

```

1 # the python module ADI 2 CSV; tag list defined in library 3/31/24 JRJ
2 # not usually a reversible process (print name: reVersionXX)
3 import ADIsensaiMod20a as cSub #ideas next
4 from dataDriver2a import tgStr
5 from dataDriver2a import tpStr
6 from dataDriver2a import tgEcpts
7 from dataDriver2a import myDat
8 #starting specs
9 # 1. convert stream format to CSV (from <T:L(:D)>V); flag for packed vs un-
10 # 2. one fixed, 1 goto, QRv assumed (in groups) inner. enhanced miniMy file
11 # 3. (who) No what (where- certs) (when-how) in data, ~3 major groups
12 # maybe output that way; split up in theirs last vs yours
13 # 5a. pack slightly un- preformat. item 6 moved down
14 # 8. at end of list loading new index, might be sparse or overflow.
15 # 9. per line arrays: tags, values, types, partial sequence
16
17 def fnd_tg(s,ln): #find tag subr; pos 1 < than tag, see #1 version subr
18     if s=='': return (False,1,s)
19     s = '<' + s #not found -1; needs < for tag too
20     p2 = ln.find(s) #the read line global
21     f1 = (p2 > -1) #ignores L... ARRL & QRZ have different formats
22     return (f1,p2+1,s) #tuple back, p2' of tag start!
23
24 # if len(lfrrt)==2:
25 #     tag1,val = lfrrt[0],lfrrt[1] #test for ARRL comment //
26 # elif len(lfrrt)==1: #error, finished
27
28 #     tag1,val = lfrrt[0],' #same names as in header
29 # else:
30 #     cSub.wrt_1linA(lfrrt,-137,q) #show this error
31

```

```

32 #         if val != '':                                #remove a single trailing space? what if
    just front?
33 #         if not cSub.Npadd(val):  val = val.strip() #try ARRL remove next
34 #         bk2=cSub.rem_ARL(val)      #2 has the discard, if any
35 #         if bk2[0]:
36 #             val = bk2[1]                                #replace val
37
38 #         else:  pass    #log this as an error? no
39 def datQualt(lf):      #from loop 1
40     'new subroutine for clarity- not necess faster?'
41     tag,v1 = 'Nil',''    #set a default
42     if len(lf)==2:
43         tag,v1 = lf[0],lf[1]        #test for ARRL comment // later
44     elif len(lf)==1:              #error, finished
45         print(45,'1 item: |',lf,'| ',id1)
46         tag,v1 = lf[0],''          #same names as in header
47     else:
48         cSub.wrt_1linA(lf,-48,q)    #show this error
49     if v1 != '':                  #remove a single trailing space? what if just
    front?
50         if cSub.Paddd(v1):  v1 = v1.strip() #try ARRL comment remove next
51         bk=cSub.rem_ARL(v1)        #2 has the discard, if any
52         if bk[0]:
53             v1 = bk[1]              #replace val
54             cSub.wrt_1linA('  ARRL comment: '+bk[2],54,g) #log comment if 0 true, dL?
55 #     else:  pass                    #log this as an error? no
56     return (tag,v1)
57
58 #         if tag2 in tgStr:                #the matching tags; dictionary created?
59 #             fdp1 = tgStr.index(tag2)+1    #1 and up, avoids the 0 flag
60 #         elif tag2 in myDat:              #block for the my-data array (new)
61 #             fdp1 = myDat.index(tag2)+1    #identify repeated data (u/c) in file

```

```

62 #             fdp1 = -1*fdp1
63 #             elif tag2 in tgEcpts:             #only members QSO_len & app_; here as tgEcpts
        only2
64 #             fdp1 = tgEcpts.index(tag2)
65 #             fdp1 += 500             #picked plus; now 2 of them!
66 #             else:             #opens up 0?
67 #             fdp1 = 0 #not found, 2 reasons: didn't get or no room
68 #             if tag2 == None:     pass     #more logic? print stmt?
69 #             else:             #build conversion table dynamically?
70 #             pass
71 def clasTg(tg):             #from loop 2
72     'new beginning of April 24- for clarity only. both being tested'
73     if tg in tgStr:             #the matching tags; dictionary created?
74         fd = tgStr.index(tg)+1     #1 and up, avoids the 0 flag
75     elif tg in myDat:             #block for the my-data array (new)
76         fd = myDat.index(tg)+1     #identify repeated data (u/c) in file
77         fd = -1*fd
78     elif tg in tgEcpts:             #only members QSO_len & app_; here as tgEcpts
        only2
79         fd = tgEcpts.index(tg)
80         fd += 500             #picked plus; now 2 of them!
81     else:             #opens up 0?
82         fd = 0 #not there. 2 reasons: didn't get or no room
83         if tg == None:     pass     #more logic? print stmt? or null?
84         else:             #build conversion table dynamically?
85             print(85,'that (tag) not found: ',tg) #, value: ',valr[id2],id2)
        #logfile? Yes
86     return (fd)
87
88 #def std_case(k,lin):             #moved to library!
89 #     ls,vl,tg,rs,tp = [],[],[],[],[]     #null lists for below; save found tag indexes
90

