

QBITS Font Editor Concept

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 QL Font sets into memory, display them to screen as a Chart. A selected character would then be shown in a Bitmap display with the option to change the bit pattern and save back to memory. After a number of Fonts had been edited in this way the whole Font Set could be saved as a Font file for use with other Programs.

QBITS QLFont Editor Display

The screen display had to be easy and intuitive to use. At start-up the Default QL Character Fonts are displayed to screen. These are accessed from the default storage device which needs to be set before running the program (see Program line 1003 Dev\$=). The Bitmap shows the bits set to 0's & 1's for the current highlighted character. Returning from the Bitmap the Character under review is displayed in its different **CSIZES** with the **KEYPad** showing the keys needed to be pressed.

QBITS QLFont Editor 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 (R)eset (E)xit

Access by pressing the bracketed Character. Information is displayed relative to the action requested. For Load and Save first select storage device with Up/Down cursor keys and Y/N.

For **LOAD** a search is made of available '_fnt' font files. A 'File Not found' will be returned if none are available. Use the Up/Down cursors to scroll through and make your choice then Y/N to load or abort. A program check is made to select the relevant Character Set Memory address, before the File is loaded and Fonts displayed to screen.

To **SAVE** use Up/Down cursors to select which Character Font set to Save. Included is a Line Editor to Rename the Font file. When ready to save a device check is made and if unavailable a 'DEVIC ERROR' is given. An 'Overwrite Y/N' is given if the file is detected as already existing. An answer Y or good to go 'Saving...' is displayed.

Press 'R' for **Reset** which prompt with 'Y/N', 'N' aborts and 'Y' reloads the default Fonts. Pressing 'E' for **Exit** 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.

QLFonts & CSIZES

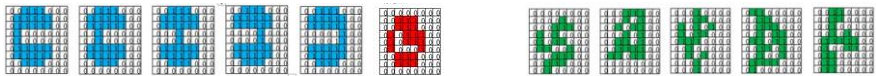
As mentioned, the Font Generator for the QL does not retrospectively produce all the bit pattern held by a character's Bitmap across the range of CSIZES.

As an example, the bit Pattern 'A' shown was Surrounded with a box. Not all CSIZES showed the whole of the box and some QL O/S had interesting outcomes.



Program use of Font Designs

I chose 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

'! # \$ % @ ' replaced for Rescue Pod



For Escape Pod '^'

(') * + _' for Ship Debray



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 by the O/S for the different CSIZES.

QBITS QLFont Editor Coding

1000 REMark **QBITS_FontEdit_v1** (QBITS Font Editor 1991 New for **BBQL** 2022)

1002 WMON:gx=0:gy=0 :REMark gx gy for High Res
1003 DIM drv\$(7,5):Dev\$='flp1_':dm%=7 :REMark Device Drives

1000 REMark **QBITS_FontEdit_v2** (QBITS Font Editor 1991 New for **QPC11** 2022)

1002 gx=40:gy=40 :REMark Set gx gy for High Res
1003 DIM drv\$(7,5):Dev\$='dos1_':dm%=7 :REMark Device Drives
1004 DIM Fnt\$(9,8),File\$(50,20):fm%=50 :REMark Font Arrays
1005 FBase1=ALCHP(875) :REMark Memory for QLFont1_fnt
1006 FBase2=ALCHP(587) :REMark Memory for QLFont2_fnt

