

# FileMaker Pro to Rev Conversion Procedure



.com Solutions Inc.

1.1	FileMaker Pro to Revolution Conversion Procedure	4
1.2	Gathering Info from FileMaker Pro	8
1.3	Converting to a Rev Stack	12
1.4	Customizing the Rev Stack Conversion Process	23
1.5	FmPro Custom Property Set	26



## FileMaker Pro to Revolution Conversion Procedure

---

This document explains the process of converting Microsoft Access and FileMaker Pro® databases into Rev ([www.runrev.com](http://www.runrev.com)) stacks with FmPro Migrator Platinum Edition. FmPro Migrator Platinum Edition converts each FileMaker layout into a Rev card, containing all of the fields, portals, text, images and buttons from the original layout.

FmPro Migrator Platinum Edition generates a full-featured database application within a few seconds, including hundreds of lines of revTalk code within each card of the generated stack file. Individual objects including Tab Panels, radio button groups, checkbox button groups and Data Grid objects also include the revTalk code required for an easy to use database front-end application.

This feature leverages the automated layout and relationship importing features of FmPro Migrator Platinum Edition, while using FileMaker Pro Advanced, Rev Studio, Rev Enterprise, or revMobile along with SQL Yoga from Blue Mango Learning.

Rev stack files generated by FmPro Migrator Platinum Edition incorporate the following commonly used database application features:

**Display Records** - When the stack is opened and a connection is made to the converted SQL database, the records associated with the converted Layout/Form are displayed in a form viewing mode. The fields, text labels, embedded graphics and images are displayed in the same manner as the original database file. Record navigation is implemented using 4 controls:

Next Button, Previous Button, Scrollbar and Record Number field entry. SQL database BLOB column fields containing images are automatically displayed.

Related records are automatically displayed within individual fields and portals/subforms are converted into Rev Data Grids. The related records displayed within Data Grids are automatically updated when advancing to another record in the parent table.

**Update Records** - Clicking into any field provides the user with the ability to update the contents of the field. Date fields are configured with a Date Picker control. Custom Value Lists from the original database are displayed as drop down menus, pop-up menus, radio button group or checkbox group depending upon the original field definition.

Related records displayed within Data Grids can be updated just as easily as records in the main form.

**Insert Records** - Individual records can be inserted as a new record in either Browse or Find operating modes. Clicking the Insert Portal Row button to the top right of any Data Grid object will add a new record to the related database table. As with the original database implementation, the primary key for the parent table will be automatically inserted into the foreign key column of the related record.

**Delete Records** - Individual records can be deleted in either Browse or Find modes. If a Delete Row button or image object was embedded in the original FileMaker Portal, then this same functionality will

also be implemented to delete the selected Data Grid row.

Query Records - Query by example record searching is automatically implemented for each card in the stack file. Once a set of records has been found as a result of the query, the navigation buttons can be used to navigate thru the found set of records. Individual records can be inserted or deleted in found set mode as well as within the standard Browse mode.

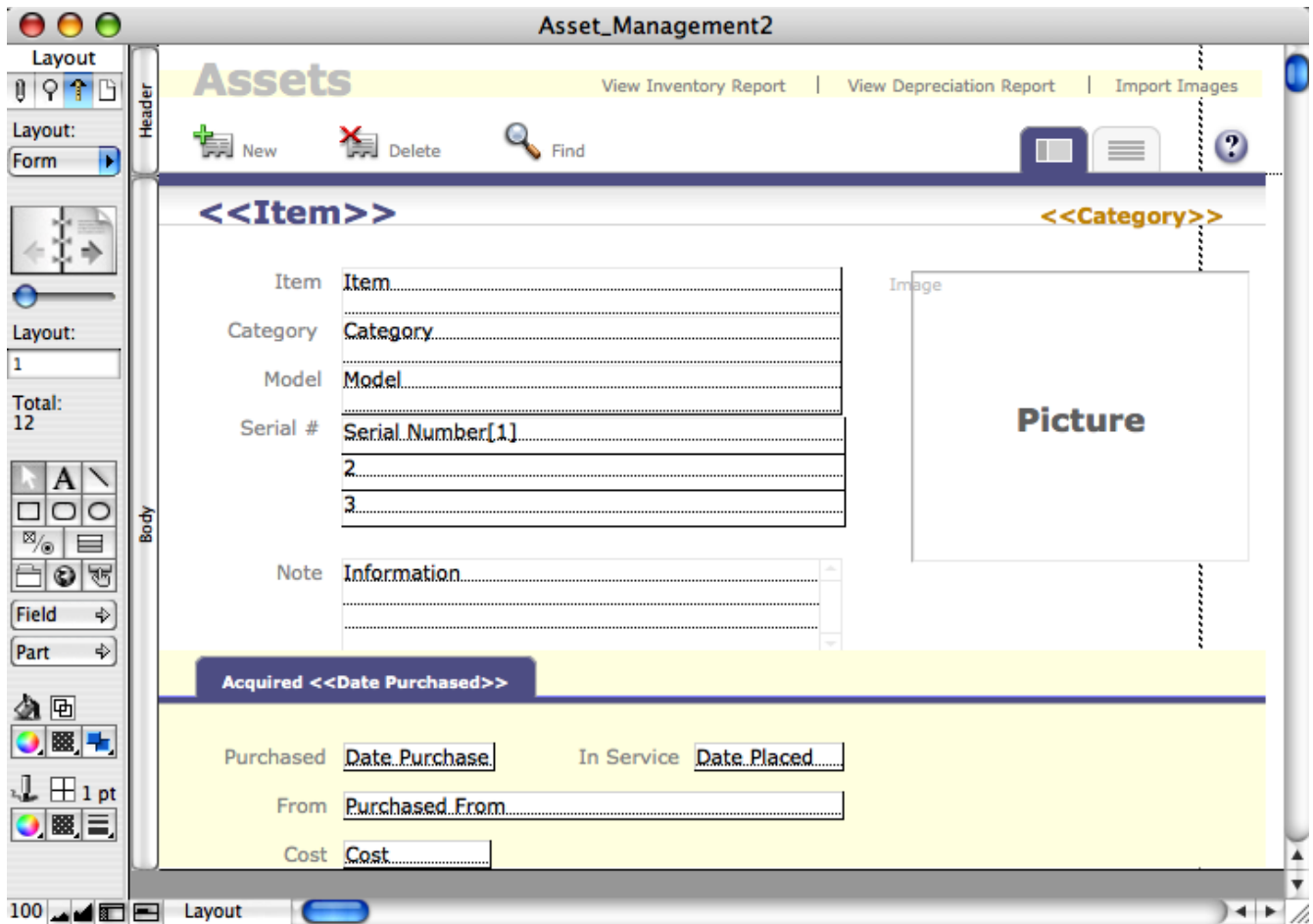
Additionally, this document provides information about the FmPro custom property set included in each card of the generated stack file.

