QR Codes for Invoices in Switzerland

As you may know the Swiss will introduce ISO-20022 and have a QR-Code with payment data on invoices containing bank accounts and amount. The specifications are not yet finished and may take another year to get done. Introduction is planned for 2019, so you should make sure your software is ready.
Based on the current spec I made the follow example script for all my plugin users:

```
# ##### SWISS QR CODE GENERATION

# Sample data for QR-Code. May need adjustment to final standard
Set Variable [ $text ; Value: "SPC" & ¶ &
"0100" & ¶ &
"1" & ¶ &
"CH1234567890123456789" & ¶ &
"Hans Müller" & ¶ &
"Bahnhofstraße" & ¶ &
"123" & ¶ &
"1234" & ¶ &
"Dorf am See" & ¶ &
"CH" & ¶ &
"Hans Müller" & ¶ &
 "Bahnhofstraße" & ¶ &
"123" & ¶ &
"1234" & ¶ &
"Dorf am See" & ¶ &
"CH" & ¶ &
"1234.56" & ¶ &
"CHF" & ¶ &
"2018-12-27" & ¶ &
"Lisa Müller" & ¶ &
"Dorfstraße" & ¶ &
"12" & ¶ &
"5678" & ¶ &
"Dorf am Berg" & ¶ &
"CH" & ¶ &
"QRR" & ¶ &
"000000000000000000000001234" & ¶ &
"ligne 28" ]
# Barcode options. We use high level as we draw over barcode.
Set Variable [ $o ; Value: MBS("Barcode.SetOptions"; 4) // ECC Level with values from 1 = low, 2 = middle, 3 = better, 4 high. ]

# CRLF line endings needed
Set Variable [ $text ; Value: MBS( "Text.ReplaceNewline"; $text; 3 ) ]
```

```
# We render barcode at 4x size for better drawing later
Set Variable [ $img ; Value: MBS("Barcode.Generate";"QRCODE"; $text; 0; 0; 0; 4; 0; 1;
"UTF-8") ]
If [ MBS("IsError") = 0 ]
	Set Variable [ $r ; Value: MBS( "GMImage.SetType"; $img; 6) ]

	# Load cross image
	Set Variable [ $cross ; Value: MBS( "GMImage.NewFromContainer";
base_test::Cross) ]
	# Scale to match size of barcode proportional
	Set Variable [ $w ; Value: MBS( "GMImage.GetWidth"; $img) ]
	Set Variable [ $k ; Value: Runden($w / 6,5; 0) // 20 point to 130 point is factor 6.5 ]
	Set Variable [ $r ; Value: MBS( "GMImage.Scale"; $cross; $k & "x" & $k) ]
	# Draw cross over barcode
	Set Variable [ $r ; Value: MBS( "GMImage.Composite"; $img; $cross;
"CenterGravity"; 1) ]

	# Save to field
	Set Field [ base_test::Swiss_QR ; MBS( "GMImage.WriteToPNGContainer"; $img;
"barcode.png" ) ]
	# Output with around 4.5cm side length so Swiss cross is 7mm big, (130 point in
PDF and 20 point for cross)

	# Clean up
	Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $cross) ]
	Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $img) ]
End If
```

As you see, we put together the data for the QR-Code. The given data is not correct (checksums) as I use sample IBANs and the reference number is just 1234. But you can collect the right data in your solution and assemble them as needed. Please use latest specs and make sure your data doesn't contain extra spaces or newline characters. Once you have the data, we convert line endings to Windows format (CRLF). Next we create a barcode in QR-Format with ECC Level High. We load the Swiss cross image overlay and scale it to the right size to composite over the barcode (needs fix in upcoming 7.6pr5 or use GMImage.CompositeXY instead). Than we write it to container, so you can place it on a layout to print or place the image on a PDF with DynaPDF.InsertImage.