



Introduction

The aspirations for a QBITS Font Editor began in the eighties. The Program was never finished for release, other events taking up ever more of my time. The concept was to Load the QLFONT sets into memory, display them to screen and Select one. A simple Bitmap would then be displayed with the option to change the bit patten and save to memory. After a number of Fonts had been Modified in this way the whole Font set could be saved as a Font file to storage for use with other Programs.

The QDOS uses an ASCII Code set for Key Control, some of which produce printable Character Fonts. The QDOS Bitmaps for these are stored in two sets Codes 32 to 127, are the common alphanumeric, some maths signs and brackets etc. the second are Codes 128 to 191 the extended character set.

QLFONT1 is 875 Bytes and QLFONT2 587 Bytes. The Format is for the first two Bytes to hold the Lowest Code number followed by Total number of Fonts. For QLFONT1 this will be 32 96 and for QLFONT2 128 64. The Bitmap for each character is 9 Bytes. This provides each with an 8x9 matrix. The QDOS character generator when using different CSIZES does not always display all the Bits. This can lead to some interesting and at time frustrating outcomes when using Modified Fonts for say Retro Gaming.

QBITS QLFONT Modifier Layout

The screen display had to be easy and intuitive to use as I hope the finished Program does. The Display above shows all of the Modifier aspects. On start up the default QL Character fonts are displayed on the left. The Bitmap on the right to show the bits set to 0's & 1's for the current highlighted character. When returning from the Bitmap the Character under review is then displayed in the different CSIZES. Top right below the (L)oad (S)ave (R)eset (E)xit information is shown applicable to the action requested.

QBITS QLFont Modifier Navigation

For the two Font sets navigate using the cursor keys and Select the Highlighted Character with Spacebar. This actions the Bitmap Grid, again navigate with cursor keys to highlight a Grid Cell and toggle the binary bit between 0 and 1 with Spacebar. Select 'N' to return without changing the existing Bitmap, 'Y' to write the changed bit pattern into memory.

QBITS (L)oad (S)ave

Actions for saving and retrieving created Font sets. Press 'L' to load from a set of existing Font files. Press 'S' to save the modified Font set(s). The presented information allows a change of storage device with Left/Right cursor keys, change between set QLFont1 or QLFont2 with Spacebar and Select a file number between 0 and 9 with Up/Down cursor keys then a 'Y/N' prompt. 'Y' will continue action. 'N' aborts the requested action. Statements are given for 'File NOT Found', 'DEVICE ERROR' and 'Overwrite Y/N'

QBITS (R)eset (E)xit

For Reset press 'R' which brings up a prompt 'Y/N' , 'N' aborts and 'R' reloads the default Fonts. Pressing 'E' will again prompt with Y/N , 'N' returns to program, 'Y' will close opened channels and free Memory before halting the Program. If desired an LRUN can be added to start a Boot or Menu Program.

QBITS Storage Devices

The default Storage Device 'Dev\$' at Code Line 1005 needs to be set so Program can access the accompanying **QLFont1&2** Files. A range of Storage Device names are provided at Code Line 1028 and loaded into an Array Drv\$(8,5) these can be changed by the user. Set 'dn%' at Code Line 1044 to chosen storage default ie. Drv\$(dn%).

QBITS Font Installation

General information on installation of Fonts, first space is allocated in the common heap (RAM memory) then for both sets the bytes required are rounded off to even values for LBYTES to work properly for example:

```
FBASE1 = ALCHP(876)    LBYTES FLP1_FONT1,FBASE1
FBASE2 = ALCHP(588)    LBYTES FLP1_FONT2,FBASE2
```

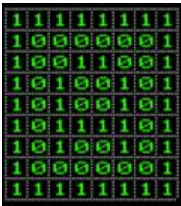
Activate with keyword **CHAR_USE#ch,FBASE1,FBASE2** and repeat for other **channels** where Font is needed. When finished release memory with the RECHP(FBASE?). Use RESPR(8&6+588) for a more permanent installation, where Fonts remain until Switch Off or a System Reset.

