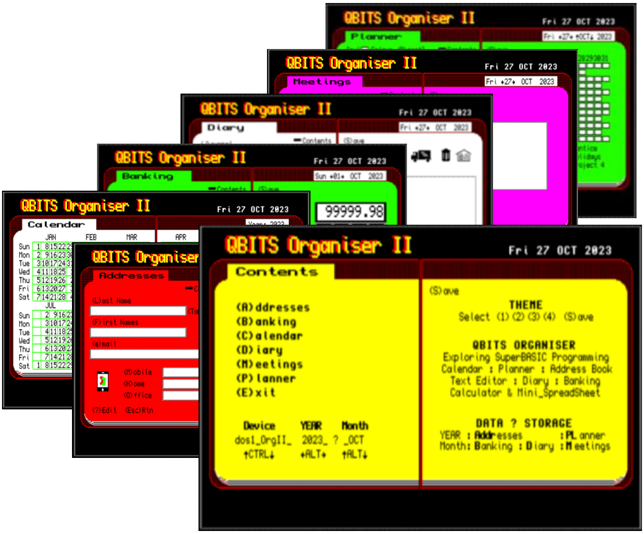


QBITS Introduction

Personal Organisers might have been around as early as 1910. A UK company founded in 1921 began producing a range of Personal Organisers styled in leather bound wallets. This design originated with a US company 'Lefax' and is perhaps one explanation as to how the abbreviation of a File of Facts became the FiloFax.

What is a Personal Organiser all about, one early description was a box or open tray kept on an office desk and divided into compartments to hold office supplies? Of more interest was a small book in which information such as names, addresses, phone numbers, dates of meetings, etc helped organise your time? Revamped by Ian Logan in the 1980's it became the most desirable accessory for the so-called Yuppie brigade.

As an electronic device a Personal Organiser contains sections for a Diary, Calendar, Contact list, a Year Planner. Back in the nineteen eighties I was enthused with writing one in SuperBASIC, but dropped any further work when PISION with its QL connection, marketed a range of Electronic Organisers called Personal Digital Assistants (PDA's).

QBITS Organiser Beginnings

The beginnings in the eighties started with code for a double page spread for Year Calendar and soon followed with Planner. A Contacts list page was next utilising a simple Line Editor. Then a Meetings scheduler was added. Graphics for a Calculator led to working with Math functions. For larger entries such as a Diary page the Line Editor was extended to work within a TextBox of varying columns and rows.

QBITS Organiser Storage

Eighties assessments revealed a year's storage of DATA might be in the order of 300Kbytes. It prompted dividing the DATA storage into monthly SAVE/LOAD files, so as to reduce working RAM allocation and for storage across multiple Microdrives.

QBITS Organiser II - Code Review

A new Contents Page provides a Menu access to each of the Organiser Sections. Apart from Calendar each can create information that is held temporally in Arrays. The Saving of data is made direct to default storage Device. Organiser II when entering a Section, automatically Searches and Loads any previously Saved DATA from the current default Device. On leaving a Section the program prompts a Save of any DATA generated.

Each Organiser Section Saves DATA in different Formats. The default Device and range of Devices is loaded at Program Start from QBITSConfig. Following the Device name is a Prefix for a Directory Filename held in DF$. The QBITS Program default is **’OrgII_**’, this can be changed to the users preferred file heading (see Code Line 1019).

All Sections use the following suffix that of the ‘**Year_**’. To this is then added the Type, ‘**Addr**’ Addresses, ‘**PL** Planner. ‘**B_**’ Banking, ‘**D_**’ Diary, ‘**M_**’ Meetings. Appended to Banking, Diary & Meetings, is the Month ‘**JAN**’ etc. The default Device, Year & Month can be changed on the Contents Page.

On the righthand side of Selected Pages the working Date is displayed. This is separate from the Current date shown above the Organiser Page. For Banking, Diary, Meetings Left/Right Cursor keys changes any displayed information to comply with the Day of the current working Date. In addition, for the Planner to change the Month use Up/Down Cursors. Calendar shows only the Year. Addresses instead of date shows the displayed record number, to change use Left/Right Cursors. To change the Search character use Up/Down cursors.

QBITS Calendar Calculations

As aligning the Day of the Month to the Day of the Week proved problematic, the weekday of the Gregorian Calendar and Key Value Method was used. To work out the correct weekday, codes for different Months and Years speed up the calculation. It takes the last two digits of the year divides by 4 and drops the remainder, to which it then adds the day of the month. A month key value is then added to the result.

Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
1	4	4	0	2	5	0	3	6	1	4	6

A leap year is one divisible by 4, but not if evenly divisible by 100 i.e. 1700,1800,1900, only centuries divisible by 400 are leap years. If Jan or Feb is in a leap year subtract 1, then add the century code.

1700s	1800s	1900s	2000s
4	2	0	6

To obtain codes for other centuries luckily the Gregorian Calendar repeats every 400 years. For example, 1650+400=2050 so code is 6.

Next, we again add the last two digits of the year, divide by 7 and use the remainder. The value being the day of the week, 1-Sunday, 2-Monday and so on with 0 Saturday.

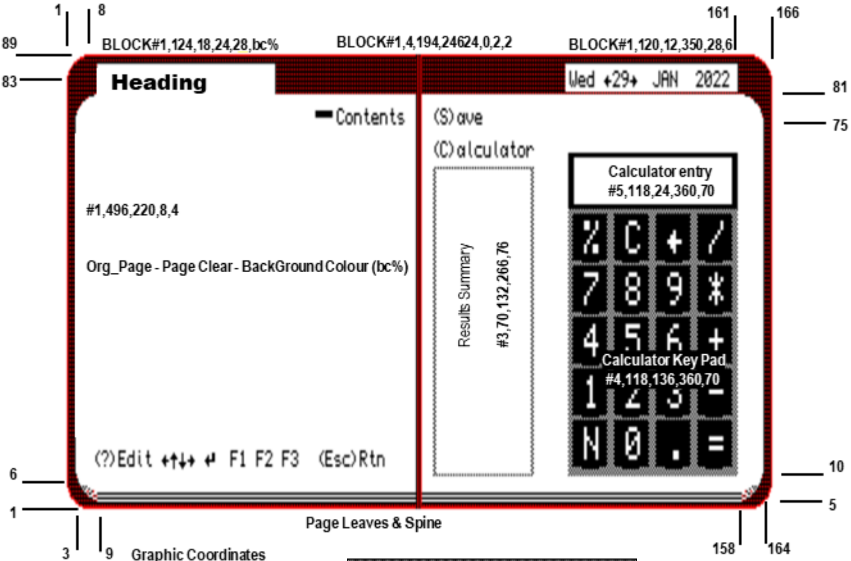
See the PROCedure Cal_day (yr%,m%,d%).

QBITS Organiser II Code 2022

```
1000 REMark QBITS_OrgII_bas (QBITS Organiser – 2023 Review QPCII)

1002 dev$='dos1_':MODE 4:gx=0:gy=0    :REMark Basic Settings

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

1006 REMark  Import QBITSConfig Settings
1007 OPEN_IN#9,dev$&'QBITSConfig':INPUT#9,gx\gy\dn$\dev$\dn%\dm%
1008 DIM drv$(15,5):FOR d=0 TO 15:INPUT#9,drv$(d):END FOR d:CLOSE#9

1010 REMark *** Appointments/Banking/Calendar/Diary/Mail List/Planner ***
1011 DIM phone$(50,6,16),name$(50,5,26),addr$(50,96),email$(50,36)    :REMark Addr
1012 DIM Tran$(31,9,3,16),Key$(4,5),Num$(9):num=0                     :REMark Banking
1013 DIM Jrnl$(31,432),Rmdr$(31,216),Wth%(31,5),CRem(31,6)            :REMark Diary
1014 DIM Meet$(31,8,32),Note$(31,256)                                 :REMark Meets
1015 DIM Plnr%(12,31),Event$(12,12),Pcol%(12)                         :REMark Planner
1016 DIM bcol%(6),wd$(7,3),mth$(12,3),md%(12),Sky$(8,10),RStr$(15)

1018 REMark *** Init Prog TexED/Calculator/OrgII_Data File & OrgII_Config ***
1019 yb%=1901:ye%=2099:DF$='OrgII_'                      :REMark Year Range : Data File Prefix
1020 ct%=65:bc%=7:rm%=50                                 :REMark ct%=A : bc% BkGnd :Max Rcrds
1021                     :FOR i=0 TO  6:bcol%(i)=7       :REMark BkGnd Colours
1022 RESTORE 1029:FOR i=1 TO  7:READ wd$(i)              :REMark Days of the Week
1023 RESTORE 1030:FOR i=1 TO 12:READ mth$(i),md%(i)      :REMark Month / Days-Month
1024 RESTORE 1033:FOR i=0 TO  8:READ Sky$(i)             :REMark Weather Patterns
1025 RESTORE 1036:FOR i=1 TO 12:READ Event$(i),Pcol%(i)  :REMark Planner
1026 RESTORE 1040:FOR b=1 TO  5:FOR a=1 TO  4:READ Key$(a,b):END FOR a:END FOR b

1028 REMark *** Days of the week : Month : End of Month ***
1029 DATA "Sun","Mon",'Tue','Wed','Thu','Fri','Sat'
1030 DATA 'JAN',31,'FEB',28,'MAR',31,'APR',30,'MAY',31,'JUN',30
1031 DATA 'JUL',31,'AUG',31,'SEP',30,'OCT',31,'NOV',30,'DEC',31
1032 REMark *** Weather Patterns ***
1033 DATA 'Changeable','Sunny','Cloudy','Windy','Showers'
1034 DATA 'Overcast','Heavy Rain','Stormy','Snow'
1035 REMark *** Planner Events : Colour Codes ***
1036 DATA 'Anniversary',1,'Birthday',2,'Doctor',3,'Dentice',4
1037 DATA 'Shopping',5,'Meetings',6,'PayDays',41,'Holidays',48
1038 DATA 'Project 1',34,'Project 2',80,'Project 3',230,'Project 4',255
1039 REMark *** Calculator Keyboard ***
1040 DATA '%','C','¼','/', '7','8','9','*', '4','5','6','+'
1041 DATA '1','2','3','-', 'N','0','.','='

1043 MODE 4:Init_Win:Init_Date:Org_Menu

1045 DEFine PROCedure Org_Exit
1046 STRIP bc%:INK 0:CURSOR 92,132:PRINT 'Y/N':PAUSE
1047 IF KEYROW(5)=64:LRUN dn$:ELSE BLOCK 20,10,92,144,bc%
1048 END DEFine
```

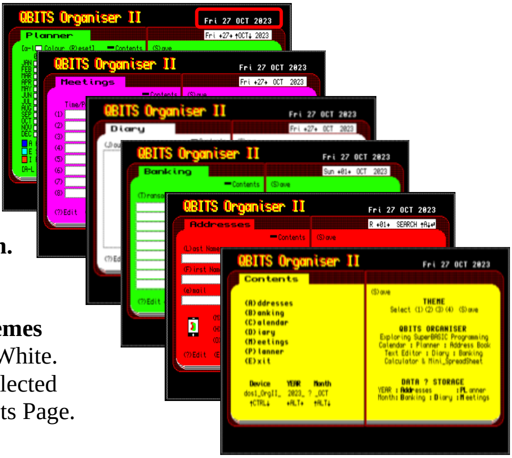
QBITS Organiser II - Screen Layouts

Using QL SuperBASIC WINDOW’s, channel#6 is opened to deploy as a blank screen background. WINDOW #1 the main operating channel is set with SCALE 100,0,0. Using ARC & LINE and the Graphics coordinates system, the first job was deciding x, y positions for drawing the organiser outer Cover and Pages. Next location for a Page Heading and areas for displaying the Organisers Current and Working Dates.



QBITS Date

Current Date is set by QL Clock.



QBITS Working Dates

For **Banking Diary & Meetings**
Change the Day of the Month.
For Planner change **Day & Month.**

QBITS Organiser Themes

Default Page colour is White.
Page Themes can be Selected
and Saved from Contents Page.

