.
- 3) For FileMaker 7+ databases, make sure that there is a TO on the RelationshipGraph which exactly matches the name of the base table being queried by FmPro Migrator.
- 4) Open the Define/Manage Database dialog and verify the existence of the base table within the FileMaker database.

## Step 3 - Create Table

Once the Get Fieldsize step has been completed, FmPro Migrator has enough information to create the SQL code defining the table in the destination database. The fieldsize info is automatically used when creating the table in the SQL database.

**Note1:** If you are using FmPro Migrator 8.01 or higher, FmPro Migrator will automatically write the tables into the dbo schema for SQL Server, and directly to the table without a schema designation for other databases.

### Step 3 - Create Table



Click the Step 3 Create Table button to create the table in the destination database.

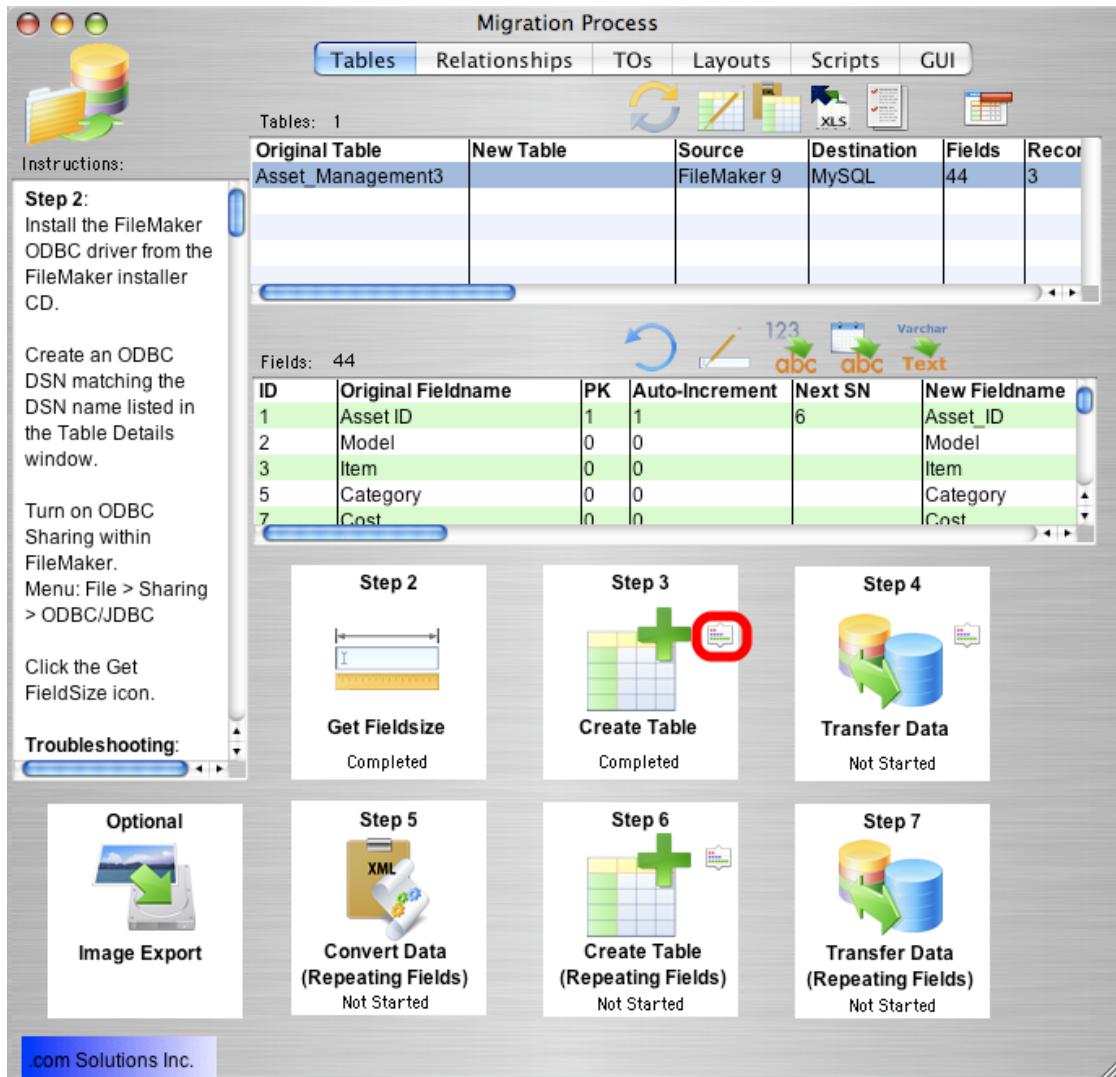
FmPro Migrator will generate the table creation SQL code, connect to the destination database

and create the table. Once the table has been successfully created, the status menu under the Step 3 Create Table button will be changed to "Completed".

Holding down the shift key while clicking the Step 3 Create Table button drops and re-creates the table in the destination database.

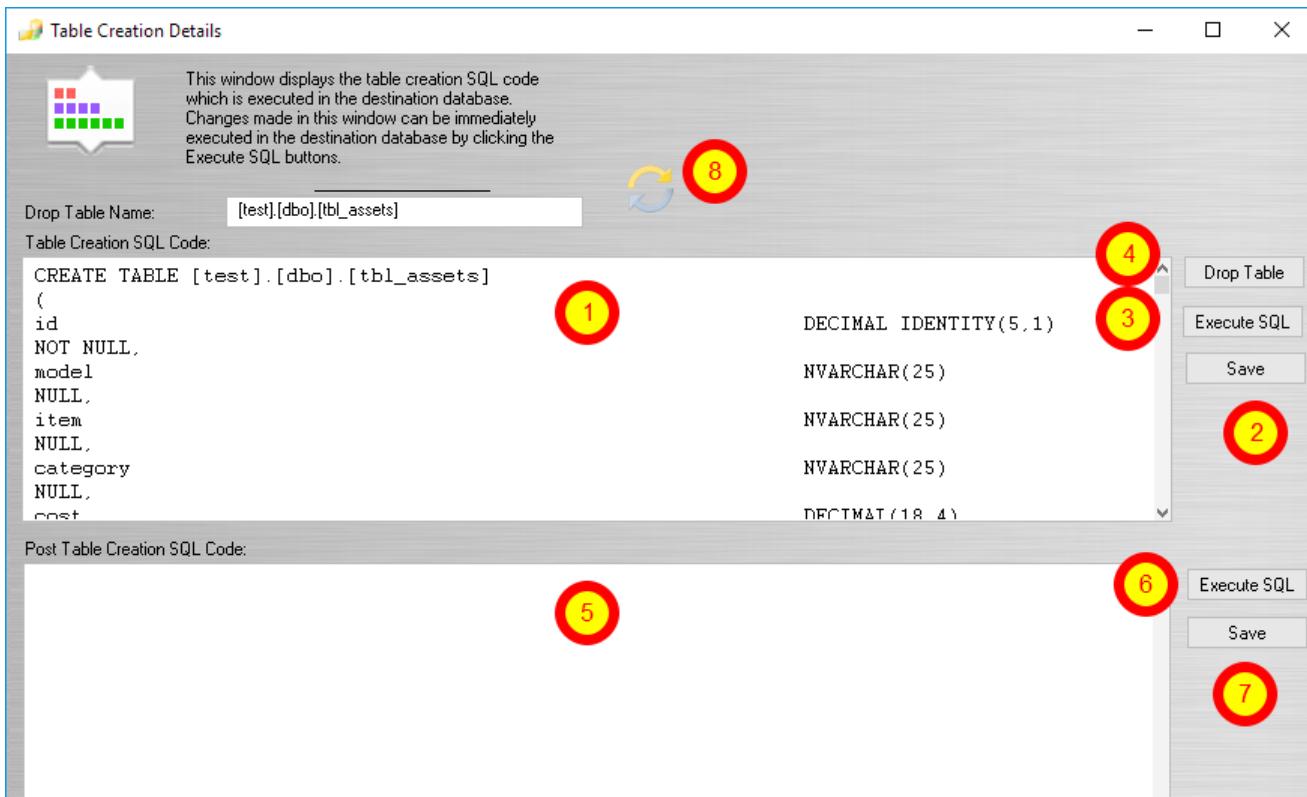
**Warning:** Dropping the table in the destination database causes the loss of all data within the table.

### Step 3 - Table Creation SQL Details



if problems occur while creating the table in the destination database, you can review the table creation SQL code generated by FmPro Migrator by clicking the Table Creation Details button, next to the Step 3 Create Table icon.

### Step 3 - Create Table - Table Creation Details Window



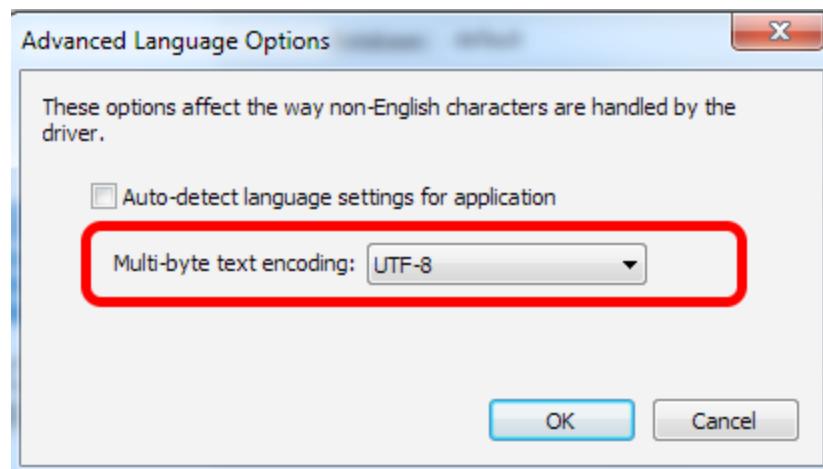
From within the Table Creation Details window, the Table Creation SQL code can be (1) manually edited, (2) saved or (3) re-executed in the destination database. The table can also be (4) dropped in the destination database.

The Post Table Creation SQL code can also be (5) manually edited, (6) executed, and (7) saved.

