

```
' xref.bas - a cross reference utility for Liberty Basic and similar
' Author: Ken B Smith - Wattle Downs, Auckland Jun 2011
',
',
' This program is free software: you can redistribute it and/or modify
' it under the terms of the GNU General Public License as published by
' the Free Software Foundation, either version 3 of the License, or
' (at your option) any later version.
',
' This program is distributed in the hope that it will be useful,
' but WITHOUT ANY WARRANTY; without even the implied warranty of
' MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
' GNU General Public License for more details.

' You should have received a copy of the GNU General Public License
' along with this program. If not, see <http://www.gnu.org/licenses/>.
' Date: 26/13/11
' 27/6/11 Based on IQPDF text conversion module
' 28/6/11 Removed snag with showing words in quotes in readascii
' 28/6/11 Tidied up the alignment on listing
' 04/7/11 Added Functions to show as 'F' and Function defined variable
s as 'f' - for Will !
```

```
global maxinput ' Number of files accepted on filelist
maxinput = 40
dim filedata$(maxinput,2)
' (x,0) - filename ! (x,1) - size ! (x,2) = date
dim info$(40,40)
dim BASname$(maxinput)
' These are decode variables
' dim results(2,maxinput) ' hold the top 20 results from upto 20 file
dim wordarray$(5000,2)
' the structure will be strange : 0 = word : 1 = line/line/line/line/line
' - line numbers as 5 byte dec : 2 is variable type G etc
' linenumber + 11111 + (len(word) * 11) - ready for main program directly.
dim stats(10)
' stats(0) = total line count / stats(1) = totalwords / stats(2) = total word count / stats(3) = Highest line count on a word
' 6 bytes hex = totalwords / 4 bytes hex largest word count / 4 bytes hex highest line count on a word
global wordfound ' flag 0 = false
global wordcount ' word count on a line
global maxwordcount ' words in a document
global totalwords, maxperline
```

```
global maxperline
global targetword$ ' word to be inserted
global targetline
' line number to go in the string if not already there
global fileIs$ ' What file are we working on ?
global Stimerval ' Start timer
global Etimerval ' End timer
global BASTxt$, BASIdx$, BASFile$, BAScount
global filesize
global filedate$
global t$
global LBkey$ ' all the Liberty reserved words and variables
global legalchr$ ' string holding chr$ allowed
global wordplace
global longestword
global metatagword
global metaid$ ' meta word ID
global funcdefline ' holder for line number where a function defined
global Qlisting ' True or False for listing on top of Xref
global Qnumbers ' do you want line numbers with that ?

LBkey$ =
" abs acs append as asc and asn atn backgroundcolor$ beep binary bmpbu
tton bmpsave boolean button call "
LBkey$ = LBkey$ +
" callback calldll case checkbox chr$ close cls colordialog combobox c
omboboxcolor$ commandline$ "
LBkey$ = LBkey$ +
" confirm cos cursor data date$ dechex$ defaultdir$ dialog dialog_fs d
ialog_modal dialog_nf dialog_nf_modal "
LBkey$ = LBkey$ +
" dialog_nf_fs dim displayheight displaywidth dll do drives$ dword dum
p else end eof err error err$ eval "
LBkey$ = LBkey$ +
" eval$ exit exp field filedialog files fontdialog end for foregroundc
olor$ function get gettrim global "
LBkey$ = LBkey$ +
" gosub goto graphics graphics_fs graphics_fs_nsb graphics_nsb graphic
s_nf_nsb graphicbox groupbox hbmpp "
LBkey$ = LBkey$ +
" hexdec hwnd if inkey$ inp input input$ inputto$ instr int joy1x joy1
y joy1z joy1button1 joy1button2 joy2x "
LBkey$ = LBkey$ +
" joy2y joy2z joy2button1 joy2button2 kill left$ len let line listbox
listboxcolor$ loadbmp locate loc "
LBkey$ = LBkey$ +
```

```
" lof log long loop lower$ lprint lr maphandle max menu midipos mid$ m
in mkdir mod name next nomainwin "
LBkey$ = LBkey$ +
" notice on open or out output platform$ playmidi playwave popupmenu p
