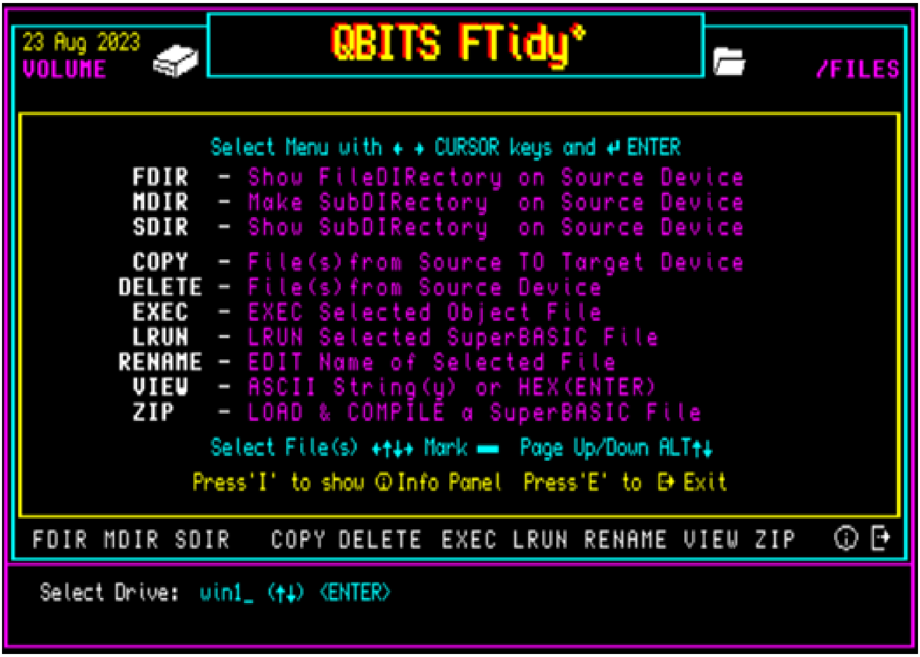
Sinclair QL Retro-Computing



23 Aug 2023
VOLUME

QBITS FTidy*

/FILES

Select Menu with ↑ ↓ CURSOR keys and ↵ ENTER

```
  FDIR  - Show FileDIRectory on Source Device
  MDIR  - Make SubDIRectory  on Source Device
  SDIR  - Show SubDIRectory  on Source Device

  COPY  - File(s)from Source TO Target Device
DELETE  - File(s)from Source Device
  EXEC  - EXEC Selected Object File
  LRUN  - LRUN Selected SuperBASIC File
RENAME  - EDIT Name of Selected File
  VIEW  - ASCII String(y) or HEX(ENTER)
  ZIP   - LOAD & COMPILE a SuperBASIC File
```

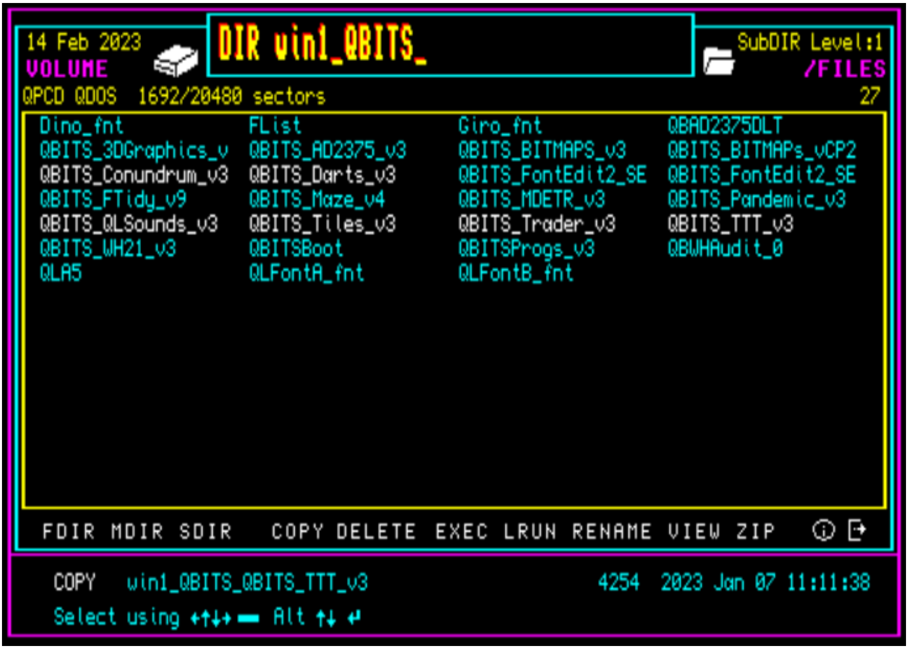
Select File(s) ←↑↓→ Mark ▬ Page Up/Down ALT↑↓

Press'I' to show ⓘ Info Panel Press'E' to ⏏ Exit

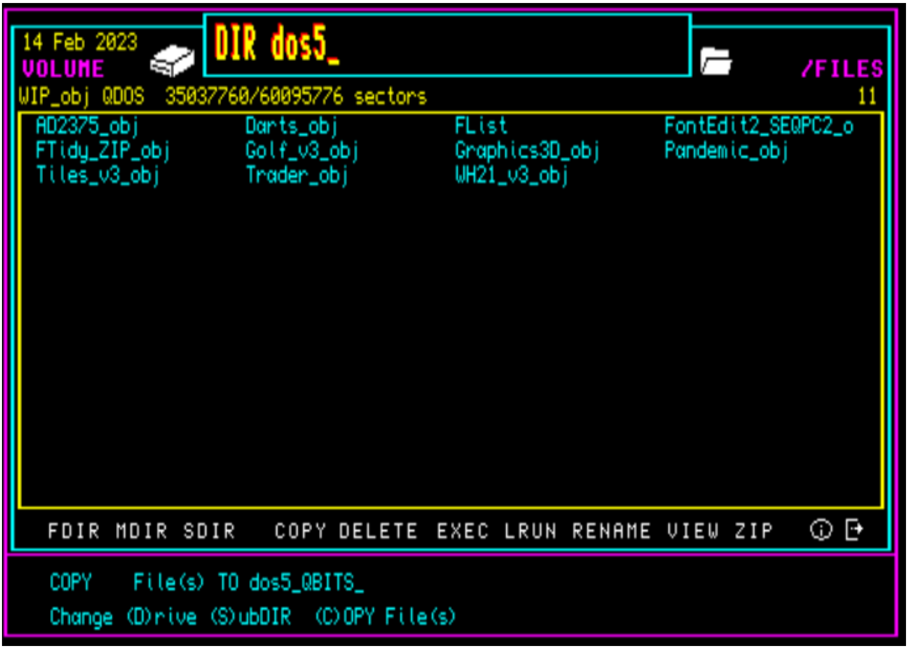
FDIR MDIR SDIR COPY DELETE EXEC LRUN RENAME VIEW ZIP ⓘ ⏏

Select Drive: win1_ (↑↓) <ENTER>

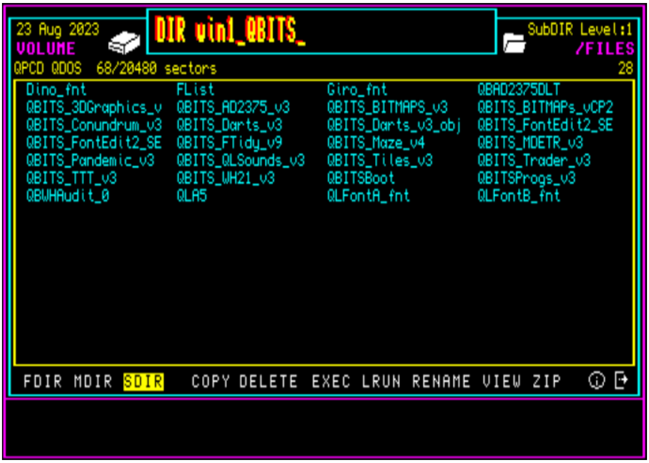
Sinclair QL Retro-Computing



```
14 Feb 2023              DIR win1_QBITS_                                SubDIR Level:1
VOLUME                                                                          /FILES
QPCD QDOS  1692/20480 sectors                                                       27
  Dino_fnt             FList               Giro_fnt             QBAD2375DLT
  QBITS_3DGraphics_v   QBITS_AD2375_v3     QBITS_BITMAPS_v3     QBITS_BITMAPs_vCP2
  QBITS_Conundrum_v3   QBITS_Darts_v3      QBITS_FontEdit2_SE   QBITS_FontEdit2_SE
  QBITS_FTidy_v9       QBITS_Maze_v4       QBITS_MDETR_v3       QBITS_Pandemic_v3
  QBITS_QLSounds_v3    QBITS_Tiles_v3      QBITS_Trader_v3      QBITS_TTT_v3
  QBITS_WH21_v3        QBITSBoot           QBITSProgs_v3        QBWHAudit_0
  QLA5                 QLFontA_fnt         QLFontB_fnt

  FDIR MDIR SDIR   COPY DELETE EXEC LRUN RENAME VIEW ZIP   ⓘ ⇨

   COPY   win1_QBITS_QBITS_TTT_v3                   4254  2023 Jan 07 11:11:38
   Select using ←↑↓→ ▬  Alt ↑↓ ↵
```



```
14 Feb 2023              DIR dos5_
VOLUME                                                                          /FILES
WIP_obj QDOS  35037760/60095776 sectors                                             11
  AD2375_obj           Darts_obj           FList                FontEdit2_SEQPC2_o
  FTidy_ZIP_obj        Golf_v3_obj         Graphics3D_obj       Pandemic_obj
  Tiles_v3_obj         Trader_obj          WH21_v3_obj

  FDIR MDIR SDIR   COPY DELETE EXEC LRUN RENAME VIEW ZIP   ⓘ ⇨

   COPY    File(s) TO dos5_QBITS_
   Change (D)rive (S)ubDIR  (C)OPY File(s)
```



```
23 Aug 2023        DIR win1_QBITS_                              SubDIR Level:1
VOLUME                                                                 /FILES
QPCD QDOS  68/20480 sectors                                                28
 Dino_fnt            FList              Giro_fnt           QBAD2375DLT
 QBITS_3DGraphics_v  QBITS_AD2375_v3    QBITS_BITMAPS_v3   QBITS_BITMAPs_vCP2
 QBITS_Conundrum_v3  QBITS_Darts_v3     QBITS_Darts_v3_obj QBITS_FontEdit2_SE
 QBITS_FontEdit2_SE  QBITS_FTidy_v9     QBITS_Maze_v4      QBITS_MDETR_v3
 QBITS_Pandemic_v3   QBITS_QLSounds_v3  QBITS_Tiles_v3     QBITS_Trader_v3
 QBITS_TTT_v3        QBITS_WH21_v3      QBITSBoot          QBITSProgs_v3
 QBWHAudit_0         QLA5               QLFontA_fnt        QLFontB_fnt

 FDIR MDIR SDIR   COPY DELETE EXEC LRUN RENAME VIEW ZIP
```

