OTP Helper

The OTP Helper provides handlers to generate and verify time-based one-time passwords conforming to RFC6238 TOTP standards. Additionally it includes a handler to draw QR codes based on the otpauth:// URI scheme. You can use this helper on its own but as the Authentication Library incorporates the OTP helper it makes more sense and it is easier to deal with the Authentication Library, it's OTP related handlers and the associated settings in application/config/authentication.lc.

Note: This helper requires LiveCode's QR Code Generator Library which needs to be stored in application/libraries. Rename the library to "QR.livecodescript", So, the path to the library is application/libraries/QR.livecodescript. Please read about how to create libraries in chapter "Creating Libraries".

Note: You don't need to load the QR library, this is done by the helper.

Preparation of the QR Code Generator Library

Find the library in the LiveCode application in Extensions/com.livecode.library.qr/qr.livecodescript.
Rename it to "QR.livecodescript".
Open the library in an editor and capitalize the name of the script in the first line.
Then add the following code to the library script:

```
global gRigA

local sStackInUse


on libraryStack
  if (gRigA is not an array) and (the environment is "server") then
    put "No direct script access allowed."
    exit to top
  end if

  if the short name of the target = the short name of me then
    if sStackInUse <> TRUE then
      put TRUE into sStackInUse

      -- LOGGING
      if the environment is "server" then
        rigLogMessage "debug", "QR Library Loaded"
      end if

      -- SET INITIAL VALUES OF THE LIBRARY VARIABLES
      _rigSetDefaultValues
      --
    end if -- if sStackInUse <> TRUE

  else
    pass libraryStack
  end if -- if the short name of the target = the short name of me
end libraryStack



-- SET INITIAL VALUES
private command _rigSetDefaultValues
  __Initialize
end _rigSetDefaultValues



command rigRunInitialQRConfig
  --Run initial configuration procedures. Don't remove this handler, even if it does nothing.
end rigRunInitialQRConfig
```

Loading the OTP Helper

This helper is loaded using the following code:

```
rigLoadHelper "otp"
```

Note: You don't need to load the helper if you use the Authentication Library. Please read about OTP authentication in chapter "One-Time Password Authentication".

Handler Reference

rigOTPgenerate pSecret, pTokenLength, pCryptoType, pCryptoNumBits, pTimeStep, pType

Generate a one-time password.

- pSecret is the decoded shared secret.
- pTokenLength (optional) is the length of the one-time password, usually this is 6 or 8, if not provided 6 is used.
- pCryptoType (optional) is "SHA" or "SHA3", though keep in mind that "SHA3" does not conform to the RFC6238 TOTP standards. So, the default value of this parameter is "SHA".
- pCryptoNumBits (optional) is the output size of the hash algorithm in bits (exception is 1 for "SHA1" with an output size of 160 bits). Possible values are 224, 256, 384, 512 or 1, the default value.
- pTimeStep (optional) defines a period in seconds that a TOTP code will be valid for. The default value is 30.
- pType is the OTP type, this is "HOTP" or "TOTP". Currently the helper supports TOTP only. So, this parameter can be ignored.

Returns FALSE in the result in case an error occurred, otherwise the result contains the one-time password.

rigOTPcompareKeys pChallenge, pKey, pTokenLength, pCryptoType, pCryptoNumBits, pTimeStep, pTimeWindow

Compare user supplied OTP with generated OTP.

- pChallenge is the one-time password the user supplied.
- pKey is the generated OTP based on stored data.
- pTokenLength (optional) is the length of the one-time password, usually this is 6 or 8, if not provided 6 is used.
- pCryptoType (optional) is "SHA" or "SHA3", though keep in mind that "SHA3" does not conform to the RFC6238 TOTP standards. So, the default value of this parameter is "SHA".
- pCryptoNumBits (optional) is the output size of the hash algorithm in bits (exception is 1 for "SHA1" with an output size of 160 bits). Possible values are 224, 256, 384, 512 or 1, the default value. Note: The values 224 and 384 do not conform to the RFC6238 TOTP standards.
- pTimeStep (optional) defines a period in seconds that a TOTP code will be valid for. The default value is 30.
- pTimeWindow (optional) is a delay window to compare OTPs not only in the current period but also with the next and previous time steps. The default value is 2.

Returns TRUE in the result if the two passwords match.

rigOTPgenerateUserKey pSecret

Generate a base32 encoded user key that conforms to the RFC6238 TOTP standards. Provide the secret as parameter.
Returns the base32 encoded shared secret in the result which can be displayed for the purpose of manually entering the secret in authenticator apps. The secret should always be transferred over a secure channel.

rigOTPqrCode(pKey, pAccount, pIssuer, pAlgo, pDigits, pPeriod, pECC, pSize, pMask)

Create QR code image data based on OTP URI data.

- pKey is the base32 encoded shared secret.
- pAccount is a user's email address or name.
- pIssuer is the name of the entity issuing user accounts.
- pAlgo (optional) defines the algorithm used. Can be "SHA1", "SHA256" or "SHA512". If not provided this parameter defaults to "SHA1".
- pDigits (optional) is the length of the one-time password, usually this is 6 or 8, if not provided 6 is used.
- pPeriod (optional) defines a period in seconds that a TOTP code will be valid for. The default value is 30.
- pECC (optional) is the error correction level. Possible values are "L", "M", "Q" and "H". If the parameter is left empty "M" is used.
- pSize (optional) defines the size of the displayed QR code image. Valid values are in the range of 1 to 10. The default value is 4.
- pMask (optional) is an integer between 0 and 7 which defines a pattern that changes the outputted matrix. The default value is "Auto".

Returns FALSE in the result in case an error occurred, otherwise the result contains the base64 encoded image data of the QR code. Here is an example:

```
rigOTPgenerateUserKey tSecret
put the result into tKey
put "user@example.com" into tAccount
put "Issuer Example" into tIssuer
put "SHA256" into tAlgo
put 8 into tDigits
put 60 into tPeriod
put "Q" into tECC
put 5 into tSize

put rigOTPqrCode(tKey, tAccount, tIssuer, tAlgo, tDigits, tPeriod, tECC, tSize) into tImgData

put "<img src='" & tImgData & "' />" into gData["qrCode"]
```