



Sinclair QL Retro Progs!



Sinclair QL Retro Progs!



QBITS Organiser

Wed 1 Jan 2020

Diary

(J)ournal

Lock down due to Coronavirus hasn't dampend my spirits, but I look forward to the day when some normality returns!

Weather Tab



Changeable

Temperature

0°C +H
-L 0°C

Wind Speed

+W
-W 0 mph

Wed 01 Jan 2020

Contents

Set Date ++++

Edit() ++++ (Esc)

(R)eminder

Check for parcel delivery and cards!
Birthday cake!
Zoom Meeting with the Family.



Health



(S)ave

Shop



Delivery



Post



Trash



QBITS Organiser

Thu 7 May 2020

Mail List

Edit() (Esc)

(e)mail

qbts.sin@gmail.com

(A)ddress

Paradise Villa
Anywhere Avenue
Loose Chippings



(P)ost/Zip Code Y42 N07

(D)istrict

(C)ountry

Contents

(Tab)

(L)ast Name

QBITS

(F)irst Names

Sinclair

01234 567890

(M)obile

(H)ome

(O)ffice

(S)ave

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z



QBITS Introduction

A little background research reveals 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 basic design originated from a US company named 'Lefax'. This was the beginnings of the 'Filofax' Personal Organiser, an abbreviation of 'file of facts'.

Revamped by Ian Logan in the nineteen eighties it became the most desirable accessory for the Yuppie brigade. Its popularity grew so much, the company renamed itself after its most famous brand. As a tool, the Personal Organiser contained sections for a Diary, Calendar, Contact list, a Year Planner, and other useful aids.

As Personal Computing became more affordable, it was an obvious contender for extending our growing world of computer dependency. Not forgiving its QL connection PSION brought out a range of Electronic Organisers or Personal Digital Assistants (PDA's). As for me, I wished to create my own version, written in QL SuperBASIC and so the **QBITS Organiser** began its journey.

QBITS Organiser Aspirations

In the eighties I delved more deeply into what a Personal Organiser was about and how I might interpret this Graphically with QL SuperBASIC. The Organiser in simple terms is just an extended Diary, not that I was very good at keeping one myself. My job involving Projects, a Diary with a Calendar helped with scheduling meetings and from that, you might devise a Planner to give an overview of continuing workload. Contact lists were always helpful, then there was the odd calculation reviewing costs or work hour/performance etc. or at home it was working out the month's finances.

For the **QBIT Organiser** early calculations revelled a year's storage of DATA might be in the order of 300KBytes with around 30KBytes for a thousand lines of code. It prompted dividing the DATA storage requirement into monthly SAVE/LOAD files, so as to reduce RAM allocation down to between 60Kbytes and 80Kbytes.

What seemed almost out of reach just a few short decades ago seems insignificant with modern computing power and their Gigabytes of RAM and Storage space. The upgraded versions of QL Hardware, and even more so the Software Emulators of today provide platforms of performance that far exceed the QL's humble beginnings.

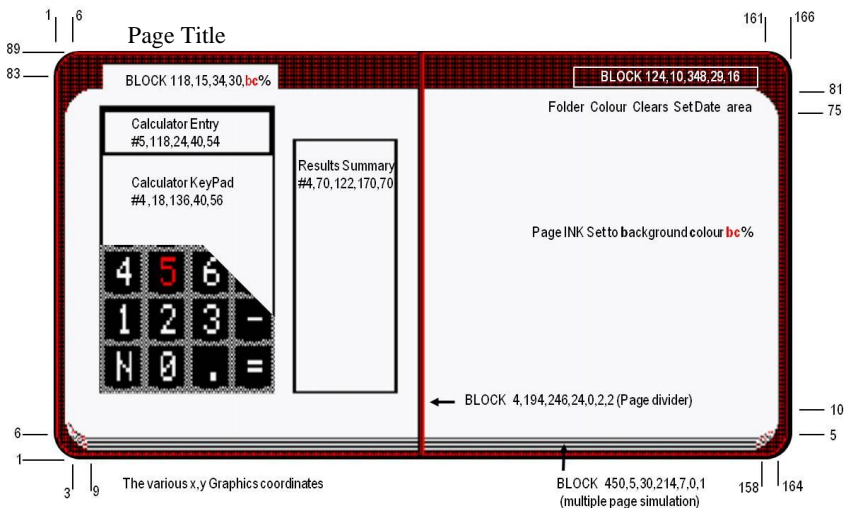
For me the coding for the **QBITS Organiser** springs from a collection of programs each interesting in their own development and usefulness in expanding my understanding of working with **QL SuperBASIC**.

QBITS Organiser Getting Started

Although ideas for various sections had layouts drawn and notes on their potential function, unfortunately early code resided on floppies that are now largely defunct. However, some hand written notes on displays together with bits of codes survived to give an overview of the original eighties Organiser Contents and the possibilities to be explored in reviving the Programme.

QBITS Organiser Screen Layouts

Using QL SuperBASIC WINDOW's, channel#6 is opened to deploy as a blank screen background. WINDOW #1 the main operating channel and 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 Title and areas for displaying the Organisers current and working dates.



The additional WINDOW# channels are used for the Calculator Graphical representation.

QBITS Organiser Control Keys

The Page layouts have a number of commonly used control keys this helped in coordinating their design. The first was changing the work Date for editing or viewing the contents of a particular day. To Edit DATA fields Selecting the bracketed character of their (H)eading seemed a logical solution. In some cases, use of the <Tab> key to Scroll possible options. Contents access <Spacebar> to select or return and <Esc> to return from other actions.

Cursor keys **Left/Right/Up/Down** <Spacebar> <Enter> <Esc> <Tab>

Edit () – key First Letter of DATA Field (H)eading... (Esc) to return

QBITS Organiser Info

What is a desktop personal organiser one description given was a box or open tray typically kept on the surface of a desk divided into compartments to hold office supplies? A little more interesting was a small book or electronic device in which information is stored, such as names, addresses, phone numbers, dates of meetings which are used to organise your time? Another stated it as software that simulates an object normally found on an office desktop, such as a calculator, notepad and appointment calendar.

So if collecting and writing up information to assist or remind what needs to be done, how should it be subdivided. Listing the possible requirements: Appointments, Notepad? A Mail list (contact names, addresses, phone numbers). A Calendar page identifying the days of the week, and then a year Planner showing Events and important Anniversaries. A Diary with a Journal for writing up the day's experiences, then how about a Reminder section. Recording the weather that might be relevant for some. A Calculator was a looked forward to addition. The idea to incorporate it as part of a Banking page with a mini spreadsheet came a little later (early nineties).

QBITS Organiser Calendar

The double page is subdivided into four quarters beginning with the calculated day of the week relevant to the year selected using <Up/Down> cursor keys and Set with <Enter>.

The Gregorian Calendar and Key Value Method was chosen to calculate the week day this uses codes for different months and years to 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 year. For example, $2450 - 400 = 2050$ 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-Mon day and so on with 0 Saturday.

QBITS Organiser Planner

The second layout pursued in the eighties was the Planner for this the year is set out with the months in rows and days as columns. Each is identified as a cell which can be colour coded. Having Set a specific Date (day/month), the selected cell colour is chosen from the available range. As the original QL Basic has limited solid colours, the Stipple patterning method was used to extend the range. The concept was to link these colour codes with a Table of Event choices. Set within the program, the latest version now allows editing of Event Titles and are Saved as part of the Planner DATA File.

QITS Organiser Mail List

An essential component of any Organiser is no doubt its contact list. The question to ask is what information it should contain and in what format. The **QBITS Organiser** page for the **Mail List** had to take into account what is practicable in the number of fields and size, yet still accommodate most people's requirements.

An obvious starting point, **Names**, Last name, First name, nickname or non de plume. A consideration to take into account was a method for searching through these names to find a related record. As **emails** are now part of life what conventions exist over length and what are the allowable printable characters such as the @ symbol. For Phone numbers, Mobile, Home & Office seem a sensible choice. **Addresses** come with a House or Apartment name or number, Street or Road, a local area; name of Village, Town or **District** of City, then the **Country**. National Postal Services use a **Post/Zip** code.

Convention for Data Fields Entries

Checking government guidelines became an interesting subject. First input to any fields are codes for printable Characters, A Alphabetic, N Numeric and S Special for example the @ symbol. For names and addresses, field lengths are 30 to 35 using alphanumeric characters and in the main use uppercase letters. A full email addresses can have 320 characters, the user name limited to 64 plus the @ symbol and domain names to 255. The special characters being Period (.), Underscore (_), Hyphen (-) and Plus sign (+).

QBITS Organiser Mail List Fields

To cover National Postal schemes, what was derived to meet most needs will not meet all local convention. Decided was a Text box with three rows for **Address** using alphanumeric and punctuation characters, Upper Case Alphanumeric characters used by the **Post/Zip** Code, **District** and **Country** single line entries.

For **emails** the present entry limit is 36 alphanumeric characters 0-9, A-Z, a-z, the Special characters being period (.) underscore (_) hyphen (-) and the @ symbol.

QBITS Organiser Mail List Page Display

The layout here is very different, entry fields are grouped on the left and right pages and each given a (**H**)eading. To Edit the respective field requires simply keying in the letter in brackets, when finished use the [**Esc**] key to return. Records can be entered sequentially or on an add hock bases not necessarily in alphabetical order. Use the <**Left/Right**> Cursor keys to display the previous or the next sequential record. Records having been entered or updated are saved when exiting the Page.

QBITS Organiser Mail List Search

Either **Last** or **First** Names can be selected with the (**T**)ab key. Then choose an Alphabet character A-Z using the <**Up/Down**> Cursor keys. Start the search with <**Enter**> key. If no records are found it will return '**Not Found!**'. If there are more entries with the same leading Alphabet letter, key <**Enter**> until the entry sought is found.

Note: Presently the Search function only matches single characters and with two fields. Adding further fields and a full string search may be a future enhancement.

QBITS Organiser Text Editing

Entry and editing data for the various fields required a Text Editor. My experience in the mid-eighties had advanced to only a simple line editor called edlin. The SuperBASIC interpreter offers INPUT and set within a defined WINDOW area could be used to some extent. However, this was somewhat restricted and not easily manipulated to meet the functions envisaged.

The concept of a multipurpose editor began to emerge and by multipurpose it needed to be able to work with different character Sets. For example, only numeric, or Upper-Case alphabet characters (with conversion of lower-case to Upper-case if necessary).

The adding of a Calculator would require a method of input, maybe a specialised form of editor, one that returned currency values to two decimal points.

QBITS Organiser DATA Storage

The Organiser is built upon a collection of tasks each performing some function that in essence creates a storage of helpful information. This, Data created is Saved and Loaded to fulfil each of the Page functions. QBITS Organiser on start-up uses a default drive to find and Load DATA Files applicable to the current Date set by the QL Clock.

The **Appointments**, **NotePad**, **Banking** and **Diary** entries written on a daily bases can generate large DATA Files. It was therefore and perhaps still is prudent to subdivide these into monthly files and so reduce the risk of Data being lost. This does introduce delays when swapping month/year DATA as current info is saved to File and the relevant new month/year Set_Date DATA File is Loaded. A single DATA file is used for the **Mail List**, 50 records generate upwards of 12.5KBytes. In comparison a year's DATA for the **Planner** is quite small, 12 Event names of 15 characters and chart entries stored as integers for each day/month cell is less than 1.5 Kbytes. Later added to this DATA File were the Page Background colours.