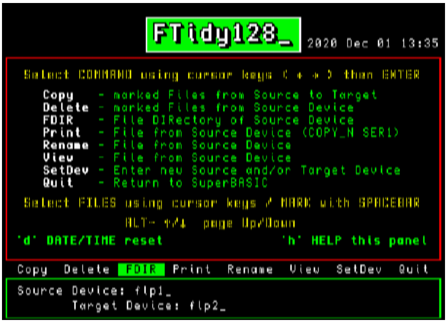
Introduction

Mid 1980's the Sinclair QL arrived with new storage devices **microdrive 'mdv1_ & mdv2_'**. It wasn't long before external **floppy drives** '**flp1_**' & '**flip2_**'were available and **Hard Drives** '**win1_**' etc, followed soon after to open up the possibility of even larger storage capacities

The **QL Technical Guide** identified the **QL Filename** as being up to **36 Bytes** in length or the equivalent number in **ASCII Characters.** Viewing files using the original SuperBASIC **DIR** command displays a single vertical list. This soon spread over several pages and mistyping a file name became a frustrating exercise in using the **DIR** command to review the misspelling of filenames. The **QBITS** approach was to develop a more friendly **File Directory Handler**.

QBITS File Management Concepts

By 1987 a collection of SuperBASIC routines to keep track and review Filenames evolved into an early File Management Tool called File Tidy later shortened to **FTidy**. It accessed a Source Device and viewed up to 160 Filenames. The screen displayed four columns of file names of up to 18 characters. However, the full Filename of up to 36 characters when selected was shown in the window below the Menu bar.



```
                FTidy128_  2020 Dec 01 13:35
 Select COMMAND using cursor keys ( ← → ) then ENTER
    Copy    - marked Files from Source to Target
    Delete  - marked Files from Source Device
    FDIR    - File DIRectory of Source Device
    Print   - File from Source Device (COPY_N SER1)
    Rename  - File from Source Device
    View    - File from Source Device
    SetDev  - Enter new Source and/or Target Device
    Quit    - Return to SuperBASIC
 Select FILES using cursor keys / MARK with SPACEBAR
            ALT- ↑/↓  page Up/Down
'd' DATE/TIME reset                'h' HELP this panel
 Copy  Delete  FDIR  Print  Rename  View  SetDev  Quit
 Source Device: flp1_
        Target Device: flp2_
```

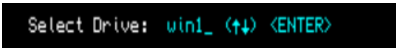
QBITS FTidy128

COPY and **DELETE** commands allowed for single File or batch processing of multiple files. This early version had **SelDev** for **Source** and **Target** selection and a **Print** command to export File lists. These were later dropped for **LOAD/LRUN** options.

A **DIR**ectory section for **SubDIR**ectories was added with the New Millennium updates. This latest version includes **EXEC** and **VIEW** with **ASCII** or **HEX** Readout plus **ZIP** to Load and Compile a SuperBASIC File.

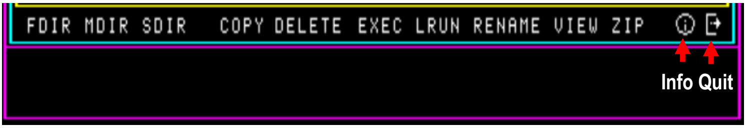
QBITS FTidySE

Screen layout is divided into four areas. Top centre the Title Box displays **QBITS FTidy$^+$** at start up and prompts for Selection of a Source device shown in the bottom action window.



Select Drive: win1_ (↑↓) <ENTER>

The main display area shows a **Help** screen of Commands with a brief description of their functions. Navigation of Menu Commands and Displayed Filenames is by ← ↑ ↓ → Cursor Keys with actions taken with ▬ Spacebar ↵ Enter Keys. To show help screen Press 'i' for ⓘ Info Press 'E' Exit to leave ⇨ the program. The **DIR**ectory and **FILE** Menu can also be Selected by Keys **F M S C D E L R V Z.**

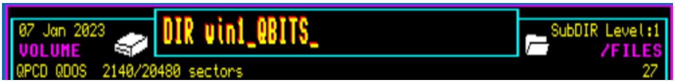


QBITS FTidySE File Directory

FDIR displays the **DIRectory** of the Default or last **Device** chosen. Full Filenames and Stats, Byte Size & Date/Time stamp entries of the Selected Storage Device are copied to **FList**. The list is then Read and Sorted Alphanumerically. A Sort program is given in the QL Users Guide Chapter 16. A few tweaks with the choice of array and variable names and Voilá! The **FList** of Filenames are selected sequentially by a **FOR loop** and compared within a **REPeat Loop** leaving the output **DFile$(n)** array sorted in Alphanumeric order. A second **FOR loop** selects any **SubDIR**ectory names and lists them in front of the **Filenames**.

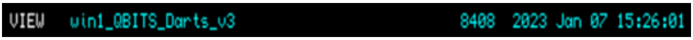
QBITS FTidySE SubDIRectories

The full identity of a file location can be 41 Characters. This begins with the Drive Device a five-character identifier, ie. mdv1_ flp1_ win1_ the fifth character being '**_**' an underscore. The next 36 Characters make up the Filename, with the first twenty-four characters considered for use as **SubDIR**ectory's. For example, 'SubDIR1_' which as with Drive names end with an underscore. If they were named with letters of the alphabet, 'A_' to 'L_' we could potentially create twelve SubDIR's. For QBITS FTidy the limit is set at six SubDIR levels.



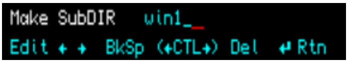
```
07 Jan 2023          DIR win1_QBITS_           SubDIR Level:1
VOLUME                                                 /FILES
QPCD QDOS  2140/20480 sectors                              27
```

The File DIRectory [**FDIR**] or SubDIRirectories [**SDIR**] if present, display the **Device name, Volume/Sectors** and **File Number** of Directory selected in the Left and Right of the upper window. The Filenames of the selected **Device DIR**ectory or **SubDIR**ectory are Read and Sorted to generate the new display of Filenames . Any selected **Filename** is shown in full with Stats **Bytes** size and **Date/Time** stamp in the lower action window.



```
VIEW   win1_QBITS_Darts_v3                 8408  2023 Jan 07 15:26:01
```

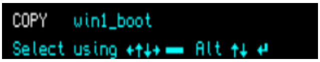
QBITS FTidySE MDIR



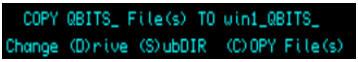
```
Make SubDIR   win1_
Edit ← →  BkSp (←CTL→) Del  ↵Rtn
```

Select **MDIR** and use the Line Editor to create a new **SubD**irectory. QBITS allocation is exampled below…

DIM SubDIR$(6,24) where SubDIR$(1) = "**SD1_**" and SubDIR$(6) access = "**SD1_SD2_SD3_SD4_SD5_SD6_**"



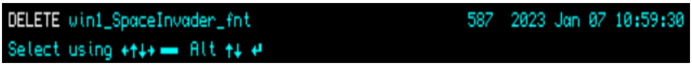
```
COPY   win1_boot
Select using ←↑↓→ ▬ Alt ↑↓ ↵
```



```
 COPY QBITS_ File(s) TO win1_QBITS_
Change (D)rive (S)ubDIR  (C)OPY File(s)
```

QBITS FTidySE COPY / DELETE

Select single or multiple files. The Filename(s) are identified by moving through the files listed and highlighting with the Spacebar. For **COPY** select a destination **Target** device with (**D**)rive and (**S**)ubDIR. Once the **Target DIR**ectory of Filenames are shown Press (**C**)opy and the selected Filenames from **Source DIR**ectory are then copied from **Source** to **Target** device. For any files of same name in destination device you are prompted with **Overwrite 'y/n'**.



```
DELETE win1_SpaceInvader_fnt                           587  2023 Jan 07 10:59:30
Select using ←↑↓→ ▬ Alt ↑↓ ↵
```

Similarly, for **DELETE** the highlighted Filename(s) are confirmed with '**y/n**' before deleting.

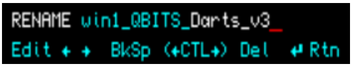
```
EXEC   win1_progs_darts_v3_obj
```

```
LRUN   win1_QBITS_Darts_v3
```

QBITS FTidySE EXEC / LRUN

Select a Filename, the full Filename up to 36 characters is displayed together with its Byte length and Time/Date stamp. EXEC is for Object files, machine code files that can be loaded and run as an independent JOB. LRUN is for a S/SuperBASIC program that loads and runs under the QL Interpreter. For both you are prompted with '**y/n**' to action.

```
RENAME win1_QBITS_Darts_v3
```



```
RENAME win1_QBITS_Darts_v3_
Edit ← →  BkSp (←CTL→) Del  ↵Rtn
```

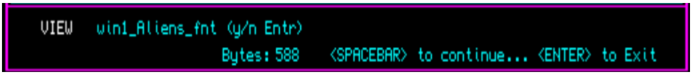
QBITS FTidySE RENAME

Select an existing Filename (**file$**) and **Edit** the string (**str$**) with the simple **QBITS Line Editor**. Actioning with ↵ Enter and the **FTidy** checks that the Filename doesn't already exist. If not a COPY with new filename is made to source and old file Deleted.