QBITS Organiser Controls

Cursor Left/Right/Up/Down <Spacebar> <Enter> <Esc>
← → ↑ ↓ ▬ ↵ **(Esc)**

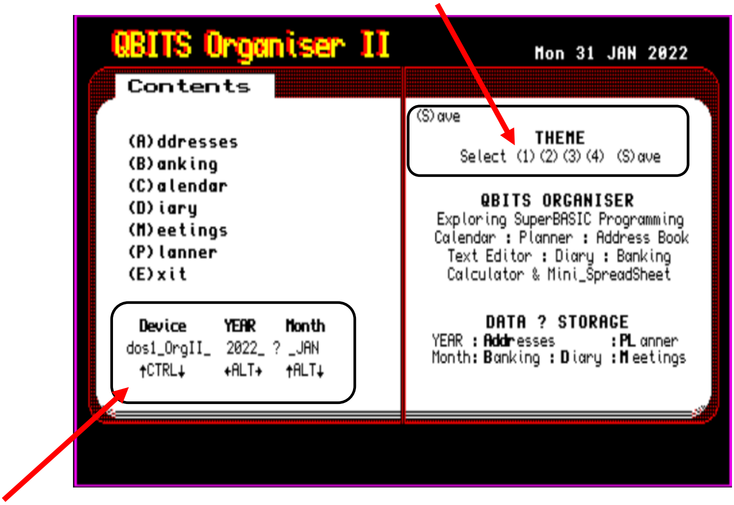
Edit (?) [key First Letter of (**H**)eading…] **(Esc)** to Return

```
1050 DEFine PROCedure Init_Win
1051 LOCal a,b,c,d,e,f
1052 OPEN#6,scr_:WINDOW#6,512,256,gx,gy:BORDER#6,1,3:PAPER#6,0:CLS#6
1053 OPEN#5,scr_:WINDOW#5,118, 24,360+gx,70+gy              :REMark QBCalc Entry
1054 OPEN#4,scr_:WINDOW#4,118,136,360+gx,72+gy              :REMark QBCalc Keys
1055 OPEN#3,scr_:WINDOW#3,70,132,266+gx,76+gy               :REMark QBCalc Results
1056 WINDOW#2,496,220,gx+8,gy+4  :PAPER#2,0:INK#2,7         :REMark SBASIC EDitor
1057 WINDOW#1,496,220,gx+8,gy+4  :PAPER#2,0:INK#2,7         :REMark QBOrg Page
1058 WINDOW#0,496, 20,gx+8,gy+226:PAPER#2,0:INK#2,7         :REMark QL Command
1059 ch%=1:SCALE#ch%,100,0,0:chk=0
1060 FOR chk=0 TO 1
1061  IF chk=0:FILL#ch%,1:INK#ch%,16:RESTORE 1065
1062  IF chk=1:FILL#ch%,0:INK#ch%,  2:RESTORE 1065
1063  FOR i=1 TO 4:READ a,b,c,d,e,f:LINE#ch%,a,b TO c,d:ARC#ch% TO e,f,-PI/2
1064 END FOR chk
1065 DATA 6,89,161,89,166,83,166,83,166,6,161,1
1066 DATA 161,1,6,1,1,6,1,6,1,83,6,89
1067 REMark Page Leaves
1068 RESTORE 1070:BLOCK#ch%,446,5,24,210,7,0,1:INK#ch%,7
1069 FOR i=1 TO 6:READ a,b,c,d:ARC#ch%,a,b TO c,d,PI/2
1070 DATA 3,10,9,2.1,3,10,9,3,3,10,9,4,158,2.1,164,10,158,3,164,10,158,4,164,10
1071 REMark Title
1072 ch%=1:QBIT$='QBITS Organiser':OVER#ch%,1:CSIZE#ch%,2,1
1073 INK#ch%,2:FOR i=0 TO 1:CURSOR#ch%,20-i,2:PRINT#ch%,QBIT$
1074 INK#ch%,6:FOR i=0 TO 1:CURSOR#ch%,22+i,3:PRINT#ch%,QBIT$
1075 OVER#ch%,0:CSIZE#ch%,0,0
1076 END DEFine

1078 DEFine PROCedure Init_Date                    Current Date from QL Clock
1079 ch%=1:BLOCK#ch%,124,10,342,10,0
1080 d$=DAY$(DATE):dmy$=DATE$:yr%=dmy$(1 TO 4):d%=dmy$(10 TO 11)
1081 FOR i=1 TO 12:IF dmy$(6 TO 8)==mth$(i):m%=i:EXIT i
1082 OVER#ch%,1:CSIZE#ch%,1,0:INK#ch%,7
1083 FOR i=0 TO 1:CURSOR#ch%,350+i,12:PRINT#ch%,d$;' ';d%;' ';mth$(m%);' ';yr%
1084 OVER#ch%,0:CSIZE#ch%,0,0:INK#ch%,0
1085 END DEFine

1100 REMark Organiser Theme

1102 DEFine PROCedure Org_Theme                    Page BkGnd Colours
1103 RESTORE 1104+(k-49):FOR i=0 TO 6:READ bcol%(i)
1104 DATA 7,7,7,7,7,7,7            :REMark Default BkGnd bc% = White
1105 DATA 6,2,4,7,6,3,4            :REMark Yellow, Red, Green, White, Yellow, Magenta, Green
1106 DATA 5,3,6,7,5,4,5            :REMark Cyan, Magenta, Yellow, Cyan, Green, Cyan
1107 DATA 4,6,5,7,4,5,6            :REMark Green, Yellow, Cyan, White, Green, Cyan, Yellow
1108 END DEFine

1110 DEFine PROCedure Org_Page(ch%,bc%)            Page Clear
1111 RESTORE 1114:FILL#ch%,1:INK#ch%,bc%
1112 FOR i=1 TO 4:READ a,b,c,d,e,f:LINE#ch%,a,b TO c,d:ARC#ch% TO e,f,-PI/1.5
1113 FILL#ch%,0:STRIP#ch%,bc%:INK#ch%,0
1114 DATA 10,81,156,81,164,75,164,75,164,10,158,5,158,5,10,5,3,10,3,10,3,75,10,81
1115 BLOCK#ch%,124,18,24,28,bc% :BLOCK#ch%,120,12,350,28,16
1116 BLOCK#ch%,4,194,246,24,0,2,2
1117 END DEFine
```

QBITS Organiser II - Theme

On start-up **Init_Date** takes the Current Date from the QL Clock and sets the **Year** & **Month**. Initialising, the Organiser Loads form an Org Config File to access the saved colour Theme or displays ‘File NOT Found’ with Page Backgrounds displayed White.

Select (1) (2) (3) (4) and (**S**)ave **Theme**. The Config File details the background colours to be deployed. See Code Line 1104 DATA 7,7,7,7,7,7,7 etc.



```
QBITS Organiser II                          Mon 31 JAN 2022

Contents                              (S)ave
                                                THEME
(A)ddresses                            Select (1) (2) (3) (4)  (S)ave
(B)anking
(C)alendar                                 QBITS ORGANISER
(D)iary                              Exploring SuperBASIC Programming
(M)eetings                           Calendar : Planner : Address Book
(P)lanner                              Text Editor : Diary : Banking
(E)xit                                 Calculator & Mini_SpreadSheet

 Device       YEAR      Month                DATA ? STORAGE
dos1_OrgII_  2022_ ?  _JAN            YEAR : Addresses        :PLanner
 ↑CTRL↓      ←ALT→    ↑ALT↓           Month: Banking : Diary :Meetings
```

QBITS Organiser II - Settings

CTRL UP/Down cursor keys to change the Default **Device.** To check try doing a **Theme** Save. If Device not valid a DEVICE ERROR will be given.

ALT Left/Right cursor keys to change **Year**. All Data files are affected by this, make sure regular backups are carried out.

ALT Up/Down cursor keys to change **Month**. Banking, Diary, and Meetings are affected by this, make sure regular backup are carried out.

QBITS Organiser II - DATA Files

The Organiser Year/Month DATA Files are Searched for when entering the relevant Organiser Page. If found they are Loaded. When leaving an Organiser Page, or by pressing ‘**S**’ a (**S**)ave Y/N prompt is displayed. If answer is ‘Yy’ (Yes) the DATA File will be Saved or a prompt to Overwrite Y/N (if already exists) or ‘DEVIC ERROR’ (if Device is not accessible).

QBITS Organiser II - DATA File Format

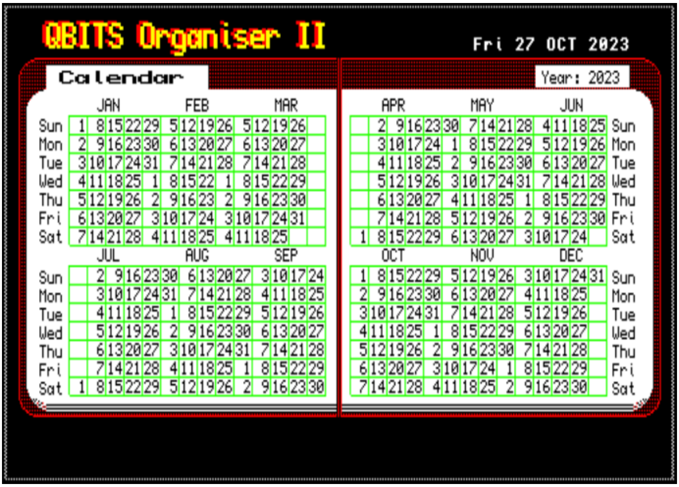
Example: win1_OrgII_2022_Addr   Address List File
Example. dos1_OrgII_2022_D_JAN   Diary Page File

```
1119 DEFine PROCedure Org_Menu
1120 ch%=1:Org_Page 1,7:Prn_Bold 1,256,60,'Initialising...'
1121 BLOCK 120,12,350,28,16:OLoad 0,'OrgConfig':Con=1
1122 REPeat Page_lp
1123   IF Con=1:Con=0:bc%=bcol%(0):Org_Page 1,bc%:Content_Page:FileCLS
1124   CURSOR#ch%,32,172:PRINT#ch%,drv$(dn%)&'Org_ '&yr%&'_  ?  _'&mth$(m%)
1125  k=CODE(INKEY$(-1))
1126   SELect ON k
1127     =65,  97:Con=1:bc%=bcol%(1):Org_Page 1,bc%:Addresses
1128     =66,  98:Con=1:bc%=bcol%(2):Org_Page 1,bc%:Banking
1129     =67,  99:Con=1:bc%=bcol%(3):Org_Page 1,bc%:Calendar
1130     =68,100:Con=1:bc%=bcol%(4):Org_Page 1,bc%:Diary
1131     =77,109:Con=1:bc%=bcol%(5):Org_Page 1,bc%:Meetings
1132     =80,112:Con=1:bc%=bcol%(6):Org_Page 1,bc%:Planner
1133     =69,101:Org_Exit
1134     =210:dn%=dn%+1:IF dn%>dm%:dn%=dm%            :REMark Ctrl Up  Device
1135     =218:dn%=dn% -1:IF dn%<0  :dn%=0             :REMark Ctrl Down
1136     =193:yr% =yr%  -1:IF yr%<yb%:yr%=yb%         :REMark Alt Left   Year
1137     =201:yr% =yr% +1:IF yr%>ye%:yr%=ye%          :REMark Alt Right Year
1138     =209:m% =m%  -1:IF m%< 1:m%= 1               :REMark Alt Up     Month
1139     =217:m% =m% +1:IF m%>12:m%=12                :REMark Alt Down Month
1140     =49,50,51,52  :Con=1:Org_Theme               :REMark Select Theme
1141     =83,115  :OSave 0,'OrgConfig':FileCLS
1142   END SELect
1143 END REPeat Page_lp
1144 END DEFine

1146 DEFine PROCedure Prn_Bold(ps,px,py,str$)          Overwrite in various Font sizes
1147 CSIZE ps,0:OVER 1
1148 FOR b=0 TO 1:CURSOR px+b,py:PRINT str$
1149 CSIZE  0,0:OVER 0
1150 END DEFine

1152 DEFine PROCedure Content_Page
1153 RESTORE 1154FOR i=1 TO 8:READ px,py,str$:CURSOR px,py:PRINT str$
1154 DATA 280, 68,'Select (1)(2)(3)(4) (S)ave'
1155 DATA 274,106,'Exploring SuperBASIC Programming'
1156 DATA 272,116,'Calendar:Year Planner:Text Editor'`
1157 DATA 278,126,'Address Book : Diary : Banking'
1158 DATA 282,136,'Mini_SpreadSheet & Calculator'
1159 DATA  38,184,'↑ CTRL↓    ←ALT→    ↑ALT↓ '
1160 DATA 270,168,'YEAR  :     esses        :   anner'
1161 DATA 270,177,'Month :  anking :  iary :  eetings'
1162:FOR i=1 TO 12:READ ps,px,py,str$:Prn_Bold ps,px,py,str$
1163 DATA 2,32,30,'Contents',1,32,60,'(A)ddresses',1,32,72,'(B)anking'
1164 DATA 1,32,84,'(C)alendar',1,32,96,'(D)iary',1,32,108,'(M)eetings'
1165 DATA 1,32,120,'(P)lanner',1,32,132,'(E)xit',1,350,58,'THEME'
1166 DATA 0,38,160,'Device   YEAR Type Month'
1167 DATA 1,308,96,'QBITS ORGANISER',1,322,158,'DATA STORAGE'
1168 DATA 0,310,168,’Addr’,0,417,168,’PL’
1169 DATA 0,310,177,’B’,0,370,177,’D’,0417,177,’M’
1170 END DEFine
```

QBITS Organiser Calendar
Set the **Year** on **Contents** Page before accessing Calendar Page. Press any key to Return.

