MirrorXref Group

John D. Baker

https://conceptcontrol.smugmug.com

SHA-256: aea8d224e8184db478022b7e41a6074c20717174d380234b4443b1cd3af53aba

May 27, 2022

Contents

MirrorXref Overview

MirrorXref is a J script that scans the directories written by the Python smugpyter program and builds a new compact version of the SQLite cross reference image metadata database mirror.db

mirror.db cross references local image files with online versions on my SmugMug image site. Local images are catalogued in ThumbsPlus SQLite databases. mirror.db matches local ThumbsPlus image and path keys with online keys.

MirrorXref Interface

BuildMirror	[17]	*backup/create/load mirror*
CheckRealDates	[18]	*check real dates*
DumpLocalImageNatural	[22]	*dump (LocalImage) as TAB delimited text*
LocalFromDir	[34]	*local files with directory match (y) and file match (x)*
MirrorStatReport	[45]	*mirror database upload summary report*
MissingImagesReport	[47]	*missing local images report*
SuspiciousPairReport	[50]	*suspicious local and online file pairings*

MirrorXref Schema

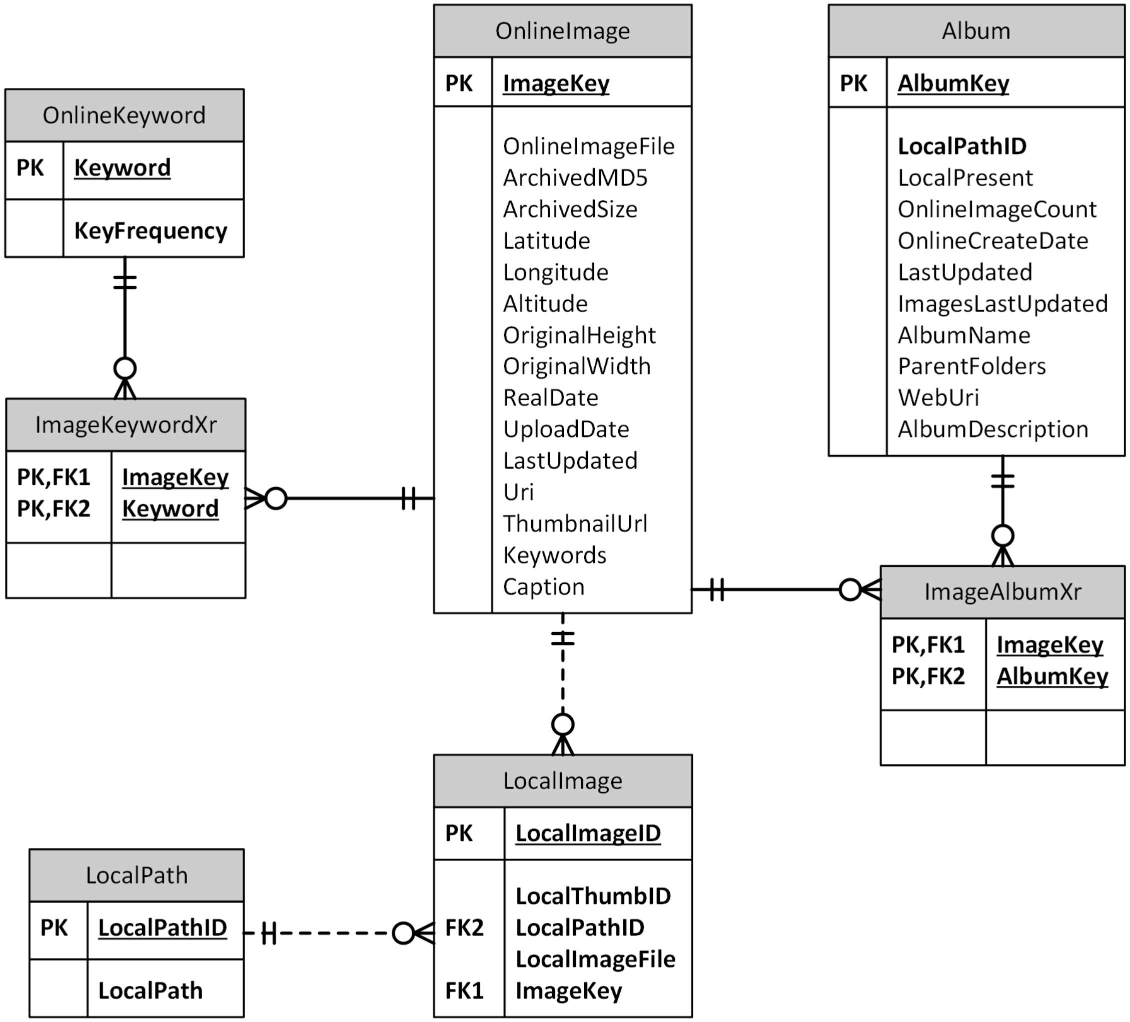

Figure 1: `mirror.db` schema see `CreateMirror_sql` on page ??.