Document Version 03

FmPro Migrator Platinum Edition 5.57

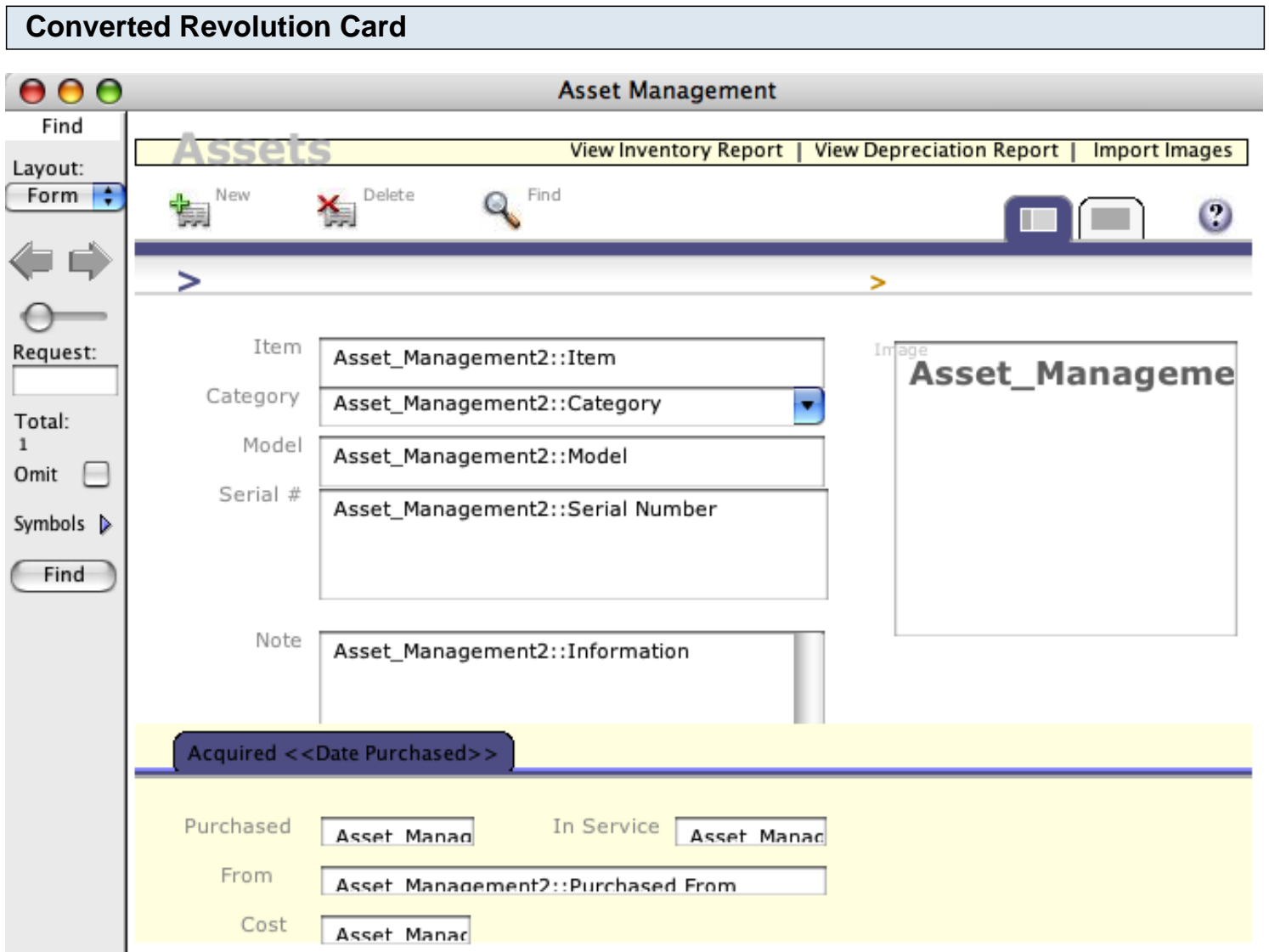
3/24/2010

### Original FileMaker Pro Layout



FmPro Migrator Developer Edition reads the XML definition of a FileMaker layout, and then converts this information into a Revolution card containing all of the fields, portals, text, images and buttons from the original layout. An original layout for the sample Asset Management database is shown in the

previous image.



The resulting Revolution card is shown in this screenshot. Each field is filled with the name of the original Tablename::Fieldname information from the original FileMaker Pro Layout.

A Status Area background object is provided at the left hand side of the card, to simulate the Browse/Find modes from the original database. Menus and scripts are provided in the template stack file for hiding/showing the status area and moving the card objects to accommodate the Status Area.

Each original layout name is used as the name for the Revolution card, and when the Layout drop down menu is selected, a "go to card" instruction is executed, taking the user to the selected card.

The card objects are placed into a Geometry Manager enabled group named "Layout\_Objects\_Group". As the card is resized, the Layout\_Objects\_Group is also resized. If the window is too small to show all of the objects, scrollbars automatically appear to enable scrolling by the user.

## Converted ScriptMaker Script Code

```
on New_Call_Log_Record
# This script creates a new Call Log record and then selects the Client field.
# Access - Script runs with Full Access to allow user-level execution of script.
# 2/28/2006 - David Simpson - .com Solutions Inc. - www.fmpromigrator.com - Initial Release
#
#
#New Record/Request
#Go to Field [ Client_Call_Log::Client_ID ]
# [ Select/perform ]
end New_Call_Log_Record

on New_Call_Log_Window
# This script opens a new Call Log window.
# 2/28/2006 - David Simpson - .com Solutions Inc. - www.fmpromigrator.com - Initial Release
#
#
lock screen
#New Window [ Name: "Call Log"; Height: 300; Width: 700; Top: 50; Left: 50 ]
go to card "Client Call Log"
unlock screen
end New_Call_Log_Window
```

If ScriptMaker scripts have been copied into FmPro Migrator Platinum Edition via the ClipBoard, then these scripts will be converted into Rev handlers. The 34 most commonly used (out of 140) ScriptMaker steps are converted directly into revTalk code. ScriptMaker commands such as Go to Layout are converted into the revTalk "go to card" command, the Freeze Window/Refresh Window commands are converted into "lock screen/unlock screen" commands. ScriptMaker commands which fill variables and fields are converted into the revTalk "put" command.

All of these converted scripts will require additional development after the conversion process has been completed by FmPro Migrator Platinum Edition, as these conversions are primarily intended to reduce the amount of typing required by the Rev developer.

# Gathering Info from FileMaker Pro

## Step 1 - Perform Table/Data Migration

The screenshot shows the 'Migration Process' window with several components:

- Instructions:**
  - Step 2:** Install the FileMaker ODBC driver from the FileMaker installer CD.
  - Create an ODBC DSN matching the DSN name listed in the Table Details window.
  - Turn on ODBC Sharing within FileMaker. Menu: File > Sharing > ODBC/JDBC
  - Click the Get FieldSize icon.
- Troubleshooting:** (Empty section)
- Tables:** 33. A table list with columns: Original Table, New Table, Source, Destination, Fields, Records. The 'Asset\_Management3' table is selected.
- Fields:** 44. A field list with columns: ID, Original Fieldname, PK, Auto-Increment, Next SN, New Fieldname. Fields include Asset ID (PK), Model, Item, Category, and Cost.
- Progress Steps:**
  - Step 2: Get FieldSize (Not Started)
  - Step 3: Create Table (Not Started)
  - Step 4: Transfer Data (Not Started)
  - Optional: Image Export
  - Step 5: Convert Data (Repeating Fields) (Not Started)
  - Step 6: Create Table (Repeating Fields) (Not Started)
  - Step 7: Transfer Data (Repeating Fields) (Not Started)

Use the instructions on the [www.fmpromigrator.com](http://www.fmpromigrator.com) Support web pages for the type of migration you want to perform. This process involves gathering metadata from the FileMaker database, creating the tables in the destination database and transferring data into the destination database.

## Step 2 - Copy Layouts Individually

**Migration Process**

Tables Relationships TOs **Layouts** Scripts GUI

Instructions: Layouts: 73

**Layout Migration:**  
[Only used for Table Consolidation Migration Projects.] Within FileMaker 7/8/9 (layout mode), select all objects on a layout then select Copy from the Edit menu. Click the Add Layout button on this screen to copy the layout definition from the clipboard into FmPro Migrator. Each layout can be copied individually in this manner, or all layouts can be imported in batch mode (see info below). Each layout is given a default name in the form: Layoutxx where xx is an incrementing number. The currently selected layout or portal name can be changed using the field above

Layout Name	Size	Fields	Portals
Address Labels	145,80	2	0
Address_Book	1081,719	60	0
Binder- new business	683,659	9	0
Bord Report	763,278	21	0
Claims Overview	672,165	11	0

Portals:

Portal Name	Fill Method	Field Titles

.com Solutions Inc.