The Calendar Page is divided into four quarterly sections with the Months and Days of a 13-week period. These are spread over 14 columns to cater for different start and end weekdays, the **Days** of each **week** calculated relevant to the **YEAR** selected.

```
1200 REMark Organiser Calendar Page    Select Year ↑↓Displays WkDays/Months

1202 DEFine PROCedure Calendar
1203 ch%=1:Prn_Bold 2,32,30,'Calendar'
1204 BLOCK#ch%,72,12,398,28,7:Calendar_Page:Cal_Year yr%
1205 STRIP#ch%,7:CURSOR#ch%,402,30:PRINT#ch%,'Year: ';yr%:PAUSE
1206 END DEFine
```



QBITS Organiser II

Fri 27 OCT 2023

Calendar

Year: 2023

		JAN					FEB					MAR		
Sun	1	8	15	22	29	5	12	19	26	5	12	19	26	
Mon	2	9	16	23	30	6	13	20	27	6	13	20	27	
Tue	3	10	17	24	31	7	14	21	28	7	14	21	28	
Wed	4	11	18	25	1	8	15	22	1	8	15	22	29	
Thu	5	12	19	26	2	9	16	23	2	9	16	23	30	
Fri	6	13	20	27	3	10	17	24	3	10	17	24	31	
Sat	7	14	21	28	4	11	18	25	4	11	18	25		

		JUL					AUG					SEP		
Sun		2	9	16	23	30	6	13	20	27	3	10	17	24
Mon		3	10	17	24	31	7	14	21	28	4	11	18	25
Tue		4	11	18	25	1	8	15	22	29	5	12	19	26
Wed		5	12	19	26	2	9	16	23	30	6	13	20	27
Thu		6	13	20	27	3	10	17	24	31	7	14	21	28
Fri		7	14	21	28	4	11	18	25	1	8	15	22	29
Sat	1	8	15	22	29	5	12	19	26	2	9	16	23	30

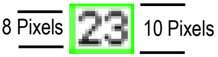
		APR					MAY				JUN			
	2	9	16	23	30	7	14	21	28	4	11	18	25	Sun
	3	10	17	24	1	8	15	22	29	5	12	19	26	Mon
	4	11	18	25	2	9	16	23	30	6	13	20	27	Tue
	5	12	19	26	3	10	17	24	31	7	14	21	28	Wed
	6	13	20	27	4	11	18	25	1	8	15	22	29	Thu
	7	14	21	28	5	12	19	26	2	9	16	23	30	Fri
1	8	15	22	29	6	13	20	27	3	10	17	24		Sat

		OCT					NOV				DEC			
1	8	15	22	29	5	12	19	26	3	10	17	24	31	Sun
2	9	16	23	30	6	13	20	27	4	11	18	25		Mon
3	10	17	24	31	7	14	21	28	5	12	19	26		Tue
4	11	18	25	1	8	15	22	29	6	13	20	27		Wed
5	12	19	26	2	9	16	23	30	7	14	21	28		Thu
6	13	20	27	3	10	17	24	1	8	15	22	29		Fri
7	14	21	28	4	11	18	25	2	9	16	23	30		Sat

```
1208 DEFine PROCedure Cal_Year(yr%)
1209 ch%=1:d%=1:wk%=1:mn%=1:x%=28:y%=46:cw%=1                    :REMark 1st Quarter
1210 Cal_Day yr%,1,1:IF yr% MOD 100=0:dm%(2)=28                   :REMark 1st Jan :Weekday wkd%
1211 CSIZE#ch%,0,0:INK#ch%,0:OVER#ch%,1:IF yr% MOD 400=0:dm%(2)=29
1212 REPeat day_lp
1213   IF cw%=1:cw%=0:FOR day=1 TO 7:BLOCK#ch%,13,9,x%+wk%*14,y%+day*10,7
1214   CURSOR#ch%,x%+wk%*14,y%+wd%*10:PRINT#ch%,FILL$(' ',2-LEN(d%))&d%
1215   wd%=wd%+1:d%=d%+1:IF wd%>7:wd%=1:wk%=wk%+1:cw%=1  :REMark cw% Wk CLS
1216   IF wk%=13 AND d%=31:FOR day=1 TO 7:BLOCK#ch%,13,9,x%+14*14,y%+day*10,7
1217   IF d%>dm%(mn%):d%=1:mn%=mn%+1                               :REMark days of month
1218   IF mn%= 4 AND d%=1:wk%=15:cw%=1:x%=  47:y%= 46              :REMark 2nd Quarter
1219   IF mn%= 7 AND d%=1:wk%=29:cw%=1:x%=-364:y%=127              :REMark 3rd Quarter
1220   IF mn%=10 AND d%=1:wk%=43:cw%=1:x%=-345:y%=127              :REMark 4th Quarter
1221   IF mn%>12:OVER#ch%,0:EXIT day_lp                            :REMark Exit
1222 END REPeat day_lp
1223 END DEFine
```

The Calendar Cells are drawn using BLOCK and Pixel coordinates system. The chosen Window size 496x220 reduces the row height for Cell Characters to 8 Pixels. To overcome this restriction the Cell is Cleared with a BLOCK command and PRINT is used with the OVER command to write the Days of the Month.

Font 6x10 Pixels



Cal_Year determines Cell BLOCK and CURSOR positioning then PRINTs the Date. The day [**d%**] and weekday [**wd%]** increments on each pass. If [**wd%**] reaches **7** it is reset to **1**, the week [**wk%**] is then incremented and clear week [**cw%**] is reset. The day [**d%**] is checked against the month [**dm%(m%)**]. If last day reached this sets the month day [**d%**] back to 1 and increments the month [**m%]**.

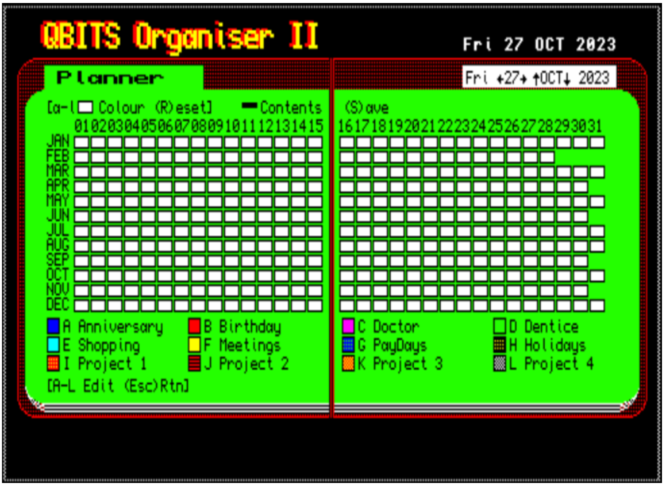
Quarters 2nd, 3rd & 4th checks sets x%,y% coordinates for repositioned Cell groups. In certain years, the occasion arises when Jan 1st is on a Sunday and March 31st falls in week 13 for this an additional Clear of Cells in week 14 is made.

C**al_day** (Year, Month, Day) calculates which Weekday a specified Date falls on.

```
1225 DEFine PROCedure Cal_Day(yr%,m%,d%)
1226 IF yr% MOD 4=0:dm%(2)=29:ELSE dm%(2)=28                :REMark Leap yr% check
1227 y$=yr%:yd%=y$(3 TO 4):year=yr%                          :REMark yr% & yd% values
1228 SELect ON year
1229   =1700 TO 1799:yc%=4                                   :REMark yc% Century key value
1230   =1800 TO 1899:yc%=2
1231   =1900 TO 1999:yc%=0
1232   =2000 TO 2099:yc%=6
1233 END SELect
1234 RESTORE 1235:FOR i=1 TO 12:READ a :IF m%=i:mk%=a:EXIT i
1235 DATA 1,4,4,0,2,5,0,3,6,1,4,6                            :REMark mk% Month key value
1236 IF dm%(2)=29:IF m%=1 OR m%=2:mk%=mk%-1:END IF          :REMark Leap Year adjustment
1237 wd%=((yd% DIV 4)+d%+mk%+yc%+yd%) MOD 7:IF wd%=0:wd%=7:REMark wd% Week Day
1238 END DEFine

1240 DEFine PROCedure Calendar_Page                         Draws the Quarterly Grids
1241 FOR i=1 TO  8:BLOCK#ch%,196,1,42,45+i*10,4
1242 FOR i=1 TO 15:BLOCK#ch%,1,70,27+i*14,55,4
1243 FOR i=1 TO  7:CURSOR#ch%,18,46+i*10   :PRINT#ch%,wd$(i)
1244 FOR i=1 TO  3:CURSOR#ch%,i*68-6,45    :PRINT#ch%,mth$(i)
1245 FOR i=1 TO  8:BLOCK#ch%,196,1,257,45+i*10,4
1246 FOR i=1 TO 15:BLOCK#ch%,1,70,242+i*14,55,4
1247 FOR i=1 TO  7:CURSOR#ch%,456,46+i*10  :PRINT#ch%,wd$(i)
1248 FOR i=1 TO  3:CURSOR#ch%,i*68+212,45  :PRINT#ch%,mth$(i+3)
1249 FOR i=1 TO  8:BLOCK#ch%,196,1,42,126+i*10,4
1250 FOR i=1 TO 15:BLOCK#ch%,1,70,27+i*14,136,4
1251 FOR i=1 TO  7:CURSOR#ch%,18,128+i*10  :PRINT#ch%,wd$(i)
1252 FOR i=1 TO  3:CURSOR#ch%,i*68-6,126   :PRINT#ch%,mth$(i+6)
1253 FOR i=1 TO  8:BLOCK#ch%,196,1,257,126+i*10,4
1254 FOR i=1 TO 15:BLOCK#ch%,1,70,242+i*14,136,4
1255 FOR i=1 TO  7:CURSOR#ch%,456,128+i*10:PRINT#ch%,wd$(i)
1256 FOR i=1 TO  3:CURSOR#ch%,i*68+212,126:PRINT#ch%,mth$(i+9)
1257 END DEFine
```

QBITS Organiser Planner

Select Year on Contents Page. Entering Planner, the Year is set out as cells in Month rows and Day columns. Each is identified can be colour coded. Having Selected a specific Date (Day/Month), chose a colour. As the original QL had a limited number of solid colours, the Stipple patterning method was used to extend the range. The Event Titles can be Edited and are Saved as part of the Planner DATA File.



QBITS Organiser II

Fri 27 OCT 2023

Planner

Fri ←27→ ↑OCT↓ 2023

[a-l□ Colour (R)eset] ▬Contents (S)ave

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
JAN	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
FEB	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□			
MAR	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
APR	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
MAY	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
JUN	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
JUL	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
AUG	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
SEP	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
OCT	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
NOV	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	
DEC	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

🟦 A Anniversary	🟥 B Birthday	🟪 C Doctor	□ D Dentice
🟦 E Shopping	🟨 F Meetings	▨ G PayDays	▨ H Holidays
▨ I Project 1	▨ J Project 2	▨ K Project 3	▨ L Project 4

[A-L Edit (Esc)Rtn]

```
1300 REMark Organiser Planner Page       Events Chart – Days of Year Colour Coded

1302 DEFine PROCedure Planner
1303 ch%=1: Prn_Bold 2,32,30,'Planner': OLoad 6,'Org_'&yr%&'_PL':FileCLS
1304 BLOCK#ch%,120,12,350,28,7:Planner_Page:Cal_Day yr%,m%,d%:Chart_Ent
1305 REPeat Plan_lp
1306  Chg_Date:k=CODE(INKEY$(-1))
1307  SELect ON k
1308   =32:OSave 6,'Org_'&yr%&'_PL':EXIT Plan_lp          :REMark Spacebar
1309   =83,115:OSave 6,'Org_'&yr%&'_PL':FileCLS           :REM ark (S)ave
1310   =97 TO 108:Plnr%(m%,d%)=k-96      :Chg_Col          :REMark (a - l) Colour Code
1311   =82,114      :Plnr%(m%,d%)=0      :Chg_Col          :REMark (R)eset Colour Code
1312   =65 TO  76 :chk=2                 :Chg_Event k-64 :REMark (A - L) Change Event Name
1313   =192:d% =d% -1:IF d%<1:d%=1                         :REMark Change Day
1314   =200:d% =d%+1:IF d%>dm%(m%):d%=dm%(m%)
1315   =208:m%=m%-1:IF m%<1:m%=1                          :REMark Change Month
1316   =216:m%=m%+1:IF m%>12:m%=12
1317  END SELect
1318 END REPeat Plan_lp
1319 END DEFine

1321 DEFine PROCedure Chg_Date
1322 Cal_Day yr%,m%,d% :STRIP#ch%,7
1323 CURSOR#ch%,352,30:PRINT#ch%,wd$(wd%);'← ';FILL$('0',2-LEN(d%))&d%;'→ '
1324 CURSOR#ch%,404,30:PRINT#ch%,'↑ ';mth$(m%);'↓ ';yr%
1325 END DEFine
```

Note: Select Day/Month with cursors then Set with one of the Event colours [a-l]

```
1327 DEFine PROCedure Chg_Col
1328 IF d%<16:pn=34:ELSE pn=46
1329 IF Plnr%(m%,d%)=0:ec%=7:ELSE ec%=Pcol%(Plnr%(m%,d%))
1330 BLOCK#ch%,8,7,53,47,ec%:BLOCK#ch%,10,5,pn+1+d%*13,59+m%*8,ec%
1331 END DEFine
```

Use [**Shift A-L**] and access QBITS **Text Editor to change any of the Twelve Event Titles**

```
1333 DEFine PROCedure Chg_Event(ev)
1334 SELect ON ev=1,5, 9 :ex%= 50:ey%=164+10*(ev DIV 4)
1335 SELect ON ev=2,6,10 :ex%=160:ey%=164+10*(ev DIV 4)
1336 SELect ON ev=3,7,11 :ex%=280:ey%=164+10*(ev DIV 4)
1337 SELect ON ev=4,8,12 :ex%=398:ey%=154+10*(ev DIV 4)
1338 bc%=bcol%(6):TextEd 1,0,12,1,ex%,ey%,Event$(ev)
1339 END DEFine
```

```
1341 DEFine PROCedure Chart_Ent                    Writes the Year Chart Entries
1342 FOR mth=1 TO 12
1343   FOR day=1 TO dm%(mth)
1344     IF day<16:pn%=34:ELSE pn%=46
1345     IF Plnr%(mth,day)=0:pc%=7:ELSE pc%=Pcol%(Plnr%(mth,day))
1346     BLOCK#ch%,12,7,pn%+day*13,58+mth*8,0
1347     BLOCK#ch%,10,5,pn%+1+day*13,59+mth*8,pc%
1348   END FOR day
1349 END FOR mth
1350 IF dm%(2)=28:BLOCK#ch%,12,7,423,74,bcol%(6)
1351 END DEFine
```

```
1353 DEFine PROCedure Planner_Page                 Creates the double page display
1354 FOR mth=1 TO 12
1355   FOR i=0 TO 1:CURSOR#ch%,24+i,56+mth*8:PRINT#ch%,mth$(mth)
1356 END FOR mth
1357 FOR n=1 TO 31
1358   IF n<10:N$='0'&n:ELSE N$=n
1359   IF n<16:pn%=34:ELSE pn%=44
1360   CURSOR#ch%,pn%+n*13,56:PRINT#ch%,N$
1361 END FOR n
1362 BLOCK#ch%,12,3,178,48,0:INK#ch%,0
1363 CURSOR#ch%,192,46:PRINT#ch%,'Contents'
1364 CURSOR#ch%,24, 46:PRINT#ch%,'[a-l   Colour (R)eset]'
1365 CURSOR#ch%,24,196:PRINT#ch%,'[A-L Edit (Esc)Rtn]'
1366 BLOCK#ch%,12,7,50,47,0:Chg_Col:RESTORE 1372:ev%=1
1367 FOR i=0 TO 11
1368    READ x:y=164+10*(i DIV 4)
1369    BLOCK#ch%,10,9,x,y,0:BLOCK#ch%,8,7,x+1,y+1,Pcol%(ev%)
1370    CURSOR#ch%,x+12,y:PRINT#ch%,CHR$(ev%+64);' ';Event$(ev%):ev%=ev%+1
1371 END FOR i
1372 DATA 26,136,256,374,26,136,256,374,26,136,256,374
1373 END DEFine
```