MirrorXref Source Code

```
NB.*MirrorXref s-- create and load sqlite mirror cross reference database.
NB.
NB. This script scans the files in the (SmugMirror) directories that
NB. were written by the Python (smugpyter) scripts and builds a single
NB. compact SQLite image and album metadata database (mirror.db).
NB.
NB. (mirror.db) cross references local image files with online versions
NB. on my SmugMug image site.
NB.
NB. verbatim:
NB.
NB. https://conceptcontrol.smugmug.com/
NB. https://github.com/bakerjd99/smugpyter
NB.
NB. interface word(s):
NB. ------------------------------------------------------------------------------
NB.  BuildMirror           - backup/create/load mirror
NB.  CheckRealDates        - check real dates
NB.  DumpLocalImageNatural - dump (LocalImage) as TAB delimited text
NB.  LocalFromDir          - local files with directory match (y) and file match (x)
NB.  MirrorStatReport      - mirror database upload summary report
NB.  MissingImagesReport   - missing local images report
NB.  SuspiciousPairReport  - suspicious local and online file pairings
NB.
NB. created: 2018jun06
NB. ------------------------------------------------------------------------------
NB. 18jun06 group class created
NB. 18jun23 main (BuildMirror) added
NB. 18jun26 schema finalized and (DumpLocalImageNatural) added
NB. 18jul10 (MissingLocalFiles_sql) added
NB. 18jul12 (MatchOnlineNoBrackets) added
NB. 18jul17 (MissingImagesReport) added
NB. 18dec01 (ParentFolders, ImagesLastUpdated, OnlineImageCount, WebUri) added to (Album) table
NB. 18dec04 (OnlineCreateDate) added to (Album) table
NB. 19feb19 (SuspiciousPairReport) added
NB. 19apr08 display J version - J 9.01 has incompatible language changes
NB. 19aug16 upload summary rate report code added
NB. 19oct03 added (Divisible)
NB. 20nov03 fix typos (Album, LocalImage) foreign keys
NB. 20nov21 add (UnClickHereImages_sql)
NB. 22may27 add (CheckRealDates)

require 'data/sqlite'

coclass 'MirrorXref'

NB.*dependents
NB. (*)=: CreateMirror_sql LocalFile_sql UpdateLocalPresent_sql LocalOnlineFile_sql
NB. (*)=: MissingLocalFiles_sql UnClickHereImages_sql
NB.*enddependents
NB. schema of mirror.db database - parsed on ';' character
CreateMirror_sql=:  (0 : 0)
create table LocalPath
  (LocalPathID integer not null primary key,
   LocalPath text not null);

create table OnlineKeyword
  (Keyword text not null primary key,
   KeyFrequency integer not null);

create table Album
  (AlbumKey text not null primary key,
   LocalPathID integer not null,
   LocalPresent integer check(LocalPresent==0 or LocalPresent==1),
   OnlineImageCount integer,
   OnlineCreateDate date,
   LastUpdated date,
   ImagesLastUpdated date,
   AlbumName text,
   ParentFolders text,
   WebUri text,
   AlbumDescription text,
   foreign key(LocalPathID) references LocalPath(LocalPathID));

create table OnlineImage
  (ImageKey text not null primary key,
   OnlineImageFile text,
   ArchivedMD5 text,
   ArchivedSize integer,
   Latitude float,
   Longitude float,
   Altitude integer,
   OriginalHeight integer,
   OriginalWidth integer,
   RealDate date,
   UploadDate date,
   LastUpdated date,
   Uri text,
   ThumbnailUrl text,
   Keywords text,
   Caption text);

create table ImageAlbumXr
  (ImageKey text not null,
   AlbumKey text not null,
   primary key(ImageKey, AlbumKey),
   foreign key(AlbumKey) references Album(AlbumKey),
   foreign key(ImageKey) references OnlineImage(ImageKey));

create table ImageKeywordXr
  (ImageKey text not null,
   Keyword text not null,
   primary key(ImageKey, Keyword),
   foreign key(Keyword) references OnlineKeyword(Keyword),
   foreign key(ImageKey) references OnlineImage(ImageKey));

create table LocalImage
  (LocalImageID integer not null primary key autoincrement,
   LocalThumbID integer not null,
   LocalPathID integer not null,
   LocalImageFile text not null,
   ImageKey text not null,
   foreign key(LocalPathID) references LocalPath(LocalPathID),
   foreign key(ImageKey) references OnlineImage(ImageKey));
)

NB. NOTE: the j sqlite addon is fussy about how sql is formatted.
NB. Running standard sql pretty printers or indenting sql in your
NB. favorite style is likely to produce code that doesn't work.
NB. Test your sql fragments to avoid this "tickeyboo!"
LocalFile_sql=:  (0 : 0)
select  a.ImageKey, OnlineImageFile, LocalPath from OnlineImage a
inner join ImageAlbumXr b on a.ImageKey = b.ImageKey
inner join Album c on c.AlbumKey = b.AlbumKey
inner join LocalPath d on d.LocalPathID = c.LocalPathID
where c.AlbumKey = ?
order by OnlineImageFile
)

UpdateLocalPresent_sql=:  'update Album set LocalPresent = 1 where AlbumKey in '
```

```
LocalOnlineFile_sql=:  (0 : 0)
select RealDate, c.ImageKey, LocalThumbID, b.LocalPathID, LocalPath, LocalImageFile, OnlineImageFile from
>..>LocalImage a
inner join LocalPath b on b.LocalPathID = a.LocalPathID
inner join OnlineImage c on c.ImageKey = a.ImageKey
order by RealDate
)

MissingLocalFiles_sql=:  (0 : 0)
select count(1) as MissingCnt, c.AlbumName, group_concat(OnlineImageFile, '|') as NoLocalFile from OnlineI
>..>mage a
inner join ImageAlbumXr b on b.ImageKey = a.ImageKey
inner join Album c on c.AlbumKey = b.AlbumKey
where a.ImageKey not in (select ImageKey from LocalImage)
group by c.AlbumName
order by MissingCnt asc
)

UnClickHereImages_sql =:  (0 : 0)
select ImageKey, OnlineImageFile, Caption from OnlineImage where Caption like "%href=%"
and ImageKey not in (select ImageKey from OnlineImage where Caption like "%(click here)%")
)

NB.*end-header


NB. returns image count, key and full name for albums
AlbumImageCount_sql=: 'select count(1) as ImageCnt, a.AlbumKey, b.AlbumName from ImageAlbumXr a inner join
```

```
>..>Album b on a.AlbumKey=b.AlbumKey group by a.AlbumKey'


NB. names of bad gallery count file
BADGALCNTS=: 'badgalcnts.txt'

NB. invald marker date for python real date failures
BOGUSMARKDATE=: 9999 1 1

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

NB. interface words (IFACEWORDSMirrorXref) group
IFACEWORDSMirrorXref=: <;._1 ' BuildMirror CheckRealDates DumpLocalImageNatural LocalFromDir MirrorStatRepo
>..>rt MissingImagesReport SuspiciousPairReport'


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

NB. name of missing local images file
LOCOMISS=: 'locomiss.txt'

NB. name of online local cross reference table
LOCOXREF=: 'locoxref.txt'
```

```
NB. macros associated with (MACROSMirrorXref) group
MACROSMirrorXref=: ,<'base36_py'

NB. name of online local manual corrections cross reference table
MANLOCOXREF=: 'manlocoxref.txt'

NB. sqlite mirror database file name
MIRRORDB=: 'mirror.db'

NB. mirror database path
MIRRORDBPATH=: 'c:/smugmirror/documents/xrefdb/'

NB. head of path - change for os
MIRRORHEAD=: 'c:/'

NB. mirror directory root path
MIRRORPATH=: 'c:/smugmirror/mirror'

NB. version, make count and date
MIRRORVMD=: '0.9.5';56;'27 May 2022 15:42:26'

NB. primary SQLite ThumbsPlus test database - copy of primary database
PRIMARYTEST=: 'c:/thumbsdbs/primarytest.tpdb8s'

NB. fully qualified sqlite primary thumbs database file name
PRIMARYTHUMBS=: 'c:/thumbsdbs/primary2018.tpdb8s'
```

```
NB. root words (ROOTWORDSMirrorXref) group
ROOTWORDSMirrorXref=: <;._1 ' BuildMirror Divisible IFACEWORDSMirrorXref LocalFromDir MACROSMirrorXref MIRR
>..>ORVMD ROOTWORDSMirrorXref UnClickHereImages_sql'


NB. name of suspect image pairs report
SUSPECTPAIRS=: 'suspects.txt'

NB. name of upload rate summary file
UPRATESUM=: 'upratesum.txt'

NB. image counts for given year {{date}} is placeholder
UploadRateCount_sql=: 'select count(1) as ImageCnt, strftime("%Y", {{date}}) as Year from OnlineImage group