QBIT Organiser Program Settings

- Note 1:** Code - Line 1003 dr% Default Storage Drive (see Line 1005 - change?)
QBTS Organiser will Load/Save DATA Files from/to Default Storage Device.
This needs to be connected and have at least 1Mbyte of free space available.
- Note2:** Code - Line 1002 gx, gy Choose for QL Platform Environment
- Note3:** Code - Line 1032 Planner Colour Codes (see Lines 1045/6 to Set)

QBITS Organiser Development

- 1986/7 Original concept Basic designs, early displays of Calendar and Planner.
- 1988 Layouts for Diary and Calculator, Part working Contents page.
- 1991 QBITS Organiser Program issues partly working, floppy code lost.
- 2020 QBITS Organiser Revamped & Revised Functioning Contents:
Text Editor, Calculator, Appointments, Banking, Calendar, Diary, Mail List, Planner. Release QBOrg_v02QPC2 & QBOrg_v02BBQL beta versions

QBITS Organiser Program Code

1000 REMark **QBOrg_v02** (1992 © QBITS Organiser - Update 2020)

1002 MODE 4: gx=20:gy=30 :dr%=1 :REMark BBQL gx=0:gy=0 QPC2 gx=20:gy=30

1003 DIM dev\$(8,5):dr%=4 :REMark Set Storage Device dr%=1 (win2_)

1004 **RESTORE** 1034: **FOR** i=1 TO 8: **READ** dev\$(i)

1005 **DATA** 'flp1_', 'flp2_', 'win1_', 'win2_', 'dos1_', 'dos2_', 'win8_', 'dos8_'

1007 REMark Organiser - Init Windows / Screen Layout / Set Date

1008 REMark TextEd - Line/Page Char Selected Text Editor

1009 REMark Calculator - Currency NN.nn (Dollar Euro Pound Yen)

1011 REMark Appointments - Daily Schedule/NotePad/Planner Event

1012 REMark Banking - Daily Accounts - Calculator

1013 REMark Calendar - Year Display - day/Week/Month

1014 REMark Diary - Journal/Reminders/Weather/Event

1015 REMark Mail - Names/Email/Mobile-Home-Office/Address/Post Codes etc

1016 REMark Planner - Yearly Schedule of Events/Dates

1017 REMark Settings - Storage - Data Files - Page BkGnds

1019 REMark DIMension Arrays

1020 DIM Apt\$(31,6,32),Note\$(31,72) :REMark Appts

1021 DIM Tran\$(31,8,3,20),Key\$(4,5),Num\$(9):num=0 :REMark Banking

1022 DIM Jml\$(31,360),Remd\$(31,216),Wth\$(31,4) :REMark Diary

1023 DIM email\$(50,36),name\$(50,5,25),addr\$(50,96),mail\$(50,6,15):REMark Mail

1024 DIM pln\$(12,31),Event\$(12,12),col\$(12) :REMark Planner

1025 DIM wd\$(7,3),mth\$(12,3),dm\$(12),bgc\$(6),Sky\$(8,10)

1027 bc%=7 :**FOR** i=1 TO 6:bgc%(i)=7 :REMark BkGnd Colours

1028 **RESTORE** 1036:**FOR** i=1 TO 7:**READ** wd\$(i) :REMark Days of the Week

1029 **RESTORE** 1037:**FOR** i=1 TO 12:**READ** mth\$(i),dm%(i) :REMark Months / Days-Month

1030 **RESTORE** 1040:**FOR** i=0 TO 8:**READ** Sky\$(i) :REMark Weather Patterns

1031 **RESTORE** 1043:**FOR** i=1 TO 12:**READ** Event\$(i) :REMark Planner Events

1032 **RESTORE** 1046:**FOR** i=1 TO 12:**READ** col%(i) :REMark Planner Colour codes

1033 **RESTORE** 1048:**FOR** b=1 TO 5:**FOR** a=1 TO 4:**READ** Key\$(a,b):**END FOR** a:**END FOR** b

1035 REMark Days of the week: Month - number of Days

1036 **DATA** 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'

1037 **DATA** 'Jan',31,'Feb',28,'Mar',31,'Apr',30,'May',31,'Jun',30

1038 **DATA** 'Jul',31,'Aug',31,'Sep',30,'Oct',31,'Nov',30,'Dec',31

1039 REMark Weather Patterns

1040 **DATA** 'Changeable', 'Sunny', 'Cloudy', 'Windy', 'Showers'

1041 **DATA** 'Overcast', 'Heavy Rain', 'Stormy', 'Snow'

1042 REMark Planner Events: Colour Codes

1043 **DATA** 'Anniversary', 'Birthday', 'Doctor', 'Dentice', 'Shopping', 'Meetings'

1044 **DATA** 'PayDays', 'Holidays', 'Project 1', 'Project 2', 'Project 3', 'Project 4'

1045 **DATA** 2,4,98,87,163,150,226,222,90,96,154,164 :REMark BBQL Mode 4 Colours

1046 **DATA** 1,2,3,4,5,6,41,48,34,80,230,255 :REMark QPC2 Mode 4 Colours

1047 REMark Calculator Keyboard

1048 **DATA** '%','C','%','/','7','8','9','★', '4','5','6','+'

1049 **DATA** '1','2','3','4','N','0',' ','='

1051 REMark Program Init

1052 dmy\$=DATE\$:yb%=1961:ye%=2050 :REMark Date Settings

1053 C\$="":dr%=8:ct%=65:bc%=7 :REMark C\$ Currency ct% Contents bc% Page Colour

1054 Init_Win : QBITS_Title : Init_Page : Org_Contents

QBITS Organiser Display

This was to be a double Page from the outset with rounded corners for both the outline holder and contents pages. The design was created by experimenting with SuperBASIC ARC, LINE, FILL and INK commands.

```
1056 DEFine PROCEDURE Init_Win
1057 LOCAL a,b,c,d,e,f
1058 OPEN#6,scr_10x10a10x10:WINDOW#8,512,256,gx,gy:PAPER#8,0:CLS#8
1059 OPEN#5,scr_10x10a10x10:WINDOW#7,118,24,40+gx,54+gy :REMark QBCalc Entry
1060 OPEN#4,scr_10x10a10x10:WINDOW#6,118,136,40+gx,56+gy :REMark QBCalc Keys
1061 OPEN#3,scr_10x10a10x10:WINDOW#5,70,122,170+gx,70+gy :REMark QBCalc Results
1062 WINDOW#2,496,220,gx+8,gy+4 :PAPER#2,0:INK#2,7 :REMark SBASIC Editor
1063 WINDOW#1,496,220,gx+8,gy+4 :PAPER#2,0:INK#2,7 :REMark QBOrg Page
1064 WINDOW#0,496,20,gx+8,gy+226 :PAPER#2,0:INK#2,7 :REMark QL Command
1065 ch%=1:SCALE#ch%,100,0,0:chk=0
1066 FOR chk=0 TO 1 Outer Folder
1067 IF chk=0:FILL#ch%,1:INK#ch%,16:RESTORE 1072
1068 IF chk=1:FILL#ch%,0:INK#ch%,2:RESTORE 1072
1069 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
1070 END FOR chk
1071 DATA 6,89,161,89,16,83,166,83,166,6,161,1
1072 DATA 161,1,6,1,1,6,1,6,83,6,89
1073 :
1074 RESTORE 1077: BLOCK#ch%,450,5,30,214,7,0,1:INK#ch%,7 :REMark Page Leaf's
1075 FOR i=1 TO 6: READ a,b,c,d: ARC#ch%,a,b TO c,d,PI/2
1076 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
1077 END DEFine
```

The QBITS Organiser Title uses INK, OVER, CSIZE as in other QBITS Progs

```
1079 DEFine PROCEDURE QBits_Title
1080 ch%=1: QBIT$=QBITS Organiser': OVER#ch%,1:CSIZE#ch%,2,1
1081 INK#ch%,2: FOR i=0 TO 2: CURSOR#ch%,20-i,3:PRINT#ch%,QBIT$
1082 INK#ch%,7: FOR i=0 TO 1: CURSOR#ch%,20+i,2:PRINT#ch%,QBIT$
1083 OVER#ch%,0:CSIZE#ch%,0,0:Page_Cls:Init_Date
1084 END DEFine
```

To the right of QBITS Title is the Current Date Set by the QL Clock dmy%=DATE\$

```
1086 DEFine PROCEDURE Init_Date
1087 ch%=1: BLOCK#ch%,100,10,376,10,0 :REMark dmy%=DATE$
1088 yr%=dmy$(1 TO 4):month$=dmy$(6 TO 8):d%=dmy$(10 TO 11)
1089 FOR i=1 TO 12: IF month$=mth$(i):m%=:EXIT i
1090 OVER#ch%,1: CSIZE#ch%,1,0: INK#ch%,7:d$=DAY$(DATE)
1091 FOR i=0 TO 1: CURSOR#ch%,350+i,10:PRINT#ch%,d$,' ';d%,' ';month$,' ';yr%
1092 OVER#ch%,0: CSIZE#ch%,0,0
1093 END DEFine
```

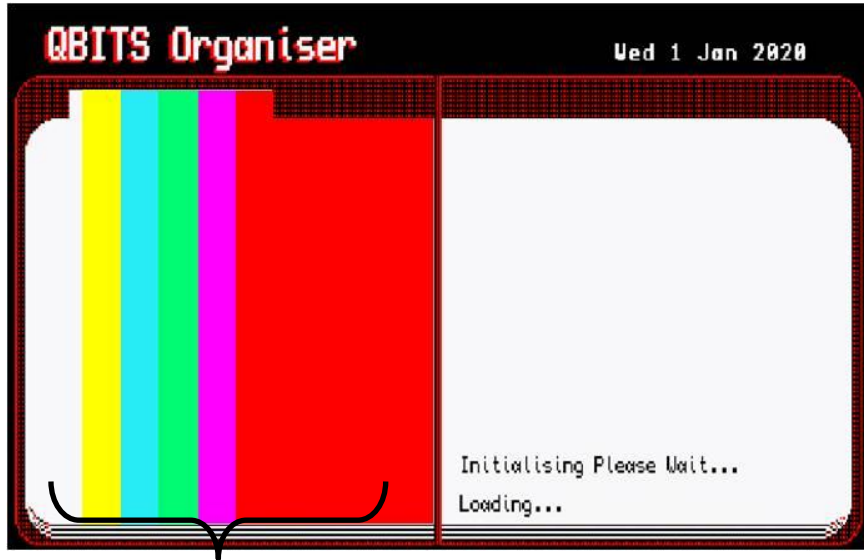
Init_Page - Start up Message Initialising Program and LOAD of DATA Files.

1095 DEFine PROCEDURE Init_Page

```
1096 ch%=1: INK#ch%,0: CURSOR#ch%,260,180:PRINT#ch%,'Initialising Please Wait...'
```

```
1097 y5%=yr%.y3%=yr%.m3%=m%.y2%=yr%.m2%=m%.y1%=yr%.m1%=m%:PAUSE 20:OrgDFile
```

```
1098 END DEFine
```



New addition the option to Set background colours [bc%] see page 31.
Sets/Clears the Double Page Title and Set Date for Page changes.

1100 DEFine PROCEDURE Page_Cls

```
1101 ch%=1: RESTORE 1105: FILL#ch%,1: INK#ch%,bc%
```

```
1102 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
```

```
1103 DATA 10,81,156,81,164,75,164,75,164,10,158,5 :REMark Page CIs
```

```
1104 DATA 158,5,10,5,3,10,3,10,3,75,10,81
```

```
1105 FILL#ch%,0:ch%=1: STRIP#ch%,bc%:INK#ch%,0
```

```
1106 BLOCK#ch%,118,15,34,30,bc%:BLOCK#ch%,124,12,348,29,16 :REMark Headings
```

```
1107 BLOCK#ch%,4,194,246,24,0,2,2: INK#ch%,0 :REMark Page Fold
```

```
1108 END DEFine
```

Prints common Page Prompts for Set Date, Edit Text Box, DATA Save etc.