QBITS Organiser Addresses

Select Year and/or Month as required on the **Contents** Page. Entering Addresses Page will LOAD DATA file if previously Saved and display any Records. Select and Edit entries as required. When leaving Page (S)ave Newly entered or updated Records.

```
1400 REMark Organiser Addresses Names/email/Mobile/Home/Office/Addresses

1402 DEFine PROCedure Addresses
1403 ch%=1:Prn_Bold 2,32,30,'Addresses':OLoad 1,'Org_'&yr%&'_Addr':FileCLS
1404 BLOCK#ch%,120,12,350,28,7:rn%=1:rs%=1:rm%=50:rf%=1:Address_Page:bc%=7
1405 REPeat Mail_lp
1406 Address_Ent rn%:k=CODE(INKEY$(-1))
1407  SELect ON k
1408   =192:rn%=rn%-1:IF rn%< 1:rn%=50                :REMark rn% Rcd Numm
1409   =200:rn%=rn%+1:IF rn%>50:rn%=1
1410   =208:IF rs%> 1:rs%=rs%-1                       :REMark rs% Rcd Search
1411   =216:IF rs%<26:rs%=rs%+1
1412   =    9:rf%=rf%+1:IF rf%>2:rf%=1:END IF :STRIP#ch%,bcol%(1):Set_Tab
1413   =  10:rc%=1:Find_Rcd rn%
1414   = 32:bc%=bcol%(1):OSave 1,'Org_'&yr%&'_Addr':EXIT Mail_lp
1415   =83,115:bc%=bcol%(1):OSave 1,'Org_'&yr%&'_Addr':FileCLS    :REMark (S)ave
1416   =76,108:TextEd 1,2,25,1, 20, 72,name$(rn%,1)                :REMark (L)ast Name
1417   =70,102:TextEd 1,0,25,1, 20, 98,name$(rn%,2)                :REMark (F)irst Name
1418   =69,101:TextEd 1,4,36,1, 20,124,email$(rn%)                 :REMark (e)mail
1419   =65, 97:TextEd 1,0,32,3,260, 98,addr$(rn%)                  :REMark (A)ddress
1420   =77,109:TextEd 1,1,15,1,146,146,mail$(rn%,1)                :REMark (M)obile
1421   =70,104:TextEd 1,1,15,1,146,160,mail$(rn%,2)                :REMark (H)ome
1422   =79,111:TextEd 1,1,15,1,146,174,mail$(rn%,3)                :REMark (O)ffice
1423   =80,112:TextEd 1,3,15,1,260,146,mail$(rn%,4)                :REMark (P)ost/Zip Code
1424   =68,100:TextEd 1,3,15,1,260,160,mail$(rn%,5)                :REMark (D)istrict
1425   =67,  99:TextEd 1,2,15,1,260,174,mail$(rn%,6)               :REMark (C)ountry
1426  END SELect
1427 END REPeat Mail_lp
1428 END DEFine
```

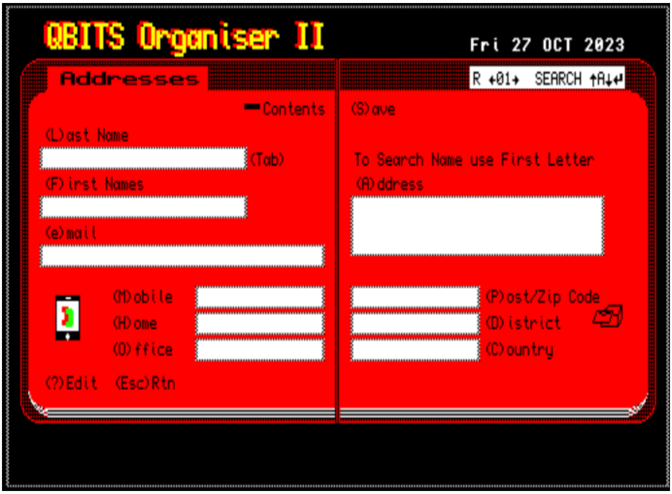
Last Name or **First Name** use **Tab** to switch between.

```
1430 DEFine PROCedure Set_Tab
1431 BLOCK#ch%,70,40,172,70,bcol%(1):CURSOR#ch%,172,47+26*rf%:PRINT#ch%,'(Tab)'
1432 END DEFine
```

Find based on the first character of the **Last Name** or **First Name**

```
1434 DEFine PROCedure Find_Rcd(rn%)
1435 REPeat rd_lp
1436   STRIP#ch%,bcol%(1):rn%=rn%+1:IF rn%>rm%:rn%=1          Note: rn% record number :rm% max
1437   IF RS$ INSTR name$(rn%,rf%,1):RETurn :ELSE rc%=rc%+1     Note: rc% records checked
1438   IF rc%>50:CURSOR#ch%,172,47+26*rf%:PRINT#ch%,' Not Found!':EXIT rd_lp
1439 END REPeat rd_lp
1440 PAUSE 50:Set_Tab
1441 END DEFine
```

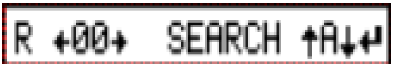
```
1443 DEFine PROCedure Address_Ent(rn%)                  Prints Field Entries to screen
1444 RESTORE 1451:STRIP#ch%,7
1445 RS$=CHR$(rs%+64):CURSOR#ch%,448,30:PRINT#ch%,RS$
1446 CURSOR#ch%,370,30:PRINT#ch%,FILL$('0',2-LEN(rn%))&rn%
1447 sc%=26:sr%=1:sx%=20:FOR i=0 TO 1:sy%=72+i*26:Prn_Str name$(rn%,i+1)
1448 sc%=36:sr%=1:sx%=20:sy%=124                   :Prn_Str email$(rn%)
1449 sc%=32:sr%=3:sx%=260:sy%=98                   :Prn_Str addr$(rn%)
1450 sc%=16:sr%=1:FOR i=1 TO 6:READ sx%,sy%        :Prn_Str phone$(rn%,i)
1451 DATA 140,146,140,160,140,174,260,146,260,160,260,174
1452 END DEFine
```



```
1454 DEFine PROCedure Address_Page
1454 Phone 30,150:Post 150,24:RESTORE 1457
1456 FOR i=1 TO 10:READ a,b,x,y:BLOCK#ch%,a+4,b+2,x-2,y-1,248
1457 DATA 150,10,20,72,150,10,20,98,216,10,20,124          :REMark LName:FName:email
1458 DATA  90,10,260,146, 90,10,260,160,90,10,260,174     :REMark Zip Area Country
1459 DATA 90,10,146,146,90,10,146,160,90,10,146,174       :REMark Mobile:Home:Office
1460 DATA 192,30,260,98                                    :REMark Addrs
1461 FOR I=1 TO 13:READ a,b,M$ :CURSOR#ch%,a,b:PRINT#ch%,M$
1462 DATA 190,46,'Contents',256,46,'(S)ave',260,86,'(A)ddress'
1463 DATA 20,60,'(L)ast Name',20,86,'(F)irst Names',20,112,'(e)mail'
1464 DATA 72,146,'(M)obile',72,160,'(H)ome',72,174,'(O)ffice'
1465DATA 360,146,'(P)ost/Zip Code',360,160,'(D)istrict',360,174,'(C)ountry'
1466 DATA 20,192,’(?)Edit  (Esc)Rtn’,260,73,'To Search Name use First Letter'
1467 Set_Tab:STRIP#ch%,7:CURSOR#ch%,352,30:PRINT#ch%,'R ←   →   SEARCH ↑    ↓← '
1478 BLOCK#ch%,12,3,176,48,0:BLOCK#ch%,2,4,466,32,0
1469 END DEFine
```

QBITS Addresses Search

Use <Left/Right> Cursors to Select a Record or Search by name.



```
R ←00→  SEARCH ↑A↓↵
```

Select an Alphabet character using the <**Up/Down>** Cursor keys, then press <**Enter**>. If no match is made '**Not Found!**' will be returned. If there are more entries with the chosen Alphabet letter, Press <**Enter**> until the one looked for is found. Alternatively, use the <**Left/Right>** Cursor keys to display the previous or the next sequential record.

To Edit the respective field key in letter held in [**?**] brackets of selected entry. When finished Exit with [**Esc**]. Records can be entered sequentially or on an add hock bases not necessarily in alphabetical order.

QBITS Organiser Contact Lists

An essential component of any Organiser is no doubt its Contact List. The question was what information should the Address page contain and in what format. An obvious starting point was, **Names**, Last name, First name, a nickname or non de plume. Then consideration for a quick search of these names to find a related record. As **emails** are now part of life what conventions exist over the length and allowable characters such as the **@** symbol. For Phone numbers, Mobile, Home & Office seemed a sensible choice.

Addresses come with a Name for a Building be it Home or Office and/or a Number, Street or Road name, a local area; Village, Town or **District** of a City, then the **Country** and **for** National Postal Services the use of **Post/Zip** codes.

Convention for Data Fields Entries

Government guidelines suggested codes for Character fields, **A** Alphabetic, **N** Numeric and **S** Special for example the **@** symbol. For names and addresses, field lengths are typically 30 to 35 alphanumeric characters and in the main use uppercase letters.

For email addresses surprisingly 320 characters, the user's name limited to 64 plus the @ symbol and domain names up to 255. Special characters identified as Period (.), Underscore (_), Hyphen (-) and Plus sign (+). For the QBITS Organiser what had to be taken into account was what was practicable for a display in the number of fields and their size, yet still accommodate most people's requirements.

The **emails** field **for** QBITS Organiser is limited to 36 alphanumeric characters A-Z, a-z, 0-9, the Special characters being period (.) underscore (_) hyphen (-) and the @ symbol.

QBITS Organiser Address Fields

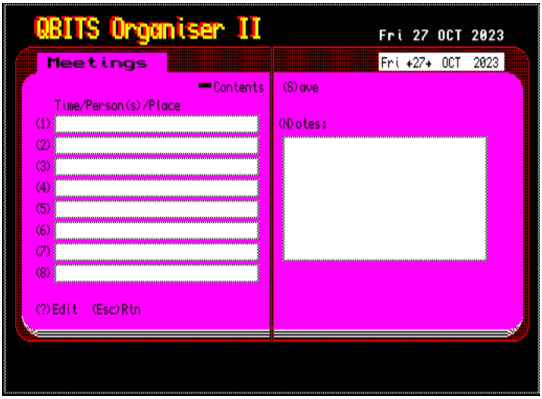
To cover most National Postal schemes, what was decided upon may not meet all local conventions. First a Text box with three rows for an **Address** using alphanumeric and punctuation characters, Numeric and Upper-Case Alphabet characters for the **Post/Zip** Code, **District** and **Country** entries.

QBITS Organiser Meetings

Select Year and/or Month as required on the **Contents** Page. Entering the Meetings Page will LOAD DATA file if previously save and display records. Select Day and Edit Meeting entries as required and Save when leaving.

```
1500 REMark Organiser Meetings  Appointments/Events/NotePad

1502 DEFine PROCedure Meetings
1503 ch%=1: Prn_Bold 2,32,30,'Meetings':OLoad 5,'Org_'&yr%&'_M_'&mth$(m%):FileCLS
1504 BLOCK#ch%,120,12,350,28,7:Meetings_Page:STRIP#ch%,7
1505 REPeat Appts_lp
1506   Chg_Day:Meetings_Ent:k=CODE(INKEY$(-1))
1507   SELect ON k
1508     =32    :OSave 5,'Org_'&yr%&'_M_'&mth$(m%):EXIT Appts_lp
1509     =83,115:OSave 5,'Org_'&yr%&'_M_'&mth$(m%):FileCLS            :REMark (S)ave
1510     =49 TO 56:n%=k-48:TextEd 1,0,32,1,42,56+n%*14,Appt$(d%,n%)  :REMark Number
1511     =78,110              :TextEd 1,0,32,8,260,84,Note$(d%)       :REMark (N)ote
1512     =192:d%=d% -1:IF d%<1:d%=1                                   :REMark ←
1513     =200:d%=d%+1:IF d%>dm%(m%):d%=dm%(m%)                        :REMark →
1514   END SELect
1515 END REPeat Appts_lp
1516 END DEFine
```