>..> by strftime("%Y", {{date}}) order by strftime("%Y", {{date}})'


AlbumImageCount=: 3 : 0

NB.*AlbumImageCount v-- execute (AlbumImageCount_sql) query.
NB.
NB. monad:  bt =. AlbumImageCount clMirrorDb

NB. require 'data/sqlite' !(*)=. sqlopen_psqlite_ sqldict__dt sqlclose__dt
NB. get album image counts and names
dt=. sqlopen_psqlite_ y
d=. sqldict__dt AlbumImageCount_sql
d [ sqlclose__dt ''
)
```

```
BackUpMirrorXrefDb=: 3 : 0

NB.*BackUpMirrorXrefDb v-- backup  mirror  database and erase  if
NB. (x) is 1.
NB.
NB. monad:  iaRc =. BackUpMirrorXrefDb (clMirrorDir ; clBackupDir)
NB.
NB.   mrd=. 'c:/smugmirror/documents/xrefdb/'         NB. mirror ddirectory
NB.   bak=. 'c:/smugmirror/documents/xrefdb/backup/'  NB. mirror backup directory
NB.   BackUpMirrorXrefDb mrd;bak
NB.
NB. dyad:  iaRc =. paClearDb BackUpMirrorXrefDb (clMirrorDir ; clBackupDir)

1 BackUpMirrorXrefDb y
:
'source target'=. tslash2&.> y
'tdate ttime'=. yyyymondd 0
bakpfx=. target,tdate,'-00-'

rc=. 0  NB. assume failure
try.

  if. -.fexist source,MIRRORDB do.
    1 [ smoutput MIRRORDB,' database missing - continue build' return.
  end.

  if. x -: 1 do.
```

```
NB. read (MIRRORDB) bytes - this copy method
NB. works well for files < 200MB and (MIRRORDB)
NB. will remain well below that threshold.
bindb=. read source,MIRRORDB
txtxf=. mantxtf=. sustxtf=. badtxtf=. uprtxtf=. ''
if. fexist source,LOCOXREF     do. txtxf=. read source,LOCOXREF end.
if. fexist source,MANLOCOXREF  do. mantxtf=. read source,MANLOCOXREF end.
if. fexist source,SUSPECTPAIRS do. sustxtf=. read source,SUSPECTPAIRS end.
if. fexist source,BADGALCNTS   do. badtxtf=. read source,BADGALCNTS end.
if. fexist source,UPRATESUM    do. uprtxtf=. read source,UPRATESUM end.
if. 1e6 > #bindb do.
  rc [ smoutput MIRRORDB,' to small' return.
end.

NB. get today's backup files !(*)=. dir
if. #files=. \:~ 1 dir target,tdate,'*.db' do.
  NB. increment day backup - allow 100 per day
  bcnt=. _1&". _2 {. '-'&beforelaststr ;0{files
  if. (bcnt < 0) +. 100 <: bcnt do.
    rc [ smoutput 'backup count invalid' return.
  end.
  bcnt=. ,'r<0>2.0' (8!:2) >:bcnt
  bakpfx=. target,tdate,'-',bcnt,'-'
end.

smoutput 'backing up -> ',bakpfx,MIRRORDB
```

```
  bindb write bakpfx,MIRRORDB
  txtxf write bakpfx,LOCOXREF
  mantxtf write bakpfx,MANLOCOXREF
  sustxtf write bakpfx,SUSPECTPAIRS
  badtxtf write bakpfx,BADGALCNTS
  uprtxtf write bakpfx,UPRATESUM

  NB. clear db source if a backup exists
  if. fexist bakpfx,MIRRORDB do.
    if. 0 > ferase source,MIRRORDB do.
      rc [ smoutput 'unable to erase ',MIRRORDB return.
    end.
  end.

elseif. x -: 0 do.

  smoutput 'removing ',MIRRORDB,' without backup'
  if. 0 > ferase source,MIRRORDB do.
      rc [ smoutput 'unable to erase ',MIRRORDB return.
  end.

elseif.do.

  rc [ smoutput 'invalid backup option' return.

end.
```

```
  rc=. 1  NB. success
catch.
  rc=. 0
end.
rc
)

BracketOverlaps=: 3 : 0

NB.*BracketOverlaps v-- files that have []'ed and nonbracketed versions.
NB.
NB. monad:  blcl =. BracketOverlaps blObj

NB. !(*)=. name
d=. sqldict__y 'select name from Thumbnail'
(0 {"1 d)=. 1 {"1 d

NB. mask files with []'s
b=. */@('[]'&e.)&> name

NB. []'ed and nonbracketed files
fb=. justfile@alltrim@('['&beforestr)&.> b#name
fn=. justfile&.> name #~ -.b

NB. extension free names that have []'ed and nonbracketed versions
fn -. fn -. fb
)
```

```
BuildMirror=: 3 : 0

NB.*BuildMirror v-- backup/create/load mirror.db
NB.
NB. This verb builds a new version of  the mirror.db SQLite local
NB. and online cross  reference database.  It is  designed to run
NB. from  jconsole.exe.  If  run  from other  J  front  ends  the
NB. smoutput messages may not show until the process completes.
NB.
NB. monad:  BuildMirror iaBackup
NB.
NB.   BuildMirror 0   NB. skip backup and build
NB.   BuildMirror 1   NB. backup and build

smoutput 9!:14 '' NB. display J version

sdr=. MIRRORPATH         NB. mirror directory
mrd=. MIRRORDBPATH       NB. mirror database directory
trg=. mrd,MIRRORDB       NB. mirror db
bak=. mrd,'backup'       NB. mirror backup directory
dmp=. mrd,LOCOXREF       NB. online local pairs
mcf=. mrd,MANLOCOXREF    NB. manual online local corrections
mli=. mrd,LOCOMISS       NB. missing local images report
ssp=. mrd,SUSPECTPAIRS   NB. suspect image pairs report
src=. PRIMARYTHUMBS      NB. thumbsplus db
urp=. mrd,UPRATESUM      NB. upload rate summary report
```

```
NB. check real dates for bogus or invalid dates
NB. all such dates should be fixed before building
if. #bf=. CheckRealDates '/real*.txt' do.
  smoutput 'invalid or bogus real dates'
  smoutput 'fix dates in following files and rerun'
  smoutput bf
  return.
end.

if. y BackUpMirrorXrefDb mrd;bak do.
  smoutput LoadMirrorXrefDb sdr;src;trg;dmp;mcf;mli;ssp;urp
else.
  smoutput 'no ',MIRRORDB,' build - backup issue(s)!'
end.
)

CheckRealDates=: 3 : 0

NB.*CheckRealDates v-- check real dates.
NB.
NB. This verb  scans all the (realdate*.txt) files in the smugmug
NB. mirror directories for the occurrence of the python SmugPyter
NB. bogus    date    of    9999-01-01.    The    python    function
NB. (get_album_image_real_dates) stopped working around  May  24,
NB. 2022 so I hacked in a marker date of (9999-01-01T00:00:00) to
NB. mark  affected images  for later manual edits.  This verb  is
NB. used to prevent such dates getting saved in (mirror.db)
NB.
NB. monad:  bt =. CheckRealDates clPfx
NB.
NB.   NB. empty table result indicates dates ok
NB.   bf=. CheckRealDates '/real*.txt'
NB.
NB. dyad:  bt =. ilYYYYMMDD CheckRealDates clPfx

BOGUSMARKDATE CheckRealDates y
:
NB. j profile !(*)=. dirtree
t=. dirtree MIRRORPATH,y

NB. date text for each file
d=. (}.@(2&{"1@readtd2)&.> 0 {"1 t) -.&.> <a:

NB. no real dates
if. 1 e. b0=. 0 ~: #&> d do.

  NB. check any real dates
  ymd=. datefrnum&.> ".@>&.> (-.&'-')@('T'&beforestr) L: 0 b0#d

  NB. all dates should be valid Gregorian
  b1=. */@valdate&> ymd

  NB. no bogus dates
  b2=. x&e.&> ymd
  NB. return any files with invalid or bogus dates
  t #~ (b2 +. -.b1) (b0#i.#b0)} b0

else.
  (0,}.$t)$''  NB. no invalid dates
end.
)

CreateMirrorXrefDb=: 3 : 0

NB.*CreateMirrorXrefDb v--  creates sqlite mirror cross reference
NB. database.
NB.
NB. monad:  blclTableNames = CreateMirrorXrefDb clMirrorDb
NB.
NB.   CreateMirrorXrefDb 'c:/SmugMirror/Documents/XrefDb/mirror.db'

('database exists -> ',y) assert -.fexist y

NB. parse schema sql - assumes ordered create statements delimited by ';'
sql=. reb&.> <;._1 ';', (LF,' ') charsub CreateMirror_sql -. CR

NB. create new empty database
dt=. sqlcreate_psqlite_ y

NB. create tables
for_create. sql do.
  create=. ;create
  ('unable to create table ->',create) assert 0 = sqlcmd__dt create
end.

tabs=. sqltables__dt ''
tabs [ sqlclose__dt ''
)

Divisible=: 3 : 0

NB.*Divisible v-- albums with image counts  that are divisible by
NB. 3 and 5.
NB.
NB. This verb finds albums with image counts that are divisibe by
NB. three and  five. See the related verb (NotDivisible) for more
NB. details.
NB.
NB. monad:  bt =. Divisible clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   Divisible trg
NB.
NB. dyad:   bt =. ia Divisible clMirrorDb
NB.
NB.   2 5 Divisible trg

3 5 Divisible y
:
NB. !(*)=. ImageCnt AlbumName
```

```
(0 {"1 d)=. 1 {"1 d=. AlbumImageCount y

b=. *./ 0 = r=. x |"0 1 ImageCnt
c=. (x *"1 >. ImageCnt %"0 1 x) - ImageCnt
(/: b # ImageCnt ) { (<"0 b # ImageCnt,.c) ,. b # AlbumName
)

DumpLocalImageNatural=: 3 : 0

NB.*DumpLocalImageNatural v--  dump (LocalImage) as TAB delimited
NB. text.
NB.
NB. (LocalImage) is the only table  in (mirror.db) that  contains
NB. information  that  cannot  be  recreated  from  TAB delimited