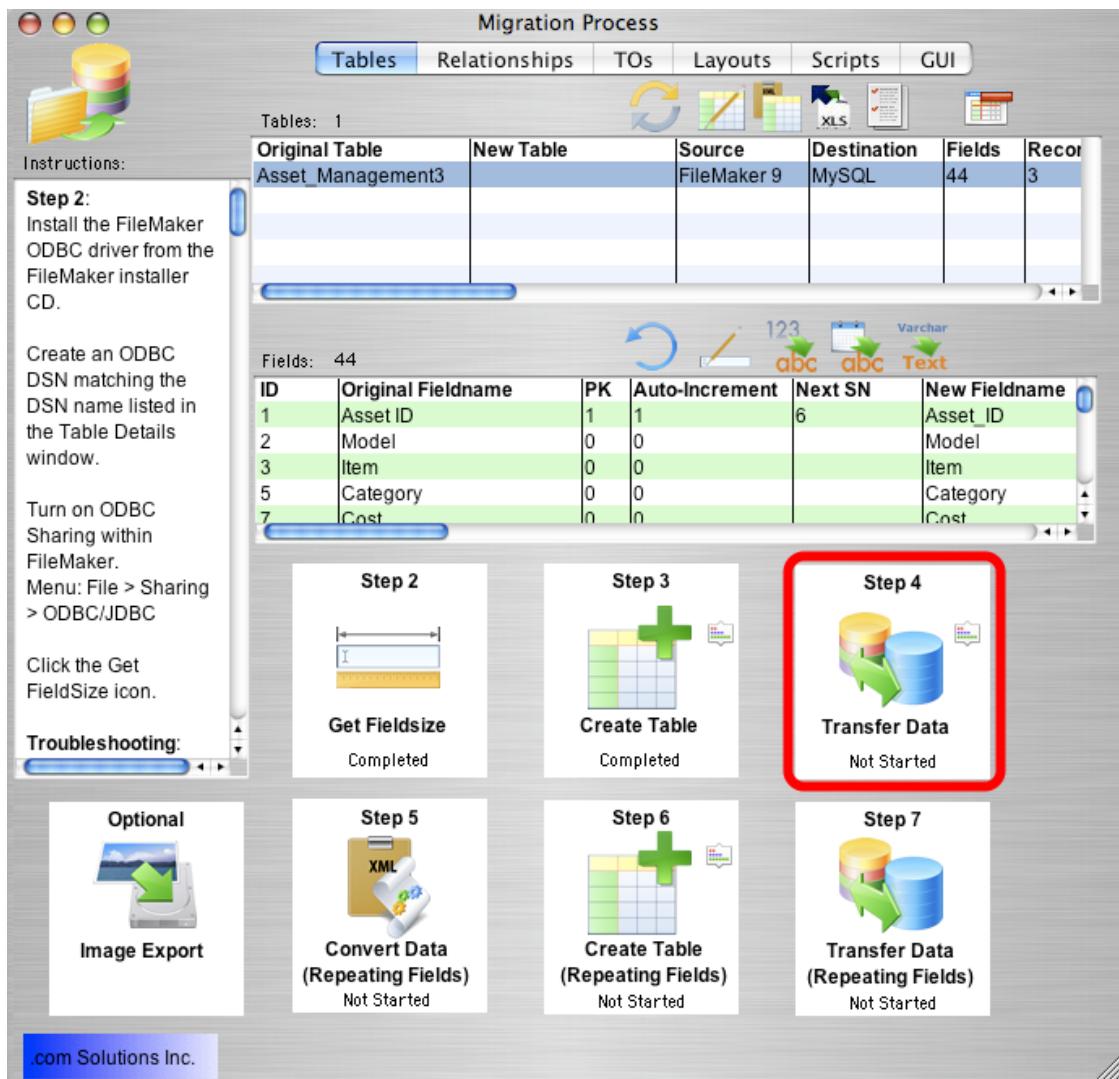
Clicking the (8) refresh button re-loads the previously stored version of the SQL code from the SQLite database.

## Step 4 - Transfer Data - MySQL

### Step 4 - Transfer Data - Unicode Characters



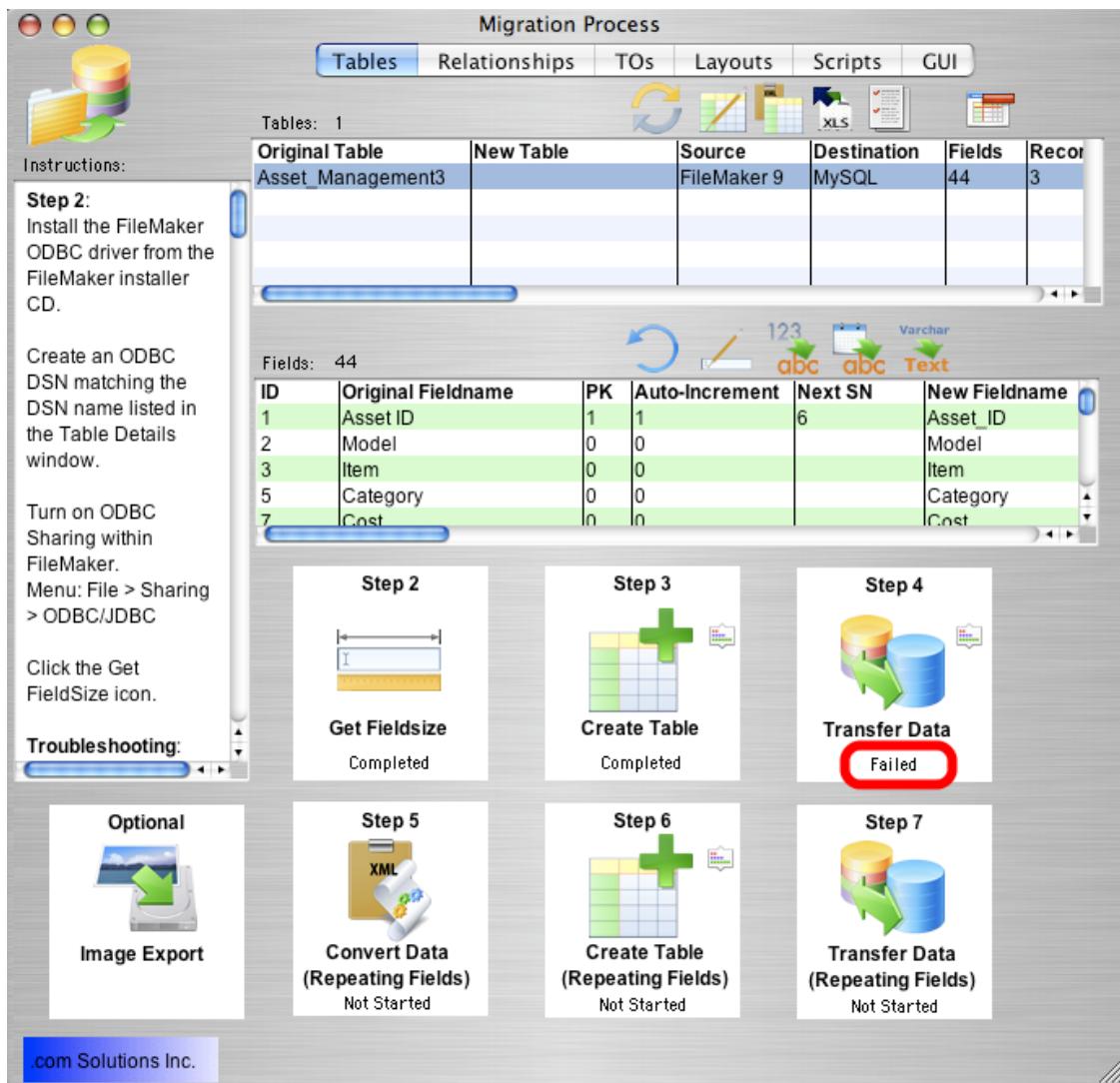
## Step 4 - Transfer Data



Click the Step 4 Transfer Data button to transfer data from the source table in the FileMaker database to the newly created table in the destination database. Once the data has been transferred successfully, the status menu below the Step 4 Transfer Data button will change from "Not Started" to "Completed".

**Note:** You need to hold down the Shift key when clicking the Transfer Data step for full Unicode compatibility. A small command prompt window will show the data transfer in progress.

## Step 4 - Transfer Data - Troubleshooting



If the data transfer process fails, an error message will be displayed containing the text of the error message returned by the destination database.

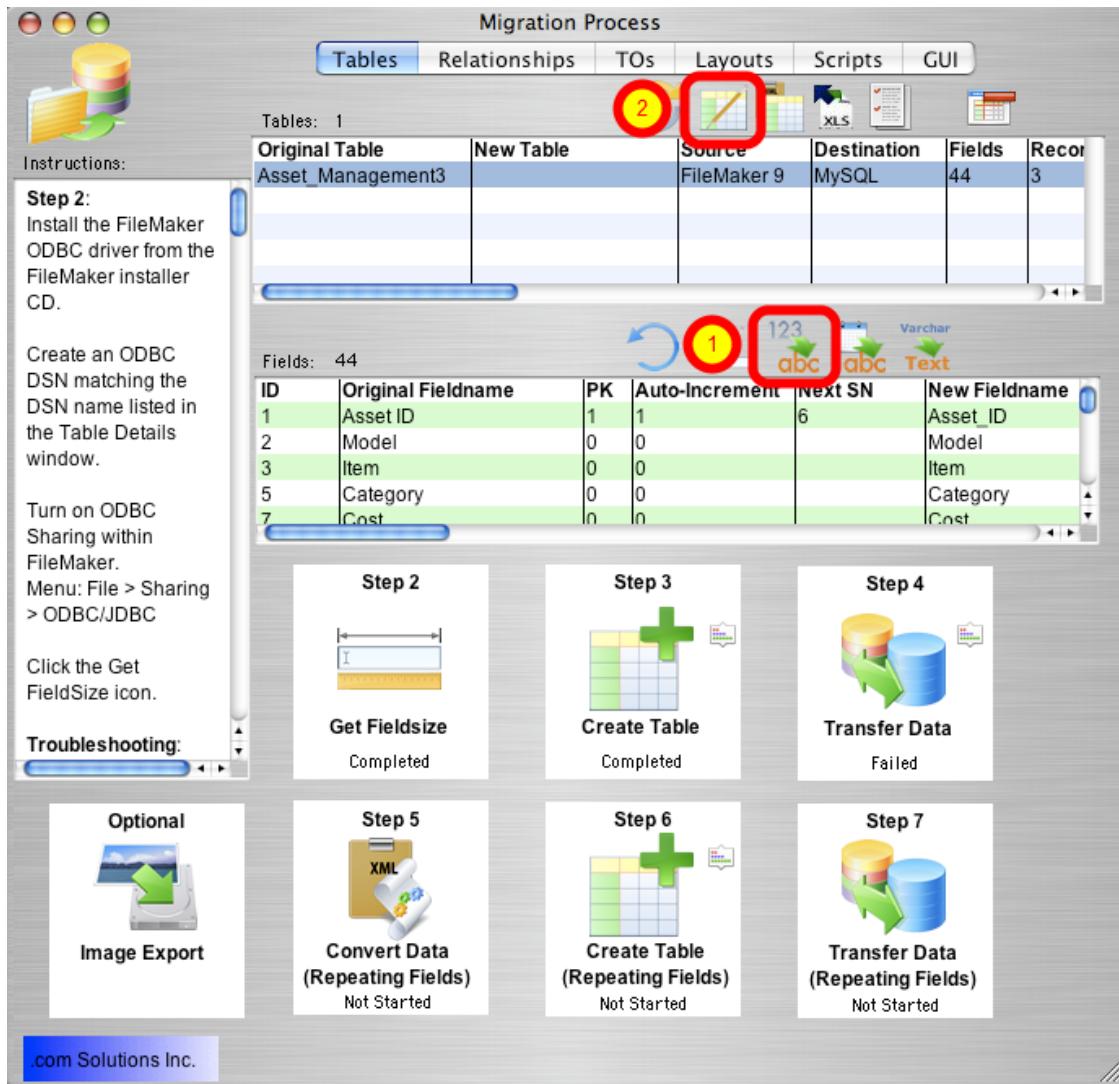
Some of the most common data transfer errors include:

FileMaker numeric fields containing non-numeric data.

FileMaker date fields containing non-date/time data.

FileMaker primary key fields containing duplicate or empty values.

## Step 4 - Transfer Data - Troubleshooting - Non-Numeric Data Type Error



The FileMaker application is very forgiving of the types of data which can be entered into each field. Therefore it is possible for non-numeric data to be entered into numeric fields if field validation has not been enabled for these fields. However SQL databases generally won't accept non-numeric data within numeric fields, which will cause an error during the data transfer process.