Note: To Edit Select (1)(2)(3)(4)(5)(6)(7)(8) (N) for Notes

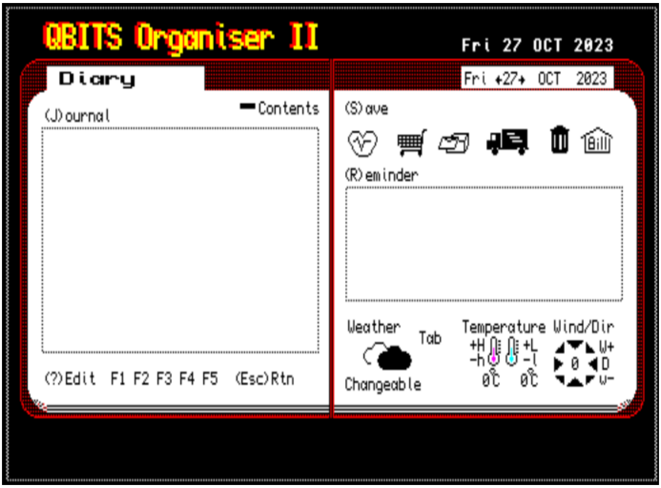


```
1518 DEFine PROCedure Chg_Day
1519 Cal_Day yr%,m%,d%:STRIP#ch%,7
1520 CURSOR#ch%,352,30:PRINT#ch%,wd$(wd%);' ← ';FILL$(0,2-LEN(d%))&d%;'→ '
1521 CURSOR#ch%,410,30:PRINT#ch%,mth$(m%);'  ';yr%
1522 END DEFine
```

```
1524 DEFine PROCedure Meetings_Ent
1525 sc%=32:sr%=1:sx%=42:FOR i=1 TO 8:sy%=56+i*14:Prn_Str Meet$(d%,i)
1526 sc%=32:sr%=8:sx%=260:sy%=84:Prn_Str Note$(d%)
1527 END DEFine

1529 DEFine PROCedure Meetings_Page
1530 CURSOR#ch%,192,46:PRINT#ch%,'Contents':BLOCK 12,3,178,48,0:FileCLS
1531 CURSOR#ch%, 40,58:PRINT#ch%,'Time/Person(s)/Place'
1532 FOR i=1 TO 8:CURSOR 20,56+i*14:PRINT '(';i;')'
1533 FOR i=1 TO 8:BLOCK 196,12,40,55+i*14,248
1534 CURSOR#ch%,252,70:PRINT#ch%,'(N)otes:':BLOCK#ch%,196,82,258,83,248
1535 CURSOR#ch%,20,192:PRINT#ch%,'(?)Edit  (Esc)Rtn'
1536 END DEFine
```

QBITS Organiser Diary

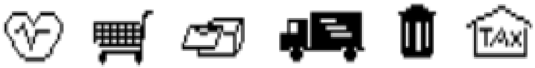
Select Year and Month as required on the Contents page. Entering Diary Page will LOAD DATA file if previously save and display records. The layout has a (**J**)ournal for reporting on the day's events and a (**R**)eminder for all those important tasks not to be forgotten. Select Day and Edit as required, (S)ave when leaving.



```
1550 REMark Organiser Diary Page      Daily Journal/Reminder/Weather/Events

1552 DEFine PROCedure Diary
1553 ch%=1:Prn_Bold 2,32,30,'Diary':OLoad 4,'Org_'&yr%&'_D_'&mth$(m%):FileCLS
1554 BLOCK#ch%,120,12,350,28,7:Diary_Page:h%=0:l%=0:w%=0:wp%=0
1555 REPeat Diary_lp
1556   Chg_Day:Diary_Ent:Wth_Set:k=CODE(INKEY$(-1))
1557   SELect ON k
1558     =32:      OSave 3,'Org_'&yr%&'_D_'&mth$(m%):EXIT Diary_lp
1559     =83,115:OSave 3,'Org_'&yr%&'_D_'&mth$(m%)
1560     =74,106:TextEd 1,0,36,12,21,62,Jrnl$(d%)                :REMark (J)ournal
1561     =82,114:TextEd 1,0,36,6,260,94,Rmdr$(d%)                :REMark (R)eminders
1562     =9:wp%=wp%+1:IF wp%>8:wp%=0:END IF    :Wth%(d%,1)=wp%:Wth_Set
1563     =  72:IF h%< 80:h%=h%+1:Wth%(d%,2)=h%  :Wth_Set  :REMark +(H)igh Temp
1564     =104:IF h%>-20:h%=h%-1:Wth%(d%,2)=h%   :Wth_Set  :REMark -(h)igh Temp
1565     =  76:IF l%< 40:l%=l%+1:Wth%(d%,3)=l%     :Wth_Set  :REMark +(L)ow Temp
1566     =108:IF l%>-40:l%=l%-1:Wth%(d%,3)=l%      :Wth_Set  :REMark -(l)ow Temp
1567     =  87:IF w%<200:w%=w%+1:Wth%(d%,4)=w% :Wth_Set  :REMark + Wind (F)actor
1568     =119:IF w%>  0:w%=w%-1:Wth%(d%,4)=w%  :Wth_Set  :REMark - Wind (f)actor
1569     = 68,100:wp%=wp%+1:IF wp%>8:wp%=0:END IF :Wth%(d%,5)=wp%:Wth_Set
1570     =192:d%=d% -1:IF d%<1:d%=1
1571     =200:d%=d%+1:IF d%>dm%(m%):d%=dm%(m%)
1572   END SELect
1573 END REPeat Diary_lp
1574 END DEFine
```

QBITS Diary Graphics

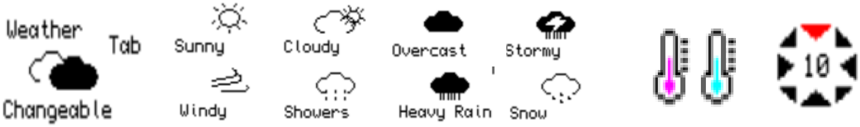
The icons were created as helpful reminders, the need of a Heath check or visit to the Gym, Shopping, Postage of Letters or Parcels, expecting a Delivery, putting out the Garbage or if a House or Car Tax payment is due.



Press '**R**' to edit the (**R**)eminder textbox.

```
1576 DEFine PROCedure Diary_Ent                        Display Field Entries
1577 sc%=36:sr%=12:sx%= 21:sy%=62:Str_Prn Jrnl$(d%)   :REMark Journal entry
1578 sc%=36:sr%= 8:sx%=260:sy%=94:Str_Prn Rmdr$(d%) :REMark Reminder entry
1579 END DEFine
```

The Weather Reporting uses Graphics to enhance the screen display and minimise the DATA storage. The Graphics are a mixture of **Vector** and **Bitmap** designs. Press the **Tab** key to scan through The Weather patterns range from Changeable, through Sunny, Cloudy, Windy, Showers, Overcast, Heavy Rain (Downpour), Stormy, to Snow.



Edit Temp and Wind speed using (+**H-h** /+**L-l** /+**W-w**), use '**Dd**' to highlight a Wind direction.

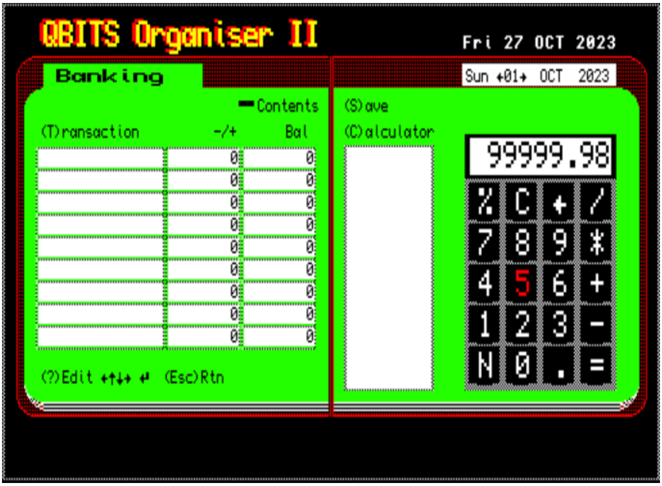
```
1581 DEFine PROCedure Wth_Set                          Weather Settings
1582 Weather Wth%(d%,1):WindDir Wth%(d%,4),Wth%(d%,5),148,14
1583 h%=Wth%(d%,2):CURSOR#ch%,354,193:PRINT#ch%,FILL$(' ',3-LEN(h%))&h%
1584 l% =Wth%(d%,3):CURSOR#ch%,384,193:PRINT#ch%,FILL$(' ',3-LEN(l%))&l%
1585 END DEFine

1447 DEFine PROCedure Diary_Page
1588 ch%=1:BLOCK#ch%,218,122, 20,61,248:BLOCK#ch%,218,62,259,93,248
1589 BLOCK#ch%,12,3,176,48,0:RESTORE 1591
1590 FOR i=1 TO 17:READ x%,y%,H$:CURSOR#ch%,x%,y%:PRINT#ch%,H$
1591 DATA 190,46,'Contents',256,46,'(S)ave',20,50,'(J)ournal', 256,82,'(R)eminder'
1592 DATA 260,164,'Weather',316,170,'Tab',350,164,'Temperature Wind/Dir'
1593 DATA 374,193,'C',356,174,'+H',356,182,'-h',458,175,'W+'
1594 DATA 404,193,'C',398,174,'+L',398,182,'-l',458,192,'w-'
1595 DATA 460,184,'D',20,192,'(?)Edit  F1 F2 F3 F4 F5  (Esc)Rtn'
1596 Therm 371,174,3:CIRCLE#ch%,126,12,.6:Therm 385,174,5:CIRCLE#ch%,136,12,.6
1597 Health 88,70:Shop 300,62:Post 112,66:Goods 370,62:Trash 420,60:CTax 154,70
1598 END DEFine
```

QBITS Organiser Banking

Select Year and Month as required on the Contents page. Entering Banking Page will LOAD DATA file if previously save and display daily entry. The Banking Page handles Transactions as a mini-spreadsheet with the Calculator used as a Currency Ready Reckoner. Use left/right cursors to select the Day if different from that displayed.

Enter Transactions by pressing (**T**). The top left cell or last currently used will be highlighted. Navigate the various cells with <**Cursor keys**> followed by <**Enter**> key. The left Columns use **TextED** allowing alphanumeric entries. The right two currency columns use the **Calculator Editor.**

For independent use of the **Calculator** press (**C**). Directly type in numbers and Arithmetic Functions or use Cursors and Enter. Use (**Esc**) to Exit either Editor. Press <**Spacebar**> or select (**S**)ave to store the current Data.



```
1600 REMark Organiser Banking Page      Daily Transactions & Calculator

1602 DEFine PROCedure Banking
1603 ch%=1::Prn_Bold 2,32,30,'Banking' :OLoad 2,'Org_'&yr%&'_B_'&mth$(m%)
1604 BLOCK#ch%,120,12,350,28,7:Bank_Page:ch%=1:tchk=0:tl=0:oy%=y2%:om%=m2%
1605 REPeat Bank_lp
1606   Chg_Day:Bank_Ent:k=CODE(INKEY$(-1))
1607   SELect ON k
1608     =     32:OSave 2,'Org_'&yr%&'_B_'&mth$(m%):EXIT Bank_lp
1609     =83,115:OSave 2,'Org_'&yr%&'_B_'&mth$(m%):FileCLS
1610     =84,116:Transactions                 :REMark (T)ransactions
1611     =67,  99:Calculator '0'              :REMark (C)alculator
1612     =192:d%=d%-1:IF d%<1:d%=1
1613     =200:d%=d%+1:IF d%>dm%(m%):d%=dm%(m%)
1614   END SELect
1615 END REPeat Bank_lp
1616 END DEFine
```

```
1618 DEFine PROCedure Bank_Ent
1619 FOR b=1 TO 9:FOR a=1 TO 3:bink%=248:OTrans a,b:END FOR a:END FOR b
1620 END DEFine
```

```
1622 DEFine PROCedure Transactions
1623 REPeat Acc_lp
1624    bink%=1:OTrans tc%,tr%:k=CODE(INKEY$(-1)):bink%=248:OTrans tc%,tr%
1625    SELect ON k
1626      =27:EXIT Acc_lp
1627      =10:IF tc%=1:bink%=0:TextEd 1,cm%,sc%,1,sx%,sy%,Tran$(d%,tr%,tc%)
1628           IF tc%>1:tchk=1:Calculator Tran$(d%,tr%,tc%):tchk=0
1629      =192:IF tc%>1:tc%=tc%-1
1630      =200:IF tc%<3:tc%=tc%+1
1631      =208:IF tr%>1:tr%=tr%-1
1632      =216:IF tr%<9:tr%=tr%+1
1633    END SELect
1634 END REPeat Acc_lp
1635 END DEFine
```

OTrans takes the field's column **tc%** and row **tr% and** calculates coordinates **sx%,sy%**, and the string column **sc%** to define string width **sw%**. Character mode **cm%** is an attribute required by **TextEd.** The boarders outline empty cells or with DATA entries of selected Day Printed to screen. To Edit and/or Exit cell press <**Enter**>. Select another cell or press <**Esc**> to exit the Spreadsheet.

```
1637 DEFine PROCedure OTrans(tc%,tr%)
1638 IF tc%=1:tl%=0:sc%=16:sw%=98:sx%=20:cm%=0
1639 IF tc%>1:tl%=9-LEN(Tran$(d%,tr%,tc%)):sc%=9:sw%=56:sx%=3+tc%*59:cm%=1
1640 IF tc%>1 AND tl%=9:Tran$(d%,tr%,tc%)='0':tl%=8
1641 sy%=62+tr%*12:BLOCK#ch%,sw%,10,sx%,sy%,7
1642 BLOCK#ch%,sw%,1,sx%-1,sy%-1,bink%:BLOCK#ch%,sw%,1,sx%-1,sy%+11,bink%
1643 BLOCK#ch%,1,11,sx%-2,sy%,bink%   :BLOCK#ch%,1,11,sx%+sw%-1,sy%,bink%
1644 STRIP#ch%,7:CURSOR#ch%,sx%,sy%:PRINT#ch%,FILL$(' ',tl%)&Tran$(d%,tr%,tc%)
1645 END DEFine
```

```
1647 DEFine PROCedure Bank_Page                         Set Up Page display
1648 ch%=1:RESTORE 1650
1649 FOR i=1  TO 7:READ a,b,H$:CURSOR#ch%,a,b:PRINT#ch%,H$
1650 DATA 190, 46,'Contents',256,46,'(S)ave',256,60,'(C)alculator'
1651 DATA  20, 60,'(T)ransaction',156,60,'-/+',212,60,'Bal'
1652 DATA  20,192,'(?)Edit ←↑↓→← (Esc)Rtn'
1653 BLOCK#ch%,12,3,176,48,0:BLOCK#ch%,2,4,104,194,0
1654 ch%=3:PAPER#ch%,7:BORDER#ch%,1,248:INK#ch%,0:CLS#ch%:CSIZE#ch%,0,0
1655 ch%=4:PAPER#ch%,248:INK#ch%,7:STRIP#ch%,0:CLS#ch%:CSIZE#ch%,2,1
1656 FOR kr=1 TO 5
1657    FOR kc=1 TO 4
1658      BLOCK#ch%,24,20,-26+kc*29,2+kr*22,0
1659      CURSOR#ch%,-20+kc*29,3+kr*22:PRINT#ch%,Key$(kc,kr)
1660    END FOR kc
1661 END FOR kr
1662 ch%=5:PAPER#ch%,7:BORDER#ch%,2,0:INK#ch%,0:CLS#ch%:CSIZE#ch%,2,1
1663 c%=2:r%=3:k=0:km=0:Num$='0':num=0:dot=0:chk=1:slen=5:tc%=1:tr%=1
1664 END DEFine
```

QBITS Text Editor Review

Addresses, Banking, Diary, Meetings and **Planner** each use differing Character Sets or String handling Formats. To meet this requirement, investigating various Editor codes produced a spiralling list each with problems of functionality. In the end a rather more pragmatic approach prevailed to obtain the minimum acceptable code to get the job done!

