

# **FmPro Migrator - How to Convert FileMaker Pro Layouts to HTML**

<b>1</b>	<b>Introduction</b>	
1.1	Intro - Layout to HTML Procedure	4
<b>2</b>	<b>Step 1</b>	
2.1	Step 1 - Importing Relationships & Value Lists	6
<b>3</b>	<b>Step 2</b>	
3.1	Step 2 - Importing Layouts	12
<b>4</b>	<b>Step 3</b>	
4.1	Step 3 - Generate HTML Files	21
<b>5</b>	<b>Post-Migration Tasks</b>	
5.1	Post-Migration Tasks - HTML File Editing	25



# Introduction



## Intro - Layout to HTML Procedure

---

This document provides an explanation of the steps required to convert FileMaker Pro layouts to HTML files using FmPro Migrator Developer Edition.

Revision 01

3/19/2009

This Layout to HTML conversion procedure assumes that the database tables and data have already been converted into a SQL database. This document doesn't attempt to duplicate the individual PDF manuals which cover the migration process for tables, data and relationships.

This procedure does provide information about gathering info about layouts, relationships and value lists from the FileMaker Pro source database along with detailed info about post-migration tasks.



# Step 1



## Step 1 - Importing Relationships & Value Lists

---

### Pre-Import Tasks

Prior to importing Relationships, Value Lists, and Custom Functions into FmPro Migrator Developer Edition, export a DDR XML file from the source FileMaker Pro database.

**Note:** This feature requires the use of FileMaker Pro Advanced/Developer, version 7 or higher. When creating the DDR, select to export all of the database objects into the DDR XML file.

If the source database is a FileMaker Pro 2,3,4,5 or 6 file - upgrade the file to FileMaker 7+ prior to exporting the DDR XML file.



## Step 1 - Import Relationships & Value Lists

**Migration Process**

Tables Relationships TOs Value Lists CFs Layouts Scripts GUI

Instructions: Relationships: 0

**Relationships Migration - FileMaker to SQL Databases:**  
Within FileMaker Pro 7+ export a DDR report if using FileMaker Developer/Advanced. FileMaker Pro users may manually add Relationships and Predicates on this screen. Click the Import Relationships button to import a DDR xml file for each FileMaker database file, in order to obtain a complete list of all Relationships, Table Occurrences and File References. Once this info has been imported, delete duplicate Table Occurrences. Select all of the Relationships, then click the Create SQL

Left Table Name	Right Table Name	Predicate Count	LT

Predicates:

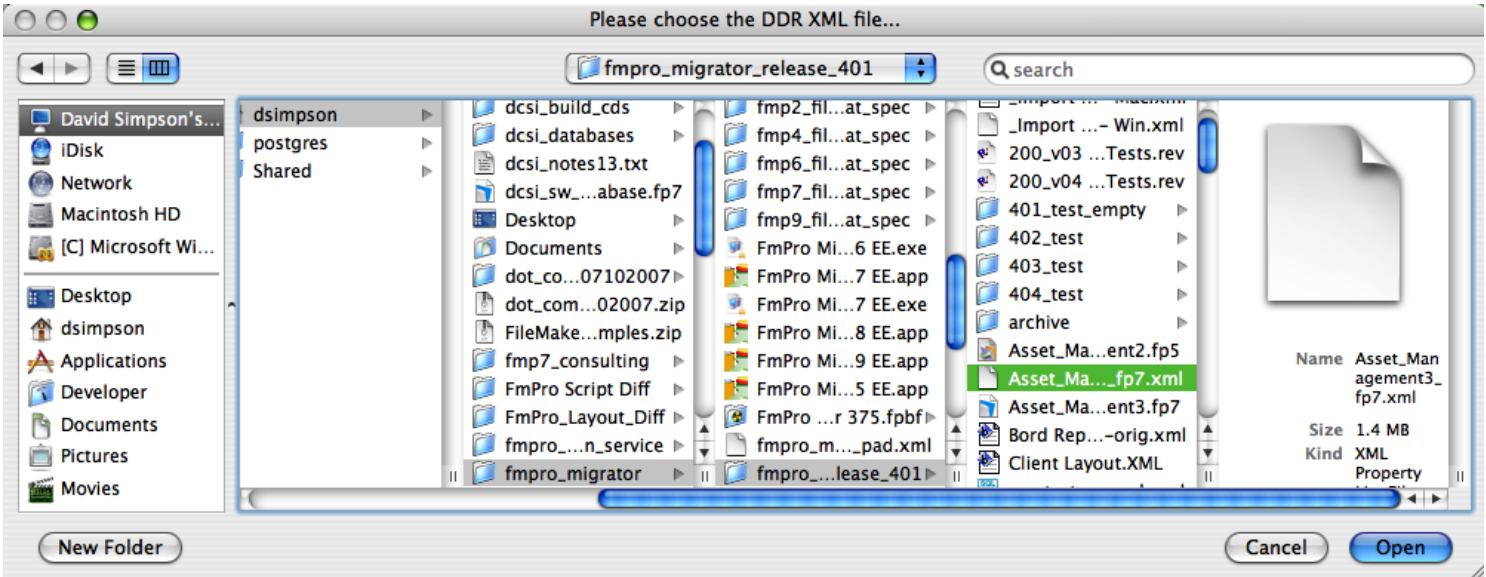
Left Table::Field	Join Type	Right Table::Field

.com Solutions Inc.

Click the Import Relationships button to import Relationships, Value Lists and Custom Functions from the DDR XML file for the source database.



