

IPOST



FileMaker Web Plug-in

User Guide



<http://www.fmwebschool.com> 800.353.7950 / 386.453.5843

IPOST Plug-in is copyright 2005-2006 of FMWebschool Inc.

Thank you for your interest in the IPOST plug-in for FileMaker versions 5-8. We initially designed this plug-in for use with our FileMaker POS system. As we worked with IPOST we realized that there were a tremendous amount of uses for this plug-in. We began using IPOST internally to download music, fill out online forms, grab daily stock reports, weather reports, images and more.

We decided to market IPOST after we realized how useful it is within our own company. We have used IPOST as a tracking system to make sure users register our software. We have used IPOST to enable users to log into http/https websites that require username and passwords retrieve the pages and load the data into FileMaker.

We are providing 12 sample files to demonstrate some of the ways you can use IPOST.

Enjoy!

The team at FMWebschool

Installing the IPOST Demo for FileMaker 7/8 Windows Version

Note: Please Exit FileMaker before installing the IPOST plug-in.

The IPOST plug-in is located inside the "IPOST" folder.



When using FileMaker 7 and 8, simply move this plug-in into the FileMaker "Extensions" folder. Here are the paths to those folders.

FileMaker 8

C:\Program Files\FileMaker\FileMaker Pro 7\Extensions

FileMaker 7

C:\Program Files\FileMaker\FileMaker Pro 7\Extensions

For FileMaker 5 and 6, you will need to place the plug-in into the "Systems" folder

FileMaker 6

C:\Program Files\FileMaker\FileMaker Pro 6\System

FileMaker 5

C:\Program Files\FileMaker\FileMaker Pro 5\System

Once you have placed the plug-in in the appropriate folder, you are ready to begin using the Example files.

Installing the IPOST Demo for FileMaker 7/8 Macintosh Version

Note: Please Exit FileMaker before installing the IPOST plug-in.

The IPOST plug-in is located inside the “IPOST” folder.



When using FileMaker 7 and 8, simply move this plug-in into the FileMaker “Extensions” folder. Here are the paths to those folders.

FileMaker 8

C:\Applications\FileMaker Pro 7\Extensions

FileMaker 7

C:\Applications\FileMaker Pro 7\Extensions

For FileMaker 5 and 6, you will need to place the plug-in into the “Systems” folder

FileMaker 6

C:\Applications\FileMaker Pro 6\Extensions

FileMaker 5

C:\Applications\FileMaker Pro 5\Extensions

Once you have placed the plug-in in the appropriate folder, you are ready to begin using the Example files.

Very Important! Before you begin working with the example files, please read the following comments below!

Macintosh users:

For the demo Macintosh files, the “Path” field will use the hard-drive name with a colon, then the name of the file. In the example files, my “hard-drive name is “snoopy”. So if I were saving a file named “tommy.gif” to my hard-drive, my path would look like this.

Ex.

Snoopy:tommy.gif

Window users:

For the demo Windows files, the “Path” will simply use C:\ followed by the name of the file.

Ex.

C:\tommy.gif

Note: For the “**Post From File example**”, the plug-in will look for a file stored on your “**C**” drive. We have included a text document for this example. Inside the IPOST folder you will find a file entitled “**sample.txt**” simply place this folder on to your “**C**” drive.

Ex.

C:\ sample.txt.

Currently, IPOST enables you to post and retrieve information from the Internet by using POST and GET commands; you can access HTTP and HTTPS pages. We have added redirection and file manipulation. To see examples of file manipulation and redirection, please make sure you take a look at the bonus examples.

IPOST Plug-in Functions and Examples

Plug-in Functions

"IPOST_Version"
"IPOST_Enable"
"IPOST_PostDataFromField"
"IPOST_PostDataFromFile"
"IPOST_PostDataFromFieldUsingGet"
"IPOST_PostDataIntoFileUsingGet"
"IPOST_PostDataIntoFileUsingGetShowProgress"

"IPOST_Version"

Parameters: NONE

Result: The version of the IPOST plug-in. Currently, the value returned is "IPOST v1.48"

Errors: None

Use this function to make sure the IPOST plug-in is installed and to make sure the version is as expected.

Example:

```
Set Field[mysolution:gPluginReply;""]
Set Field[mysolution:gPluginReply;External("IPOST_Version";"")]
If[GetAsNumber (mysolution:gPluginReply)<1.38]
    If [mysolution:gPluginReply = ""]
        //Do something because the plug-in is not even installed
    End If
    //Do something if you care that a lower version is installed
End If
```

"IPOST_Enable"

Parameters: Username|Password

Result: "REGISTERED", "REGISTERED DEMO", "INVALID"

Errors: "\$\$ErrorParameterSize",

Use this function in a start up script to "unlock" the IPOST plug-in. This function only needs to be called once, however the plug-in will become "locked" again once the FileMaker application is quit.

Example to use as demo:

```
Set Field[mysolution:gPluginReply;External("IPOST_Enable"; "Demo|Demo")]
```

"IPOST_PostDataIntoFileUsingGet"

(FileMaker Example files 1,2,4,5,6,7,9 demonstrate this function.)