QBITS Text Editor Considerations

Previously written **QBITS** code for editing a string was a starting point, this included a character **Add**/**Delete** function anywhere within the string. The challenge now was to introduce a **Wrap** for multiple rows, and a **New Line** function, while making sure the character string remained within a defined screen area.

Other thoughts that came to mind. Generally, most keyboards come with a dedicated **Delete** and **Backspace.** The **QL** Keyboard uses **CTRL Right** and **CTRL Left** Cursor keys. For the **QBITS TextED** it was decided to extend these actions to the Function keys. **F1** to act as a Backspace, Deleting the character left of cursor position. **F2** to Delete character at cursor position. **F3** Delete to End of Line (opposite to a New Line function) and **F4** Delete all of the current string within the text box.

When deleting a String (**F4**) this presented an opportunity to store it in a buffer and RESTORE using (**F5)**. This meant a character string could be transferred to other fields. However, problems arise between different field sizes and control of designated character sets. Therefore, this function had to be restricted, primarily if a string is longer than transferred field space, the string is truncated.

To identify the screen position within a current string an **underline** style cursor was chosen. This to move horizontally and vertically anywhere inside the columns and rows of the assigned screen area. However, it must also be retained within the columns and rows of the current string length.

QBITS Text Editor ASCII Codes

Published International standards for data fields helped in deciding the ASCII code sets for Names, Postal Address, Post/Zip Codes, Telephone numbers and Email address. For control of the differing code sets, a method of scalable character groups is deployed beginning with numbers 0-9 and printable characters such as space, hyphen, period, then adding upper-case alphabetic characters followed by lower case, then finally building to a full set to include all punctuation characters, mathematical symbols and brackets etc.

‘**space**’ **! ” # $ % & ‘ ( ) * + , - . /**	codes	32 - 47
0123456789	codes	48 - 57
: ; < = > @	codes	58 - 64
ABCDEFGHIJKLMNOPQRSTUVWXYZ	codes	65 - 90
[\ ] ^ £	codes	91 - 96
abcdefghijklmnopqrstuvwxyz	codes	97 - 122
{ | } ~	codes	123 - 126

This in general terms were the goals set for the **QBITS Text Editor**.

QBITS Text Editor Character String

In SuperBASIC an input of character codes grouped as a string variable is suffixed with ‘$’. Therefore, the first step is **Add**ing characters to the string; i.e. **str$**:

```
k$=INKEY$        str$=k$ to begin then, str$=str$&k$ to continue adding characters
```

To add characters within the string requires a little more manipulation. First, a variable to locate the current character position **cp%** then the string is added to as follows:

```
 k$=INKEY$       str$=str$(1 to cp%-1)&k$&str$(cp% TO):cp%=cp%+1
```

QBITS Text Editor Character Delete

In a similar way characters in the string can be deleted, here another variable is needed to identify the string length **sl%**. This creates a number of possible **Delete** outcomes:

```
IF cp%=1                    : str$=str$(2 TO) : sl%=sl%-1
IF cp%>1 AND cp%<sl%        : str$=str$(1 to cp%-1)&str$(cp%+1 TO) : sl%=sl%-1
IF cp%>1 AND cp%=sl%        : sl%=sl%-1 : str$=(1 to sl%)
IF str$=”                   : sl%=0 : cp%=1
```

Further constraints are the number of columns s**c%** and rows sr**%** and maximum string length **sm%**, which set the boundaries within which string manipulation can take place.

QBITS Text Editor New Line

For a **New Line** the **cp%** character position is moved to the beginning of the next row**.** For this, variable **cn%** is introduced to count character spaces to extend the string into the next row (still checking that s**c%,sr%** & **sm%** boundaries are not exceeded).

```
IF sl%=0: Return
IF sl% DIV sc%<sr%-1                        Check row position sr%-1
   cn%=sc%-cp% MOD sc%+1                    Count number of spaces to end of row
   IF cp%<=sl%:str$=str$(1 TO cp%-1)&FILL$(‘ ‘,cn%)&str$(cp% TO)
   IF cp% > sl%:str$=str$(1 TO cp%-1)&FILL$(‘ ‘,cn%)
   cp%=cp%+cn% : sl%=sl%+cn%
END IF
```

QBITS Text Editor Character Display

Characters are printed to screen using the **TextEd** attributes to identify **SuperBASIC** output channel **ch%**, the SuperBASIC **ASCII** code set selected by IF **cm%** statements and acted upon by **SELect ON** k Input Key. The size of screen area is calculated from columns **sc%** and rows **sr%,** and its start screen location **sx%,sy%.** The string mechanism for **Wrap** is controlled by a check row variable **cr%**.

QBITS Text Editor Position Cursor

The QBITS Editor uses the BLOCK command to provide an underscore visual indicator. CSIZE 0,0 (six pixels wide) is used to calculation current character cursor coordinates c**x%, cy%.** The variable **cp%** is now used for string position, for calculating cursor pixel position and for checking against **str$** boundaries: 0, s**c%, sr%, sl%** and **sm%**.

Variables Summary: k$, str$, cp%, sl%, sc%, sr%, sm%, cn%, sx%, sy%, cr%

```
1700 REMark QBITS Text Editor     Set channel,chr mode,col,row,x,y,chr string

1702 DEFine PROCedure TextEd(ch%,cm%,sc%,sr%,sx%,sy%,str$)
1703 cx%=sx%:cy%=sy%:cp%=1
1704 REPeat Edit_lp
1705    Prn_Str str$:Prn_Cur:k=CODE(INKEY$(-1))
1706    SELect ON k
1707      = 10:IF sr%>1:Add_Nln                        :REMark New Line
1708      = 27:EXIT Edit_lp                            :REMark End Edit
1709      = 32 TO 125:k$=":Chk_Chr:IF k$>":Add_Chr     :REMark Add Character
1710      =194,232   :IF cp%>1:cp%=cp%-1  :Del_Chr     :REMark F1 Delete Left
1711      =202,236                        :Del_Chr     :REMark F2 Delete Above CURSOR
1712      =240                            :Del_EOL     :REMark F3 Delete End of Line
1713      =244                            :Del_Str     :REMark F4 Delete Page
1714      =248                            :Res_Str     :REMark F5 Restore Page
1715      =192:IF cp%> 1      :cp%=cp%-1               :REMark Move Left
1716      =200:IF cp%<=sl% :cp%=cp%+1                  :REMark Move Right
1717      =208:IF cp%>sc%  :cp%=cp%-sc%                :REMark Move Up
1718      =216:IF cp%<sl%-sc%:cp%=cp%+sc%:ELSE IF sl%>0:cp%=sl%  :REMark Move Down
1719    END SELect
1720 END REPeat Edit_lp
1721 BLOCK#ch%,sc%*6,sr%*10,sx%,sy%,bc%:STRIP#ch%,bc%:Prn_Str str$
1722 END DEFine
```

Note: Text Box Cleared and returned Character String Printed with BkGnd Colour Bc%

```
1724 DEFine PROCedure Chk_Chr                           Checks for Character Set
1725  IF cm%=0                :SELect ON k=32 TO 127    :k$=CHR$(k):RETurn
1726 IF cm%=1 OR cm%>2        :SELect ON k=32,43,45,46,48 TO  57 :k$=CHR$(k)
1727  IF cm%>1                :SELect ON k=65 TO  90    :k$=CHR$(k)
1728  IF cm%=2 OR cm%=3       :SELect ON k=97 TO 122    :k$=CHR$(k-32)
1729  IF cm%>3                :SELect ON k=97 TO 122    :k$=CHR$(k)
1730  IF cm%=4                :SELect ON k=64           :k$=CHR$(k)
1731  IF cm%=5                :SELect ON k=42,47,61     :k$=CHR$(k)
1732 END DEFine

1734 DEFine PROCedure Add_Chr                           Adds Character to String
1735 IF cp%=1 AND sl%=0:str$=str$&k$
1736 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&k$&str$(cp% TO sl%)
1737 IF cp%>=1 AND cp%=sl%:str$=str$(1 TO cp%-1)&k$&str$(cp%)
1738 IF cp%>1 AND cp%>sl%:str$=str$&k$
1739 IF cp%=sm%:str$(cp%)=k$
1740 IF sl%<sm%:sl%=sl%+1:ELSE sl%=sm%
1741 IF cp%<sm%:cp%=cp%+1:ELSE cp%=sm%
1742 END DEFine

1744 DEFine PROCedure Del_Chr                           Deletes Character form string
1745 IF cp%=sl%:str$=str$(1 TO sl%-1):sl%=sl%-1
1746 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&str$(cp%+1 TO sl%):sl%=sl%-1
1747 IF cp%=sm%:str$=str$(1 TO sm%-1):cp%=cp%-1:sl%=sm%-1
1748 IF cp%=1 AND sl%=1:str$=" ":sl%=0
1749 END DEFine
```

```
17510 DEFine PROCedure Del_EOL                      Delete to End of Line [F3]
1752 IF cp%>=sl%:RETurn
1753 buf$=str$:cn%=1+sc%-cp% MOD sc%:IF cn%>sl%-cp%:cn%=sl%-cp%
1754 str$=FILL$(' ',sl%):Str_Prn str$
1755 str$=buf$(1 TO cp%-1)&buf$(cp%+cn% TO LEN(buf$)):sl%=LEN(str$)
1756 END DEFine

1758 DEFine PROCedure Del_Str                        Clear Text Box [F4]
1759 buf$=str$:str$="":sl%=0:cp%=1:BLOCK#ch%,sc%*6,sr%*10,sx%,sy%,7
1760 END DEFine

1762 DEFine PROCedure Res_Str                       Restore Text from Buffer [F5]
1763 sl%=LEN(str$):IF sl%>sm%:str$=buf$(1 TO sm%):sl%=sm%:ELSE str$=buf$
1764 END DEFine

1766 DEFine PROCedure Add_Nln                       Add New line
1767 IF sl%=0:RETurn :ELSE cn%=1+sc%-cp% MOD sc%
1768 IF sl% DIV sc%<sr%-1 OR sl%+cn%<sm%
1769    IF cp%<=sl%
1770       str$=str$(1 TO cp%-1)&FILL$(' ',cn%)&str$(cp% TO sl%)
1771       cp%=cp%+cn%:sl%=LEN(str$)
1772    END IF
1773   IF cp%> sl%:str$=str$(1 TO cp%)&FILL$(' ',cn%):cp%=cp%+cn%:sl%=cp%:END IF
1774 END IF
1775 END DEFine

1777 DEFine PROCedure Prn_Str(str$)                 Print String in Text Box
1778 BLOCK#ch%,sc%*6,sr%*10,sx%,sy%,7:STRIP#ch%,7   :REMark Text Clear
1779 sm%=sc%*sr%:sl%=LEN(str$):IF sl%>sm%:sl%=sm%   :REMark String Length
1780 cr%=sl% DIV sc%:IF cr%=sr%:cr%=sr%-1            :REMark Carriage Return
1781 FOR r=0 TO cr%
1782   IF sl%>0 AND sl% MOD sc%>=0
1783     CURSOR#ch%,sx%,sy%+r*10:PRINT#ch%,str$(1+sc%*r TO sc%+sc%*r)
1784   END IF
1785 END FOR r
1786 IF sl%=0:CURSOR#ch%,sx%,sy%:PRINT#ch%,' ':END IF
1787 END DEFine

1789 DEFine PROCedure Prn_Cur                       Reposition String Position Cursor
1790 BLOCK#ch%,6,1,cx%,cy%+9,7
1791 IF cp% MOD sc%=0:cx%=sx%+sc%*6-6             :cy%=sy%+cp% DIV sc%*10-10
1792 IF cp% MOD sc%>0:cx%=sx%+cp% MOD sc%*6-6:cy%=sy%+cp% DIV sc%*10
1793 IF cp%=1        :cx%=sx%                :cy%=sy%
1794 IF cp%>=sm% :cx%=sx%+sc%*6-6 :cy%=sy%+sr%*10-10
1795 BLOCK#ch%,6,1,cx%,cy%+9,2
1796 END DEFine
```