## Step 1 - Select DDR XML File



The relationships will be imported from the DDR file and added to the list of relationships on the Relationships tab. TOs, value lists and custom functions will also be imported from the DDR XML file at the same time, and displayed under the Value Lists and CFs tabs in the Migration Process window.





## Step 1 - List of Value Lists

The screenshot shows the 'Migration Process' window with the 'Value Lists' tab selected. The interface includes a top navigation bar with tabs for Tables, Relationships, TOs, Value Lists, CFs, Layouts, Scripts, and GUI. Below the navigation bar are icons for a refresh cycle and a database cylinder. The main area is divided into an 'Instructions' pane on the left and a list of value lists on the right. The list shows 47 value lists, with the first six being 'Custom' and the last three being 'Static Value Lists'.

Value List Name	Type
Category List	Custom
Checkbox 1/0	Custom
Claim ID	Field
Claim Type	Custom
Claims Adjuster	Custom
Static Value List:	
Office Furniture	
Computers	
Telephone	

**Value Lists Migration:**  
[Only used for FileMaker Table Consolidation Migrationdjuster1Adjus Projects.] Value Lists are imported automatically when reading an exported DDR on the Relationships tab. Static Value Lists are migrated to the new FileMaker Pro database.

.com Solutions Inc.

Clicking the Value Lists tab shows the list of value lists which were imported from the DDR XML file. The Custom Value Lists displayed in this list will be used by FmPro Migrator to create drop down lists on the HTML files where they are used. The relationships and custom functions are not used in the



creation of the HTML files.



# Step 2



## Step 2 - Importing Layouts

FmPro Migrator Developer Edition supports the importing of layouts individually from the clipboard or in batch mode by using QuicKeys(MacOSX)/WinAutomation(Windows).

### Step 2 - Layouts Tab

Migration Process

Tables Relationships TOs **Layouts** Scripts GUI

Instructions: Layouts: 5 Client Table:  1 2 Pass#1

**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
Addresss Book	1081,719	60	0
Client	1200,1478	6	6
Client - Financial Info	1328,641	3	3
Information Overview	920,1423	5	5
Notification List View	1506,192	12	0

Portals:

Portal Name	Fill Method	Field Titles
Portal1	On Record Advance	
Portal2	On Record Advance	
Portal3	On Record Advance	
Portal4	On Record Advance	
Portal5	On Record Advance	

Layout Image

.com Solutions Inc.



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

Clicking on a layout in the list displays details for the layout, including a [3] thumbnail version of the layout image.



## Step 2 - Click Import All Layouts Button

**Migration Process**

Tables Relationships TOs **Layouts** Scripts GUI

Instructions: Layouts: 5 Client Table: Client Pass#1

**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 Book	1081,719	60	0
Client	1200,1478	6	6
Client - Financial Info	1328,641	3	3
Information Overview	920,1423	5	5
Notification List View	1506,192	12	0

Portals:

Portal Name	Fill Method	Field Titles
Portal1	On Record Advance	
Portal2	On Record Advance	
Portal3	On Record Advance	
Portal4	On Record Advance	
Portal5	On Record Advance	

Layout Image

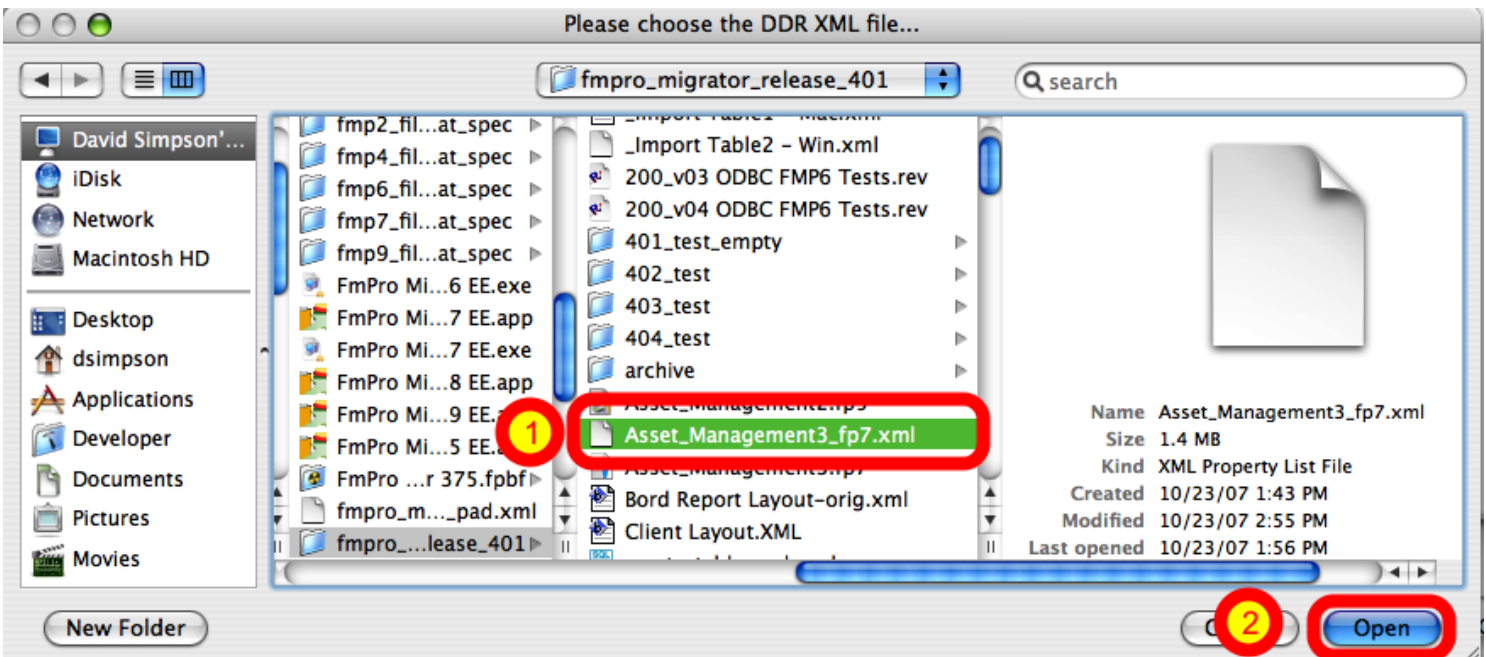
The Import All Layouts button starts the automated process of importing all of the layouts from the FileMaker database which is currently open. Open the source FileMaker Pro database file which you are migrating, then click the Import All Layouts button.