rint printerdialog printerfont$ prompt "
LBkey$ = LBkey$ +
" ptr put radiobutton randomize read readjoystick redim rem restore re
sume return right$ rmdir rnd run "
LBkey$ = LBkey$ +
" scan seek select short sin sort space$ spreadsheet sqr statictext st
ep str$ stopmidi struct stylebits sub "
LBkey$ = LBkey$ +
" tab tan text textbox textboxcolor$ texteditor texteditorcolor$ text_
fs text_nsb text_nsb_ins then time$ "
LBkey$ = LBkey$ +
" timer titlebar to trace trim$ txcount ul ulong unloadbmp until upper
$ upperleftx upperlefty ur ushort using "
LBkey$ = LBkey$ +
" val version$ void wait wend while window windowheight windowwidth wi
ndow_nf winstring word word$ "
legalchr$ =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_'[].$#
" + chr$(34)
```

```
[setup.main.Window]
```

```
'-----Begin code for #main
```

```
nomainwin
WindowWidth = 740
WindowHeight = 150
UpperLeftX=int((DisplayWidth-WindowWidth)/2)
UpperLeftY=int((DisplayHeight-WindowHeight)/2)
```

```
'-----Begin GUI objects code
```

```
call removespaces
call findfiles ' find all BASs in the directory
```

```
TextboxColor$ = "white"
textbox #main.fileisbox, 325, 15, 380, 25
textbox #main.dotbox, 20, 60, 275, 25
textbox #main.timebox, 325, 60, 380, 25
radiobutton #main.NoListing, "No Listing?", [NoListSet], [
NoListReset], 325, 37, 100, 25
radiobutton #main.Listing, "Listing?", [ListSet], [ListReset], 460,
```

```
37, 60, 25
radiobutton #main.Numbers, "With Line Numbers?", [LNSet], [
LNReset], 580, 37, 150, 25

combobox #main.filebox, BASname$(), [selectfile], 20, 12, 275, 250
'-----End GUI objects code

open "XRef2 - Liberty BASIC Cross-
Index Creation
Tool - V2.1 [c] Ken B Smith - June 2011" for window as #main
print #main, "font ms_sans_serif 10"
print #main, "trapclose [quit.main]"
#main.Numbers,"hide"
#main.NoListing, "set"
[main.inputLoop] 'wait here for input event
wait
[NoListSet]
#main.Numbers,"hide"
Qlisting = 0
Qnumbers = 0
wait

[NoListReset]

wait

[ListSet]
#main.Numbers,"show"
Qlisting = 1
Qnumbers = 0
wait

[ListReset]

wait

[LNSet]
Qnumbers = 1
wait

[LNReset]

wait

[selectfile] 'Perform action for the combobox named 'filebox'
```

```
print #main.filebox, "contents? fileIs$"
print #main.filebox, "selectionindex? fileselect"
if fileIs$ = "All Files" then
call allfiles
else
BASFile$ = fileIs$
' filesize = val(info$(fileselect,1))
' filedate$ = info$(fileselect,2)
call start
end if
close #main

end
[quit.main] 'End the program
close #main
end

'-----allfiles-----

sub allfiles ' do all files in list in one go

for xx = 1 to BAScount ' This is the total filecount of "BAS" files
BASFile$ = BASname$(xx)
filesize = val(info$(xx,1))
filedate$ = info$(xx,2)
call start
call pause 1000
next xx

end sub

' ----- start -----
sub start
redim wordarray$(5000,2) ' Flush the word buffer
redim stats(10)
longestword = 0

BASTxt$ = left$(BASFile$,instr(BASFile$,".")) + "bas"
BASIdx$ = left$(BASFile$,instr(BASFile$,".")) + "idx"
Stimerval = time$("ms")
print #main.fileisbox, "Working on - " + BASFile$

Etimerval = time$("ms")
filetime = (Etimerval - Stimerval) / 1000
```

```
call readascii ' read the txt file to make the idx

Etimerval = time$("ms")
filetime = (Etimerval - Stimerval) / 1000
call writeidx
print #main.timebox, BASFile$ + " took " + str$(filetime) +
" secs or " + str$(int(stats(1)/filetime)) + " lines/sec"
call pause 2000
end sub

' ----- writeidx -----
sub writeidx

open BASIdx$ for output as #dbf

' add the originating filename and todays date / time
print #dbf,BASFile$; " X-REF created "; date$(); " "; time$()
print #dbf, ""
print #dbf," DECODE OF Symbology"
print #dbf," -----"
print #dbf," [G] - Variable defined as GLOBAL"
print #dbf," [D] - Variable Dimensioned ($)"
print #dbf," [d] - Variable or Constant used in Dimension Declaration"
print #dbf," [S] - Sub Routine name"
print #dbf,
" [s] - Variable or Constant used in Sub Routine Declaration"