NB. SmugMirror text files  and ThumbsPlus databases. (LocalImage)
NB. matches online images to local images. Many can be matched by
NB. file name  but many  require  tedious  human  inspection. The
NB. mapping is not one-to-one some online images, like panoramas,
NB. are  derived   from  many  local  images.  This  verb   dumps
NB. (LocalImage) in the  form of TAB delimited  natural keys that
NB. preserves the online local pairing in durable version control
NB. friendly form.
NB.
NB. monad:  bl =. DumpLocalImageNatural (clDumpFile ; clMirrorDb)
NB.
NB.   dmp=. 'c:/smugmirror/documents/xrefdb/locoxref.txt'
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   DumpLocalImageNatural dmp;trg
```

```
'dmp trg'=. y
dt=. sqlopen_psqlite_ trg
oc=. sqlsize__dt 'OnlineImage'
ld=. sqldict__dt LocalOnlineFile_sql
rc=. sqlclose__dt ''

NB. box unboxed columns and format as char
hd=. 0 {"1 ld [ td=. 1 {"1 ld
pos=. I. 0 = L.&> td
td=. (":L:0(<"0)&.>pos{td) pos} td
td=. |: >td

NB. write as TAB delimited file
(hd,td) writetd2 dmp

NB. number of distinct local images
utd=. # ~. (hd i. <'ImageKey') {"1 td

NB. unique match percentage below should equal this result
NB. %/;(#&> > 1{sqlread__dt 'select distinct ImageKey from LocalImage');sqlsize__dt 'OnlineImage'

NB. record count; unique match percentage; file
(#td);(100 * utd%oc);dmp
)

LoadAlbum=: 3 : 0
```

```
NB.*LoadAlbum v-- load (Album) table.
NB.
NB. (Album) references (LocalPath) execute (LoadLocalPath) before
NB. this verb.
NB.
NB. monad:  iaRc =. LoadAlbum (clMirrorDir ; clMirrorDb)
NB.
NB.   sdr=. 'c:/smugmirror/mirror'  NB. smugmug mirror directory
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   LoadAlbum sdr;trg

'sdr trg'=. y

NB. j profile !(*)=. dirtree
NB. all mirror directory albums contain an (ainfo-*.txt) file
ai=. 0 {"1 dirtree (tslash2 sdr),'ainfo-*.txt'

NB. standarize path format
pf=. tolower@('/'&beforelaststr)@('/'&afterstr)&.> ai

NB. fetch local path table
dt=.sqlopen_psqlite_ trg
dat=. sqldict__dt 'LocalPath'

NB. match ainfo paths with local path table
(0 {"1 dat)=. 1 {"1 dat
```

```
NB. table columns !(*)=. LocalPath LocalPathID
LocalPath=. tolower&.> LocalPath
'LocalPath(s) not found - probably new album added' assert (#LocalPath) > >./px=. LocalPath i. pf
LocalPathID=. px { LocalPathID

NB. read all TAB delimited ainfo files
hd=. 0{readtd2 0{ai
dat=. ,@}.@readtd2&> ai
'path id directory ainfo file mismatch' assert (#dat) = #LocalPathID

NB. NOTE: renamed columns - first row original - next new name
NB. ImageCount        Date              Name       Description
NB. OnlineImageCount  OnlineCreateDate  AlbumName  AlbumDescription
cx=. hd i. ;:'AlbumKey ImageCount Date LastUpdated ImagesLastUpdated Name ParentFolders WebUri Description'
msg=. 'missing (AlbumKey, Name, LastUpdated, ImagesLastUpdated ParentFolders, WebUri, Description) col(s)'
msg assert (#hd) > >./cx

NB. match local path thumbsplus ids to albums
dat=. <"1 |: 1 0 0 1 1 1 1 1 1 1 1 (#^:_1)"1 cx {"1 dat
dat=. (<LocalPathID) (1)} dat

NB. set all local sample images flag present to 0 - assume they are missing
dat=. (<0 #~ #LocalPathID) (2)} dat

NB. convert online image counts
dat=. (<_1&".&> ;3{dat) (3)} dat
```

```
NB. insert in Album table
cln=. 'AlbumKey LocalPathID LocalPresent OnlineImageCount OnlineCreateDate'
cln=. ;:cln,' LastUpdated ImagesLastUpdated AlbumName ParentFolders WebUri AlbumDescription'
rc=. sqlinsert__dt 'Album';cln;<dat
rc [ sqlclose__dt ''
)

LoadImageAlbumXr=: 3 : 0

NB.*LoadImageAlbumXr v-- load (ImageAlbumXr) table.
NB.
NB. monad:  iaRc =. LoadImageAlbumXr (clDb ;< btclDat)

'trg dat'=. y
dt=. sqlopen_psqlite_ trg
rc=. dt insqltd 'ImageAlbumXr';<dat
rc [ sqlclose__dt ''
)

LoadImageKeywordXr=: 3 : 0

NB.*LoadImageKeywordXr v-- load (ImageKeywordXr) table.
NB.
NB. monad:  iaRc =. LoadImageKeywordXr (clDb ; <blclDat)
NB.
NB.   sdr=. 'c:/smugmirror/mirror'                          NB. mirror directory
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   'img iax'=. MirrorImageXrTables sdr
NB.   LoadOnlineKeyword trg;<img


'trg dat'=. y
dat=. }.dat [ head=. 0{dat

'image keys'=. <"1 |: (head i. ;:'ImageKey Keywords') {"1 dat
keys=. <;._1&.> (';' ,&.> keys) -.&.> ' '
image=. (#&> keys) # image

NB. insert image keyword xref
dt=. sqlopen_psqlite_ trg
rc=. sqlinsert__dt 'ImageKeywordXr';(;:'ImageKey Keyword');<image;<;keys
rc [ sqlclose__dt ''
)

LoadLocalPath=: 3 : 0

NB.*LoadLocalPath v-- load (LocalPath) table.
NB.
NB. (LoadLocalPath)  uses values  from a ThumbsPlus SQLite (Path)
NB. table. ThumbsPlus builds a complete table of unique paths and
NB. and  path  primary  keys.  All  the  paths  (mirror.db)  will
NB. reference will be in this table.
NB.
NB. monad:  iaRc =. LoadLocalPath (clThumbsDb ; clMirrorDb)
NB.
NB.   src=. 'c:/thumbsdbs/working2018.tpdb8s'           NB. thumbsplus
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   LoadLocalPath src;trg

'src trg'=. y

NB. fetch thumbsplus data
ds=. sqlopen_psqlite_ src
dat=. sqlreads__ds 'select idPath, name from Path'
rc=. sqlclose__ds ''

NB. convert path delimiters
paths=. allwhitetrim&.> <"1 '\/' charsub ;1{1{dat
ids=. , ;0{1{dat

NB. insert in LocalPath
dt=. sqlopen_psqlite_ trg
rc=. sqlinsert__dt 'LocalPath';(;:'LocalPathID LocalPath');<ids;<paths
rc [ sqlclose__dt ''
)

LoadMirrorXrefDb=: 3 : 0

NB.*LoadMirrorXrefDb v--  loads  new  mirror  cross  reference
NB. database.
NB.
NB. monad:  btNameCnts = LoadMirrorXrefDb (clMirrorDr ; clThumbsDb ; clMirrorDb)
NB.
NB.   sdr=. 'c:/smugmirror/mirror'                      NB. mirror directory
NB.   src=. 'c:/thumbsdbs/primary2018.tpdb8s'           NB. thumbsplus db
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db' NB. mirror db
NB.   dmp=. 'c:/smugmirror/documents/xrefdb/locoxref.txt'  NB. online local pairs
NB.   mcf=. 'c:/smugmirror/documents/xrefdb/manlocoxref.txt'  NB. missing local online corrections
NB.   mli=. 'c:/smugmirror/documents/xrefdb/locomiss.txt'   NB. missing local images report
NB.   ssp=. 'c:/smugmirror/documents/xrefdb/suspects.txt'   NB. suspicious image pair report
NB.   urp=. 'c:/smugmirror/documents/xrefdb/upratesum.txt'  NB. upload summary report
NB.   LoadMirrorXrefDb sdr;src;trg;dmp;mcf;mli;ssp;urp

'sdr src trg dmp mcf mli ssp urp'=. y

NB. thumbs db must exist
if. 0 e. fexist src do. smoutput 'thumbs db missing' return. end.

smoutput 'Creating mirror.db ...'
tabs=. CreateMirrorXrefDb trg

smoutput 'Loading path and albums ...'
LoadLocalPath src;trg
LoadAlbum sdr;trg

smoutput'Loading keywords ...'
'img iax'=. MirrorImageXrTables sdr
kw=. }. ((0{img) i. <'Keywords') {"1 img
LoadOnlineKeyword trg;<kw
```

```
smoutput 'Loading Images and cross references ...'
LoadOnlineImage trg;<img
LoadImageAlbumXr trg;<iax
LoadImageKeywordXr trg;<img

smoutput 'Checking existence of downloaded images ...'
UpdateLocalPresent trg

smoutput 'Matching online and local files with same file names ...'
MatchOnlineLocal mcf;src;trg

smoutput 'Matching online and local files after [] bracket suffix removal ...'
MatchOnlineNoBrackets mcf;src;trg

smoutput 'Matching manual local online files corrections ...'
MatchManual mcf;src;trg

smoutput 'Dumping current online local image pairs as TAB delimited text -> ',dmp
smoutput DumpLocalImageNatural dmp;trg

smoutput 'Writing missing images report -> ',mli
0 0$MissingImagesReport mli;trg

smoutput 'Writing suspicious pairs report -> ',ssp
0 0$SuspiciousPairReport ssp;trg

smoutput 'Writing upload summary report -> ',urp
```

```
0 0$MirrorStatReport urp;trg
)

LoadOnlineImage=: 3 : 0

NB.*LoadOnlineImage v-- load (OnlineImage) table.
NB.