```

```

91 #BEGIN PROGRAM: start row (cr after eor); setup the lookup array
92 j0,sell2 = len(tgStr),cSub.sell2 #like 49-0; +XX to prt references
93 jSav,last_dup,fli=j0,'',cSub.rtnPflgs(2) #was True,True,True,False,True,False
94 prtSupMsg,spltEnds,verbs,repLen,goStd,noCms =
    fli[0],fli[1],fli[2],fli[3],fli[4],fli[5]
95
96 nwMd = cSub.ovrrFlgs(2,sell2) #call new manual update subr
97 if crs < 0: pass #add interm var!
98 else: #reset the flags listed above; copy sel from library
99     crs = rtnPflgs(2) #define flags
100     prtSupMsg,spltEnds,verbs,repLen,goStd,noCms =
        crs[0],crs[1],crs[2],crs[3],crs[4],crs[5]
101     print( 100,'command changes accepted, code:',nwMd[1]) #provide notice of change?
        add code conf?
102     sell2 = nwMd[0] #copy local from global update
103 kyvl,tags,valr,outIm,linebuf,tOn = [None]*j0,['']*j0,['']*j0,'','','2400' #/
104 maiCols = cSub.crt_dic1(tgStr,tpStr) #tag string & type; created dictionary- use?
105 print ( ' begin new ADI --> CSV conversion run-----prog2',cSub.tmStr() )
106 print (106,' new internal lookup dict1: ',maiCols) #log tag? long list
107 nwLn,hdrdt,recNo,ver = '\n',True,1,'26dF'
108
109 vnum = input('enter file number XY of hamOutput/ now input? ')
110 vname = 'hamOutput'+str(vnum)+'.adi'
111 lname = 'hamLogfil'+str(vnum)+'.txt' #usage should match 1st pgm & library subr
112 oname = 'hamComfil'+str(vnum)+'.csv'
113 qname = 'hamDivOut'+str(vnum)+'.txt' #supplemental output
114
115 f = open(vname,'r') #datafile; logging is next
116 g = open(lname,'a') #open or create
117 h = open(oname,'a') #all append
118 q = open(qname,'a') #2nd output
119 cSub.wrt_1linA('log file: '+lname,-119,q) #h is next

```

```

120 cSub.wrt_1linA('output file: '+oname, -120, g)
121                                     #also see around line 190
122 cSub.fmt_out(tgStr, h, False)       #write out the tag string, or buf_mgr
123
124 # middle: start a while loop on the input data rows; assume 1 /n
125 print (125, 'Program2 version:', ver, 'to CSV- from the QRZ verbose format') #new
    section; buffer already 'cleared' but...
126 while True:                        #readline loop: needs a break (as do programmers)
127 # goal: read-in either format. App_ data out to a different stream
128     cSub.line = f.readline()        #would assemble lines last, with pgm
    flag... replace line wt sum
129     if not cSub.line: break         #see the book on why a line can be boolean
130
131 # send all output in a single (long) line; the header is without an extra space for
    now
132 # needs a field skip for app_data_ used in other programs
133 # not possible to manipulate data- but select the best sets, stuck with users concept
134 # process header... add capability for multiple lines; no translation
135     if fnd_tg('eoh', cSub.line)[0] or fnd_tg('EOH', cSub.line)[0]: #read until <eoh> /
    <EOH> is received
136
137     cSub.wrt_1linA(' in user header: '+str(recNo), -137, g)
138     if verbs:
139         if linebuf == '': pass      #not quite right?
140         else: cSub.line = linebuf+cSub.line
141         linebuf = ''                #reset, then continue
142     cSub.wrt_1line(142, g)           #before split
143     hdrdt = False                   #set header flag to data
144     mnyFd = cSub.line.split('<')     #leading space and trailing newline
145     if spltEnds: mnyFd = mnyFd[1:-1] #see program flag above- 4testing only
    (off) of previous attempt
146     cSub.wrt_1linA(mnyFd, -146, g)  #2 extra fields, beg & end: split #1

