

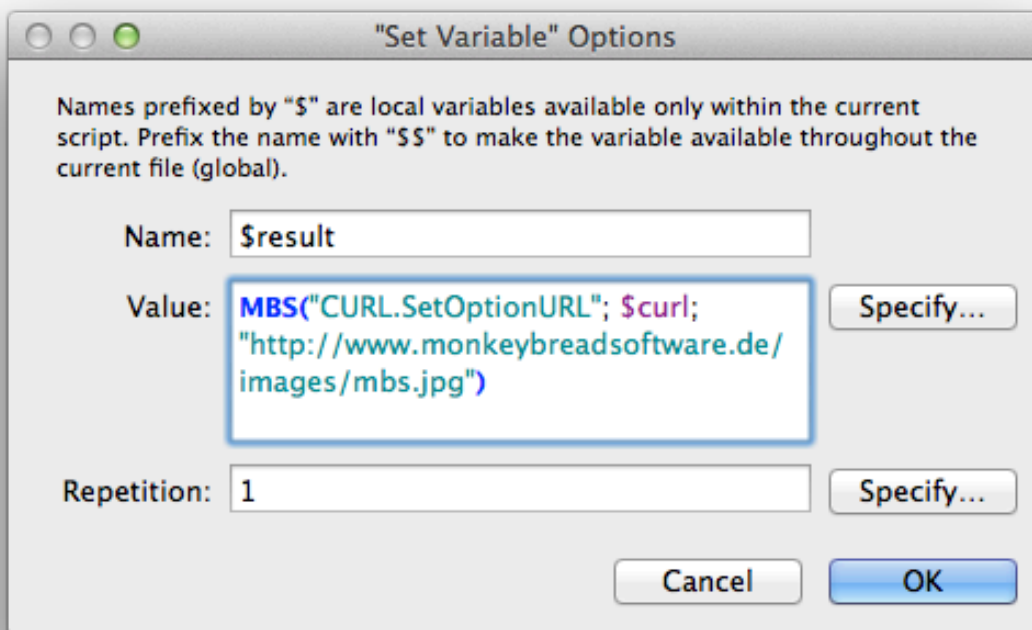
CURL Tutorial for Filemaker with MBS Plugin

Today we want to show you how to use our [CURL functions](#) in the MBS Filemaker Plugin to do some simple tasks. First we download a picture file and then we upload a file with ftp.

Part 1: Download

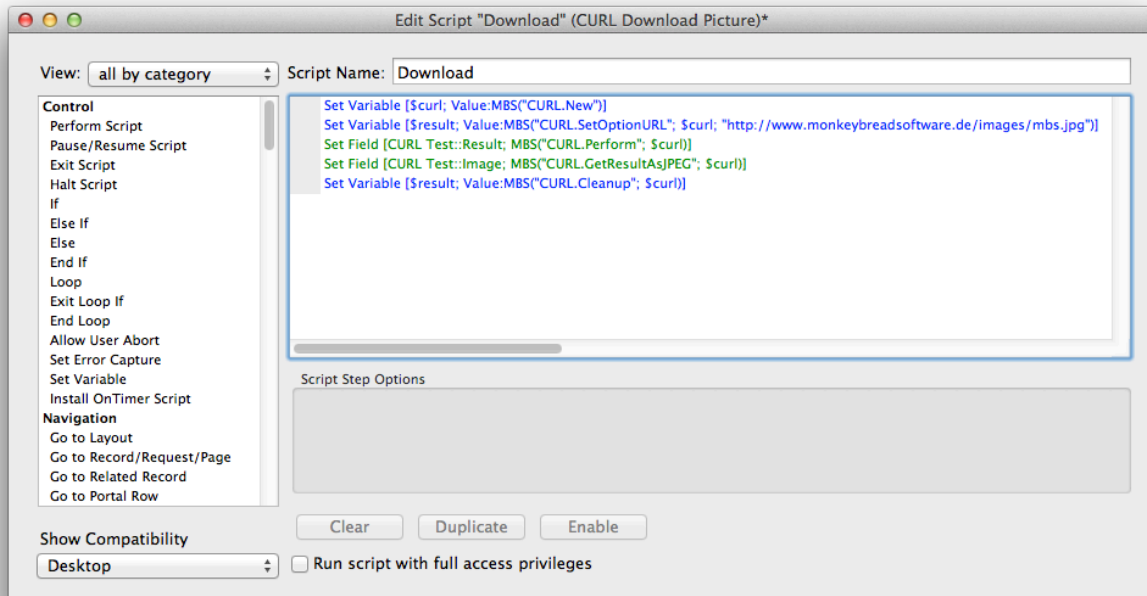
To download a picture from a web server, we only need 5 script steps. First script step is to call [CURL.New](#) to start a new session for the transfer. This function gives back a handle value which is simply the number assigned to the transfer and you need it later to reference the transfer. This way the plugin can handle several transfers in parallel. Like most plugin calls you can simply put this function into a Set Variable script step and assign the result to a variable (\$curl in our sample) for later use.