1110 DEFine PROCEDURE Page_Set(h%,H\$)

```
1111 ch%=1:CSIZE#ch%,2,0:INK#ch%,0:STRIP#ch%,bc%:OVER#ch%,1
```

```
1112 FOR i=0 TO 1:CURSOR#ch%,40+i,32:PRINT#ch%,H$
```

```
1113 OVER#ch%,0 :CSIZE#ch%,0,0:STRIP#ch%,bc%:RESTORE 1117
```

```
1114 FOR i=1 TO h%: READ a,b,H$:CURSOR#ch%,a,b:PRINT#ch%,H$
```

```
1115 DATA 260,196,'(S)ave',260,46,'Contents' :IF h%>1:BLOCK#ch%,12,3,318,50,0
```

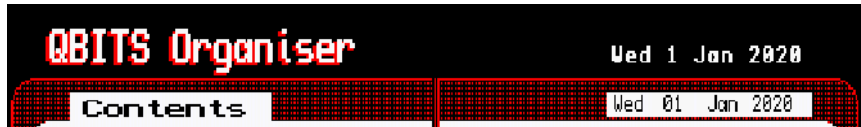
```
1116 DATA 380, 46,'Set Date+ →↑↓↵ ' :IF h%>2:BLOCK#ch%, 2,4,464,48,0
```

```
1117 DATA 360, 58,'Edit( )←↑↓↵ (Esc)' :IF h%>3:BLOCK#ch%, 2,4,438,60,0
```

```
1118 END DEFine
```

QBITS Organiser Date Setting

To change Organiser entry Dates, use <Cursor keys> (Day/Month become highlighted) then Set with the <Enter> key. Scrolling Up or Down through the Jan/Dec Months Selects other Years. <Left/Right> Cursor keys change the Day. A 30-day limit is set for April, June, September, November, February 28, or 29 in a leap year, the rest 31.



```

1120 DEFine PROCEDURE Set_Date(chk)
1121 ch%=1:stc%=7:CSIZE#ch%,0,0:STRIP#ch%,7:INK#ch%,0
1122 BLOCK#ch%,120,10,348,30,7: Cal_day yr%, m%, d%
1123 REPEAT date_ip
1124 IF chk>0
1125   stc%=5:Cal_day yr%,m%,d%
1126   IF m%=2 AND d%>28:d%=dm%(2)      Note: Array dm%(m%) month day lengths
1127   SELECT ON m%=4,6,9,11:IF d%>30:d%=30
1128 END IF
1129 Prn_Date: IF chk>0: k=CODE(INKEY$(-1)):ELSE RETURN      chk=0 Just displays Date
1130 SELECT ON k      chk>0 highlights day/month action
1131   =216 :m%=m%-1:IF m%< 1:yr%=yr%-1:m%=12      Select month
1132   =208 :m%=m%+1:IF m%>12:yr%=yr%+1:m%=1
1133   =200 :IF d%<31:d%=d%+1      Select day
1134   =192 :IF d%> 1:d%=d%-1
1135   = 10 :stc%=7:chk=0:Prn_Date:EXIT date_ip
1136 END SELECT
1137 END REPEAT date_ip
1138 END DEFine

```



```

1140 DEFine PROCEDURE Prn_Date
1141 CURSOR#ch%,351,30:PRINT#ch%,d$
1142 STRIP#ch%,stc%:CURSOR#ch%,374,30:PRINT#ch%, ' ';FILL$(0',2-LEN(d%))&d%,' '
1143 CURSOR#ch%,400,30:PRINT#ch%, ' ';mth$(m%);' '
1144 STRIP#ch%,7:CURSOR#ch%,432,30:PRINT#ch%,yr%
1145 END DEFine

```

To display the day of the week Sun/Mon/Tue/Wed/Thu/Fir/Sat, the Key Value method is used with code values added for different Months, and Years of different Centuries.

```

1147 DEFine PROCEDURE Cal_day(yr%, m%, d%)
1148 IF yr% MOD 4=0:dm%(2)=29:ELSE dm%(2)=28      :REMark Set Feb 28/29 days
1149 y$=yr%-yd%=y$(3 TO 4):year=yr%      :REMark yd% Decade key value
1150 SELECT ON year
1151   =1700 TO 1799: yc%=4      :REMark yc% Century key value
1152   =1800 TO 1899: yc%=2
1153   =1900 TO 1999: yc%=0
1154   =2000 TO 2099: yc%=6
1155 END SELECT
1156 RESTORE 1157: FOR i=1 TO 12: READ a :IF m%=i:mk%=a:EXIT i
1157 DATA 1,4,4,0,2,5,0,3,6,1,4,6      :REMark mk% Month key
1158 IF dm%(2)=29:IF m%=1 OR m%=2:mk%=mk%-1:END IF      :REMark mk% Leap Year
1159 wd%=(yd% DIV 4)+d%+mk%+yc%+yd%) MOD 7:IF wd%=0:wd%=7:REMark wd% Week Day
1160 END DEFine

```

QBITS Organiser Contents

The **QBITS Organiser** opening double page displays **Contents** on the left-hand Page. Use the up/down cursors keys to move the Contents indicator and action with Spacebar or alternatively press keys **A B C D M P S Q** for direct access.

As an **Organiser** is all about creating prompts and reminders it was thought apt that any scheduled **Appointments**, meetings, should be a top-level priority. The right-hand page displays three groups of information, **Appointments**, **NotePad** entries and an **Event** reminder. The last derived from the **Planner** Page entry.



Appointments	Select Date	- Edit 1- 6 Entries + NotePad + Events reminder
Banking	Select Date	- Transactions Remark/Sum /Balance + Calculator
Calendar	Select Year	- Display week days / months of Year
Diary	Select Day	- Daily entries/Journal/Reminders//Weather
Mail List	Select Name	- First Name(s)/Mobile/Home/Office Phone email/Address/District/Post Zip code/Country
Planner	Select Day	- Event (Colour code) / Change Event Name
Settings	Select Year	- Load DATA / Change Page Background Colour
Quit	Exit	- Close Channels End QBITS Organiser Program

For each Page most actions are triggered by direct use of the cursor keys (such as Date settings) or activating other actions by keying the letter of a (**H**)eading, returning via the <Esc> key. To return to Contents Page press <Spacebar>.

The (S)ave option at the bottom of page allows updates to the current DATA File. The DATA is stored monthly for **Appointments**, **Banking** and **Diary** Page entries. If the **Set_Date** is changed to a new month/year **QBITS Organiser** will save the current DATA and attempt to open one for the new **Set_Date**.

```

1162 DEFine PROCEDURE Org_Contents
1163 REMark Organiser Main Menu [Select ↑↓ — or ABCDMPQ]
1164 ch%=1: Init_Contents: Set_Date 0: Appt: Events
1165 REPEAT Menu_lp
1166   Org_Ptr 1:k=CODE(INKEY$(-1)):IF k=32:k=ct%
1167   SElect ON k
1168     =65, 97      :ct%=65:Org_Ptr 2      :Appt: Events: Appointments
1169     =66, 98      :ct%=66:bc%=bgc%(2) :Banking
1170     =67, 99      :ct%=67:bc%=bgc%(3) :Calendar
1171     =68,100      :ct%=68:bc%=bgc%(4) :Diary
1172     =69,77,109   :ct%=69:bc%=bgc%(5) :Mail
1173     =70,80,112   :ct%=70:bc%=bgc%(6) :Planner
1174     =71,83,115   :ct%=71:Org_Ptr 2      :OrgSetUp:chk=1:Appt:Events
1175     =72,81,113:INK#0,7:STRIP#0,0      :Org_Quit
1176     =208:IF ct%>65:ct%=ct%-1
1177     =216:IF ct%<72:ct%=ct%+1
1178   END SElect
1179 END REPEAT Menu_lp
1180 END DEFine

1182 DEFine PROCEDURE Org_Quit
1183 FOR i=3 TO 9: CLOSE#: END FOR i:bc%=0:CLS#2:PRINT#0,'bye...':STOP      :REMark NEW
1184 END DEFine

```

Note: The Pointer for Selecting QBITS Organiser Pages is designed to point in either direction. Two of the Contents Pages Appointments and Settings use only the right-hand Page. See Pages 11 & 30.

```

1186 DEFine PROCEDURE Org_Ptr(chk)
1187 OVER#ch%,1: BLOCK#ch%,24,156,202,50,bc%:INK#ch%,0
1188 IF chk=1:FOR i=0 TO 3:CURSOR#ch%,200+i,32+(ct%-64)*20:PRINT#ch%, '<' Left
1189 IF chk=2:FOR i=0 TO 3:CURSOR#ch%,216+i,32+(ct%-64)*20:PRINT#ch%, '>' Right
1190 OVER#ch%,0: BLOCK#ch%,10,3,208,35+(ct%-64)*20,0
1191 END DEFine

```

```

1193 DEFine PROCEDURE Init_Contents
1194 ch%=1:bc%=bgc%(1):Page_Cls:Page_Set 4,'Contents':RESTORE 1203
1195 CSIZE#ch%,2,0: INK#ch%,0: STRIP#ch%,bc%:OVER#ch%,1
1196 FOR a=1 TO 8
1197   BLOCK#ch%,160,14,32,30+a*20,248: BLOCK#ch%,156,12,34,31+a*20,7
1198   READ C$:FOR b=0 TO 1:CURSOR#ch%,40+b,32+a*20:PRINT#ch%,C$
1199 END FOR a
1200 OVER#ch%,0: CSIZE#ch%,0,0
1201 DATA 'Appointments', 'Banking', 'Calendar', 'Diary'
1202 DATA 'Mail List', 'Planner', 'Settings', 'Quit'
1203 END DEFine

```

QBITS Text Editor Review

Appointments Banking Diary Mail List and **Planner** use differing Character Sets or String handling Formats. To meet this requirement, I considered the possibility of translating Editors from other sources, however after a number of frustrating attempts and a spiralling code lists, a rather more pragmatic approach prevailed. What was the minimum acceptable requirement in functionality, but still able to get the basic 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 how to introduce a **Wrap** function for multiple rows, and add a **New Line** while still making sure the character string remained within a defined screen area.

Another thought, generally 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** Deletes character at cursor position. **F3** Deletes to End of Line (opposite to a New Line) and **F4** Deletes 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 it 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 to multiple row fields and if string is longer than transferred field space, the string is truncated.

To identify screen position within a current string an **underline** style cursor was chosen. This is 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 TextED Character String

In SuperBASIC an input of character codes grouped as a string variable is suffixed with '\$'. Therefore, the first step is **Adding** 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 TextED 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 **sc%** and rows **sr%** and maximum string length **sm%**, which set boundaries within which string manipulation can take place.

QBITS TextED 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 (checking that **sc%,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 TextED 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 TextED 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 **cx%, cy%**. The variable **cp%** now doubles as string position and for calculating cursor pixel position and for checking against **str\$** boundaries: 0, **sc%**, **sr%**, **sl%** and **sm%**.

