CURL Custom Function

Recently a client suggested to make a custom function to call CURL functions anywhere with just a function call. But instead of building a convenience function into the plugin, we provide you the recipe on how to make such a custom function, so you can change it.

Our function will do GET and POST with text in and out. That is the most requested case for various REST or SOAP web services. You may also add variations to return result as container to taking a container for upload or a file path as in- or output. So here is our easy script:

CURLExecute(URL, Post Text, HTTP Headers)

```
Let([
curl = MBS("CURL.New");
// put the URL in for this request
r = MBS("CURL.SetOptionURL"; curl; URL);
// and some headers if needed
r = MBS( "CURL.SetOptionHTTPHeader"; curl; HTTP Headers );
// for POST, we use post text. Keep it empty if you do GET.
r = If( Length(Post Text) = 0; ""; MBS("CURL.SetOptionPostFields"; curl;
Post Text));
// run the request
r = MBS("CURL.Perform"; curl);
// we return text as result
result = MBS("CURL.GetResultAsText"; curl; "UTF8");
// and put debug log in global variable
$$debug = MBS("CURL.GetDebugAsText"; curl);
// free the object
r = MBS("CURL.Release"; curl)];
result)
```

As you see we setup a CURL session, set the URL and headers with optional POST fields data. Then we run the request and return back the text. We store the debug messages in a $$debug global variable.

You may do variants of this to do various other things. Like swap out the CURL.GetResultAsText function with the CURL.GetResultAsContainer to get the result as a container:

```
result = MBS("CURL.GetResultAsContainer"; curl);
```

Now let's build an FTP upload function, which also does FTPS and SFTP if needed, which is more secure for you. In general we prefer SFTP over FTPS since it only needs one port and not two like FTP.

CURLExecute FTP Upload(URL, Data, UserName, Password)

```
Let([
curl = MBS("CURL.New");
// put the URL in for this request
r = MBS("CURL.SetOptionURL"; curl; URL);
// open input file or container
r = If(MBS( "FM.DataType"; Data ) = "container";
MBS( "CURL.SetInputFile"; curl; Data ); // upload container
MBS( "CURL.OpenInputFile"; curl; Data )); // open file on disk
// Set user name and password
r = MBS("CURL.SetOptionUserName"; curl; User Name);
r = MBS("CURL.SetOptionPassword"; curl; Password);
// don't wait a minute to connect
r = MBS("CURL.SetOptionConnectionTimeout"; curl; 10);
// set SFTP to use user name and not a private key
r = MBS( "CURL.SetOptionSSHAuthTypes"; curl; 2+8 );
// we want an upload
r = MBS("CURL.SetOptionUpload"; curl; 1);
// run the request
result = MBS("CURL.Perform"; curl);
// and put debug log in global variable
$$debug = MBS("CURL.GetDebugAsText"; curl);
// free the object
r = MBS("CURL.Release"; curl)];
result)
```

As you see, we use a few little tricks: First we check with FM.DataType what type of value you passed. This may be a container or text with a native file path, so we can call the correct MBS function to either open the container as an input or the file on disk. Then we set username and password for the upload. To avoid endless wait time, we set connection timeout to 10 seconds so you don't wait forever, if the server doesn't respond. And for SFTP we disable the private key login, so user name and password is used. Result is the result from CURL.Perform in this case.

Please try these custom functions. They'll be included in a sample database with the next plugin download. And please don't hesitate to contact us with your questions.