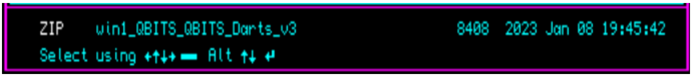
```
VIEW   win1_QBITS_Darts_v3
```

QBITS FTidySE VIEW

Reading the opening lines of a File was seen as a necessary addition to help in recognising a file content for what it was. **S/SuperBASIC** Files are **Plain Text**, others are **Data** lists or **Executable** computer readable machine code. **FTidy View** gives the option to display as **ASCII characters (y)** or **Hex Bytes (Enter).**



```
   VIEW  win1_Aliens_fnt (y/n Entr)
                        Bytes: 588   <SPACEBAR> to continue... <ENTER> to Exit
```



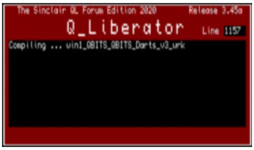
```
   ZIP   win1_QBITS_QBITS_Darts_v3                    8408  2023 Jan 08 19:45:42
   Select using ←↑↓→ ▬ Alt ↑↓ ↵
```

QBITS FTidySE ZIP

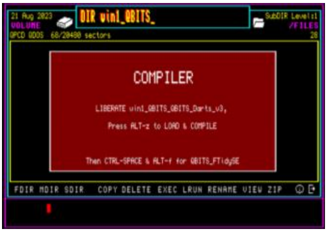
Select an existing S/SupeBASIC Filename ie. with a '_bas' suffix, confirm (y/n Entr) 'n' returns to **FTidy Menu**. The compiler **Q_LIBERATOR** is installed if absent. Several programs can share **QLib_RUN** if installed or it can be linked as part of the Compile process. Press '**y**' if Runtime is resident or '**Enter**' if to be linked.

```
COMPILE QBITS_QBITS_Darts_v3 (y/n Enter)
```

When completed return to
QBITS FTidySE with **ALT-f** and
EXEC Filename_obj
the Compiled Program.



```
The Sinclair QL Forum Edition 2020      Release 3.45a
              Q_Liberator        Line 1157
Compiling ... win1_QBITS_QBITS_Darts_v3_wrk
```



```
21 Aug 2023    DIR win1_QBITS_                SubDIR Level:1
VOLUME                                                /FILES
QPCD QDOS  68/20480 sectors                               28

                       COMPILER

           LIBERATE win1_QBITS_QBITS_Darts_v3,
             Press ALT-z to LOAD & COMPILE

          Then CTRL-SPACE & ALT-f for QBITS_FTidySE
```

QBITS FTidySE Background Notes:

The QL User's Manual introduces the QL operating system and the SuperBASIC Interpreter and Line Editor for entering program code. This was the start to my own venture into writing a complete program. Programming revolves around presentation and problem solving and can be broken down into a number of steps or tasks. This separation of the common and more complicated features that interplay between similar and differing aspects of a program offers the opportunity for coding in manageable units. Early BASIC's used GOTO and GOSUB, to access by line numbers these groups of code, but the control structures in S/SuperBASIC make these statements virtually redundant. Code blocks are written and handled as named **PROCedures** and **FuNctions.**

The processing of program is dependent and triggered by the interaction between a user and the information presented to screen. On opening a new program, the first action is to initialise the screen environment, set and load any arrays and variables, then follow with a Menu that directs keyboard input to select the next action.

Therefore, screen layout becomes fundamental to the graphics displayed. This begins with separating areas of the screen, SuperBASIC uses **WINDOW**'s which can be highlighted with **BORDER**'s and given different **PAPER** colour backgrounds. Presenting information to screen the graphics are quite often just the grouping of character strings.

Character Graphics

SuperBASIC provides the opportunity to deploy character strings in many ways with the use of **INK**, **STRIP**, **CSIZE** and **CURSOR**. The options with **CLS** helps when Printing strings to the same CURSOR location. **CLS** 3 clears the whole of the cursor line while CLS 4 clears leftover characters from the previous string to the end of line. Then when encountering strings of varying length to a specific screen area a more succinct approach can be achieved. The command **LEN**(String$) returns a string length in the number of characters and with this knowledge for a set maximum *Number* of characters you can use the **FILL$** command to infill with added spaces or with set characters such as zero's

PRINT String$(1 to **LEN**(String$)& **FILL$**(' ',*Numbe*r-**LEN** (String$))

Another useful option is when printing a set of numbers where the number of characters increase as it builds from units, ten, hundreds, thousands etc. SuperBASIC delivers the number in the form of a string 1,10,100,1000 which can be manipulated to grow right to left.

CURSOR 100,20:**PRINT FILL$**(' ',4-**LEN**(number%)&number%

The CURSOR also has a further option when used with the Graphic coordination system. This positions the printed string in relation to a drawn Graphic coordinate.

CURSOR x,y,ox,oy:**PRINT** String$ (ox,oy being the offset to x,y coordinates.)

The use of **OVER** with **FOR** loops can produce **Bold** or even a **3D** effect on screen Printouts.

.

Handling Screen Positioning

QBITS FTidy maps a list of Filenames across the screen in rows and columns, each column having a defined number of characters or string length. If a name exceeds this length, then the name is truncated. If the name string is less, the missing characters are filled with spaces to overwrite the possibility of characters left over from a previously displayed Filename. If the list of Filenames exceeds the available screen space it poses additional problems. Reaching the bottom of the window the bottom row can be scrolled up and a new group of filenames added. To keep track a file pointer identifies the position in the Filename list and by adding a line pointer a screen row can now be identified by subtracting the Line pointer from the File List pointer. A similar arrangement is utilised for scrolling the screen down so as to add rows at the top of the page.

Note: A program having run its course needs to close down, release RAM etc and restore the system back to previous settings

Note: File & DIRectory Info

QL Files appeared as an array of bytes on a physical storage device such as microdrives mdv1_, floppy disks flp1_ and Hard drives win1_ etc.. The file storage system is composed of 512-byte blocks, so addressing is via an associated File Pointer by Block number (sector) and Byte number within that block. The QL Tech Guide describes a 64 Bytes header as follows:

$00	long	file length
$04	byte	file access key (not yet implemented - set at 0)
$05	byte	file type
$06	8bytes	file type-dependant information
$0E	2+36 Bytes	filename
$34	long	reserved for update Date (not yet implemented)
$38	long	reserved for reference Date (not yet implemented)
$3C	long	reserved for backup Date (not yet implemented)

The file types allowed at the time of original QL:

2	a relocatable object file
1	an executable program where the first longword of the type-dependant information holds the default size of data space required for the program.
0	for anything else

QBITS FTidySE use of **LIBERATOR**

An early aspiration was to include a link to Compile SuperBASIC Programs hence the Menu Command ZIP. In computing it is a term used for File(s) or Routines compressed into a package. By definition it can imply to go faster, add to this there was early Spectrum Compiler called ZIP and written by N Goodwin. I hope he won't mind!

The available Compliers being SUPERCHARGE TURBO or LIBERATOR. To achieve compatibility across all QL Platforms is not an easy Task. For the present QBITS_FTidy relies on the SMSQ/E O/S being in place and uses the QLIBERATOR Sinclair QL Forum Edition 2020 for QPC. Users should refer to the User Manuals for a better understanding.

QBITS FTidySE Procedures

Init_win	Initialises Program setting the screen display
F_Title	Displays QBITS Title / DIR headings
F_Info	Displays Info/Help screen
Cmd_Menu	Main Program Loop
SelDrv	Selects Source or Target Devices
FileDIR	Generates File List of Device DIRectory or a Sub DIRectory
F_Sort	Arranges Filenames AlphaNumerically with SubDIR(s) first
MakeDIR	Uses Line Editor to create a New SubDIRectory
SubDIR	Selects and displays the Filenames of a SubDIRectory
Sub_up	Move to Drive DIR or Higher-Level SubDIR
Sub_dn	Move to a Lower-Level SubDIR
F_Select	Use Cursor keys to Select a SubDIR or Filename
Fscr_posn	Calculate screen position of SubDIR or Filename
Fscr_up	Scroll up one row
Fscr_dn	Scroll down one row
F_write	Print SubDIR or Filename and Stats o screen position
F_clear	Prints updated File List to screen
F_Chk	Returns y/n Enter answer [no=0 yes=1 Enter=2
F_Copy	Selects and Confirms Filename(s) y/n? from Source Device
F_Target	Selects destination - Targeted Device DIR/SubDIR
F_Copy2	Copies File(s) to Target Device DIR SubDIR with overwrite y/n?
F_Delete	Deletes Selected Files(s) from Source Device y/n?
F_Lrun(act)	**EXEC** or **L**oad/**RUN** selected File from displayed list
F_Rename	Uses Line editor to Rename a selected Filename
F_View	Prints ASCII or Hex Code of selected Filename to screen
F_ZIP	Selected SuperBASIC File linked into **LIBERATOR** Compiler
Ln_Ed	Line Editor action menu
Str_chk	Checks if last Character of string is '_' and deletes
Ln_Prn	Prints Filename to Screen
Ln_Cur	Prints Current Cursor Position to Screen
Add_chr	Adds a character anywhere within String or at end.
Del_chr	Deletes a character anywhere in string
KQuit	Key Graphic Image created for Quit ⇥
KInfo	Key Graphic Image created for Info/Help ⓘ
GDrive	Hard Drive symbol
GFolder	File Folder Symbol

QBITS FTidySE Code

```
1000 REMark QBITS_FTidySE_bas (QBITS File Tidy SE v2 QPC2 2023)

1002 WMON::MODE 4:gx=0:gy=0     :REMark Basic Screen Settings

1004 REMark DIM Arrays : Set max% number of Files
1005 max%=300
1006 DIM DFile$(max%,2,36),fink%(max%),CFile$(max%,2,36),cink%(max%)
1007 DIM Comd$(58),key$(3,52),help$(9,48),Time$(20)
1008 DIM DD$(5),DDIR$(24),SD$(5),SDIR$(24),TD$(5),TDIR$(24),str$(36)
```

Note: DFile$(max%,2,36)
Filename=1 ↑
Stats =2
Max ASCII Characters

```
1010 REMark Local Device Settings
1011 DIM Drv$(15,5):RESTORE 1012:FOR d=0 TO 15:READ Drv$(d)
1012 DATA 'mdv1_','mdv2_','flp1_','flp2_','win1_','win2_','win3_','win4_'
1013 DATA 'drv1_','drv2_','drv3_','drv4_','drv5_','drv6_','drv7_','drv8_'
1014 dev$='flp1_':dn%=2:DD$=dev$:DDIR$=''     :REMark Default Drive
1015 dn$='flp1_QBITSProgs'                    :REMark Exit Route

1017 WHEN ERRor :CONTINUE:END WHEN
```

Note: WHEN ERRor will CONTINUE instead of the Interpreter halting the program. Device Error is printed if FDIR fails to find a Device or any Files.