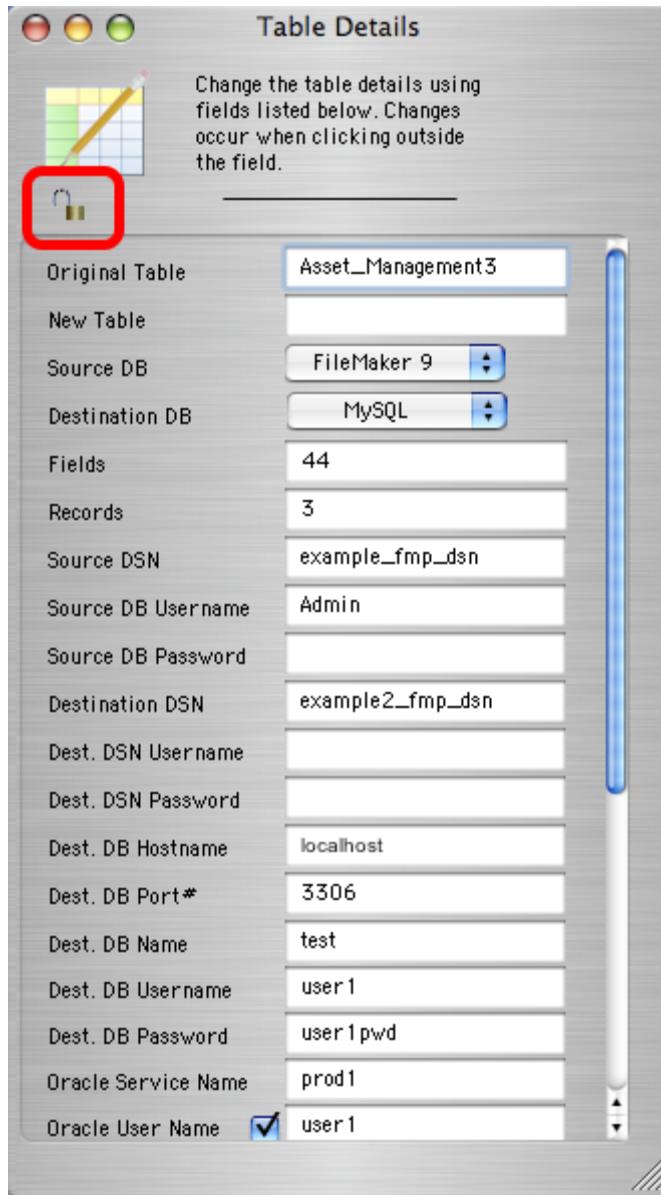
If this type of error occurs:

Option 1: Correct the data within the FileMaker database in order to resolve the problem. Then drop and re-create the table in the destination database and click the Step 4 Transfer Data button again.

Option 2: Correct the data after transferring the data to the destination database. To get the data

transferred to the destination database, click the Set Numeric Fields to Text button, above the Fields list. This button instructs FmPro Migrator to change all of the numeric field types to varchar field types in the table creation SQL code for the destination table.

#### Step 4 - Transfer Data - Lock Table Details



After clicking the Numeric to Varchar, Date to Varchar or Varchar to TEXT buttons, open the Table Details window and click the lock icon. Setting the lock icon prevents FmPro Migrator from re-setting the field types back to their original values when clicking the Step 3 Create Table button.

Hold down the Shift key and click the Step 3 Create Table button to drop and re-create the table in the destination database, then click the Step 4 Transfer Data button again.

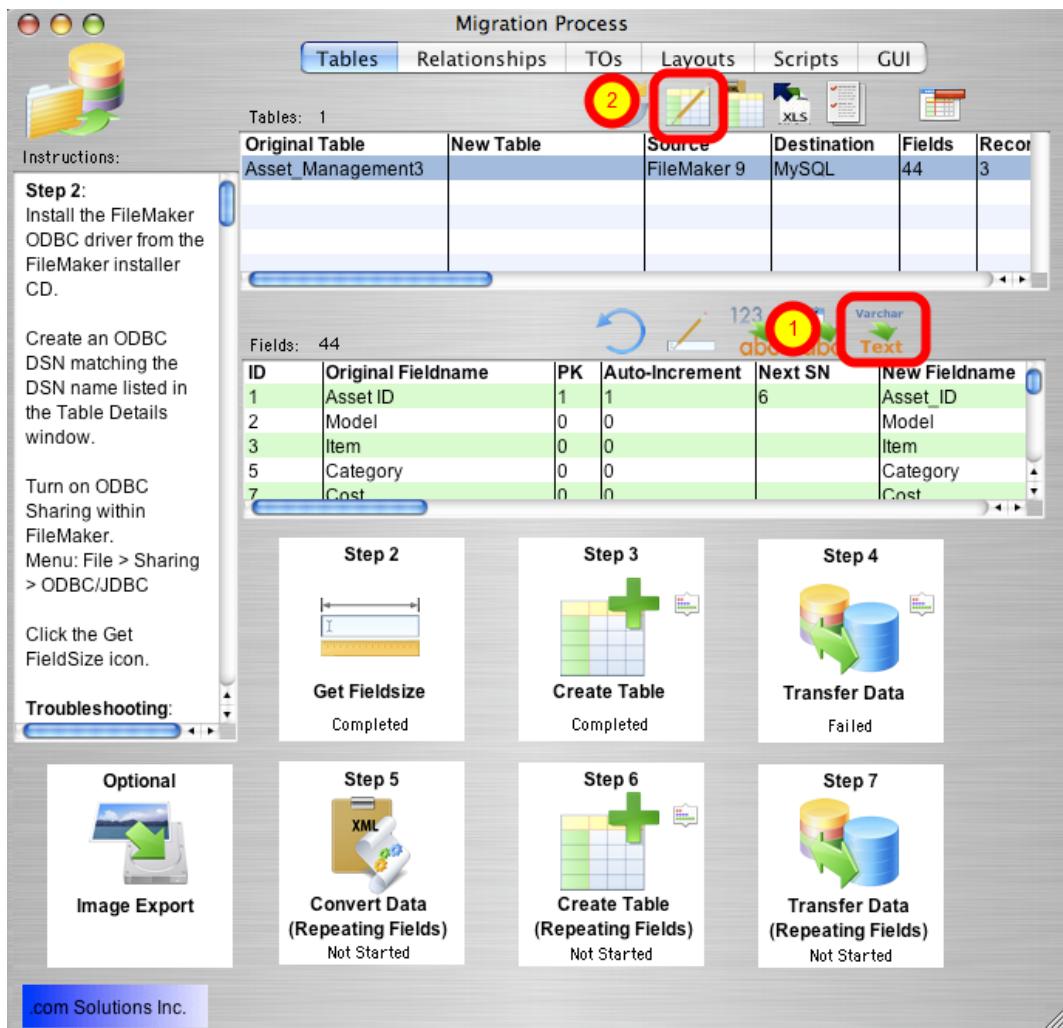
Note: You must generate table creation SQL code by clicking the Step 3 Create Table button at least once, before clicking the lock icon on the Table Details window.

#### Step 4 - Transfer Data - Troubleshooting - Non-Date/Time Data Type Error

If data transfer errors occur as a result of having non-date/time data within Date/Time fields, clicking the Set Date/Time Fields to Varchar button will change these column types to varchar columns in the destination database.

Open the Table Details window, click the Lock icon, then hold down the Shift key while clicking the Step 3 Create Table button.

#### Step 4 - Transfer Data - Varchar to TEXT Conversion

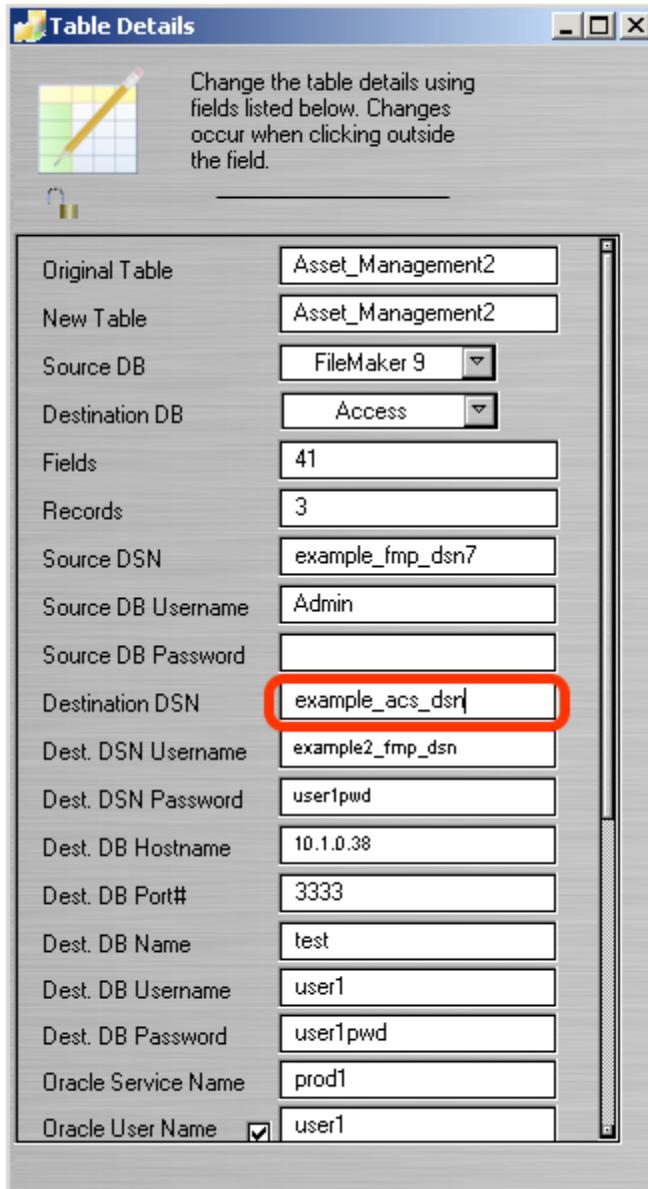


It is not usually necessary to change Varchar columns to TEXT column types in the destination database. But occasionally you might need to perform this task. If you wanted to perform a quick migration, without running the Get Fieldsize step, you could just convert all of the Varchar columns

to TEXT column types without worrying about the exact size of the data being transferred. FmPro Migrator will utilize the correct column type in the destination database for storing the largest amount of text.

After clicking the Varchar to TEXT button, open the Table Details window, click the Lock icon, then hold down the Shift key while clicking the Step 3 Create Table button. This process will drop and re-create the table in the destination database, creating the new table with the requested column type changes.

#### Step 4 - Transfer Data - Troubleshooting



Error Message: Count Error (1).

If an unexpected error occurs during the data transfer process, there is an alternate data transfer

process built into FmPro Migrator which can be used. Hold down the shift key when clicking on the Transfer Data button. The alternate data transfer method will open a separate console window and will run a compiled Perl program to transfer the data between databases.

If transferring data to an Access database, create an ODBC DSN to the destination database, as shown in the Table Details window. Open the Table Details window by double-clicking on the table in the list of tables.

#### **Step 4 - Transfer Data Troubleshooting - Shift Key Method**

If some other unexpected data transfer error occurs, you might hold down the shift key when clicking the Data Transfer button.