```

```

147
148     ct = len(mnyFd)-1                #always 1 less, typ.
149     hi = ct
150     while ct >= 0:                   #start col counter hi
151         id = hi-ct
152         lftrt = mnyFd[ct].split('>')    #right is value. split #2 header
153         tag1,val = lftrt[0],lftrt[1]    #header has no values?
154         ts = cSub.ele_unpak1(tag1,val)  #check flag? split3; log tag1, print val?
155         so = '|' + mnyFd[id] + '|'
156         cSub.wrt_1linA(so, -156, q)
157         ct += -1                     #reversed QRZ format (verbose)
158     recNo += 1
159
160     #data records; if <EOH> is received, flip flag & stop (a 2nd file started?). more
work
161     elif fnd_tg('eor',cSub.line)[0] or fnd_tg('EOR',cSub.line)[0]: #line. during this
(store all) into a line array.
162         if verbs:                     #see comments above
163             if linebuf == '': pass
164             else: cSub.line = linebuf+cSub.line
165             linebuf,hdrdt = '',True
166             cSub.wrt_1line(-166,g)      #DATA SECT., extra space?
167             if noCms: cSub.line = cSub.line.replace(',', '~') #remove commas here, use ~
instead
168             if goStd: rSt=cSub.std_case(168,cSub.line,g) #translate and standardize;
check rs!
169             if goStd and (len(rSt) < 5): #why 5?
170                 print (170,'standardize failed- no change dot line')
171             else: cSub.line = rSt
172
173     mnyFd = cSub.line.split('<')        #temp, split #1; leading space

```

```

174         if spltEnds: mnyFd = mnyFd[1:-1]    #split2 produces a leading blank &
        trailing 'eo h/r +'
175 # 6. line algorithm:
176     #read in a whole line; then split by the < char- manyFields; check lengths
177     #split each element next by >, you get the 'value' coming off- on right
178     #residue left with TLD ala colons; split these with subr- tested2; as error check
179
180     cSub.wrt_1linA(mnyFd, -180, q)
181     cSub.wrt_1linA(' user record: '+str(recNo), 181, g)
182 # process a record; +at the same time, collect the tag array (pass 1)
183     jSv2 = jSav+3+jSav+jSav//2 #set array size at 250% wow!
184     #show app_ fields that are skipped and their values; std+spcl
185     #show dates for all, store the destination for each in a col if available
186     #app_ /APP_ received, echo to a different stream; could have continues in front?
187     tags, valr=['']*jSv2, ['']*jSv2        #column algorithm next; init @line
188 # col ID marks the owner of the broken dataset- index. place in order?
189 #-----go through col by col of input (twice?); show translations
190     ct1 = len(mnyFd)-1                    #and trailing newline?
191     h1 = ct1                             #save width
192     while ct1 >= 0:                       #ends ~30 lines, col counter
193         id1 = h1-ct1                     #going up
194         lftr = mnyFd[id1].split('>')      #where to put new split#2
195         bk3 = datQualt(lftr)             #new
196         tag1, val = bk3[0], bk3[1]
197         ts = cSub.ele_unpak1(tag1, val)    #split #3, tuple out- save in an array
198     #6b. report issues, after that L can be discarded?
199         if ts[0]:
200             fms = 'error re format '+'|'+str(ts[1])+ '|' #must exist, so skip to
        else
201             cSub.wrt_1linA(fms, -201, g)
202         else:                             #load row arrays here,
        positions 1 & 4, d?

```

```

203         if ts[1]=='time_on':  tOn = ts[4]           #save the on-time to
compute length- always
204         if repLen and ts[1]=='time_off':
205             bk=cSub.getQS0len(tOn,ts[4])           #compute L, hmm,mm,f-
tuple back
206             tsp = (False,'QS0_len',len(bk[1]),ts[3],bk[1])
207             #ignore flag; redefine bk[1]? can't redef a tuple
208             tags[id1],valr[id1] = tsp[1],tsp[4]
209         else:  tags[id1],valr[id1] = ts[1],ts[4]    #discard L finally, loads
backwards!
210             #if ct1 == hi:      pass          #or 0; watch for QS0_start & QS0_len
(QS0_end)
211             #else:  print (ts)
212             cSub.wrt_1linA(ts,-212,g)            #log some samples, ts is nice
213             ct1 += -1                            #end of loop1
214
215             #make results of loop avail: arrays from above; loop ends
216             dupStr = ''                          #restart, loop2
217 #----- method I. best, recNo still counts lines. ARRL l/c QRZ caps issue?
218             #processing 1 line at a time; show variations in tag names from database.
219             ct2 = len(tags)-1                     #hope valr has the same length?
220             inOut = [None]*(ct2+1)               #backwards- presize. big enuf? or just []
221             h2 = ct2                             #start loop from the avail subset tags in
the database
222             while ct2 >= 0:  #go through the stored data tags. backwards, but doesn't
matter
223                 id2 = h2-ct2
224                 tag2 = tags[id2]                 #look up each tag, or create dictionary- tgStr.
use in and index
225                 fdp1 = clasTg(tag2)              #new
226                 if fdp1 <0:  #repeating data      don't print again if dupe
227                     print(227,'ADI tag processed',tag2,valr[id2],id2,'repeating data?')