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

QBITS TextED

1205 REMARK **QBITS Text Editor** Set channel, chr mode, col, row, x, y, chr string

```
1207 DEFINE PROCEDURE TextEd(ch%,cm%,sc%,sr%,sx%,sy%,str$)
1208 sl%=LEN(str$):IF sl%>0:cp%=sl%:ELSE cp%=1:sl%=0
1209 sm%=sc%*sr%:cr%=0:cx%=sx%:cy%=sy%: STRIP#ch%,bc%
1210 REPEAT Edit_lp
1211 Str_Prn str$:Str_Cur:k=CODE(INKEY$(-1))
1212 SElect ON k
1213 = 10 :Add_crt :REMark New Line
1214 = 27:Str_Prn str$:RETurn :REMark End Edit
1215 = 32 TO 125:k$="":Chk_chr:IF k$>" :Add_chr :REMark Character Set
1216 =194,232:IF cp%>1:cp%=cp%-1 :Del_chr :REMark F1 Delete Left
1217 =202,236 :Del_chr :REMark F2 Delete Above CURSOR
1218 =240 :Del_eol :REMark F3 Delete End of Line
1219 =244 :Del_pge :REMark F4 Delete Page
1220 =248 :Str_Restore :REMark F5 Restore Ppage
1221 =192:IF cp%>1:cp%=cp%-1 :REMark Move Left
1222 =200:IF cp%<=sl%:cp%=cp%+1 :REMark Move Right
1223 =208:IF cp%>sc%:cp%=cp%-sc% :REMark Move Up
1224 =216:IF cp%<sl%-sc%:cp%=cp%+sc%:ELSE cp%=sl% :REMark Move Down
1225 END SElect
1226 END REPEAT Edit_lp
1227 END DEFINE
```

```
1229 DEFINE PROCEDURE Chk_chr Controls the Character Sets
1230 IF cm%=0 :SElect ON k=32 TO 127 :k$=CHR$(k):RETurn
1231 IF cm%=1 OR cm%>2 :SElect ON k=32,43,45,46,48 TO 57 :k$=CHR$(k)
1232 IF cm%>1 :SElect ON k=65 TO 90 :k$=CHR$(k)
1233 IF cm%=2 OR cm%=3 :SElect ON k=97 TO 122 :k$=CHR$(k-32)
1234 IF cm%>3 :SElect ON k=97 TO 122 :k$=CHR$(k)
1235 IF cm%=4 :SElect ON k=64 :k$=CHR$(k)
1236 IF cm%=5 :SElect ON k=42,47,61 :k$=CHR$(k)
1237 END DEFINE
```

```
cm%=0 32-127 Full ASCII printable Characters ' ' <space> to Copywrite © <Shift Esc>
cm%=1 OR >2 32 [ ], 43 [+], 45 [-], 46 [.] , 48-57 [0123456789]
cm%>1 65-90 [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
cm%=2 OR 3 97-122 [abcdefghijklmnopqrstuvwxyz] -32 change to [ABC-XYZ]
cm%>3 97-122 [abcdefghijklmnopqrstuvwxyz]
cm%=4 64 [@]
cm%=5 42 [*], 47 [/], 61 [=]
```

```
1239 DEFINE PROCEDURE Add_chr
1240 IF cp%= 1 AND sl%=0 :str$=str$&k$
1241 IF cp%>=1 AND cp%<sl%:str$=str$(1 TO cp%-1)&k$&str$(cp% TO sl%)
1242 IF cp%>=1 AND cp%=sl%:str$=str$(1 TO cp%-1)&k$&str$(cp%)
1243 IF cp%> 1 AND cp%>sl%:str$=str$&k$
1244 IF cp%=sm%:str$(cp%)=k$
1245 IF sl% <sm% :sl% =sl%+1:ELSE sl%=sm%
1246 IF cp%<sm%:cp%=cp%+1:ELSE cp%=sm%
1247 END DEFINE
```



```

1249 DEFine PROCEDURE Del_chr
1250 IF cp% = sl%:str$=str$(1 TO sl%-1):sl%=sl%-1
1251 IF cp% >= 1 AND cp% < sl%:str$=str$(1 TO cp%-1)&str$(cp%+1 TO sl%):sl%=sl%-1
1252 IF cp% = sm%:str$=str$(1 TO sm%-1):cp% = cp%-1:sl% = sm%-1
1253 IF cp% = 1 AND sl% = 1:str$="" :sl%=0
1254 END DEFine

1256 DEFine PROCEDURE Del_eol
1257 IF cp% >= sl%:RETURN
1258 buf$=str$:cn%=1+sc%-cp% MOD sc%:IF cn% > sl%-cp%:cn%=sl%-cp%
1259 str$=FILL$(' ',sl%):Str_Prn str$
1260 str$=buf$(1 TO cp%-1)&buf$(cp%+cn% TO LEN(buf$)):sl%=LEN(str$)
1261 END DEFine

1263 DEFine PROCEDURE Del_pge
1264 buf$=str$:str$="" :sl%=0:cp%=1:BLOCK#ch%,sc%*6,sr%*10,sx%,sy%,7
1265 END DEFine

1267 DEFine PROCEDURE Str_Restore
1268 sl%=LEN(str$):IF sl% > sm%:str$=buf$(1 TO sm%):sl%=sm%:ELSE str$=buf$
1269 END DEFine

1271 DEFine PROCEDURE Add_crt
1272 IF sl%=0:RETURN :ELSE cn%=1+sc%-cp% MOD sc%
1273 IF sl% DIV sc% < sr%-1 OR sl%+cn% < sm%
1274   IF cp% <= sl%
1275     str$=str$(1 TO cp%-1)&FILL$(' ',cn%)&str$(cp% TO sl%)
1276     cp%=cp%+cn%:sl%=LEN(str$)
1277   END IF
1278   IF cp% > sl%:str$=str$(1 TO cp%)&FILL$(' ',cn%):cp%=cp%+cn%:sl%=cp%:END IF
1279 END IF
1280 END DEFine

1282 DEFine PROCEDURE Str_Prn(str$)
1283 STRIP#ch%,7:sm%=sc%*sr%:sl%=LEN(str$):IF sl% > sm%:sl%=sm%
1284 cr%=sl% DIV sc%:IF cr% = sr%:cr% = sr%-1
1285 FOR r=0 TO cr%
1286   IF sl% > 0 AND sl% MOD sc% >= 0
1287     CURSOR#ch%,sx%,sy%+r*10:PRINT#ch%,str$(1+sc%*r TO sc%+sc%*r)
1288   END IF
1289 END FOR r
1290 IF sl%=0:CURSOR#ch%,sx%,sy%: PRINT#ch%, ' ':END IF
1291 END DEFine

1293 DEFine PROCEDURE Str_Cur
1294 BLOCK#ch%,6,1,cx%,cy%+9,7
1295 IF cp% MOD sc%=0:cx%=sx%+sc%*6-6 :cy%=sy%+cp% DIV sc%*10-10
1296 IF cp% MOD sc% > 0:cx%=sx%+cp% MOD sc%*6-6:cy%=sy%+cp% DIV sc%*10
1297 IF cp%=1 :cx%=sx% :cy%=sy%
1298 IF cp% >= sm% :cx%=sx%+sc%*6-6 :cy%=sy%+sr%*10-10
1299 BLOCK#ch%,6,1,cx%,cy%+9,2
1300 END DEFine

```

QBITS Calculator

The Editor used by the Calculator is for finances, working with the chosen currencies Dollar, Euro, Pound and Yen, requires values to two decimal places. The Arithmetic functions include Add [+], Subtract [-], Multiply [*], Divide [/]. Percentage of a number, [*] times a value of [%] to give a result. Other keys, Equal sign [=], Decimal Point [.] , [N]egate +/-number, [C]lear all, [←] Delete last character.

1302 REMark **QBITS Calculator** Currency NN.nn Dollar Euro Pound Yen

```

1304 DEFine PROCEDURE Calculator(Num$)
1305 REPeat Calc_ip
1306 INK#6,2:Calc_Key c%,r%:PRINT#7,FILL$(' ',9-LEN(Num$))&Num$
1307 IF tchk=1:CURSOR#1,sx%,sy%:PRINT#1,FILL$(' ',9-LEN(Num$))&Num$
1308 k=CODE(INKEY$(1)):INK#6,7:Calc_Key c%,r%:IF k=10:k=CODE(Key$(c%,r%))
1309 SElect ON k
1310 =27:INK#6,7      : Calc_Key c%,r%:ch%=1:Calc_Dec:EXIT Calc_ip
1311 =194             : Calc_BkSp           :REMark [»] Delete
1312 =67,99          : Calc_Cls            :REMark [C]lear
1313 =78,110          : Calc_Negate         :REMark [N]egate
1314 =37,42,43,45,47,61: Calc_Func         :REMark [%/*+.=] Function
1315 =46:IF dot=0     : Calc_Dot:chk=0:dot=1:REMark [.] Decimal Point
1316 =48 TO 57       : Calc_Add           :REMark [0 to 9]
1317 =192:c%=c%-1 : IF c%<1:c%=4         :REMark Key column
1318 =200:c%=c%+1 : IF c%>4:c%=1
1319 =216:r% =r%+1 : IF r%>5:r%=1         :REMark Key row
1320 =208:r% =r%-1 : IF r%<1:r%=5
1321 END SElect
1322 END REPeat Calc_ip
1323 IF Num$>0:km=32: Calc_Dec
1324 END DEFine

```



Note: The **REPeat Calc_ip** checks for Numeric Input or Maths Function, keyed directly or via highlighted key using <Cursor keys> and <Enter>.

```

1326 DEFine PROCEDURE Calc_Key(c%,r%)      Updates the highlighted Key on KeyPad
1327 ch%=6: CURSOR#ch%,-20+c%*29,3+r%*22:PRINT#ch%,Key$(c%,r%)
1338 END DEFine

```

```

1330 DEFine PROCEDURE Calc_BkSp              Deletes last Character
1331 Num$=Num$(1 TO LEN(Num$)-1)
1332 IF LEN(Num$)=0:Num$='0':chk=1:dot=0:slen=5
1333 IF LEN(Num$)=1 AND '.' INSTR Num$=1:Num$='0':chk=1:dot=0:slen=5
1334 IF '.' INSTR Num$=0:dot=0:slen=5
1335 IF '.' INSTR Num$>1:slen=Num$-LEN(INT(Num$))+2
1336 END DEFine

```

```

1338 DEFine PROCEDURE Calc_Cls              Clears all numbers /screens
1339 CLS#5:CLS#7:Num$='0':num=0:dot=0:slen=5
1340 END DEFine

```

```

1342 DEFINE PROCEDURE Calc_Negate                                Toggles +/- value of number
1343 IF Num$=0: RETURN
1344 IF Num$(1)='-':Num$=Num$(2 TO LEN(Num$)):ELSE Num$='-'&Num$
1345 END DEFINE

```

```

1347 DEFINE PROCEDURE Calc_Dot                                  Adds the decimal point
1348 IF chk=1:Num$=0.:slen=4:RETURN
1349 IF Num$-INT(Num$)=0:Num$=Num$&'.'.slen=LEN(Num$)+2
1350 END DEFINE

```

```

1352 DEFINE PROCEDURE Calc_Add                                  Adds a numerical value
1353 IF chk=1:chk=0:Num$="&CHR$(k):ELSE IF LEN(Num$)<slen:Num$=Num$&CHR$(k)
1354 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.

```

1356 DEFINE PROCEDURE Calc_Err
1357 IF Num$>99999.89 OR Num$<-99999.89: Num$='!': ELSE num=Num$
1358 END DEFINE

```

```

1360 DEFINE PROCEDURE Calc_Dec
1361 IF Num$='!':PRINT#5,'= Overflow!':Num$=0:num=0:RETURN
1362 IF '-' INSTR Num$=0 AND Num$<.1:Num$=0':RETURN :REMark QL (1E6 etc)
1363 IF Num$-INT(Num$)=0:dec$=''.00':ELSE dec$=Num$('.' INSTR Num$ TO LEN(Num$))
1364 IF LEN(dec$)<3:dec$=dec$&'0'
1365 Num$=INT(Num$)&dec$(1 TO 3):PRINT#5,CHR$(km);FILL$(' ',10-LEN(Num$))&Num$
1366 END DEFINE

```

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

```

1368 DEFINE PROCEDURE Calc_Func
1369 IF km=47 AND Num$=0:chk=1: RETURN
1370 IF num=0:km=32:Calc_Err:Calc_Dec:chk=1:dot=0:slen=5:num=Num$:km=k:RETURN
1371 IF k=37:km=37:k=61 :REMark '%' Percentages
1372 IF km<>61: Calc_Dec Num$:chk=1 :REMark '/' * + - Functions
1373 SELECT ON km
1374 =37: Num$=num*Num$/100 :REMark '%' 'n1*n2/100'
1375 =47: Num$=num/Num$ :REMark '/' 'n1/n2'
1376 =42: Num$=num*Num$ :REMark '*' 'n1*n2'
1377 =43: Num$=num+Num$ :REMark '+' 'n1+n2'
1378 =45: Num$=num-Num$ :REMark '-' 'n1-n2'
1379 END SELECT
1380 Calc_Err:km=61: Calc_Dec:km=k:chk=1 :REMark Results
1381 END DEFINE

```

```

1383 DEFINE PROCEDURE Currency(C$)
1384 ch%=1: CSIZE#ch%,2,1: INK#ch%,0: STRIP#ch%,bc%
1385 CURSOR#ch%,16,52:PRINT#ch%,C$:CSIZE#ch%,0,0
1386 IF C$='C': BLOCK#ch%, 9 ,1,16,60,0: BLOCK#ch%,8,1,16,62,0
1387 IF C$='Y': BLOCK#ch%,10,1,18,61,0: BLOCK#ch%,10,1,18,63,0
1388 END DEFINE

```

£ € ¥

Note: This is the code for Banking Pages layout, which includes the Calculator Keypad.