1008 WHEN ERROr
1009 eck=1:CONTINUE
1010 END WHEN

1012 Init_Screen:FontMain

1014 DEFine PROCEDURE Init_Screen

1015 WINDOW#0,512,32,gx,gy+224 :PAPER#0,0 :CSIZE#0,0,0:BORDER#0,1,3:CLS#0
1016 WINDOW#1,512,224,gx,gy :PAPER#1,0 :CSIZE#1,0,0:BORDER#1,1,3:CLS#1
1017 WINDOW#2,512,224,gx,gy :PAPER#2,0 :CSIZE#2,0,0:BORDER#2,1,3:CLS#2
1018 OPEN#3,scl_ :WINDOW#3,276,178,gx+12,gy+32:CSIZE#3,3,0:INK#3,4
1019 OPEN#4,scl_ :WINDOW#4,136,112,gx+330,gy+88:CSIZE#4,3,0:INK#4,4
1020 CSIZE#2,2,1:OVER#2,1
1021 INK#2,2:FOR i=0 TO 1:CUSOR#2,6+i,8:PRINT#2,'QBITS QLFont Editor'
1022 INK#2,6:FOR i=0 TO 1:CUSOR#2,8+i,6:PRINT#2,'QBITS QLFont Editor'
1023 CSIZE#2,0,0:OVER#2,0:INK#1,7
1024 CUSOR#1,296,18:PRINT#1,'(L)oad (S)ave (R)eset (E)xit'
1025 CUSOR#1,12,208:PRINT#1,'Select ←↑ ↓→':BLOCK#1,12,3,70,212,7
1026 **RESTORE 1028**:FOR d=0 TO 7:READ str\$:drv\$(d)=str\$
1027
1028 DATA 'mdv1_', 'mdv2_', 'flp1_', 'flp2_', 'win1_', 'win2_', 'dos1_', 'dos4_'
1029 END DEFine

1031 DEFine PROCEDURE FontReset(ck)

1032 IF ck=1:CUSOR#1,420,32:PRINT#1,'Y/N':PAUSE:IF KEYROW(5)<>64:RETurn
1033 LBYTES Dev\$&'QLFont1_fnt',FBase1:cn1=96:cn1\$='QLFont1_fnt'
1034 LBYTES Dev\$&'QLFont2_fnt',FBase2:cn2=64:cn2\$='QLFont2_fnt'
1035 FontSets cn1,cn2:FontName cn1\$,cn2\$:FontGrid:cn=32
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#1:RECHP FBase1:RECHP FBase2:LRUN Dev\$&'QBITSProgs_v3'
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%=6:cx=0:cy=2:x=1:y=1:FontReset 0:chk=0:eck=0 :REMark dn% default drive
1045 REPEAT Main_lp
1046 IF cn=128:cy=9:cx=0
1047 IF cy=8 AND cx>5:cx=5:cn=127
1048 IF cy=13 AND cx>3:cx=3:cn=191
1049 cn=32+cx+(cy-2)*15:BLOCK#1,220,24,280,32,0:INK 7
1050 IF cn>127:cn=cn-9
1051 CURSOR#1,312,68:PRINT#1,'Character Dec: ',cn,' Hex: ',HEX$(cn,8):'
1052 FontChar cx,cy,7:FontPeek
1053 K=CODE(INKEY$(-1)):BLOCK#3,260,22,0,0,0:BLOCK 140,10,140,208,0
1054 FontChar cx,cy,0:IF K<>32:CLS#4:FontGrid
1055 SELECT ON K
1056 =192:cx=cx-1:cn=cn-1 :IF cx<0 :cx=0
1057 =200:cx=cx+1:cn=cn+1 :IF cx>14:cx=14
1058 =208:cy=cy-1:cn=cn-15:IF cy<2 :cy=2
1059 =216:cy=cy+1:cn=cn+15:IF cy>13:cy=13
1060 = 32:FontMod:KeyPad :REMark Modify Char
1061 =108,76:FontLoad :REMark (L)oad
1062 =115,83:FontSave :REMark (S)ave
1063 =114,82:FontReset 1 :REMark (R)eset
1064 =101,69:FontExit :REMark (E)xit
1065 END SElect
1066 END REPEAT Main_lp
1067 END DEFine

```

Note: Switch from Main Character display to Bitmap display. To switch back press 'N' for no change or 'Y' to update Character. Once Changes to a Font set are complete the updated Fonts can be saved.