```



```

228         dupStr = cSub.buf_mgr(tag2,dupStr,False,True)           #remember the last
1's printed?
229         dupStr = cSub.buf_mgr(valr[id2],dupStr,False,True)      #push right again
230         elif fdp1 >0: pass                                     #stored data already
231         else: pass                                             #think empty, not found? 0 done above
232             #get list of 0's                                   or extra tags in data stream; or 2's
233             inOut[id2] = fdp1                                   #store that data code
234     #the values) can be accessed later (a subset of tgStr)
235     pass                                                       #if not in dictionary: placed a -1
236     ct2 += -1                                                  #end of ct2 (first) loop
237
238     cSub.wrt_1linA(inOut,-238,q)                                #output a read record report?
239     print (239,' processed',h1,'tags & values, maximum:',jSv2)
240     if dupStr == last_dup:
241         if prtSupMsg: cSub.wrt_1linA(' duplicate suppressed',241,g) #same, do
nothing
242         prtSupMsg = False
243     else:
244         cSub.wrt_1linA(dupStr,-244,g)
245         prtSupMsg = True
246     #outInA = cSub.revAset(None,inOut,True)                    #reverse the list?
247     last_dup = dupStr                                          #remember...
248
249 #----- #get valr from stored if present.
250     ct3,outIm = jSav-1,''                                     #output width- image of CSV
251     h3 = ct3
252     while ct3 >= 0:                                           #go thru all the out-data slots
253         up = h3-ct3                                           #the index, up or id3
254         curOutTag = tgStr[up]                                  #left to right, or up
255         if curOutTag in tags:                                  #find index in tags for value next
256             id = tags.index(curOutTag)                        #output value; use tags.index() to buffer
257             v1 = valr[id]

```

```

258         if v1 == '': outIm = cSub.buf_mgr('',outIm,False,True)      #advance
259         else:         outIm = cSub.buf_mgr(v1,outIm,False,True)      #push
right
260     else:             #not in list, data WILL be lost
261         if outIm == '': outIm = str(recNo) #to start the buffer
262         else: outIm = cSub.buf_mgr('',outIm,False,True)      #advance
263         #log any input tag (and values) skipped, to file q
264         ct3 += -1      #post dec; end of loop 3
265         #the output array is probably less than the database size
266         h.write(outIm) #output the buffer
267         h.write(nwLn)  #needed one
268         recNo += 1
269     else:             #ignore blank lines, or long run-on
270         if cSub.line == '': pass #log if this happens?
271         elif cSub.line == nwLn: pass #blank fields (lines) are/were skipped
272         else:         #skip blanks eventually
273             if recNo==1:
274                 print (274,'flags:
',cSub.det_newln(cSub.line),cSub.det_newlnA(cSub.line),'title(1):' )
275                 cSub.wrt_1line(275,g) #1 is line location
276                 recNo = 2
277             else:     #partials: read all after <eoh> & stop
after reading <eor>.
278                 if verbs: #assemble a line here...
279                     line1 = ' '+cSub.rem_newln(cSub.line) #remove nwLn
on the end
280                     linebuf = cSub.buf_mgr(line1,linebuf,False,False) #add right?
281                     #print (281,'after, buffer:',linebuf,'|') #restarts
@next 'line' after <eor> just read.
282                 else: pass
283                 #print (283,'continue:',recNo,nwLn,cSub.line,nwLn )
284 # print out fields, scan list, and data types if any present

```

```
285     outIm = cSub.buf_mgr(' ',outIm,True,False)    #clear buffer for rerun?
286 #     recNo += 1                                     #moved to accomodate verbose formats
287     pass                                             #end of main (input rows) loop
288 f.close()    #pass 2 breaks down the data?
289 g.close()    #log file
290 h.close()    #output file
291 q.close()    #diversion file (odd tags or data)
292 # problem, long data streams rx without getting < or > occasionally.
293
294 # can preconvert miniMy to a CSV file, if needed
295 #method2. see #1 above- this not used...
296 # array above TAGS (lc) is target; determine where None's stop?
297 # write out std tag positions tgStr & names first
298 # go col by col of the output array J image; use outIm
299 # use 'my' output of #1 as the input
300 #     A. go thru all stored tags for the read col, blank ignore
301 #     B. if a match, print that value with a ',' too
302 # in either case, go to the next col or if at end
303 # write out the final physical file h in append & close
304
305 # next step or idea: do 2 files in a sequence. later, allow app_data save
306 # if data has a comma, enclose it in " quotes, but problematic? also <,>
```