



TurboForth V1.0

Introduction

Presented here are some File IO helper words for use with TurboForth V1.0. Please note that the syntax of the word FILE is expected to change in V1.1 to facilitate much simpler use of run-time file IO characteristics (currently it is very difficult to use the word FILE with dynamic (at run time) data (e.g. file names etc.))

The words given here are:

- * **COPY-FILE** - Copies a DV80 file to another file.
- * **BLK>FILE** - exports a range of disk blocks to a DV80 file.
- * **FILE>BLK** - imports a DV80 file into disk blocks
- * **CPYBLK** - copies a range of disk blocks either to the same block file (to facilitate moving of blocks) or to a different blocks file.

Note that the code that follows below is a *suite* of words that should be loaded together. For efficiency, they use common data buffers, and common error checking routines. Therefore, they cannot be loaded individually, they must be loaded together (i.e. linked with the `-->` word).

COPY-FILE (-)

Description

Copies a DV80 called DSK2.EXPORT to (currently) CLIP. This word developed primarily to provide write access to the Windows clipboard via Classic99. For example, after exporting a range of Blocks to a DV80 in the Classic99 emulator, it is possible to immediately export them to the 'Windows side' by pasting them into the clip board. As coded, this code does exactly that.

Usage

Place in a block of your choice. When loaded, simply enter COPY-FILE from the command line.

```
: CHKIN  IF ." Cant open input file"      ABORT THEN ;
: CHKOUT IF ." Cant open output file"     ABORT THEN ;
: CHKRD  IF ." Cant read from input file" ABORT THEN ;
: CHKWRT IF ." Cant write to output file" ABORT THEN ;
CREATE BUF80 80 ALLOT
S" DSK2.EXPORT 8DI0VS" FILE CFI
S" CLIP. DV80O" FILE CFO
: COPY-FILE
  CFI #OPEN CHKIN  CFO #OPEN CHKOUT
  BEGIN CFI #EOF?  0= WHILE
    BUF80 CFI #GET ABORT" Error reading input file"
    BUF80 COUNT CFO #PUT ABORT" Error writing to output"
  REPEAT
  CFI #CLOSE CFO #CLOSE ;
.( COPY-FILE loaded) -->
```

BLK>FILE (startblock endblock -)**Description**

Exports blocks from the block file currently in **USE** as ASCII text to the DV80 file named in the code (currently DSK2.EXPORT).

Usage

From the command line, enter:

startblock endblock BLK>FILE

E.g. **11 29 BLK>FILE**

The above example will export the contents of the blocks 11 to 29 inclusive to DSK2.EXPORT.

```
S" DSK2.EXPORT DV80SO" FILE EXP  : SZ ZEROS ! ;
: BAR S" --BLOCK-----" 2DUP >R >R DROP 8 + SWAP -1 SZ
  N>S 0 SZ ROT SWAP CMOVE S"  " EXP #PUT DROP R> R> EXP #PUT ;
: BLK>FILE ( start end -- )
  EXP #OPEN CHKOUT
  1+ SWAP DO I BAR DROP
    I BLOCK 16 0 DO
      DUP BUF80 64 VMBR  BUF80 64 -TRAILING DUP
      IF EXP #PUT ELSE 2DROP S"  " EXP #PUT THEN DROP 64 +
    LOOP DROP
  LOOP
  EXP #CLOSE ;
.( BLK>FILE Loaded) -->
```

FILE>BLK (block -)**Description**

Imports a DV80 text file called DSK2.IMPORT into the blocks system. The file is imported beginning at the block specified, and continuing for as many blocks as required. Lines greater than 64 characters are truncated to 64 characters.