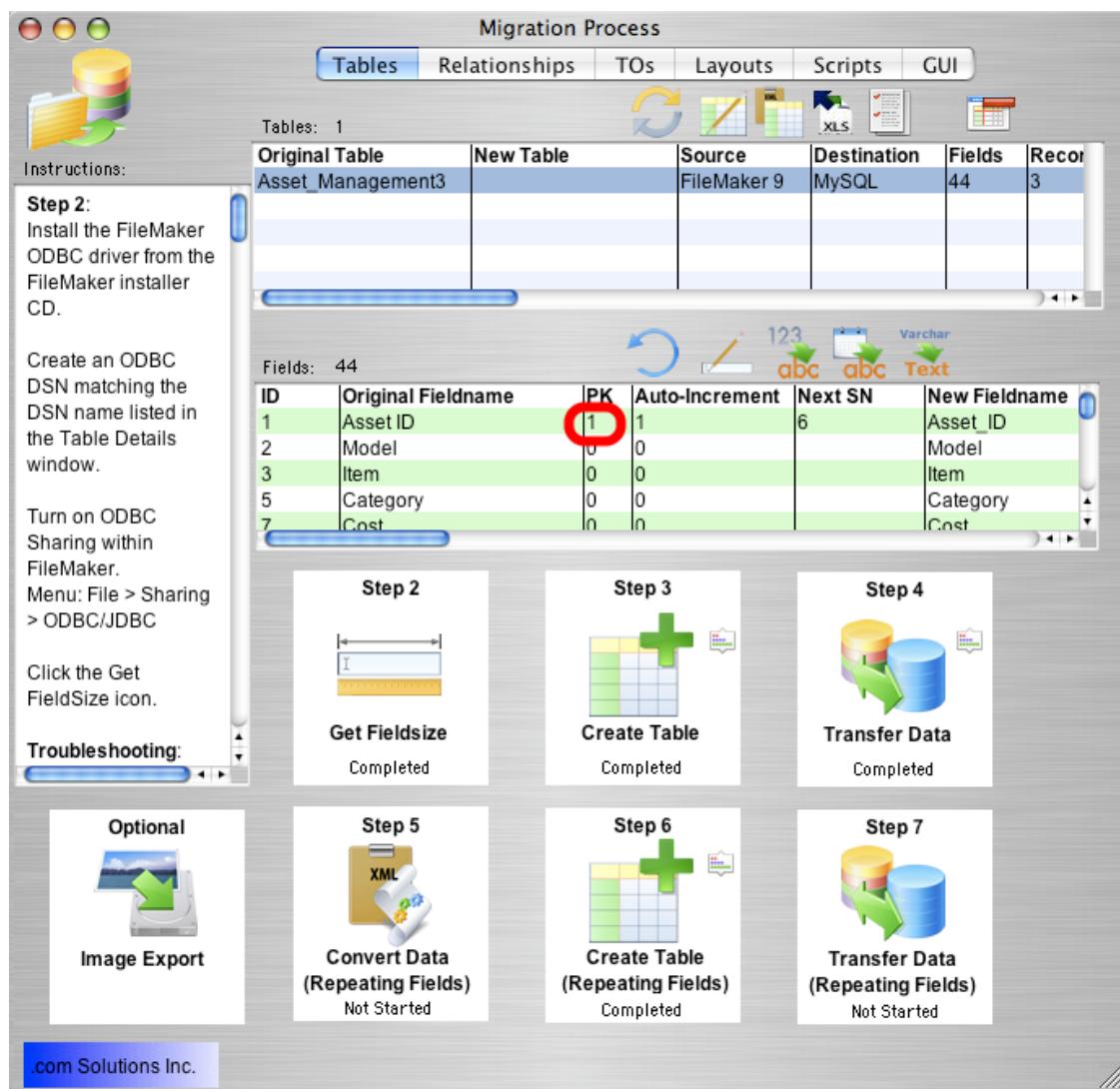
After you do this, a command prompt (Windows) or console window (macOS) will be displayed showing the data transfer using xferDataPerl scripts.

## Step 5 - Convert Data (Repeating Fields)

The buttons for Steps 5, 6 and 7 will only be visible if the source FileMaker Pro database table contains repeating fields.

FmPro Migrator converts non-relational repeating fields data into child records referencing the parent table record using the primary key of the parent table. The child table is created in the destination database using the naming convention *parent\_table\_repeating*. Each individual repeating fields data value is written into a separate record within this table.

### Repeating Fields Processing - Preparation Steps



In order to accurately transfer repeating fields data, FmPro Migrator needs to determine the Primary Key field within the source FileMaker database table. FmPro Migrator identifies the Primary Key field as being a field in which Unique and Not Empty data validation is configured. The Primary Key field contains a "1" in the PK column as shown in the field list.

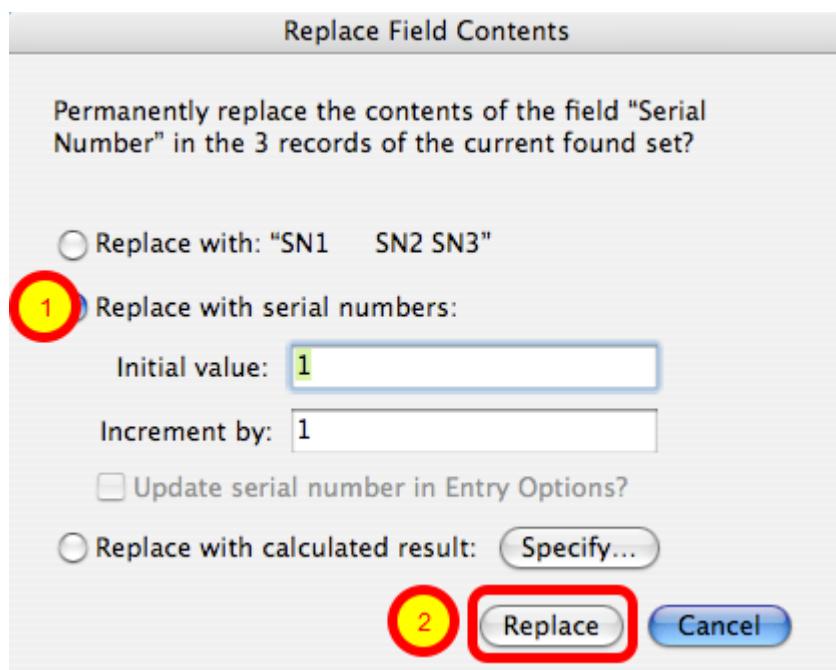
If the Primary Key field contains empty or non-unique values, then the data needs to be corrected, or a new Primary Key field needs to be defined. Only one field should be defined as the Primary Key, and this field should be defined as a numeric field, as SQL databases don't generally handle Auto-Incrementing Text and Numeric values within the same field (as FileMaker easily does).

To create a new Primary Key field within the FileMaker database, create a numeric field having Unique and Not Empty field validation. Disable the Unique and Not Empty validation for the original Primary Key field.

Select the Records > Show All Records menu item, to make sure that all records will be affected by the Replace Field Contents feature.

After creating the new field, put the cursor into this new field, then select the Records > Replace Field Contents menu item.

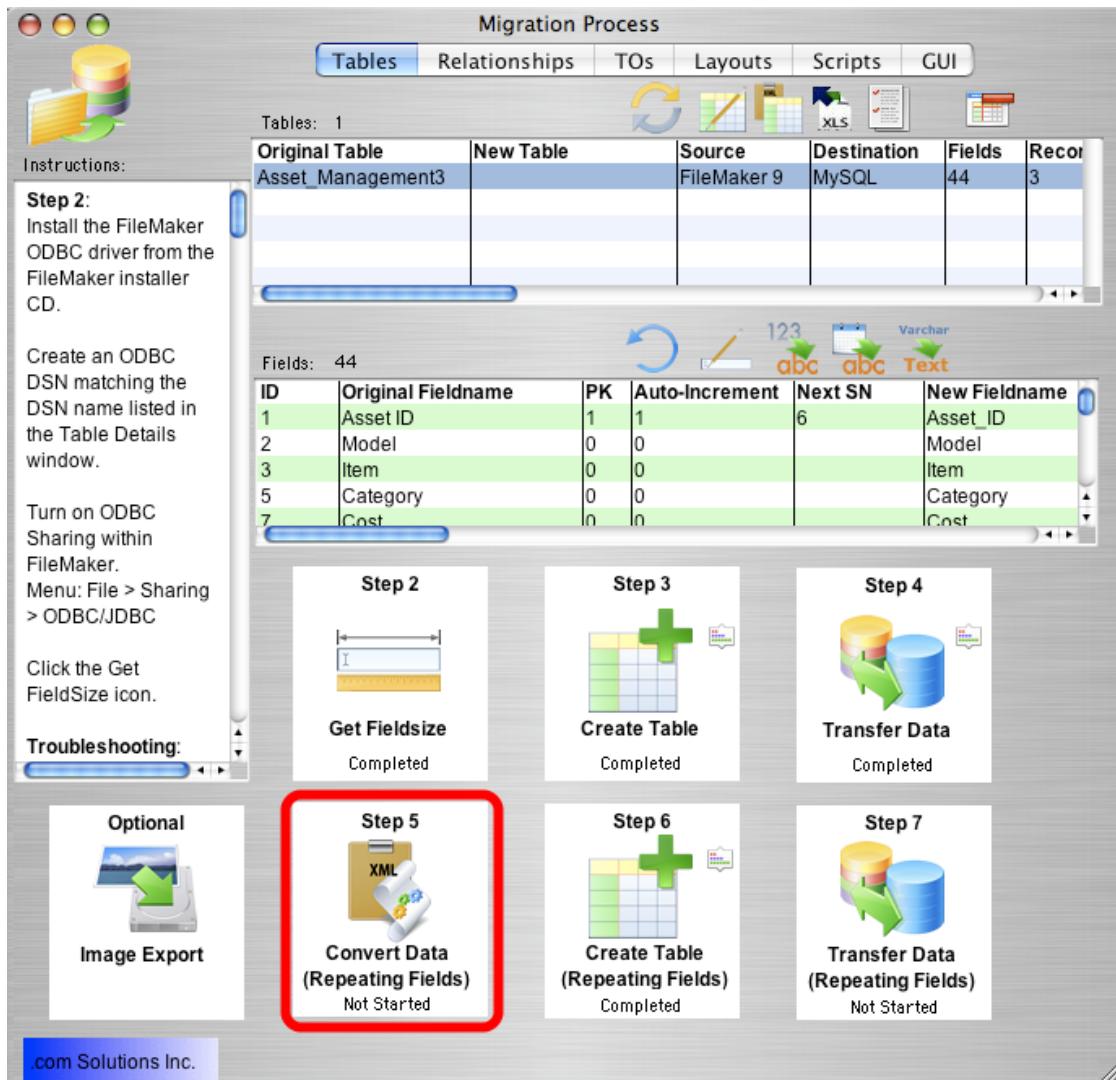
### Repeating Fields Processing - Creating New Primary Key



Using Replace Field Contents, replaces the empty values within the new Primary Key field with an incrementing series of numbers, within all records of the source table.

After adding this new Primary Key field and replacing field contents, you will need to start the migration process over starting with Step 1, since the structure of the source table has changed and FmPro Migrator needs to be aware of any changes which have been made to the source table.

## Step 5 - Convert Data (Repeating Fields) - FileMaker 7+ Only



For FileMaker 7+ source database tables, Step 5 processing is designed to change the contents of repeating fields data so that it can be transferred thru the FileMaker 7+ ODBC driver. FileMaker 7+ ODBC drivers are no longer capable of transferring data from repeating fields within the FileMaker database. To work-around this ODBC Driver limitation, FmPro Migrator moves all of the repeating fields data into the first repeating data value, and separates each repeat value with a TAB data separator.

When the Step 5 Convert Data (Repeating Fields) button is clicked, FmPro Migrator generates FileMaker Script Workspace scripts to convert the data within each repeating field data and puts this script onto the ClipBoard.



Repeating Fields conversion scripts have been put onto the clipboard.

- 1) Change Numeric and Date/Time fields to Text within FileMaker.
- 2) Paste these scripts into the Script Workspace within FileMaker Pro.
- 3) Run the script named:  
RF Convert – for Table: tbl\_Assets

OK

Click the Ok button to the informational dialog.

Open the FileMaker Script Workspace, and paste the script from the ClipBoard into FileMaker.

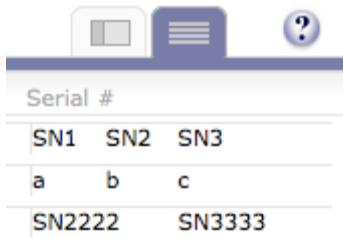
Manually change each of the repeating fields from Numeric or Date/Time format to Text within FileMaker. It is not necessary to re-gather info from FileMaker after making this change.

Run the script shown in the dialog (the one you just pasted into the Script Workspace).

**Warning:** Running the RF Convert script for a particular table is a task which should be performed **one-time only**. It should not be run more than once. To verify whether the RF Convert script has previously been run on a table, check to see if all of the repeating field data values have been moved to the first repeating field occurrence within the database.

If FileMaker gets interrupted while running the script, it's generally not possible to start the process over again with the remaining records. For instance, the script could stop if a data validation error occurs with one of the other fields of the database table. If this type of problem occurs, then it will be necessary to get a copy of the database file, copy and run this script again after solving the problem.