print #dbf," [F] & [f] as above, but for Functions"
print #dbf, ""
' now the word list and line data
longestword = longestword + 2 ' establish minimum spacing
for x = 1 to stats(1)
if wordarray$(x,2) <> "" then
wordarray$(x,0) = "[" + wordarray$(x,2) + "]" " + wordarray$(x,0)
else
wordarray$(x,0) = " " + wordarray$(x,0)
end if
if wordarray$(x,2) = "D" or wordarray$(x,2) = "F" then
wordarray$(x,0) = wordarray$(x,0) + "()"
end if
print #dbf, wordarray$(x,0);space$(longestword - len(
wordarray$(x,0)));wordarray$(x,1)
next x
print #dbf, ""
print #dbf, " ----- "
print #dbf, ""
' Now list out the program for those without line numbers
```

```
if Qlisting = 0 goto [endlist]
targetline = 0
open BASTxt$ for input as #df ' this is the target ascii file
while eof(#df) = 0
line input #df, a$ ' take the whole line and find out what it is
targetline = targetline + 1
if Qnumbers = 1 then
print #dbf, str$(targetline);space$(5 - len(str$(targetline)));" ";
end if
print #dbf,a$
wend
print #dbf, ""
print #dbf, " ----- "
print #dbf, ""
close #df
[endlist]
close #dbf
end sub
```

```
' ----- readascii -----
```

```
sub readascii
totalwords = 0
maxwordcount = 0
maxperline = 0
targetline = 0
funcdefline = 0

open BASTxt$ for input as #df ' this is the target ascii file

while eof(#df) = 0
line input #df, a$ ' take the whole line and find out what it is
targetline = targetline + 1
' print #main.dotbox, "Working on line - ";str$(targetline);
print #main.dotbox,"Working on line - ";using("#####",targetline)
wordcount = 1
metatagword = 0
metaid$ = ""
funcdefline = 0
a$ = trim$(a$) ' get the spaces out
rotateflag = 1 ' used to kill quoted text
if len(a$) < 2 goto [nextline] ' nothing there
```

```
' Now a special case. I have a label in a handler ie #main.gr, "When m
ouseMove [pixelFollow]"
```

```
' I need to subvert the rotateflag and allow the quotes and all within
to be allowed
if instr(left$(a$,1),"#") > 0 then rotateflag = 2
' quotes cancelled for this line

' First remove all the punctuation in a line. This is easier than spli
tting targetword$
for y = 1 to len(a$)
length = len(a$)
z$ = mid$(a$,y,1)
' I need to dump everything between quotes. rotateflag shows beginning
and end of quotes as -1 and + 1 respectively
if asc(z$) = 34 then
a$ = ADDSPACE$(a$,y)
rotateflag = rotateflag * -1 ' show quote flag active
goto [nextchr]
end if

if rotateflag = -1 then ' quote flag is active ?
a$ = ADDSPACE$(a$,y)
goto [nextchr]
end if

if z$ = "'" then
a$ = left$(a$,y-1)
exit for ' lose the rest of the line
end if

if instr(legalchr$,z$) = 0 then ' this is not a legal character so ...
if y = 1 then ' on the start of a line add a space instead
a$ = " " + mid$(a$,2)
goto [nextchr]
end if

if y = length then ' at the end of a line add a space
a$ = left$(a$,length - 1) + " "
goto [nextchr]
end if

a$ = ADDSPACE$(a$,y) ' add a space everywhere else
end if
[nextchr]
next y

' Now we can process a line that only contains words, numbers and keyw
ords
```

```
targetword$ = "dummy" ' give targetword$ a value
while targetword$ <> ""
targetword$ = word$(a$,wordcount)
targetword$ = trim$(targetword$)
t$ = " " + lower$(targetword$) + " "
' set up a temp string in lowercase to look for keywords
' the space either side ensures we don't pick up bits of words
if instr(LBkey$,t$) > 0 then ' this is a keyword
t$ = trim$(t$)
funcdefline = targetline ' keyword ID on this line
if t$ = "global" then
metatagword = wordcount + 1 ' the next word to be actioned
metaid$ = "G" ' metaid$ = G is a global
end if
if t$ = "dim" then
metatagword = wordcount + 1
metaid$ = "D"
end if
if t$ = "sub" then
metatagword = wordcount + 1
metaid$ = "S"
end if
if t$ = "function" then
metatagword = wordcount + 1
metaid$ = "F"
end if

goto [nextword] ' this was a keyword - ignore it
end if

if len(targetword$) = 0 goto [nextword]
' somehow the word was zero length - so ignore that too.