QBITS Calculator

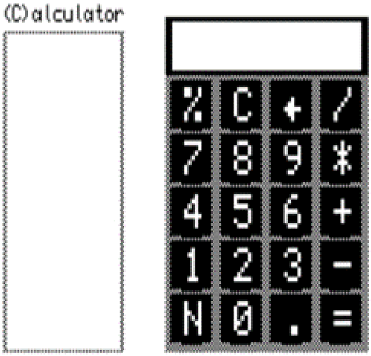
The Calculator Editor is for working with currencies printed to two decimal places. The Arithmetic Functions are [**+**] Add, [**-**] Subtract, [*****] Multiply, [/] Divide. Percentage a number [*] times a value of [%]. Other keys, [=] Equal sign, [.] Decimal Point, [**N**]egate **-/+**number, [**C**]lear all, [🡸] Delete last character.

```
1800 REMark QBITS Calculator (99999.89)

1802 DEFine PROCedure Calculator(Num$)
1803 REPeat Calc_lp
1804   INK#4,2:Calc_Key c%,r%:PRINT#5,FILL$(' ',9-LEN(Num$))&Num$
1805   IF tchk=1:CURSOR#1,sx%,sy%:PRINT#1,FILL$(' ',9-LEN(Num$))&Num$
1806   k=CODE(INKEY$(-1)):INK#4,7:Calc_Key c%,r%:IF k=10:k=CODE(Key$(c%,r%))
1807   SELect ON k
1808     =27:INK#4,7:Calc_Key c%,r%:ch%=1:EXIT Calc_lp
1809     =194                :Calc_BkSp                  :REMark [»] Delete
1810     =67,99              :Calc_Cls                   :REMark [C]lear
1811     =78,110             :Calc_Negate                :REMark [N]egate
1812     =37,42,43,45,47,61 :Calc_Func                   :REMark [%/*+-=] Function
1813     =46:IF dot=0        :Calc_Dot   :chk=0:dot=1    :REMark [.] Decimal Point
1814     =48 TO 57           :Calc_Add                   :REMark [0 to 9]
1815     =192:c%=c%-1:IF c%<1:c%=4                       :REMark Key column
1816     =200:c%=c%+1:IF c%>4:c%=1
1817     =216:r%=r%+1:IF r%>5:r%=1                       :REMark Key row
1818     =208:r%=r%-1:IF r%<1:r%=5
1819   END SELect
1820 END REPeat Calc_lp
1821 IF Num$>0:km=32:Calc_Dec
1822 END DEFine
```

Note: REPeat Calc_lp checks for Numeric Input or Maths Function, keyed directly or via **Cursor Keys & Enter**.

Note: Printing Numbers when Growing from Right to Left

```
PRINT#5,FILL$(' ',9-LEN(Num$))&Num$
```



```
1824 DEFine PROCedure Calc_Key(c%,r%)          Updates the Highlighted Key on KeyPad
1825 ch%=4:CURSOR#ch%,-20+c%*29,3+r%*22:PRINT#ch%,Key$(c%,r%)
1826 END DEFine

1828 DEFine PROCedure Calc_BkSp                Deletes Last Character
1829 Num$=Num$(1 TO LEN(Num$)-1)
1830 IF LEN(Num$)=0:Num$='0':chk=1:dot=0:slen=5
1831 IF LEN(Num$)=1 AND '-' INSTR Num$=1:Num$='0':chk=1:dot=0:slen=5
1832 IF '.' INSTR Num$=0:dot=0:slen=5
1833 IF '.' INSTR Num$>1:slen=Num$-LEN(INT(Num$))+2
1834 END DEFine
```

```
1836 DEFine PROCedure Calc_Cls                    Resets Numbers Num$,num, Clears Work Areas
1837 CLS#3:CLS#5:Num$='0':num=0:dot=0:slen=5
1838 END DEFine

1840 DEFine PROCedure Calc_Negate                  Toggles +/- Value of Number
1841 IF Num$='0':RETurn
1842 IF Num$(1)='-':Num$=Num$(2 TO LEN(Num$)):ELSE Num$='-'&Num$
1843 END DEFine

1845 DEFine PROCedure Calc_Dot                    Adds Decimal Point
1846 IF chk=1:Num$='0.':slen=4:RETurn
1847 IF Num$-INT(Num$)=0:Num$=Num$&'.':slen=LEN(Num$)+2
1848 END DEFine

1850 DEFine PROCedure Calc_Add                    Adds a Numerical Value
1851 IF chk=1:chk=0:Num$=CHR$(k):ELSE IF LEN(Num$)<slen:Num$=Num$&CHR$(k)
1852 END DEFine
```

Note: To avoid QL SuperBASIC printing values as self-describing integers (1E6 etc.) and QPC2 SMSQ/E apparent rounding up of 99999.99 to 100000. The QBITS Calculator is limited to string values of only eight numeric characters ranged between +99998.99 and -99998.99 exceeding these values incurs an Overflow.

```
1854 DEFine PROCedure Calc_Err
1855 IF Num$>99999.89 OR Num$<-99999.89:Num$='!':ELSE num=Num$
1856 END DEFine
```

Note: Formatting to two decimal points requires checking if a decimal point already exists within the Character String (Num$) and then to what number of decimal places.

```
1858 DEFine PROCedure Calc_Dec                    Identifies and Adds Full Decimal Notation
1859 IF Num$='!':PRINT#3,'= Overflow!':Num$='0':num=0:RETurn
1860 IF '-' INSTR Num$=0 AND Num$<.1:Num$='0':RETurn    :REMark QL (1E6 etc)
1861 IF Num$-INT(Num$)=0:dec$='.00' :ELSE dec$=Num$('.' INSTR Num$ TO LEN(Num$))
1862 IF LEN(dec$)<3:dec$=dec$&'0'
1863 Num$=INT(Num$)&dec$(1 TO 3):PRINT#3,CHR$(km);FILL$(' ',10-LEN(Num$))&Num$
1864 END DEFine
```

Note: km holds the math function [**% / * + -**] num & Num$ the numeric values (n1 n2).

```
1866 DEFine PROCedure Calc_Func                   Carries out the Math Functions
1867 IF km=47 AND Num$='0':chk=1:RETurn
1868 IF num=0:km=32:Calc_Err:Calc_Dec:chk=1:dot=0:slen=5:num=Num$:km=k:RETurn
1869 IF k=37:km=37:k=61                           :REMark '%' Percentages
1870 IF km<>61:Calc_Dec Num$:chk=1                :REMark '/ * + -' Functions
1871 SELect ON km
1872   =37:Num$=num*Num$/100    :REMark '%' n='n1*n2/100'
1873   =47:Num$=num/Num$        :REMark '/'   'n1/n2'
1874   =42:Num$=num*Num$        :REMark '*'   'n1*n2'
1875   =43:Num$=num+Num$        :REMark '+'   'n1+n2'
1876   =45:Num$=num-Num$        :REMark '-'   'n1-n2'
1877 END SELect
1878 Calc_Err:km=61:Calc_Dec:km=k:chk=1           :REMark Results
1879 END DEFine
```

QBIT Organiser Data Storage

Dependant on Device/Year/Month selected Data Files are Loaded if previously Saved relevant to the Organiser Page being accessed. The field entries are then displayed in accordance to the Record / Day selected. Array fields are not Deleted or Overwritten when leaving an Organiser Page.

Note: LOAD Config DATA on Initialising Organiser or from Contents Page with change of Theme.

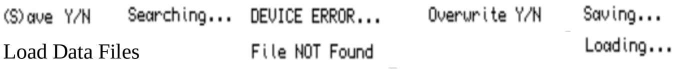
```
1900 REMark Organiser Data Storage

1902 DEFine PROCedure FCheck
1903 STRIP#ch%,bc%:CURSOR#ch%,256,46:PRINT#ch%,'Searching...     '
1904 PAUSE 30:eck=0:ck%=0:DELETE drv$(dn%)&'FList'
1905 OPEN_NEW#9,drv$(dn%)&'FList':DIR#9,drv$(dn%):CLOSE#9
1906 OPEN_IN#9,drv$(dn%)&'FList'
1907 REPeat dir_lp
1908    IF EOF(#9):CLOSE#9:ck=0:EXIT dir_lp
1909    INPUT#9,DFile$:IF DFile$==OrgFile$:CLOSE#9:ck=1:EXIT dir_lp
19107END REPeat dir_lp
1911 END DEFine

1913 DEFine PROCedure OLoad(OFT,OrgFile$)                    Check File exists in DIR List
1914 ch%=1:FCheck
1915 IF ck=0 OR eck=1
1916    CURSOR 256,46:PRINT 'File NOT Found':eck=0:PAUSE 50:RETurn
1917 END IF
1918 OPEN_IN#9,drv$(dn%)&OrgFile$
1919 CURSOR#ch%,256,46:PRINT#ch%,'Loading...  ';drv$(dn%)&OrgFile$
1920 IF OFT=0:FOR a=0 TO 6:INPUT#9,bcol%(a)                 :REMark Config
1921 IF OFT=1                                               :REMark Addr List
1922    FOR a=1 TO 50:FOR b=1 TO 5:INPUT#9,name$(a,b):END FOR b:END FOR a
1923    FOR a=1 TO 50:FOR b=1 TO 6:INPUT#9,mail$(a,b):END FOR b:END FOR a
1924    FOR a=1 TO 50:INPUT#9,email$(a)\addr$(a)
1925 END IF
1926 IF OFT=2                                               :REMark Banking
1927    FOR a=1 TO 31
1928     FOR b=1 TO 8:FOR c=1 TO 3:INPUT#9,Tran$(a,b,c):END FOR c:END FOR b
1929    END FOR a
1930 END IF
1931 IF OFT=4                                               :REMark Diary
1932    FOR a=1 TO 31:INPUT#9,Jrnl$(a)
1933    FOR a=1 TO 31:INPUT#9,Rmdr$(a)
1934    FOR a=1 TO 31:FOR b=1 TO 5:INPUT#9,Wth%(a,b):END FOR b:END FOR a
1935 END IF
1936 IF OFT=5                                               :REMark Meetings
1937    FOR a=1 TO 31:FOR b=1 TO 6:INPUT#9,Appt$(a,b):END FOR b:END FOR a
1938    FOR a=1 TO 31:INPUT#9,Note$(a):END FOR a
1939 END IF
1940 IF OFT=6                                               :REMark Planner
1941    FOR a=1 TO 12:FOR b=1 TO 31:INPUT#9,Plnr%(a,b):END FOR b:END FOR a
1942    FOR a=1 TO 12:INPUT#9,Pcol%(a)\Event$(a)
1943 END IF
1944 CLOSE#9:PAUSE 50:FileCLS
1945 END DEFine
```

Note: On leaving a Page press <**Spacebar**> or by select (**S**)ave you are prompted to Store the current Data. Either the data will be saved as a New file or if previously saved then an Overwrite Y/N prompt will be given.

If the Device or Medium is not found a **DEVICE ERROR** will be given. To save Data return without Saving to the **Contents** Page and select a Device that is attached (**Do not change Year or Month**). Again return to **Organiser Page** last accessed, the field entries should still be there so that they can now be Saved.

```
1947 DEFine PROCedure OSave(OFT,OrgFile$)
1948 ch%=1:STRIP#ch%,bc%:CURSOR#ch%,298,46:PRINT#ch%,'Y/N  ';drv$(dn%);OrgFile$
1949 PAUSE:IF KEYROW(5)<>64:BLOCK#ch%,20,10,300,46,bc%:RETurn :ELSE FCheck
1950 IF eck=1
1951   CURSOR#ch%,256,46:PRINT#ch%,'DEVICE ERROR...':eck=0:PAUSE 50:RETurn
1952 END IF
1953 IF ck=1
1954   CURSOR#ch%,256,46:PRINT#ch%,'Overwrite Y/N  ':PAUSE:IF KEYROW(5)<>64:RETurn
1955 END IF
1956 DELETE drv$(dn%)&OrgFile$:OPEN_NEW#9,drv$(dn%)&OrgFile$
1957 CURSOR#ch%,256,46:PRINT#ch%,'Saving...       ';OrgFile$
1958 IF OFT=0:FOR a=0 TO 6:PRINT#9,bcol%(a)               :REMark Config
1959 IF OFT=1                                             :REMark Addr List
1960    FOR a=1 TO 50:FOR b=1 TO 5:PRINT#9,name$(a,b):END FOR b:END FOR a
1961    FOR a=1 TO 50:FOR b=1 TO 6:PRINT#9,mail$(a,b):END FOR b:END FOR a
1962    FOR a=1 TO 50:PRINT#9,email$(a)\addr$(a)
1963 END IF
1964 IF OFT=2                                             :REMark Banking
1965    FOR a=1 TO 31
1966      FOR b=1 TO 8:FOR c=1 TO 3:PRINT#9,Tran$(a,b,c):END FOR c:END FOR b
1967    END FOR a
1968 END IF
1969 IF OFT=4                                             :REMark Diary
1970    FOR a=1 TO 31:PRINT#9,Jrnl$(a)\Rmdr$(a)
1971    FOR a=1 TO 31:FOR b=1 TO 5:PRINT#9,Wth%(a,b):END FOR b:END FOR a
1972 END IF
1973 IF OFT=5                                             :REMark Meetings
1974    FOR a=1 TO 31:FOR b=1 TO  8:PRINT#9,Appt$(a,b):END FOR b:END FOR a
1975    FOR a=1 TO 31:PRINT#9,Note$(a):END FOR a
1976 END IF
1977 IF OFT=6                                             :REMark Planner
1978    FOR a=1 TO 12:FOR b=1 TO 31:PRINT#9,Plnr%(a,b):END FOR b:END FOR a
1979    FOR a=1 TO 12:PRINT#9,Pcol%(a)\Event$(a)
1980 END IF
1981 CLOSE#9:PAUSE 50:FileCLS
1982 END DEFine
```

File Actions



```
(S)ave Y/N    Searching...    DEVICE ERROR...    Overwrite Y/N    Saving...
Load Data Files               File NOT Found                      Loading...
```

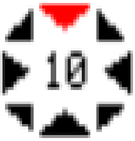
```
1984 DEFine PROCedure FileCLS
1985 CURSOR#ch%,256,46:PRINT#ch%,'(S)ave';FILL$(' ',15)
1986 END DEFine
```

QBITS Organiser Graphics

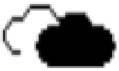
These use a combination of **Vector** and **Bitmap** style designs. **Vectors** use SuperBasic ARC, CIRCLE, LINE and **Bitmaps** use the BLOCK command.