Parameters: URL|Data|filepath

Result: Data returned from URL

Errors: "\$\$There was a problem with the URL given", "\$\$out of memory", "\$\$Session error.", "\$\$Server error.", "\$\$Connection error.", "\$\$Server request error.", "\$\$Header error.", "\$\$Send request error.", "\$\$The URL did not reply.", "\$\$Write data error.", "\$\$File error.", "\$\$Write error.",

"\$\$The server responded with a status code of ###"

Use this function to retrieve data from the web using the "GET" method. Good for downloading any binary data.

Example or retrieving an image based on data sent:

Set

```
Field[mysolution:gPluginReply;External("IPOST_PostDataIntoFileUsingGet"  
; "http://www.marketwatch.com/charts/int-basic.chart" & "|" &  
"symp=AAPL&time=1&freq=9&size=1&style=1013" & "|" & "C:\~APPL.gif")]
```

Result:

"GIF" & "5 or 6 additional characters"

Different file types will reply with ("ÿØÿà" = JPEG), ("RIFF" & some extra characters = .wav), ("PK" & some extra characters = .ZIP) etc..

The downloaded file can be found in the path specified. In this case the resulting file is C:\~APPL.gif

Example or retrieving a static image

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataIntoFileUsingGet"  
;"http://www.blogography.com/photos4/Stewie.gif" & "|" & "" & "|" &  
"C:\\~Stewie.gif")]
```

Result:

"GIF" & "5 or 6 additional characters"

Different file types will reply with ("ÿØÿà" = JPEG), ("RIFF" & some extra characters = .wav),

("PK" & some extra characters = .ZIP) etc..

The downloaded file can be found in the path specified. In this case the resulting file is

C:\\~Stewie.gif.

Stewie hates his broccoli

"IPOST_PostDataFromField"

(FileMaker Example file 10 demonstrate this function.)

Parameters: URL|Data

Result: Data returned from URL

Errors:"\$\$There was a problem with the URL given","\$\$out of memory","\$\$Session
error.,"\$\$Server error.,"\$\$Connection error.,"\$\$Server request error.,"\$\$Header
error.,"\$\$Send request error.,"\$\$The URL did not reply.,"\$\$Write data error.,"\$\$Write
error.,"\$\$The server responded with a status code of ###"

Use this function to send data to a webform from a field (or hard coded script) using the "POST"
method.

Example:

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataFromField";"http:  
//www.backofficeware.com/formsubmit.asp " & "|" &  
"MessageType=Praise&Subject=%28Other%29&SubjectOther=IPOST+Plug-in  
&Comments=Please+reply+to +this+ post.  
&Username=Anonymous&UserEmail=somebody@somewhere.com&UserTel=6095551212  
&UserFAX=n%2Fa")]
```

Reply:

```
"<p>MessageType is Praise</p><p>Subject is (Other)</p><p>SubjectOther  
is IPOST Plug-in</p><p>Comments is Please reply to this  
post.</p><p>Username is Anonymous</p><p>UserEmail is  
somebody@somewhere.com</p><p>UserTel is 6095551212</p><p>UserFAX is  
n/a</p><p>:D</p>"
```

"IPOST_PostDataFromFile"

(FileMaker Example file 10 demonstrate this function.)

Parameters: URL|filepath

Result: Data returned from URL

Errors:"\$\$There was a problem with the URL given","\$\$out of memory","\$\$Session error.,"\$\$Server error.,"\$\$Connection error.,"\$\$Server request error.,"\$\$Header error.,"\$\$Send request error.,"\$\$The URL did not reply.,"\$\$Write data error.,"\$\$File error.,"\$\$Write error.,"
"\$\$The server responded with a status code of ###"

Use this function to send data to a webform from a text file using the "POST" method.

NOTE: always use \"' when using this function.

Example using a file named "sample.txt" saved on the "C:" drive that contains the same data as the "IPOST_PostDataFromField" example above:

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataFromFile";"http://  
/www.backofficeware.com/formsubmit.asp " & "|" & "C:\\sample.txt")]
```

Reply

(if sample.txt contains same string as used in the
IPOST_PostDataFromField example) :
"<p>MessageType is Praise</p><p>Subject is (Other)</p><p>SubjectOther
is IPOST Plug-in</p><p>Comments is Please reply to this
post.</p><p>Username is Anonymous</p><p>UserEmail is
somebody@somewhere.com</p><p>UserTel is 6095551212</p><p>UserFAX is
n/a</p><p>:D</p>"

"IPOST_PostDataFromFieldUsingGet"

(FileMaker Example files 8 & 3 demonstrate this function.)

Parameters: URL|Data

Result: Data returned from URL

Errors:"\$\$There was a problem with the URL given","\$\$out of memory","\$\$Session error.,"\$\$Server error.,"\$\$Connection error.,"\$\$Server request error.,"\$\$Header error.,"\$\$Send request error.,"\$\$The URL did not reply.,"\$\$Write data error.,"\$\$Write error.,"\$\$The server responded with a status code of ###"

Use this function to send data to a webform from a field (or hard coded script) using the "GET" method. Use this to retrieve webpage sources and online text files.

Online text file example:

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataFromFieldUsingGet  
";"http://www.backofficeware.com/hello.txt " & "|")]  
Reply:
```

"Hello IPOST users."