' we have a valid word - file it

call findword

[nextword]
wordcount = wordcount + 1
wend

[nextline]
wend
close #df
end sub
```

```
' ----- find word -----
' Find a targetword$ in array wordarray$(x,y) where 0 = word / 1 = lin
es and make a place for it at wordpos if not there
sub findword
wordfound = 0
wordplace = 0
oldpointer = 0
cpointer = stats(1) / 2 ' stats(1) is the total word count to date
cstep = stats(1) / 4 ' half way and increment n/4

while cstep > 0.25
oldpointer = int(cpointer)
if targetword$ = wordarray$(oldpointer,0) then exit while ' A hit

if targetword$ > wordarray$(oldpointer,0) then
cpointer = cpointer + cstep
else
cpointer = cpointer - cstep
end if
cstep = cstep / 2
wend

wordplace = oldpointer ' hold that number for a moment
if targetword$ = wordarray$(oldpointer,0) then
' We have a hit at oldpointer
wordfound = oldpointer ' hold where we found it
goto [makeword]
else
if targetword$ = wordarray$(oldpointer + 1,0) then
' it could be one below
wordfound = oldpointer + 1
goto [makeword]
end if
end if

' If we get here then the word wasn't found in the existing list

stats(1) = stats(1) + 1 ' stats(1) is total words
wordfound = stats(1) ' that is where we are
wordarray$(wordfound,0) = targetword$
hexline$ = str$(targetline)
hexline$ = space$(5 - len(hexline$)) + hexline$ ' pad to 5 chrs

wordarray$(wordfound,1) = hexline$
maxwordcount = maxwordcount + 1
if len(targetword$) > longestword then longestword = len(
```

```
targetword$) ' print formatting
goto [alldone]

[makeword]
' now add the line number if not already added for this line

hexline$ = str$(targetline)
hexline$ = space$(5 - len(hexline$)) + hexline$ ' pad to 5 chrs

if instr(wordarray$(wordfound,1),hexline$) = 0 then ' not there ?
wordarray$(wordfound,1) = wordarray$(wordfound,1) +
  hexline$ ' add the hex
end if

[alldone]
' now pick up the status of the metaid$ in wordarray$(wordfound,2)
' if the word is on the same line as the definition itself then someth
ing must be done
if funcdefline = targetline and metaid$ <> "" then
' we are on a keyword line

if metatagword = wordcount then
' this is the word directly after a keyword = name
wordarray$(wordfound,2) = metaid$
' give this the capital value if allocated
else
if metaid$ = "G" then
wordarray$(wordfound,2) = metaid$ ' special case
else
wordarray$(wordfound,2) = lower$(metaid$) ' otherwise the lowercase
end if
end if
end if

if wordfound = stats(1) then call sortme ' new entry - sort it
end sub

' ----- Home Brew sort to get around problems in orderi
ng with LB sort cmd -----
'sort wordarray$( ),wordplace,stats(1),0 is what we are replacing
' very nasty - but will do for now
sub sortme
[startsort]
sorted = 1
for sx = stats(1) to 2 step -1
if wordarray$(sx,0) < wordarray$(sx - 1,0) then
```

```
' and wordarray$(sx - 1,0) <> "" then
' swap elements 0 & 1 & 2
temp0$ = wordarray$(sx - 1,0)
temp1$ = wordarray$(sx - 1,1)
temp2$ = wordarray$(sx - 1,2)
wordarray$(sx - 1,0) = wordarray$(sx,0)
wordarray$(sx - 1,1) = wordarray$(sx,1)
wordarray$(sx - 1,2) = wordarray$(sx,2)
wordarray$(sx,0) = temp0$
wordarray$(sx,1) = temp1$
wordarray$(sx,2) = temp2$
sorted = 0
else
sorted = 1
exit for
end if
next sx
if sorted = 0 goto [startsort]

end sub
```

```
' ----- findfiles -----
```

```
sub findfiles
```

```
files DefaultDir$, "*.BAS",info$()
BAScount = val(info$(0,0))
for x = 0 to BAScount
for y = 0 to 2
filedata$(x,y) = info$(x,y)
next y
BASname$(x) = info$(x,0)
next x
BASname$(0) = "All Files"
end sub
```

```
' -----removespaces -----
```

```
sub removespaces
```

```
' remove spaces from filenames in BASname$() and replace with _
