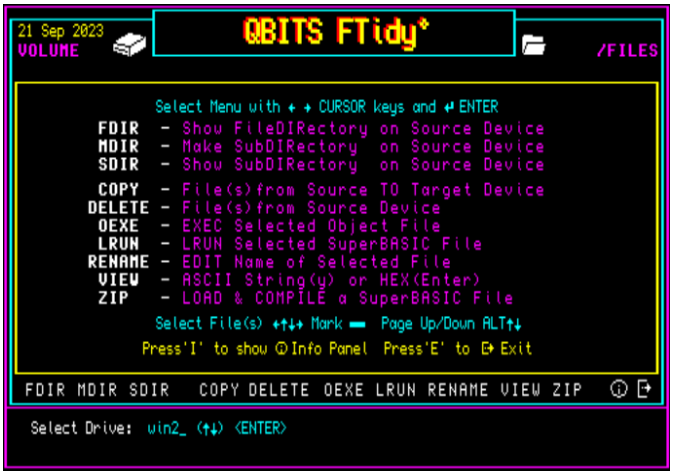


Introduction

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. **QBITS FTidy** was submitted to QUANTA shortly before Howard Clase published his **FTidy** program in QLWorld Sep1988.

QBITS FTidy128

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. **COPY** and **DELETE** commands allowed for single File or batch processing of multiple files. **SelDev** was used for **Source** and **Target** selection. The **Print** command was for Printer export of File lists. These were later dropped for **LOAD/LRUN**.

The **FDIR** with **SubDIR**ectories was added with Millennium updates. The latest version now includes **EXEC** and **VIEW** with **ASCII** or **HEX** Readout plus a **ZIP** option 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.

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 by ▬ Spacebar ↵ Enter Keys. The Menu can also be Selected by individual Keys **F M S C D O L R V Z**. For Help screen Press 'I' for ⓘ Info, Press 'E' ⍈ to leave Exit the program.

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 was given in the QL Users Guide Chapter 16. A few tweaks with Array and Variable Names and Voilá! The Filenames of **FList** are selected sequentially by a **FOR loop** and compared within a **REPeat Loop** 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**.

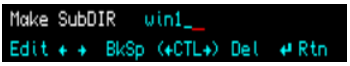
QBITS FTidySE SubDIRectories

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**rectory levels. **QBITS FTidy** limit is set at six **SubDIR** levels.

The File DIRectory [**FDIR**] or SubDIRirectories [**SDIR**] if present, display under **Volume** the **Device name, Volume/Sectors** and under **Files** the **Number** held in Directory. 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.

QBITS FTidySE MDIR

Using the Line Editor create a new **SubD**irectory name. QBITS six level **SubDIR** allocation…



DIM SubDIR$(6,24) where SubDIR$(1) = "**SD1**_" and SubDIR$(6) access = "**SD1_SD2_SD3_SD4_SD5_SD6**_"

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 Filenames are shown Press (**C**)opy and the selected Filenames from **Source DIR**ectory are then copied across to the **Target** device. Any files of same name in destination device prompts for further action with an **Overwrite 'y/n'**.

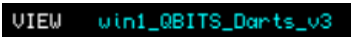
Similarly, for **DELETE** the highlighted Filename is confirmed with '**y/n**' before Deleting.

QBITS FTidySE OEXE / LRUN

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

QBITS FTidySE RENAME

Select an existing Filename (**file$**) and **Edit** the string (**str$**) with the **QBITS Line Editor**. Action 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 the old file Deleted.



QBITS FTidySE VIEW

View was seen as a useful option in being able to OPEN and read the contents a File. **S/SuperBASIC** Files are in **Plain Text**, others can be **Data** lists or a mix of code blocks and ASCII characters as in **Executable** computer readable machine code. **FTidy View** gives the option to display as **ASCII characters (y)** or in **Hex Bytes (Enter).**

QBITS FTidySE ZIP

Select a **SupeBASIC** File ie. with a **'_bas'** suffix. A Prompt (y/n Entr) appears. Press '**y**' if Runtime is resident or '**Enter**' if to be linked with program. Several programs can share **QLib_RUN** if installed or it can be linked as part of the Compile process. The compiler **Q_LIBERATOR** is installed if absent. Pressing 'n' returns to **FTidy Menu** bar.

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

QBITS FTidySE use of **LIBERATOR**
An early notion was to include a link to Compile SuperBASIC Programs. In computing ZIP is a term for File(s) or Routines compressed into a package and descriptively can imply to go faster. A Spectrum BASIC Compiler written by N Goodwin was called ZIP. I hope he won't object to QBITS reuse of the word!

My choice of Compliers being SUPERCHARGE, TURBO or LIBERATOR and aim to maintain some compatibility across various QL Platforms. For the present QBITS_FTidySE relies on the SMSQ/E O/S being in place and use of the QLIBERATOR Sinclair QL Forum Edition 2020 for QPC2. Users should refer to the User Manuals for a better understanding of limitations with other environments.

QBITS FTidySE and **Handling of Screen Positioning**
Filenames are listed across the screen in columns and rows, 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 a Filename List exceeds the available screen space, upon reaching the bottom of the window the bottom row can be scrolled up and a new group of filenames added. To keep track a Line pointer identifies the position in the Filename list and a Row pointer for the screen position. A filename position can be identified by subtracting the Row pointer from the File Line pointer. A similar arrangement is utilised for scrolling the screen down so as to add rows at the screen top.