```

1390 DEFINE PROCEDURE Init_Trans
1391 ch%=1:RESTORE 1395:OVER#ch%,1
1392 FOR i=0 TO 8:READ a,b,H$:CURSOR#ch%,a,b:PRINT#ch%,H$ (H)eadings & keys
1393 DATA 160,56,'(C)alculator',260,80,'(T)ransaction',394,80,'-/+',450,80,'Bal'
1394 DATA 32,196,'Currency Dollar Euro Pound Yen'
1395 DATA 93,196,'D',135,196,'E',165,196,'P',201,196,'Y'
1396 OVER#ch%,0
1397 ch%=5:PAPER#ch%,7:BORDER#ch%,1,248:INK#ch%,0:CLS#ch%:CSIZE#ch%,0,0
1398 ch%=6:PAPER#ch%,248:INK#ch%,7:STRIP#ch%,0:CLS#ch%:CSIZE#ch%,2,1
1399 FOR kr=1 TO 5
1400 FOR kc=1 TO 4
1401 BLOCK#ch%,24,20,-26+kc*29,2+kr*22,0
1402 CURSOR#ch%,-20+kc*29,3+kr*22:PRINT#ch%,Key$(kc,kr) Keypad Display
1403 END FOR kc
1404 END FOR kr
1405 ch%=7:PAPER#ch%,7:BORDER#ch%,2,0:INK#ch%,0:CLS#ch%:CSIZE#ch%,2,1
1406 c%=2:r%=3:k=0:km=0:Num$='0':num=0:dot=0:chk=1:slen=5:tc%=1:tr%=1
1407 END DEFINE

```

QBITS Organiser Appointments



QBITS Organiser Appointments

Select **Appointments** from the **Contents** Page by use of <Up/Down> cursor keys and <Spacebar> or simply press the (A) key. To change the **Set_Date** use the four <Cursor keys> then the <Enter> key to Set and return. Any **Appointments** or **NotePad** entries previously created for the **Set_Date** are displayed.

To Edit Appointments, press characters (1 – 6) or (N) for NotePad. When your field edit is complete exit using the <Esc> key.

Use (S)ave to update the Organiser DATA Files.

1409 REMark **Organiser Appointments** Apppts/Events/NotePad

1411 DEFine PROCEDURE Appointments

1412 REPEAT Apts_lp

1413 k=CODE(INKEY\$(-1))

1414 SELECT ON k

1415 =32 :OrgSave 1,'Org'&yr%&'_A_'&mth\$(m%):EXIT Apts_lp

1416 =83,115:OrgSave 1,'Org'&yr%&'_A_'&mth\$(m%)

1417 =49 TO 54:n%=k-48:TextEd 1,0,32,1,284,74+n%*10,Apt\$(d%,n%)

1418 =78,110 :TextEd 1,0,36,2,260,166,Note\$(d%)

1419 =192,200,208,216 :y1%=yr%:m1%=m%: Set_Date 1:Appt:Events

1420 END SELECT

1421 END REPEAT Apts_lp

1422 END DEFine

1424 DEFine PROCEDURE Appt

1425 oy%=y1%:om%=m1%:OrgDChk 1

1426 BLOCK#ch%,216,264,260,83,bc%:BLOCK#ch%,216,60,262,84,7 :REMark Appt Cls/Prt

1427 STRIP#ch%,bc%:CURSOR#ch%,260,70:PRINT#ch%,'Appointments'

1428 CURSOR#ch%,420,150:PRINT#ch%,'(N)otePad':STRIP#ch%,7

1429 FOR i=1 TO 6:CURSOR#ch%,260,74+i*10:PRINT#ch%,(';i:') :Apt\$(d%,i)

1430 BLOCK#ch%,218,22,259,165,248:BLOCK#ch%,216,20,260,166,7 :REMark Note Cls/Prt

1431 sc%=36:sr%=2:sx%=260:sy%=166:Str_Prn Note\$(d%)

1432 END DEFine

1434 DEFine PROCEDURE Events

1435 BLOCK#ch%,96,10,278,150,bc%:STRIP#ch%,bc% :REMark Event Cls/Prt

1436 CURSOR#ch%,280,150:PRINT#ch%,Event\$(pln%(m%,d%))

1437 pcol=col%(pln%(m%,d%)):IF pcol=0:pcol=7 :REMark Colour Code

1438 BLOCK#ch%,12,8,265,150,0:BLOCK#ch%,10,6,266,151,pcol

1439 END DEFine

QBITS Organiser Banking

One of the aspirations in the eighties was for the QBITS Organiser Banking page to be some sort of a spreadsheet with columns for currency. The first problem was growing number values from right to left. Fixed by use of FILL\$ and LEN (Str\$):-

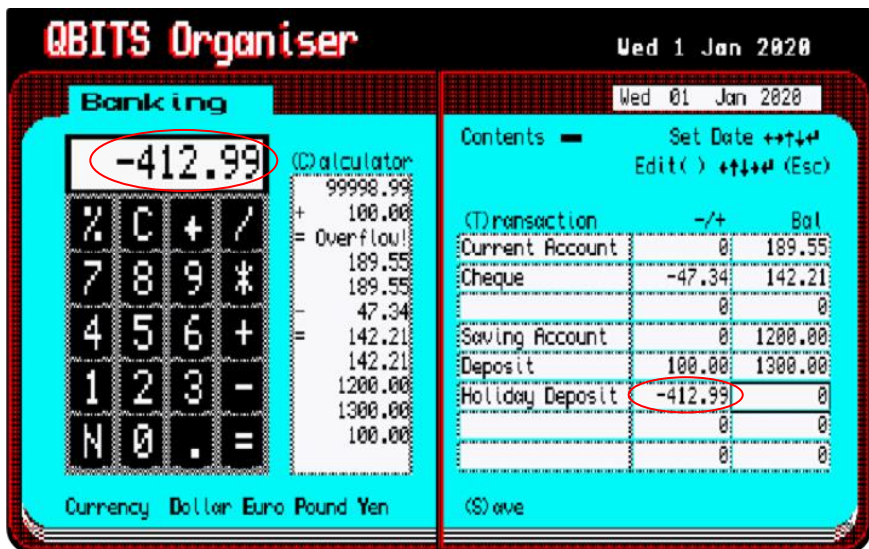
PRINT FILL\$(' ',n-LEN (Num\$))&Num\$ where n=max field spaces of entry.

Formatting to two decimal points required checking if a decimal point already exists within the Character String (Num\$) and then to what number of decimal places. The end resulting in a number with two decimal places:-

```
IF Num$-INT(Num$)=0:dec$='.00':ELSE dec$=Num$('.' INSTR Num$ TO LEN(Num$))
IF LEN(dec$)<3:dec$=dec$&'0'
Num$=INT(Num$)&dec$(1 TO 3):PRINT FILL$( ' ',n-LEN(Num$))&Num$
```

QBITS Banking Layout

The Banking page is set out with Transactions as a mini-spreadsheet on the right and a graphic representation of a Calculator on the left to be used as a currency ready reckoner. Set a different Date with <Cursor keys> & <Enter key>. Select [D][E][P][Y] to display one of the International Currency Symbols.



To **Edit** a Transactions press (T). The top left cell or last currently used will be highlighted. Navigate around the various cells with <Cursor keys> followed by the <Enter> key. The left Column uses the QBITS TextED allowing alphanumeric entries. For the two currency columns, the **Calculator Editor** is used.

For independent use of the Calculator press (C) and carry out computations again using <Esc> to return. Choose the (S)ave option to update any Edits to the DATA File for Banking Transactions.

1443 **DEFine PROCEDURE Banking**

1444 ch%=1:Page_Cls:Page_Set 4,'Banking':Init_Trans:Set_Date 0

1445 tchk=0:tl=0:oy%=y2%:om%=m2%:OrgDChk 2

1446 **REPEAT Bank_Ip**1447 **FOR** b=1 **TO** 8:**FOR** a=1 **TO** 3:bin=248:OrgTrans a,b: **END FOR** a:**END FOR** b

1448 k=CODE(INKEY\$(-1))

1449 **SElect ON** k

1450 =32 :OrgSave 2,'Org'&yr%&'_B_'&mth\$(m%):Org_Contents

1451 =83,115 :OrgSave 2,'Org'&yr%&'_B_'&mth\$(m%) :REMark (S)ave

1452 =68,100 :Currency '\$' :REMark (D)ollar

1453 =69,101 :Currency 'C' :REMark (E)uro

1454 =80,112 :Currency 'P' :REMark (P)ound

1455 =89,121 :Currency 'Y' :REMark (Y)en

1456 =84,116 :Transactions :REMark (T)ransactions

1457 =67, 99 :Calculator '0' :REMark (C)alculator

1458 =192,200,208,216:y2%=yr%:m2%=m%:Set_Date 1:oy%=y2%:om%=m2%:OrgDChk 2

1459 **END SElect**1460 **END REPEAT Bank_Ip**1461 **END DEFine**1463 **DEFine PROCEDURE Transactions**1464 **REPEAT Acc_Ip**

1465 bin=0:OrgTrans tc%,tr%:k=CODE(INKEY\$(-1)):bin=248:OrgTrans tc%,tr%

1466 **SElect ON** k

1467 =27 :EXIT Acc_Ip

1468 =10 :IF tc%=1:bin=0:TextEd 1,cm%,sc%,1,sx%,sy%,Tran\$(d%,tr%,tc%)

1469 IF tc%>1:tchk=1:Calculator Tran\$(d%,tr%,tc%):tchk=0

1470 =192:IF tc%>1:tc%=tc%-1

1471 =200:IF tc%<3:tc%=tc%+1

1472 =208:IF tr%>1:tr%=tr%-1

1473 =216:IF tr%<8:tr%=tr%+1

1474 **END SElect**1475 **END REPEAT Acc_Ip**1476 **END DEFine**

The **PROCEDURE OrgTrans** identifies the coordinates **sx%**,**sy%** from the field's column **tc%** and row **tr%**, and string columns **sc%** to define string width **sw%**. Character mode **cm%** is an attribute required for **TextEd**. The cell outlines are produced and any **Set_Date** entries Printed to screen. To exit either editors press <Esc>. Select another cell or press <Esc> a second time to exit the spreadsheet. Now you can either Set another Date or return to the **Contents** Page by pressing <Spacebar>.

1478 **DEFine PROCEDURE OrgTrans(tc%,tr%)**

1479 IF tc%=1 tl%=0:sc%=16:sw%=98:sx%=260:cm%=0 :REMark TextED

1480 IFtc%>1: tl%=9-LEN(Tran\$(d%,tr%,tc%)):sc%=9:sw%=56:sx%=246+tc%*57:cm%=1

1481 If tc%>1 AND tl%=9: Tran\$(d%,tr%,tc%)='0' :REMark Calc Editor

1482 sy%=79+tr%**12:B LOCK#ch%,sw%,10,sx%,sy%,7

1483 BLOCK#ch%,sw%+1,1,sx%-1,sy%-1,bin:BLOCK#ch%,sw%+1,1,sx%-1,sy%+11,bin

1484 BLOCK#ch%, 1,10,sx%-2,sy%,bin :BLOCK#ch%, 1,10,sx%+sw%-1,sy%,bin

1485 STRIP#ch%,7: CURSOR#ch%,sx%,sy%:PRINT#ch%,FILL\$(' ',tl%)&Tran\$(d%,tr%,tc%)

1486 **END DEFine**

QBITS Organiser Diary

Designs for the Diary began with a full double page; however, the layout is now left as separate pages with a **(J)**ournal page for reporting the day's events on the left and a **(R)**eminder page on the right. The **BLOCK** command is used to create bordered screen areas where Text is to be entered and displayed. Each is given a **(H)**eading with the bracketed character being the key to activate **QBITS TextED** for entering information.

To change Date Settings, see details on page 9.

1488 REMark **Organiser Diary Page** Daily Journal/Reminder/Weather/Events

1490 DEFine PROCEDURE Diary