Repeat the following steps to import layouts from additional source database files.

Once all of the layouts have been imported from each source database file, they should be reviewed to determine if there are any layouts having duplicate names. Manually rename any duplicate layouts within the list of layouts. A thumbnail image of each layout and the Display Records From: TO are available on the Layouts screen in order to assist the developer when identifying each layout.

## Step 2 - Select the DDR XML File

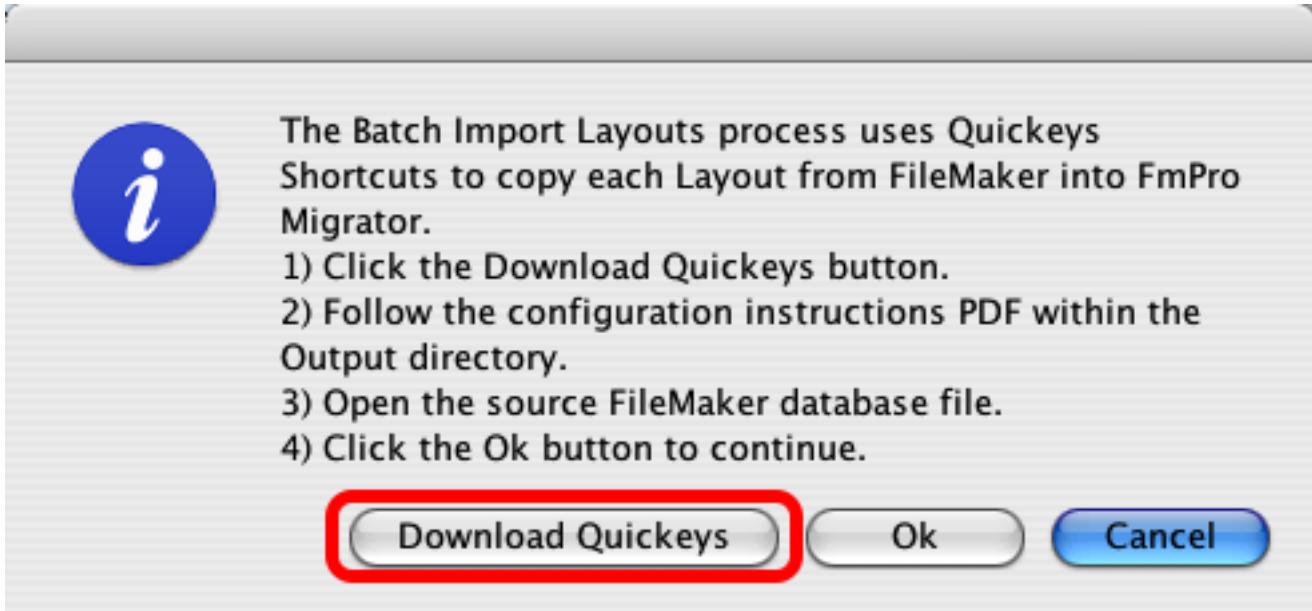


FmPro Migrator needs to read the table occurrence associated with each layout from the DDR XML file which was previously exported.

Select the DDR XML file, then click the Open button.



**Step 2 - Click Download QuicKeys Button (Mac OS X) - [1st Time Setup Step]**



The Download QuicKeys dialog is displayed by FmPro Migrator on Mac OS X. Click the Download QuicKeys button if you don't already have QuicKeys 3.03 (or higher) installed on Mac OS X.