```

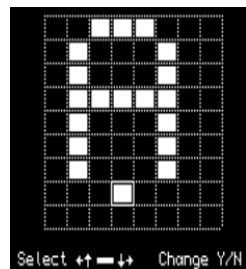
1069 DEFine PROCEDURE FontMod
1070 CURSOR#1,332,208:PRINT#1,'Select ←↑ ↓→ Change Y/N'
1071 BLOCK#1,12,3,368,212,7 :REMark Spacebar
1072 REPEAT Chg_lp
1073 FontBit x,y,7:K=CODE(INKEY$(-1)):FontBit x,y,248
1074 SELECT ON K
1075 =192:x=x-1:IF x<1:x=1
1076 =200:x=x+1:IF x>8:x=8
1077 =208:y=y-1:IF y<1:y=1
1078 =216:y=y+1:IF y>9:y=9
1079 =110,78:EXIT Chg_lp :REMark (N)o Return
1080 =121,89:FontPoke:EXIT Chg_lp :REMark (Y)es Change Font
1081 = 32:Bitswap :REMark Swap 0-1
1082 END SElect
1083 END REPEAT Chg_lp
1084 BLOCK#1,148,10,332,208,0:CLS#4:FontGrid:FontSize
1085 END DEFine

```

```

1087 DEFine PROCEDURE Bitswap
1088 IF Fnt$(y,x)='0':Fnt$(y,x)='1':ELSE Fnt$(y,x)='0'
1089 IF Fnt$(y,x)='0':BLOCK#4,13,9,4+(x-1)*16,4+(y-1)*12,0
1090 IF Fnt$(y,x)='1':BLOCK#4,13,9,4+(x-1)*16,4+(y-1)*12,7
1091 END DEFine

```



```

1093 DEFINE PROCEDURE FontSets(cn1,cn2)
1094 fx=2:fy=26:CHAR_USE#3,FBase1,FBase2:CLS#3:CSIZE#3,3,0:CLS#4:INK#1,7
1095 FOR c=32 TO 31+cn1
1096   CURSOR#3,fx,fy:PRINT#3,CHR$(c)
1097   fx=fx+18:IF fx>260:fx=2:fy=fy+12
1098 END FOR c
1099 fx=2:fy=110
1100 FOR c=128 TO 127+cn2
1101   CURSOR#3,fx,fy:PRINT#3,CHR$(c)
1102   fx=fx+18:IF fx>260:fx=2:fy=fy+12
1103 END FOR c
1104 END DEFINE

```



```

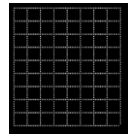
1106 DEFINE PROCEDURE FontName(cn1$,cn2$)
1107 CURSOR#1,156,130:PRINT#1,FILL$(' ',20-LEN(cn1$))&cn1$
1108 CURSOR#1,156,190:PRINT#1,FILL$(' ',20-LEN(cn2$))&cn2$
1109 END DEFINE

```

```

1111 DEFINE PROCEDURE FontGrid
1112 FOR i=0 TO 8:BLOCK#4,1,108,2+i*16,2,248
1113 FOR i=0 TO 9:BLOCK#4,128,1,2,2+i*12,248
1114 END DEFINE

```



```

1116 DEFINE PROCEDURE FontChar(cx,cy,ci)
1117 BLOCK#3,17,1,1+cx*18,2+cy*12,ci:BLOCK#3,17,1, 1+cx*18,12+cy*12,ci
1118 BLOCK#3,1,11,1+cx*18,2+cy*12,ci:BLOCK#3,1,11,18+cx*18, 2+cy*12,ci
1119 END DEFINE

```



```

1121 DEFINE PROCEDURE FontBit(x,y,ci)
1122 BLOCK#4,16,1,-14+x*16,-10+y*12,ci:BLOCK#4,16,1,-14+x*16,2+y*12,ci
1123 BLOCK#4,1,12,-14+x*16,-10+y*12,ci:BLOCK#4,1,12,2+x*16,-10+y*12,ci
1124 END DEFINE

