

Sending email with a huge custom function

A client surprised me by presenting me a big calculation to send emails:

SMail(htmlText, EmRecipient, FromEmail, FromName, plainText, Subject, SMTPServer, SMTPName, SMTPpw, EmailID, Send)

Let ([

```
$ToEmail = MBS ( "FM.executesql" ; "SELECT \"Email\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'TO' " ) ;  
$ToName = MBS ( "FM.executesql" ; "SELECT \"Name\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'TO' " ) ;  
$ToEmailcc = MBS ( "FM.executesql" ; "SELECT \"Email\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'CC' " ) ;  
$ToNamecc = MBS ( "FM.executesql" ; "SELECT \"Name\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'CC' " ) ;  
$ToEmailbcc = MBS ( "FM.executesql" ; "SELECT \"Email\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'BCC' " ) ;  
$ToNamebcc = MBS ( "FM.executesql" ; "SELECT \"Name\" FROM \"RM\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Type\" = 'BCC' " ) ;  
$AttCont = MBS ( "FM.sql.execute" ; "" ; "SELECT \"Container\" FROM  
\"AT\" WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NULL " ) ;  
AttCount = MBS( "FM.SQL.RowCount" ; $AttCont ) ;  
$AttPath = MBS ( "FM.executesql" ; "SELECT \"Path\" FROM \"AT\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NOT NULL " ) ;  
AttCountP = ValueCount ( $AttPath ) ;  
$AttNameC = MBS ( "FM.executesql" ; "SELECT \"Name\" FROM \"AT\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NULL " ) ;  
$AttTypeC = MBS ( "FM.executesql" ; "SELECT \"Type\" FROM \"AT\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NULL " ) ;  
$AttNameP = MBS ( "FM.executesql" ; "SELECT \"Name\" FROM \"AT\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NOT NULL " ) ;  
$AttTypeP = MBS ( "FM.executesql" ; "SELECT \"Type\" FROM \"AT\"  
WHERE \"EmailID\" = \" & EmailID & \" AND \"Path\" IS NOT NULL " ) ;
```

```
// htmltext = EM::HTMLText ; EmRecipient = EM::Recipient ; FromEmail =  
EM::FromEmail ; FromName = EM::FromName ; plaintext = EM::PlainText ;  
Subject = EM::Subject ; SMTPServer = EM::SMTP Server ; SMTPName =  
EM::SMTP Username ; SMTPpw = EM::SMTP Password ; eid = EmailID ;
```

```
// Run in file Build and send HTML Email with inline graphic
```

```
html = htmlText ; // Build HTML by replacing placeholder with name of  
recipient
```

```

name = MBS("Text.EncodeToHTML"; EMRecipient) ;
html = Substitute(html; "%name%"; name) ;
$EmailID = MBS("SendMail.CreateEmail") ;// Build Email
r = MBS("SendMail.SetFrom"; $EmailID; FromEmail; FromName) ;
r = MBS("SendMail.SetHTMLText"; $EmailID; html) ;
r = MBS("SendMail.SetPlainText"; $EmailID; PlainText) ;
r = MBS("SendMail.SetSubject"; $EmailID; Subject) ;
r = MBS("SendMail.SetSMTPServer"; $EmailID; SMTPServer) ;
r = MBS("SendMail.SetSMTPUsername"; $EmailID; SMTPName) ;
r = MBS("SendMail.SetSMTPPassword"; $EmailID; SMTPpw ) ;// Add
Recipients
re = MBS( "FM.Loop"; "n"; 1 ; ValueCount ( $ToEmail); 1; "let ( [ a1 =
getvalue ( $ToEmail ; n ) ; a2 = getvalue ( $ToName ; n ) ] ; if ( not
isempty ( a1 ) ; MBS("\SendMail.AddTO\"; $EmailID; a1; a2 ) )" ) ;
rec = MBS( "FM.Loop"; "n"; 1 ; ValueCount ( $ToEmailcc); 1; "let ( [ a1 =
getvalue ( $ToEmailcc ; n ) ; a2 = getvalue ( $ToNamecc ; n ) ] ; if ( not
isempty ( a1 ) ; MBS("\SendMail.AddCC\"; $EmailID; a1; a2 ) )" ) ;
recc = MBS( "FM.Loop"; "n"; 1 ; ValueCount ( $ToEmailbcc); 1; "let ( [ a1 =
getvalue ( $ToEmailbcc ; n ) ; a2 = getvalue ( $ToNamebcc ; n ) ] ; if ( not
isempty ( a1 ) ; MBS("\SendMail.AddBCC\"; $EmailID; a1; a2 ) )" ) ;

// Add Attachments
rc= If ( AttCount > 0 ; MBS( "FM.Loop"; "n"; 0 ; AttCount - 1 ; 1; "let ( [ a1
= MBS("\FM.SQL.Field\"; $AttCont; n; 0 ) ] ; if ( not Isempty( a1) ;
MBS("\SendMail.AddAttachmentContainer\"; $EmailID; a1 ; getvalue
( $AttNameC ; n+1 ) ; getvalue ( $AttTypeC ; n+1 ) ;
\"79680F27-98FD-4E86-99D5-54D40E5536B5/\" & getvalue ( $AttNameC ;
n+1 ) ) )" ) ) ;
rp = If ( AttCountP > 0 ; MBS( "FM.Loop"; "n"; 1 ; AttCountP; 1; "let ( a1 =
getvalue ( $AttPath ; n ) ; if ( not Isempty( a1 ) ;
MBS("\SendMail.AddAttachmentFile\"; $EmailID; a1 ; getvalue ( $AttNameP;
n ) ; getvalue ( $AttTypeP; n ) ;
\"79680F27-98FD-4E86-99D5-54D40E5536B5/\" & getvalue ( $AttNameP ;
n ) ) )" ) ) ;

// Show Email
EmailSource
= MBS( "Text.ReplaceNewline"; MBS("SendMail.GetSource"; $EmailID); 1 ) ]
; // Send Email
If ( Send = 1 ; Let ( [
$curl = MBS("CURL.New") ;
r = MBS("SendMail.PrepareCURL"; $EmailID; $curl) ;