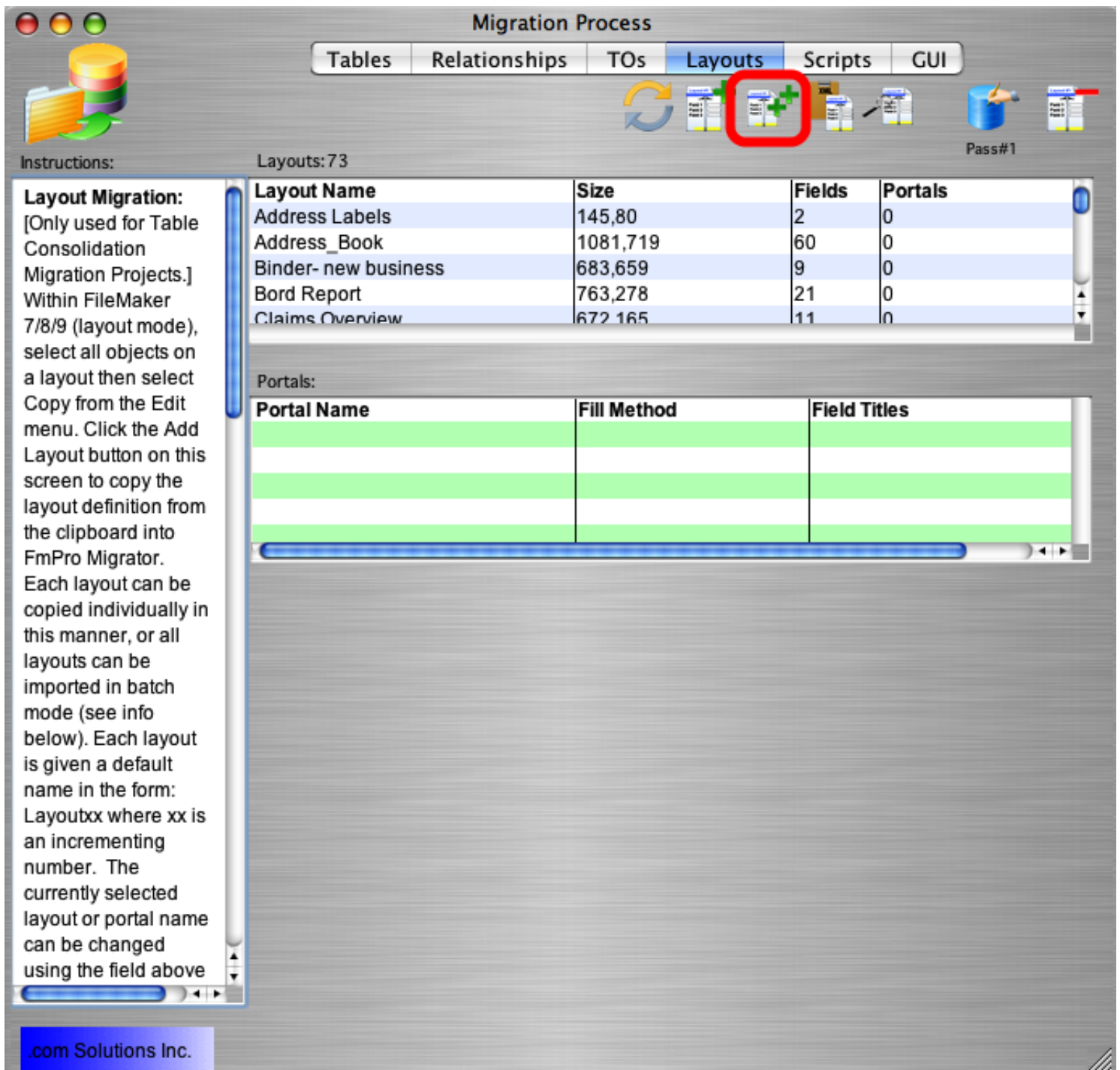
Click the Add Layout from Clipboard button, to import layouts one at a time from FileMaker Pro into FmPro Migrator Platinum Edition.

Select all of the objects on a Layout, then copy the layout onto the Clipboard within FileMaker Pro. Once the Layout has been copied, click the Add Layout from Clipboard button on the Layout tab within FmPro Migrator Platinum Edition.

Once the Layout has been added to the list, give the Layout a name, matching its name in the original FileMaker Pro database. This individual layout capture process is best suited for importing small numbers of layouts from the original FileMaker Pro database.

**Note:** Layout and ScriptMaker names should not include Unicode characters, as these names will become the names of cards and handlers within the Revolution stack.

## Step 2 - Copy Layouts in Batch Mode



**Migration Process**

Tables Relationships TOs **Layouts** Scripts GUI

Instructions: Layouts: 73

**Layout Migration:**  
[Only used for Table Consolidation Migration Projects.]  
Within FileMaker 7/8/9 (layout mode), select all objects on a layout then select Copy from the Edit menu. Click the Add Layout button on this screen to copy the layout definition from the clipboard into FmPro Migrator. Each layout can be copied individually in this manner, or all layouts can be imported in batch mode (see info below). Each layout is given a default name in the form: Layoutxx where xx is an incrementing number. The currently selected layout or portal name can be changed using the field above

Layout Name	Size	Fields	Portals
Address Labels	145,80	2	0
Address_Book	1081,719	60	0
Binder- new business	683,659	9	0
Bord Report	763,278	21	0
Claims Overview	672,165	11	0

Portals:

Portal Name	Fill Method	Field Titles

.com Solutions Inc.

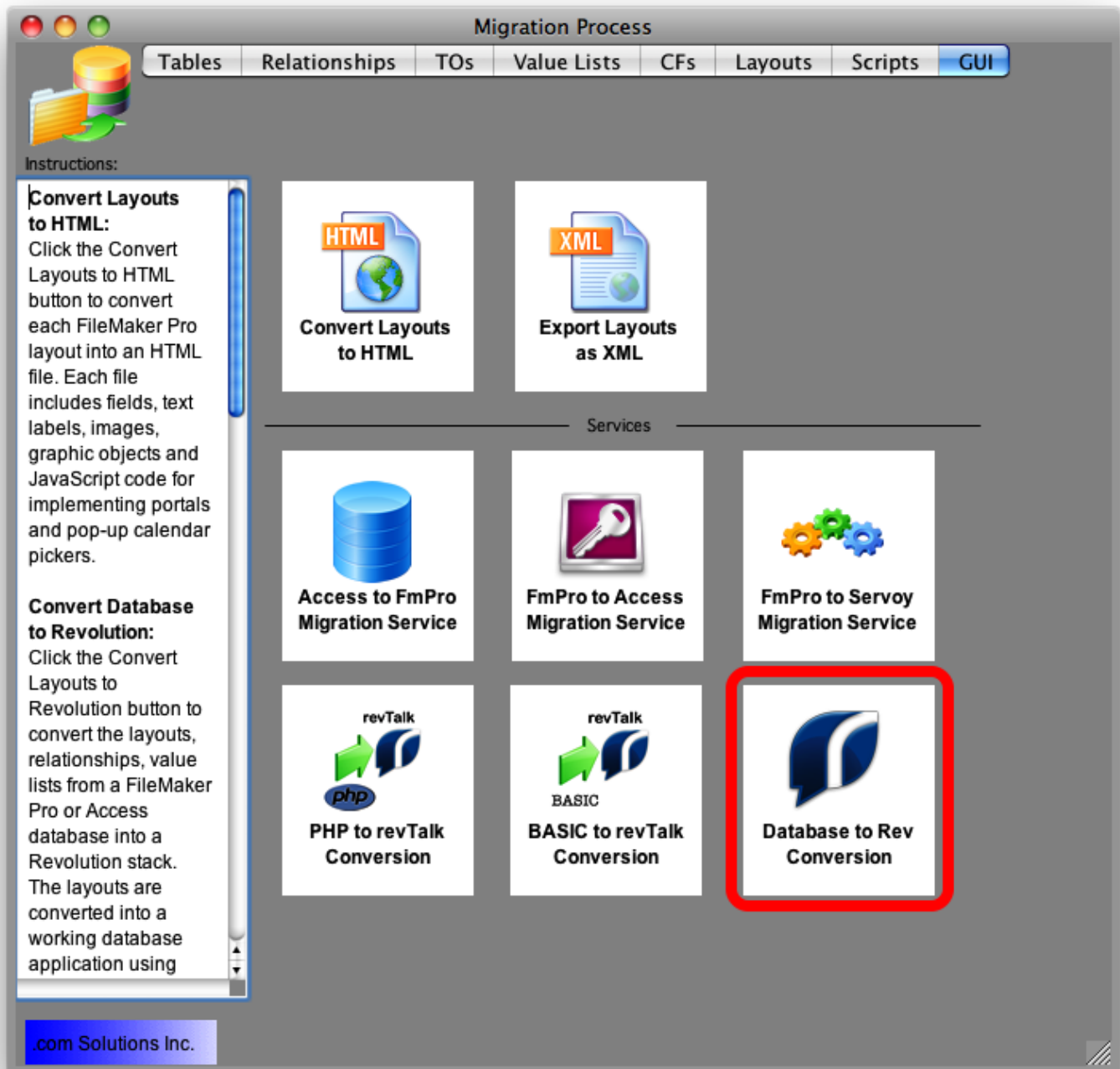
Click the Import All Layouts button to import all of the layouts from the currently open FileMaker Pro database file. This batch capture process utilizes QuickKeys/WinAutomation shortcuts to control the FileMaker Pro database while capturing the layouts into FmPro Migrator Platinum Edition. This process involves the installation and setup of either QuickKeys or WinAutomation software on your computer

(depending upon operating system in use), and the use of FileMaker Pro Advanced to create a DDR XML file.

When this automated procedure is used, each layout will be imported into FmPro Migrator Platinum Edition and the layout will automatically be given the correct name as it was read from the DDR XML file exported by FileMaker Pro Advanced.

## Converting to a Rev Stack

### Step 3 - Convert Database to Rev



Click the Database to Rev Conversion button to open the Database to Rev Conversion window.

