gpxutils Group

John D. Baker

https://github.com/bakerjd99/jacks/blob/master/geotagged/gpxutils.ijs

SHA-256: 8dbe6e87f69bbee8653aab48be279bf13119088ead06d2f692b59e7623aaf311

November 11, 2020

Contents

gpxutils Overview

gpxutils is a J script that that formats Garmin style waypoint GPX files from CSV files and the SmugMug Sqlite mirror.db database. The resulting GPX files can be loaded into the Motion-GPS iPhone app and other GPS devices that import GPX data..

To run gpxutils inspect interface word comments.

gpxutils is generated from JOD dictionaries gps and utils

```
NB. open JOD dictionaries and generate gpxutils script
load 'general/jod'
od ;:'gps utils'
mls 'gpxutils'
```

A generated gpxutils script and sample mirror.db database are here:

- https://github.com/bakerjd99/jacks/blob/master/geotagged/gpxutils.ijs
- https://github.com/bakerjd99/smugpyter/blob/master/testdata/mirror.zip/mirror.db

gpxutils Interface

```
allrecent   [5] all recent images from last waypoint generation
gpxfrmirror [11] extracts geotagged images from mirror_db and generates gpx
gpxfrpoicsv [12] converts poi csv files to gpx
gpxfrrecent [14] gpx from recent waypoints
```