NB. monad:  iaRc =. LoadOnlineImage (clDb ;< btclDat)

'trg dat'=. y
dat=. <"1 |: }. dat [ head=. 0{dat

NB. convert numeric columns
pos=. head i. ;:'ArchivedSize Altitude OriginalHeight OriginalWidth'
dat=. ((SQLITE_NULL_INTEGER_psqlite_&".&>) L: 1 [ pos{dat) pos} dat
pos=. head i. ;:'Latitude Longitude'
dat=. ((SQLITE_NULL_FLOAT_psqlite_&".&>) L: 1 [ pos{dat) pos} dat

dt=. sqlopen_psqlite_ trg
rc=. sqlinsert__dt 'OnlineImage';head;<dat
rc [ sqlclose__dt ''
)

LoadOnlineKeyword=: 3 : 0

NB.*LoadOnlineKeyword v-- load (OnlineKeyword) table.
NB.
NB. monad:  LoadOnlineKeyword (clDb ; <blclKeywords)
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   keys=. 'boo;hoo;foo' ; 'yoo;key;me'               NB. keyword lists
NB.   LoadOnlineKeyword trg;<keys

'trg keys'=. y

if. #keys=. <;._1 (;';' ,&.> keys) -. ' ' do.
  NB. insert unique keywords and frequencies
  'kw fq'=. ofreq s: keys
  dt=. sqlopen_psqlite_ trg
  rc=. sqlinsert__dt 'OnlineKeyword';(;:'Keyword KeyFrequency');<(5&s: kw);fq
  rc [ sqlclose__dt ''
else.
  0
end.
)

LocalFrOnline=: 4 : 0

NB.*LocalFrOnline  v--  local  file  names from  online keys  and
NB. filenames.
NB.
NB. Local file names are  derived  from online names by prefixing
NB. the name with the image key, followed by a lower case base 36
NB. case mask, followed  by the online  file name. All blanks are
NB. replaced with hyphen '-' characters.
NB.
NB. dyad:  bclObj LocalFrOnline clAlbumKey
NB.
NB.   db=. sqlopen_psqlite_ 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   db LocalFrOnline 'BFs8q4'

NB. fetch album
'invalid (AlbumKey)' assert 0 < #y

NB. use of sqlparm sometimes leaves databases open
NB. dat=. sqlparm__x LocalFile_sql;SQLITE_TEXT_psqlite_;y
sql=. ('/?/''',y,'''') changestr LocalFile_sql
dat=. sqldict__x sql

nofiles=. 0$<''
if. 0 e. #&> 1 {"1 dat do. nofiles return. end.

NB. table columns !(*)=. ImageKey OnlineImageFile LocalPath
(0 {"1 dat)=. boxopen&.> 1 {"1 dat

if. #ImageKey do.
  pfx=. ImageKey ,&.> '-' ,&.> (b36casemask ImageKey) ,&.> '-'
  (<MIRRORHEAD) ,&.> LocalPath ,&.> '/' ,&.> pfx ,&.> ' -'&charsub&.> OnlineImageFile
else.
  nofiles
end.
)
```

```
LocalFromDir=: 3 : 0

NB.*LocalFromDir  v--  local  files with directory match (y)  and
NB. file match (x).
NB.
NB. A  utility  for browsing  local  files.  Used  from resolving
NB. mismatches.
NB.
NB. monad:  btcl =. LocalFromDir clPathPat
NB.
NB.   LocalFromDir 'Petroglyph'
NB.
NB. dyad:  btcl =. clFilePat LocalFromDir clPathPat
NB.
NB.   'hand' LocalFromDir 'Petroglyph'

'' LocalFromDir y
:
if. #x do. d=. ' and a.name like ''%',x,'%''' else. d=. '' end.
sql=. 'select * from Thumbnail a inner join Path b on a.idPath=b.idPath where b.name like ''%', y ,'%''',d
db=. sqlopen_psqlite_ PRIMARYTEST
r=. sqldict__db sql
sqlclose__db ''
'path name'=. (1 {"1 r) #~ (0 {"1 r) = <'name'
name ,. path
)

MatchManual=: 3 : 0
```

```
NB.*MatchManual  v--  insert  manual  online   local  image  pair
NB. corrections.
NB.
NB. This verb reads a  TAB delimited  file  of manual online  and
NB. local image  pairings and inserts valid pairs into the mirror
NB. cross reference database. Manual  corrections  are  necessary
NB. for online images with file names that do not follow patterns
NB. handled by (MatchOnlineLocal) and (MatchOnlineNoBrackets).
NB.
NB. monad:  iaRc =. MatchManual (clFixFile ; clThumbsDb ; clMirrorDb)
NB.
NB.   src=. 'c:/thumbsdbs/primary2018.tpdb8s'
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   mcf=. 'c:/smugmirror/documents/xrefdb/manlocoxref.txt'
NB.   MatchManual mcf;src;trg

'mcf src trg'=. y
rc=. 0  NB. success

if.-.fexist mcf do.
  rc=._1 [ smoutput 'Manual correction(s) file missing - skipping adjustments ... ' return.
end.

NB. manual corrections - remove commented lines -- cannot comment header
mc=. ];._2 tlf (read mcf) -. CR
mc=. parsetd ctl mc #~ -. 0 {"1 (,:'--') E. mc
```

```
mc=. }. mc [ hd=. 0 { mc
'lp lf of'=. hd i. ;:'LocalPath LocalImageFile OnlineImageFile'

NB. look up path IDs and online image keys
NB. (*)=. LocalPathID LocalPath OnlineImageFile ImageKey LocalThumbID
dt=. sqlopen_psqlite_ trg
du=. sqldict__dt 'LocalPath'
dv=. sqldict__dt 'OnlineImageFile,ImageKey from OnlineImage'
rc=. sqlclose__dt''

(0{"1 du)=. 1 {"1 du
'missing manual local path(s)' assert (#LocalPath) > pos=. LocalPath i. lp {"1 mc
lpid=. pos { LocalPathID

(0{"1 dv)=. 1 {"1 dv
'invalid manual online image(s)' assert (#OnlineImageFile) > pos=. OnlineImageFile i. of {"1 mc
olik=. pos { ImageKey

NB. look up thumbsplus image ID - used as LocalThumbID
ds=. sqlopen_psqlite_ src
du=. sqldict__ds 'idThumb as LocalThumbID, name as LocalImageFile, idPath as LocalPathID from Thumbnail'
rc=. sqlclose__ds''

NB. use symbols - local file list is large
(0{"1 du)=. 1 {"1 du
LocalImageFile=. s: LocalImageFile
```

```
'invalid manual local image(s)' assert (#LocalImageFile) > pos=. LocalImageFile i. s: lf {"1 mc
lifp=. pos

NB. match on path id and local file name
pos=. (LocalPathID ,. i.~ LocalImageFile) i. lpid ,. lifp
'missing manual local path image pairs(s)' assert pos < #LocalImageFile
ltid=. pos { LocalThumbID

NB. insert manual local online pairs in LocalImage table
cln=. ;:'LocalPathID LocalThumbID LocalImageFile ImageKey'
du=. lpid;ltid;(lf {"1 mc);<olik
dt=. sqlopen_psqlite_ trg
rc=. sqlinsert__dt 'LocalImage';cln;<du
rc [ sqlclose__dt ''
)

MatchOnlineLocal=: 3 : 0

NB.*MatchOnlineLocal v-- match online files with local files.
NB.
NB. Online files are, with few  exceptions, JPG files while local
NB. files are mostly  TIF files. This verb exactly matches online
NB. file names with  local names. This exact  match picks up over
NB. 1500  images which is  better than  I  expected. Matching all
NB. online  images  to  local  files  will be  an ongoing  little
NB. project.
NB.
NB. verbatim: online files are:
```

```
NB.
NB.  1. often renamed
NB.  2. retyped (tif,nef -> jpg)
NB.  3. can derive from many originals (panoramas)
NB.
NB. monad: iaRc = MatchOnlineLocal (clThumbsDb ; clMirrorDb)
NB.
NB.   src=. 'c:/thumbsdbs/primary2018.tpdb8s'
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   mcf=. 'c:/smugmirror/documents/xrefdb/manlocoxref.txt'
NB.   MatchOnlineLocal mcf;src;trg

'mcf src trg'=. y
ds=. sqlopen_psqlite_ src
dt=. sqlopen_psqlite_ trg

NB. local file names and ids !(*)=. idPath idThumb name
ld=. sqldict__ds 'idPath, idThumb, name from Thumbnail'
(0 {"1 ld)=. 1 {"1 ld
rc=. sqlclose__ds ''
local=. justfile&.> name

NB. online ImageKey and file names !(*)=. ImageKey OnlineImageFile
td=. sqldict__dt 'ImageKey, OnlineImageFile from OnlineImage'
(0 {"1 td)=. 1 {"1 td

if. fexist mcf do.
```

```
  NB. manual corrections
  mc=. }. mc [ hd=. 0 { mc=. readtd2 mcf
  of=. hd i. <'OnlineImageFile'
  NB. remove any manual correction files
  bm=. -.OnlineImageFile e. of {"1 mc
  OnlineImageFile=. bm # OnlineImageFile
  ImageKey=. bm # ImageKey
end.

OnlineImageFile=. justfile&.> OnlineImageFile

NB. local/online exact matches
pos=. local i. OnlineImageFile [ rc=. 0
if. #mch=. pos -. #local do.

  NB. insert matched
  NB. NOTE: renamed columns - first row original - next new name
  NB. idPath       idThumb
  NB. LocalPathID  LocalThumbID
  clns=. ;:'LocalPathID LocalThumbID LocalImageFile ImageKey'
  dat=. (mch{idPath);(mch{idThumb);(mch{name);<(pos < #local)#ImageKey
  dat=. 'LocalImage';clns;<dat
  rc=. sqlinsert__dt dat

end.

rc [ sqlclose__dt ''
)

MatchOnlineNoBrackets=: 3 : 0

NB.*MatchOnlineNoBrackets v-- match online files with local files
NB. without [] suffix.
NB.
NB. For many years  I attempted  to pair online images with local
NB. images by affixing a bracketed suffix with the online  key in