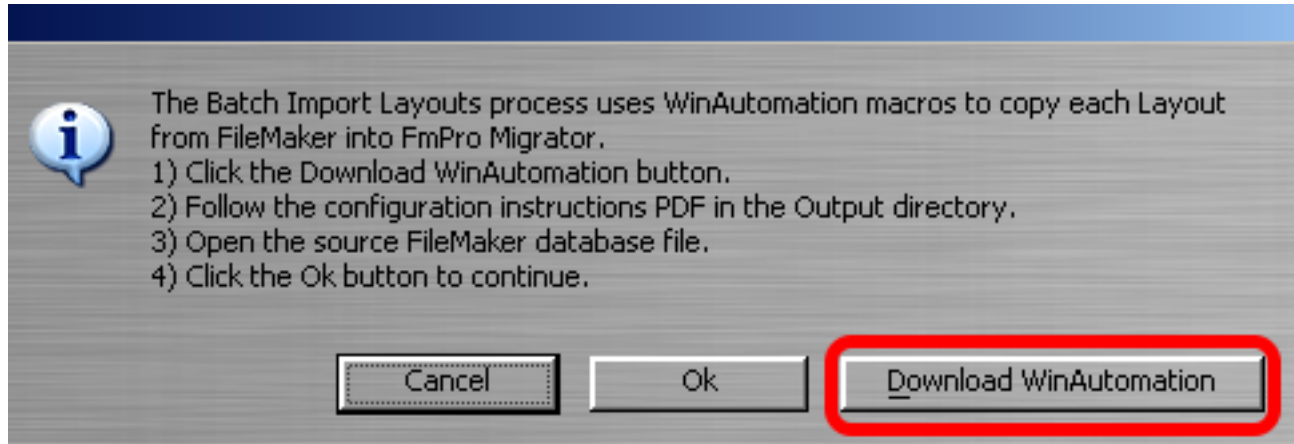
Click this download button even if you do have QuicKeys installed, because clicking this button triggers FmPro Migrator to create the FmPro Migrator QuicKeys Installation.pdf and FmPro Migrator QuicKeys Shortcuts - Mac.qkexport files within the output directory. Whenever the QuicKeys Shortcuts have been updated within FmPro Migrator, you will need to click on this button in order to extract the latest shortcuts from FmPro Migrator for installation on your computer.

Follow the instructions within the FmPro Migrator QuicKeys Installation.pdf file to import the QuicKeys shortcuts from the FmPro Migrator QuicKeys Shortcuts - Mac.qkexport file into QuicKeys and set up the QuicKeys variables. These QuicKeys shortcuts use 10 QuicKeys variables which must be created manually prior to using the shortcuts.





## Step 2 - Click Download WinAutomation Button (Windows) - [1st Time Setup Step]



The Download WinAutomation dialog is displayed by FmPro Migrator on Windows. Click the Download WinAutomation button if you don't already have WinAutomation 1.2.0.622 (or higher) installed on Windows.

Click this download button even if you do have WinAutomation installed, because clicking this button triggers FmPro Migrator to create the [FmPro Migrator WinAutomation Installation.pdf](#) and [FmPro Migrator WinAutomation Jobs.zip](#) files within the output directory. Whenever the WinAutomation Jobs have been updated within FmPro Migrator, you will need to click on this button in order to extract the latest versions of these jobs from FmPro Migrator for installation on your computer.

Follow the instructions within the [FmPro Migrator WinAutomation Installation.pdf](#) file to import the WinAutomation jobs from the [FmPro Migrator WinAutomation Installation Jobs.zip](#) file into WinAutomation.

## Step 2 - Start WinAutomation Console Application (Windows) - [1st Time Setup Step]

The WinAutomation Console application will install and startup the WinAutomation Service the first time it is launched. You may close the WinAutomation Console after importing the WinAutomation jobs.



**Step 2 - Click the Copy All Layouts Button (2nd Time)**

**Migration Process**

Tables Relationships TOs **Layouts** Scripts GUI

Instructions: Layouts: 5 Client Table: Client Pass#1

Layout Name	Size	Fields	Portals
Addresss Book	1081,719	60	0
Client	1200,1478	6	6
Client - Financial Info	1328,641	3	3
Information Overview	920,1423	5	5
Notification List View	1506,192	12	0

Portals:

Portal Name	Fill Method	Field Titles
Portal1	On Record Advance	
Portal2	On Record Advance	
Portal3	On Record Advance	
Portal4	On Record Advance	
Portal5	On Record Advance	

Layout Image

Select the DDR XML file, then click the Ok button on the Batch Import dialog.

FmPro Migrator will then use the QuicKeys/WinAutomation shortcuts to drive the FileMaker user



interface in the same manner as a user would manually copy each of the Layouts from FileMaker Pro into FmPro Migrator.

## Step 2 - Controlling Batch Capture Mode



FmPro Migrator takes over the computer's keyboard/mouse when controlling FileMaker Pro to capture or create layouts in FileMaker Pro. If you need to cancel the process while it is running, click the Done button at the bottom of the Layouts tab.





# Step 3



### Step 3 - Generate HTML Files

#### Step 3 - Click Convert Layouts to HTML Button

**Migration Process**

Tables Relationships TOs Value Lists CFs Layouts Scripts **GUI**

Instructions:

**Convert Layouts to HTML:**  
Click the Convert Layouts to HTML button to convert each FileMaker Pro layout into an HTML file. Each file includes fields, text labels, images, graphic objects and JavaScript code for implementing portals and pop-up calendar pickers.

**Convert to Revolution:**  
Click the Convert Database to Revolution button to convert the entire FileMaker Pro database into a Revolution stack. Layouts are re-created as cards and ScriptMaker scripts are converted into Revolution handlers within the

Buttons:  
Generate Perl CGI Scripts  
Export Layouts as XML  
Convert Database to Revolution  
**Convert Layouts to HTML**