gpxutils Source Code

```
NB.*gpxutils  s--  generate  gpx  waypoint  files  from  various
NB. sources.
NB.
NB. This group formats Garmin  style waypoint  gpx files from CSV
NB. files and my  SmugMug sqlite  mirror database.  The resulting
NB. gpx  files can be loaded into the Motion-GPS  iPhone  app and
NB. other GPS devices that import gpx data.
NB.
NB. verbatim: interface words
NB.
NB.  allrecent   - all recent images from last waypoint generation
NB.  gpxfrmirror - extracts geotagged images from mirror_db and generates gpx
NB.  gpxfrpoicsv - converts poi csv files to gpx
NB.  gpxfrrecent - gpx from recent waypoints
NB.
NB. created: 2019Dec11
NB. changes: ------------------------------------------------
NB. 19dec18 added (allrecent)

require 'data/sqlite'
coclass 'gpxutils'
NB.*end-header

NB. get all images from mirror - select columns
AllMirror_sql=: 'select Latitude, Longitude, RealDate, UploadDate, OnlineImageFile from OnlineImage'
```

```
NB. carriage return character
CR=: 13{a.

NB. header template for gpx xml
GPXHEADER=: 60 103 112 120 32 120 109 108 110 115 61 34 104 116 116 112 58 47 47 119 119 119 46 116 111 112
>..> 111 103 114 97 102 105 120 46 99 111 109 47 71 80 88 47 49 47 49 34 32 120 109 108 110 115 58 120 115 105
>..> 61 34 104 116 116 112 58 47 47 119 119 119 46 119 51 46 111 114 103 47 50 48 48 49 47 88 77 76 83 99 104
>..>101 109 97 45 105 110 115 116 97 110 99 101 34 32 99 114 101 97 116 111 114 61 34 74 32 87 97 121 112 111
>..>105 110 116 115 34 32 118 101 114 115 105 111 110 61 34 49 46 49 34 32 120 115 105 58 115 99 104 101 109 9
>..>7 76 111 99 97 116 105 111 110 61 34 104 116 116 112 58 47 47 119 119 119 46 116 111 112 111 103 114 97 10
>..>2 105 120 46 99 111 109 47 71 80 88 47 49 47 49 32 104 116 116 112 58 47 47 119 119 119 46 116 111 112 111
>..> 103 114 97 102 105 120 46 99 111 109 47 71 80 88 47 49 47 49 47 103 112 120 46 120 115 100 34 62 13 10 13
>..> 10 60 109 101 116 97 100 97 116 97 62 13 10 60 108 105 110 107 32 104 114 101 102 61 34 104 116 116 112 5
>..>8 47 47 119 119 119 46 106 115 111 102 116 119 97 114 101 46 99 111 109 34 62 13 10 60 116 101 120 116 62
>..>74 32 40 103 112 120 117 116 105 108 115 41 32 108 97 115 116 32 119 97 121 112 111 105 110 116 32 61 32 1
>..>23 123 100 97 116 101 125 125 60 47 116 101 120 116 62 13 10 60 47 108 105 110 107 62 13 10 13 10 60 47 10
>..>9 101 116 97 100 97 116 97 62 13 10{a.


NB. valid gpx name characters
GPXNAMECHARS=: ' -()0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'

NB. get geotagged images from mirror - rows in desc upload date
GpxGeotaggedMirror_sql=: 'select Latitude, Longitude, RealDate, UploadDate, OnlineImageFile from OnlineImag
>..>e where Keywords like "%geotagged%"'
```

```
NB. interface words (IFACEWORDSgpxutils) group
IFACEWORDSgpxutils=: <;._1 ' allrecent gpxfrmirror gpxfrpoicsv gpxfrrecent'

NB. line feed character
LF=: 10{a.

NB. gpx file written by (gpxutils)
MIRRORGPXFILE=: 'c:/pd/coords/gpx/geotagged smugmug images.gpx'

NB. root words (ROOTWORDSgpxutils) group
ROOTWORDSgpxutils=: <;._1 ' AllMirror_sql GpxGeotaggedMirror_sql IFACEWORDSgpxutils ROOTWORDSgpxutils allre
>..>cent gpxfrmirror gpxfrpoicsv gpxfrrecent write'


NB. retains string (y) after last occurrence of (x)
afterlaststr=: ] }.~ #@[ + 1&(i:~)@([ E. ])

allrecent=: 3 : 0

NB.*allrecent v-- all recent images from last waypoint generation.
NB.
NB. monad:  bt =. allrecent clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   allrecent trg
NB.
```

```
NB. dyad:  bt =. clGpxFile gpxfrrecent clMirrorDb
NB.
NB.   lastgpx=. 'c:/pd/coords/gpx/geotagged test images.gpx'
NB.   lastgpx allrecent trg

MIRRORGPXFILE allrecent y
:
waydate=. waystmp gpx=. read x NB. extract last waypoint date

NB. the last upload date is shifted forward to partly compensate
NB. for the mixture of UTC and local dates. The times in the database
NB. come from many time zones and many timestamps are just approximations.
sql=. AllMirror_sql , ' where UploadDate > date("',waydate,'", ''+16 hours'') order by UploadDate desc '
sql fst y
)

NB. trims all leading and trailing blanks
alltrim=: ] #~ [: -. [: (*./\. +. *./\) ' '&=

NB. signal with optional message
assert=: 0 0"_ $ 13!:8^:((0: e. ])`(12"_))