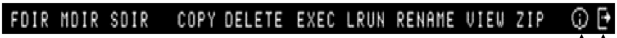
```
1019 REMark Import QBITSConfig Settings - QPC2
1020 OPEN_IN#9,'ram2_QBITSConfig':INPUT#9,gx\gy\dn$\dev$\dn%\dm%
1021 FOR d=0 TO 15:INPUT#9,Drv$(d):END FOR d:CLOSE#9

1023 REMark FTidy Setup & Comand Menu

1025 Init_win:Cmd_Menu

1027 DEFine PROCedure Init_win
1028 OPEN#5,scr_:WINDOW#5,512,256,gx,gy          :PAPER#5,0:BORDER#5,1,3:CLS#5
1029 OPEN#4,scr_:WINDOW#4,504,214,gx+4,gy+7      :PAPER#4,0:BORDER#4,1,5:CSIZE#4,1,0
1030 OPEN#3,scr_:WINDOW#3,280,26,gx+114,gy+2    :PAPER#3,0:BORDER#3,1,5:CLS#3
1031 WINDOW#2,504,220,gx+4,gy+2 :PAPER#2,0                          :CSIZE#2,0,0:INK#2,7
1032 WINDOW#1,496,162,gx+8,gy+42:PAPER#1,0       :BORDER#1,1,6:CSIZE#1,0,0:INK#1,7
1033 WINDOW#0,512, 34,gx,gy+222 :PAPER#0,0       :BORDER#0,1,3:CSIZE#0,0,0:INK#0,7
1034 OVER#4,1:INK#4,3
1035 FOR i=0 TO 1:CURSOR#4,  4+i,12:PRINT#4,'VOLUME'
1036 FOR i=0 TO 1:CURSOR#4,450+i,12:PRINT#4,'/FILES'
1037 OVER#4,0:INK#2,6:Time$=DATE$
1038 CURSOR#2,6,7:PRINT#2,Time$(10 TO 11)&Time$(5 TO 9)&Time$(1 TO 4)
1039 SCALE#2,100,0,0:SCALE#1,100,0,0:F_Title 2,66,'QBITS FTidyº':F_Info
1040 END DEFine
```

07 Jan 2023

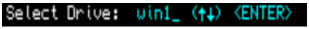
```
1042 DEFine PROCedure F_Title (cs%,x%,Title$)
1043 CLS#3:CSIZE#3,cs%,1:OVER#3,1
1044 INK#3,2 :FOR i=0 TO 1:CURSOR#3,x%+i,1:PRINT#3,Title$
1045 INK#3,6 :FOR i=1 TO 2:CURSOR#3,x%+i,2:PRINT#3,Title$:OVER#3,0
1046 END DEFine
```

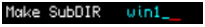
Note: cs% - Character SIZE: x% - horizontal offset: Title$ - Character string.

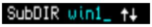
```
1048 DEFine PROCedure F_Info
1049 help$(0)=' FDIR     - Show FileDIRectory on Source Device'
1050 help$(1)=' MDIR     - Make SubDIRectory  on Source Device'
1051 help$(2)=' SDIR     - Show SubDIRectory  on Source Device'
1052 help$(3)=' COPY     - File(s)from Source TO Target Device'
1053 help$(4)='DELETE    - File(s)from Source Device'
1054 help$(5)=' EXEC     - EXEC Selected Object File'
1055 help$(6)=' LRUN     - LRUN Selected SuperBASIC Prog'
1056 help$(7)='RENAME    - Edit Name of Selected File'
1057 help$(8)=' VIEW     - ASCII String(y) or HEX(ENTER)'
1058 help$(9)=’ ZIP      - QLBASIC COMPILER [Not Implemented]’
1059 OVER#1,1:CSIZE#1,1,0:CLS#1:sp=20
1060 FOR hp=0 TO 8
1061   INK#1,7:FOR i=0 TO 1:CURSOR#1,54+i,sp+hp*11:PRINT#1,help$(hp,1 TO 9);
1062   INK#1,3:PRINT#1,help$(hp,10 TO):IF hp=2:sp=26
1063 END FOR hp
1064 OVER#1,0:CSIZE#1,0,0:INK#1,5:GDrive 7,31,92:GFolder 7,136,91
1065 CURSOR#1,106,  8:PRINT#1,'Select Menu with ←→ CURSOR keys and ← ENTER'
1066 CURSOR#1,106,128:PRINT#1,'Select File(s)  ←↑ ↓→  Mark   Page Up/Down ALT ↑ ↓'
1067 BLOCK#1,2,4,334,10,5:BLOCK#1,12,3,256,132,5:INK#1,6
1068 CURSOR#1, 96,142:PRINT#1,"Press'I' to show  Info Panel  Press 'E' to   Exit"
1069 KInfo 1,6,94,8:KExit 1,6,167,8
1070 cmd=1:mm%=1:mx%=9:lptr%=0:INK#0,7:px%=108:DD$="
1071 END DEFine
```

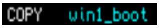


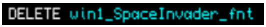
```
1073 DEFine PROCedure Cmd_Menu
1074 DIM cx%(9),Cmd$(9,6):RESTORE 1075:FOR i=1 TO 9:READ cx%(i),Cmd$(i)
1075 DATA 16,'FDIR',56,'MDIR',96,'SDIR',150,'COPY',190,'DELETE'
1076 DATA 246,'EXE',286,'LRUN',326,'RENAME',382,'VIEW'
1077 STRIP#4,0:INK#4,7:FOR i=1 TO 9:CURSOR#4,cx%(i),200:PRINT#4,Cmd$(i)
1078 KInfo 2,7,156,4.6:KExit 2,7,162,4.4:cmd=1
1079 REPeat Cmd_lp
1080   STRIP#4,6:INK#4,0:CURSOR#4,cx%(cmd),200:PRINT#4,Cmd$(cmd)
1081   k$=(INKEY$(#0,-1)):k=CODE(k$)
1082   STRIP#4,0:INK#4,7:CURSOR#4,cx%(cmd),200:PRINT#4,Cmd$(cmd)
1083   SELect ON k
1084     =192:cmd=cmd-1:IF cmd<1:cmd=9          :REMark← Left   Cursor min-max
1085     =200:cmd=cmd+1:IF cmd>9:cmd=1          :REMark →Right Cursor max=min
1086     =73,105   :h%=1:F_Info:h%=0            :             :REMark (I)nfo
1087     =69,101   :IF dn$=’’:STOP :ELSE LRUN dn$ :STOP   :REMark(E)xit
1088     =10,65 TO 122:IF k<>10:cmd=1+(k$ INSTR 'FfMmSsCcDdEeLlRrVv')DIV 2
1089         SELect ON cmd
1090           =  1:SelDrv                      :CLS#0
1091           =  2::MakeDIR                    :CLS#0
1092           =  3::SubDIR                     :CLS#0
1093           =  4:IF ftot%>0:F_Copy           :CLS#0
1094           =  5:IF ftot%>0:F_Delete         :CLS#0
1095           =  6:IF ftot%>0:F_Run 2          :CLS#0   :REMark EXEC
1096           =  7:IF ftot%>0:F_Run 1          :CLS#0   :REMark # LOAD y LRUN
1097           =  8:IF ftot%>0:F_Rename         :CLS#0
1098           =  9:IF ftot%>0:F_View           :CLS#0   :REMark # HEX  y Char String
1099           =10:IF ftot%>0:F_ZIP             :CLS#0   :REMark Compiler
1100         END SELect
1101   END SELect
1102 END REPeat Cmd_lp
1103 END DEFine
```

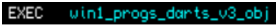

Select Drive: win1_ (↑↓) <ENTER>

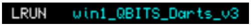

Make SubDIR win1_

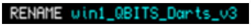

SubDIR win1_ ↑↓

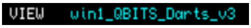

COPY win1_boot

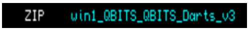

DELETE win1_SpaceInvader_fnt


EXEC win1_progs_darts_v3_obj

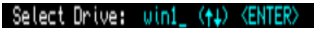

LRUN win1_QBITS_Darts_v3


RENAME win1_QBITS_Darts_v3


VIEW win1_QBITS_Darts_v3

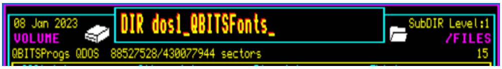

ZIP win1_QBITS_QBITS_Darts_v3

```
1105 REMark QBITS FTidy DIRectory Management

1107 DEFine PROCedure SelDrv
1108 DIM SubDIR$(6,24):dl%=0:OD$=DD$:dch=0
1109 IF cmd=1:INK#0,7:CURSOR#0,18,6:PRINT#0,'Select Drive: ':px%=108
```



Select Drive: win1_ (↑↓) <ENTER>

```
1110 REPeat Dr_lp
1111   INK#0,5:CURSOR#0,px%,6:PRINT#0,Drv$(dn%)&' (⬆⬇) <ENTER>':CLS#0,4
1112   k=CODE(INKEY$(#0,-1))
1113   SELect ON k
1114     =10:DD$=Drv$(dn%):EXIT Dr_lp          :REMark Enter Select Drive
1115     =208:dn%=dn%-1:IF dn%< 0:dn%=15       :REMark Up
1116     =216:dn%=dn%+1:IF dn%>15:dn%= 0       :REMark Down
1117   END SELect
1118 END REPeat Dr_lp
1119 IF OD$=DD$:RETurn :ELSE DDIR$='':FileDIR
1120 END DEFine
```

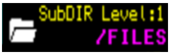


08 Jan 2023
VOLUME
QBITSProgs QDOS 88527528/430077944 sectors
DIR dos1_QBITSFonts_
SubDIR Level:1
/FILES
15

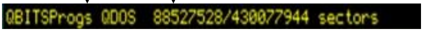
```
1122 DEFine PROCedure FileDIR
1123 CLS#0:DELETE DD$&DDIR$&'FList':F_Title 1,4,'DIR '&DD$&DDIR$
1124 IF cmd<2:INK#0,5:CURSOR#0,24,6:PRINT#0,'Files being Selected...'
```