## Step 5 - Convert Data (Repeating Fields) - Verifying Data Conversion



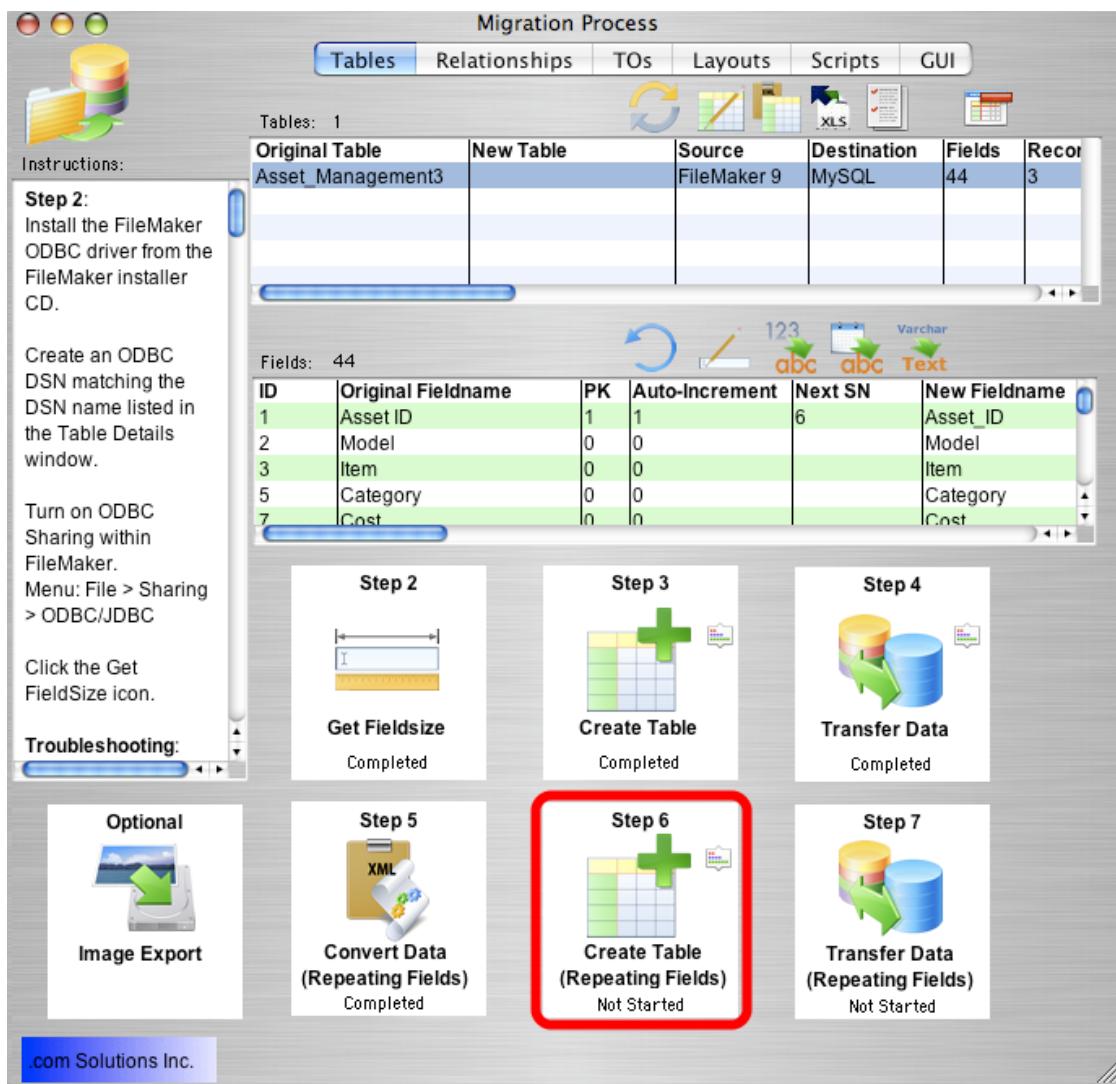
SN1	SN2	SN3
a	b	c
SN2222		SN3333

This image shows the contents of a repeating field shown on the FileMaker layout as "Serial #", which has been successfully converted with the RF Convert ScriptMaker script. If the RF convert script had not been run, then there would be vertical lines separating each repeating value. But since all of the repeating field data values have been placed into the first repeat occurrence, these vertical lines are not visible.

FmPro Migrator sets the status of the Step 5 Convert Data (Repeating Fields) step to "In Progress" when you click the button to generate and put the script onto the ClipBoard. FmPro Migrator doesn't know when you have actually run the RF Convert script within the Script Workspace, so you should click the pop-up menu and set the status of this step to "Completed" once you have completed this task. Keeping the status selections updated can be helpful if you get interrupted while performing the migration, so that you will know which steps you have completed for each table to be migrated.

## Step 6 - Create Table (Repeating Fields)

### Step 6 - Create Table (Repeating Fields)



Click the Step 6 Create Table (Repeating Fields) button to create the repeating fields table in the destination database.

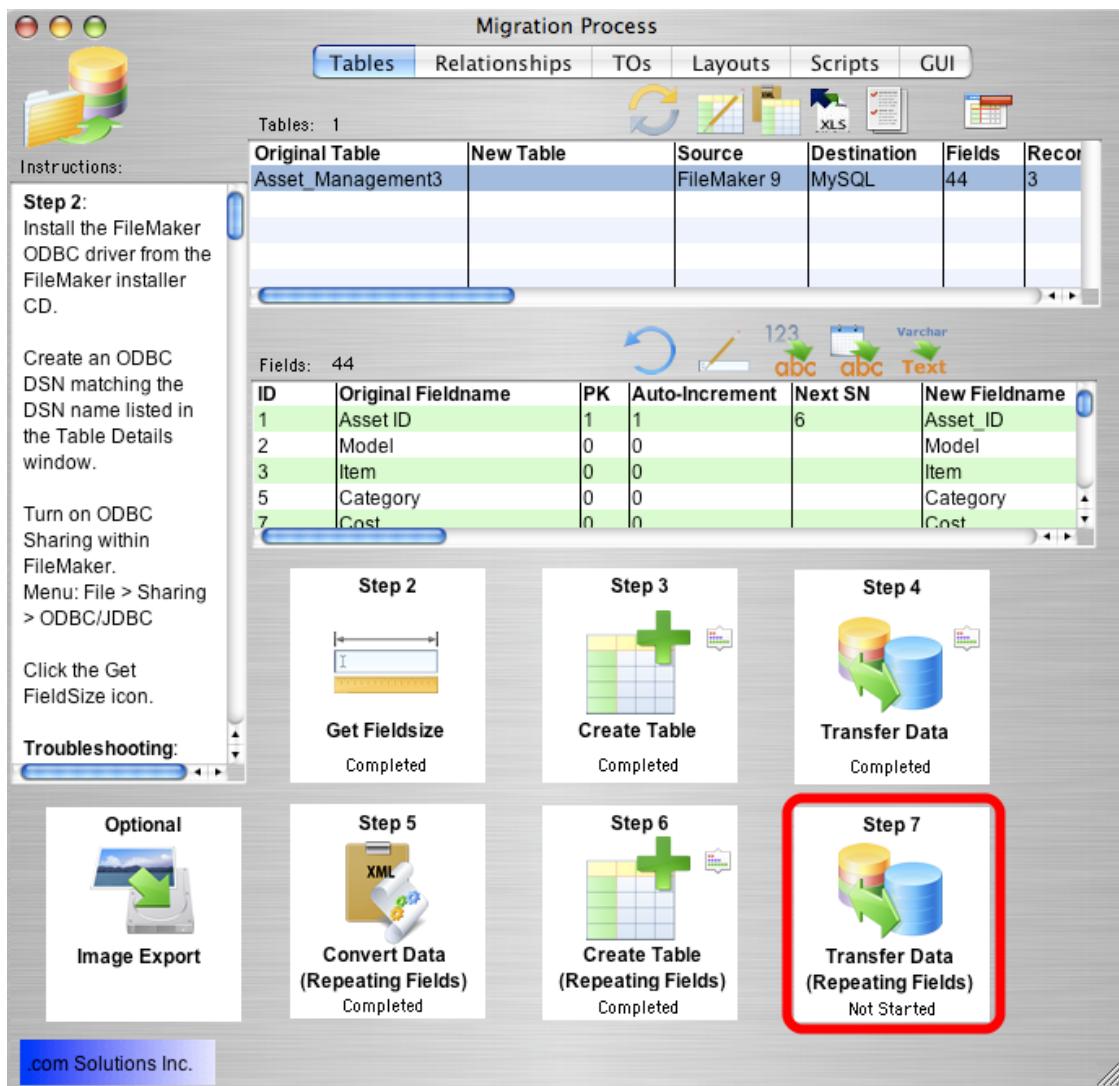
FmPro Migrator will generate the table creation SQL code, connect to the destination database and create the table. Once the repeating fields table has been successfully created, the status menu under the Step 6 Create Table (Repeating Fields) button will be changed to "Completed".

Holding down the shift key while clicking the Step 6 Create Table (Repeating Fields) button drops and re-creates the table in the destination database.

**Warning:** Dropping the table in the destination database causes the loss of all data within the table.

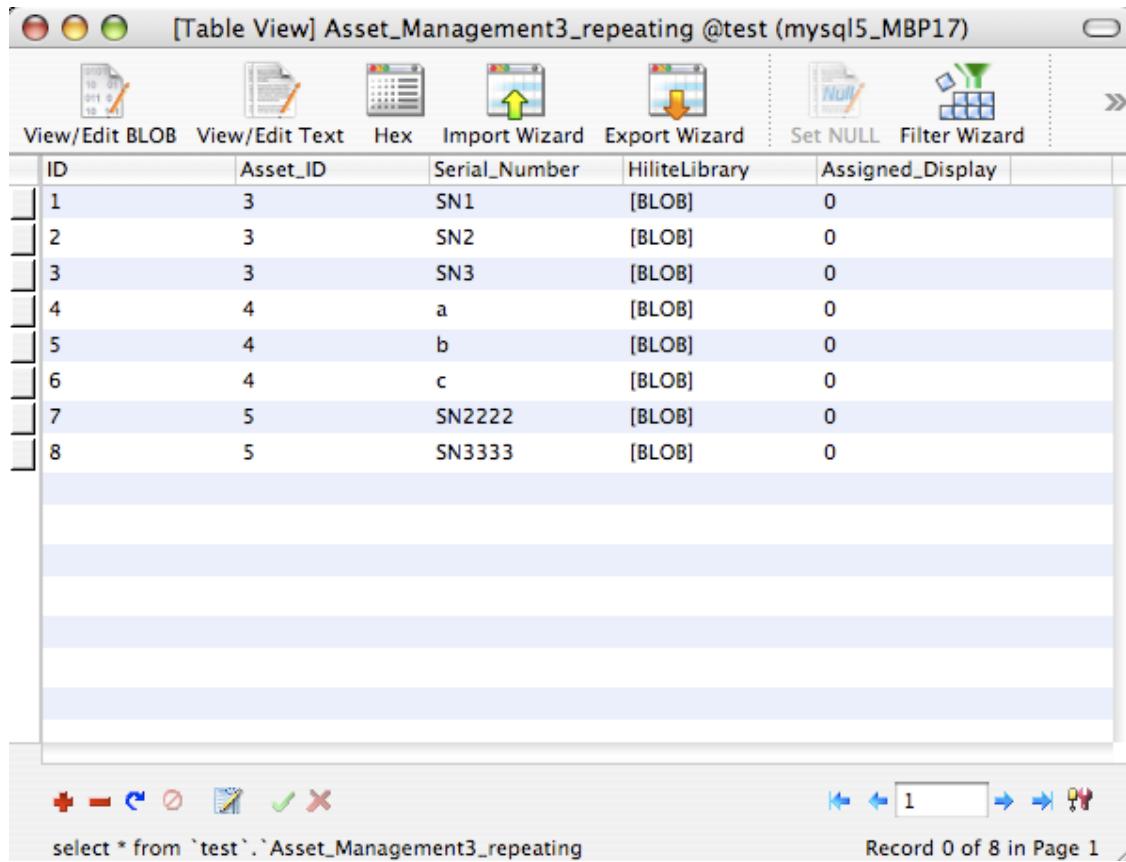
## Step 7 - Transfer Data (Repeating Fields) - MySQL

### Step 7 - Transfer Data (Repeating Fields)



Click the Step 7 Transfer Data (Repeating Fields) button to transfer data from the source table in the FileMaker database to the newly created repeating fields table in the destination database. Once the data has been transferred successfully, the status menu below the Step 7 Transfer Data (Repeating Fields) button will change from "Not Started" to "Completed".