Next we need to define the target URL. For that we use the function [CURL.SetOptionURL](#) and pass both the reference value and the target URL as text. The call looks like this:



To start the transfer we call [CURL.Perform](#) with the \$curl variable again. If everything goes right, we get back error code zero for no error. Then

we can call [CURL.GetResultAsJPEG](#) to get the jpeg image as a container value to be assigned to a container field. As the last step we call [CURL.Cleanup](#) to end the session. As you see we always pass \$curl, the reference value from the first call to CURL.New each time. The complete script looks like this:



When you try it, you'll notice that the plugin blocks Filemaker for a second to do the transfer. You can read about [CURL.PerformInBackground](#) if you like to keep the download running in background. Also the MBS Plugin has functions to show a [progress bar](#) while downloading.

Part 2: Upload

The upload example will be a little bit longer as we use more plugin functions and we want to have debug output for seeing what the problems are. Especially with FTP uploads it is good to see the log as the FTP servers may respond with some error like an unknown user account or wrong password.

We start with [CURL.New](#) again to start a new curl session. Then we set the URL with [CURL.SetOptionURL](#) for the upload. In our example this URL looks like this: "ftp://monkeybreadsoftware.de/folder/test.jpg". As you see we pass the complete URL to the file we want to upload including the path.

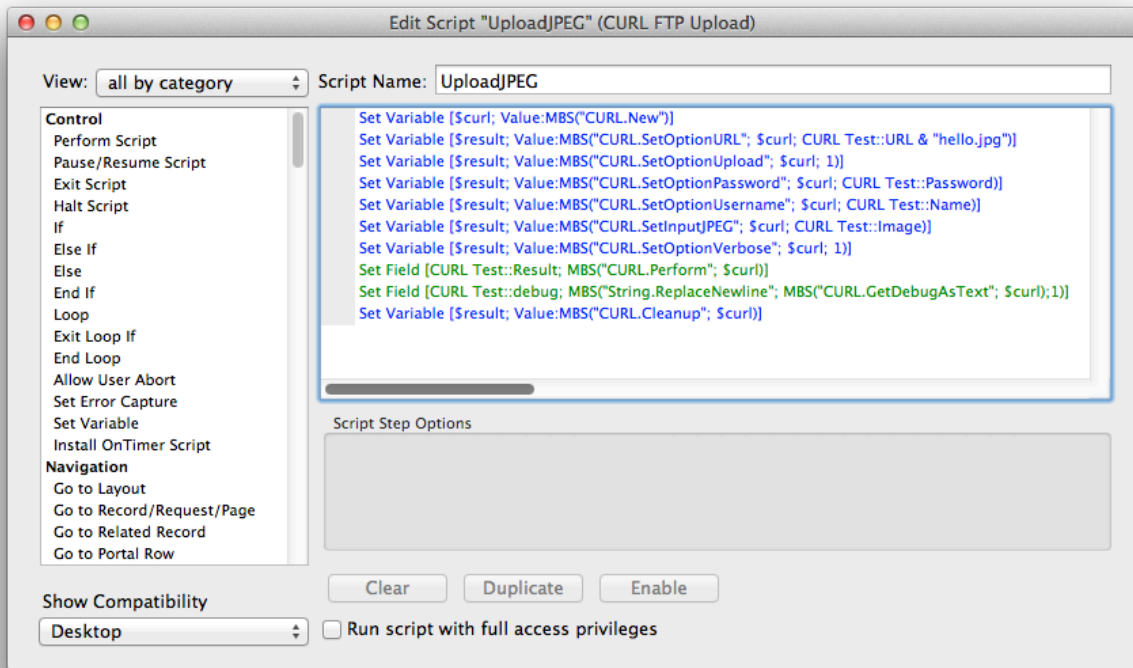
Next we set credentials. For that we call [CURL.SetOptionPassword](#) and [CURL.SetOptionUsername](#) passing the username and password. Please do not include the password or username in the URL. While you can do that, you get into trouble when the name includes a @ character.

As this is an upload, we call [CURL.SetOptionUpload](#) and pass 1 as value to enable upload. For the file we simply pass a JPEG file we have inserted in a container. So we call [CURL.SetInputJPEG](#) and pass the container field. The plugin then picks the JPEG data from the field value and puts it in the input buffer.

Before starting the transfer we enable debug messages. We call [CURL.SetOptionVerbose](#) which gives more details on the transfer. The plugin collects all messages from CURL or the server, so we can later read them.

Now we can run the transfer and call [CURL.Perform](#). After the transfer is done, we call [CURL.GetDebugAsText](#) to get back the collected debug messages and we store them in a field. There you can now read the whole FTP transcript and see error details in case of a problem. Finally we call [CURL.Cleanup](#) to end our session.

The complete script:



The log actually looks like this:

```
About to connect() to macsw.de port 21 (#0)
Trying 93.187.234.106... connected
Connected to macsw.de (93.187.234.106) port 21 (#0)
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 50 allowed.
220-Local time is now 15:26. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 10 minutes of inactivity.
USER xxxxxx
331 User 12144-webcam OK. Password required
PASS xxxxxx
230 OK. Current restricted directory is /
PWD
257 "/" is your current location
Entry path is '/'
EPSV
Connect data stream passively
229 Extended Passive mode OK (|||50049|)
Trying 93.187.234.106... connected
Connecting to 93.187.234.106 (93.187.234.106) port 50049
TYPE I
200 TYPE is now 8-bit binary
STOR hello.jpg
150 Accepted data connection
Remembering we are in dir ""
226-File successfully transferred
226 0.438 seconds (measured here), 80.03 Kbytes per second
Connection #0 to host macsw.de left intact
```

If you have questions, please do not hesitate to contact us.