NB. retains string before first occurrence of (x)
beforestr=: ] {.~ 1&(i.~)@([ E. ])

NB. boxes open nouns
boxopen=: <^:(L. = 0:)
```

```
changestr=: 4 : 0

NB.*changestr v-- replaces substrings - see long documentation.
NB.
NB. dyad:  clReps changestr cl
NB.
NB.   NB. first character delimits replacements
NB.   '/change/becomes/me/ehh' changestr 'blah blah ...'

pairs=. 2 {."(1) _2 [\ <;._1 x      NB. change table
cnt=._1 [ lim=. # pairs
while. lim > cnt=.>:cnt do.         NB. process each change pair
  't c'=. cnt { pairs               NB. /target/change
  if. +./b=. t E. y do.             NB. next if no target
    r=. I. b                        NB. target starts
    'l q'=. #&> cnt { pairs         NB. lengths
    p=. r + 0,+/\(<:# r)$ d=. q - l NB. change starts
    s=. * d                         NB. reduce < and > to =
    if. s = _1 do.
      b=. 1 #~ # b
      b=. ((l * # r)$ 1 0 #~ q,l-q) (,r +/ i. l)} b
      y=. b # y
      if. q = 0 do. continue. end.  NB. next for deletions
    elseif. s = 1 do.
      y=. y #~ >: d r} b            NB. first target char replicated
    end.
    y=.(c $~ q *# r) (,p +/i. q)} y  NB. insert replacements
```

```
  end.
end. y                                NB. altered string
)

NB. enclose all character lists in blcl in " quotes
dblquote=: '"'&,@:(,&'"')&.>

eletags=: 4 : 0

NB.*eletags v-- encloses xml text (y) in xml element tag.
NB.
NB. dyad:  clTag eletags clXml

tag=. alltrim x
'<',tag,'>',y,'</',tag,'>'
)

fmtmirrorgpx=: 3 : 0

NB.*fmtmirrorgpx v-- formats mirror_db sql query results as gpx.
NB.
NB. monad:  fmtmirrorgpx btSqlDict

NB. insure any singletons are shaped
ix=. I. (0 {"1 y) e. ;:'RealDate UploadDate OnlineImageFile'
y=. (boxopen&.> (<ix;1){y) (<ix;1)} y
y=. (,&.> 1 {"1 y) (<a:;1)} y
```

```
NB. quit if no data
if. +./ 0 = #&> 1 {"1 y do. '' return. end.

NB. !(*)=. Latitude Longitude RealDate UploadDate OnlineImageFile
(0 {"1 y)=. 1 {"1 y

NB. clean file names
names=. '['&beforestr@justfile&.> OnlineImageFile
names=. alltrim&.> names -.&.> names -.&.> <GPXNAMECHARS
'names cannot be null' assert -. 0 e. #&> names

NB. format latitude and longitude
wpt=.  (<LF,'<wpt lat=') ,. (dblquote 8!:0 Latitude) ,. (<' lon=') ,. (dblquote 8!:0 Longitude) ,.  <'>'

NB. format dates for gpx
RealDate=. alltrim@((,&'Z')@('+'&beforestr))&.> RealDate
UploadDate=. alltrim@((,&'Z')@('+'&beforestr))&.> UploadDate

NB. use real date unless empty else use upload date
ix=. I. 0 = #&> RealDate
RealDate=. (ix{UploadDate) ix} RealDate
wpt=.  wpt ,. 'time'&eletags&.> RealDate

NB. waypoint names & descriptions
wpt=. wpt ,. _1 |."1 names ,"0 1 |. tags 'name'
```

```
NB. symbols
wpt=. wpt ,. <'sym' eletags 'waypoint'
wpt=.  wpt ,. <'</wpt>'

NB. last waypoint upload date
gpxhead=. ('/{{date}}/', }: ;0{UploadDate) changestr GPXHEADER

NB. return gpx
gpxhead,(;wpt),LF,'</gpx>'
)

fsd=: 4 : 0

NB.*fsd v-- fetch sqlite dictionary array.
NB.
NB. dyad:  clSql fsd clDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   sql=. 'select ImageKey, OriginalWidth, OriginalHeight, OnlineImageFile, Keywords from OnlineImage'
NB.   sql fsd trg

NB. require 'data/sqlite' !(*)=. sqlclose__db sqldict__db sqlopen_psqlite_
d [ sqlclose__db '' [ d=. sqldict__db x [ db=. sqlopen_psqlite_ y
)

fst=: 4 : 0
```

```
NB.*fst v-- fetch sqlite reads table.
NB.
NB. dyad:  bt =. clSql fst clDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   sql=. 'select ImageKey, OriginalWidth, OriginalHeight, OnlineImageFile, Keywords from OnlineImage'
NB.   sql fst trg

NB. require 'data/sqlite' !(*)=. sqlclose__db sqlreads__db sqlopen_psqlite_
d [ sqlclose__db '' [ d=. sqlreads__db x [ db=. sqlopen_psqlite_ y
)

gpxfrmirror=: 3 : 0

NB.*gpxfrmirror v-- extracts geotagged images from mirror_db and generates gpx.
NB.
NB. monad:  clGpx =. gpxfrmirror clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   gpx=. gpxfrmirror trg
NB.   (toHOST gpx) write 'c:/pd/coords/gpx/geotagged smugmug images.gpx'
NB.
NB. dyad:  clGpx =. iaN gpxfrmirror clMirrorDb
NB.
NB.   10 gpxfrmirror trg

0 gpxfrmirror y NB. all waypoints default
:
```

```
NB. limit waypoints
sql=. GpxGeotaggedMirror_sql , ' order by UploadDate desc ' , ;(0<x){'';' limit ',":x