## Step 7 - Transfer Data (Repeating Fields) - Troubleshooting



The screenshot shows a table named 'Asset\_Management3\_repeating' in the 'test' database. The table has columns: ID, Asset\_ID, Serial\_Number, HilitLibrary, and Assigned\_Display. The data consists of 8 rows, each with a different ID and Asset\_ID, and a Serial\_Number value followed by '[BLOB]'. The HilitLibrary column contains the value 0 for all rows. The Assigned\_Display column also contains the value 0. The table is displayed in 'Table View' mode, with various toolbar icons for blob, text, hex, import, export, and filtering.

ID	Asset_ID	Serial_Number	HilitLibrary	Assigned_Display
1	3	SN1	[BLOB]	0
2	3	SN2	[BLOB]	0
3	3	SN3	[BLOB]	0
4	4	a	[BLOB]	0
5	4	b	[BLOB]	0
6	4	c	[BLOB]	0
7	5	SN2222	[BLOB]	0
8	5	SN3333	[BLOB]	0

select \* from `test`.`Asset\_Management3\_repeating`

Record 0 of 8 in Page 1

All of the same troubleshooting steps apply to transferring repeating fields data as were mentioned within the Step 4 instructions. But one additional consideration involves reviewing the data transferred into the repeating fields.

Reviewing the data validates whether the each of the steps were processed correctly. For instance, if only 1 repeating value was transferred to the destination table, then this could mean that the RF Convert script wasn't run or the contents of the source table fields weren't converted into Text prior to processing. If either of these problems occur, it is a simple matter to go back and re-do those processing steps, drop & re-create the table then press the Step 7 button to transfer the data again.

This screenshot of a repeating fields table displayed within Navicat shows that the individual repeating fields data values for the Serial\_Number column were correctly separated into individual child records related to the parent table via the primary key (Asset\_ID column). In this example, you should ignore the BLOB column named HilitLibrary as repeating fields data for FileMaker container fields can't be transferred between databases.

## Image Transfer (Optional Step)

---

Images (and other types of data within FileMaker 7+) can be transferred directly into supported SQL databases [but not Microsoft Access]. In order to perform this type of export, you need to have at least one container field within the FileMaker Pro database table and a primary key within the table.

### Image Transfer - Prerequisites

Binary files of all types are supported when the FileMaker Insert File... feature has been used to insert the file. This feature would commonly be used for storing spreadsheets and other files.

The database table must have a primary key (including numeric and non-numeric primary keys consisting of UUIDs).

FmPro Migrator is able to transfer FileMaker container field data from embedded container fields.

The transfer of the following container field data is not supported:

External Container Fields

Files stored as a Reference

Very old unsupported embedded data types including: including: XMLO - FileMaker layout, JP2 - JPEG 2000, Quicktime, PICT, MacPaint, FPX - FlashPix, .SGI, TGA- Targa

## Image Transfer

Migration Process

Tables Relationships TOs Value Lists CFs Layouts Scripts GUI

Tables: 7

Original Table New Table Source Destination Fields Records

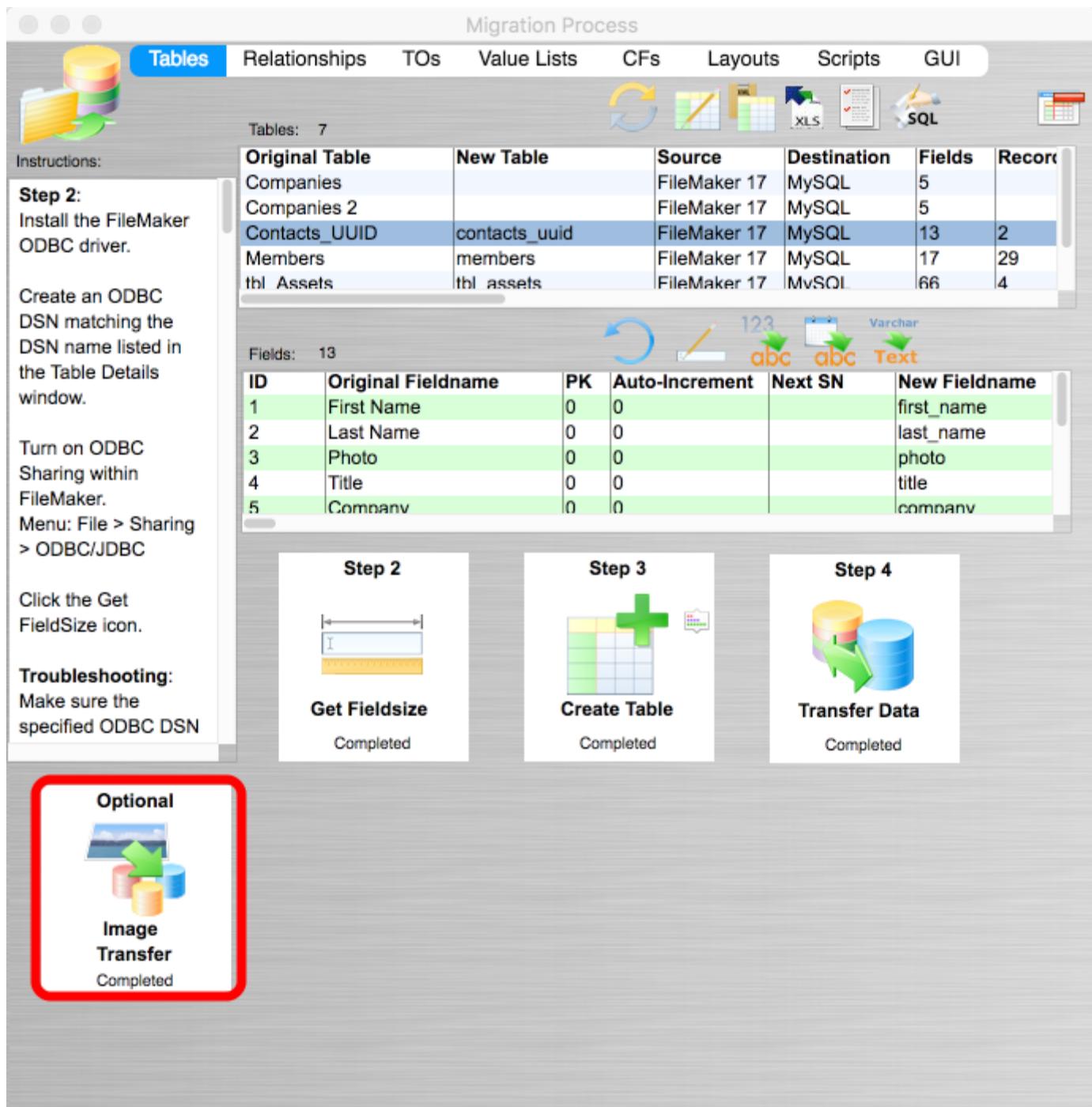
Companies		FileMaker 17	MySQL	5	
Companies 2		FileMaker 17	MySQL	5	
Contacts_UUID	contacts_uuid	FileMaker 17	MySQL	13	2
Members	members	FileMaker 17	MySQL	17	29
tbl Assets	tbl_assets	FileMaker 17	MySQL	66	4

Fields: 13

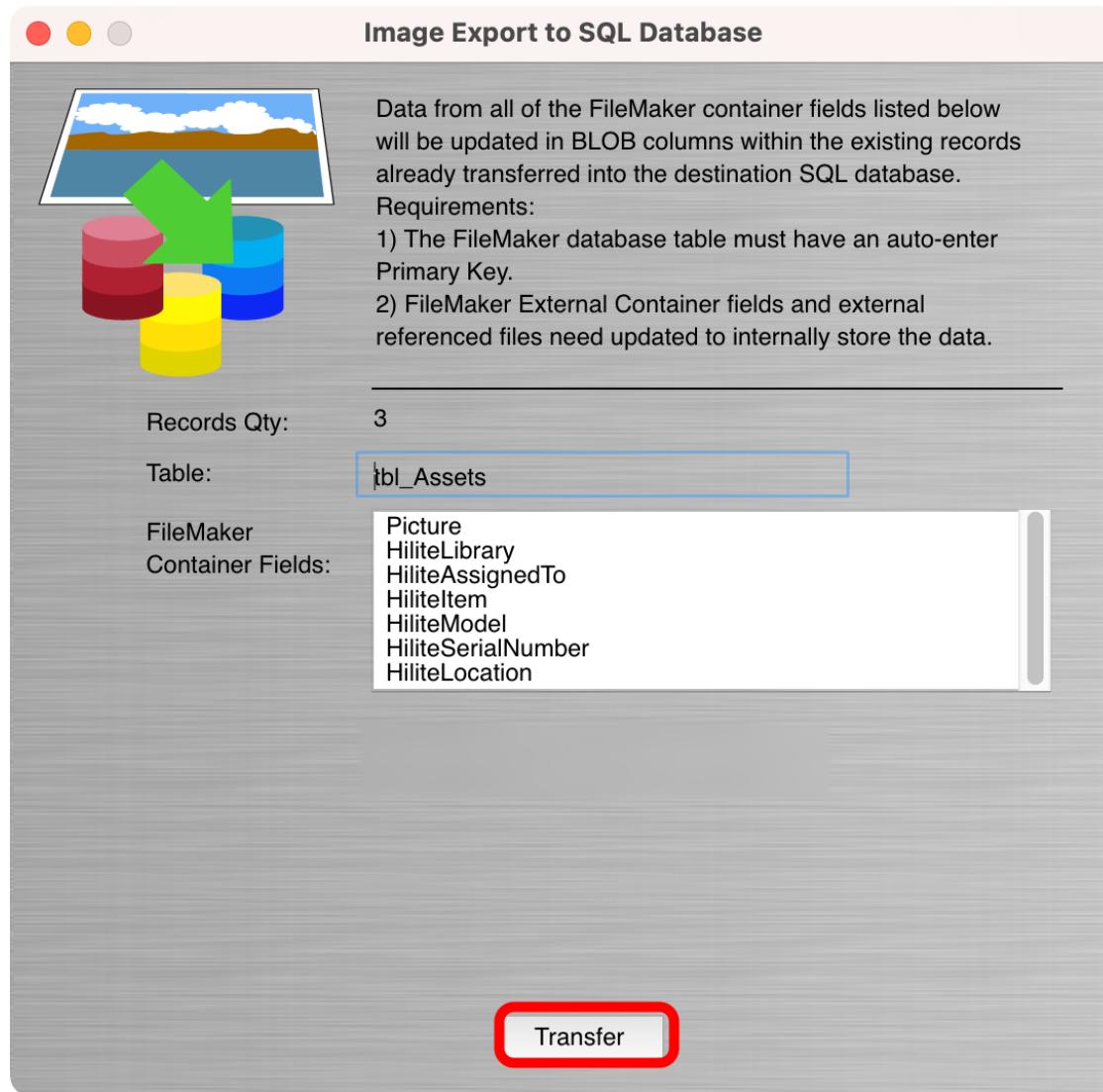
ID	Original Fieldname	PK	Auto-Increment	Next SN	New Fieldname
1	First Name	0	0		first_name
2	Last Name	0	0		last_name
3	Photo	0	0		photo
4	Title	0	0		title
5	Company	0	0		company