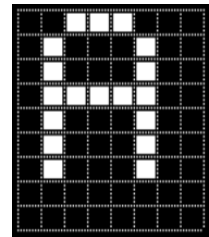
```



```

1126 DEFINE PROCEDURE FontPeek
1127 a1%=PEEK(FBase1):a2%=PEEK(FBase2)
1128 IF cn<128:addr=FBase1+2+(cn-a1%)*9:ELSE addr=FBase2+2+(cn-a2%)*9
1129 OVER#4,1
1130 FOR r=0 TO 8
1131   Fnt$(r+1)=BIN$(PEEK(addr+r),8)
1132   FOR c=0 TO 7:IF Fnt$(r+1,c+1)<>'0':BLOCK#4,13,9,4+c*16,4+r*12,7
1133 END FOR r
1134 OVER#4,0
1135 END DEFINE

```



```

1137 DEFINE PROCEDURE FontPoke
1138 a1%=PEEK(FBase1):a2%=PEEK(FBase2)
1139 IF cn<128:addr=FBase1+2+(cn-a1%)*9:ELSE addr=FBase2+2+(cn-a2%)*9
1140 FOR r=0 TO 8:POKE(addr+r),BIN(Fnt$(r+1))
1141 FontSets cn1,cn2:FontName cn1$,cn2$
1142 END DEFINE

```

Note a1% & a2% checks for lowest Char code



1144 **DEFine PROCEDURE FontSize**

1145 **CURSOR**#1,148,40:PRINT#1,'CSIZES':RESTORE 1147

1146 **FOR** i=1 **TO** 8:READ a,b,c,d:CSIZE#3,a,b:C**CURSOR**#3,c,d:PRINT#3,CHR\$(cn)

1147 **DATA** 0,0,2,4, 0,1,14,0, 1,0,28,4, 1,1,40,0

1148 **DATA** 2,0,56,4, 2,1,72,0, 3,0,94,4, 3,1,114,0

1149 **END DEFine**

```
(L)oad (S)ave (R)eset (E)xit
LBYTES dos1_Giro_fnt
Select ↑ 11 ↓ Y/N
```

1151 **DEFine PROCEDURE FontLoad**

1152 chk=0:eck=0:Lchk=0:sf%=0:ft%=0:fm%=50:**FOR** i=1 **TO** 50:File\$(i)=""

1153 **INK** 7:C**CURSOR** 300,46:PRINT 'Select ↑ ↓ Y/N':**INK** 5:**SelDrv**

1154 **FntList** 1:**SelFont** 1,'LBYTES ':**IF** Lchk=1:**GrpChk**:**ELSE** **RETURN**

1155 **BLOCK** 200,24,292,32,0:C**CURSOR** 310,32:PRINT#1,'Loading...':**PAUSE** 30

1156 **IF** sg%<127:LBYTES drv\$(dn%)&File\$(sf%),FBase1:cn= 32:cn1=96:cn1\$=File\$(sf%)

1157 **IF** sg%>=127:LBYTES drv\$(dn%)&File\$(sf%),FBase2:cn=128:cn2=64:cn2\$=File\$(sf%)

1158 **FontSets** cn1,cn2:FontName cn1\$,cn2\$:FontGrid:eck=0:**INK** 7

1159 **END DEFine**

```
(L)oad (S)ave (R)eset (E)xit
SBYTES dos1_QLFont1_fnt
Select ↑ 01 ↓ Y/N (E)dit ← →
```

1161 **DEFine PROCEDURE FontSave**

1162 chk=0:eck=0:Lchk=0:sf%=1:ft%=2:File\$(1)=cn1\$:File\$(2)=cn2\$

1163 **INK** 7:C**CURSOR** 300,46:PRINT 'Select ↑ ↓ Y/N (E)dit ← →'

1164 **BLOCK** 12,3,453,50,7:**BLOCK** 2,4,481,48,7:**INK** 5:**SelDrv**

1165 **SelFont** 2,'SBYTES ':**IF** Lchk=0:**RETURN**

1166 **FntList** 2:**BLOCK** 200,10,300,46,0:C**CURSOR** 312,46

1167 **IF** eck=1:PRINT#1,'DEVICE ERROR...':**PAUSE** 50:**RETURN**