fmtmirrorgpx sql fsd y
)

gpxfrpoicsv=: 3 : 0

NB.*gpxfrpoicsv v-- converts poi csv files to gpx.
NB.
NB. This  verb converts  comma delimited point of interest  (POI)
NB. *.csv files to Garmin compatible gpx files. Example POI files
NB. can be downloaded from:
NB.
NB. http://www.poi-factory.com/poifiles
NB.
NB. monad:  clGpx =. gpxfrpoicsv clCsvfile
NB.
NB.   gpx=. gpxfrpoicsv 'c:\pd\coords\poicsv\ca_park_m.csv'
NB.
NB. dyad:  clGpx =. iaRows gpxfrpoicsv clCsvfile
NB.
NB.   gpx=. 10 gpxfrpoicsv 'c:\pd\coords\poicsv\ca_park_m.csv'

0 gpxfrpoicsv y NB. format all waypoints default
:
NB. read csv file
csv=. parsecsv read y
```

```
if. 0<x do. csv=. (x<.#csv) {. csv end.

NB. sanity test latitude and longitude
lbcheck=. -. _9999 e. , _9999 ".&> 0 1 {"1 csv
'invalid longitude latitude number representations' assert lbcheck

NB. clean names
names=. 2 {"1 csv
names=. alltrim&.> names -.&.> names -.&.> <GPXNAMECHARS
'names cannot be null' assert -. 0 e. #&> names

NB. format latitude and longitude
csv=. (dblquote 0 1 {"1 csv) (1 0)}"1 csv
wpt=.  (<LF,'<wpt lat=') ,. (0{"1 csv) ,. (<' lon=') ,. (1{"1 csv) ,.  <'>'

NB. times set to now
wpt=.  wpt ,. <'time' eletags nstmp=. gpxtimestamp 6!:0''

NB. waypoint names & descriptions
wpt=. wpt ,. _1 |."1 names ,"0 1 |. tags 'name'
NB. wpt=. wpt ,. _1 |."1 (alltrim&.> 3 {"1 csv) ,"0 1 |. tags 'desc'

NB. symbols
wpt=. wpt ,. <'sym' eletags 'waypoint'
wpt=.  wpt ,. <'</wpt>'

NB. waypoint format date
```

```
gpxhead=. ('/{{date}}/', }:nstmp) changestr GPXHEADER

NB. return gpx
gpxhead,(;wpt),LF,'</gpx>'
)

gpxfrrecent=: 3 : 0

NB.*gpxfrrecent v-- gpx from recent waypoints.
NB.
NB. monad:  clGpx =. gpxfrrecent clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   gpx=. gpxfrrecent trg
NB.   (toHOST gpx) write 'c:/pd/coords/gpx/recent geotagged images.gpx'
NB.
NB. dyad:  clGpx =. clGpxFile gpxfrrecent clMirrorDb
NB.
NB.   lastgpx=. 'c:/pd/coords/gpx/geotagged test images.gpx'
NB.   lastgpx gpxfrrecent trg

MIRRORGPXFILE gpxfrrecent y
:
waydate=. waystmp gpx=. read x NB. extract last waypoint date

NB. the last upload date is shifted forward to partly compensate
NB. for the mixture of UTC and local dates. The times in the database
NB. come from many time zones and many timestamps are just approximations.
```

```
sql=. GpxGeotaggedMirror_sql , ' and UploadDate > date("',waydate,'", ''+16 hours'') order by UploadDate d
>..>esc '
fmtmirrorgpx sql fsd y
)


gpxtimestamp=: 3 : 0

NB.*gpxtimestamp v-- format time for Garmin gpx as: yyyy-mm-ddThr:mn:scZ
NB.
NB. monad: cl =. gpxtimestamp nlTime | ntTime
NB.
NB.   gpxtimestamp 6!:0 ''
NB.
NB.   gpxtimestamp 10 # ,: 6!:0 ''  NB. table

r=. }: $y
t=. _6 [\ , 6 {."1 y
d=. '--T::' 4 7 10 13 16 }"1 [ 4 3 3 3 3 3 ": <.t
c=. {: $d
d=. ,d
d=. '0' (I. d=' ')} d
'Z' ,"1~ (r,c) $ d
)

NB. file name from fully qualified file names
justfile=: (] #~ [: -. [: +./\ '.'&=)@(] #~ [: -. [: +./\. e.&':\')
```

```
parsecsv=: 3 : 0

NB.*parsecsv v--  parses comma delimited  files. (x) is the field
NB. delimiter. Lines are delimited with either CRLF or LF
NB.
NB. monad:  btcl =. parsecsv cl
NB. dyad:   btcl =. ca parsecsv cl
NB.
NB.   ',' parsecsv read 'c:\comma\delimted\text.csv'


',' parsecsv y
:
'separater cannot be the " character' assert -. x -: '"'

NB. CRLF delimited *.csv text to char table
y=.  x ,. ];._2 y -. CR

NB. bit mask of unquoted " field delimiters
b=. -. }. ~:/\ '"' e.~  ' ' , , y
b=. ($y) $ b *. , x = y

NB. use masks to cut lines
b <;._1"1 y
)

NB. reads a file as a list of bytes
read=: 1!:1&(]`<@.(32&>@(3!:0)))
```

```
NB. xml BEGIN and END tags
tags=: '<'&,@,&'>' ; '</'&,@,&'>'

NB. extract waypoint date from gpx metadata header
waystmp=: [: alltrim '=' afterlaststr '</text>' beforestr ]

NB. writes a list of bytes to file
write=: 1!:2 ]`<@.(32&>@(3!:0))

NB.POST_gpxutils post processor.

smoutput IFACE=:  (0 : 0)
NB. (gpxutils) interface word(s):
NB. ----------------------------
NB. allrecent    NB. all recent images from last waypoint generation
NB. gpxfrmirror  NB. extracts geotagged images from mirror_db and generates gpx
NB. gpxfrpoicsv  NB. converts poi csv files to gpx
NB. gpxfrrecent  NB. gpx from recent waypoints
)

cocurrent 'base'
coinsert  'gpxutils'
```

Index