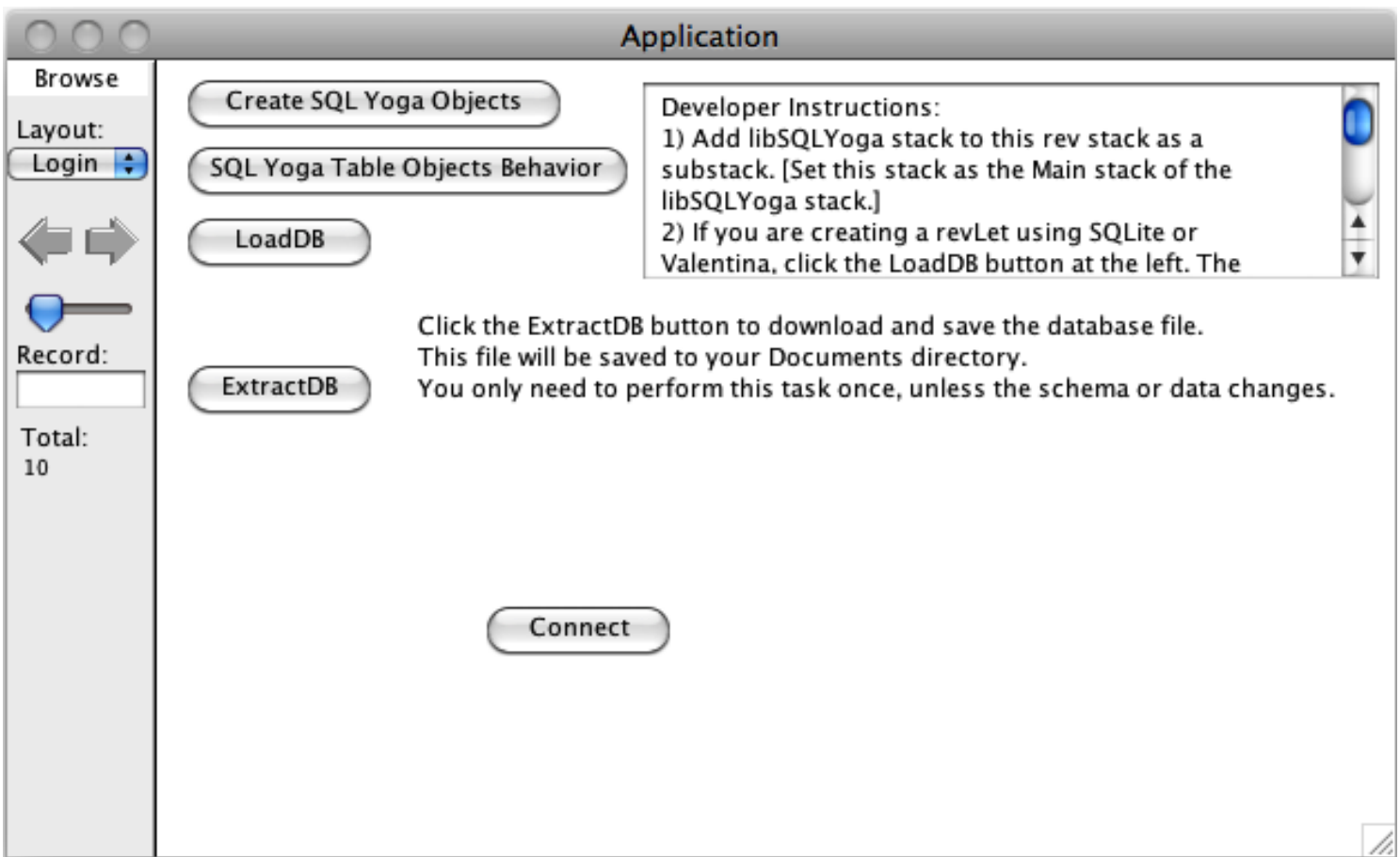
## Database to Rev Conversion Window

Some of the features available in the Database to Rev Migration window include:

- (1) The Layouts Qty label displays the number of layouts which will be converted. This number represents the number of layouts which have been captured and stored in the FmPro Migrator project file.
- (2) Output file types include: Application, revLet, revMobile. The Output File Type menu provides options for generating a stack file to be used as a desktop application, revLet or revMobile app. Selecting the revLet option generates a stack having additional buttons visible for the user and Rev developer to embed or extract a SQLite or Valentina database file from the stack - if one of these database files has been selected for the migration.

- (3) Processing Type menu: Licensed or Demo. In demo mode up to 5 layouts and scripts can be processed.
- (4) Copy License Key from Clipboard - Pastes in the license key, which will be validated and displayed in the field to the right of the clipboard icon.
- (5) License Key validation message text - This text displays the License Key expiration date and maximum quantity of layouts which can be processed with this license key.
- (6) Processing Status - Lists the number of layouts, scripts and elapsed time for the processing.
- (7) Online Quote Link - Produces a printable quote form for ordering license keys.
- (8) Export Template Stack button - Exports the template stack into a file named Application.rev in the output directory (overwrites existing copies).
- (9) Migrate button - Click this button to perform the migration.

**The Generated Stack File**



The first time you open the generated stack file, there are a few tasks which need to be completed to finish the conversion process and prepare for using and deploying the stack as either an application or revLet. Double-click the Application.rev file which was written into the output directory, and it will open within the Rev IDE.

- 1) Add SQL Yoga as a substack to the Application stack.

2) Add your SQL Yoga license key into the SQL Yoga custom properties of the Application.rev stack (if you have one). If you don't have a SQL Yoga license key, don't worry, you can still use SQL Yoga in demo mode with up to 10 database records, and periodic reminder messages every 10 minutes.

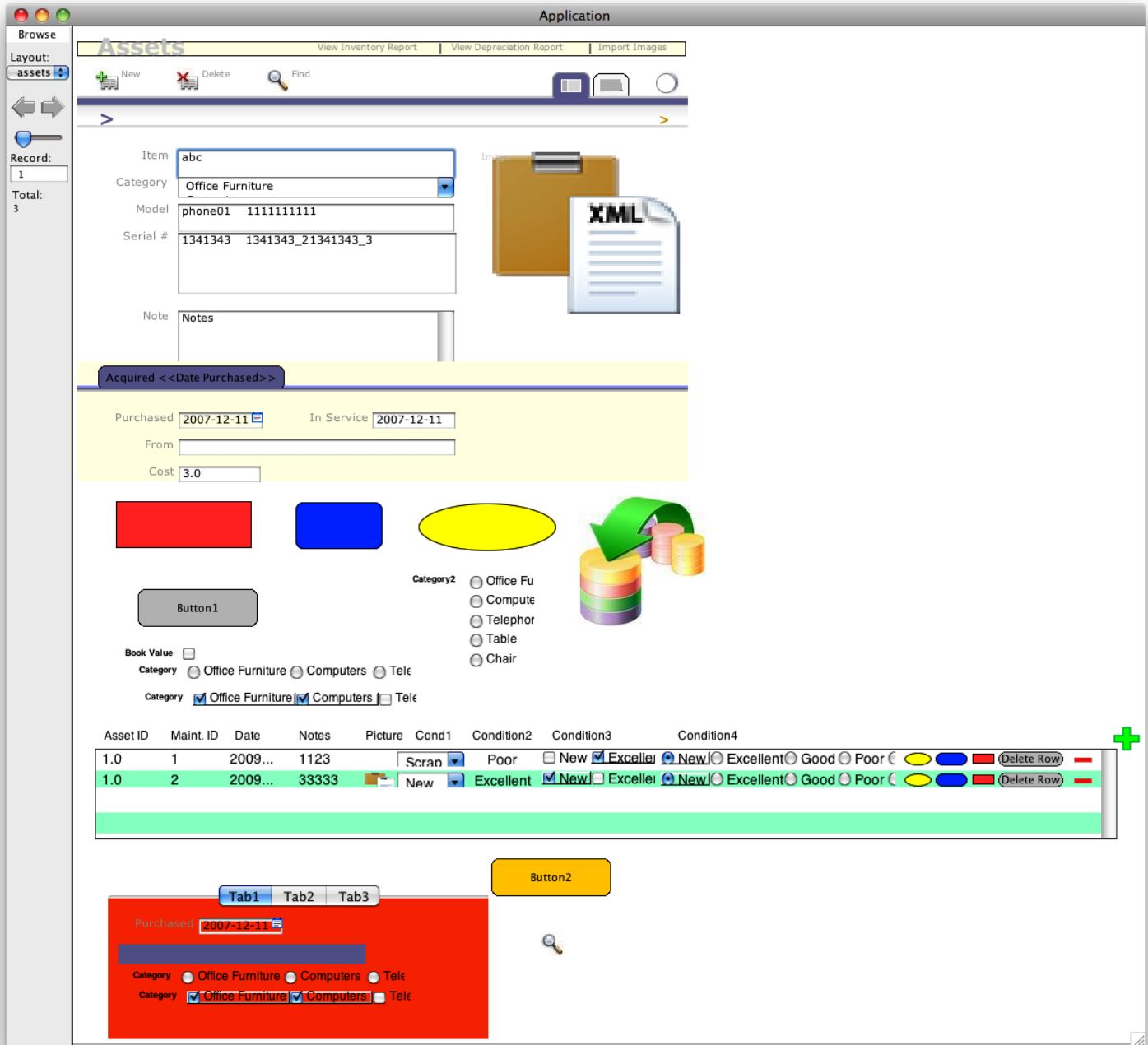
3) If you are creating a revLet and you want to include a SQLite or Valentina database to be used locally with the revLet, you should click the LoadDB button, which will load the embedded database into a custom property within the stack file. Once this button has been clicked, the button is hidden, because the end-user doesn't need to see it. The LoadDB and ExtractDB buttons are only visible when generating a stack which will be used as a revLet. And you don't even have to use them at that time, because your revLet could connect directly to a SQL database thru a network connection.