Note: With the **QLFont1** & **QLFont2** assigned to RAM the Bitmap for each individual character can be Read and Overwritten with the PEEK and POKE commands. Each entry is identified by an offset from the **QLFont1** & **2** assigned Base address **FBASE1** & **FBASE2** given by **ALCHP**. The first 2 Bytes holding the Lowest code & Total number of Characters font, followed by the Bitmaps each being multiples of 9 Bytes.

QBITS QLFont Modifier CSIZES

As mentioned, the Font Generator for the QL does not retrospectively produce all the bit pattern held by a character's Bitmap across the range of CSIZES.

As an example, the bit Pattern 'A' shown was Surrounded with a box.



Not all CSIZES showed the box and some have an interesting outcome.

QPC11 SMSQ/E

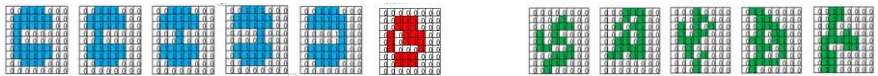


Qemulator



Program use of Font Designs

Apart from what Character Font were originally intended, I have chosen two programs Giro Rescue and Dino that use redefined Fonts. Here Giro Rescue modifies the QLFont1 and Dino replaces the QLFont2 extended set.



Giro Rescue

! # \$ % @for Rescue Pod



For Escape Pod ^

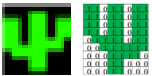
() * + _ for Ship Debruy



Dino



Uses all of the extended Fonts.



QPC11 SMSQ/E



Qemulator



Conclusion, when using Fonts to create Retro Game features be aware of the bit pattern displays generated for the different CSIZES.

QBITS QLFont Modifier Coding

1000 REMark **QBITS_FontMod_v1** (QBITS Font Modifier c1989 new for 2022)

1002 gx=0:gy=0 :REMark Set gx gy for High Res Platforms
1003 DIM drv\$(8,5) :REMark Devise Drives Set dn% for default drive
1004 DIM Fnt\$(9,8) :REMark Array for Font Pattern
1005 Dev\$='flp1_' :REMark default storage Device
1006 FBase1=ALCHP(875) :REMark Heap Memory to hold QLFont grp 1
1007 FBase2=ALCHP(587) :REMark Heap Memory to hold QLFont grp 2





QPC11_v2
gx=40:gy=40

Dev\$='dos1_'

1009 WHEN ERRor
1010 eck=1:CONTINUE
1011 END WHEN

1013 Init_Screen:FontMain

1015 DEFine PROCEDURE Init_Screen

1016 WINDOW#0,512,32,gx,gy+224 :PAPER#0,0 :CSIZE#0,0,0:BORDER#0,1,3:CLS#0
1017 WINDOW#1,512,224,gx,gy :PAPER#1,0 :CSIZE#1,0,0:BORDER#1,1,3:CLS#1
1018 WINDOW#2,512,224,gx,gy :PAPER#2,0 :CSIZE#2,0,0:BORDER#2,1,3:CLS#2
1019 OPEN#3,scr_:WINDOW#3,276,178,gx+12,gy+22 :CSIZE#3,3,0:INK#3,4
1020 OPEN#4,scr_:WINDOW#4,136,112,gx+340,gy+88 :CSIZE#4,3,0:INK#4,4
1021 CSIZE#2,2,1:OVER#2,1
1022 INK#2,2:FOR i=0 TO 1:CUSOR#2,6+i,8:PRINT#2,'QBITS QLFont Modifier'
1023 INK#2,6:FOR i=0 TO 1:CUSOR#2,8+i,6:PRINT#2,'QBITS QLFont Modifier'
1024 CSIZE#2,0,0:OVER#2,0
1025 CUSOR#1,312,18:PRINT#1,'(L)oad (S)ave (R)eset (E)xit'
1026 CUSOR#1,12,208:PRINT#1,'Select     :BLOCK#1,12,3,80,212,7
1027 RESTORE 1028:FOR d=1 TO 8:READ str\$:drv\$(d)=str\$
1028 DATA 'mdv1_', 'mdv2_', 'flp1_', 'flp2_', 'win1_', 'win2_', 'dos1_', 'dos2_'
1029 END DEFine