Step 2: Get Fieldsize Step 3: Create Table Step 4: Transfer Data

Optional Image Transfer Completed

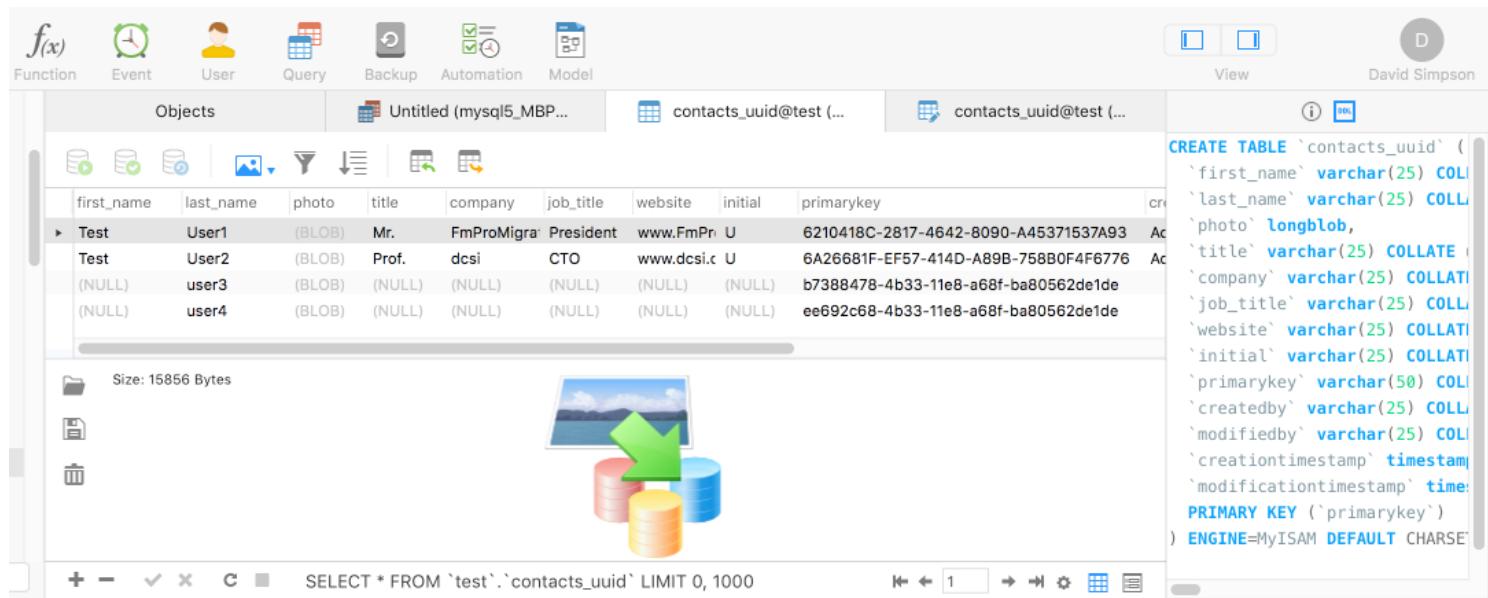


Click on the Image Transfer button to transfer images from the FileMaker Pro database table. This button will not be displayed if there are no container fields within the table.



Click the Transfer button. FmPro Migrator will read thru all of the records and transfer container field data for all of the records into the SQL database.

## Imager Transfer - and UUID PK Displayed - MySQL



first_name	last_name	photo	title	company	job_title	website	initial	primarykey
Test	User1	(BLOB)	Mr.	FmProMigr	President	www.FmPr	U	6210418C-2817-4642-8090-A45371537A93
Test	User2	(BLOB)	Prof.	dcsi	CTO	www.dcsi.c	U	6A26681F-EF57-414D-A89B-758B0F4F6776
(NULL)	user3	(BLOB)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	b7388478-4b33-11e8-a68f-ba80562de1de
(NULL)	user4	(BLOB)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	ee692c68-4b33-11e8-a68f-ba80562de1de

Size: 15856 Bytes

SELECT \* FROM `test`.`contacts\_uuid` LIMIT 0, 1000

```
CREATE TABLE `contacts_uuid` (
  `first_name` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `last_name` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `photo` longblob,
  `title` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `company` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `job_title` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `website` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `initial` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `primarykey` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `createdby` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `modifiedby` varchar(25) COLLATE utf8mb4_unicode_ci NOT NULL,
  `creationtimestamp` timestamp NOT NULL,
  `modificationtimestamp` timestamp NOT NULL,
  PRIMARY KEY (`primarykey`),
  ENGINE=MyISAM DEFAULT CHARSET=utf8mb4
)
```

For MySQL database servers, FileMaker fields having an Auto-Enter Get (UUID) calculation will be created with a MySQL database trigger having the same functionality. The first 2 records were created in FileMaker and transferred into MySQL. The last 2 records were created directly in MySQL. Notice that the UUID values look a little different - in MySQL only the first section of the UUID changes value.

## MySQL Troubleshooting

---

### **MySQL max\_allowed\_packet error - During Data Transfer**

The max\_allowed\_packet error could be displayed while transferring data from FileMaker to the MySQL destination database. This error is most commonly noticed when transferring large amounts of data within FileMaker container fields, but it could occur with other column types too.

Solution: The max-allowed-packet MySQL variable may be increased by entering the updated value into the mysqld configuration file (my.ini on Windows or /etc/my.conf on UNIX):  
--max-allowed-packet=8M

This configuration parameter needs to be updated by the MySQL database administrator.

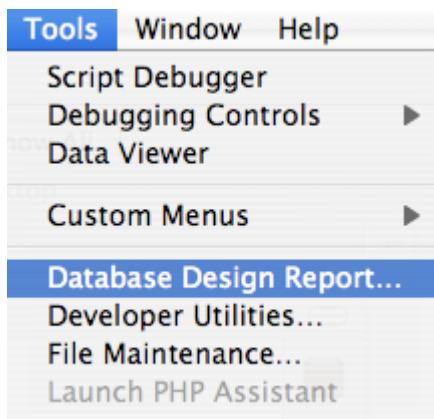
An alternative to storing the images within the database is to export the images from the FileMaker database and storing them outside the MySQL database. FmPro Migrator can export images from container fields to individual files on your hard disk.

## Migrating Relationships - FileMaker to MySQL

FmPro Migrator reads FileMaker relationship info from DDR XML files exported by FileMaker Developer/Advanced 7+. This relationship info is then used to create SQL code to re-create the relationships within the MySQL database.

**Note:** If you are using FmPro Migrator to create a CakePHP web application, you should skip this step. CakePHP implements the MySQL relationships within its application code created by FmPro Migrator.

### Migrating Relationships - Export DDR XML File



Using FileMaker 7+, select Database Design Report... from the Tools menu.

**Note:** Export the DDR file for the database to be migrated before removing Relationships and Table Occurrences from the Relationship Graph.

## Database Design Report

Create an XML or HTML report on the structure of your database(s). The file can be viewed in a web browser. Only files open with full access privileges can be included. A file is marked with a "\*" when a subset of its tables are selected.

### Available files:

<input checked="" type="checkbox"/> Asset_Management3.fp7	1
---	---

### Include fields from tables in selected file:

--

### Include in report:

<input checked="" type="checkbox"/> Accounts	2
<input checked="" type="checkbox"/> Custom Menu Sets	
<input checked="" type="checkbox"/> Custom Menus	
<input checked="" type="checkbox"/> Data Sources	
<input checked="" type="checkbox"/> Extended Privileges	
<input checked="" type="checkbox"/> Functions	
<input checked="" type="checkbox"/> Layouts	

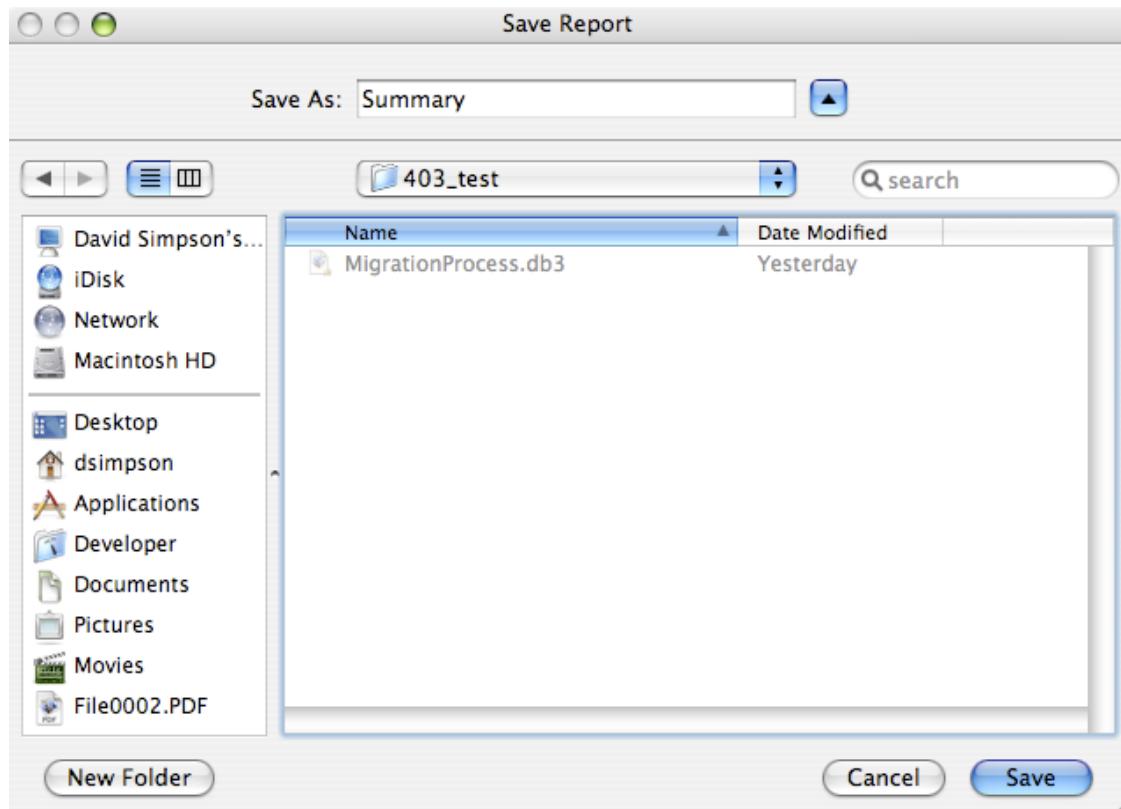
### Report Format:

