

Compress and Send

If your user needs to compress one file and send it via email, they can compress the file in Finder or Explorer and attach the file to an email and send it off. But what if you need to do this 100 times and you need a script to do this?

Send XML File as Zip

Let's start with a XML (or JSON) export from FileMaker which you like to email with [MBS FileMaker Plugin](#). You could just add the XML directly as an attachment with [SendMail.AddAttachmentText](#) function and call the job done:

```
Set Variable [ $r ; Value: MBS( "SendMail.AddAttachmentText"; $email; $XML; "UTF-8"; "export.xml"; "application/xml" ) ]
```

But let's use our new [Archive.CompressText](#) function to compress the text to a zip file directly. Then we can attach the zip with [Archive.CompressContainer](#) function to a new email and send it later. When compressing we use zip as format with deflate as compression algorithm, which is the standard one for zip files. There is an option we use to set compression level to 9 for high. Yes, if data would be already compressed, we could use level 0 for no compression. We use UTF-8 for the encoding of the XML file, but please make sure the XML file has UTF-8 specified as encoding in the header line if it exists. Here is the sample script:

```
# get some XML
```

```
Set Variable [ $XML ; Value: "<test>Hello</test>" ]
```

```
# Compress it
```

```
Set Variable [ $CompressedXML ; Value: MBS( "Archive.CompressText"; "zip"; "deflate"; "export.zip"; "export.xml"; $xml; "UTF-8"; "compression-level=9" ) ]
```

```
# now add to email
```

```
Set Variable [ $email ; Value: MBS( "SendMail.CreateEmail" ) ]
```

```
Set Variable [ $r ; Value: MBS( "SendMail.AddAttachmentContainer"; $Email; $CompressedXML; "export.zip"; "application/zip" ) ]
```

```
Set Variable [ $r ; Value: MBS( "SendMail.SetPlainText"; $email; "Dear " & Contacts::FullName & ¶ & ¶ & "Attached you find the export from " & Get(CurrentDate ) & ¶ & ¶ & "Greetings" & ¶ & "Your FileMaker Server" ) ]
```

Send PDF File as Zip

Now let us repeat this with a PDF file. We pick it from a container and pass it to [Archive.CompressContainer](#) function. The file name for the file in the archive is taken from the container, so if needed you can use [Container.Rename](#) to change that before passing if needed. Again we attach the zip to the email like in this example script:

```
# get some PDF from a field
Set Variable [ $PDF ; Value: Reports::ReportContainer ]
# Compress it
Set Variable [ $CompressedPDF ; Value: MBS( "Archive.CompressContainer";
"zip"; "deflate"; "report.zip"; $pdf) ]
# now add to email
Set Variable [ $email ; Value: MBS( "SendMail.CreateEmail") ]
Set Variable [ $r ; Value: MBS( "SendMail.AddAttachmentContainer"; $Email;
$CompressedPDF; "report.zip"; "application/zip" ) ]
Set Variable [ $r ; Value: MBS( "SendMail.SetPlainText"; $email; "Dear " &
Contacts::FullName & ¶ & ¶ & "Attached you find the report from " &
Get(CurrentDate ) & ¶ & ¶ & "Greetings" & ¶ & "Your FileMaker Server") ]
```

Send PDF and XML together

For the final example script we combine both and files, the PDF and the XML file into one zip archive. We do that with using [Archive.Create](#) to start the new zip archive. This time with compression and text encoding option to use UTF-8 to encode file names. Then we add both the XML text and the PDF container to the archive. And when we close the archive, we get back the archive as container value. The rest is the same as above with adding the container to the email, all without using temporary files.

```
# get some XML and PDF
Set Variable [ $PDF ; Value: Reports::ReportContainer ]
Set Variable [ $XML ; Value: "<test>Hello Hello Hello</test>" ]
# Compress it
Set Variable [ $r ; Value: MBS( "Archive.Create"; "zip"; "deflate"; "export.zip";
"compression-level=9¶hdrcharset=UTF-8") ]
Set Variable [ $r ; Value: MBS( "Archive.AddText"; $xml; "UTF-8"; "export.xml") ]
Set Variable [ $r ; Value: MBS( "Archive.AddContainer"; $PDF ) ]
Set Variable [ $Archive ; Value: MBS( "Archive.Close"; "report.zip" ) ]
# now add to email
Set Variable [ $email ; Value: MBS( "SendMail.CreateEmail") ]
Set Variable [ $r ; Value: MBS( "SendMail.AddAttachmentContainer"; $Email;
$Archive; "report.zip"; "application/zip" ) ]
```

```
Set Variable [ $r ; Value: MBS( "SendMail.SetPlainText"; $email; "Dear " &
Kontakte::Vorname & ¶ & ¶ & "Attached you find the export from " &
Get(CurrentDate ) & ¶ & ¶ & "Greetings" & ¶ & "Your FileMaker Server") ]
```

We hope you enjoy the new function and the example given here. Let us know if you have questions.