```
1491 ch%=1:Page_Cls:Page_Set 4,'Diary':Init_Diary:Set_Date 0
1492 oy%=y3%:om%=m3%:OrgDChk 3:h%=0:l%=0:w%=0:Diary_Ent:Events
1493 REPEAT Diary_lp
1494 STRIP#ch%,bc%:Wth_Set:k=CODE(INKEY$(-1))
1495 SElect ON k
1496 =32 :OrgSave 3,'Org'&yr%&'_D_'&mth$(m%):Org_Contents
1497 =83,115:OrgSave 3,'Org'&yr%&'_D_'&mth$(m%)
1498 =74,106:TextEd 1,0,36,10,21,58,Jrnl$(d%) :REMark (J)ournal
1499 =82,114:TextEd 1,0,36,6,260,82,Remd$(d%) :REMark (R)eminders
1500 = 72 :IF h%<40:h%=h%+1 :Wth%(d%,1)=h% :REMark +(H)igh Temp
1501 =104 :IF h%>20:h%=h%-1 :Wth%(d%,1)=h% :REMark -(h)igh Temp
1502 = 76 :IF l%<40:l%=l%+1 :Wth%(d%,2)=l% :REMark +(L)ow Temp
1503 =108 :IF l%>20:l%=l%-1 :Wth%(d%,2)=l% :REMark -(l)ow Temp
1504 = 87 :IF w%<200:w%=w%+1:Wth%(d%,3)=w% :REMark + Wind (F)actor
1505 =119:IF w%> 1:w%=w%-1:Wth%(d%,3)=w% :REMark - Wind (f)actor
1506 =9:IF wn%>8:wn%=1:Wth%(d%,4)=wn: ELSE wn%=wn%+1:Wth%(d%,4)=wn%
1507 =192,200,208,216:y3%=yr%:m3%=m%:Set_Date 1:Diary_Ent:Events
1508 END SElect
1509 END REPEAT Diary_lp
1510 END DEFine
```

1512 DEFine PROCEDURE Diary_Ent

```
1513 oy%=y3%:om%=m3%:OrgDChk 3
1514 ch%=1:BLOCK#ch%,216,100,21,58,7:Events :REMark CLS Diary/Events
1515 sc%=36:sr%=12:sx%= 21:sy%=62:Str_Prn Jrnl$(d%) :REMark Journal entry
1516 sc%=36:sr%= 8:sx%=260:sy%=82:Str_Prn Remd$(d%) :REMark Reminder entry
1517 END DEFine
```

A new addition is the Weather Reporting using Graphics to enhance the screen display and minimise the DATA storage. The Graphics are a mixture of **Vector** and **Bitmap** designs. The Weather patterns range from a blank labelled Changeable, through Sunny, Cloudy, Windy, Showers, Overcast, Heavy Rain (Downpour), Stormy, to Snow. Edit Temp and Wind speed using upper & lowercase characters **(h/l/w)**.

1519 DEFine PROCEDURE Wth_Set

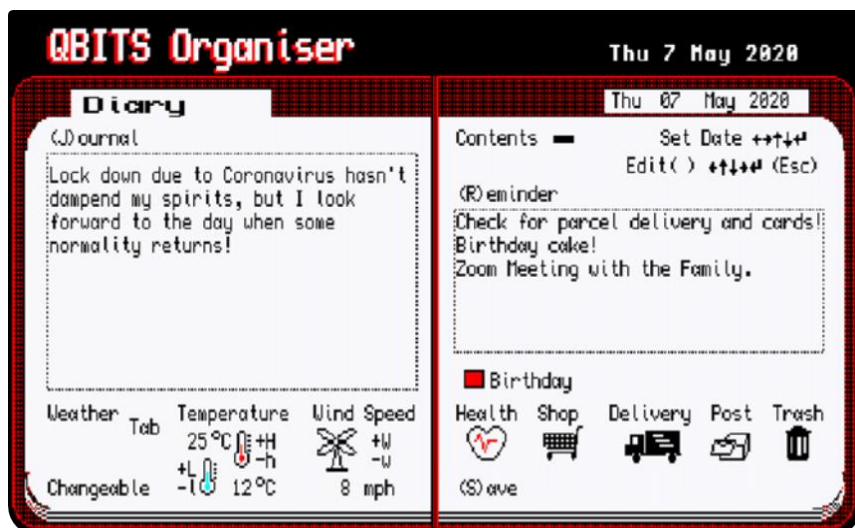
```
1520 h% =Wth%(d%,1): CURSOR#ch%,96,176:PRINT#ch%,FILL$(' ',3-LEN(h%))&h%
1521 l% =Wth%(d%,2): CURSOR#ch%,122,196:PRINT#ch%,FILL$(' ',3-LEN(l%))&l%
1522 w% =Wth%(d%,3): CURSOR#ch%,180,196:PRINT#ch%,FILL$(' ',3-LEN(w%))&w%
1523 wn%=Wth%(d%,4): Weather wn%
1524 END DEFine
```



```

1526 DEFine PROCEDURE Init_Diary
1527 ch%=1: RESTORE 1535
1528 BLOCK#ch%,218,102, 20,57,248: BLOCK#ch%,216,100, 21,58,7
1529 BLOCK#ch%,218, 62,259,81,248: BLOCK#ch%,216, 60,260,82,7
1530 FOR i=1 TO 16: READ x%,y%,H$: CURSOR#ch%,x%,y%:PRINT#ch%,H$
1531 DATA 20,46,'(J)ournal', 260,70,'(R)eminder',20,164,'Weather'
1532 DATA 68,170,'Tab',96,164,'Temperature',176,164,'Wind Speed'
1533 DATA 122,176,'C',142,176,'+H',142,184,'-h', 96,188,'+L',96,196,'-l'
1534 DATA 148,196,'C',210,176,'+W',210,184,'-w',206,196,'mph'
1535 DATA 260,164,'Health Shop Delivery Post Trash'
1536 CIRCLE#ch%,40,19,.7: CIRCLE#ch%,49,10,.7: Windmill 60,12
1537 Health 90,19,2: Shop 312,174: Goods 360,176: Post 138,14: Trash 454,174
1538 Therm 130,176,3: Therm 130,176,5
1539 END DEFine

```



Press the **Tab** key to scan through various Weather Patterns, choose one that most represents the day.



(See pages 34-36 for the Vector and Bitmap designs)

Below the **(R)**eminder Text box an Event box and Heading taken from the Planner Page. The right-hand page set of Graphics are to help prompt a response as well as an enhancement to the screen display.

QBITS Organiser Mail List.

1541 REMark **Organiser Mail List Page** Names/email/Mobile/Home/Office/Addresses

1543 DEFINE PROCEDURE Mail

1544 ch%=1:rs%=1:rn%=50:rf%=1:Page_Cls:Page_Set 2,'Mail List'

1545 Init_Mail: Rcd_Sel:rn%=1

1546 REPEAT Mail_Ip

1547 Rcd_Prn rn%:k=CODE(INKEY\$(-1))

1548 SElect ON k

1549 =192:rn%=rn% -1:IF rn%< 1:rn%=50 :REMark record num

1550 =200:rn%=rn%+1:IF rn%>50:rn%=1

1551 =208:IF rs%> 1:rs%=rs%-1 :REMark record select

1552 =216:IF rs%<26:rs%=rs%+1

1553 =9 :IF rf%=1:rf%=2:Rcd_Sel:ELSE rf%=1:Rcd_Sel :REMark Tab between last /First Names

1554 =10:rc%=1:Rcd_Find rn%

1555 =32 : OrgSave 4,'Org&yr%&'_ML':Org_Contents

1556 =83,115: OrgSave 4,'Org&yr%&'_ML'

1557 =69,101: TextEd 1,4,36,1, 20, 77,email\$(rn%) :REMark (e)mail

1558 =76,108: TextEd 1,2,25,1,260, 77,name\$(rn%,1) :REMark (L)ast Name

1559 =70,102: TextEd 1,0,25,1,260,102,name\$(rn%,2) :REMark (F)irst Name

1560 =65, 97: TextEd 1,0,32,3, 44,104,addr\$(rn%) :REMark (A)ddress

1561 =77,109: TextEd 1,1,15,1,260,142,mail\$(rn%,1) :REMark (M)obile

1562 =70,104: TextEd 1,1,15,1,260,156,mail\$(rn%,2) :REMark (H)ome

1563 =79,111: TextEd 1,1,15,1,260,170,mail\$(rn%,3) :REMark (O)ffice

1564 =80,112: TextEd 1,3,15,1,146,142,mail\$(rn%,4) :REMark (P)ost/Zip Code

1565 =68,100: TextEd 1,3,15,1,146,156,mail\$(rn%,5) :REMark (D)istrict

1566 =67, 99: TextEd 1,2,15,1,146,170,mail\$(rn%,6) :REMark (C)ountry

1567 END SElect

1568 END REPEAT Mail_Ip

1569 END DEFINE

Note: Search by Name, Last or First Names use **(Tab)**. Select an Alphabet character using the <Up/Down> Cursor keys, then press <Enter> key trigger Search for a match, if no records are found, '**Not Found!**' it will return. If there are more entries with chosen Alphabet letter, key <Enter> until the one is found. Alternatively, use the <Left/Right> Cursor keys to display the previous or the next sequential record.

Last Name or First Name use Tab to switch between.

1571 DEFINE PROCEDURE Rcd_Sel

1572 BLOCK#ch%,6,40,420,76,bc%

1573 STRIP#ch%,bc%:INK#ch%,0:CURSOR#ch%,420,54+24*rf%:PRINT#ch%,''' rf% record find

1574 END DEFINE

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

1576 DEFINE PROCEDURE Rcd_Find(rn)

1577 REPEAT rd_Ip

1578 STRIP#ch%,bc%:m%=rn%+1: IF rn%>rn%:rn%=1 m% record number : rn% max

1579 IF RS\$==name\$(m%,rf%,1) :RETURN :ELSE rc%=rc%+1 rc% records checked

1580 IF rc%>50: CURSOR#ch%,350,42+24*rf%:PRINT#ch%, 'Not Found!': EXIT rd_Ip

1581 END REPEAT rd_Ip

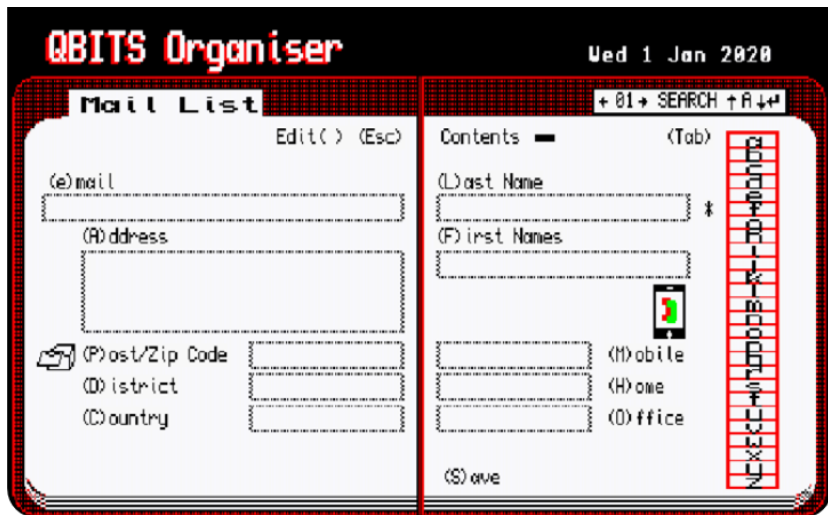
1582 PAUSE 50: BLOCK#ch%,64,10,348,64,bc%:BLOCK#ch%,64,10,348,90,bc%

1583 END DEFINE