First the image is sketched, then redrawn over a Grid to map the coordinates. **Vectors** and **Graphics coordinates** use x y Axis for Window positioning, **Bitmap** images use top left location of the **Pixel coordinates** system.

```
2000 REMark QBITS Organiser Graphics

2002 DEFine PROCedure WindDir(wn%,wd%,x,y)
2003 LOCal a,b,c,d,e,f,l:RESTORE 2008:BLOCK 18,10,430,184,bcol%(4)
2004 FOR w=1 TO 8
2005    READ a,b,c,d,e,f:IF w=wd%:INK 2:ELSE INK 0
2006    FILL 1:LINE a+x,b+y TO c+x,d+y TO e+x,f+y TO a+x,b+y:FILL 0
2007 END FOR w
2008 DATA 0,3.8,-2,5.5,+2,5.5,3,3,3,5,5,3
2009 DATA 3.5,0,5.5,-1.5,5.5,1.5,3,-3,3,-5,5,-3
2010 DATA 0,-3.5,2,-5.5,-2,-5.5,-3,-3,-3,-5,-5,-3
2011 DATA -3.5,0,-5.5,-1.5,-5.5,1.5,-3,3,-3,5,-5,3
2012 l=INT(LEN(wn%)*6)/2:STRIP bcol%(4):INK 0:CURSOR x,y,-l,-4:PRINT wn%
2013 END DEFine
```

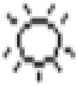


```
2015 DEFine PROCedure Weather(wp)
2016 ch%=1:INK#ch%,gc%:wx%=98:wy%=16:wr%=2
2017 STRIP#ch%,bcol%(4):BLOCK#ch%,46,30,270,174,bcol%(4)
2018 SELect ON wp
2019    =0:Cloud       ch%,wx%-2,wy%,wr%        :Overcast ch%,wx%+2,wy%-1,wr%
2020    =1:Sunny       ch%,wx%,wy%,wr%
2021    =2:Sunny       ch%,wx%+4,wy%+1,wr%*.7 :Cloud ch%,wx%,wy%-1,wr%
2022    =3:Windy       ch%,wx%,wy%,wr%
2023    =4:Showers     ch%,wx%,wy%,wr%
2024    =5:Overcast    ch%,wx%,wy%,wr%
2025    =6:Downpour   ch%,wx%,wy%,wr%
2026    =7:Stormy      ch%,wx%,wy%,wr%
2027    =8:Snow        ch%,wx%,wy%,wr%
2028 END SELect
2029 CURSOR#ch%,258,195:PRINT#ch%,Sky$(wp)&FILL$(' ',10-LEN(Sky$(wp)))
2030 END DEFine
```

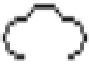


The Weather Patters

```
2032 DEFine PROCedure Sunny(ch%,wx%,wy%,wr%)
2033 CIRCLE#ch%,wx%,wy%,wr% :r1%=wr%*1.5:r2%=wr%*2
2034 FOR i=1 TO 9
2035    LINE#ch%,wx%+r1%*COS(i*60),wy%+r1%*SIN(i*60)
2036    LINE#ch% TO wx%+r2%*COS(i*60),wy%+r2%*SIN(i*60)
2037 END FOR i
2038 END DEFine
```

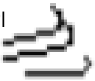


```
2040 DEFine PROCedure Cloud(ch%,wx%,wy%,wr%)
2041 ARC#ch%,wx%-wr%*1.2,wy%-wr% TO wx%-wr%,wy%+wr%,-PI
2042 ARC#ch% TO wx%+wr%,wy%+wr%,-PI*.8 TO wx%+wr%*1.2,wy%-wr%,-PI
2043 END DEFine:
```

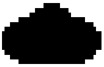


Weather Patterns cont...

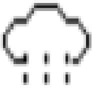
```
2045 DEFine PROCedure Windy(ch%,wx%,wy%,wr%)
2046 LINE#ch%,wx%-wr%*2,wy%+wr%/2 TO wx%,wy%+wr%:ARC#ch% TO wx%,wy%+wr%*2,PI
2047 LINE#ch%,wx%-wr%*2,wy%-wr%/4 TO wx%+wr%*.6,wy%+wr%/4
2048 ARC#ch% TO wx%+wr%*.6,wy%+wr%*1.4,PI
2049 LINE#ch%,wx%-wr%,wy%-wr% TO wx%+wr%,wy%-wr%:ARC#ch% TO wx%+wr%*1.5,wy%,PI
2050 END DEFine
```



```
2052 DEFine PROCedure Overcast(ch%,wx%,wy%,wr%)
2053 FILL#ch%,1:Cloud ch%,wx%,wy%,wr%
2054 LINE#ch%,wx%-wr%,wy%-wr% TO wx%+wr%,wy%-wr%:FILL#ch%,0
2055 END DEFine
```

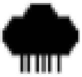


```
2057 DEFine PROCedure Showers(ch%,wx%,wy%,wr%)
2058 Cloud ch%,wx%,wy%,wr%
2059 FOR i=-1 TO 1:LINE#ch%,wx%+wr%*i,wy%-wr%*.8 TO wx%+wr%*i,wy%-wr%*1.2
2060 FOR i=-1 TO 1:LINE#ch%,wx%+wr%*i,wy%-wr%*1.5 TO wx%+wr%*i,wy%-wr%*2
2061 END DEFine
```

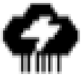


The Text with this is given as Heavy Rain

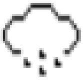
```
2063 DEFine PROCedure Downpour(ch%,wx%,wy%,wr%)
2064 Overcast ch%,wx%,wy%,wr%
2065 FOR i=-1 TO 1 STEP .4:LINE#ch%,wx%+wr%*i,wy%-wr% TO wx%+wr%*i,wy%-wr%*2
2066 END DEFine
```



```
2068 DEFine PROCedure Stormy(ch%,wx%,wy%,wr%)
2069 Downpour ch%,wx%,wy%,wr%:INK#ch%,7:FILL#ch%,1
2070 LINE#ch%,wx%,wy% TO wx%-wr%,wy% TO wx%,wy%+wr% TO wx%+wr%/2,wy%+wr%
2071 LINE#ch% TO wx%,wy%+wr%/2 TO wx%+wr%,wy%+wr%/2 TO wx%-wr%/4,wy%-wr%
2072 LINE#ch% TO wx%,wy%:FILL#ch%,0:INK#ch%,0
2073 END DEFine
```

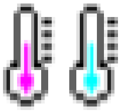


```
2075 DEFine PROCedure Snow(ch%,wx%,wy%,wr%)
2076 Cloud ch%,wx%,wy%,wr%
2077 CIRCLE#ch%,wx%-wr%*.8,wy%-wr%*1.5,wr%/6:CIRCLE#ch%,wx%,wy%-wr%,wr%/6
2078 CIRCLE#ch%,wx%+wr%*.8,wy%-wr%*1.5,wr%/6:CIRCLE#ch%,wx%,wy%-wr%*2,wr%/6
2079 END DEFine
```



Bitmap Thermostat Symbols with Colours shown for High & Low.

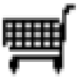
```
2081 DEFine PROCedure Therm(wx,wy,wc)
2082 RESTORE 2083:FOR i=1 TO 18:READ a,b,c,d,e:BLOCK#ch%,a,b,wx+c,wy+d,e
2083 DATA 3,1,3,0,0,  1,8,2,1,0,  1,8,6,1,0,  2,1,1,9,0,  2,1,6,9,0
2084 DATA 1,4,0,10,0, 1,4,8,10,0, 2,1,1,14,0, 2,1,6,14,0, 3,1,3,15,0
2085 DATA 3,13,3,1,7, 7,3,1,10,7, 1,8,4,6,wc, 3,2,3,11,wc
2086 DATA 2,1,8,2,0,  2,1,8,4,0,  2,1,8,6,0,  2,1,8,8,0
2087 END DEFine
```



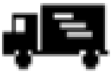
The Diary (R)eminders etc

```
2089 DEFine PROCedure Health(wx,wy)
2090 INK#ch%,0:ARC#ch%,wx,wy-1 TO wx+3.6,wy,-PI TO wx+7.4,wy-1,-PI
2091 LINE#ch% TO wx+5.6,wy-4.2:ARC#ch% TO wx+2,wy-4.2,-PI/3:LINE#ch% TO wx,wy-1
2092 LINE#ch%,wx+1.2,wy-1 TO wx+1.8,wy-1 TO wx+2.2,wy TO wx+3.2,wy-3.5
2093 LINE#ch% TO wx+4,wy-1 TO wx+6,wy-1:INK#ch%,0
2094 END DEFine
```

```
2096 DEFine PROCedure Shop(wx,wy)
2097 RESTORE 2098:FOR i=1 TO 20:READ a,b,x,y,c:BLOCK#ch%,a,b,wx+x,wy+y,c
2098 DATA 3,1,20,0,0, 2,1,19,1,0, 1,6,19,2,0, 1,6,18,6,0, 1,5,0,4,0
2099 DATA 18,1,0,3,0, 18,1,0,5,0, 18,1,0,7,0, 16,1,2,9,0, 14,1,3,11,0
2100 DATA 1,5,3,4,0, 1,5,6,4,0, 1,5,9,4,0, 1,5,12,4,0, 1,5,15,4,0
2101 DATA 1,5,18,4,0, 3,2,2,12,0, 1,4,3,11,0, 3,2,16,12,0, 1,4,17,11,0
2102 END DEFine
```

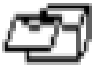


```
2104 DEFine PROCedure Goods(wx,wy)
2105 RESTORE 2106:FOR i=1 TO 13:READ a,b,x,y,c:BLOCK#ch%,a,b,wx+x,wy+y,c
2106 DATA 4,5,0, 6,0, 8,8,3,2,0, 3,3, 4, 3,7, 20,10,12,0,0
2107 DATA 5,2,4,11,0, 3,4,5,10,0, 5,2,25,11,0,  3, 4,26,10,0
2108 DATA 15,1,9,10,0, 3,1,30,10,0, 6,1,16,2,7, 8,1,18,4,7, 8,1,20,6,7
2109 END DEFine
```

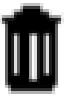


Also used on Address page

```
2111 DEFine PROCedure Post(wx,wy)
2112 LINE#ch%,wx+1,wy+4 TO wx+5,wy+4 TO wx+2.5,wy+3 TO wx+1,wy+4
2113 LINE#ch%,wx+1,wy+4 TO wx,wy+1 TO wx+4,wy+1 TO wx+5,wy+4
2114 LINE#ch%,wx+2,wy+.5 TO wx+2,wy TO wx+6.5,wy TO wx+6.5,wy+3 TO wx+5,wy+3
2115 LINE#ch%,wx+6.5,wy TO wx+7.5,wy+2 TO wx+7.5,wy+5 TO wx+6.5,wy+3
2116 LINE#ch%,wx+7.5,wy+5 TO wx+4,wy+5 TO wx+2.5,wy+4
2117 END DEFine
```



```
2119 DEFine PROCedure Trash(wx,wy)
2120 RESTORE 2121:FOR i=1 TO 8:READ a,b,x,y,c:BLOCK#ch%,a,b,wx+x,wy+y,c
2121 DATA 5,1,5,0,0, 9,1,3,1,0, 15,1,0,2,0, 13,11,1,3,0, 11,1,2,14,0
2122 DATA 1,7,3,5,7, 2,8,7,5,7, 1,7,11,5,7
2123 END DEFine
```



```
2125 DEFine PROCedure Bill(wx,wy)
2126 LINE#ch%,wx-4.5,wy TO wx,wy+2 TO wx+4,wy
2127 LINE#ch%,wx-3.5,wy TO wx-3.5,wy-4 TO wx+3,wy-4 TO wx+3,wy
2128 CIRCLE#ch%,wx-1.8,wy-.8,.8:CIRCLE#ch%,wx-1.8,wy-2.5,.8
2129 LINE wx,wy-2.8 TO wx,wy-1:POINT wx,wy-.5
2130 LINE wx+1,wy-2.8 TO wx+1,wy+.2:LINE wx+2,wy-2.8 TO wx+2,wy+.2
2131 END DEFine
```

Used on Address Page

```
1928 DEFine PROCedure Phone(wx,wy)
1929 RESTORE 1944:FOR i=1 TO 12:READ a,b,x,y,c:BLOCK#ch%,a,b,wx+x,wy+y,c
1930 DATA 19,24,0,0,0, 17,16,1,3,7, 3,2,8,21,7, 1,4,9,20,7
1931 DATA 3,10,7,6,2, 2,3,5,6,2, 2,3,5,13,2, 4,1,8,1,7
1932 DATA 3,10,10,7,4, 1,8,13,8,4, 2,3,8,7,4, 2,3,8,14,4
1933 END DEFine
```