Online page source example:

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataFromFieldUsingGet  
";"http://www.fmwebschool.com/aboutus.php" & "|")]
```

NOTE: There is no data sent in the above two examples, just the url.

Online search query example (search for FileMaker on Yahoo.com):

```
Set  
Field[mysolution:gPluginReply;External("IPOST_PostDataFromFieldUsingGet  
";"http://search.yahoo.com/search" & "|" &  
"p=FileMaker&sm=Yahoo%21+Search&fr=FP-tab-web-t&toggle=1&cop=&ei=UTF-  
8")]
```

NOTE: There is no '?'

"IPOST_PostDataIntoFileUsingGetShowProgress"

Parameters: URL|Data|filepath

Result: Data returned from URL

Errors: "\$\$There was a problem with the URL given", "\$\$out of memory", "\$\$Session error.", "\$\$Server error.", "\$\$Connection error.", "\$\$Server request error.", "\$\$Header error.", "\$\$Send request error.", "\$\$The URL did not reply.", "\$\$Write data error.", "\$\$File error.", "\$\$Write error.",
"\$\$\$The server responded with a status code of ###"

Use this function to retrieve large sets of data from the web using the "GET" method. This function displays a progress bar at the bottom of the FMP Application window

Example or retrieving a movie from FMWebSchool

```
Field[mysolution:gPluginReply;External("IPOST_PostDataIntoFileUsingGetS  
howProgress" "http://www.fmwebschool.com/IP_demo/ipostdemo.mov"& "|" &  
"|" & "C:\IPOSTMOV.mov")]
```

Result:

""

The downloaded file can be found in the path specified. In this case the resulting file is C:\IPOSTMOV.mov

Note: .MOV does not return any value into the reply field. In this case, no reply can be considered success because we know we are downloading a .mov file and the IPOST plug-in would reply with an error if there was one to report.

Try:

```
Field[mysolution:gPluginReply;External("IPOST_PostDataIntoFileUsingGetS  
howProgress" "http://www.fmwebschool.com/IP_demo/ipostdemo2.mov"& "|" &  
"|" & "C:\IPOSTMOV.mov")]
```

Result:

\$\$The server responded with a status code of 404."

Working with Web Forms

You can almost bet that when you come to a website that contains a "submit" button there is going to be a web form. Luckily, you can use IPOST to send information to the response page. If you have never worked with HTML and you are not familiar with the HTML language, this can be somewhat challenging.

You see, in order to send data to the web page, you are going to need to know how to find the <FORM> in the source code, the input lines, information being submitted in the form and the url of the response page.

Let's go step-by-step. First you will need to open the webpage in your browser. Once the page is open, select "view source". The web page will open as a text document. Go ahead and save that document. Make sure to pay attention as to where you are saving the file.

Next, open up the text document. Below is an example of how the page might look.

Here is an example.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<form method="post" action="response_page.php">
<input type="text" name="first_name">
<input type="text" name="last_name">
<input type="text" name="phone">
<select name="state">
<option value="CA">CA</option>
<option value="CA">FL</option>
<option value="CA">NC</option>
</select>
<input type="submit" name="submit" value="Submit Form">
</form>

</body>
</html>
```

Now that you have the page open we are ready to begin the process of finding the code we need.

You will need to locate the beginning and the ending of the **form**. This is pretty simple. The form begins with **<form>** and ends with **</form>** many web sites have multiple forms on their pages, so you will need to make sure you are selecting the correct form.

Below, I have selected my form. I have deleted all of the other text on the page, so I am just working with what I need.

```
<form method="post" action="response_page.php">
<input type="text" name="first_name">
<input type="text" name="last_name">
<input type="text" name="phone">
<select name="state">
<option value="CA">CA</option>
<option value="CA">FL</option>
<option value="CA">NC</option>
</select>
<input type="submit" name="submit" value="Submit Form">
</form>
```

This form includes 3 text fields and 1 value list with three options.

```
<input type="text" name="first_name">
```

In this case, the key is 'first_name'.

There are two other text keys in the form. They are 'last_name' and 'phone'.

The value list is structured differently than the text boxes, but it still contains a 'name=' which denotes the key. The key for the value list is 'state'.

The key value pairs can be combined by placing an ampersand (&) between the pairs.

Ex.

```
first_name=Stephen&last_name=Knight&phone=8003537950&state=FL
```

As an example, this string used with the PostDataFromField function would look like this.

```
Set Field[mysolution:gPluginReply;External("IPOST_PostDataFromField";
"http://www.backofficeware.com/formsubmit.asp " & "|" &
"first_name=Stephen&last_name=Knight&phone=8003537950&state=FL ")]
```

This script will post data to a web page.

Getting Help

If you have questions about IPOST, please email FMWebschool at info@fmwebschool.com You can also ask questions on our forum at: <http://fmwebschool.com/frm/index.php> If you do have questions, or are experiencing problems with the plug-in, please be as specific as possible.