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 storage system is composed of 512-byte blocks, addressing is via an associated File Pointer by Block number (sector) and Byte number within that block. The QL Tech Guide describes the 64 Bytes File 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 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
KExit	Key Graphic Image created for Exit
KInfo	Key Graphic Image created for Info/Help
GDrive	Hard Drive symbol
GFolder	File Folder Symbol

QBITS FTidySE_bas Code

```
1000 REMark QBITS_FTidySE_bas [QBITS FTidy SE 2023 Review - QPC2]

1002 dev$='win2_':MODE 4:gx=0:gy=0    :REMark Basic Settings

1004 WHEN ERRor :eck=1:CONTINUE:END WHEN
```

Note: **dev$** is set to the default source device which is accessed to Import **QBITSConfig** settings. If FDIR fails to find a Device or any Files then it returns **DEVICE ERROR** and instead of the Interpreter halting the program WHEN ERRor will CONTINUE the Program.

```
1006 REMark Import QBITSConfig Settings - QPC2
1007 OPEN_IN#9,dev$&'QBITSConfig':INPUT#9,gx\gy\dn$\dev$\dn%\dm%
1008 DIM Drv$(dm%,5):FOR d=0 TO dm%:INPUT#9,Drv$(d):END FOR d:CLOSE#9

1010 REMark DIM Arrays : Set max% number of Files
1011 max%=300
1012 DIM DFile$(max%,2,36),fink%(max%),CFile$(max%,2,36),cink%(max%)
1013 DIM Comd$(58),key$(3,52),help$(9,48),Time$(20)
1014 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

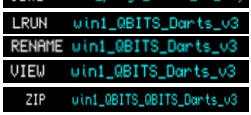
```
1016 REMark FTidy Setup & Comand Menu

1018 Init_win:Cmd_Menu

1020 DEFine PROCedure Init_win
1021 OPEN#5,scr_:WINDOW#5,512,256,gx,gy          :PAPER#5,0:BORDER#5,1,3:CLS#5
1022 OPEN#4,scr_:WINDOW#4,504,214,gx+4,gy+7      :PAPER#4,0:BORDER#4,1,5:CSIZE#4,1,0
1023 OPEN#3,scr_:WINDOW#3,280,26,gx+114,gy+2     :PAPER#3,0:BORDER#3,1,5:CLS#3
1024 WINDOW#2,504,220,gx+4,gy+2 :PAPER#2,0                  :CSIZE#2,0,0:INK#2,7
1025 WINDOW#1,496,162,gx+8,gy+42:PAPER#1,0       :BORDER#1,1,6:CSIZE#1,0,0:INK#1,7
1026 WINDOW#0,512, 34,gx,gy+222 :PAPER#0,0       :BORDER#0,1,3:CSIZE#0,0,0:INK#0,7
1027 OVER#4,1:INK#4,3
1028 FOR i=0 TO 1:CURSOR#4,  4+i,12:PRINT#4,'VOLUME'
1029 FOR i=0 TO 1:CURSOR#4,450+i,12:PRINT#4,'/FILES'
1030 OVER#4,0:INK#2,6:Time$=DATE$
1031 CURSOR#2,6,7:PRINT#2,Time$(10 TO 11)&Time$(5 TO 9)&Time$(1 TO 4)
1032 SCALE#2,100,0,0:SCALE#1,100,0,0:F_Title 2,66,'QBITS FTidy':F_Info
1033 END DEFine
```


```
1035 DEFine PROCedure F_Title (cs%,x%,Title$)
1036 CLS#3:CSIZE#3,cs%,1:OVER#3,1
1037 INK#3,2 :FOR i=0 TO 1:CURSOR#3,x%+i,1:PRINT#3,Title$
1038 INK#3,6 :FOR i=1 TO 2:CURSOR#3,x%+i,2:PRINT#3,Title$:OVER#3,0
1039 END DEFine
```

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

```
1041 DEFine PROCedure F_Info
1042 help$(0)=' FDIR       - Show FileDIRectory on Source Device'
1043 help$(1)=' MDIR       - Make SubDIRectory  on Source Device'
1044 help$(2)=' SDIR       - Show SubDIRectory  on Source Device'
1045 help$(3)=' COPY       - File(s)from Source TO Target Device'
1046 help$(4)='DELETE      - File(s)from Source Device'
1047 help$(5)=' EXEC       - EXEC Selected Object File'
1048 help$(6)=' LRUN       - LRUN Selected SuperBASIC File'
1049 help$(7)='RENAME      - Edit Name of Selected File'
1050 help$(8)=' VIEW       - ASCII String(y) or HEX(Enter)'
1051 help$(9)=' ZIP        - Load & COMPILE a SuperBASIC File'
1052 OVER#1,1:CSIZE#1,1,0:CLS#1:sp=20
1053 FOR hp=0 TO 8
1054   INK#1,7:FOR i=0 TO 1:CURSOR#1,54+i,sp+hp*10:PRINT#1,help$(hp,1 TO 9);
1055   INK#1,5:PRINT#1,help$(hp,10 TO):IF hp=2:sp=24
1056 END FOR hp
1057 OVER#1,0:CSIZE#1,0,0:INK#1,5:GDrive 7,31,92:GFolder 7,136,91
1058 CURSOR#1,106,  8:PRINT#1,'Select Menu with ← → CURSOR keys and ← ENTER'
1059 CURSOR#1,106,128:PRINT#1,'Select File(s)  ←↑ ↓→ Mark    Page Up/Down ALT ↑ ↓'
1060 BLOCK#1,2,4,334,10,5:BLOCK#1,12,3,256,132,5:INK#1,6
1061 CURSOR#1, 96,142:PRINT#1,"Press'I' to show   Info Panel   Press 'Q' to   Quit"
1062 KInfo 1,6,94,8:KExit 1,6,167,8: cmd=1:mm%=1:mx%=9:lptr%=0:INK#0,7:px%=108:DD$=''
1064 END DEFine
```


