

As our emulators and laptops and pen drives set us ever more free to take our QLs and files with us wherever we go, one of the problems which arises is that of ensuring we have the latest copies of our files and documents on whichever QL or emulator we happen to be using at the time.

I often take a QL On A Stick with me when away, work on my QL files using QPC2 and a QXL.WIN on the USB pen drive, then have to make sure any updated files are copied onto my main QXL.WIN at home before I work on them at home. When it's just one file, that's easy enough to remember. When it's a lot of files, it's not so easy unless you are meticulous enough to keep a detailed list of all files changed so that you can make sure you copy the most recent ones. Even then, it is easy to make a mistake.

Somewhat embarrassingly, this happened to me fairly recently with my Q-Dock utility, and it cost me quite some time to undo my mistake, which further delayed release of the program. The problem was that Q-Dock has quite a lot of files and at home I'd worked on a number of changes to the graphics before I went away for the weekend, taking a laptop with me and duly made several more changes.

When I returned, it was a couple of days before I touched the program again and of course I'd forgotten by then what changes I'd made to which files, so I decided to copy the whole lot. Unfortunately, during a "senior moment" I copied things the wrong way round and ended up with the oldest versions of many of the graphics - it took me ages to sort it all out.

So, once the mess was sorted out, it became a priority to think of a way to reduce the risk of that happening again.

Since most QL systems save a file update date value in the headers of files when they are created or updated, I thought it ought to be possible to compare dates of matching files in the same directories on both drives and automatically copy files in the right direction to ensure the latest versions of each file are maintained across both drives.

Synchronising files like this is a common feature of modern computing with its use of mobile technology and you need to ensure that you always have the most recent copies of important files wherever you are. Our requirements are thankfully a bit simpler normally - we don't necessarily have to store files in the "cloud" and other such fairly complex procedures, just across two devices such as a desktop computer and a pen drive or mobile hard disk.

I wrote a little BASIC program which is essentially a two way copier program, which takes file update dates into account to decide whether to copy and update files. It will also create new directories if it spots that new directories have been created.

The program starts off by asking for the names of the two drives to be synchronised, followed by the name of a directory. If you specify a directory name, only that directory and sub-directories are copied. If you make a blank entry for the directory name the entire drive is scanned. A recursive directory tree routine is used for the scanning - the procedure `Extended_DIR` calls itself to scan each sub-directory as it is encountered.

The scan works two ways - it runs through the source drive first and once that's been done, the process is reversed and the second drive is scanned. Files on the source drive which don't exist on the destination drive are copied and the file update date is made the same (to prevent repeated reverse copying on the second pass with swapped drives). If a copy of a file already exists on the destination drive, the file update dates are compared and if the file on the destination drive is older, it is overwritten with a copy of the newer version of the file, and the file update dates are made the same.

The second call to the `Extended_DIR` procedure merely reverses the process - once files have been

checked and copied from source to destination, we need to repeat the process the other way to ensure that if there are more recent versions of some files on the target drive, they can be updated to the other drive too, so the process is fully bi-directional and it doesn't really matter which drive name you enter first..

The program uses Toolkit 2 extensions such as FUPDT and SET_FUPDT, although these are built into SBASIC on SMSQ/E systems and provided by the toolkit software on many disk drives, so the program should work on almost any system which has Toolkit 2 extensions and level 2 directories.

```
100 REMark Synchronise two drives/directories
110 CLS : CLS #0
120 INPUT #0,'Synchronise drive > ';dr$
130 INPUT #0,'With drive > ';tdr$
140 INPUT #0,'Directory (leave blank for all of drive) > ';dir$
150 IF LEN(dir$) > 0 THEN
160   IF dir$(LEN(dir$)) <> '_' THEN dir$ = dir$ & '_'
170 END IF
180 :
190 PRINT'Synchronising one way...'\
200 synced = 0 : REMark how many files copied
210 Extended_DIR dr$,tdr$,dir$
220 PRINT \ 'Synchronising the other way...'\
230 Extended_DIR tdr$,dr$,dir$
240 PRINT synced;' files synchronised in total.'
250 :
260 STOP
270 :
280 DEFine PROCedure Extended_DIR (fdrive$,tdrive$,direc$)
290   LOCAl loop,ch,d$,fp,n$
300   ch = FOP_DIR (fdrive$&&direc$) : REMark open channel to directory
310   IF ch < 0 THEN RETURN : REMark unable to open directory
320   fp = 14 : REMark file position in directory for filename
330   REPEAT loop
340     BGET #ch\fp : IF EOF(#ch) THEN CLOSE #ch : EXIT loop
350     GET #ch,d$ : REMark get directory entry name
360     IF LEN(d$) > 0 THEN
370       REMark a directory entry length of 0 may be a deleted file
380       BGET #ch\fp-9: REMark file type byte
390       IF CODE(INKEY$(#ch)) = 255 THEN
400         REMark Directory - may need to create this in destination
410         IF FTEST(tdrive$&d$) <> 0 THEN
420           PRINT' Creating ';d$;'...' : MAKE_DIR tdrive$&d$
430         END IF
440         REMark follow the directory tree
450         Extended_DIR fdrive$,tdrive$,d$ : REMark enter directory
460       ELSE
470         REMark File - does it exist on target?
480         BGET #ch\fp+38 : updt1 = GET_LONG(#ch)
490         IF FTEST(tdrive$&d$) = -7 THEN
500           REMark file does not exist on destination, so copy
510           PRINT'Copying ';d$;'...' : COPY fdrive$ & d$ TO tdrive$ & d$
520           REMark ensure both copies have same update date
530           SET_FUPDT \ tdrive$&d$,updt1
540           synced = synced+1
550         ELSE
```

```

560      REMark Exists, but only copy if file on destination is older
570      updt2 = FUPDT( \ t$&d$)
580      IF updt2 < updt1 THEN
590          REMark destination is older, so replace with source copy
600          PRINT'Copying ';d$;'...' : COPY_O fdrive$&d$ TO tdrive$&d$
610          REMark ensure both copies have same update date
620          SET_FUPDT \ tdrive$&d$,updt1
630          synced = synced+1
640      END IF
650  END IF
660  END IF
670  END IF
680  fp = fp + 64
690  END REPEAT loop
700 END DEFINE Extended_DIR

```

N.B. uses a DJToolkit extension called GET_LONG in line 480 so requires use of DJToolkit as it stands.

However, if you don't have DJToolkit installed, this is easily changed to use a common Toolkit 2 extension instead. Simply change line 480 to use the GET command from Toolkit 2 instead:

```

480      BGET #ch\fp+38 : GET #ch,updt1

```