

# CURL in Detail

What CURL plugin can do for you.

# About CURL

- CURL command line tool
  - included with Mac OS X, Linux
  - available for Windows
- libCURL, the library behind
  - C library

# About CURL

- Transfers data in various protocols
- SSL with Certificates
- Proxies
- Cookies
- Authentications

# About CURL

- File Transfer Resume
- Proxy Tunneling
- HTTP POST/PUT
- HTTP Form based upload
- FTP upload

# Protocols

- DICT
- FILE
- FTP(s)
- Gopher
- HTTP(s)
- IMAP(s)
- LDAP(s)
- POP3(s)
- RTMP & RTSP
- SCP
- SFTP
- SMTP(s)
- Telnet
- TFTP

# LibCURL

- Portable C Library for Mac, Windows, Linux and others
- Thread Safe
- IPv6 Compatible
- Feature Rich
- Fast
- Well documented

# MBS CURL Plugin

- CURLSMBS class
- Built in CURL, SSH and SSL libraries.
- Background transfers with Threading
- Various simple & advanced interfaces

# MBS CURL Plugin

- Simple download:
- ```
dim c as new CURLSMBS
c.CollectOutputData = true
c.OptionURL = "http://www.mbsplugins.de/"
dim e as integer = c.Perform
dim h as integer = c.GetInfoResponseCode
dim d as string = c.OutputData
Break
```
- See result in debugger.

# Options

- `OptionURL` - the target URL
- `OptionUpload` - upload or download?
- `OptionPOST`, `OptionPUT`, `OptionCustomRequest`
- `OptionTimeOut`
- `OptionVerbose` - enable debug messages
- `OptionFollowLocation` - follow redirects

# Input / Output

- Via Events
- Collect in memory and store in property (beware of size limits!)
- Provide input via variable in memory
- Provide input via form data
- Stream directly from/to file

# Receiving Debug Messages

- event DebugMessage(int type as integer, data as string)
- CollectDebugData = true  
Later query DebugData property.
- CreateMTDebugOutputFile &  
CloseMTDebugOutputFile

# Receiving Header

- event Header(data as string) as integer
- CollectHeaderData = true  
Later query HeaderData property.
- CreateMTHeaderOutputFile &  
CloseMTHeaderOutputFile

# Receiving Data (Download)

- event Write(data as string) as integer
- CollectOutputData = true  
Later query OutputData property.
- CreateMTOutputFile & CloseMTOutputFile

# Data Upload

- event Read(count as integer) as string  
event RestartRead() as boolean  
event Seek(pos as Int64, whence as integer) as Integer
- property InputData as String  
method SetInputData(data as MemoryBlock)  
method SetInputData(data as string)
- OpenMTInputFile
- OptionPostFields - for POST
- FormAdd & FormFinish

# Perform transfer

- Perform
  - Can block GUI
  - YieldTime property
- PerformMT
  - run on Xojo Thread
  - Performs transfer on background preemptive thread
  - no read event!

# Information while transfer

- event Progress(dltotal as double, dlnow as double, ultotal as double, ulnow as double) as boolean
- Return true to cancel transfer. Or set Cancel property.
- Live data via DebugMessage, Write & Header events.

# Info after transfer

- `GetInfoEffectiveURL` - URL used
- `GetInfoRedirectCount`
- `GetInfoResponseCode` - HTTP Status codes
- various times
- content length uploaded/downloaded

# Build form data

- `d.FormAdd(d.kFormCopyName, "name", d.kFormCopyContents, "Mr. Tester")`
- `d.FormAdd(d.kFormCopyName, "email", d.kFormCopyContents, "test@test.invalid")`
- `d.FormAdd(d.kFormCopyName, "myfile", d.kFormBuffer, file.Name, d.kFormBufferPtr, buf, d.kFormBufferLength, lenb(buf))`
- `d.FormFinish`

# FTP Operations

- Run native FTP commands
- OptionQuote - instead of transfer  
OptionPreQuote - before transfer  
OptionPostQuote - after transfer
- SetOptionPostQuote array("DELETE Temp.txt")
- dim commands() As String  
commands.Append "RNFR Temp.txt"  
commands.append "RNTO MyFile.txt"  
d.SetOptionPostQuote(commands)

# IMAP Email access

- List emails:

OptionURL = "imap://" + Server + "/INBOX"

OptionCustomRequest = "FETCH 1:\* FLAGS"

- Download Email with ID:

OptionURL = "imap://" + Server + "/INBOX;UID="+UID

- Create Folder:

OptionURL = "imap://" + Server + "/"

OptionCustomRequest = "CREATE MyFolder"

# Send Email via SMTP

- First build email source code e.g. with `EmailMessage` class
- Use email source as input data for transfer
- Set `OptionMailFrom` and `OptionMailRecipients`
- Use URL with SMTP/SMTPS prefix
- Run upload transfer

# Send Email via SMTP

- `dim em as new EmailMessage`  
`em.AddRecipient ToName.text`  
`em.FromAddress = FromName.text`  
`em.Subject = Subject.text`  
`em.BodyPlainText = content.text`
- `d.InputData = em.source`  
`d.OptionMailFrom = FromName.text`  
`d.SetOptionMailRecipients array(ToName.text)`  
`d.OptionURL = "smtp://" + server.Text`  
`d.OptionUpload=true`

# Send Email via SMTP

- New CURLEMailMBS class
- add text, attachments, recipients, subject and more
- UTF-8 encoding
- Built in Quoted Printable Encoding

# SSL

- Without security:
  - `d.OptionSSLVerifyHost = 0`  
`d.OptionSSLVerifyPeer = 0`
- With security:
  - get cacert.pem file or use own certificate
  - `d.OptionSSLVerifyHost = 2`  
`d.OptionSSLVerifyPeer = 1`  
`d.OptionCAInfo = cacert.NativePath`

# Resume Upload/Download

- Set `OptionResumeFromLarge` before transfer
- Provide data starting given position (Upload)
- Receive data starting given position (Download)

# iOS?

- Plugin SDK not yet available
- No CURL on iOS for declares
- NSURLConnection until CURL is available

**Questions?**