```
1066 DEFine PROCedure Cmd_Menu
1067 DIM cx%(9),Cmd$(10,6):RESTORE 1075:FOR i=1 TO 10:READ cx%(i),Cmd$(i)
1068 DATA 16,'FDIR',50,'MDIR',90,'SDIR',144,'COPY',182,'DELETE'
1069 DATA 240,'EXE',280,'LRUN',320,'RENAME',376,'VIEW',416,'ZIP'
1070 STRIP#4,0:INK#4,7:FOR i=1 TO 10:CURSOR#4,cx%(i),200:PRINT#4,Cmd$(i)
1071 KInfo 2,7,158,4.6:KExit 2,7,164,4.4:SelDrv:CLS#0
1072 REPeat Cmd_lp
1073   STRIP#4,6:INK#4,0:CURSOR#4,cx%(cmd),200:PRINT#4,Cmd$(cmd)
1074   k$=(INKEY$(#0,-1)):k=CODE(k$)
1075   STRIP#4,0:INK#4,7:CURSOR#4,cx%(cmd),200:PRINT#4,Cmd$(cmd)
1076   SELect ON k
1077     =192:cmd=cmd-1:IF cmd<1:cmd=10      :REMark← Left   Cursor min-max
1078     =200:cmd=cmd+1:IF cmd>10:cmd=1      :REMark →Right Cursor max=min
1079     =73,105   :h%=1:F_Info:h%=0         :              :REMark (I)nfo
1080     =69,101   :IF dn$='':STOP :ELSE LRUN dn$ :STOP   :REMark(E)xit
1081     =10,65 TO 122:IF k<>10:cmd=1+(k$ INSTR 'FfMmSsCcDdEeLlRrVvZz')DIV 2
1082       SELect ON cmd
1083         = 1:SelDrv                :CLS#0
1084         = 2::MakeDIR              :CLS#0
1085         = 3::SubDIR               :CLS#0
1086         = 4:IF ftot%>0:F_Copy     :CLS#0
1087         = 5:IF ftot%>0:F_Delete   :CLS#0
1088         = 6:IF ftot%>0:F_Run 2    :CLS#0  :REMark EXEC
1089         = 7:IF ftot%>0:F_Run 1    :CLS#0  :REMark LRUN
1090         = 8:IF ftot%>0:F_Rename   :CLS#0
1091         = 9:IF ftot%>0:F_View     :CLS#0  :REMark ASCII(y) HEX(Entr)
1092         =10:IF ftot%>0:F_ZIP      :CLS#0  :REMark Compiler
1093       END SELect
1094   END SELect
1095 END REPeat Cmd_lp
1096 END DEFine
```



```
1098 REMark QBITS FTidy DIRectory Management

1100 DEFine PROCedure SelDrv
1101 DIM SubDIR$(6,24):dl%=0:OD$=DD$:dch=0
1102 IF cmd=1:INK#0,7:CURSOR#0,18,6:PRINT#0,'Select Drive: ':px%=108
1103 REPeat Dr_lp
1104   INK#0,5:CURSOR#0,px%,6:PRINT#0,Drv$(dn%)&' (⬆⬇) <ENTER>':CLS#0,4
1105   k=CODE(INKEY$(#0,-1))
1106   SELect ON k
1107     =10:DD$=Drv$(dn%):EXIT Dr_lp          :REMark Enter Select Drive
1108     =208:dn%=dn%-1:IF dn%< 0:dn%=15    :REMark Up
1109     =216:dn%=dn%+1:IF dn%>15:dn%= 0    :REMark Down
1110   END SELect
1111 END REPeat Dr_lp
1112 IF OD$=DD$:RETurn :ELSE DDIR$='':FileDIR
1113 END DEFine
```

Select Drive: win1_ (↑↓) <ENTER>

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