NB. square [] brackets. When I  migrated many  of my images  from
NB. Flickr  to SmugMug I  stripped  the bracket  suffix from many
NB. file names.  By  removing  bracket  suffixes  more  than  700
NB. matches  can  be added  to  what  (MatchOnlineLocal)  already
NB. found.
NB.
NB. monad:  iaRc = MatchOnlineNoBrackets (clThumbsDb ; clMirrorDb)
NB.
NB.   src=. 'c:/thumbsdbs/primary2018.tpdb8s'
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   fix=. 'c:/smugmirror/documents/xrefdb/manlocoxref.txt'
NB.   MatchOnlineNoBrackets fix;src;trg

'mcf src trg'=. y
ds=. sqlopen_psqlite_ src
dt=. sqlopen_psqlite_ trg

NB. files with []'ed and nonbracketed versions
ovb=. BracketOverlaps ds
```

```
NB. local file names and ids !(*)=. idPath idThumb name
ld=. sqldict__ds 'select idPath, idThumb, name from Thumbnail where name like ''%[%]%'''
(0 {"1 ld)=. 1 {"1 ld
rc=. sqlclose__ds ''
if. 0=#name do. rc=. sqlclose__dt '' return. end.

NB. strip [] suffixes from local file names
local=. alltrim@('['&beforestr)&.> name

NB. removing suffixes might create duplicates
bm=. ~:local
if. -.1 e. bm do. rc=. sqlclose__dt '' return. end.
local=.   bm # local
name=.    bm # name
idPath=.  bm # idPath
idThumb=. bm # idThumb

NB. remove any previously matched images !(*)=. LocalImageFile
td=. sqldict__dt 'select LocalImageFile from LocalImage where LocalImageFile like ''%[%]%'''
(0 {"1 td)=. 1 {"1 td
bm=. -.name e. LocalImageFile
bm=. bm +. -.local e. justfile&.> LocalImageFile
local=.   bm # local
name=.    bm # name
idPath=.  bm # idPath
idThumb=. bm # idThumb
```

```
if. 0=#name do. rc=. sqlclose__dt '' return. end.

NB. online ImageKey and file names !(*)=. ImageKey OnlineImageFile
td=. sqldict__dt 'select ImageKey, OnlineImageFile from OnlineImage'
(0 {"1 td)=. 1 {"1 td

OnlineImageFile=. justfile&.> OnlineImageFile

if. fexist mcf do.
  NB. manual corrections
  mc=. }. mc [ hd=. 0 { mc=. readtd2 mcf
  of=. hd i. <'OnlineImageFile'
  NB. remove any manual corrections
  bm=. -.OnlineImageFile e. justfile&.> of {"1 mc
  OnlineImageFile=. bm # OnlineImageFile
  ImageKey=. bm # ImageKey
end.

NB. removing suffixes can over insert when both the []'ed
NB. and non bracketed files exist - the long term fix
NB. is to find and rename the mismatched files
bm=. -.OnlineImageFile e. ovb
OnlineImageFile=. bm # OnlineImageFile
ImageKey=. bm # ImageKey
if. 0=#OnlineImageFile do. rc=. sqlclose__dt '' return. end.

NB. local/online exact matches after suffixes removed
```

```
pos=. local i. OnlineImageFile [ rc=. 0
if. #mch=. pos -. #local do.

  NB. insert matched
  NB. NOTE: renamed columns - first row original - next new name
  NB. idPath       idThumb
  NB. LocalPathID  LocalThumbID
  clns=. ;:'LocalPathID LocalThumbID LocalImageFile ImageKey'
  dat=. (mch{idPath);(mch{idThumb);(mch{name);<(pos < #local)#ImageKey
  dat=. 'LocalImage';clns;<dat
  rc=. sqlinsert__dt dat

end.

rc=. sqlclose__dt ''
)

MirrorImageXrTables=: 3 : 0

NB.*MirrorImageXrTables v--  scan  manifest/realdate  files  and
NB. return image and imagealbum xref tables.
NB.
NB. monad:  ((<btclImage), <btclXref) =. MirrorImageXrTables clRoot
NB.
NB.   'img xref'=. MirrorImageXrTables 'c:/SmugMirror/Mirror'

NB. z locale !(*)=. dirtree
img=. 2#<0 0$a:
```

```
if. #files=. dirtree (tslash2 y),'manifest-*.txt' do.

  NB. for each manifest file there is a real date file
  'missing real date file(s)' assert fexist rdfiles=. RealDateFrManifest 0 {"1 files

  NB. all manifest/real date files have same layout get headers
  head=. 0{readtd2 ;0{0{files
  'image album'=. head i. ;:'ImageKey AlbumKey'
  'missing (ImageKey, AlbumKey) column(s)' assert (image,album) < #head

  NB. read all manifest files
  img=. ; }.@readtd2&.> 0 {"1 files

  NB. image album cross reference - insure unique rows
  iax=. ~. ((image,album){head) , (image,album) {"1 img

  NB. reduce on image key - there are duplicate
  NB. images in the python smugpyter directories
  b=. head ~: <'AlbumKey'
  img=. b #"1 img [ head=. b # head
  img=. img #~ ~: image {"1 img

  NB. insert new column in table schema order
  b=. 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
  img=. b (#^:_1)"1 img
  head=. b (#^:_1) head
  NB. rename FileName -->> OnlineImageFile and insert new column
  head=. (<'OnlineImageFile') (head i. <'FileName')} head
  head=. (<'RealDate') (I. -.b)} head

  NB. update column positions
  'image imdate'=. head i. ;:'ImageKey RealDate'

  NB. merge in real dates
  rdhead=. 0{readtd2 ;0{0{rdfiles
  'rdimage rddate'=. rdhead i. ;:'ImageKey RealDate'
  'missing (ImageKey, RealDate) column(s)' assert (rdimage,rddate) < #rdhead
  rd=. ; }.@readtd2&.> rdfiles
  pos=. (image {"1 img) i. rdimage {"1 rd
  if. +./b=. pos < #img do.
    pos=. b # pos
    img=. (b # rddate {"1 rd) (<pos;imdate)} img
  end.

  NB. image table and xref
  (<head,img),<iax
else.
  img
end.
)

MirrorStatReport=: 3 : 0

NB.*MirrorStatReport v-- mirror database upload summary report.
NB.
NB. monad:  cl =. MirrorStatReport (clReportFile ; clMirrorDb)
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   urp=. 'c:/smugmirror/documents/xrefdb/upratesum.txt'
NB.   MirrorStatReport urp;trg

'urp trg'=. y

cnts=. TableCounts trg
'all yrc'=. WeekUploadRates trg
gst=. NonEmptyGalleryStats trg
wy=. weeksinyear {. 6!:0 ''
gp=. 2#LF

r=.LF,'mirror.db upload summary for: ',timestamp ''

r=.r,gp,'Online image count:              ', ": ;((0{cnts) i. <'OnlineImage'){ 1{cnts

r=.r,LF,'Overall weekly cy upload rate:   ', ":ur=. {: {: all
r=.r,LF,'Overall cy upload count:         ', ":  _2 {  {: all
r=.r,LF,'Estimated overall cy uploads:    ', ":1 round ur * wy
r=.r,LF
r=.r,LF,'This year weekly cy upload rate: ', ":yr=. {: {: yrc
r=.r,LF,'This year cy upload count:       ', ":  _2 {  {: yrc
r=.r,LF,'Estimated year cy uploads:       ', ":1 round yr * wy
```

```
r=.r,gp,'Nonempty gallery count statistics'
r=.r,LF,ctl gst

r=.r,gp,'Table row counts'
portchars '' [ cchars=. 9!:6 ''
r=.r,LF,ctl ":cnts

r=.r,gp,'First five nondivisible galleries'
r=.r,LF,ctl ":5 {. NotDivisible trg

NB. write upload rate summary report
r [ (toHOST r) write urp [ 9!:7 cchars
)

MissingImagesReport=: 3 : 0

NB.*MissingImagesReport v-- missing local images report.
NB.
NB. monad:  clReportFile =. MissingImagesReport (clMissingFile ; clMirrorDb)
NB.
NB.   mli=. 'c:/smugmirror/documents/xrefdb/locomiss.txt'
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   MissingImagesReport mli;trg

'mli trg'=. y

NB. switch to portable box characters
portchars '' [ cchars=. 9!:6 ''
```

```
dt=. sqlopen_psqlite_ trg
dat=. ctl ": sqlreads__dt MissingLocalFiles_sql
9!:7 cchars [ sqlclose__dt ''

NB. write report text
r [ (r=. toHOST dat) write mli
)

NonEmptyGalleryStats=: 3 : 0

NB.*NonEmptyGalleryStats v-- summary statistics of nonempty galleries.
NB.
NB. monad:  ctStats =. NonEmptyGalleryStats clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   NonEmptyGalleryStats trg

NB. require 'data/sqlite' !(*)=. ImageCnt sqlopen_psqlite_ sqldict__dt sqlclose__dt
dt=. sqlopen_psqlite_ y
d=. sqldict__dt AlbumImageCount_sql
(0 {"1 d)=. 1 {"1 d
dstat ImageCnt [ sqlclose__dt ''
)

NotDivisible=: 3 : 0

NB.*NotDivisible v-- list albums with  image counts that are  not
NB. divisible by three and five.
NB.
NB. This  verb  finds  albums  with  image counts  that  are  not
NB. divisibe by  three  and  five.  This  weird  requirement  was
NB. motivated by  how the SmugMug iPhone  App displays galleries.
NB. It breaks the images into rows of three or five. I don't like
NB. incomplete terminal rows.
NB.
NB. monad:  bt =. NotDivisible clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   NotDivisible trg