<Please Select>

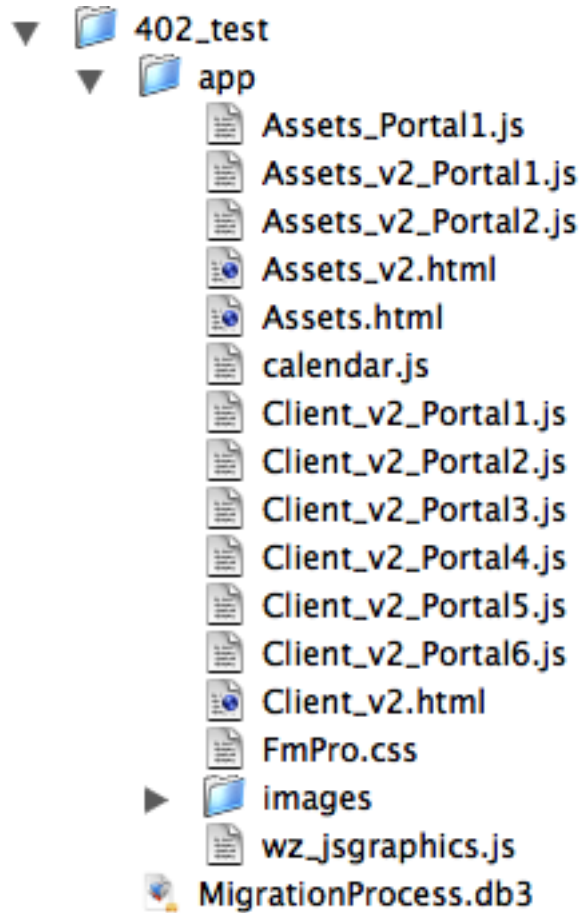
.com Solutions Inc.

Click the Convert Layouts to HTML button. Within a few seconds, an HTML form will be generated for



each FileMaker Pro layout within a new /app directory inside FmPro Migrator's previously selected output directory.

### Step 3 - Generated HTML Files List



An HTML file is created for each FileMaker Pro layout, along with a JavaScript file for each portal and images which have been extracted from each layout.

Layout images are converted into PNG, JPG or GIF files and stored unduplicated within the /images subdirectory. If a particular image such as a company logo image is used on multiple layouts, it is stored only once within the /images directory. Changing the one single copy of this image file will automatically affect the displayed image on every HTML file where it is referenced.

The open source Walter Zorn VectorGraphics library [wz\\_jsgraphics.js](#) JavaScript file is used for drawing lines, circles/ovals, rectangles and rounded rectangle objects on the web page.



The open source calendar.js JavaScript file is used for displaying a calendar picker for any field which had a drop down calendar configured within the original layout.

The \_Portal.js files are used to implement an [ExtJS Data Grid](#) for each layout portal. [ExtJS software must be licensed separately for commercial projects, please see the <http://www.extjs.com/products/license.php> page for more info.]