```

1585 DEFINE PROCEDURE Rcd_Prn(m%) Prints Mail Field Entries to screen
1586 RS$=CHR$(rs%+64):STRIP#ch%,7:CURSOR#ch%,443,30:PRINT#ch%,RS$
1587 CURSOR#ch%,366,30:PRINT#ch%,FILLS('0',2-LEN(m%))&m%
1588 BLOCK#ch%,216,10, 20, 77,7 :CURSOR#ch%, 20, 77:PRINT#ch%,email$(m%)
1589 BLOCK#ch%,120,10,260, 77,7 :CURSOR#ch%,260, 77:PRINT#ch%,name$(m%,1)
1590 BLOCK#ch%,120,10,260,102,7 :CURSOR#ch%,260,102:PRINT#ch%,name$(m%,2)
1591 RESTORE 1601:BLOCK#ch%,192,30,44,104,7
1592 sc%=32:sr%=3:sx%=44:sy%=104:Str_Prn addr$(m%)
1593 FOR i=1 TO 6
1594 READ a,b:BLOCK#ch%,90,10,a,b,7:CURSOR#ch%,a,b:PRINT#ch%,mail$(m%,i)
1595 END FOR i
1596 DATA 260,142,260,156,260,170,146,142,146,156,146,170
1597 END DEFINE

```



```

1599 DEFINE PROCEDURE Init_Mail
1600 ch%=1: CSIZE#ch%,2,0: INK#ch%,0: OVER#ch%,1 :REMark 50Rcds
1601 FOR i=122 TO 97 STEP -1
1602 BLOCK#ch%,32,8,434,i*6-534,2:BLOCK#ch%,28,7,436,i*6-534,7
1603 FOR j=0 TO 0: CURSOR#ch%,444+j,i*6-534:PRINT#ch%,CHR$(j)
1604 END FOR i
1605 BLOCK#ch%,32,2,434,97*6-534,2: OVER#ch%,0: CSIZE#ch%,0,0: RESTORE 1613
1606 FOR I=1 TO 10: READ a,b,x,y:BLOCK#ch%,a+4,b+2,x-2,y-1,248:BLOCK#ch%,a,b,x,y,7
1607 FOR I=1 TO 12: READ a,b,H$:CURSOR#ch%,a,b:PRINT#ch%,H$
1608 DATA 150,10,260,77,150,10,260,102,90,10,260,142,90,10,260,156,90,10,260,170
1609 DATA 216,10, 20,77,192,34, 44,102,90,10,146,142,90,10,146,156,90,10,146,170
1610 DATA 256,66,'(L)ast Name',256,90,'(F)irst Names',360,142,'(M)obile'
1611 DATA 360,156,'(H)ome',360,170,'(O)ffice',20,66,'(e)mail',40,90,'(A)ddress'
1612 DATA 40,142,'(P)ost/Zip Code',40,156,'(D)istrict',40,170,'(C)ountry'
1613 DATA 396,46,'(Tab)',160,46,'Edit( ) (Esc)'
1614 Post 5,30: Phone 390,116
1615 STRIP#ch%,7: BLOCK#ch%,120,12,352,29,0: BLOCK#ch%,116,10,354,30,7
1616 CURSOR#ch%,356,30:PRINT#ch%, '← →SEARCH↑ ↓↵': BLOCK#ch%,2,4,464,32,0
1617 END DEFINE

```

QBITS Organiser Planner

This is a colour coded **Year Chart** with day cells allocated to specific Events. These could be repeating on a week or monthly bases or run over several days, weeks even months. All of which is to assist in avoiding those clashes with workload and with a quick glance be reminded of those important anniversaries. Taking the best of the nineteen-eighties layouts the Planner code has been though a number of changes and revamped to follow what are now common page attributes, date change etc.

First a review of the associated array settings. The Original QL Mode 4 Prime colours were limited, so **Stipple** colour options are employed (see DATA Lines below).

```
DIM Event$(12,12),pln%(12,31),col%(12)           :REMark Planner Arrays
RESTORE 146: FOR i=1 TO 12: READ Event$(i)         :REMark Planner Events
RESTORE 149: FOR i=1 TO 12: READ col%(i)           :REMark Planner Colour Codes

Planner Events & Colour Codes
DATA 'Anniversary','Birthday','Doctor','Dentice','Shopping','Meetings'
DATA 'PayDays','Holidays','Project 1','Project 2','Project 3','Project 4'
DATA 2,4,98,87,163,150,226,222,90,96,154,164       :REMark BBQL Mode 4 Colours
DATA 1,2,3,4,5,6,41,48,34,80,230,255              :REMark QPC2 Mode 4 Colours
```

Select Day/Month /Year and Set one of the Event colours by keying [**A-L**] in lowercase.

```
1619 REMark Organiser Planner Page      Events Chart - Year Colour Coded Days

1621 DEFine PROCedure Planner
1622 ch%=1:Page_Cls:Page_Set 3,'Planner':oy%=yr%:OrgDChk 5:Init_Planner:Chart
1623 REPEAT Plan_lp
1624 IF d%<15:pn=44:ELSE pn=60
1625 IF pln%(m%,d%)=0:pcol=7:ELSE pcol=col%(pln%(m%,d%))
1626 BLOCK#ch%,10,5,pn+1+d%*13,59+m%*8,pcol:BLOCK#ch%,8,6,31,48,pcol
1627 k=CODE(INKEY$(-1))
1628 SELECT ON k
1629 =32          :OrgSave 5,'Org'&yr%&'_PL':Org_Contents
1630 =83,115      :OrgSave 5,'Org'&yr%&'_PL'
1631 =97 TO 108:pln%(m%,d%)=k-96          :REMark (a - l) Colour Code
1632 =82,114      :pln%(m%,d%)=0          :REMark (R)eset Colour Code
1633 =65 TO 76:chk=2:Event k-64          :REMark (A - L) Change Event Name
1634 =192,200,208,216:y5%=yr%:Set_Date 1:oy%=y5%:OrgDChk 5:Init_Planner:Chart
1635 END SELECT
1636 END REPEAT Plan_lp
1637 END DEFine
```

The **Twelve Events** labelled **A-L**, can be Set with personal preferences as part of the SuperBASIC Program code, or they can now be edited from within the program using [**Shift A-L**] and QBITS TextED.

```
1639 DEFine PROCedure Event(ev)
1640 SELECT ON ev=1,5, 9 :ex=50:ey%=164+10*(ev% DIV 4)
1641 SELECT ON ev=2,6,10 :ex=160:ey%=164+10*(ev% DIV 4)
1642 SELECT ON ev=3,7,11 :ex=280:ey%=164+10*(ev% DIV 4)
1643 SELECT ON ev=4,8,12 :ex=398:ey%=154+10*(ev% DIV 4)
1644 TextEd 1,0,12,1,ex%,ey%,Event$(ev)
1645 END DEFine
```

1647 **DEFINE PROCEDURE Chart**

Writes the Year Chart Entries

1648 **FOR** mth=1 TO 12

1649 **FOR** day=1 TO dm%(mth)

1650 IF day<15:pn%=44:ELSE pn%=60

1651 IF pln%(mth,day)=0:pcol=7:ELSE pcol=col%(pln%(mth,day))

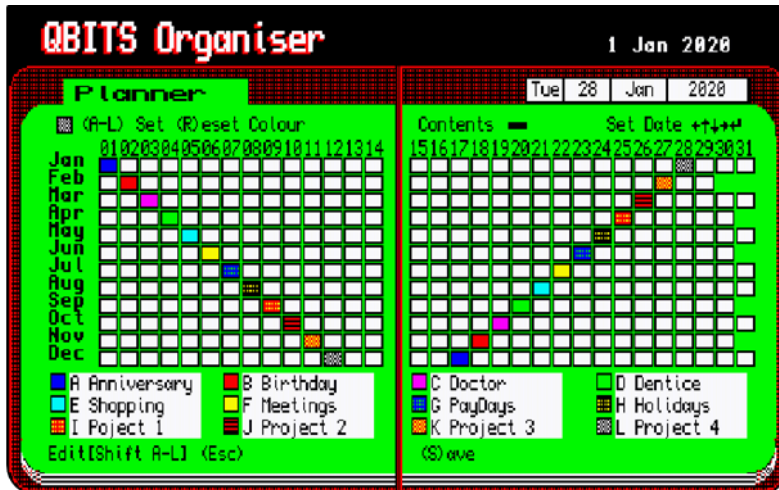
1652 BLOCK#ch%,12,7,pn%+day*13,58+mth*8,0

1653 BLOCK#ch%,10,5,pn%+1+day*13,59+mth*8,pcol

1654 **END FOR** day

1655 **END FOR** mth

1656 **END DEFINE**



1658 **DEFINE PROCEDURE Init_Planner**

Creates the double page display

1659 ch%=1:**Set_Date** 0

1660 CSIZE#ch%,1,0:OVER#ch%,1:INK#ch%,0:STRIP#ch%,bc%

1661 **FOR** mth=1 TO 12

1662 **FOR** i=0 TO 1:CORSOR#ch%,24+i,56+mth*8:PRINT#ch%,mth\$(mth)

1663 **END FOR** mth

1664 CSIZE#ch%,0,0:OVER#ch%,0:STRIP#ch%,bc%

1665 **FOR** n=1 TO 31

1666 IF n<10:N\$='0'&n:ELSE N\$=n

1667 IF n<15:pn%=44:ELSE pn%=59

1668 CURSOR#ch%,pn%+n*13,57:PRINT#ch%,N\$

1669 **END FOR** n

1670 CURSOR#ch%, 44, 46 :PRINT#ch%, '(A-L) Set (R)eset Colour'

1671 CURSOR#ch%, 24,196 :PRINT#ch%, 'Edit[Shift A-L] (Esc)'

1672 BLOCK#ch%,10,8,30,47,0: **RESTORE** 1683:ev%=1

1673 **FOR** i=0 TO 11

1674 **READ** x:y=164+10*(i DIV 4):STRIP#ch%,7:BLOCK#ch%,72,10,x+24,y,7

1675 BLOCK#ch%,10,8,x,y,0:BLOCK#ch%,8,6,x+1,y+1,col%(ev%)

1676 CURSOR#ch%,x+12,y :PRINT#ch%,CHR\$(ev%+64);' ';Event\$(ev%):ev%=ev%+1

1677 **END FOR** i

1678 DATA 26,136,256,374,26,136,256,374,26,136,256,374

1679 **END DEFINE**

QBITS Organiser Calendar

The Calendar Year covers a double page, in the late eighties this was an indulgence to explore SuperBASIC Graphics, coupled with my ambition to create symmetrical layouts. The display is in 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. To change the Year, use the <Up/Down> Cursor keys and Set with the <Enter> key.

1681 REMark **Organiser Calendar Page** Select Year ↑↓ Displays Weekdays/Months

1683 **DEFine PROCEDURE Calendar**

1684 ch%=1:ym%=1700:yx%=2099:**Page_Cls:Init_Cal:Cal_Year** yr%:ch%=1:INK#ch%,0

1685 **REPEAT Cal_lp**

1686 **CURSOR**#ch%,392,30:**PRINT**#ch%, ' ↑:yr%; ↓:BLOCK#ch%,2,4,458,32,0

1687 k=CODE(INKEY\$(-1))

1688 **SElect ON k**

1689 =208:IF yr%<yx%:yr%=yr%+1

1690 =216:IF yr%>ym%:yr%=yr%-1

1691 =10:Cal_Year yr%:ch%=1

:REMark Select year

1692 =32:ym%=1961:yx%=2050:**Init_Date:Org_Contents**

1693 **END SElect**

1694 **END REPEAT Cal_lp**

1695 **END DEFine**

The Selected Year's weekday for the 1st of January provided by **Cal_day year,1,1** (see page 9). Checks are made for non-leap year centuries see lines 1699 & 1700.

Cal_Year **PRINT**'s out the days of the month. Each column of seven days is first cleared with a **FOR** loop and **BLOCK** command. The day of the week [**wd%**] increments on each pass from which row pixel location is calculated. On reaching 7 it is reset to 1, the week [**wk%**] is then incremented and the clear week [**cw%**] is reset (line 1704). The day [**d%**] is also incremented and checked against the month [**dm%(m%)**]. If last day reached this sets the month day [**d%**] to 1 and increments the month [**m%**] (line 1706). Further checks for the monthly quarters add new increments to the Pixel coordinates for **PRINT**ing to screen (see lines 1707-1709).

In certain years, the occasion arises when Jan 1st is on a Sunday and March 31st falls in week 13 of 1st Qtr, for this an additional week 14 column clear is added (see line 1705).

1697 **DEFine PROCEDURE Cal_Year (yr%)**

1698 ch%=3:d%=1:wk%=1:m%=1:x%=35:y%=50:cw%=1 :REMark 1st Quarter

1699 **Cal_day yr%,1,1**: IF yr% Mod 100=0:dm%(2)=28 :REMark 1st Jan week day wk d%

1700 **CSIZE**#ch%,0,0:INK#ch%,0:OVER#ch%,1:IF yr% MOD 400=0:dm%(2)=29 **Note: use OVER command**

1701 **REPEAT day_lp**

1702 IF cw%=1:cw%=0:**FOR day**=1 TO 7:**BLOCK**#ch%,12,8,1+x%+wk%*14,1+y%+day*10,7

1703 **CURSOR**#ch%,x%+wk%*14,y%+wd%*10:**PRINT**#ch%,FILL\$(' ',2-LEN(d%))&d%

1704 wd%=wd%+1:d%=d%+1:IF wd%>7:wd%=1:wk%=wk%+1:cw%=1 :REMark cw% clear week

1705 IF wk%=13 AND d%=31: **FOR day**=1 TO 7:**BLOCK**#ch%,12,8,1+x%+14*14,1+y%+day*10,7

1706 IF d%>dm%(m%):d%=1:m%=m%+1

:REMark days of month

1707 IF m%= 4 AND d%=1:wk%=15:cw%=1:x%= 54:y%= 50

:REMark 2nd Quarter

1708 IF m%= 7 AND d%=1:wk%=29:cw%=1:x%=-356:y%=131

:REMark 3rd Quarter

1709 IF m%=10 AND d%=1:wk%=43:cw%=1:x%=-338:y%=131

:REMark 4th Quarter

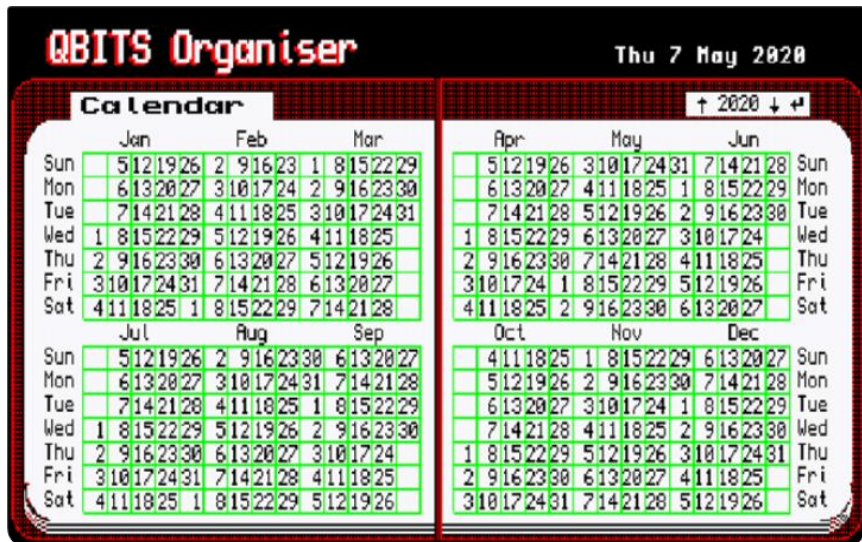
1710 IF m%>12: OVER#ch%,0: EXIT day_lp

:REMark **Reset Over**: Exit

1711 **END REPEAT day_lp**

1712 **END DEFine**

The Calendar grid is drawn with lines using Block and Pixel coordinates system. The screen page dimensions restricted the number of character rows. However, with a little ingenuity characters can be made to fit. Each Grid Cell needs to accommodate two characters this would be 12x10 pixels. The number of rows 16 or 160 pixels just fits within the Organiser Page, but includes the Grid lines so each cell is reduced to 12x8 pixels. Luckily **CSIZE** 0,0 or characters of 6x10 pixels do not use all the rows. Each cell is cleared with a **BLOCK** command and the characters **PRINT**ed using the **OVER 1** command.