```
1115 DEFine PROCedure FileDIR
1116 CLS#0:DELETE DD$&DDIR$&'FList':F_Title 1,4,'DIR '&DD$&DDIR$
1117 IF cmd<2:INK#0,5:CURSOR#0,24,6:PRINT#0,'Files being Selected...'
1118 OPEN_NEW#9,DD$&DDIR$&'FList':STAT#9,DD$&DDIR$:WSTAT#9,DD$&DDIR$:CLOSE#9
1119 OPEN_IN #9,DD$&DDIR$&'FList':INPUT#9,DName$\DSec$:n=1:ftot%=0
1120 REPeat DIR_lp
1121   IF EOF(#9) OR n>max%:ftot%=n-1:CLOSE#9:EXIT DIR_lp
1122   INPUT#9,DFile$(n,1):fink%(n)=5
1123   IF '>' INSTR DFile$(n,1)=0:INPUT#9,DFile$(n,2):n=n+1:ELSE n=n+1
1124 END REPeat DIR_lp
1125 BLOCK#2,88,10,414,7,0:IF dl%>0:CURSOR#2,414,7:PRINT#2,'SubDIR Level:';dl%
1126 BLOCK#2,480,10,4,29,0:INK#2,6 :CURSOR#2,4,29:PRINT#2,DName$;'  ';DSec$
1127 CURSOR#2,466,29:PRINT#2,FILL$(' ',5-LEN(ftot%))&ftot%
1128 IF ftot%=0:stot%=0:CLS:F_Title 1,4,'DEVICE ERROR':RETurn
1129 F_Sort:nm%=1:nx%=ftot%:lptr%=0:CLS#1:F_clear:n=1
1130 END DEFine
```

Files being Selected...

SubDIR Level:1 /FILES

QBITSProgs QDOS 88527528/430077944 sectors

```
1132 DEFine PROCedure F_Sort
1133 FOR sn=1 TO ftot%
1134   p=sn:comp$=DFile$(p,1):info$=DFile$(p,2)
1135   REPeat Sort_lp
1136     IF comp$>=DFile$(p-1,1):EXIT Sort_lp
1137     DFile$(p,1)=DFile$(p-1,1):DFile$(p,2)=DFile$(p-1,2):p=p-1
1138   END REPeat Sort_lp
1139   DFile$(p,1)=comp$:DFile$(p,2)=info$
1140 END FOR sn
1141 ntop=1:nsel=1:stot%=0
1142 FOR sn=1 TO ftot%
1143   IF '>' INSTR DFile$(sn,1)
1144     comp$=DFile$(sn,1):info$=DFile$(sn,2):nsel=sn-1
1145     FOR p=nsel TO ntop STEP -1
1146       DFile$(p+1,1)=DFile$(p,1):DFile$(p+1,2)=DFile$(p,2)
1147     END FOR p
1148     DFile$(ntop,1)=comp$:DFile$(ntop,2)=info$:ntop=ntop+1:stot%=stot%+1
1149   END IF
1150 END FOR sn
1151 END DEFine
```

```
Make SubDIR   win1_
```

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

```
1153 DEFine PROCedure MakeDIR
1154 md%=24-LEN(DDIR$):px%=138+LEN(DDIR$)*6
1155 IF md%<3:CURSOR#0,24,6:PRINT#0,'Lowest Level Reached':PAUSE 50:RETurn
1156 INK#0,7 :CURSOR#0,24,6:PRINT#0,'Make SubDIR    ';:INK#0,5:PRINT#0,DD$&DDIR$
1157 cp%=1:sl%=0:sm%=md%:str$='':Ln_Ed:IF LEN(str$)=0:RETurn
1158 CURSOR#0,px%+LEN(str$)*6,6:PRINT#0,' (y/n)':K_Chk
1159 IF chk=1
1160   FOR n=1 TO stot%:IF DDIR$&str$ INSTR DFile$(n,1):RETurn
1161   MAKE_DIR DD$&DDIR$&str$:FileDIR
1162 END IF
1163 END DEFine
```

```
SubDIR win1_ ↑↓
```

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

```
1165 DEFine PROCedure SubDIR
1166 INK#0,7:CURSOR#0,24,6:PRINT#0,'SubDIR ';:INK#0,5:PRINT#0,DD$&DDIR$;
1167 INK#0,7:PRINT#0,' ⬆⬇ ':CLS#0,4:k=CODE(INKEY$(#0,-1))
1168 IF k=208 AND dl%>=0   :Sub_up          dl% DIRectory Level
1169 IF k=216 AND stot%>=0:Sub_dn          stot% Sub Total
1170 END DEFine

1172 DEFine PROCedure Sub_up
1173 SubDIR$(dl%)='':dl%=dl%-1:DDIR$=SubDIR$(dl%):FileDIR:RETurn
1174 END DEFine

1176 DEFine PROCedure Sub_dn
1177 IF stot%<1 OR dl%=6:RETurn
1178 px%=96:mark%=5:n=1:nm%=1:nx%=stot%:st%=1:F_select:st%=0
1179 INK#0,5:CURSOR#0,px%,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1180 IF chk=1
1181   DDIR$=DFile$(n,1,1 TO flen%-3)&'_':dl%=dl%+1:nm%=stot%:nx%=ftot%:FileDIR
1182   SubDIR$(dl%)=DDIR$:CURSOR#0,px%+12,6:PRINT#0,DDIR$:CLS#0,4
1183 END IF
1184 nm%=1:nx%=ftot%:F_clear:n=1
1185 END DEFine
```

```
SubDIR dos2_Test -> (y/n)
```

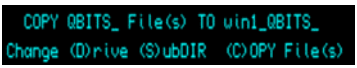
QBITS FTidy SubDIRectories

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, QBITS File Tidy is limited this to just SIX Sub levels, 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_**"