HTML  
 XML 3

File Handling:  Automatically open report when done

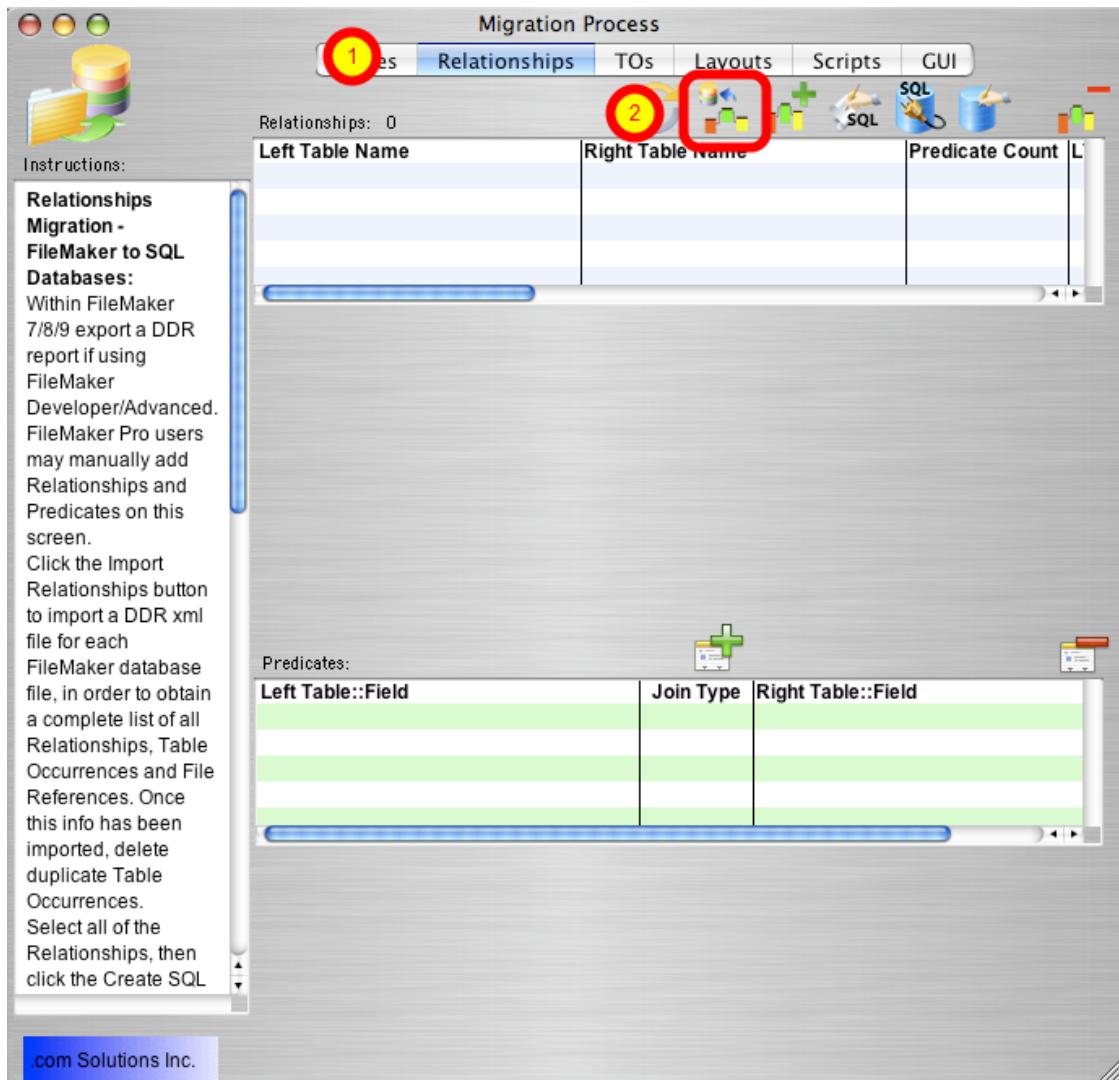
Cancel 4 Create

Within the DDR Export dialog, make sure that the database file is (1) checked for export, (2) along with all objects, (3) XML instead of HTML report format, then click the (4) Create button.



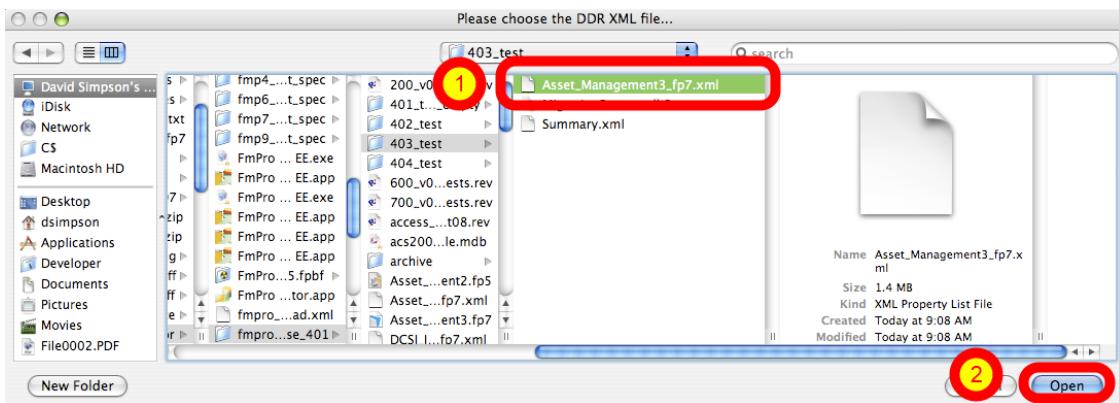
Select the output directory, then click the Save button.

## Migrating Relationships - Import DDR XML File



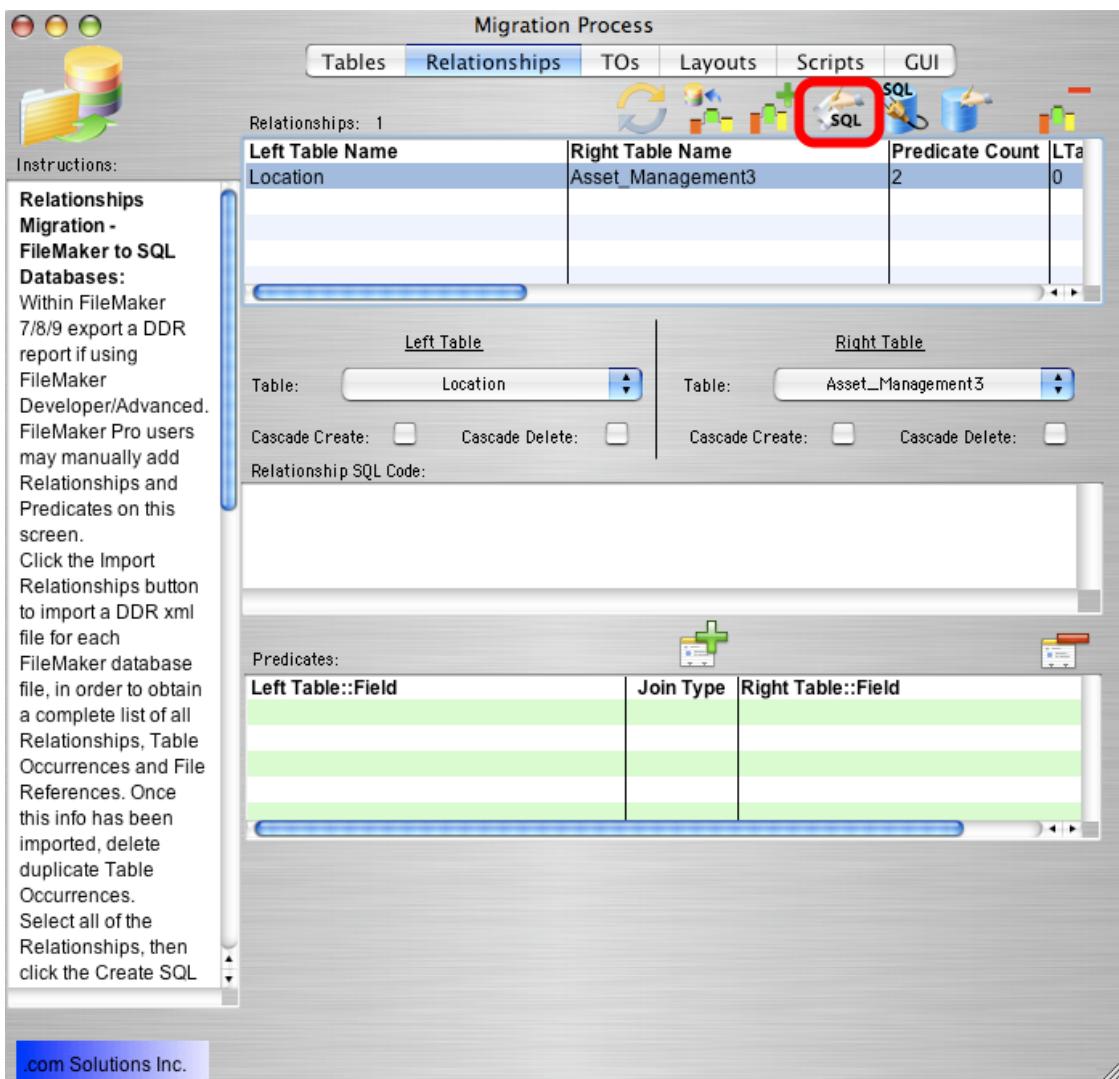
Within FmPro Migrator, (1) click on the Relationships tab, then (2) click on the Import Relationships button.

Note: Relationships will only be imported correctly if all of the base tables have already been created within the Tables tab.



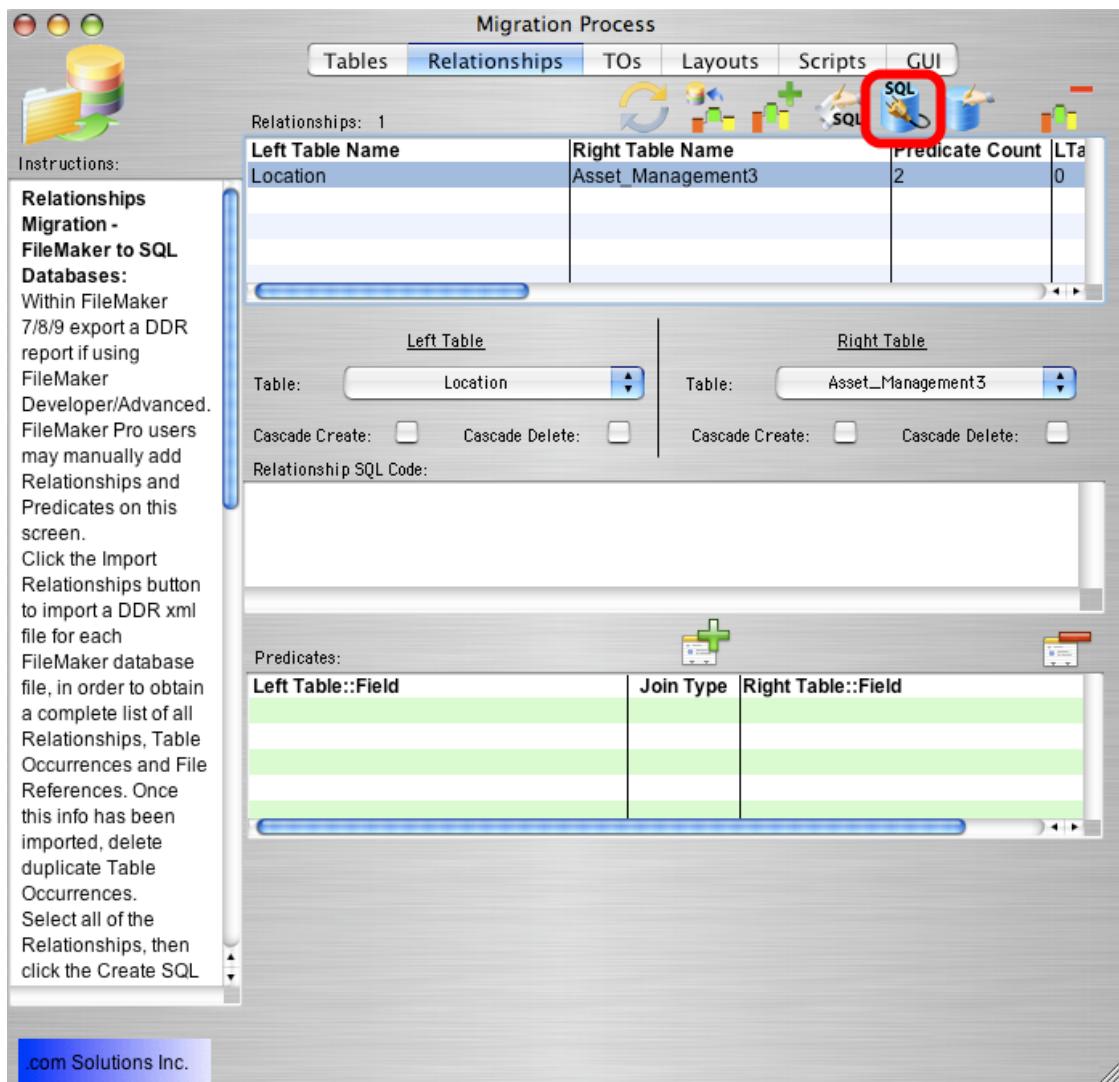
(1) Select the exported DDR XML file, then (2) click the Open button.  
Don't select the Summary.xml file.

### Migrating Relationships - Generate Relationship SQL Code



Select one ore more relationships from the list, click on the Generate Relationship SQL button.

## Migrating Relationships - Create Relationships in MySQL



Click the Transfer Relationships button to create the selected relationships in the MySQL database.

Note: To create relationships for MySQL tables, each table needs to be configured as an InnoDB table type instead of MyISAM and the MySQL database server needs to be at version 5 or higher.