Files being Selected...

```
1125 OPEN_NEW#9,DD$&DDIR$&'FList':STAT#9,DD$&DDIR$:WSTAT#9,DD$&DDIR$:CLOSE#9
1126 OPEN_IN #9,DD$&DDIR$&'FList':INPUT#9,DName$\DSec$:n=1:ftot%=0
1127 REPeat DIR_lp
1128   IF EOF(#9) OR n>max%:ftot%=n-1:CLOSE#9:EXIT DIR_lp
1129   INPUT#9,DFile$(n,1):fink%(n)=5
1130   IF '>' INSTR DFile$(n,1)=0:INPUT#9,DFile$(n,2):n=n+1:ELSE n=n+1
1131 END REPeat DIR_lp
1132 BLOCK#2,88,10,414,7,0:IF dl%>0:CURSOR#2,414,7:PRINT#2,'SubDIR Level:';dl%
1133 BLOCK#2,480,10,4,29,0:INK#2,6 :CURSOR#2,4,29:PRINT#2,DName$;'  ';DSec$
1134 CURSOR#2,466,29:PRINT#2,FILL$(' ',5-LEN(ftot%))&ftot%
1135 IF ftot%=0:stot%=0:CLS:F_Title 1,4,'DEVICE ERROR':RETurn
1136 F_Sort:nm%=1:nx%=ftot%:lptr%=0:CLS#1:F_clear:n=1
1137 END DEFine
```

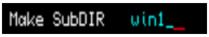


SubDIR Level:1
/FILES

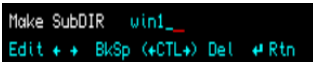


QBITSProgs QDOS 88527528/430077944 sectors

```
1139 DEFine PROCedure F_Sort
1140 FOR sn=1 TO ftot%
1141   p=sn:comp$=DFile$(p,1):info$=DFile$(p,2)
1142   REPeat Sort_lp
1143     IF comp$>=DFile$(p-1,1):EXIT Sort_lp
1144     DFile$(p,1)=DFile$(p-1,1):DFile$(p,2)=DFile$(p-1,2):p=p-1
1145   END REPeat Sort_lp
1146   DFile$(p,1)=comp$:DFile$(p,2)=info$
1147 END FOR sn
1148 ntop=1:nsel=1:stot%=0
1149 FOR sn=1 TO ftot%
1150   IF '>' INSTR DFile$(sn,1)
1151     comp$=DFile$(sn,1):info$=DFile$(sn,2):nsel=sn-1
1152     FOR p=nsel TO ntop STEP -1
1153       DFile$(p+1,1)=DFile$(p,1):DFile$(p+1,2)=DFile$(p,2)
1154     END FOR p
1155     DFile$(ntop,1)=comp$:DFile$(ntop,2)=info$:ntop=ntop+1:stot%=stot%+1
1156   END IF
1157 END FOR sn
1158 END DEFine
```

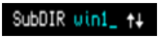


Make SubDIR   win1_

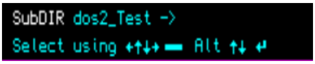


Make SubDIR   win1_
Edit ← → BkSp (←CTL→) Del ↵ Rtn

```
1160 DEFine PROCedure MakeDIR
1161 md%=24-LEN(DDIR$):px%=138+LEN(DDIR$)*6
1162 IF md%<3:CURSOR#0,24,6:PRINT#0,'Lowest Level Reached':PAUSE 50:RETurn
1163 INK#0,7 :CURSOR#0,24,6:PRINT#0,'Make SubDIR   ';:INK#0,5:PRINT#0,DD$&DDIR$
1164 cp%=1:sl%=0:sm%=md%:str$='':Ln_Ed:IF LEN(str$)=0:RETurn
1165 CURSOR#0,px%+LEN(str$)*6,6:PRINT#0,' (y/n)':K_Chk
1166 IF chk=1
1167   FOR n=1 TO stot%:IF DDIR$&str$ INSTR DFile$(n,1):RETurn
1168   MAKE_DIR DD$&DDIR$&str$:FileDIR
1169 END IF
1170 END DEFine
```

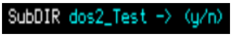


SubDIR win1_ ↑↓



SubDIR dos2_Test ->
Select using ←↑↓→ ▬ Alt ↑↓ ↵

```
1172 DEFine PROCedure SubDIR
1173 INK#0,7:CURSOR#0,24,6:PRINT#0,'SubDIR ';:INK#0,5:PRINT#0,DD$&DDIR$;
1174 INK#0,7:PRINT#0,' ⬆⬇ ':CLS#0,4:k=CODE(INKEY$(#0,-1))
1175 IF k=208 AND dl%>=0   :Sub_up            dl% DIRectory Level
1176 IF k=216 AND stot%>=0:Sub_dn             stot% Sub Total
1177 END DEFine

1179 DEFine PROCedure Sub_up
1180 SubDIR$(dl%)='':dl%=dl%-1:DDIR$=SubDIR$(dl%):FileDIR:RETurn
1181 END DEFine

1183 DEFine PROCedure Sub_dn
1184 IF stot%<1 OR dl%=6:RETurn
1185 px%=96:mark%=5:n=1:nm%=1:nx%=stot%:st%=1:F_select:st%=0
1186 INK#0,5:CURSOR#0,px%,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1187 IF chk=1
1188   DDIR$=DFile$(n,1,1 TO flen%-3)&'_':dl%=dl%+1:nm%=stot%:nx%=ftot%:FileDIR
1189   SubDIR$(dl%)=DDIR$:CURSOR#0,px%+12,6:PRINT#0,DDIR$:CLS#0,4
1190 END IF
1191 nm%=1:nx%=ftot%:F_clear:n=1
1192 END DEFine
```



SubDIR dos2_Test -> (y/n)

QBITS FTidy SubDIRectories

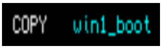
The full identity of a file location can be 41 Characters. This begins with the Drive Device a five-character identifier, ie. mdv1_ flp1_ win1_ dos1_ the fifth character '_' always being an underscore. The next 36 Characters make up the Filename, of which the first twenty-four characters can be considered for **SubDIR**ectory use. For example, 'SubDIR1_' which as with Drive names have to end with an underscore. If they were named with letters of the alphabet, 'A_' to 'L_' we could potentially create twelve SubDIR levels. However, for QBITS File Tidy I limited this to just SIX Sub levels, creating further levels will not be accessed by **SDIR.**

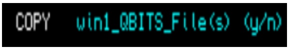
DIM SubDIR$(6,24) where SubDIR$(1) = "**SD1_**" and SubDIR$(6) access = "**SD1_SD2_SD3_SD4_SD5_SD6_**"

```
1194 REMark QBITS FTidy Filename Display

1196 DEFine PROCedure F_select                         filename: select and highlight in screen location
1197 IF cmd>3:INK#0,7:CURSOR#0,24,6:PRINT#0,Cmd$(cmd):CLS#0,4     Note: Check a File Command
1198 INK#0,5:CURSOR#0,66,6:PRINT#0,DD$&DDIR$                           Device & SubDIR
1199 REPeat Sel_lp
1200  Fscr_posn:fink%(n)=7:F_write:fink%(n)=5:k=CODE(INKEY$(#0,-1))
1201  SELect ON k
1202   =192:Fscr_posn:F_write:n=n  -1          :REMark Back       1
1203   =200:Fscr_posn:F_write:n=n +1           :REMark Forward    1
1204   =208:Fscr_posn:F_write:n=n  -4          :REMark Up         1 Row
1205   =216:Fscr_posn:F_write:n=n +4           :REMark Down       1 Row
1206   =209:Fscr_posn:F_write:n=n -60          :REMark Up         1 Page
1207   =217:Fscr_posn:F_write:n=n+60           :REMark Down       1 Page
1208   = 32:fink%(n)=mark%:F_write:n=n+1       :REMark            mark Filename mark% in Highlight Colour
1209   = 10:fink%(n)=7:CURSOR#0,0,20:CLS#0,4:RETurn               fink%(n) File Ink Highlight Colour
1210  END SELect
1211 END REPeat Sel_lp
1212 END DEFine

1214 DEFine PROCedure Fscr_posn                    calculate screen position
1215 IF n<nm%:n=nm%                               n file number nm% min nx% max
1216 IF n>nx%:n=nx%                               Note: n floating point – necessary with QDOS FOR loops
1217 fptr%=n-1:frow%=(fptr% DIV 4)
1218 IF frow%>(15+lptr%):Fscr_up:n=fptr%+1:Fscr_posn    frow% file row fptr% file pointer lptr% line pointer
1219 IF frow%<(  0+lptr%):Fscr_dn:n=fptr%+1:Fscr_posn
1220 srow%=frow%-lptr%:scol%=(fptr% MOD 4)*20         srow% screen row scol% screen column
1221 END DEFine

1223 DEFine PROCedure Fscr_up
1224 lptr%=lptr%+1:SCROLL#1,-10:n=(lptr%+15)*4:srow%=15
1225 FOR i=0 TO 3:scol%=i*20:n=n+1:F_write
1226 END DEFine

1228 DEFine PROCedure Fscr_dn
1229 lptr%=lptr%-1:SCROLL#1,10:n=(lptr%)*4:srow%=0
1230 FOR i=0 TO 3:scol%=i*20:n=n+1:F_write
1231 END DEFine

1233 DEFine PROCedure F_write
1234 IF n>ftot% OR n<1:RETurn
1235 flen%=LEN(DFile$(n,1)):slen%=LEN(DDIR$)          flen% Filename length slen% SubDIRectory length
1236 IF flen%-slen%>18:flen%=18+slen%
1237 INK#1,fink%(n):CURSOR#1,8+scol%*6,srow%*10       fink% file ink print colour
1238 PRINT#1,DFile$(n,1,1+slen% TO flen%)&FILL$(' ',18+slen%-flen%)          Filename - Main Screen
1239 INK#0,5:CURSOR#0,px%,6:PRINT#0,DFile$(n,1):CLS#0,4:IF st%<>1:RETurn       Filename - Action Window
1240 INK#0,5:CURSOR#0,24,20:PRINT#0,'Select using ←↑↓→ ▬ Alt ↑↓ ↵':CLS#0,4
1241 BLOCK#0,12,3,130,24,5:BLOCK#0,2,4,198,22,5:CURSOR#0,300,6:PRINT#0,DFile$(n,2)  File Stats - Action
1242 END DEFine                               Use of BLOCK for Spacebar and Return Tail

1244 DEFine PROCedure F_clear                      Clear marked files
1245 FOR sc=1 TO ftot%:fink%(sc)=5                 ftot% File Total sc set clear (ink colour)
1246 fs%=(lptr%*4)+1:fe%=(lptr%+16)*4:IF fe%>ftot%:fe%=ftot%
1247 FOR n=fs% TO fe%:Fscr_posn:F_write            fs% file start fe% file end
1248 END DEFine
```

