

Sending Email with FileMaker via MBS Plugin

While FileMaker has a built in script step to send email, it's fairly limited. You can specify only one attachment and just plain text for the email content.

As an alternative we want to show you how to use MBS Plugin functions to build and email and send it through your SMTP server via CURL functions. The MBS Plugin supports multiple recipients, multiple attachments and email content can be provided as html, rich or plain text. No worries about text encoding as the MBS Plugin encodes the email content as UTF-8, so all the umlauts and smileys survive.



Build Email

To start, you call the plugin function [SendMail.CreateEmail](#), which returns an identifier for the current email we build. This identifier (a text) is then passed to other functions in the next steps to identify the email. By using identifiers for all emails, you can have several scripts use the email functions and each has its own email.

To set the from address, you call the function [SendMail.SetFrom](#) with your email address. You can also include the name for display. The email is used as is, but the name is encoded to preserve any special characters.

To specify the subject, you call the function [SendMail.SetSubject](#). Text is encoded, so even smileys survive here.

Content

To define the content of your email, please call at least one of those three functions: [SendMail.SetPlainText](#), [SendMail.SetHTMLText](#) or [SendMail.SetRichText](#). Plain text is for simply text to send. This allows small emails and also helps if the receiver automatically scans emails. As the MBS Plugin can also download emails via POP3 or IMAP, you can build a robot which downloads emails from email account to process them. If you pass rich text for the email via [SendMail.SetRichText](#) function, please pass RTF data. If you need to build RTF text, you can use the plugin function [Text.TextToRTF](#). For HTML, you use [SendMail.SetHTMLText](#) function and pass HTML. To get HTML you can for example use FileMaker's [GetAsCSS](#) function or the plugin function

[Text.TextToHTML](#). HTML, Rich and plain text can be mixed. In that case the email app will choose one to display.

For sending newsletter and other emails to clients, you may want to build templates with placeholders. Then you can easily replace those place holders with the real values. For encoding values from fields for inserting into HTML template, please use [String.EncodeToHTML](#) function to escape special characters.

Recipients

You can add recipients to the email: To, CC, BCC and Reply-To. Either you call dedicated functions [SendMail.AddTo](#), [SendMail.AddReplyTo](#), [SendMail.AddCC](#), [SendMail.AddBCC](#) or you use the more general [SendMail.AddRecipient](#) function. In that case you pass the type of address as parameter. In all cases, please provide email and optional the name of the recipient.

Other Header values

If you like, you can add some special header values. For example [SendMail.SetInReplyTo](#) can be used to specify which email you reply to. For that you pass a message ID.

For new emails, we generate a message ID automatically when you set from email address. Using the function [SendMail.SetMessageID](#), you can specify your own if needed.

For other header values, please use the [SendMail.AddHeader](#) function. For example you can set a company name via `MBS("SendMail.AddHeader"; $EmailID; "X-Company: Your Company Name")` or a high priority using `MBS("SendMail.AddHeader"; $EmailID; "X-Priority: 1")`.

Attachments

If you like to add an attachment, please call [SendMail.AddAttachmentContainer](#) or [SendMail.AddAttachmentFile](#). For first function you pass a container as parameter with the file to upload. This can be anything like image, video, audio, PDF or other documents. For the second variant, please pass a native file path, e.g. `"/Users/test/Desktop/test.pdf"`. For all attachments, you can pass a file name and a mime type. If you pass no mime type, the plugin defaults to `"application/octet-stream"`. If you pass `"image/jpeg"` for a JPEG image or `"application/pdf"` for a PDF file, the email client can display the attachment.

Server

To specify which server to use, please call [SendMail.SetSMTPServer](#) function and pass server domain name, e.g. smtp.mac.com. If you like to use SSL, please pass second parameter with value 1. Next please call [SendMail.SetSMTPUsername](#) to pass user name for login. Call [SendMail.SetSMTPPassword](#) to provide the password. Our plugin does never store those passwords. It just keeps them in memory for login process later.

CURL

For the final sending, we pass the email to CURL. So first you call [CURL.New](#) and get back a new session identifier for the new CURL session. Next you call [SendMail.PrepareCURL](#) function to pass the email to the CURL session. Now the plugin setups the transfer for the email with content of email, server details and other options.

If you like you can set more CURL options and then call [CURL.Perform](#) function to send email. After the transfer, you can use [CURL.GetDebugAsText](#) to get debug messages. So if [CURL.Perform](#) returns an error, you can lookup there details and maybe see what caused a problem.

Cleanup

When you are done with email sending, please cleanup. So you call both [CURL.Cleanup](#) and [SendMail.Release](#) to release the CURL session and the email. Of course you can reuse both for sending another email.

SSL/TLS

Of course it is easy to send email unencrypted to server. But sometimes we want a little bit privacy, so we turn on SSL when calling [SendMail.SetSMTPServer](#) function. If your SMTP server has a non standard port, please use [CURL.SetOptionPort](#) and pass the right port, e.g. 587.

For TLS we have to connect without SSL and later upgrade to SSL on request. For that please pass 0 on [SendMail.SetSMTPServer](#) for SSL option. Now also call [CURL.SetOptionFTPSSL](#) with value 3 to allow to use all SSL variants, so the email server can define which SSL variation he knows and the plugin can upgrade to the matching SSL version.

For best security, please download certificate from CURL website. Now use [CURL.SetOptionCAInfo](#) to pass the file path to this file. If you now pass 2 for [CURL.SetOptionSSLVerifyHost](#) and 1 for [CURL.SetOptionSSLVerifyPeer](#). Now you should get an error if SSL verification doesn't work and someone tries a man in the middle attack on you.

Example

Be sure to check the example database included with the plugin. It uses two sub tables for the recipients and attachments and loops over those in the script to build the email. Good luck!