NB.
NB. dyad:   bt =. ia NotDivisible clMirrorDb
NB.
NB.   4 NotDivisible trg

3 5 NotDivisible y
:
NB. !(*)=. ImageCnt AlbumName
(0 {"1 d)=. 1 {"1 d=. AlbumImageCount y

NB. works for integer (x) without common divisors
'common factor(s)' assert -.pwcf x
a=. x , */x

b=. *./ 0 < r=. a |"0 1 ImageCnt
c=. (a *"1 >. ImageCnt %"0 1 a) - ImageCnt
c=. (/: |."1 b # c) { (<"0 b # ImageCnt,.c) ,. b # AlbumName
c ,~ '[Count]';(<"0 a),<'[Album Name]'
)

RealDateFrManifest=: 3 : 0

NB.*RealDateFrManifest v-- real date files from manifest files.
NB.
NB. monad:  pl =. RealDateFrManifest blclManifestFiles
NB.
NB.   RealDateFrManifest 0 {"1 dirtree jpath '~temp/manifest-*.txt'

NB. assumes use of (dirtree) - hence / path delimiter
paths=. tslash2&.> '/'&beforelaststr&.> y
paths ,&.> 'realdate-'&,&.> (#'manifest-') }.&.> '/'&afterlaststr&.> y
)

SuspiciousPairReport=: 3 : 0

NB.*SuspiciousPairReport v--  suspicious  local  and  online file
NB. pairings.
NB.
NB. Online images  are almost  entirely jpg files  with some gif,
NB. png and avi files.  Typically I create local tif files that I
NB. upload as jpg. Hence, any computed local online pairing  that
NB. does not pair files of these types is questionable.
NB.
NB. Suspicious files need to be manually inspected and faulty
NB. pairings should be corrected in the (manlocoxref.txt) file.
NB.
NB. monad:  clSuspectFile =. SuspiciousPairReport (clSuspectFile ; clMirrorDb)
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   ssp=. 'c:/smugmirror/documents/xrefdb/suspectpairs.txt'
NB.   SuspiciousPairReport ssp;trg

'ssp trg'=. y

NB. get local online file pairing
dt=. sqlopen_psqlite_ trg
du=. sqldict__dt LocalOnlineFile_sql
rc=. sqlclose__dt''

NB. !(*)=. LocalPath LocalImageFile OnlineImageFile RealDate
(0 {"1 du)=. 1 {"1 du

NB. suspect file extension mask
ext=. tolower@justext&.> LocalImageFile
b=. ext e. (~.ext) -. ;:'jpg tif gif png avi'

NB. suspicious file pairing table
h=. ;:'LocalPath LocalImageFile OnlineImageFile RealDate'
du=. h , /:~ b # LocalPath ,. LocalImageFile ,. OnlineImageFile ,. RealDate
```

```
NB. switch to portable box characters
portchars '' [ cchars=. 9!:6 ''
du=. ctl ": du
9!:7 cchars

NB. write suspect report
r [(r=. toHOST du) write ssp
)

TableCounts=: 3 : 0

NB.*TableCounts v-- table row counts.
NB.
NB. monad:  bt =. TableCounts clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'
NB.   TableCounts trg

NB. require 'data/sqlite' !(*)=. sqlclose__dt sqlopen_psqlite_ sqlsize__dt sqltables__dt
dt=. sqlopen_psqlite_ y
if. #tabs=. sqltables__dt '' do. cnts=. tabs ,: sqlsize__dt&.> tabs else. cnts=. 0 0$a: end.
cnts [ sqlclose__dt ''
)

UpdateLocalPresent=: 3 : 0
```

```
NB.*UpdateLocalPresent v--  set (LocalPresent) in (Album) to 1 if
NB. all local images exist.
NB.
NB. monad:  iaRc =. UpdateLocalPresent clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   UpdateLocalPresent trg

NB. get albums !(*)=. AlbumKey AlbumName
rc=. 0
dt=. sqlopen_psqlite_ y
al=. sqldict__dt 'AlbumKey, AlbumName from Album'
(0 {"1 al)=. 1 {"1 al
if. 0=#AlbumKey do. rc [ sqlclose__dt '' return. end.

bok=. 0 #~ #AlbumKey
for_pos. i.#AlbumKey do.
  ak=. ;pos{AlbumKey
  an=. ;pos{AlbumName
  smoutput 'Checking ',an,' ...'
  files=. dt LocalFrOnline ak
  b=. fexist files
  if. (0 e. b) *. 0 < #files do.
    smoutput 'missing local album sample file(s)'
    smoutput >files # -.b
  else.
    NB. all album files present
    bok=. 1 pos} bok

    NB. NOTE: faster to scan all files and issue one update
    NB. WARNING: this method leaves (mirror.db) open despite
    NB. being closed in all cases - this is not a problem when
    NB. run from (buildmirror.bat) that terminates the entire
    NB. process and closes (mirror.db).
    NB. rc=. sqlparm__dt UpdateLocalPresent_sql;SQLITE_TEXT_psqlite_;ak
    NB. if. rc ~:0 do. break. end.

  end.
end.

NB. issue one update
if. 1 e. bok do.
  sql=. UpdateLocalPresent_sql,'(',(}.;',' ,&.> quote&.> bok#AlbumKey),')'
  rc=. sqlcmd__dt sql
end.

rc [ sqlclose__dt ''
)

WeekUploadRates=: 3 : 0

NB.*WeekUploadRates v-- computes weekly image upload rates.
NB.
NB. monad:  (ft ; ft) =. WeekUploadRates clMirrorDb
NB.
NB.   trg=. 'c:/smugmirror/documents/xrefdb/mirror.db'  NB. smugmug mirror
NB.   WeekUploadRates trg

NB. require 'data/sqlite' !(*)=. sqlopen_psqlite_ sqldict__dt sqlclose__dt
NB. get uploads per year and images per year - sorted on year
dt=. sqlopen_psqlite_ y
sql=. '/{{date}}/UploadDate' changestr UploadRateCount_sql
uy=. sqldict__dt sql
sql=. '/{{date}}/RealDate' changestr UploadRateCount_sql
iy=. sqldict__dt sql
sqlclose__dt ''

NB. all uploads per year - first biases rates - last incomplete
(0 {"1 uy)=. 1 {"1 uy
ImageCnt=. }. ImageCnt [ Year=. ".&> }. Year
LastCnt=. {: ImageCnt [ LastYear=. {:Year
uy=. }: Year ,. ImageCnt ,. ImageCnt % weeksinyear Year
wn=. {: weeknumber today 0
uy=. uy , LastYear , LastCnt , LastCnt % wn

NB. images per year rates for common years - first year NULL
cy=. Year
(0 {"1 iy)=. 1 {"1 iy
ImageCnt=. }. ImageCnt [ Year=. ".&> }. Year
b=. Year e. cy
ImageCnt=. b # ImageCnt [ Year=. b # Year
LastCnt=. {: ImageCnt [ LastYear=. {:Year
iy=. }: Year ,. ImageCnt ,. ImageCnt % weeksinyear Year
iy=. iy , LastYear , LastCnt , LastCnt % wn

NB. rate tables
(<uy),<iy
)


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


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


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


NB. trims all leading and trailing white space
allwhitetrim=: ] #~ [: -. [: (*./\. +. *./\) ] e. (9 10 13 32{a.)"_


antimode=: 3 : 0

NB.*antimode v-- finds the least frequently occurring  item(s) in
NB. a list.
NB.
NB. monad:  ul =. antimode ul
NB.
NB.   antimode ?.500#100
NB.   antimode ;:'blah blah blah yada yada wisdom'


if. 0 < # y =. ,y do.    NB. no antimodes for null lists
  f =. #/.~ y            NB. nub frequency
  (~. y) #~ f e. <./ f   NB. lowest frequency items
else. y
end.
)

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

b36casemask=: 3 : 0

NB.*b36casemask v-- case mask encoded as base 36 string.
NB.
NB. This verb must match the python (case_mask_encode)  function.
NB. See the macro (base36_py) for details.
NB.
NB. monad:  blcl =. b36casemask blclKeys
NB.
NB.   b36casemask  <'7ZdVjvN'
NB.
NB.   keys=. <;._1 ' bKq6rXc 6wpcfgH SCVLN82 6bb47N7 gQhBGBw GSM47GW z78dtXv hH8C2Sz M58pxN7 JvZb6gp'
NB.   b36casemask keys
```

```
c=. b36fd #. 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' e.~ > ,y

NB. suppress leading zeroes to match python
c=. ' ' (<(I. '0' = 0 {"1 c);0)} c
(<"1 c) -.&.> ' '
)

NB. base 36 from decimal: b36fd 3x^90
b36fd=: 36x&#.@('0123456789abcdefghijklmnopqrstuvwxyz'&i.)^:_1

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

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
)

charsub=: 4 : 0

NB.*charsub v-- single character pair replacements.
NB.
NB. dyad:  clPairs charsub cu
NB.
NB.   '-_$ ' charsub '$123 -456 -789'

'f t'=. ((#x)$0 1)<@,&a./.x
t {~ f i. y
)

NB. character table to newline delimited list
ctl=: }.@(,@(1&(,"1)@(-.@(*./\."1@(=&' '@])))) # ,@((10{a.)&(,"1)@]))

NB. YYYYMMDD to YYYY MM DD - see long document
datefrnum=: 0 100 100&#:@<.

NB. deviation about mean
dev=: -"_1 _ mean

dstat=: 3 : 0

NB.*dstat v-- descriptive statistics
NB.
NB. monad: ct =. dstal nl
NB.
NB.   dstat  ?.1000#100
NB.
NB. dyad:  ct =.  faRound dstat nl
NB.
NB.   0.1 dstat  ?.1000#100

0.0001 dstat y
:
t=. '/sample size/minimum/maximum/1st quartile/2nd quartile/3rd quartile/first mode'
t=. t , '/first antimode/mean/std devn/skewness/kurtosis'
min=. <./
max=. >./
t=. ,&':  ' ;._1 t
v=. $,min,max,q1,median,q3,({.@mode2),({.@antimode),mean,stddev,skewness,kurtosis