```
(L)oad (S)ave (R)eset (E)xit
SBYTES flp2_QLFont1_fnt
DEVICE ERROR...
```

1168 **IF** chk=1:PRINT#1,'Overwrite Y/N' :**PAUSE**:**IF** KEYROW(5)<>64:**RETURN**

```
(L)oad (S)ave (R)eset (E)xit
SBYTES dos1_QLFont1_fnt
Overwrite Y/N
```

1169 **INK** 5:**DELETE** drv\$(dn%)&File\$(sf%)

1170 **BLOCK** 200,24,292,32,0:C**CURSOR** 310,32:PRINT#1,'Saving...':**PAUSE** 30

1171 **IF** sf%=1:SBYTES drv\$(dn%)&File\$(1),FBase1,875:cn1\$=File\$(1)

1172 **IF** sf%=2:SBYTES drv\$(dn%)&File\$(2),FBase2,587:cn2\$=File\$(2)

1173 **FontName** cn1\$,cn2\$:**INK** 7

1174 **END DEFine**

```
(L)oad (S)ave (R)eset (E)xit
dos1_
Select ↑ ↓ Y/N
```

1176 **DEFine PROCEDURE SelDrv**

1177 **REPEAT** drv_ip

1178 **CURSOR** 342,32:PRINT drv\$(dn%)

1179 K=CODE(INKEY\$(-1))

1180 **SELECT ON K**

1181 =208:dn%=dn%-1:**IF** dn%<0:dn%=dm%

1182 =216:dn%=dn%+1:**IF** dn%>dm%:dn%=0

1183 =121,99,110,78:**EXIT** drv_ip

1184 **END SELECT**

1185 **END REPEAT** drv_ip

1186 **END DEFine**

Note For Load SelfFont returns 'No Files Found' if FrontList fails to find any on selected device. If there are any '_fnt' files found Scroll Up/Down with cursor keys and select one of choice to load. For Save select one of the two font files you can save with exiting name to a new device or if modified to the present one by use (E)dit line editor to rename the file. If device not available DEVICE ERROR will be displayed. If file already exist you are prompted to Overwrite Y/N.

```

1188 DEFine PROCEDURE SelfFont(act%,Act$)
1189 sf%=1:BLOCK 200,10,296,32,0
1190 IF ft%<1:CURSOR 310,32:PRINT ' No Files Found...':PAUSE 30:RETurn
1191 REPeat File_lp
1192 CURSOR 292,32:PRINT Act$&drv$(dn%)&File$(sf%):CLS 4
1193 CURSOR 363,46:PRINT FILL$(0',2-LEN(sf%))&sf%
1194 K=CODE(INKEY$(-1))
1195 SElect ON K
1196 =208:sf%=sf%-1:IF sf%<1:sf%=ft%
1197 =216:sf%=sf%+1:IF sf%>ft%:sf%=1
1198 =101,69:IF act%=2:EditName File$(sf%)
1199 =110,78:Lchk=0:EXIT File_lp
1200 =121,99:Lchk=1:EXIT File_lp
1201 END SElect
1202 END REPeat File_lp
1203 IF File$(sf%)=QLFont1_fnt'OR File$(sf%)=QLFont2_fny':Lchk=0
1204 END DEFine

```

```

(L)oad (S)ave (R)eset (E)xit
No Files Found...
Select ↑ ↓ Y/N

```

```

'No
'Yes

```

Note: LINE 1203 checks that the default font files are not Overwritten!