In addition to converting the FileMaker layouts, FmPro Migrator Platinum Edition also converts each FileMaker ScriptMaker script into revTalk scripts. These scripts will be placed into two output files will be written to the output directory:

FmPro Original Scripts.txt file - This is a text version of the original ScriptMaker scripts, and is made available for documentation purposes so that Revolution developers can review the original unconverted scripts.

FmPro Converted Scripts.txt - This text file contains the converted Revolution code. Any instructions which could not be converted will be commented and the original comments will be retained within the scripts.

## Creating SQL Yoga Objects



The first time the stack is opened within the Rev IDE, click the Create SQL Yoga Objects button. Then click the Connect button.

You should then see the first converted layout, with data filling the fields. Clicking the navigation buttons should take you to the next record, and related records should be displayed within the Data Grid.

## Supported FileMaker Pro Layout Object Types

The following FileMaker Pro Layout object types are currently supported by the conversion process:

- Fields
- Portals
- Text
- Lines
- Rectangles
- Rounded Rectangles
- Ovals/Circles
- Grouped Objects
- Images
- Buttons
- Tab Controls
- WebView

Each FileMaker Pro layout object is re-created as an equivalent Rev card object, using the formatting and style attributes of the original object.

Stylized FileMaker Pro layout text objects are converted into Rev text labels, having the embedded text styles of the original object defined with the HTMLText property.

The supported field display options include Check Box Set, Radio Button Set, Pop-Up Menu, Drop Down List and Calendar. Pop-Up menus and Calendar objects are created with embedded Rev script code to automatically populate the underlying field.

## Unsupported FileMaker Pro Layout Object Types

The following FileMaker Pro Layout object types are not currently supported by the conversion process:

**Conditional Formatting** - Conditional formatting can be used to automatically resize fields and change the appearance or color of field data based upon a calculation formula. The automated resizing or movement of objects can be done by using Revolution's Geometry Manager. Within Revolution, the movement of objects can even be made dependent upon the movement of other nearby objects, offering more control of objects than is available within FileMaker Pro.

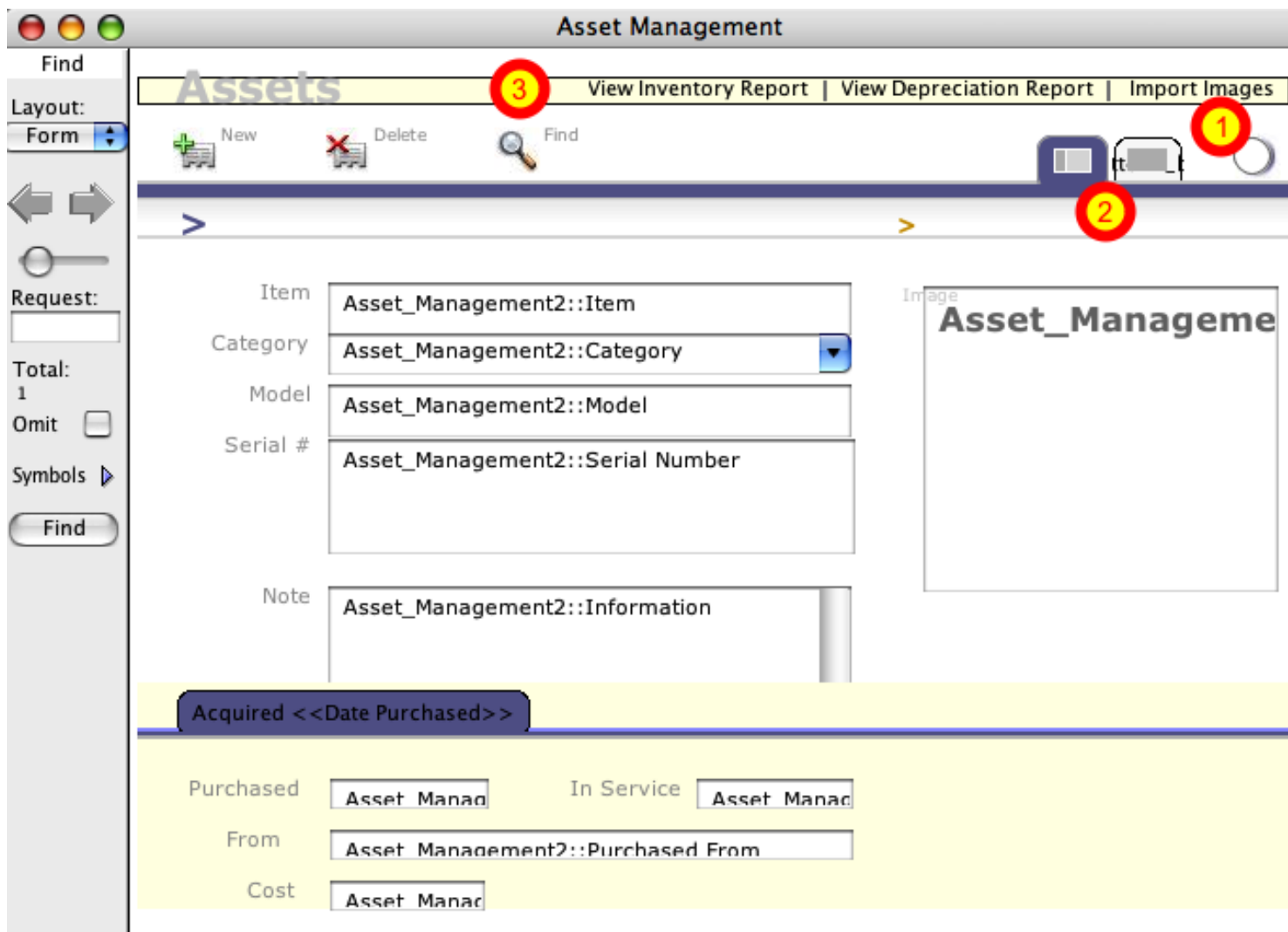
**Dynamic Value Lists** - Value Lists within FileMaker Pro can be implemented as either static or dynamic value lists. Static value lists are similar to pull down menus in Revolution populated with a static list of values. FmPro Migrator imports static value lists from the DDR XML file and assigns the values to each appropriate object on the new Rev card. Dynamic value lists can be implemented in Revolution by running a handler to run a SQL query prior to displaying a menu.

**Merge Fields** - Merge fields include static text and dynamically gathered information from fields, which gets merged together and displayed together on the layout. A similar feature could be implemented with custom properties and setProp handlers within a Rev text label (field) object.

**Chart Object** - This is a new FileMaker Pro 11 object. With extra development, a similar feature could be implemented within a Rev stack by using chartsEngine from derbrill Multimedia.

**Embedded Page#, Date, Record# Layout objects** - These objects should be manually replaced with some other object like a text label holding the data to be displayed. Plus the addition of a little bit of revTalk code to display the proper Page#, Date etc. However the record# is presently displayed within the record number field in the status area, under the record navigation slider.