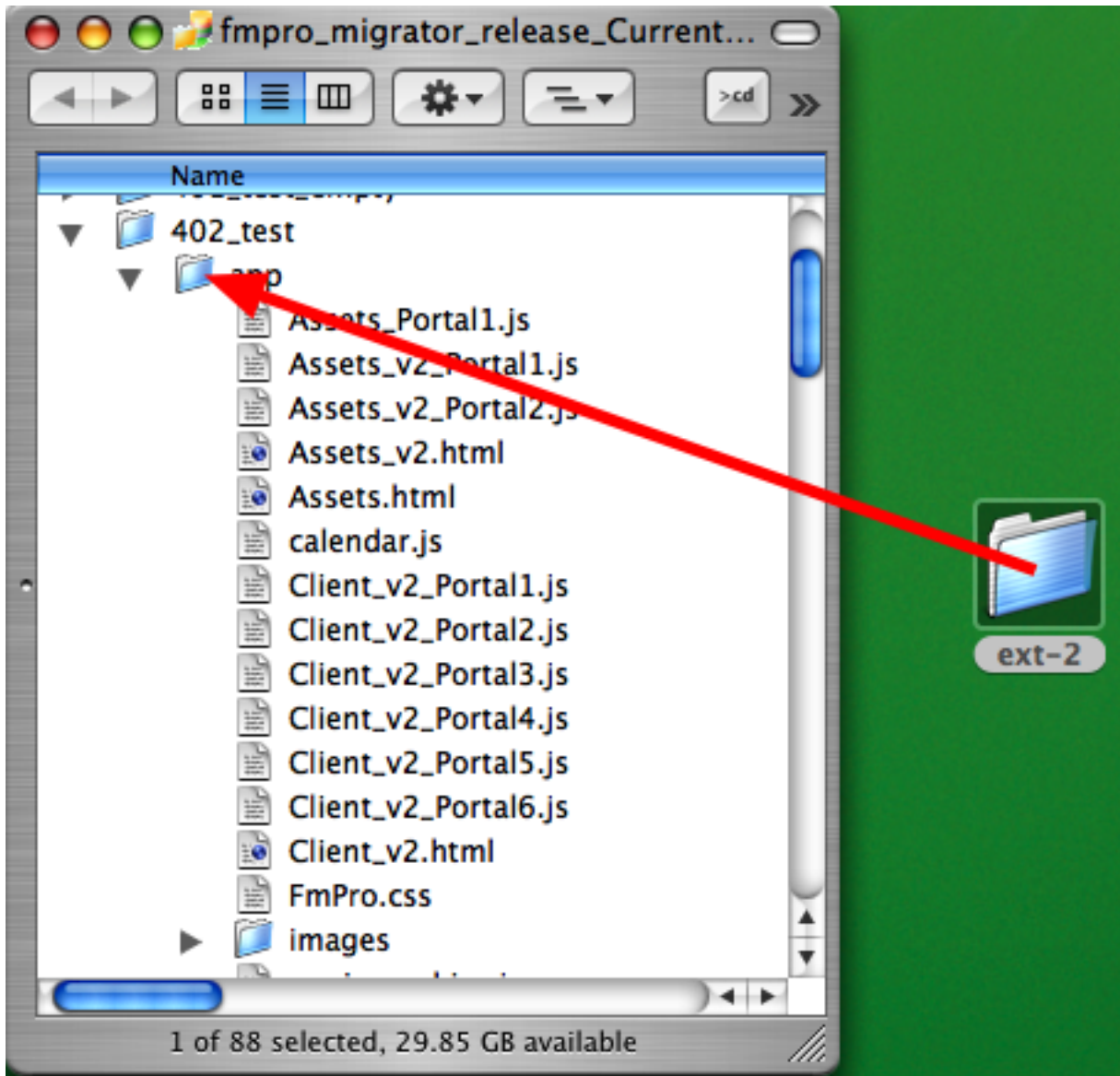
# Post-Migration Tasks





## Post-Migration Tasks - HTML File Editing

### Download & Install ExtJS



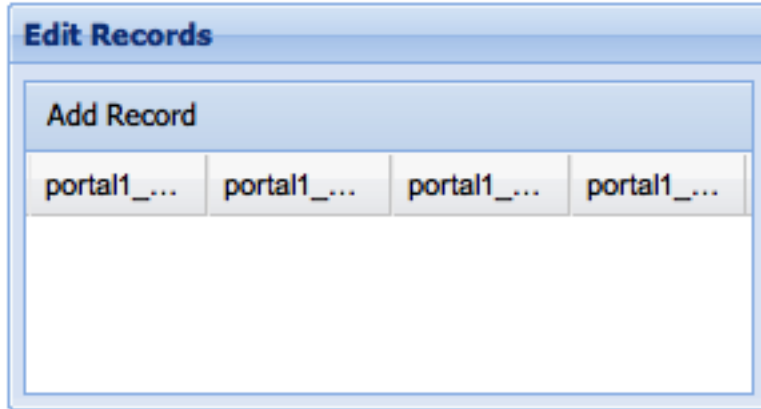
Download and install ExtJS 2.2.1 or higher within the /app folder.

The ExtJS JavaScript library is used to display a JavaScript Data Grid object in place of the original FileMaker Pro portal object.

**Note:** The ExtJS library is available at no charge for open source projects, but a [license](#) needs to be purchased for use in commercial development projects.



## ExtJS Data Grid Customization - Default Display



By default, FmPro Migrator generates ExtJS compatible JavaScript files utilizing the portal field names as data grid column headers.



## Editing the Assets\_v2\_Portal2.js File

```
// the column model has information about grid columns
// dataIndex maps the column to the specific data field in
// the data store (created below)
var colModel = new Ext.grid.ColumnModel([
  1 header: "Maint Date",
    dataIndex: 'portal2_maintenance_record__date_fld',
    width: 68,
    sortable: true,
    editor: new fm.TextField({
      allowBlank: true
    })
  2 header: "Maint. ID",
    dataIndex: 'portal2_maintenance_record__maintenance_id_fld',
    width: 67,
    sortable: true,
    editor: new fm.TextField({
      allowBlank: true
    })
  3 header: "Notes",
    dataIndex: 'portal2_maintenance_record__notes_fld',
    width: 67,
    sortable: true,
    editor: new fm.TextField({
      allowBlank: true
    })
  4 header: "Asset ID",
    dataIndex: 'portal2_maintenance_record__asset_id_fld',
    width: 67,
    sortable: true,
    editor: new fm.TextField({
      allowBlank: true
    })
])
```

Editing the header: attribute for each column will change the default column name into a user-friendly column header. The columns for this portal have been renamed to read: Maint Date, Maint. ID, Notes and Asset ID.



## ExtJS Data Grid Customization - Updated Column Headers

Maint Date	Maint. ID	Notes	Asset ID
------------	-----------	-------	----------

Save the changes to the .js file and refresh the browser display to see the updated Data Grid.

Additional changes will also need to be made within each Data Grid .js file in order to implement loading of the data from a database server, in place of the default behavior which loads data from an xml file.

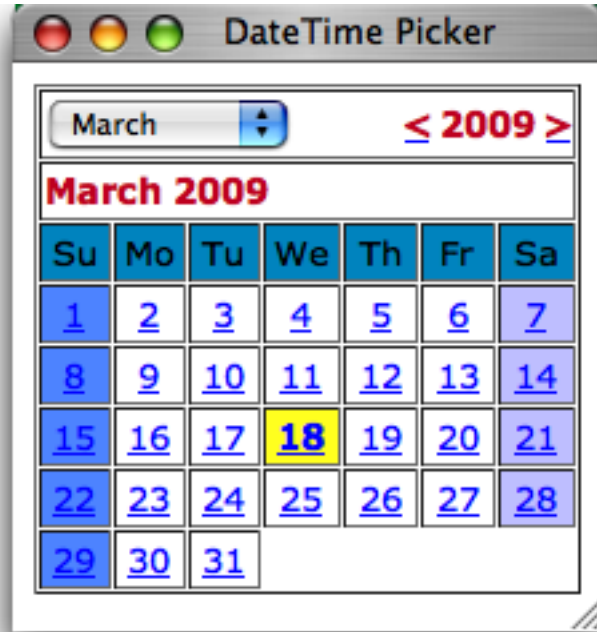
## ExtJS Data Grid - Vertical Size

The ExtJS Data Grid takes up more vertical space than the original layout portal. An ExtJS Data Grid needs to be at least 120 pixels tall in order to be useful. It is generally easier to move objects around on the original FileMaker Pro layout to accommodate this extra vertical space. Once the changes have been made to the original layout, copy the changed layouts back into FmPro Migrator and re-generate the HTML files again.



