

FX.php

A FREE, Open Source PHP database abstraction class for accessing FileMaker Pro and other databases by Chris Hansen with Chris Adams, Gjermund Thorsen, and others.

FileMaker Pro has quite a following, and with good reason: it mixes the power of a relational database with phenomenal ease-of-use. PHP is a free, embedded scripting language in use on nearly half of all Apache web servers (which themselves account for about 60% of web servers.) PHP can access just about any data source, but there was no easy way for it to pull data from FileMaker until 2001. FX.php was the first PHP class which allowed FileMaker enthusiasts to access their data via PHP. More recent versions have also added support for MySQL, PostgreSQL, OpenBase, ODBC data sources, and more.

FX.php is a PHP class which parses the XML output by FileMaker Pro's web companion into a multi-level array which is easily manipulated using PHP. Records can be searched, edited, created, and deleted. In addition, a number of other actions can be performed including script execution, and opening and closing databases. (See the documentation for more details.) When any action is performed, the returned array is organized in up to four arrays, as follows (these are the relevant index or key values):

- Level 1: (optional, depending on the return type)
'linkNext', 'linkPrevious', 'foundCount', 'fields', 'data', 'URL', 'errorCode', 'valueLists'
- Level 2: (of 'data')
RecordID.ModificationID
- Level 3:
fieldName
- Level 4: (optional, depending on the return type)
Numbers, starting at zero; one for each related or repeating value

So, a reference to a specific value might look like one of the following:

```
$dataArray['12.3']['First_Name']  
$dataArray['data']['12.3']['First_Name'][0]
```

Look at the sample code to get a better feel for how things work. You can also see it at work on my site:

www.iviking.org/FX/

Another resource for becoming familiar with the way that FX.php returns data is the excellent fxparser included with the release. This tool was created by Lance Hallberg and is highly recommended. Note that you'll find documentation, etc. for this tool inside the fxparser directory of the FX distribution. Some of the syntax is a bit dated, but it will still work just fine.

Of course, to take advantage of this class, you'll need to install PHP. PHP is available for a number of platforms. The source code for PHP and links to Windows binaries and information on other distributions are located on this page:

<http://www.php.net/downloads.php>

PHP is included with recent releases of Mac OS X. More information and additional Mac OS X binaries are available here (this site will also tell you how to configure PHP on Mac OS X):

<http://www.entropy.ch/software/macosex/php/>

If you'd like to run PHP in the MacOS classic environment, you'll need to install WebTen from Tenon Intersystems:

www.tenon.com/products/webten/

Finally, PHP runs just fine on Windows servers. FX.php has been tested running with PHP installs on both Microsoft IIS and Apache running on Windows.

IMPORTANT: In my experience running this combination on old versions of the Mac OS (up to and including under Mac OS X 10.0.? -- this has been **fixed** since version 10.1.2), it is important that you not attempt to access FileMaker Pro and your web server over the same network interface. (i.e. you'll either need a second machine, or a second network card.) On my old PowerMac G3 Server running Mac OS X, I got a kernel panic when I attempted to use a single interface. Each interface should, of course, have it's own IP address.

There are example databases and several example PHP pages included with this README.

Configuring FileMaker 7/8/9 Server or Server Advanced is beyond the scope of a simple README, but the main thing to keep in mind in order to use FX.php with a database is to ensure that the account that will be used for web access has 'fmxml' extended privileges. Please see the documentation for the version that you're using for more details.

In order to use these files under FileMaker 5/6 do the following:

- 1) Open the 'Book_List.fp5' database (and Tester.fp5, if desired) in FileMaker Pro.
- 2) Make sure that the Web Companion is active under plug-ins in FileMaker's application preferences (FileMaker has reserved port 591 and FX uses that port by default, to set the port used by the Web Companion, select Web Companion and click the configure button in the plug-ins preference screen.)
- 3) Make sure that Web Companion sharing is enabled for the example database.

In either case, here's how to set up and access the FX.php demo files:

- 1) Place the 'FX' folder at the root level of your web server.
- 2) Make sure that all users (or at least the web user) have read access to the 'FX' directory.
- 3) Enter the IP address of your FileMaker Pro machine in server_data.php after the equals sign in this expression: \$serverIP = ; Make sure that the line ends with a semi-colon.
- 4a) For FileMaker 7, if the web server where FM7SA is installed is listening on a port other than 80, you'll need to change the value of \$webCompanionPort to that port.
- 4b) On FileMaker 5/6 Unlimited, if you configured the Web Companion to use a port other than 591, you'll need to change the value for \$webCompanionPort accordingly.
- 4) You should now be able to access the files from your browser like this:

`http://your.web.site/FX/` (for the book list demo)
`http://your.web.site/FX/Tutorials/FXExamples.php` (usable with most FM DBs)

Adding the functionality of FX.php to your own PHP files is as simple as using an 'include' statement in a PHP file. Although there are multiple functions that can be used to accomplish this, I would suggest using 'include_once()' or 'require_once()', since PHP will throw up an error if there are multiple declarations of a single class.

Also, there are a number of things that may be helpful to know about FX.php itself (look at the example files to see these at work):

- 1) First, declare an instance of the FX class like this (substituting the IP address of your FileMaker machine):

```
$SomeQuery = new FX("172.0.0.1");
```

- 2) Specify database and layout details:

```
$SomeQuery->SetDBData("DatabaseName", "LayoutName");
```

- 3) Specify the parameters that you wish to pass to the database using one line like this for each field:

```
$SomeQuery->AddDBParam("FieldName", "FieldValue");
```

- 4) Specify sort fields and sort orders if you desire (there are ways that PHP can sort your results for you, too):

```
$SomeQuery->AddSortParam("SortFieldName", "descending");
```

5) Submit the query and store the result using the following (passing any ONE of: delete, duplicate, update, perform_find, show_all, show_any, new as a string):

```
$ReturnedData = $SomeQuery->DoFXAction('perform_find'); // find a record
```

There are other commands, too, but these should get you started. Look through FX.php. The actions that FX can perform are grouped towards the bottom of the file. Also, be sure to make use of the functions reference included with the file, as it goes through the possible parameters for each function. Feel free to email me any questions that you may have:

FX@iviking.org

Finally, there is no way that I could completely cover all of the possibilities that PHP opens to you. Fortunately, there is a wealth of information out there. PHP's on-line manual is excellent (I don't feel that way about any other on-line documentation that I've seen.) It can be found at:

www.php.net/manual/en/

This is the URL of the English Language version of the manual. Other languages are linked from the site, too. There are also a number of books on PHP. I've found these two very helpful:

PHP Pocket Reference by Rasmus Lerdorf (the creator of PHP) and published by O'Reilly

PHP4 Bible by Tim Converse and Joyce Park, pulished by IDG

But don't just take my word for the books you get; spend some time in your local bookstore comparing books and find one that makes sense to you. I would suggest that you take a good look at the index of each book (you're liable to refer to it frequently.)

I hope all of this is of some help. Good luck and happy programming!

--Chris Hansen