```

1206 DEFine PROCEDURE FntList(act%)
1207 CURSOR 310,32:PRINT 'Seaching...':CLS 4:PAUSE 20
1208 DELETE drv$(dn%)&'FList'
1209 OPEN_NEW#9,drv$(dn%)&'FList':DIR#9,drv$(dn%):CLOSE#9
1210 OPEN_IN#9, drv$(dn%)&'FList'
1211 REPeat dir_lp
1212 IF act%=1
1213 IF EOF(#9) OR sf%>fm%:ft%=sf%-1:CLOSE#9:EXIT dir_lp
1214 INPUT#9,F$:fl%=LEN(F$)
1215 IF fl%<=20 AND '_fnt' INSTR F$>0:File$(sf%)=F$:sf%=sf%+1
1216 END IF
1217 IF act%=2
1218 IF EOF(#9):CLOSE#9:chk=0:EXIT dir_lp
1219 INPUT#9,F$:IF F$==File$(sf%):CLOSE#9:chk=1:EXIT dir_lp
1220 END IF
1221 END REPeat dir_lp
1222 END DEFine

```

```

(L)oad (S)ave (R)eset (E)xit
Seaching...
Select ↑ ↓ Y/N

```

Note: FntList carries out a search for '_fnt' files for LOAD and checks for SAVE if the Font File already exists before writing to storage. Prompting with Overwrite Y/N if necessary.

```

1224 DEFine PROCEDURE GrpChk
1225 OPEN_IN#9,drv$(dn%)&File$(sf%):sg%=CODE(INKEY$(#9)):CLOSE#9
1226 END DEFine

```

Note: Before Loading to FBase1 or FBase2 memory address GrpChk check Lowest Character held in first byte of file. If less than 127 file LBYTES to address FBase1 if greater to FBase2 memory address.

Note The Filename Editor restricts characters to those used for filenames. Position the underline with left/Right cursors to a character then Add a new or Delete the existing character. Adding a Character expands the string to the right, Deleting shrinks the string. To Delete the QL uses CTRL Right cursor for character above underline to make thing easier I have utilised the spacebar as a Delete key. CTRL Left Backspace still applies to Delete character to left of underline.

```

1228 DEFine PROCEDURE EditName(str$)
1229 sm%=16:sl%=LEN(str$):cp%=1
1230 temp$=str$:sl%=(' _fnt' INSTR str$)-1:str$=str$(1 TO sl%)
1231 REPeat Ed_Ip
1232 Ln_Prn:Ln_Cur:k$=INKEY$(#0,-1):K=CODE(k$)
1233 SELEct ON K
1234   = 10:EXIT Ed_Ip
1235   = 48 TO 57, 65 TO 90,95, 97 TO 122:Add_chr
1236   =194 :IF cp%>1:cp%=cp%-1:Del_chr
1237   =202,32:Del_chr
1238   =192 :IF cp%>1:cp%=cp%-1
1239   =200 :IF cp%<sl%+1:cp%=cp%+1
1240 END SELEct
1241 END REPeat Ed_Ip
1242 IF sl%=0:str$=temp$:ELSE str$=str$&'_fnt'
1243 END DEFine

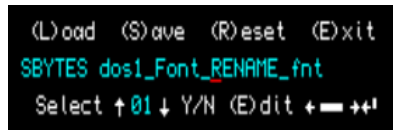
```

Delete Left
Delete

```

1245 DEFine PROCEDURE Ln_Prn
1246 IF LEN(str$)>sm%:str$=str$(1 TO sm%):cp%=sm%
1247 INK 5:CURSOR 364,32:PRINT str$&'_fnt':CLS 4
1248 END DEFine

```



```

(L)oad (S)ave (R)eset (E)xit
SBYTES dos1_Font_RENAME_fnt
Select + 01 + Y/N (E)dit + - + +

```

```

1250 DEFine PROCEDURE Ln_Cur
1251 BLOCK 6,1,364+cp%*6-6,41,2
1252 END DEFine

```

```

1254 DEFine PROCEDURE Add_chr
1255 IF cp% = 1 AND sl%=0 :str$=str$&k$
1256 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&k$&str$(cp% TO sl%)
1257 IF cp%>=1 AND cp%=sl%:str$=str$(1 TO cp%-1)&k$&str$(cp%)
1258 IF cp%> 1 AND cp%>sl%:str$=str$&k$
1259 IF cp%=sm%:str$(cp%)=k$
1260 IF sl%<sm%:sl%=sl%+1:ELSE sl%=sm%
1261 IF cp%<sm%:cp%=cp%+1:ELSE cp%=sm%
1262 END DEFine

```