### DateTime Picker Display

Purchased



Each FileMaker Pro date field configured with a drop down Calendar, is configured with a calendar icon and a link to the [Calendar.js JavaScript DateTime Picker](#), written by [TengYong Ng](#).

### DateTime Picker Icon - with Overlapping Rectangle Graphic

Acquired <<Date Purchased>>		Depreciated <<Depreciation>>	
Purchased	<input type="text"/>	In Service	<input type="text"/>
From	<input type="text"/>		
Cost	<input type="text"/>		

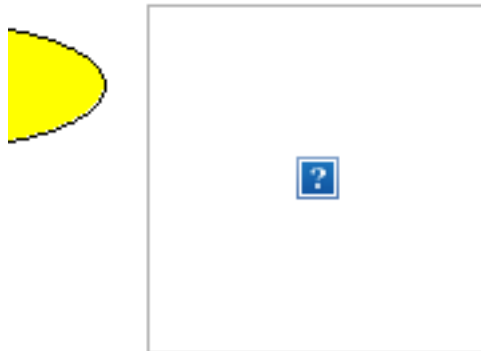
The [Walter Zorn JavaScript VectorGraphics library](#) (wz\_jsgraphics.js file) is used for drawing rectangles, circles/ovals and lines on the background of the HTML web page in order to duplicate the equivalent functionality within the FileMaker Pro layout. However some FileMaker Pro layouts utilize background objects like rectangles behind the fields of the layout, as shown in this screenshot. When a graphic object like the yellow colored rectangle is drawn behind a field having the calendar icon, the calendar icon image is not visible as it is covered up by the rectangle.



**DateTime Picker icon - with Rectangle Graphic Moved**

One way to work around the display issue with these two objects is to move or resize the yellow background rectangle as shown in this screenshot.

**Layout Images Stored as Reference**



On this sample layout, the bottom icon is displayed twice on the same layout when viewed within FileMaker Pro. But the top icon is only stored as a reference to the original file and is not actually embedded within the original layout. For this reason, the image is not extracted or available for display



on the web page. Therefore, if images are not displayed on a particular web page, check to make sure that the original layout image wasn't stored as a reference. For any missing images, the layouts should be modified to embed the images within the layout, then re-copy the layouts into FmPro Migrator and re-generate the HTML files.

### Overlapping Image & Graphic Objects - Web Browser View



Within a web browser, this circular object appears to be missing the image object which should be displayed within the circle. The circle is being drawn with the JavaScript VectorGraphics library, and it is covering the circular graphic image which should be displayed within the circle.

### Overlapping Image & Graphic Objects - Dreamweaver Design View

```
<!-- Image: image5_img.jpg -->  

```

The screenshot shows the Dreamweaver design view interface. At the top, the source code for an image element is displayed. Below the code is the 'PROPERTIES' panel for the selected image. The panel includes fields for 'CSS-P Element', 'L' (581px), 'W' (24px), 'Z-Index' (1), 'Bg image', 'T' (41px), 'H' (24px), 'Vis' (default), and 'Bg color'. A red circle with the number '2' is placed over the 'Z-Index' field. Below the properties panel, the design view shows a toolbar with 'Delete' and 'Find' buttons. A blue square with a question mark icon is highlighted with a red circle and the number '1'.

Within the Dreamweaver design view, the underlying graphic image can be seen, because the JavaScript VectorGraphics library is not being run within design view. Once the graphic image has



been selected (1), open the properties window and set the Z-index to 1 (2).

### Overlapping Image & Graphic Objects - Updated Web Browser View



Once the HTML file changes have been saved by Dreamweaver, refresh the web browser window, or switch to Dreamweaver's Live View mode to see the changes. The image now shows up and the gradated image below the purple line has also been changed to display over top of the drawn image of the purple line.

Even though the image now shows up, the drawn circle is now covered up under the ? image graphic image. For situations of this type, it would be better to draw the entire object within a graphics program and paste the finished graphic onto the FileMaker Pro layout.

### Text Labels with Multiple Sizes/Styles



Text labels having multiple font, sizes or text styles will be rendered with the 1st text style read from the FileMaker Pro layout XML description of the object. If a section of the text is supposed to be displayed as a smaller sized font, the text will overflow to the next line.