```
1187 REMark QBITS FTidy Filename Display

1189 DEFine PROCedure F_select                          filename: select and highlight in screen location
1190 IF cmd>3:INK#0,7:CURSOR#0,24,6:PRINT#0,Cmd$(cmd):CLS#0,4     Note: Check a File Command
1191 INK#0,5:CURSOR#0,66,6:PRINT#0,DD$&DDIR$                          Device & SubDIR
1192 REPeat Sel_lp
1193  Fscr_posn:fink%(n)=7:F_write:fink%(n)=5:k=CODE(INKEY$(#0,-1))
1194  SELect ON k
1195    =192:Fscr_posn:F_write:n=n  -1          :REMark Back      1
1196    =200:Fscr_posn:F_write:n=n +1          :REMark Forward   1
1197    =208:Fscr_posn:F_write:n=n  -4          :REMark Up        1 Row
1198    =216:Fscr_posn:F_write:n=n +4          :REMark Down      1 Row
1199    =209:Fscr_posn:F_write:n=n -60         :REMark Up        1 Page
1200    =217:Fscr_posn:F_write:n=n+60          :REMark Down      1 Page
1201    = 32:fink%(n)=mark%:F_write:n=n+1      :REMark           mark Filename mark% in Highlight Colour
1202    = 10:fink%(n)=7:CURSOR#0,0,20:CLS#0,4:RETurn             fink%(n) File Ink Highlight Colour
1203  END SELect
1204 END REPeat Sel_lp
1205 END DEFine

1207 DEFine PROCedure Fscr_posn                        calculate screen position
1208 IF n<nm%:n=nm%                                   n file number nm% min nx% max
1209 IF n>nx%:n=nx%                                   Note: n floating point – necessary with QDOS FOR loops
1210 fptr%=n-1:frow%=(fptr% DIV 4)
1211 IF frow%>(15+lptr%):Fscr_up:n=fptr%+1:Fscr_posn  frow% file row fptr% file pointer lptr% line pointer
1212 IF frow%<(  0+lptr%):Fscr_dn:n=fptr%+1:Fscr_posn
1213 srow%=frow%-lptr%:scol%=(fptr% MOD 4)*20         srow% screen row scol% screen column
1214 END DEFine

1216 DEFine PROCedure Fscr_up
1217 lptr%=lptr%+1:SCROLL#1,-10:n=(lptr%+15)*4:srow%=15
1218 FOR i=0 TO 3:scol%=i*20:n=n+1:F_write
1219 END DEFine

1221 DEFine PROCedure Fscr_dn
1222 lptr%=lptr%-1:SCROLL#1,10:n=(lptr%)*4:srow%=0
1223 FOR i=0 TO 3:scol%=i*20:n=n+1:F_write
1224 END DEFine

1226 DEFine PROCedure F_write
1227 IF n>ftot% OR n<1:RETurn
1228 flen%=LEN(DFile$(n,1)):slen%=LEN(DDIR$)           flen% Filename length slen% SubDIRectory length
1229 IF flen%-slen%>18:flen%=18+slen%
1230 INK#1,fink%(n):CURSOR#1,8+scol%*6,srow%*10        fink% file ink print colour
1231 PRINT#1,DFile$(n,1,1+slen% TO flen%)&FILL$(' ',18+slen%-flen%)          Filename - Main Screen
1232 INK#0,5:CURSOR#0,px%,6:PRINT#0,DFile$(n,1):CLS#0,4:IF st%<>1:RETurn    Filename - Action Window
1233 INK#0,5:CURSOR#0,24,20:PRINT#0,'Select using ←↑↓→ ▬ Alt ↑↓ ↵ ':CLS#0,4
1234 BLOCK#0,12,3,130,24,5:BLOCK#0,2,4,198,22,5:CURSOR#0,300,6:PRINT#0,DFile$(n,2)  File Stats - Action
1235 END DEFine    Use of BLOCK for Spacebar and Return Tail

1237 DEFine PROCedure F_clear                          Clear marked files
1238 FOR sc=1 TO ftot%:fink%(sc)=5                     ftot% File Total sc set clear (ink colour)
1239 fs%=(lptr%*4)+1:fe%=(lptr%+16)*4:IF fe%>ftot%:fe%=ftot%
1240 FOR n=fs% TO fe%:Fscr_posn:F_write                fs% file start fe% file end
1241 END DEFine
```

```
1243 REMark QBITS FTidy File Mamgement

1245 DEFine PROCedure K_Chk                              Note: INKEY$ Check
1246 REPeat chk_lp
1247   k=CODE(INKEY$(#0,-1))
1248   SELect ON k=78,110:chk=0:EXIT chk_lp              Note: Answer 'n,N' no
1249   SELect ON k=89,121:chk=1:EXIT chk_lp              Note: Answer 'y,Y' yes
1250   SELect ON k=10:IF cmd=9 OR cmd=10:chk=2:EXIT chk_lp:  Note: Answer 'Enter' for VIEW & ZIP
1251 END REPeat chk_lp
1252 END DEFine

