Talking to Drupal
Pause On Error NYC 2009

Andy Chase
The Proof Group LLC

July 17, 2009

Abstract

With recent versions of FileMaker Pro it has gotten substantially easier to move information back and forth to web-facing applications without necessarily having to resort to using Web Publishing. External SQL Sources makes it possible to talk directly to MySQL, which for many one-off, custom web applications is all you need. But for well-established Content Management Platforms such as Drupal[1] it can get a little more complicated. The goal of this session is to look at some of the considerations involved with getting data in and out of Drupal from FileMaker, and in and out of FileMaker from Drupal.

Contents

[1] http://drupal.org

1 Why Drupal?

There are numerous mature, open source Content Management Systems available, each with their own strengths and weaknesses. Drupal's strength lies in its extensibility and in its community.

1.1 Free as in Speech

Not only is Drupal free of cost, it is released under the Gnu Public License, which means that you are free to re-use, redistribute, modify and customize it to your heart's content - and with the knowledge that a given version of Drupal will *always* be free. This gives you the benefit of a huge, worldwide community of fellow Drupal users, and it gives your clients the peace of mind of knowing that they are not locked into a proprietary system.

1.2 Extensibility

Drupal's system of hooks provides access to many low-level, core system operations. Combined with its powerful Forms API, it's possible to write modules that extend or improve upon nearly any aspect of a Drupal site *without hacking the core code*.

Because contributions to the Drupal project are all released under the Gnu Public License, all of the modules which get contributed back to the project are free for everyone to use, and because everyone has access to the source code, bugs and security issues often get noticed and fixed much more quickly than similar issues in a closed-source solution.

1.3 Community

Drupal excels in providing infrastructure to foster a cohesive codebase, setting coding standards, and providing the developer and user communities with numerous avenues for finding help (web forums, mailing lists, IRC chats, et cetera.) As a result there is better communication among developers, less duplication of effort, and overall the project's code is more consistent.

2 Pulling from Drupal to FileMaker

2.1 External SQL Sources

The easiest way to pull data from Drupal's database into FileMaker is by setting it up as an External SQL Source[2]. You'll need to configure an ODBC connection to Drupal's MySQL database using one of the following drivers:

For Windows, you can use the MySQL Connector/ODBC 5.1[3] For Mac OS X you'll need to use the Actual ODBC Driver For Open Source Databases.[4]

2.2 MySQL Access and Permissions

You'll need to make sure that your MySQL login allows access from IP address of the machine hosting your FileMaker database; many web hosting companies restrict remote access to MySQL for security reasons. Depending on your host, it may be possible to enable remote access to all IPs (convenient for testing and development, but less secure), or to a specific IP address (less convenient, but more secure.)

Your MySQL login will minimally need SELECT permissions for all of the tables on the Drupal database. If you don't plan to write any data to the Drupal database, it's a good idea to use a MySQL login that *only* has SELECT permissions. This will prevent you from inadvertently writing to Drupal's core tables, which can have unpredictable consequences.

2.3 MySQL Indexes

You may encounter problems adding table occurrences to your FileMaker graph from the Drupal database if the MySQL table doesn't have a unique index. The Drupal core database is generally well defined and should behave well in the graph, but some third-party modules may define tables without proper indexes. If you encounter such a table, consider taking the time to roll a patch[5] for that module and contribute it back to the community, rather than just fixing it for your own database. It is possible to specify which column(s) should be used as the primary key in FileMaker, but it makes more sense to do it on the MySQL side; performance will be better all around.

2.4 Get to Know the Drupal Database

The Drupal database has a lot of tables, and it can sometimes be hard to track down where the data you're looking for lives. In many cases it may be split up among several tables. The Schema module[6] provides helpful details about

[2] http://www.filemaker.com/support/technologies/sql.html
[3] http://dev.mysql.com/downloads/connector/odbc/5.1.html.
[4] http://su.pr/5rM8aI
[5] http://drupal.org/node/128209
[6] drupal.org/project/schema

Drupal's tables, and also comes in handy for developing a custom schema if you decide to use ESS to push data to Drupal.

2.5 SOAP

The FM::Nexus Web Services Plugin[7] provides a straightforward way to access web data using the SOAP protocol. The Drupal Services module[8], still in development, provides a framework for making Drupal data available by SOAP. As the Services module matures this may be a very promising technique for pulling Drupal data into FileMaker.

3 Pushing to Drupal from FileMaker

Pushing data to Drupal gets a lot trickier, because Drupal has its own fairly specific ways of manipulating data as it gets stored and loaded for display on the web.

3.1 Don't Write to Core Tables!

While it may be tempting to drag the `node` table into your graph, add a layout, and start editing `title` and `body` fields willy-nilly, **don't do it!** Between Drupal's Hooks system[9] and node versioning capabilities, it would be extremely difficult (if not impossible) to replicate all of the ways Drupal manipulates data on its way into or out of the database with all of the modules enabled for a particular site. Writing directly to core tables may have unpredictable results, and has the potential to break your site.

If you can't write directly to the Drupal database, then what can you do? Fortunately you have a few options:

3.1.1 Use a Cache Table

If you need to manipulate native Drupal data structures such as users or nodes from FileMaker, one approach would be to write a Drupal module that defines a new table that acts as a simple cache. You can safely write data to this table via ESS, and then implement `hook_cron` in your Drupal module to examine the cache table and run its contents through the appropriate Drupal APIs.

While you don't get the satisfaction of having your data changes appear live online instantaneously, this approach does allow you to take advantage of Drupal's native data structures and all of the contributed modules that enhance them (for example Views[10] and CCK.[11])

[7] http://fmnexus.com/products/webservices/
[8] http://drupal.org/project/services
[9] http://api.drupal.org/api/group/hooks
[10] http://drupal.org/project/views
[11] http://drupal.org/project/cck

3.1.2 Define your own non-core API

If you want to display arbitrary data from your FileMaker database on the web, you can write a Drupal module which defines a new table as above, and then defines the appropriate permissions, callbacks, and theming functions to display data from that table on your Drupal site.

3.1.3 Use XML-RPC

Drupal provides a native XML-RPC endpoint out of the box, and using the XML-RPC hook[12] you can easily expose Drupal methods to remote clients... including FileMaker! Using the SmartPill plugin[13] and a library such as the Incutio XML-RPC Library[14] you can pass data from FileMaker to Drupal by XML-RPC calls. Alternately, you could write FileMaker calcs that build Python scripts an execute them on the fly; Python's native xmlrpclib[15] library is tremendously powerful and easy to use.

4 Pulling/Pushing to FileMaker From Drupal

4.1 `FX.php` and the FileMaker PHP API

If you're using FileMaker Server Advanced (7 or later) or FileMaker 5 and 6 with Web Companion, you could write a custom Drupal module which uses the `FX.php`[16] library or the FileMaker API for PHP[17] to access a layout in your FileMaker database.

4.2 This Just In

The Web Services Manager from 360works[18] makes it possible to call FileMaker scripts remotely using the SOAP protocol (this is the inverse of the functionality that FM::Nexus' plugin provides.) Using the Drupal SOAP Client module[19], it should be possible to write a module that can talk to FileMaker with SOAP calls.

[12] http://su.pr/2vUmZD
[13] http://www.scodigo.com/products/smartpill-php
[14] http://su.pr/1bKZyZ
[15] http://docs.python.org/library/xmlrpclib.html
[16] http://www.iviking.org/FX.php/
[17] http://www.filemaker.com/support/technologies/php.html
[18] http://www.360works.com/web-services-manager/
[19] http://drupal.org/project/soapclient