### Text Labels with Multiple Sizes/Styles - Adjusted Size Font Tag

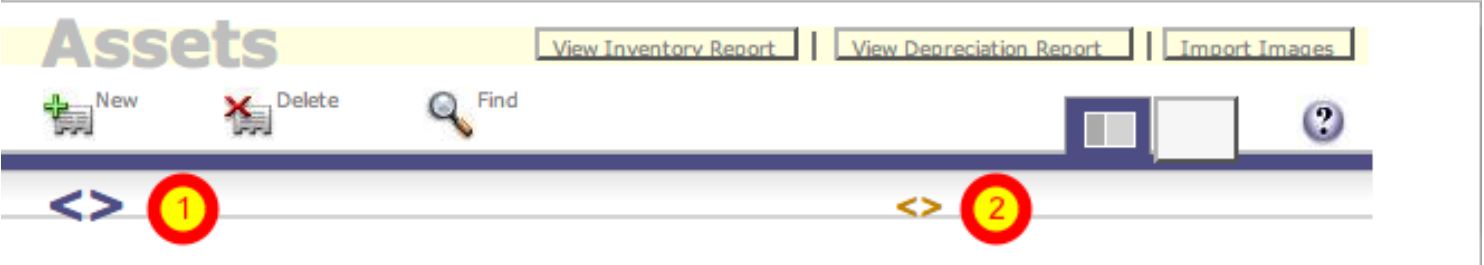
```
<!-- Text: text22_lbl -->  
<p id="text22_lbl" style="position: absolute; top: 850px; left: 368px; width: 250px; height: 24px;  
text-align: left; vertical-align: top; color: #FFCC00; border-width: 0px border-color: #008000;  
z-index: 3; font: bold 16pt Arial; background-color: transparent; font-family: Arial; font-size:  
16pt;">Insurance Policies <font size=-3>(Active)</font></p>  
<!-- Portal: Portal1 -->  
<div id="Portal1_editor-grid" style="position: absolute; top: 895px; left: 0px; width: 349px;  
height: 211px; z-index: 2">  
<script type="text/javascript" src="Client_v2_Portal1.js"></script>  
</div>  
  
<!-- Text: text23_lbl -->
```



Adding an additional `<font size=-3>` HTML tag resizes the trailing portion of the text to match the original FileMaker Pro layout text.

### FileMaker Pro Layout Merge Fields

```
50 <!-- Text: text3_lbl -->  
51 <p id="text3_lbl" style="position: absolute; top: 83px; left: 400px; width: 190px; height: 24px;  
text-align: left; vertical-align: top; color: #CC9900; border-width: 1px border-color: #000000;  
z-index: 3; font: bold 10pt Verdana; background-color: transparent"><<Category>></p>
```



Merge fields don't have a direct correlation to an HTML object. Since merge fields are enclosed within `<<>>` characters within layout text objects, each of these objects are reproduced as incompletely displayed HTML objects within the HTML document. One potential alternative would be a PHP statement which would display information at the specified location on the web page. However putting multiple fields together allowing the text to flow in the same manner as the merge field in the original



layout would require additional work.

## Printable Web Graphics

```
    jg_doc.setPrintable(false); // set to true if web page needs printed
    jg_doc.print(); // draws directly onto the web page
}

var jg_doc = new jsGraphics(); // draw directly into document

fmProGraphicsFunction();

//-->
</script>
```

Each HTML file which contains FileMaker Pro line, circle/oval and rectangle graphic objects will include a JavaScript function named `fmProGraphicsFunction`. This function uses the JavaScript `VectorGraphics` library (`wz_jsgraphics.js` file) to draw these objects on the background of the HTML page. In order to improve performance, the `setPrintable(false)` function is executed by default. If all of the web page objects need to be printed to a printer, this function should be changed by passing the true parameter to the `setPrintable()` function.



## Database Objects

The screenshot shows the 'Insert' menu of a software application. The menu items are as follows:

- Tag... ⌘E
- Image ⌘I
- Image Objects (Unsorted) ▶
- Media ▶
- Table ⌘T
- Table Objects ▶
- Layout Objects ▶
- Form ▶
- Hyperlink
- Email Link
- Named Anchor ⌘A
- Date
- Server-Side Include
- Comment
- HTML ▶
- Template Objects ▶
- Recent Snippets ▶
- Spry ▶
- InContext Editing ▶
- Data Objects ▶**
- Customize Favorites...
- Get More Objects...

The 'Data Objects' submenu is open, displaying the following options:

- Recordset
- Stored Procedure
- Dynamic Data ▶
- Repeat Region
- Show Region ▶
- Recordset Paging ▶
- Go To ▶
- Display Record Count ▶
- Master Detail Page Set
- Insert Record ▶
- Update Record ▶
- Delete Record
- User Authentication ▶
- XSL Transformation



Dreamweaver CS4 (and many other tools) provides dynamic data objects to enable the development of database backed websites. These features are documented in the Dreamweaver manual and are beyond the scope of this manual.

## Unsupported Layout Objects

There are a few layout objects which are not supported during the HTML conversion process:

**Tab Control** - The Tab Control functionality can be duplicated by making multiple copies of the layout containing the Tab Control, and laying out the elements of each tab on a separate layout. This is the same technique which was utilized within FileMaker Pro database layouts prior to the inclusion of the Tab Control feature, introduced with FileMaker Pro 8.

**Field Tab Order** - The XML definition of a FileMaker Pro layout does not include field tab order information. To set a specific tab order for fields contained within the form, add the `tabindex` HTML tag for each field using an HTML editor.

**Rounded Rectangle Objects** - Rounded Rectangle objects are drawn by the JavaScript Vectorgraphics library as regular rectangles, similar to the way they are displayed by the FileMaker Pro Instant Web Publishing feature. If the display of rounded rectangle objects is critical to the look of the resulting HTML page, these types of objects should be replaced with graphic images pasted directly onto the FileMaker Pro layout.

**Object Z-index (stacking order)** - The XML definition of a FileMaker Pro layout does not include object stacking order (z-index) information. FmPro Migrator automatically assigns z-index values for each object according to the following table:

[Lower numbers are at the bottom of the stack, covered up by objects with a higher z-index value.]

<b><u>Object Type</u></b>	<b><u>Z-index</u></b>
Text Label	5
Button	4
Field	3
Field - Calendar	3
Menu	3
Checkbox	3



Radio Button	3
Portal	2
Line	1
Rectangle	1
Oval/Circle	1
Graphic Image	none (0)