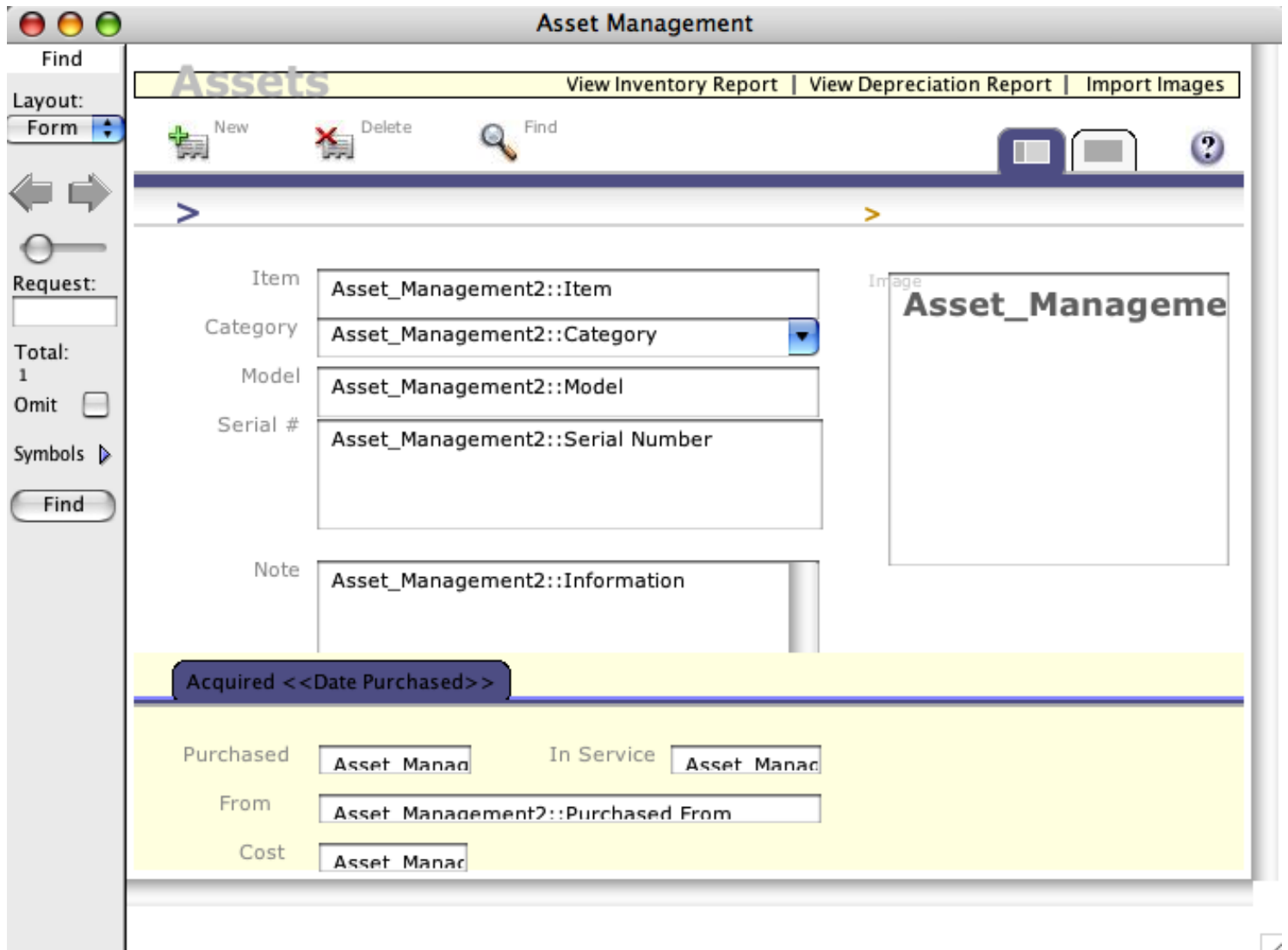
### Cosmetic Changes Required - Example #1



In general, each created object within the Rev stack will appear very similar to its counterpart in the original FileMaker Pro Layout. However, some changes may be required within the generated stack file. This image of a converted Revolution stack file shows 3 changes which should be made:

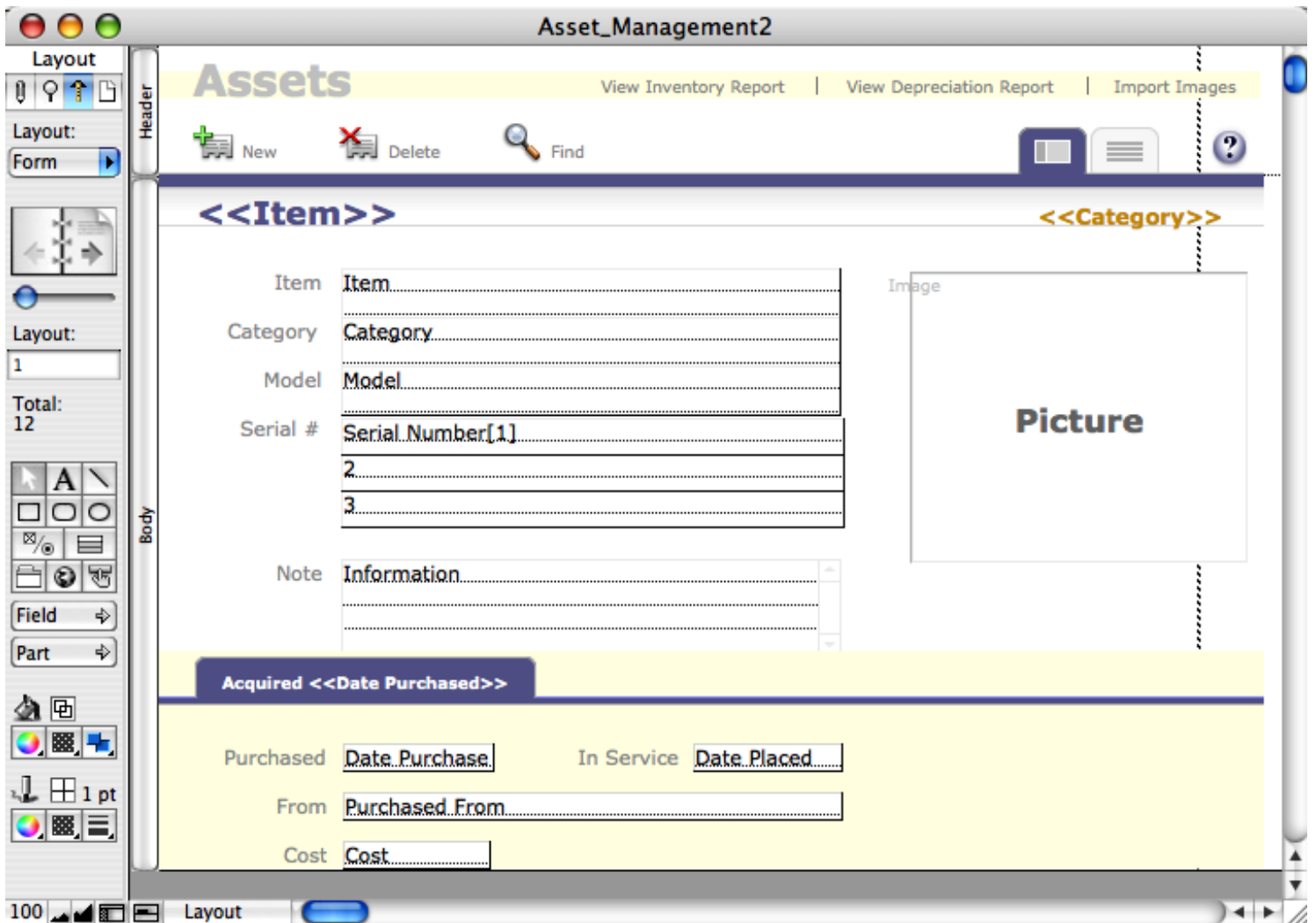
- 1) The Help button icon has been created, but the (?) image is sitting under the shadowed circle object. FmPro Migrator Platinum Edition builds each layout object in the order it is found within the Layout XML file. Therefore some objects may be created in the wrong order for display purposes. The solution is to change the layer of each object so that they display properly. The shadowed circle object was also moved left by 1 pixel.
- 2) There is a button sitting under the graphic of the folder icon, and this button has its showName property enabled. Setting this property to false makes the button display more cleanly on the card.
- 3) The text of the 3 text buttons across the top of the card are displayed a little bit too high.

### Cosmetic Changes Applied - Example #1



This image shows the stack after the 3 cosmetic changes were made.

## Original FileMaker Pro Layout



And for comparison purposes, here is an image of the original FileMaker Pro Layout.