t,. ": x round ,. v , y
)

NB. boxes UTF8 names
fboxname=: ([: < 8 u: >) ::]

NB. erase files - cl | blcl of path file names
ferase=: 1!:55 ::(_1:)@(fboxname&>)@boxopen

NB. 1 if file exists 0 otherwise
fexist=: 1:@(1!:4) ::0:@(fboxname&>)@boxopen

insqltd=: 4 : 0

NB.*insqltd v-- insert btcl table into sqlite table.
NB.
NB. dyad:  iaRc =. baObj insqltd (clSqlTable ;< btclData)
NB.
NB.   db=. sqlopen_psqlite_ jpath '~temp\dntest.db'
NB.   dat=. readtd2 'C:\Users\john2000\j64-807-user\temp\dn_dv.txt'
NB.   db insqltd 'dntest';<dat

'tab dat'=. y
st=. sqltables__x ''
'missing database table' assert st e.~ <tab
cl=. sqlcols__x tab
'missing table column(s)' assert cl e.~ 0{dat
sqlinsert__x tab;(0{dat);< <"1 |: }.dat
)

NB. extracts the extension from qualified file names
justext=: ''"_`(] #~ [: -. [: +./\. '.'&=)@.('.'&e.)

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

NB. kurtosis
kurtosis=: # * +/@(^&4)@dev % *:@ssdev

NB. mean value of a list
mean=: +/ % #

NB. median value of a list
median=: -:@(+/)@((<. , >.)@midpt {  /:~) ::_:
```

```
NB. mid-point
midpt=: -:@<:@#

mode2=: 3 : 0

NB.*mode2 v-- finds  the  most frequently occurring item(s) in  a
NB. list.
NB.
NB. monad:  ul =. mode2 ul
NB.
NB.   mode2 ?.500#100
NB.   mode2 ;:'I do what I do because I am what I am'

if. 0 < # y =. ,y do.     NB. null lists have no modes
  f =. #/.~ y             NB. nub frequency
  (~. y) #~ f e. >./ f    NB. highest frequency items
else. y
end.
)

NB. like (freq) but results in descending frequency
ofreq=: [: (([: < [: \: [: ; 1 {  ]) { &.> ]) ~. ; #/.~

NB. parse TAB delimited table text - see long document
parsetd=: [: <;._2&> (a.{~9) ,&.>~ [: <;._2 [: (] , ((10{a.)"_ = {:) }. (10{a.)"_) (13{a.) -.~ ]

NB. portable box drawing characters
portchars=: [: 9!:7 '+++++++++|-'"_ [ ]
```

```
NB. 1 if (x) has at least one pair with common factor(s) - see long document
pwcf=: 1 < [: >./ [: +/ [: +./@e.&>/~ 0 -.&.>~ [: <"1 [: ~."1 q:

NB. first quartile
q1=: median@((median > ]) # ]) ::_:

NB. third quartile
q3=: median@((median < ]) # ]) ::_:

NB. quotes character lists for execution
quote=: ''''&,@(,&'''')@(#~ >:@(=&''''))

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

NB. read TAB delimited table files - faster than (readtd) - see long document
readtd2=: [: <;._2&> (a.{~9) ,&.>~ [: <;._2 [: (] , ((10{a.)"_ = {:) }. (10{a.)"_) (13{a.) -.~ 1!:1&(]`<@.(
>..>32&>@(3!:0)))


reb=: 3 : 0

NB.*reb v-- removes redundant blanks - leading, trailing multiple
NB.
NB. monad:  reb cl
NB. dyad:  ua reb ul
```

```
' ' reb y
:
y=. x , y
b=. x = y
}.(b*: 1|.b)#y
)

NB. round (y) to nearest (x) (e.g. 1000 round 12345)
round=: [ * [: (<.) 0.5 + %~

NB. skewness
skewness=: %:@# * +/@(^&3)@dev % ^&1.5@ssdev

NB. session manager output
smoutput=: 0 0 $ 1!:2&2

NB. sum of square deviations (2)
ssdev=: +/@:*:@dev

NB. standard deviation (alternate spelling)
stddev=: %:@:var

timestamp=: 3 : 0

NB.*timestamp v-- formats timestamp as dd mmm yyyy hr:mn:sc
NB.
NB. monad:  cl =. timestamp zu | nlTime
NB.
NB.   timestamp ''               NB. empty now
NB.   timestamp 2007 9 16        NB. fills missing
NB.   timestamp 1953 7 2 12 33

if. 0 = #y do. w=. 6!:0'' else. w=. y end.
r=. }: $ w
t=. 2 1 0 3 4 5 {"1 [ _6 [\ , 6 {."1 <. w
d=. '+++::' 2 6 11 14 17 }"1 [ 2 4 5 3 3 3 ": t
mth=. _3[\'   JanFebMarAprMayJunJulAugSepOctNovDec'
d=. ,((1 {"1 t) { mth) 3 4 5 }"1 d
d=. '0' (I. d=' ') } d
d=. ' ' (I. d='+') } d
(r,20) $ d
)

NB. appends trailing line feed character if necessary
tlf=: ] , ((10{a.)"_ = {:) }. (10{a.)"_

NB. converts character strings to CRLF delimiter
toCRLF=: 2&}.@:;@:((13{a.)&,&.>@<;.1@((10{a.)&,)@toJ)

NB. converts character strings to host delimiter
toHOST=: toCRLF
```

```
NB. converts character strings to J delimiter LF
toJ=: ((10{a.) I.@(e.&(13{a.))@]}  ])@:(#~ -.@((13 10{a.)&E.@,))

today=: 3 : 0

NB.*today v-- returns todays date.
NB.
NB. monad:  ilYYYYMMDD =. today uu
NB.
NB.   today 0    NB. ignores argument
NB.
NB. dyad:  iaYYYYMMDD =. uu today uu
NB.
NB.   0 today 0

3&{.@(6!:0) ''
:
0 100 100 #. <. 3&{.@(6!:0) ''
)

todayno=: 3 : 0

NB.*todayno v-- convert dates to day numbers, converse  (todate).
NB.
NB. WARNING: valid only for  Gregorian dates after  and including
NB. 1800 1 1.
NB.
NB. monad:  todayno ilYYYYMMDD
NB.
NB.   dates=. 19530702 19520820 20000101 20000229
NB.   todayno 0 100 100 #: dates
NB.
NB. dyad:  pa todayno itYYYYMMDD
NB.
NB.   1 todayno dates

0 todayno y
:
a=. y
if. x do. a=. 0 100 100 #: a end.
a=. ((*/r=. }: $a) , {:$a) $,a
'y m d'=. <"_1 |: a
y=. 0 100 #: y - m <: 2
n=. +/ |: <. 36524.25 365.25 *"1 y
n=. n + <. 0.41 + 0 30.6 #. (12 | m-3),"0 d
0 >. r $ n - 657378
)

tolower=: 3 : 0

NB.*tolower v-- convert to lower case.
NB.
NB. monad: cl =. tolower cl

x=. I. 26 > n=. ((65+i.26){a.) i. t=. ,y
```

```
($y) $ ((x{n) { (97+i.26){a.) x}t
)

NB. appends trailing / iff last character is not \ or /
tslash2=: ([: - '\/' e.~ {:) }. '/' ,~ ]

valdate=: 3 : 0

NB.*valdate v-- validates lists or tables of YYYY MM DD Gregorian
NB. calendar dates.
NB.
NB. monad:  valdate il/it
NB.
NB.   valdate 1953 7 2
NB.   valdate 1953 2 29 ,: 1953 2 28  NB. not a leap year

s=. }:$y
'w m d'=. t=. |:((*/s),3)$,y
b=. */(t=<.t),(_1 0 0<t),12>:m
day=. (13|m){0 31 28 31 30 31 30 31 31 30 31 30 31
day=. day+(m=2)*-/0=4 100 400|/w
s$b*d<:day
)

NB. var
var=: ssdev % <:@#
```

```
NB. day of week - see long document:  weekday 1953 7 2
weekday=: 7 | 3 + todayno

weeknumber=: 3 : 0

NB.*weeknumber v-- gives the year and weeknumber of date.
NB.
NB. A week  belongs to  a year iff 4 days of  the week  belong to
NB. that year. From J library (dates).
NB.
NB. Note: does not generalize to array arguments.
NB.
NB. verbatim: see
NB.
NB. http://www.phys.uu.nl/~vgent/calendar/isocalendar.htm
NB.
NB. monad: il =. weeknumber ilYYYYMMDD
NB.
NB.   weeknumber 2005 1 2
NB.   weeknumber 1953 7 2


yr=. {.y
sd=. 1 ((i.~weekday){]) ((<:yr),.12,.29+i.3),yr,.1,.1+i.4
wk=. >.7%~>: y -&todayno sd
if. wk >weeksinyear yr do.
  (>:yr),1
elseif. wk=0 do.
```

```
  (,weeksinyear)<:yr
elseif. do.
  yr,wk
end.
)

NB. gives number of weeks in year - see long document
weeksinyear=: 52 + [: +./"1 [: [ 4 = [: weekday (2 2$1 1 12 31) ,"0 1/~ ]

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

NB. writes tables as TAB delimited LF terminated text - see long document
writetd2=: ] (1!:2 ]`<@.(32&>@(3!:0)))~ [: ([: (] , ((10{a.)"_ = {:) }. (10{a.)"_) [: }.@(,@(1&(,"1)@(-.@(*
>..>./\."1@(=&' '@])))) # ,@((10{a.)&(,"1)@])) [: }."1 [: ;"1 (a.{~9)&,@":&.>) [


yyyymondd=: 3 : 0

NB.*yyyymondd v-- today in (yyyymondd;hrmnss) format.
NB.
NB. Yet another date format verb. We can never have enough!
NB.
NB. monad:  (clDate ; clTime) =. yyyymondd uuIgnore

fmt=.'r<0>2.0'
months=. _3 [\ '   janfebmaraprmayjunjulaugsepoctnovdec'
```

```
'yy mn dd'=. 3{.now=. 6!:0''
date=. (":yy),(mn{months),,fmt (8!:2) dd
time=. }.;':' ,&.> fmt (8!:0) _3 {. now
date;time
)
```

```
NB.POST_MirrorXref post processor.

smoutput IFACE=:  (0 : 0)
NB. (MirrorXref) interface word(s): 20220527j154226
NB. -------------------------------
NB. BuildMirror            NB. backup/create/load mirror
NB. CheckRealDates         NB. check real dates
NB. DumpLocalImageNatural  NB. dump (LocalImage) as TAB delimited text
NB. LocalFromDir           NB. local files with directory match (y) and file match (x)
NB. MirrorStatReport       NB. mirror database upload summary report
NB. MissingImagesReport    NB. missing local images report
NB. SuspiciousPairReport   NB. suspicious local and online file pairings
)

cocurrent 'base'
coinsert  'MirrorXref'
```

Index