1254 DEFine PROCedure F_Copy                    Select Single/Multiple Files
1255 px%=96:mark%=7:nm%=stot%+1:st%=1:F_select:st%=0
1256 CURSOR#0,px%,6:PRINT#0,DDIR$;'File(s) (y/n)':CLS#0,4:fnum=n:K_Chk
1257 IF chk=1:cn%=0:ELSE nm%=1:F_clear:n=fnum:RETurn
1258 FOR n=stot%+1 TO ftot%
1259   IF fink%(n)=7
1260     cn%=cn%+1:CFile$(cn%,2)=DFile$(n,2)             Note: Copy Stats
1261     CFile$(cn%,1)=DFile$(n,1,1+LEN(DDIR$) TO LEN(DFile$(n,1)))  Note: Copy Filename (less SubDIR)
1262   END IF
1263 END FOR n
1264 SD$=DD$:SDIR=DDIR$:TD$=DD$:TDIR$=DDIR$:F_Target
1265 END DEFine

1267 DEFine PROCedure F_Target                  Select Target DIRectory
1268 REPeat tag_lp
1269   CURSOR#0,24, 6:PRINT#0,'  COPY       Files(s)   TO  '&TD$&TDIR$:CLS#0,4
1270   CURSOR#0,24,20:PRINT#0,'Change (D)rive (S)ubDIR (C)OPY File(s)'
1271   k=CODE(INKEY$(#0,-1))
1272   SELect ON k
1273     =68,100:px%=138:SelDrv   :TD$=DD$                :REMark D (Drive)
1274     =83,115:px%=170:SubDIR :TDIR$=DDIR$              :REMark S (SubDIR)
1275     =67, 99:EXIT tag_lp                              :REMark C (Copy) Exit loop
1276   END SELect
1277 END REPeat tag_lp
1278 IF SD$&SDIR$=TD$&TDIR$:nm%=1:F_clear:n=fnum:RETurn :ELSE CLS#0:F_Copy2
1279 END DEFine

1281 DEFine PROCedure F_Copy2                    COPY the selected file(s)
1282 FOR n2=1 TO cn%
1283   str$=CFile$(n2,1):chk=1                          Note: CFile$(n2,1)=DFi;e$(n2,1) less SDIR$
1284   CURSOR#0,24,6:PRINT#0,'  COPY   '&str$&' TO '&TD$&TDIR$:CLS#0,4
1285   FOR n1=stot%+1 TO ftot%
1286     IF str$==DFile$(n1,1,1+LEN(TDIR$) TO LEN(DFile$(n1,1)))   Note: Target with same Filename
1287       INK#0,3:CURSOR#0,6,20:PRINT#0,CFile$(n2,2)&'     '&DFile$(n1,2)
1288       INK#0,5:CURSOR#0,340,6:PRINT#0,' Overwrite y/n':K_Chk
1289       IF chk=0:NEXT n2
1290     END  IF                  Note: chk=1 Filename exist 'y=yes' Delete then Copy
1291     IF chk=1:DELETE TD$&TDIR$&str$:COPY SD$&SDIR$&str$ TO TD$&TDIR$&str$
1292   END FOR n1
1293 END FOR n2
1294 FileDIR                     Note: Display Updated Volume Info & Filenames of Target device
1295 END DEFine
```



```
1297 DEFine PROCedure F_Delete                                  DELETE win1_SpaceInvader_fnt
1298 px%=96:mark%=7:nm%=stot%+1:st%=1:F_select:st%=0
1299 fnum=n:fdel%=ftot%
1300 FOR n=stot%+1 TO ftot%
1301  IF fink%(n)=7
1302   CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1303   IF chk=1:DELETE DD$&DFile$(n,1):fdel%=fdel%-1:fink%(n)=0:Fscr_posn:F_write
1304   IF chk=0:fink%(n)=5:Fscr_posn:F_write
1305  END IF
1306 END FOR n
1307 IF LEN(DDIR$)>0 AND fdel%=0:DELETE DD$&DDIR$:dl%=dl%-1:DDIR$=SubDIR$(dl%)
1308 IF fdel%<ftot%:FileDIR:ELSE nm%=1:F_clear:n=fnum
1309 END DEFine

1311 DEFine PROCedure F_Run(act)                                 Action Filename EXEC or LRUN
1312 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1313 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1314 IF chk=1 AND act=2:EXEC DD$&DFile$(n,1)   OEXE   win1_progs_darts_v3_obj
1315 IF chk=1 AND act=1:LRUN DD$&DFile$(n,1)   LRUN   win1_QBITS_Darts_v3
1316 fink%(n)=5:Fscr_posn:F_write
1317 END DEFine
```

Note: Select OEXE to EXEC Program as a JOB that allows CTRL-C to switch back to SuperBASIC Interpreter. Exit the program cancels the JOB. The LRUN Menu command is not actionable from QBITSFTidySE_obj.