## Cosmetic Changes Required - Example #2

The screenshot shows a FileMaker Pro application window titled "Application". The main content area is a "Detail Form" for a client. The form is divided into several sections:

- Client Information:** Fields for Client ID, First Name, Last Name, Stable Name, Address #1, Address #2, City, State, Zip, Home Phone#, Office Phone#, Fax#, Cell #, Email, Email2, Email3, Full Name, and Address.
- Account Information:** Fields for Inactive (checkbox), By (Client::Inactive), Creation Date (Client::Creation), and Account Rep (Client::Account Rep mer).
- Horses Owned (Active):** A section titled "Horses Owned (Active)" with a sub-section "Horses Owned By This Client:" containing a table with columns "ID" and "Name".
- Horses Owned (Inactive):** A section titled "Horses Owned (Inactive)" with a sub-section "Inactive Horses Owned By This Client:" containing a table with columns "ID" and "Name".

Five red circles with numbers 1 through 5 highlight specific areas:

- 1) A field labeled "GlobalTable::dbMaintenanceMessage" overlapping a text label.
- 2) A horizontal line with a red background.
- 3) Two text label buttons wrapped within a field.
- 4) A horizontal line with a green background.
- 5) Another horizontal line with a green background.

Here is another converted Layout stack file which needs different cosmetic changes:

- 1) FmPro Migrator Platinum Edition creates each Rev field as an opaque field, in order to insure that fields having a background color are displayed properly. Text objects however, are created with their opaque property set to false, allowing other objects to show thru. But the dbMaintenanceMessage object shown on this layout is not a Text object, it is a field object, so it has been created to be opaque - thus hiding the text label which it overlaps. Therefore in this location, the dbMaintenanceMessage field should have its opaque property set to false.
- 2) This horizontal line is supposed to be colored green, but the correct color was not properly located within the original XML code. This same problem also occurs with the objects marked (4) and (5). However the remaining horizontal line objects on this same layout (not shown) did retain their correct colors.
- 3) These two text label buttons ended up wrapping within the field. Changing the text margin from 2 to 4 and widening the fields solved these issues.

## Cosmetic Changes Applied - Example #2

Application

GlobalTable::dbMaintenanceMessage

**Client Record - Detail Form**

GlobalTable::nDocServerError

Client ID: Client::Client

First Name: Client::First Name

Last Name: Client::Last Name

Stable Name: Client::Stable Name

Address #1: Client::Address #1

Address #2: Client::Address #2

City: Client::City

State: Client::State

Zip: Client::Zip

Home Phone#: Client::Home Phone Number

Office Phone#: Client::Office Phone Number

Fax#: Client::Fax Number

Cell #: Client::Cell Phone Number

Glott Email: Client::Email, Client::Email De

Email2: Client::Email2, Client::Email2 I

Email3: Client::Email3, Client::Email3 I

Full Name: Client::Full Name

Address: Client::Address

Inactive  By Client::Inactiv

Creation Date: Client::Creation

Account Re: Client::Account Rep\_mer

Date: Client::Inactive Date

Balance: Client::Client B

Horses Owned (Active)

Horses Owned By This Client:

ID	Name	TSI (FM)

Horses Owned (Inactive)

Inactive Horses Owned By This Client:

ID	Name	TSI (FM)

This image shows the stack after the 5 cosmetic changes were made.