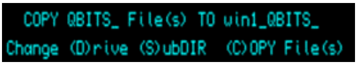
```
1250 REMark QBITS FTidy File Mamgement

1252 DEFine PROCedure K_Chk                                  Note: INKEY$ Check
1253 REPeat chk_lp
1254  k=CODE(INKEY$(#0,-1))
1255  SELect ON k=78,110:chk=0:EXIT chk_lp                  Note: Answer 'n,N' no
1256  SELect ON k=89,121:chk=1:EXIT chk_lp                  Note: Answer 'y,Y' yes
1257  SELect ON k=10:IF cmd=9 OR cmd=10:chk=2:EXIT chk_lp:   Note: VIEW & ZIP
1258 END REPeat chk_lp
1259 END DEFine
```

```
1261 DEFine PROCedure F_Copy                   Select Single/Multiple Files
1262 px%=96:mark%=7:nm%=stot%+1:st%=1:F_select:st%=0
1263 CURSOR#0,px%,6:PRINT#0,DDIR$;'File(s) (y/n)':CLS#0,4:fnum=n:K_Chk
1264 IF chk=1:cn%=0:ELSE nm%=1:F_clear:n=fnum:RETurn
1265 FOR n=stot%+1 TO ftot%
1266   IF fink%(n)=7
1267     cn%=cn%+1:CFile$(cn%,2)=DFile$(n,2)                  Note: Copy Stats
1268     CFile$(cn%,1)=DFile$(n,1,1+LEN(DDIR$) TO LEN(DFile$(n,1)))  Note: Copy Filename (less SubDIR)
1269   END IF
1270 END FOR n
1271 SD$=DD$:SDIR=DDIR$:TD$=DD$:TDIR$=DDIR$:F_Target
1272 END DEFine
```

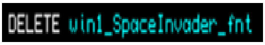

COPY   win1_boot


COPY   win1_QBITS_File(s) (y/n)

```
1274 DEFine PROCedure F_Target              Select Target DIRectory
1275 REPeat tag_lp
1276   CURSOR#0,24, 6:PRINT#0,'  COPY  '&SDIR$&' File(s) TO '&TD$&TDIR$:CLS#0,4
1277   CURSOR#0,24,20:PRINT#0,'Change (D)rive (S)ubDIR (C)OPY File(s)'
1278   k=CODE(INKEY$(#0,-1))
1279   SELect ON k
1280    =68,100:px%=138:SelDrv   :TD$=DD$             :REMark D (Drive)
1281    =83,115:px%=170:SubDIR :TDIR$=DDIR$           :REMark S (SubDIR)
1282    =67, 99:EXIT tag_lp                           :REMark C (Copy) Exit loop
1283   END SELect
1284 END REPeat tag_lp
1285 IF SD$&SDIR$=TD$&TDIR$:nm%=1:F_clear:n=fnum:RETurn :ELSE CLS#0:F_Copy2
1286 END DEFine
```


COPY QBITS_ File(s) TO win1_QBITS_
Change (D)rive (S)ubDIR (C)OPY File(s)

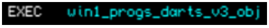
```
1288 DEFine PROCedure F_Copy2                  COPY the selected file(s)
1289 FOR n2=1 TO cn%
1290   str$=CFile$(n2,1):chk=1                 Note: CFile$(n2,1)=DFi;e$(n2,1) less SDIR$
1291   CURSOR#0,24,6:PRINT#0,'  COPY   '&str$&' TO '&TD$&TDIR$:CLS#0,4
1292   FOR n1=stot%+1 TO ftot%
1293    IF str$==DFile$(n1,1,1+LEN(TDIR$) TO LEN(DFile$(n1,1)))   Note: Target with same Filename
1294      INK#0,3:CURSOR#0,6,20:PRINT#0,CFile$(n2,2)&'     '&DFile$(n1,2)
1295      INK#0,5:CURSOR#0,340,6:PRINT#0,' Overwrite y/n':K_Chk
1296      IF chk=0:NEXT n2
1297    END  IF                  Note: chk=1 If Filename doesn't exist - Overwrite. Delete then Copy
1298    IF chk=1:DELETE TD$&TDIR$&str$:COPY SD$&SDIR$&str$ TO TD$&TDIR$&str$
1299   END FOR n1
1300 END FOR n2
1301 FileDIR                     Note: Display Updated Volume Info & Filenames of Target device
1302 END DEFine
```



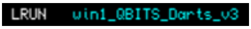
DELETE win1_SpaceInvader_fnt

```
1304 DEFine PROCedure F_Delete
1305 px%=96:mark%=7:nm%=stot%+1:st%=1:F_select:st%=0
1306 fnum=n:fdel%=ftot%
1307 FOR n=stot%+1 TO ftot%
1308  IF fink%(n)=7
1309   CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1310   IF chk=1:DELETE DD$&DFile$(n,1):fdel%=fdel%-1:fink%(n)=0:Fscr_posn:F_write
1311   IF chk=0:fink%(n)=5:Fscr_posn:F_write
1312  END IF
1313 END FOR n
1314 IF LEN(DDIR$)>0 AND fdel%=0:DELETE DD$&DDIR$:dl%=dl%-1:DDIR$=SubDIR$(dl%)
1315 IF fdel%<ftot%:FileDIR:ELSE nm%=1:F_clear:n=fnum
1316 END DEFine
```

```
1318 DEFine PROCedure F_Run(act)                                   Action Filename EXEC or LRUN
1319 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1320 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1321 IF chk=1 AND act=2:EXEC DD$&DFile$(n,1)
1322 IF chk=1 AND act=1:LRUN DD$&DFile$(n,1)
1323 fink%(n)=5:Fscr_posn:F_write
1324 END DEFine
```

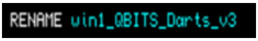


EXEC   win1_progs_darts_v3_obj



LRUN   win1_QBITS_Darts_v3

Note: EXEC runs Program as a Job and allows CTRL-C to switch back to SuperBASIC Interpreter. The LRUN Menu command is not functional if not actionable from QBITSFTidySE_obj.



RENAME win1_QBITS_Darts_v3

```
1326 DEFine PROCedure F_Rename
1327 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1328 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1329 IF chk=0:fink%(n)=5:Fscr_posn:F_write:RETurn                   IF NO Clear Highlight & RETurn
1330 INK#0,5:CURSOR#0,px%,6:PRINT#0,DDIR$:CLS#0,4
1331 str$=DFile$(n,1,1+LEN(DDIR$) TO LEN(DFile$(n,1)))
1332 sl%=LEN(str$):cp%=sl%+1:sm%=36-LEN(DDIR$):px%=px%+LEN(DDIR$)*6:Ln_Ed      Edit Filename
1333 IF str$='':str$=DFile$(n%,1):fink%=5:Fscr_posn:F_write:n=fnun:RETurn
1334 FOR n1=1 TO ftot%:
1335    IF str$==DFile$(n1,1)
1336       INK#0,5:CURSOR#0,24,20:PRINT#0,'Filename Exists':CLS#0,4      Note: Check if Rename Exists
1337       PAUSE 50:fink%(n)=5:Fscr_posn:F_write:RETurn
1338    END IF
1339 END FOR n1
1340 COPY DD$&DFile$(n,1) TO DD$&str$:DELETE DD$&DFile$(n,1):FileDIR
1341 END DEFine
```

Note: A Filename Change uses Copy and then Delete and does not check the available storage on target device. Please take note if a Large File is being renamed this might lead to the action being rejected and possible loss of Data!.

VIEW   win1_QBITS_Darts_v3

```
1343 DEFine PROCedure F_View
1344 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1345 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n Enter)':CLS#0,4
1346 fnum=0:K_Chk:IF chk=0:fink%(n)=5:Fscr_posn:F_write:RETurn
1347 CURSOR#0,240,20:PRINT#0,'<SPACEBAR> to continue... <ENTER> to Exit'
1348 CURSOR#0,160,20:PRINT#0,'Bytes: ':CLS#1
1349 OPEN_IN#9,DD$&DFile$(n,1):char%=0:fline%=0:row%=0:fbyts=0
1350 REPeat View_lp
1351   k$=INKEY$(#9,-1):IF EOF(#9):CLOSE#9:K_Chk:IF chk=2:EXIT View_lp
1352   IF chk=1:PRINT#1,k$;                                          ASCII Printout
1353   IF chk=2:CURSOR#1,6+char%*15,row%:PRINT#1,HEX$(CODE(k$),8)    HEX Printout
1354   char%=char%+1:fbyts=fbyts+1:CURSOR#0,200,20:PRINT#0,fbyts
1355   IF chk=1 AND char%>=74 OR chk=1 AND k$=CHR$(10)               ASCII New Line
1356     char%=0:fline%=fline%+1
1357     IF fline% MOD 16=0:IF INKEY$(-1)=CHR$(10):CLOSE#9:EXIT View_lp   ASCII New Page
1358   END IF
1359   IF chk=2 AND char% MOD 32=0                                   HEX New Line
1360     char%=0:fline%=fline%+1:row%=row%+10
1361     IF fline% MOD 16=0:IF INKEY$(-1)=CHR$(10):CLOSE#9:EXIT View_lp   HEX New Page
1362     IF row%>150:row%=150:SCROLL -10
1363   END IF
1364 END REPeat View_lp
1365 nm%=1:CLS#1:F_clear:n=fnum
1366 END DEFine
```

ZIP   win1_QBITS_QBITS_Darts_v3

```
1368 DEFine PROCedure F_ZIP
1369 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0:ALTKEY 'z'
1370 INK#0,7:CURSOR#0, 24,6:PRINT#0,'COMPILE ';DFile$(n,1);' (y/n Enter)':CLS#0,4
1371 K_Chk:IF chk=0:RETurn :END IF :IF chk=2:RTme=1:ELSE RTme=0
1372 CLS:WINDOW 364,138,72+gx,54+gy:PAPER 208:CLS:BORDER 1,7
1373 CSIZE 2,1:CURSOR 132,20:PRINT 'COMPILER':CSIZE 0,0
1374 CLS#0:CURSOR#0,66,6:QLIB_USE:IF eck=1:Load_Qlib(RTme):eck=0:PAUSE 60
1375 CLS#0:CURSOR#0,66,6:cl%=INT(LEN(DFile$(n,1))/2)
1376 CURSOR 24, 60:PRINT FILL$(' ',18-cl%)&'LIBERATE ';DD$&DFile$(n,1);',';
1377 CURSOR 92, 80:PRINT 'Press ALT-z to LOAD & COMPILE'
1378 CURSOR 56,120:PRINT 'Then CTRL-SPACE & ALT-f for QBITS_FTidySE'
1379 ALTKEY 'z','LOAD '&DD$&DFile$(n,1);'LIBERATE '&DD$&DFile$(n,1)&',';CHR$(10)
1380 STOP
1381 END DEFine
```

Note: QL Platform Requires SMSQ/E O/S and uses QLIBERATOR Sinclair QL Forum Edition 2020 for QPC.