1031 DEFine PROCEDURE FontReset

[Load Default QLFonts](#)

1032 IF chk=1:CUSOR#1,420,32:PRINT#1,'Y/N':PAUSE:IF KEYROW(5)<>64:RETurn
1033 LBYTES drv\$(dn%)&'QLFont1',FBase1:cn1=96
1034 LBYTES drv\$(dn%)&'QLFont2',FBase2:cn2=64
1035 FontSets cn1,cn2 :FontGrid:cn=32:sg%=1:sf%=0:Files\$='QLFont1_'
1036 END DEFine

1038 DEFine PROCEDURE FontExit

1039 CUSOR#1,460,32:PRINT#1,'Y/N':PAUSE:IF KEYROW(5)<>64:RETurn
1040 CLOSE#4:CLOSE#3:CLS#2:RECHP FBase1:RECHP FBase2:STOP
1041 END DEFine

Note: Exit closes open channels and releases heap memory. STOP can be replaced with an LRUN command to return to another program such as LRUN flp1_Boot or win1_Progs_Menu etc.

```

1043 DEFine PROCEDURE FontMain
1044 dn%=3:sx=0:sy=0:x=1:y=1:chk=0:eck=0:FontReset
1045 REPEAT Main_Ip
1046   cn=32+cx+(cy-2)*15:BLOCK#1,160,20,320,32,0
1047   IF cn>127:cn=cn-9
1048   IF cy=8 AND cx>5:cx=5:cn=127
1049   IF cy=13 AND cx>3:cx=3:cn=191
1050   CURSOR#1,318,68:PRINT#1,'Character Dec: ',cn,' Hex: ',HEX$(cn,8),'
1051   FontChar cx,cy,7:FontPeek
1052   K=CODE(INKEY$(-1))
1053   FontChar cx,cy,0:IF K<>32:CLS#4:FontGrid
1054   SElect ON K
1055     =192:cx=cx-1:cn=cn-1:IF cx<0:cx=0
1056     =200:cx=cx+1:cn=cn+1:IF cx>14:cx=14
1057     =208:cy=cy-1:cn=cn-15:IF cy<0:cy=0
1058     =216:cy=cy+1:cn=cn+15:IF cy>11:cy=11
1059     =32:FontMod:REMark Modify Character
1060     =108,76:FontFiles 1:REMark (L)oad
1061     =115,83:FontFiles 2:REMark (S)ave
1062     =114,82:FontReset 1:REMark (R)eset
1063     =101,69:FontExit:REMark (E)xit
1064   END SElect
1065 END REPEAT Main_Ip
1066 END DEFine

```

QPC11_v2
dn%=7

Note: cn ASCII Char code number
checks for correct code
in the display positions

```

(L)oad (S)ave (R)eset (E)xit
LBYTES dos1_0LFont2_0 Y/N
++ Select ==>+<

```

```

(L)oad (S)ave (R)eset (E)xit
SBYTES dos1_0LFont2_0 Y/N
++ Select ==>+<

```

Note: Switch from Main Character display to Bitmap display and change the Bitmap. To switch back press 'N' for no change or 'Y' to update Character. Once Changes to set are complete the updated Fonts can be saved. Press 'S' and select device Font Set 1 or 2 and file num 0 to 9. If device not available DEVICE ERROR will be displayed. If file already exist you are prompted to Overwrite Y/N. Press 'L' to load a Font set.

```

1068 DEFine PROCEDURE FontMod
1069 CURSOR#1,332,208:PRINT#1,'Select <=> Change Y/N'
1070 BLOCK#1,12,3,400,212,7:FontPeek
1071 REPEAT Chg_Ip
1072   FontBit x,y,7:K=CODE(INKEY$(-1)):FontBit x,y,248
1073   SElect ON K
1074     =192:x=x-1:IF x<1:x=1
1075     =200:x=x+1:IF x>8:x=8
1076     =208:y=y-1:IF y<1:y=1
1077     =216:y=y+1:IF y>9:y=9
1078     =110,78:EXIT Chg_Ip:REMark (N)o Return
1079     =121,89:FontPoke:EXIT Chg_Ip:REMark (Y)es Change Font
1080     =32:Bitswap:REMark Swap 0-1
1081   END SElect
1082 END REPEAT Chg_Ip
1083 BLOCK#1,148,10,332,208,0:CLS#4:FontGrid
1084 END DEFine

