



# FmPro Migrator

## Platinum Edition

### The Quick Prototyping Tool for PHP Web Applications

Use FileMaker® Pro as a Rapid Application Development Tool for PHP Web Applications

#### Top 10 Features

- Scale-up existing database solutions to PHP web apps supporting 12 SQL database servers.
- Export charts to pChart-generated PNG image files in batch mode for high traffic web sites.
- Convert FileMaker Charts into server-side pChart charts or client-based Fusion Charts.
- Convert Portals into jqGrid or ExtJS grid objects.
- Tab Controls and their enclosed objects are converted recursively any number of levels deep.
- Static and Dynamic Value Lists are converted into PHP code.
- Table Occurrences and Relationships are converted into MVC architecture Model files.
- Convert FileMaker Pro scripts into PHP code, including commonly used button script steps.
- Build PHP web applications supporting any web browser or mobile device.
- Save money by building a working web application from your original database which can be further enhanced by PHP developers anywhere in the world.

#### Layouts

#### Navigation Toolbar

#### Value Lists

Value List Test Fields

- vi\_test fld1\_numeric: 1 Value #1
- vi\_test fld2\_numeric: Value #2
- vi\_test fld1\_text: abc
- vi\_test fld2\_text: MacBook Pro 17
- WinBook W360

#### Charts

#### Radio Buttons & Checkboxes

Radio Btn 2 Column

- Office Furniture
- Computers
- Telephones
- Table
- Chair
- Router

Checkbox 2 column

- Office Furniture
- Computers
- Telephones
- Table
- Chair
- Router

#### Layout Scripts

#### Scripts

```

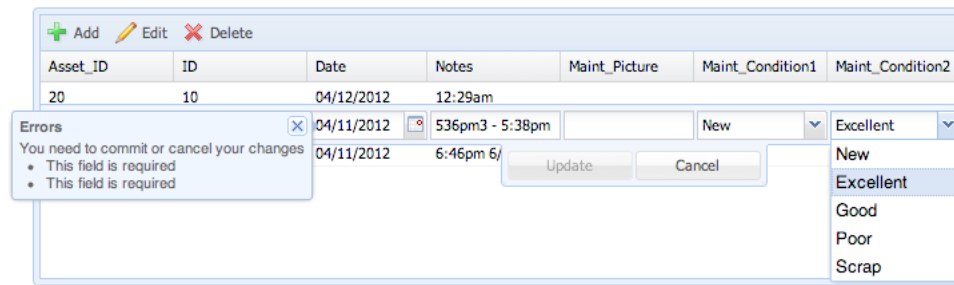
app
├── app_controller.php
├── config
├── controllers
├── converted_scripts
│   ├── affiliation_query.php
│   ├── clear_sort_indicator.php
│   ├── depreciation_report.php
│   ├── find.php
│   ├── go_to_form_layout.php
│   ├── go_to_information_layout.php
│   ├── go_to_list_layout.php
│   ├── import_images.php
│   ├── inventory_report.php
│   ├── license_query.php
│   ├── multiple_dwelling_query.php
│   ├── occupancy_classification_query.php
│   ├── open_script.php
│   ├── open_url_test.php
│   ├── other_items_query.php
│   ├── rf_convert_for_field_serial_number.php
│   ├── rf_convert_for_table_asset_management2.php
│   ├── sort_by_assigned_to_ascending.php
│   ├── sort_by_assigned_to_descending.php
│   ├── sort_by_assigned_to.php
│   └── sort_by_category_ascending.php

```

82 Layouts Processed in 11.3 Sec.  
40 Scripts Processed (402 lines).  
61,997 Lines of Application Code Generated.

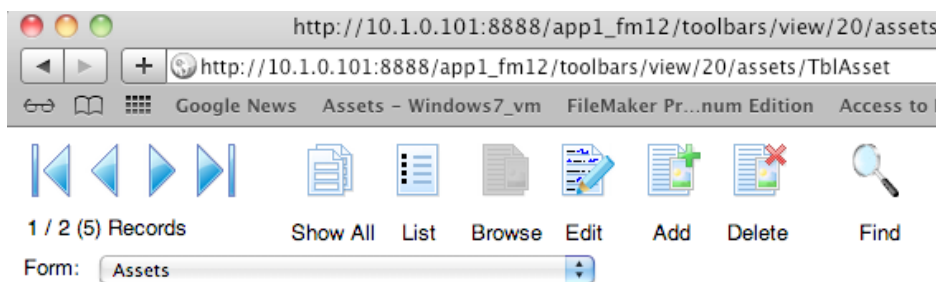
## FileMaker Portal to jqGrid/ExtJS Grid Conversion

FileMaker portals are converted into grid objects using either the jQuery jqGrid PHP library or the ExtJS grid. The jqGrid is a high performance PHP grid providing the capability of handling millions of records thru a direct connection to supported SQL database servers.



The ExtJS grid provides compatibility with all CakePHP supported database servers by using the CakePHP Models as its data source. The ExtJS grid even supports non-database data sources since it reads its data directly from CakePHP models.

## Query By Example Record Searching & Foundsets



Fields, menus, radio button and checkbox sets are used by the query processing code.

Radio button and checkbox sets are searchable in the same manner as within the original FileMaker database.

The primary keys for the records found by the query are stored within the foundset session variable. Only the foundset records will be displayed when navigating between records. Clicking on the Show All Records button on the navigation toolbar view clears the foundset session variable and displays all records using the index view.

While in foundset mode, the record position within the foundset along with the total number of records are displayed in the navigation toolbar. This slight change in the display of the navigation toolbar provides a quick visual indication that the application is in foundset operating mode. Foundsets and the current record position are retained between forms using the same model, as within the original database.

## Gritter Dialogs & Field Validation

The jquery.gritter.js plug-in provides a professional looking non-modal alert dialog for web applications. The Gritter dialog looks similar to a MacOS X Growl alert dialog - however it is cross-platform and cross-browser compatible. This plug-in is fully integrated into the generated web application wherever flash messages are sent from within controller code.

By default, Info dialogs stay on the screen for a few seconds and then fade out. Error dialogs stay on the screen until closed by the user. The behavior and look of the Gritter dialogs is fully customizable by changing the image files in the webroot directory or the flash\_info.ctp or flash\_error.ctp files within the views/elements directory.

Field validation and validation error message text is read directly from the original FileMaker Pro field definition and is written into each model file.