```

```
files DefaultDir$, "*.BAS",info$()
for x = 1 to val(info$(0,0))
BASname$(x) = info$(x,0)
newname$ = BASname$(x)
space = instr(newname$, " ")
```

[illegible]

```
#main.filebox 99 139 140
#main.fileisbox 91 179
#main.Listing 95
#main.NoListing 94 106
#main.Numbers 96 105 110 120
#main.timebox 93 189
[alldone] 391 403
[endlist] 219 234
[ListReset] 95 125
[ListSet] 95 119
[LNReset] 96 133
[LNSet] 96 129
[main.inputLoop] 107
[makeword] 372 376 393
[NoListReset] 94 115
[NoListSet] 94 109
[nextchr] 274 279 290 295 300
[nextline] 260 343
[nextword] 330 333 339
[quit.main] 152
[selectfile] 99 137
[setup.main.Window] 75
[startsort] 426 446
0 111 112 122 174 205 207 210 212 219 220 222 241 242 243 244 245 249
255 257 264 287 310 333 351 352 353 359 361 370 374 384 399 429 431 43
4 437 440 446 456 457 458 461 463 470 471 474
0.25 357
1 121 130 160 162 189 203 212 224 225 251 254 259 264 267 269 273 277
283 288 294 314 318 322 326 340 354 355 374 375 382 383 388 389 399 40
0 419 427 428 429 431 432 433 434 435 436 438 442 470 475 491
[d] 10 32 173
100 94
1000 165 182 187
12 99
15 91
150 81 96
[d] 2 25 30 82 83 163 172 202 204 205 209 260 264 289 354 366 409 412
414 428 433 436 439 458
20 92 99
2000 190
25 91 92 93 94 95 96
250 99
275 92 99
325 91 93 94
34 73 271
37 94 95 96
```

```

380 91 93
4 355
[d] 40 24 26
460 95
5 226 386 397
[d] 5000 30 172
580 96
60 92 93 95
740 80
[F] ADDSPACE$() 272 278 298 490 491
a$ 223 228 250 258 260 264 267 268 269 272 278 283 289 294 298 306
[S] allfiles 142 158
[G] BAScount 44 160 456 457
[G] BASFile$ 44 144 161 176 177 179 189 199
[G] BASIdx$ 44 177 196
[D] BASname$() 27 99 161 461 463 471 472 478
[G] BASTxt$ 44 176 221 247
cpointer 354 358 362 364
cstep 355 357 362 364 366
[G] Etimerval 43 181 182 186 187
[D] filedata$() 25 459
[G] filedate$ 46 163
[G] fileIs$ 41 141 144
[G] filesize 45 162
filetime 182 187 189
[S] findfiles 88 453
[S] findword 337 350
[G] funcdefline 54 245 257 312 406
hexline$ 385 386 388 396 397 399 400
hide 105 110
[D] info$() 26 162 163 455 456 459 461 469 470 471
[G] LBkey$ 48 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 310
[G] legalchr$ 49 73 287
length 268 293 294
[G] longestword 51 174 202 212 390
[d] maxinput 23 24 25 27
[G] maxperline 37 38 243
[G] maxwordcount 36 242 389
[G] metaid$ 53 256 315 319 323 327 406 409 411 412 414
[G] metatagword 52 255 314 318 322 326 408
[s] mil 482 485
newname$ 472 473 475 476 478
oldpointer 353 358 359 361 369 370 371 374 375
[S] pause 165 190 482
[G] Qlisting 55 111 121 219
[G] Qnumbers 56 112 122 130 225

```

[S] readascii 184 240
[S] removespaces 87 467
rotateflag 259 264 273 277
[f] Sa\$ 490 491
[G] Stimerval 42 178 182 187
[f] Sy 490 491
set 106
show 120
sorted 427 440 442 446
[S] sortme 419 425
space 473 474 475 476
[S] start 147 164 171
[D] stats() 32 173 189 203 354 355 382 383 419 428
sx 428 429 431 432 433 434 435 436 437 438 439 445
t 483 485
[G] t\$ 47 308 310 311 313 317 321 325
[G] targetline 40 220 224 226 244 251 253 312 385 396 406
[G] targetword\$ 39 304 305 306 307 308 333 359 361 370 374 384 390
temp0\$ 431 437
temp1\$ 432 438
temp2\$ 433 439
[G] totalwords 37 241
[D] wordarray\$() 30 172 204 205 207 209 210 212 359 361 370 374 384 388 399 400 409 412 414 429 431 432 433 434 435 436 437 438 439
[G] wordcount 35 254 306 314 318 322 326 340 408
[G] wordfound 34 351 371 375 383 384 388 399 400 409 412 414 419
[G] wordplace 50 352 369
[S] writeidx 188 194
x 203 204 205 207 209 210 212 213 457 459 461 462 470 471 472 478 479
xx 160 161 162 163 166
y 267 269 272 278 283 288 293 298 301 458 459 460
z\$ 269 271 282 287