Usage

From the command line, enter:

<block> FILE>BLK

e.g. **21 FILE>BLK**

```
0 VALUE LC  0 VALUE BLKNUM  0 VALUE BLKADDR
S" DSK2.IMPORT DV80I" FILE IMP  : CLRBUF BUF80 66 32 FILL ;
: NEW-BLK FLUSH BLKNUM BLOCK TO BLKADDR 1 +TO BLKNUM ;
: READ-LINE CLRBUF BUF80 IMP #GET ABORT" Cant read file" ;
: PUT-LINE BLKADDR BUF80 1+ 64 VMBW 1 +TO LC 64 +TO BLKADDR ;
: FLUSH-BLK LC 16 = IF BLKNUM 1- BUF? DROP DIRTY FLUSH 0 TO LC
  NEW-BLK THEN ;
: FILE>BLK TO BLKNUM 0 TO LC
  IMP #OPEN CHKIN NEW-BLK
  BEGIN IMP #EOF? 0= WHILE READ-LINE PUT-LINE FLUSH-BLK
  REPEAT IMP #CLOSE ;
.( FILE>BLK Loaded) -->
```

CPYBLK

Description

Facilitates copying of blocks, either within the same blocks same, or to a different blocks file. This is very useful if you find you want to insert a blank block in the middle of an application to allow more code to be added.

Usage

CPYBLK <source-start> <source-end> <source-filename> <dest-start> <dest-filename>

E.g. **CPYBLK 11 31 DSK1.BLOCKS 22 DSK2.BACKUP**

Copies blocks 11 to 31 inclusive from DSK1.BLOCKS to blocks 22 onwards on DSK2.BACKUP

To copy within the same blocks file, simply specify the same file name:

E.g. **CPYBLK 11 31 DSK1.BLOCKS 22 DSK1.BLOCKS**

Caution

Care should be exercised if copying to the same blocks file where the source and destination blocks overlap. E.g.

CPYBLK 11 31 DSK1.BLOCKS 12 DSK1.BLOCKS

One might think this will move all the blocks forward by one block, however, it will simply result in block 11 being copied throughout the range. This is because CPYBLK starts copying at the start block, moving towards the end block, rather than start at the end, moving towards the beginning. Modification of the code is left as an exercise for the reader.

Where a range overlaps, one should copy the blocks elsewhere *first*, then copy from the new (copy) location to the desired location.

```
0 CONSTANT strtblk 0 CONSTANT endblk 0 CONSTANT dstblk
VARIABLE srcfn 2 CELLS ALLOT VARIABLE dstfn 2 CELLS ALLOT
: BLW 32 WORD ;
: CPYBLK BLW NUMBER DROP TO strtblk BLW NUMBER DROP TO endblk
  BLW SWAP srcfn ! srcfn 2+ ! BLW NUMBER DROP TO dstblk
  BLW SWAP dstfn ! dstfn 2+ ! CR endblk 1+ strtblk DO
  srcfn @ srcfn 2+ @ USE I BLOCK DROP I .
  dstfn @ dstfn 2+ @ USE 0 dstblk SETBLK 0 DIRTY FLUSH
  dstblk 1+ TO dstblk LOOP ." Copied. " CR
  srcfn @ srcfn 2+ @ USE ; .( CPYBLK loaded)
: HIDE 32 WORD FIND DROP >LINK HIDDEN ; IMMEDIATE
HIDE CHKIN HIDE CHKOUT HIDE CHKRD HIDE CHKWRT HIDE BUF80
HIDE CFI HIDE CFO HIDE EXP HIDE SZ HIDE BAR HIDE LC
HIDE BLKNUM HIDE BLKADDR HIDE IMP HIDE CLRBUF HIDE NEW-BLK
HIDE READ-LINE HIDE PUT-LINE HIDE FLUSH-BLK HIDE strtblk
HIDE endblk HIDE dstblk HIDE srcfn HIDE dstfn HIDE BLW HIDE HIDE
```

Disclaimer

This code is presented as is. It is possible to destroy your files if you are not careful! The code presented above performs no error checking, it is very basic. Therefore you are advised to exercise caution when using these words.