```

Modify the Bitmap of Selected Char

Toggle Bits on Selected Grid Location

```

1086 DEFine PROCEDURE Bitswap
1087 IF Fnt$(y,x)='0':Fnt$(y,x)='1':ELSE Fnt$(y,x)='0'
1088 CURSOR#4,-12+x*16,-8+y*12:PRINT#4,Fnt$(y,x)
1089 END DEFine

```

1091 **DEFINE PROCEDURE FontSets(cn1,cn2)**

Display Char Fonts

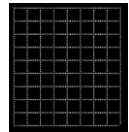
```
1092 fx=2:fy=26:CHAR_USE#3,FBase1,FBase2:CLS#3:CLS#4
1093 FOR c=32 TO 31+cn1
1094   CURSOR#3,fx,fy:PRINT#3,CHR$(c)
1095   fx=fx+18:IF fx>260:fx=2:fy=fy+12
1096 END FOR c
1097 CURSOR#1,172,121:PRINT#1,'FontSet 1 32 96':fx=2:fy=110
1098 FOR c=128 TO 127+cn2
1099   CURSOR#3,fx,fy:PRINT#3,CHR$(c)
1100   fx=fx+18:IF fx>260:fx=2:fy=fy+12
1101 END FOR c
1102 CURSOR#1,172,190:PRINT#1,'FontSet 2 128 64'
```



1105 **DEFINE PROCEDURE FontGrid**

Grid for Bitmap

```
1106 FOR i=0 TO 8:BLOCK#4,1,108,2+i*16,2,248
1107 FOR i=0 TO 9:BLOCK#4,128,1,2,2+i*12,248
1108 END DEFINE
```



1110 **DEFINE PROCEDURE FontChar(cx,cy,ci)**

Highlight Char Font

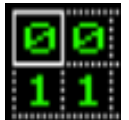
```
1111 BLOCK#3,18,1,1+cx*18,1+cy*12,si:BLOCK#3,18,1,1+cx*18,13+cy*12,si
1112 BLOCK#3,1,12,1+cx*18,1+cy*12,si:BLOCK#3,1,12,19+cx*18,1+cy*12,si
1113 END DEFINE
```



1115 **DEFINE PROCEDURE FontBit(x,y,ci)**

Highlight Grid Bit

```
1116 BLOCK#4,16,1,-14+x*16,-10+y*12,ci:BLOCK#4,16,1,-14+x*16,2+y*12,ci
1117 BLOCK#4,1,12,-14+x*16,-10+y*12,ci:BLOCK#4,1,12,2+x*16,-10+y*12,ci
1118 END DEFINE
```



1120 **DEFINE PROCEDURE FontPeek**

Display Char Bitmap

```
1121 IF cn<128:addr=FBase1+2+(cn-31)*9:ELSE addr=FBase2+2+(cn-127)*9
1122 OVER#4,1
1123 FOR r=0 TO 8
1124   Fnt$(r+1)=BIN$(PEEK(addr+r),8):CURSOR#4,4,4+r*12:PRINT#4,Fnt$(r+1)
1125 END FOR r
1126 OVER#4,0
1127 END DEFINE
```



1129 **DEFINE PROCEDURE FontPoke**

Change Char Bitmap

```
1130 IF cn<128:addr=FBase1+2+(cn-31)*9:ELSE addr=FBase2+2+(cn-127)*9
1131 FOR r=0 TO 8:POKE(addr+r),BIN(Fnt$(r+1))
1132 FontSets cn1,cn2
1133 END DEFINE
```



1135 **DEFINE PROCEDURE Fontsize**

```
1136 FOR i=0 TO 3:CSIZE#3,i,0:CURSOR#3,86+(i*36)+i*3,7:PRINT#3,CHR$(cn)
1137 FOR i=0 TO 3:CSIZE#3,i,1:CURSOR#3,98+(i*40)+i,2 :PRINT#3,CHR$(cn)
1138 CURSOR#1,12,38:PRINT#1,'Show CSIZES'
```