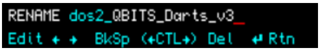
```
1383 DEFine PROCedure Load_Qlib(RTme)
1384 add1=RESPR(15064):LBYTES dev$&'Qlib_sys',add1:CALL add1
1385 add2=RESPR(49004):LBYTES dev$&'Qlib_obj',add2:CALL add2
1386 REMark Q_LIBERATOR  Settings
1387 IF RTme=0:QLIB_USE dev$,dev$,72+gx,54+gy,"0011010100" :REMark RunTime Off
1388 IF RTme=1:QLIB_USE dev$,dev$,72+gx,54+gy,"0011110100" :REMark RunTime On
1389 END DEFine
```

Note: QLIB_USE attributes : Load_Device for QLIB_OBJ, Load_Device for QLIB_HELP, Window x,y coordinates, “Option Bits” 1 to 10 -STAT -DEBUG -LINES -NAMES -RUN -AUTOF -BEEP -WINDS 9&10 -00 Reserved.

1391 REMark **QBITS FTidy Line Editor**

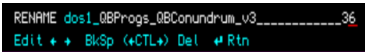
Note: The Line Editor restricts characters to numeric 0 to 9 [ASCII 48-57], the UPPER/lower-case Alphabet A-z [ASCII 65-90 & 97-122] plus underscore '_' [95]. Position the character highlight (Underline **Ln_Cur**) with Left Right Cursors, then Add (**Add_Chr**) a new or Delete (**Del_Chr**) existing Character.

```
1393 DEFine PROCedure Ln_Ed
1394 INK#0,5:CURSOR#0,24,20
1395 PRINT#0,'Edit ← → BkSp (← CTL→) Del ↵ Rtn':BLOCK#0,2,4,198,22,5
1396 REPeat Ed_lp
1397   Ln_Prn:Ln_Cur:k$=INKEY$(#0,-1):k=CODE(k$)
1398   SELect ON k
1399     = 10:Str_chk:EXIT Ed_lp
1400     = 48 TO 57, 65 TO 90,95, 97 TO 122:Ln_Prn:Add_chr        ASCII codes available for Filenames
1401     =194:IF cp%>1:cp%=cp%-1:Del_chr                          cp% cursor position
1402     =202:Del_chr
1403     =192:IF cp%>1:cp%=cp%-1
1404     =200:IF cp%<sl%+1:cp%=cp%+1                              sl%  string length
1405   END SELect
1406 END REPeat Ed_lp
1407 END DEFine
```



```
RENAME dos2_QBITS_Darts_v3_
Edit ← →  BkSp (←CTL→) Del  ↵ Rtn
```

```
1409 DEFine PROCedure Str_chk
1410 REPeat str_lp
1411  IF '_' INSTR str$(LEN(str$))=1:cp%=sl%:Del_chr:Ln_Prn     Note: Removes any EOL '_'
1412  IF '_' INSTR str$(LEN(str$))=0:PAUSE 30:EXIT str_lp
1413END REPeat str_lp
1414 END DEFine
```



```
RENAME dos1_QBProgs_QBConundrum_v3____________36
Edit ← →  BkSp (←CTL→) Del  ↵ Rtn
```

```
1416 DEFine PROCedure Ln_Prn
1417 IF LEN(str$)>sm%:str$=str$(1 TO sm%):cp%=sm%              Note: Truncate to sm% max string length
1418 INK#0,7:CURSOR#0,px%,6:PRINT#0,str$:CLS#0,4
1419 END DEFine

1421 DEFine PROCedure Ln_Cur
1422 BLOCK#0,8,1,px%+cp%*6-6,15,2                               Note: px% x start position
1423 END DEFine

1425 DEFine PROCedure Add_chr
1426 IF cp% =1 AND sl%=0     :str$=str$&k$                          add to string
1427 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&k$&str$(cp% TO sl%)  add in string
1428 IF cp%>=1 AND cp%=sl%:str$=str$(1 TO cp%-1)&k$&str$(cp%)         add one before end
1429 IF cp%> 1 AND cp%>sl%:str$=str$&k$                             add to end of string
1430 IF cp%=sm%:str$(cp%)=k$                                        change last character
1431 IF sl%<sm%:sl%=sl%+1:ELSE sl%=sm%                     sl% string length sm% max length
1432 IF cp%<sm%:cp%=cp%+1:ELSE cp%=sm%                     cp% character position
1433 END DEFine

1435 DEFine PROCedure Del_chr
1436 IF cp%=sl%:str$=str$(1 TO sl%-1):sl%=sl%-1                              delete end of string
1437 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&str$(cp%+1 TO sl%):sl%=sl%-1  delete in string
1438 IF cp%=sm%:str$=str$(1 TO sm%-1):cp%=cp%-1:sl%=sm%-1                    delete last character
1439 IF cp%=1 AND sl%=1:str$="":sl%=0                                        Null string
1440 END DEFine
```

```
1442 REMark QBITS FTidy Graphics
```

```
1444 DEFine PROCedure KExit(ch,col,x,y)
1445 INK#ch,col:CURSOR#ch,x,y,1,-5:PRINT#ch,'½':LINE#ch,x,y TO x,y-.4
1446 LINE#ch,x+1.2,y+1.8 TO x-1,y+1.8 TO x-1,y-2 TO x+1.6,y-2
1447 END DEFine
```

```
1449 DEFine PROCedure KInfo(ch,col,x,y)
1450 INK#ch,col:CIRCLE#ch,x,y,2:LINE#ch,x,y-1.2 TO x,y:POINT#ch,x,y+.5
1451 END DEFine
```

```
1453 DEFine PROCedure GDrive(col,x,y)
1454 FILL#2,1:INK#2,col
1455 LINE#2,x-4,y TO x,y+2 TO x+4,y+1 TO x+4,y-1 TO x,y-3.5 TO x-4,y-2 TO x-4,y
1456 FILL#2,0:INK#2,0
1457 LINE#2,x-4,y TO x,y-1 TO x+4,y+1:LINE#2,x,y-3.5 TO x,y-1
1458 LINE#2,x-3.6,y-1.5 TO x-.5,y-2.6:INK#2,7
1459 END DEFine
```

```
1461 DEFine PROCedure GFolder(col,x,y)
1462 FILL#2,1:INK#2,col
1463 LINE#2,x-3,y+2 TO x-2.6,y+2.4 TO x-1,y+2.4 TO x,y+2 TO x+2,y+2
1464 LINE#2 TO x+2,y+1 TO x+3,y+1 TO x+2,y-1.8 TO x-3,y-1.8 TO x-3,y+2
1465 FILL#2,0:INK#2,0
1466 LINE#2,x-3,y-1.8 TO x-2,y+1 TO x+4,y+1:INK#2,7
1467 END DEFine
```

Note: As an exercise the Graphics below were seen as possible symbols for use with the Pointer Environment.

```
2000 REMark QBITS Pointer Graphics
```

```
2002 DEFine PROCedure GDisk(ch,col,x,y)
2003 FILL#ch,1:INK#ch,col:LINE#ch,x-2,y+2 TO x+1,y+2 TO x+2,y+1
2004 LINE#ch TO x+2,y-2 TO x-2,y-2 TO x-2,y+2:FILL#ch,0
2005 END DEFine
```

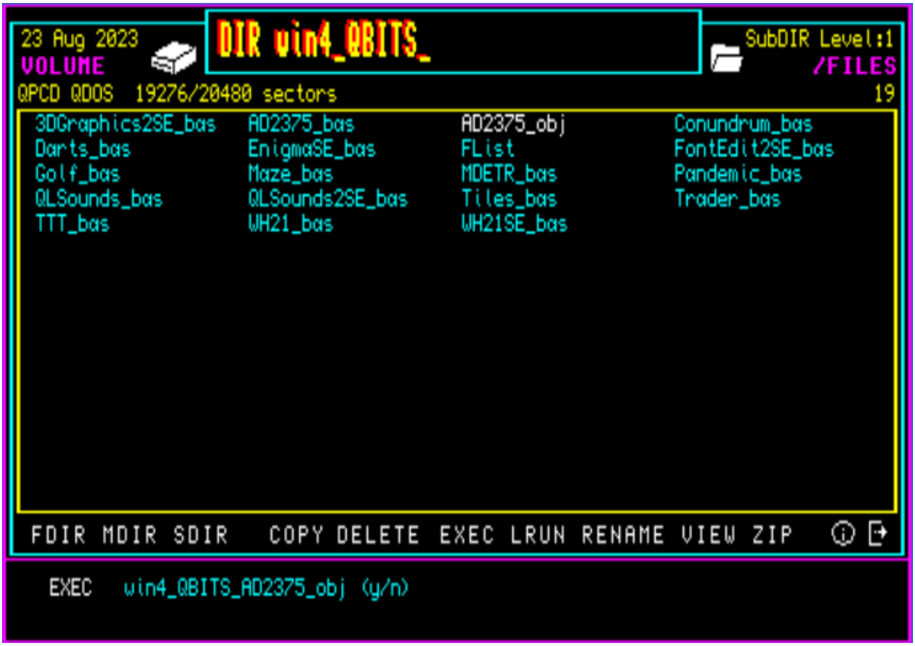
```
2007 DEFine PROCedure GSave(ch,col,x,y)
2008 GDisk ch,col,x,y:INK#ch,0:LINE#ch,x-1,y TO x+1,y
2009 LINE#ch,x-1,y TO x+1.5,y:LINE#ch,x-1,y-1 TO x+1.5,y-1
2010 END DEFine
```

```
2012 DEFine PROCedure GLoad(ch,col,x,y)
2013 GDisk ch,col,x,y:INK#ch,0:LINE#ch,x-1.6,y+.5 TO x+1.2,y+.5
2014 LINE#ch TO x-1.6,y-1.6 TO x+1.2,y-1.6 TO x-1.6,y+.5
2015 END DEFine
paper
```

```
2017 DEFine PROCedure GCopy(ch,col,x,y)
2018 GDisk ch,col,x,y:INK#ch,0:LINE#ch,x-.5,y-.5 TO x+1.8,y-.5
2019 LINE#ch,x-1.5,y+1 TO x-1.5,y-1.5 TO x+1.8,y-1.5
2020 END DEFine
```