```
1319 DEFine PROCedure F_Rename                                  RENAME win1_QBITS_Darts_v3
1320 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1321 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n)':CLS#0,4:K_Chk
1322 IF chk=0:fink%(n)=5:Fscr_posn:F_write:RETurn                IF NO Clear Highlight & RETurn
1323 INK#0,5:CURSOR#0,px%,6:PRINT#0,DDIR$:CLS#0,4
1324 str$=DFile$(n,1,1+LEN(DDIR$) TO LEN(DFile$(n,1)))
1325 sl%=LEN(str$):cp%=sl%+1:sm%=36-LEN(DDIR$):px%=px%+LEN(DDIR$)*6:Ln_Ed        Edit Filename
1326 IF str$='':str$=DFile$(n%,1):fink%=5:Fscr_posn:F_write:n=fnun:RETurn
1327 FOR n1=1 TO ftot%:
1328    IF str$==DFile$(n1,1)
1329       INK#0,5:CURSOR#0,24,20:PRINT#0,'Filename Exists':CLS#0,4      Note: Check if Rename Exists
1330       PAUSE 50:fink%(n)=5:Fscr_posn:F_write:RETurn
1331    END IF
1332 END FOR n1
1333 COPY DD$&DFile$(n,1) TO DD$&str$:DELETE DD$&DFile$(n,1):FileDIR
1334 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!.

```
1336 DEFine PROCedure F_View
1337 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0
1338 CURSOR#0,96,6:PRINT#0,DFile$(n,1)&' (y/n Enter)':CLS#0,4
1339 fnum=0:K_Chk:IF chk=0:fink%(n)=5:Fscr_posn:F_write:RETurn
1340 CURSOR#0,240,20:PRINT#0,'<SPACEBAR> to continue... <ENTER> to Exit'
1341 CURSOR#0,160,20:PRINT#0,'Bytes: ':CLS#1
1342 OPEN_IN#9,DD$&DFile$(n,1):char%=0:fline%=0:row%=0:fbyts=0
1343 REPeat View_lp
1344   k$=INKEY$(#9,-1):IF EOF(#9):CLOSE#9:K_Chk:IF chk=2:EXIT View_lp
1345   IF chk=1:PRINT#1,k$;                                          ASCII Printout
1346   IF chk=2:CURSOR#1,6+char%*15,row%:PRINT#1,HEX$(CODE(k$),8)    HEX Printout
1347   char%=char%+1:fbyts=fbyts+1:CURSOR#0,200,20:PRINT#0,fbyts
1348   IF chk=1 AND char%>=74 OR chk=1 AND k$=CHR$(10)               ASCII New Line
1349     char%=0:fline%=fline%+1
1350     IF fline% MOD 16=0:IF INKEY$(-1)=CHR$(10):CLOSE#9:EXIT View_lp   ASCII New Page
1351   END IF
1352   IF chk=2 AND char% MOD 32=0                                   HEX New Line
1353     char%=0:fline%=fline%+1:row%=row%+10
1354     IF fline% MOD 16=0:IF INKEY$(-1)=CHR$(10):CLOSE#9:EXIT View_lp   HEX New Page
1355     IF row%>150:row%=150:SCROLL -10
1356  END IF
1367 END REPeat View_lp
1358 nm%=1:CLS#1:F_clear:n=fnum
1359 END DEFine
```

VIEW win1_QBITS_Darts_v3

```
1361 DEFine PROCedure F_ZIP
1362 px%=96:mark%=5:nm%=stot%+1:st%=1:F_select:st%=0:ALTKEY 'z'
1363 INK#0,7:CURSOR#0, 24,6:PRINT#0,'COMPILE ';DFile$(n,1);' (y/n Enter)':CLS#0,4
1364 K_Chk:IF chk=0:RETurn :END IF :IF chk=2:RTme=1:ELSE RTme=0
1365 CLS:WINDOW 364,138,72+gx,54+gy:PAPER 208:CLS:BORDER 1,7
1366 CSIZE 2,1:CURSOR 132,20:PRINT 'COMPILER':CSIZE 0,0
1367 CLS#0:CURSOR#0,66,6:QLIB_USE:IF eck=1:Load_Qlib(RTme):eck=0:PAUSE 60
1368 CLS#0:CURSOR#0,66,6:cl%=INT(LEN(DFile$(n,1))/2)
1369 CURSOR 24, 60:PRINT FILL$(' ',18-cl%)&'LIBERATE ';DD$&DFile$(n,1);',';
1370 CURSOR 92, 80:PRINT 'Press ALT-z to LOAD & COMPILE'
1371 CURSOR 56,120:PRINT 'Then CTRL-SPACE & ALT-f for QBITS_FTidySE'
1372 ALTKEY 'z','LOAD '&DD$&DFile$(n,1);'LIBERATE '&DD$&DFile$(n,1)&',';CHR$(10)
1373 STOP
1374 END DEFine
```

ZIP win1_QBITS_QBITS_Darts_v3

Note: Select OEXE (y/n Enter) – Enter will Compile with Qlib_Runtime. QL Platform Requires SMSQ/E O/S and QLIBERATOR Sinclair QL Forum Edition 2020 for QPC2.

```
1376 DEFine PROCedure Load_Qlib(RTme)
1377 add1=RESPR(15064):LBYTES dev$&'Qlib_sys',add1:CALL add1
1378 add2=RESPR(49004):LBYTES dev$&'Qlib_obj',add2:CALL add2
1379 REMark Q_LIBERATOR  Settings
1380 IF RTme=0:QLIB_USE dev$,dev$,72+gx,54+gy,"0011010100" :REMark RunTime Off
1381 IF RTme=1:QLIB_USE dev$,dev$,72+gx,54+gy,"0011110100" :REMark RunTime On
1382 END DEFine
```

Note: QLIB_USE attributes: Load_Device for QLIB_OBJ, Load_Device for QLIB_HELP, Window x,y coordinates, “Option Bits” 1 to 8 - STATS-DEBUG-LINES-NAMES-RUN-AUTO-BEEP-WINDS - [9&10 Reserved].

```
1384 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.