# Customizing the Rev Stack Conversion Process

---

## FmPro\_Migrator\_Preferences.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<Preferences>
<DBToRev type="DBToRev">
<StackTemplate filepath="" />
<SQLYogaFilename filepath="/Applications/Revolution Enterprise/Third Party/sql_yoga/sql_yoga.rev"/>
<SQLYogaLicenseKey Username="EnterUserName" SQLYoga_License="EnterLicenseKey"/>
<ValentinaLicenseKey LinuxLicense="EnterLicenseKey" MacLicense="EnterLicenseKey" WinLicense="EnterLicenseKey" />
</DBToRev>
</Preferences>
```

FmPro Migrator utilizes an optional XML preferences file which can be used to customize the DB To Rev conversion process. This XML template file can be placed into the FmPro Migrator directory within the Documents (or My Documents on Windows) directory to customize the generated stack file. These customizations can include the automated addition of License Keys for SQL Yoga and Valentina database libraries so that you don't have to manually enter these keys within each stack you create.

### Using a Different StackTemplate File

The StackTemplate option enables Rev developers to extract and customize the standard template stack embedded within FmPro Migrator. Click on the Export Template Stack button on the Database to Rev window to extract the existing template stack file to the output directory (it will overwrite any existing file). You should move or rename this file and update the XML preferences file to point to the modified version of the file.

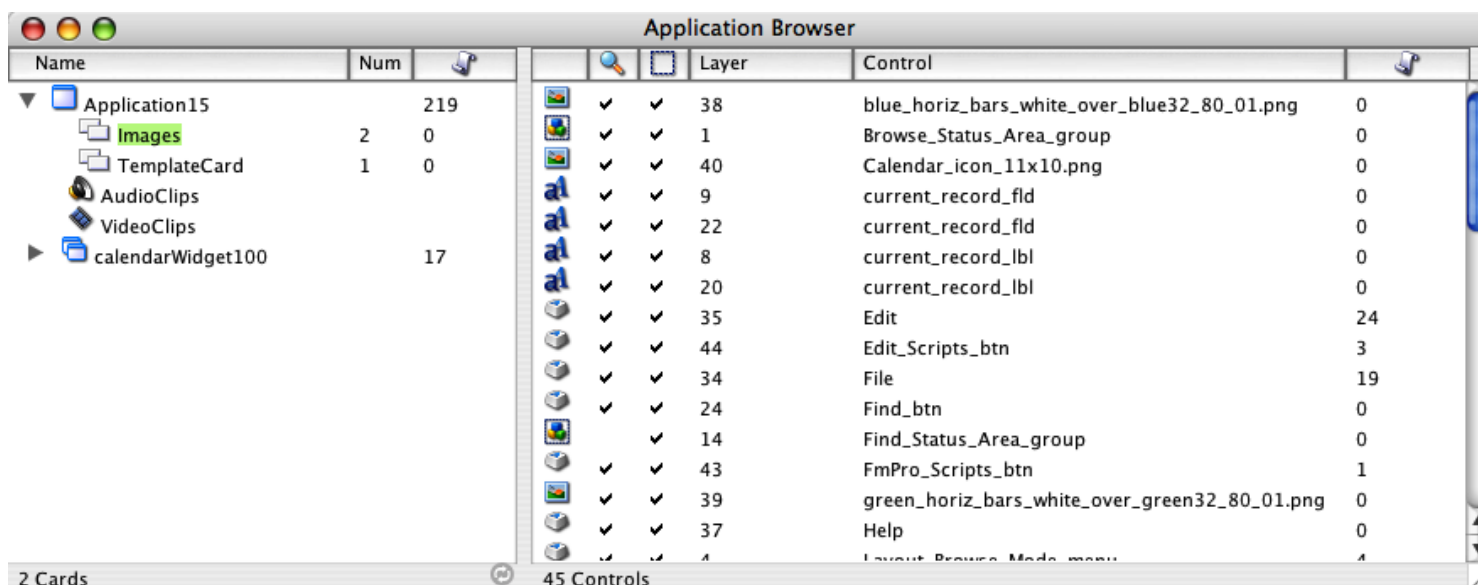
The custom version of your template stack file will be copied into the output directory, renamed as Application.rev and used as the destination stack where all of the converted layout objects will be created.

This means that you can add that one line command from article #3 into the showAllRecords handler in the card stack, and that correction along with any other changes you make will automatically be used as your new template stack. You might also want to change the look of the status area. You could create a new status area across the top of the card, to simulate the look of newer versions of FileMaker Pro. Or you might turn the status area into a floating palette.

### Embedding the SQL Yoga Library

The SQLYogaFilename option provides a place to specify the location of the SQL Yoga library to FmPro Migrator so that SQL Yoga can be added to your generated stack automatically. Otherwise, you will need to manually add SQL Yoga to your stack each time you perform a conversion.

## The Template Stack



FmPro Migrator Platinum Edition uses an internally-stored template file for the Rev stack which it generates from the FileMaker Pro Layouts. This file is written to the output directory as a file named Application.rev at the start of the conversion process.

Rev developers can supply their own stack in place of the Application.rev file, in order to make use of their own customized handlers and graphics. FmPro Migrator Platinum Edition always deletes and overwrites any existing Application.rev stack file.

A replacement template stack can be specified in the FmPro\_Migrator\_Preferences.xml file.

A replacement template stack should have the following features, in order to be compatible with FmPro Migrator Platinum Edition:

**setLayoutList** - stack-level handler - This handler puts the list of cards into the Layout list menus of the Browse and Find groups. If you don't want to have this feature, you can simply include a handler having this same name but containing no executable code.

**displayCalendar** - stack-level function - This handler displays the calendar pop-up menu within fields which have this feature configured within the original FileMaker Pro Layout. This function returns empty or the date picked by the user.

**browserInit**

**browserGo**

**browserEnsure**

**browserFinalize**

**browserBack**

**browserForward**

**browserRefresh****browserStop**

These stack-level handlers control one or more revBrowser controls on a card. Each handler takes a parameter (pBrowserNum) as the number representing the revBrowser control, and is used to define its name on the card. There could be multiple FileMaker WebViewer objects on a single Layout, therefore each WebViewer will be converted into a separate Revolution revBrowser control having a different name on the card.

**statusAreaWidth** - stack-level customProperty - This custom property indicates the width of the Status Area at the left side of the window. The Layout\_Objects\_Group is automatically moved over to accommodate the width of the Status Area after all of the cards have been created. Set this custom property to 0 if you don't want to have a Status Area within your template stack.

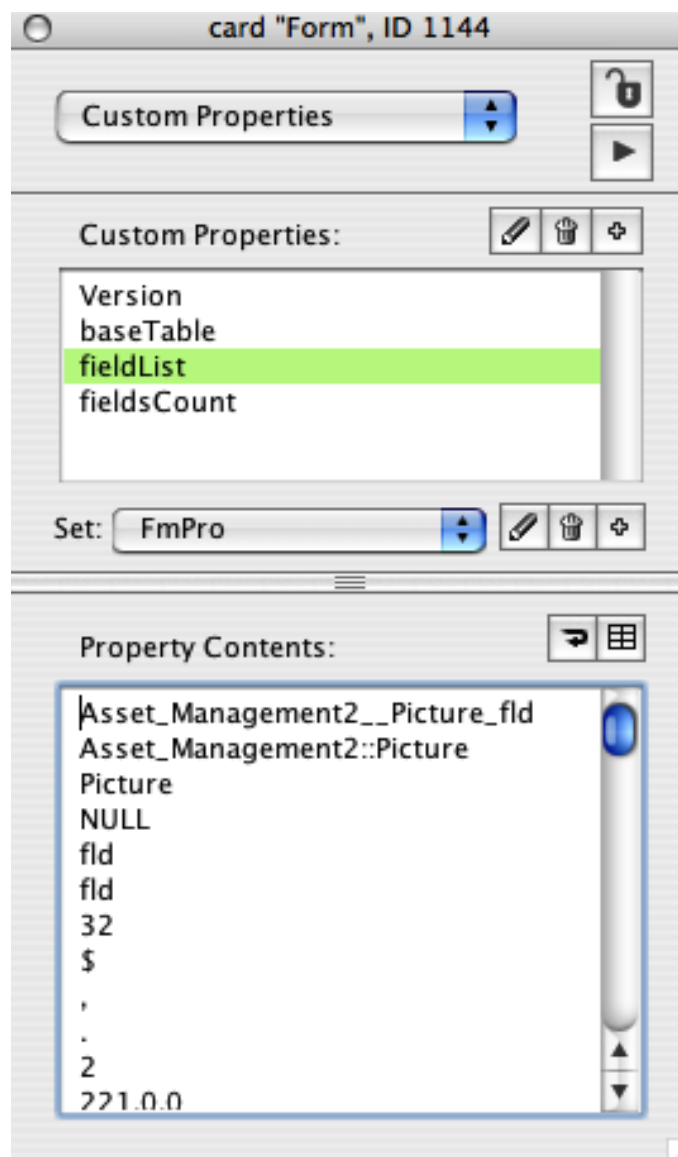
**templateStackName** - stack-level customProperty - This custom property provides the name of the template stack file.

## FmPro Custom Property Set

Each card in the generated Rev stack file includes a custom property set named FmPro. This custom property set includes detailed info about the field objects which have been created on the card by FmPro Migrator Platinum Edition. This information is expected to be useful for Rev developers who want to create data entry/validation and data display handlers for each of the fields on the card. This info is also used by the handlers which clear field data upon entering Find mode and when filling each field with data.

Having the fields listed in a custom property set also means that developers don't have to manually type in each of the field object names into their code either. Just loop thru the lines of data in the custom property set to work with the names and data types for each field on the card.

### FmPro - customPropertySet Overview



There are 4 customProperties within the FmPro customPropertySet.

## Version - customProperty

The Version customProperty defines the version of the customPropertySet schema. The current value for the Version customProperty is 1, this value will change as enhancements are made to the FmPro customPropertySet.

## baseTable - customProperty

The baseTable customProperty represents the name of the base table referenced by the original layout.

## fieldList - customProperty

fieldsList - Custom Property (Columns 1 - 7)	Rev Object Name	Original FmPro Name	SQL Table Name	SQL Yoga Relationship Name	SQL Table Column Name	SQL Table Column Type	Rev Object Type
Example Data -->>	Field1 fld	Client::ID	client	client	ID	DECIMAL	fld
	Field2_menu	Client::Client_Name	client	client	Client_Name	DATE	btn

fieldsList - Custom Property (Columns 8 - 16)	Usage Type	Entry Options Flag	Currency Symbol	Thousands Separator	Decimal Point	Decimal Digits	Negative Color (RGB)	True String	False String
Example Data -->>	fld	0	\$	,	.	2	255, 0, 0	Yes	No

fieldsList - Custom Property (Columns 17 - 22)	Date Separator Char	Date Element Separator1	Date Element Separator2	Date Element Separator3	Time Separator Char	Time AM String
Example Data -->>	/	<<-- space	<<-- space	<<-- space	:	AM

fieldsList - Custom Property (Column 23)	Time PM String
Example Data -->>	PM

The fieldsList is a TAB delimited list providing 23 parameters for each field which has been created on the card. These parameters are shown in the screenshot with sample values and are listed below:

- (1) **Rev Object Name** - The name given to the Revolution field when it was created.
- (2) **Original FileMaker Pro Field Name** - This is the name of the Table Occurrence::Fieldname in the original FileMaker Pro database.
- (3) **SQL Table Name** - The converted name of the original table, as it appears in the SQL database.
- (4) **SQL Yoga Relationship Name** - If the field is in another table, the relationship name to use to get the data from the table.
- (5) **SQL Table Column Name** - The converted name of the SQL database column name.
- (6) **SQL Table Column Type** - The type of data contained in the field, matching the field types for the destination SQL database.
- (7) **Revolution Object Type** - The type of Revolution object. Use this object type and its name when referencing the object.
- (8) **Usage Type** - The way in which the object is used on the card, either as a field or a menu. This parameter indicates whether you need to put data into the field (for a field) or into the label of an object (for a menu).

- (9) **Entry Options Flag** - The entry options for entry of data within a field, as defined by the fieldObj flags XML parameter in the original layout. Multiple options may be applied to a field simultaneously. To determine if a particular value is selected, use the bitAND Revolution instruction. See the following screenshot for more details.
- (10) **Currency Symbol** - The symbol used for currency values.
- (11) **Thousands Separator Symbol** - The symbol used to separate thousands, in numeric values.
- (12) **Decimal Point Symbol** - The symbol used for decimal points in numeric values.
- (13) **Number of Decimal Digits** - The number of values displayed to the right of the decimal point symbol
- (14) **Negative Color (RGB)** - The RGB values, separated by commas which should be used for negative numeric values.
- (15) **True String** - The text string used for true values.
- (16) **False String** - The text string used for false values.
- (17) **Date Separator Char** - The symbol used for separating short date values. (i.e. MM/DD/YYYY).
- (18) **Date Element Separator1** - The first date separator character string, used when displaying dates in long format.
- (19) **Date Element Separator2** - The second date separator character string, used when displaying dates in long format.
- (20) **Date Element Separator3** - The third date separator character string, used when displaying dates in long format.
- (21) **Time Separator Char** - The symbol used for separating time values (i.e. HH:MM:SS)
- (22) **Time AM String** - The suffix string used for displaying AM time values.
- (23) **Time PM String** - The suffix string used for displaying PM time values.

### Entry Options Details - (fieldObj flags values)

fieldObj flags	Result
0	Enter field - Browse & Find Modes - No other Options checked
4	Entry in Find Mode Only - No other options checked.
16	Entry in Browse Mode Only - No other options checked
20	No Entry in Field - No Goto Next Object options
52	Enter field - Unchecked for Browse & Find Modes - TAB key Goto Next Object checked
84	No Entry in Field - Return Key Goto next Object checked
148	No Entry in Field - Enter Key Goto Next Object Option is checked
256	Vertical Scrollbar on Field
1024	Display calendar icon when entering field

Multiple entry options can be applied to the same field. To decode these values, use the Revolution bitAND command as follows:

```
put varFieldEntryOptions bitAnd 4 into tResult
if tResult = 4 then set the lockText of field Field1_fld to true
```