```

1264 DEFine PROCEDURE Del_chr
1265 IF cp%=sl%:str$=str$(1 TO sl%-1):sl%=sl%-1
1266 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&str$(cp%+1 TO sl%):sl%=sl%-1
1267 IF cp%=sm%:str$=str$(1 TO sm%-1):cp%=cp%-1:sl%=sm%-1
1268 IF cp%=1 AND sl%=1:str$="":sl%=0
1269 END DEFine

```

Note A Font Editor would not seem complete if the KEY(s) for Printable Characters were not identified.

1271 DEFine PROCEDURE KeyPad

```
1272 SElect ON cn=32 TO 96 :Key$=CHR$(cn)
1273 SElect ON cn=32 :Key$='Spacebar'
1274 SElect ON cn=33,34 :Key$='Shift '&(cn-32)
1275 SElect ON cn=36,37 :Key$='Shift '&(cn-33)
1276 SElect ON cn=38 :Key$='Shift 7'
1277 SElect ON cn=40 :Key$='Shift 9'
1278 SElect ON cn=41 :Key$='Shift 0'
1279 SElect ON cn=42 :Key$='Shift 8'
1280 SElect ON cn=43 :Key$='Shift ='
1281 SElect ON cn=58 :Key$='Shift ,'
1282 SElect ON cn=60 :Key$='Shift ,'
1283 SElect ON cn=62 :Key$='Shift .'
1284 SElect ON cn=63 :Key$='Shift /'
1285 SElect ON cn=64 :Key$='Shift '"
1286 SElect ON cn=65 TO 90 :Key$='Shift '&CHR$(cn)
1287 SElect ON cn=94 :Key$='Shift 6"
1288 SElect ON cn=95 :Key$='Shift -"
1289 SElect ON cn=96 :Key$='Shift 3"
1290 SElect ON cn=97 TO 122 :Key$=CHR$(cn-32)
1291 SElect ON cn=123 :Key$='Shift ['
1292 SElect ON cn=124 :Key$='Shift \'
1293 SElect ON cn=125 :Key$='Shift ]'
1294 SElect ON cn=126 :Key$='Shift #'
1295 SElect ON cn=127 :Key$='Shift Esc'
1296 SElect ON cn=128 :Key$='CTRL Esc'
1297 SElect ON cn=129 :Key$='CTRL Shift 1'
1298 SElect ON cn=130 :Key$='CTRL Shift '"
1299 SElect ON cn=131 :Key$='CTRL Shift 3'
1300 SElect ON cn=132 TO 153 :Key$='CTRL '&CHR$(cn-96)
1301 SElect ON cn=154 :Key$='CTRL Shift ,'
1302 SElect ON cn=155 :Key$='CTRL ;'
1303 SElect ON cn=156 :Key$='CTRL Shift ,'
1304 SElect ON cn=157 :Key$='CTRL ='
1305 SElect ON cn=158 :Key$='CTRL Shift .'
1306 SElect ON cn=159 :Key$='CTRL Shift \'
1307 SElect ON cn=160 :Key$='CTRL Shift 2'
1308 SElect ON cn=161 TO 186 :Key$='CTRL Shift '&CHR$(cn-96)
1309 SElect ON cn=187 :Key$='CTRL ['
1310 SElect ON cn=188 :Key$='CTRL ]'
1311 SElect ON cn=189 :Key$='CTRL \'
1312 SElect ON cn=190 :Key$='CTRL Shift 6'
1313 SElect ON cn=191 :Key$='CTRL Shift -'
1314 :
1315 CURSOR 148,208:PRINT 'KEYPad: '&Key$
1316 END DEFine
```

KEYPad: Shift A

QBITS - Storage Devices

The default Storage Device 'Dev\$' at Code Line 1003 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%) [Dev\$].

QBITS - Font Installation

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

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_fnt** & **QLFont2_fnt** 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.