1139 **END DEFINE**

```

1141 DEFine PROCEDURE FontFiles(act) Select Font File
1142 IF act=1:Act$='LBYTES':ELSE Act$='SBYTES'
1143 CURSOR#1,372,42:PRINT#1,'↔ Select ↑↓ ':BLOCK#1,12,3,430,46,7
1144 REPeat File_lp
1145 CURSOR#1,324,32:PRINT#1,Act$&' '&drv$(dn%)&File$&sf%&' Y/N'
1146 K=CODE(INKEY$(-1))
1147 SElect ON K
1148 = 32:IF sg%=1:sg%=2:File$='QLFont2_':ELSE sg%=1:File$='QLFont1_'
1149 =192:dn%=dn% -1:IF dn%<1:dn%=8
1150 =200:dn%=dn%+1:IF dn%>8:dn%=1
1151 =208:sf%=sf%+1 :IF sf%>9:sf%=0
1152 =216:sf%=sf% -1 :IF sf%<0:sf%=9
1153 =110,78:RETurn
1154 =121,89:EXIT File_lp
1155 END SElect
1156 END REPeat File_lp
1157 IF act=1:FCheck:FontLoad:ELSE FCheck:FontSave
1158 END DEFine

```

```

1160 DEFine PROCEDURE FCheck Font File Check
1161 BLOCK#1,160,20,320,32,0:CURSOR#1,320,32:PRINT#1,'Seaching...'
1162 PAUSE 20:DELETE drv$(dn%)&'FList'
1163 OPEN_NEW#99,drv$(dn%)&'FList':DIR#99,drv$(dn%):CLOSE#99
1164 OPEN_IN#99,drv$(dn%)&'FList'
1165 REPeat dir_lp
1166 IF EOF(#99):CLOSE#99:chk=0:EXIT dir_lp
1167 INPUT#99,Fchk$:IF Fchk$==File$&sf%:CLOSE#99:chk=1:EXIT dir_lp
1168 END REPeat dir_lp
1169 END DEFine

```

```

(L)oad (S)ave (R)eset (E)xit
Seaching...

```

```

1171 DEFine PROCEDURE FontLoad
1172 IF ck=0 OR eck=1
1173 CURSOR#1,320,32:PRINT#1,'File NOT Found...':PAUSE 50:eck=0:RETurn
1174 END IF
1175 BLOCK#1,160,20,320,32,0:CURSOR#1,320,32:PRINT#1,'Loading...'
1176 IF sg%=1:LBYTES drv$(dn%)&'QLFont1_ '&sf%,FBase1:cn1=96
1177 IF sg%=2:LBYTES drv$(dn%)&'QLFont2_ '&sf%,FBase2:cn2=64
1178 FontSets cn1,cn2 :FontGrid:cn=32
1179 END DEFine

```

```

(L)oad (S)ave (R)eset (E)xit
File NOT Found...

```

```

(L)oad (S)ave (R)eset (E)xit
Loading...

```

```

1181 DEFine PROCEDURE FontSave
1182 IF eck=1
1183 CURSOR#1,320,32:PRINT#1,'DEVICE ERROR...':PAUSE 50:eck=0:RETurn
1184 END IF
1185 IF ck=1
1186 CURSOR#1,320,32:PRINT#1,'Overwrite Y/N':PAUSE:IF KEYROW(5)<>64:RETurn
1187 END IF
1188 DELETE drv$(dn%)&File$&sf%
1189 BLOCK#1,160,20,320,32,0:CURSOR#1,320,32:PRINT#1,'Saving...'
1190 IF sg%=1:SBYTES drv$(dn%)&'QLFont1_ '&sf%,FBase1,875
1191 IF sg%=2:SBYTES drv$(dn%)&'QLFont2_ '&sf%,FBase2,587
1192 END DEFine

```

```

(L)oad (S)ave (R)eset (E)xit
DEVICE ERROR...

```

```

(L)oad (S)ave (R)eset (E)xit
Overwrite Y/N

```

```

(L)oad (S)ave (R)eset (E)xit
Saving...

```