```
1386 DEFine PROCedure Ln_Ed
1387 INK#0,5:CURSOR#0,24,20
1388 PRINT#0,'Edit ← → BkSp (← CTL→) Del ↵ Rtn':BLOCK#0,2,4,198,22,5
1389 REPeat Ed_lp
1390   Ln_Prn:Ln_Cur:k$=INKEY$(#0,-1):k=CODE(k$)
1391   SELect ON k
1392     = 10:Str_chk:EXIT Ed_lp
1393     = 48 TO 57, 65 TO 90,95, 97 TO 122:Ln_Prn:Add_chr        ASCII codes available for Filenames
1394     =194:IF cp%>1:cp%=cp%-1:Del_chr                          cp% cursor position
1395     =202:Del_chr
1396     =192:IF cp%>1:cp%=cp%-1
1397     =200:IF cp%<sl%+1:cp%=cp%+1                              sl%  string length
1398   END SELect
1399 END REPeat Ed_lp
1400 END DEFine
```

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

```
1402 DEFine PROCedure Str_chk
1403 REPeat str_lp
1404   IF '_' INSTR str$(LEN(str$))=1:cp%=sl%:Del_chr:Ln_Prn      Note: Removes any EOL '_'
1405   IF '_' INSTR str$(LEN(str$))=0:PAUSE 30:EXIT str_lp
1406 END REPeat str_lp
1407 END DEFine
```

```
RENAME dos1_QBProgs_QBConundrum_v3__________36
Edit ← →  BkSp (←CTL→) Del  ↵ Rtn
```

```
1409 DEFine PROCedure Ln_Prn
1410 IF LEN(str$)>sm%:str$=str$(1 TO sm%):cp%=sm%
1411 INK#0,7:CURSOR#0,px%,6:PRINT#0,str$:CLS#0,4
1412 END DEFine
```

Note: Truncate to sm% max string length

```
1414 DEFine PROCedure Ln_Cur
1415 BLOCK#0,8,1,px%+cp%*6-6,15,2
1416 END DEFine
```

Note: px% x start position

```
1418 DEFine PROCedure Add_chr
1419 IF cp%  =1 AND sl%=0      :str$=str$&k$                          add to string
1420 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&k$&str$(cp% TO sl%)  add in string
1421 IF cp%>=1 AND cp%=sl%:str$=str$(1 TO cp%-1)&k$&str$(cp%)         add one before end
1422 IF cp%>  1 AND cp%>sl%:str$=str$&k$                             add to end of string
1423 IF cp%=sm%:str$(cp%)=k$                                         change last character
1424 IF sl%  <sm%:sl%=sl%+1  :ELSE sl%=sm%                  sl% string length sm% max length
1425 IF cp%<sm%:cp%=cp%+1:ELSE cp%=sm%                      cp% character position
1426 END DEFine
```

```
1428 DEFine PROCedure Del_chr
1429 IF cp%=sl%:str$=str$(1 TO sl%-1):sl%=sl%-1                                delete end of string
1430 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&str$(cp%+1 TO sl%):sl%=sl%-1  delete in string
1431 IF cp%=sm%:str$=str$(1 TO sm%-1):cp%=cp%-1:sl%=sm%-1                      delete last character
1432 IF cp%=1 AND sl%=1:str$="":sl%=0                                          Null string
1433 END DEFine
```

```
1435 REMark QBITS FTidy Graphics

1437 DEFine PROCedure KExit(ch,col,x,y)
1438 INK#ch,col: LINE#ch,x+1,y+2 TO x-1,y+2 TO x-1,y-2 TO x+2,y-2
1439 LINE#ch,x,y TO x+3,y:LINE#ch,x+2,y+1 to x+3,y to x=2,y-1 to x+2,y+1
1440 END DEFine

1442 DEFine PROCedure KInfo(ch,col,x,y)
1443 INK#ch,col:CIRCLE#ch,x,y,2:LINE#ch,x,y-1.2 TO x,y:POINT#ch,x,y+.5
1444 END DEFine

1446 DEFine PROCedure GDrive(col,x,y)
1447 FILL#2,1:INK#2,col
1448 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
1449 FILL#2,0:INK#2,0
1450 LINE#2,x-4,y TO x,y-1 TO x+4,y+1:LINE#2,x,y-3.5 TO x,y-1
1451 LINE#2,x-3.6,y-1.5 TO x-.5,y-2.6:INK#2,7
1452 END DEFine

1454 DEFine PROCedure GFolder(col,x,y)
1455 FILL#2,1:INK#2,col
1456 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
1457 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
1458 FILL#2,0:INK#2,0
1459 LINE#2,x-3,y-1.8 TO x-2,y+1 TO x+4,y+1:INK#2,7
1460 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

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
```

Floppy Disk