```

```

// Maybe use alternative SMTP port?
r = MBS("CURL.SetOptionPort"; $curl; 587) ;
// This turns TLS on and requires connection to be encrypted
r = MBS("CURL.SetOptionUseSSL"; $curl; 3) ;
// force TLS v1.2
r = MBS("CURL.SetOptionSSLVersion"; $curl; 6) ;
// This disables certificate verification, so we accept any:
r = MBS("CURL.SetOptionSSLVerifyHost"; $curl; 0) ;
r = MBS("CURL.SetOptionSSLVerifyPeer"; $curl; 0) ;
// Better with certificates if you have some:
// r = MBS( "CURL.SetOptionCAInfo"; $curl; "/Library/FileMaker Server/
certificates.pem" ) ;
// r = MBS("CURL.SetOptionSSLVerifyHost"; $curl; 2) ;
// r = MBS("CURL.SetOptionSSLVerifyPeer"; $curl; 1) ;
// Run the transfer
r = MBS("CURL.Perform"; $curl) ;
// get debug messages
DebugInput
= MBS("Text.ReplaceNewline";MBS("CURL.GetInputAsText"; $curl); 1) ;
DebugMessages = MBS("CURL.GetDebugAsText"; $curl; "UTF-8") ;
r = MBS("CURL.Release"; $curl) ;
// Cleanup
r = MBS("SendMail.Release"; $EmailID) ;
r = MBS( "FM.SQL.Release"; $AttCont ) ;
write = MBS( "FM.ExecuteFileSQL"; "" ; "UPDATE \"EM\" SET \"EmailSource\"
= ?, \"DebugInput\" = ?, \"DebugMessages\" = ? WHERE \"EmailID\" = ?
"; 9; 13; EmailSource; DebugInput; DebugMessages; EmailID )
]; "" ) ) )

```

Basically this is a whole script converted in a custom function. As you see a few parameters are passed as arguments and others are queried by SQL queries from records based on an EmailID field. Then it builds the email with all the values. The attachments are a bit tricky since they walk over a SQL result set dynamically to grab the containers from it. The email is then send via our CURL functions and on the end, the record for the email is updated via SQL.

Is this cool? Yes! Do we recomend to do it this way? No.

A custom function to quickly send email anywhere in FileMaker is great, but then I would read SMTP settings from a global configuration table. And I would pass recipients as parameters and on the end create the log record for the email via our [FM.InsertRecord](#) function.

Please note that this example has minimal error handling. It easier works in total or fails somewhere. In a script you would check results and alert the user maybe with a dialog.

What do you think about this?