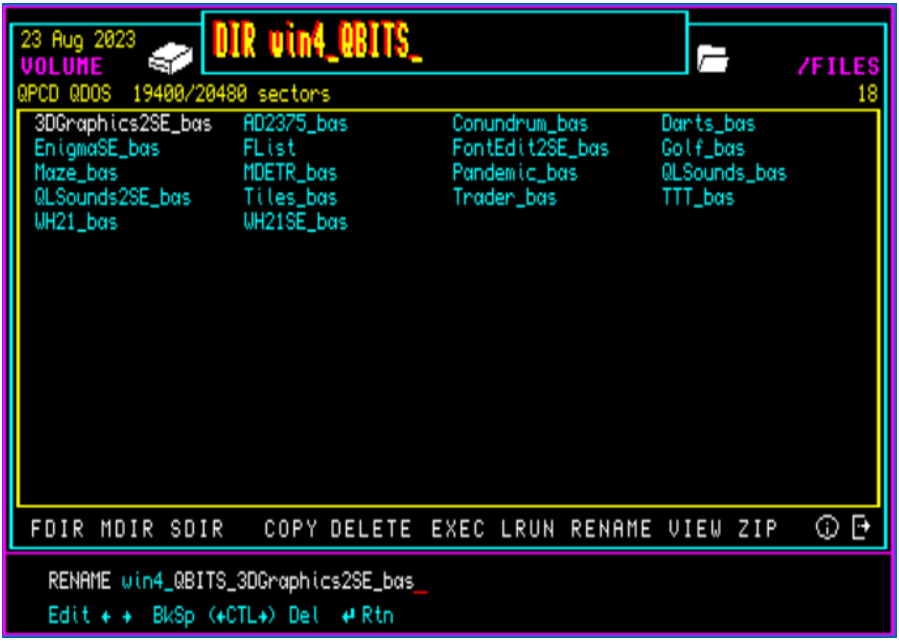
```
2022 DEFine PROCedure GDelete(ch,col,x,y)
2023 GDisk ch,col,x,y:INK#ch,0:LINE#ch,x-.4,y+1 TO x+.6,y+1
2024 LINE#ch,x-1,y+.5 TO x+1,y+.5 TO x+1,y-1.5 TO x-1,y-1.5
2025 LINE#ch TO x-1,y+.5:LINE#ch,x,y-1.5 TO x,y+.5
2026 END DEFine
```

```
2028 DEFine PROCedure GRename(ch,col,x,y)
2029 GDisk ch,col,x,y:INK#ch,0:LINE#ch,x,y+.5 TO x,y-1.5
2030 LINE#ch,x-1.2,y+.5 TO x+1.2,y+.5:LINE#ch,x-1.2,y-1.5 TO x+1.2,y-1.5
2031 END DEFine
```



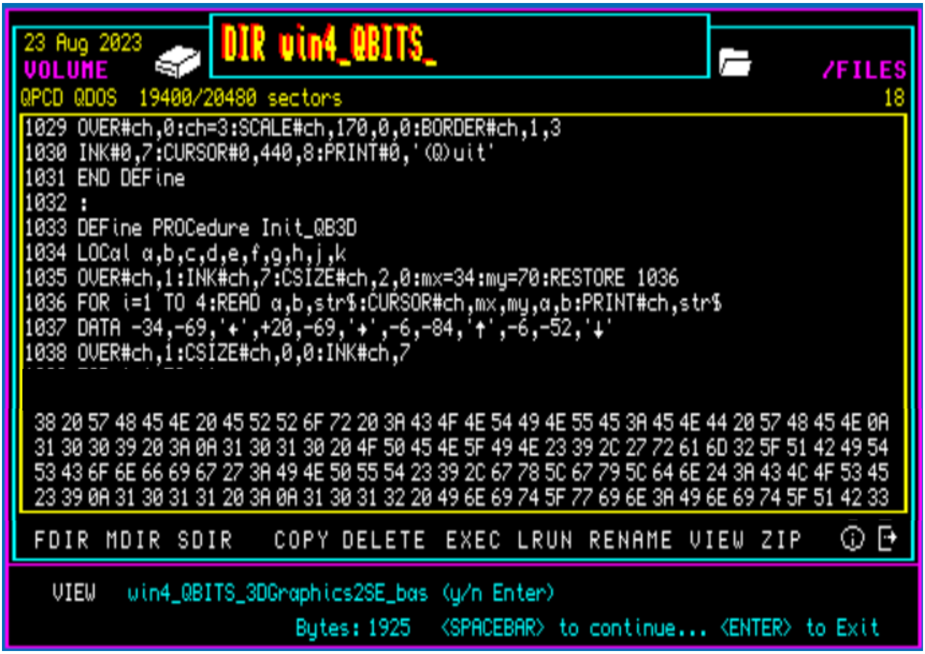
```
23 Aug 2023          DIR win4_QBITS_                SubDIR Level:1
VOLUME                                                      /FILES
QPCD QDOS   19276/20480 sectors                                 19
 3DGraphics2SE_bas  AD2375_bas         AD2375_obj         Conundrum_bas
 Darts_bas          EnigmaSE_bas       FList              FontEdit2SE_bas
 Golf_bas           Maze_bas           MDETR_bas          Pandemic_bas
 QLSounds_bas       QLSounds2SE_bas    Tiles_bas          Trader_bas
 TTT_bas            WH21_bas           WH21SE_bas
```

```
  EXEC   win4_QBITS_AD2375_obj  (y/n)
```



```
23 Aug 2023          DIR win4_QBITS_
VOLUME                                                      /FILES
QPCD QDOS   19400/20480 sectors                                 18
 3DGraphics2SE_bas  AD2375_bas         Conundrum_bas      Darts_bas
 EnigmaSE_bas       FList              FontEdit2SE_bas    Golf_bas
 Maze_bas           MDETR_bas          Pandemic_bas       QLSounds_bas
 QLSounds2SE_bas    Tiles_bas          Trader_bas         TTT_bas
 WH21_bas           WH21SE_bas
```

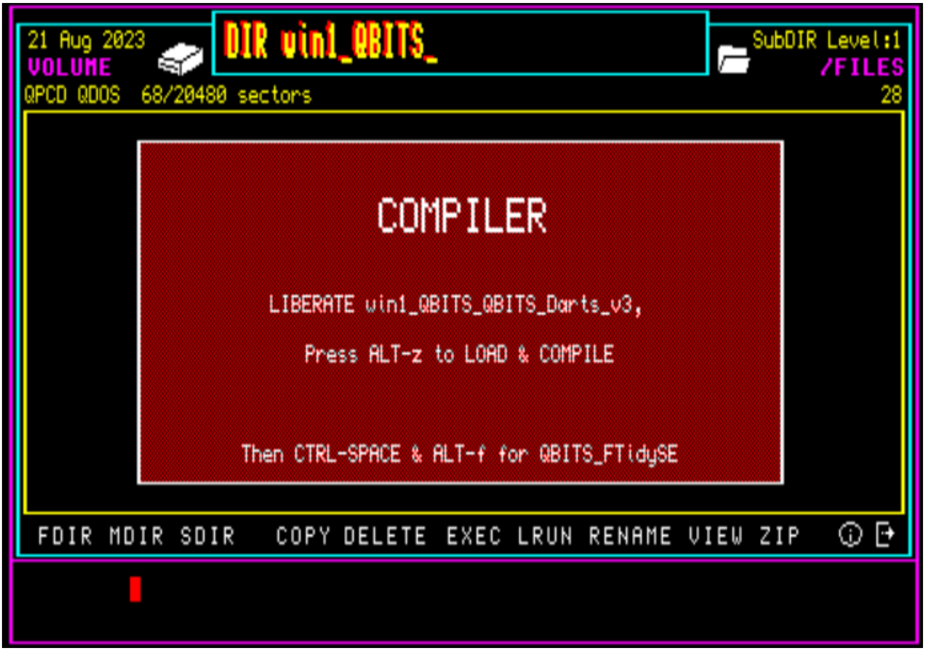
```
  RENAME win4_QBITS_3DGraphics2SE_bas_
  Edit ← →  BkSp (←CTL→) Del  ↵ Rtn
```



```
1029 OVER#ch,0:ch=3:SCALE#ch,170,0,0:BORDER#ch,1,3
1030 INK#0,7:CURSOR#0,440,8:PRINT#0,'(Q)uit'
1031 END DEFine
1032 :
1033 DEFine PROCedure Init_QB3D
1034 LOCal a,b,c,d,e,f,g,h,j,k
1035 OVER#ch,1:INK#ch,7:CSIZE#ch,2,0:mx=34:my=70:RESTORE 1036
1036 FOR i=1 TO 4:READ a,b,str$:CURSOR#ch,mx,my,a,b:PRINT#ch,str$
1037 DATA -34,-69,'←',+20,-69,'→',-6,-84,'↑',-6,-52,'↓'
1038 OVER#ch,1:CSIZE#ch,0,0:INK#ch,7

 38 20 57 48 45 4E 20 45 52 52 6F 72 20 3A 43 4F 4E 54 49 4E 55 45 3A 45 4E 44 20 57 48 45 4E 0A
 31 30 30 39 20 3A 0A 31 30 31 30 20 4F 50 45 4E 5F 49 4E 23 39 2C 27 72 61 6D 32 5F 51 42 49 54
 53 43 6F 6E 66 69 67 27 3A 49 4E 50 55 54 23 39 2C 67 78 5C 67 79 5C 64 6E 24 3A 43 4C 4F 53 45
 23 39 0A 31 30 31 31 20 3A 0A 31 30 31 32 20 49 6E 69 74 5F 77 69 6E 3A 49 6E 69 74 5F 51 42 33
```

VIEW win4_QBITS_3DGraphics2SE_bas (y/n Enter)

Bytes: 1925 <SPACEBAR> to continue... <ENTER> to Exit


21 Aug 2023
VOLUME
DIR win1_QBITS_
SubDIR Level:1
/FILES
QPCD QDOS 68/20480 sectors
28

COMPILER

LIBERATE win1_QBITS_QBITS_Darts_v3,

Press ALT-z to LOAD & COMPILE

Then CTRL-SPACE & ALT-f for QBITS_FTidySE

FDIR MDIR SDIR COPY DELETE EXEC LRUN RENAME VIEW ZIP