```

1714 DEFINE PROCEDURE Init_Cal                                     :REMark Draw Grid
1715 ch%=3:OVER#ch%,1:CSIZE#ch%,2,0:INK#ch%,0:STRIP#ch%,bc%
1716 FOR i=0 TO 1:CURSOR#ch%,46+i,35:PRINT#ch%,'Calendar'
1717 OVER#ch%,0: CSIZE#ch%,0,0:STRIP#1,7
1718 FOR i=1 TO 8: BLOCK#ch%,196,1,50,49+i*10,4
1719 FOR i=1 TO 15: BLOCK#ch%,1,70,35+i*14,59,4
1720 FOR i=1 TO 7: CURSOR#ch%,26,49+i*10 :PRINT#ch%,wd$(i)
1721 FOR i=1 TO 3: CURSOR#3,i*68+2,49 :PRINT#ch%,mth$(i)
1722 FOR i=1 TO 8: BLOCK#ch%,196,1,265,49+i*10,4
1723 FOR i=1 TO 15: BLOCK#ch%,1,70,250+i*14,59,4
1724 FOR i=1 TO 7: CURSOR#ch%,46,49+i*10 :PRINT#ch%,wd$(i)
1725 FOR i=1 TO 3: CURSOR#ch%,i*68+220,49 :PRINT#ch%,mth$(i+3)
1726 FOR i=1 TO 8: BLOCK#ch%,196,1,50,130+i*10,4
1727 FOR i=1 TO 15: BLOCK#ch%,1,70,35+i*14,140,4
1728 FOR i=1 TO 7: CURSOR#ch%,26,130+i*10 :PRINT#ch%,wd$(i)
1729 FOR i=1 TO 3: CURSOR#ch%,i*68+2,130 :PRINT#ch%,mth$(i+6)
1730 FOR i=1 TO 8: BLOCK#ch%,196,1,265,130+i*10,4
1731 FOR i=1 TO 15: BLOCK#ch%,1,70,250+i*14,140,4
1732 FOR i=1 TO 7: CURSOR#ch%,46,130+i*10: PRINT#ch%,wd$(i)
1733 FOR i=1 TO 3: CURSOR#ch%,i*68+220,130: PRINT#ch%,mth$(i+9)
1734 END DEFINE

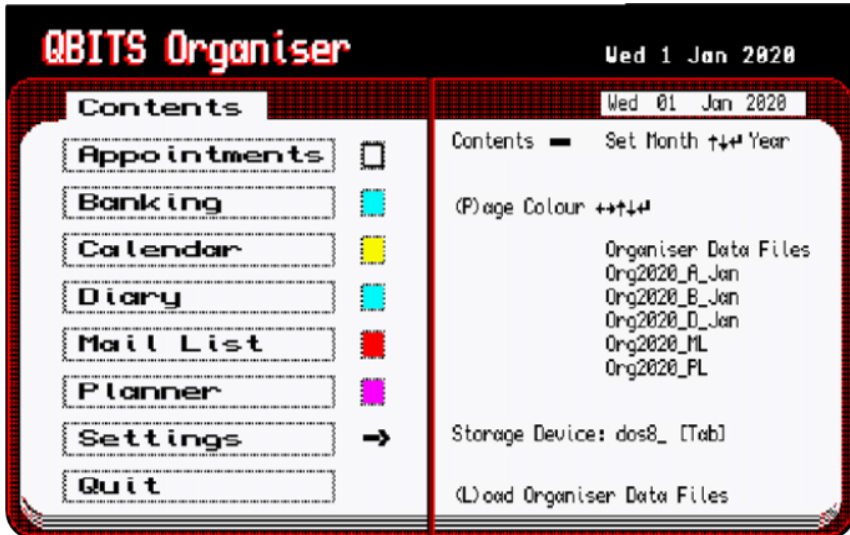
```

QBITS Organiser Settings

Select the **Settings** Page from the **Contents** using <Up/Down> Cursor keys and then the <Spacebar> or by hitting the (S) key. This provides options to arrange different Page background colours, change the current month/year DATA Files, and LOAD them.

QBITS Organiser Default Drive

The array **dev\$** is filled with a number of Device names. These can be altered to one's own preference see Prog DATA at line 1034. To Select a drive, use the (Tab) key.



QBITS Organiser Page Background

Select (P) to access the Page Background settings. Use <Up/Down> Cursor keys to Scroll the Pages and <Left/Right> Cursor keys to Pan the Background Colours, Press <Enter> to Set new arrangement.

QBITS Organiser DATA Files

The Organiser month/year DATA Files are created by and appended by the program. The **Set_Date** at start-up is initialised to the same as QL Clock date. QBITS Organiser tries to LOAD the Page DATA Files from the default Device Drive. The File Check [OrgFChk] will return 'NOT Found!' if the files are not located on the default Drive. The Organiser Pages will then generate new DATA Files.

When swapping in or out of Organiser Pages the month/year DATA File is checked against that currently in use [OrgDChk]. If NOT the same or when setting a different month/year within the Page, QBITS Organiser will try to SAVE the current DATA, before attempting to LOAD the respective DATA File before you can proceed. If NOT Found a new DATA File will be created.

Note: Used to Change Month/Year, Select and LOAD DATA files. Change Page Background Colours.

1736 REMark QBITS Organiser Settings

1738 DEFINE PROCEDURE OrgSetUp

```
1739 ch%=1:bc%=bgc%(1):BLOCK#ch%,224,160,256,46,bc%
1740 CSIZE #ch%,0,0:INK#ch%,0:STRIP#ch%,bc%:OVER#ch%,0 :CURSOR#ch%,260,46
1741 PRINT#ch%,'Contents  ── Set Month↑↓↕ Year' :BLOCK#ch%,2,4,428,48,0
1742 CURSOR#ch%,260, 72:PRINT#ch%,'(P)age Colour ↔↑↓↕' :BLOCK#ch%,2,4,374,76,0
1743 CURSOR#ch%,350, 92:PRINT#ch%,',Organiser Data Files' :BLOCK#ch%,12,3,318,50,0
1744 CURSOR#ch%,260,172:PRINT#ch%,',Storage Device: [Tab]'
1745 FOR i=1 TO 6
1746   BLOCK#ch%,14,12,207,31+i*20,248: BLOCK#ch%,12,10,208,32+i*20,bgc%(i)
1747 END FOR i
1748 REPEAT Dev_Ip
1749 CURSOR#ch%,260,196:PRINT#ch%,',(L)oad Organiser Data Files'
1750 CURSOR#ch%,350,102:PRINT#ch%,',Org&yr%&'_A_'&mth$(m%)
1751 CURSOR#ch%,350,112:PRINT#ch%,',Org&yr%&'_B_'&mth$(m%)
1752 CURSOR#ch%,350,122:PRINT#ch%,',Org&yr%&'_D_'&mth$(m%)
1753 CURSOR#ch%,350,132:PRINT#ch%,',Org&yr%&'_ML'
1754 CURSOR#ch%,350,142:PRINT#ch%,',Org&yr%&'_PL'
1755 CURSOR#ch%,356,172:PRINT#ch%,dev$(dr%):STRIP#ch%,bgc%(1)
1756 k=CODE(INKEY$(-1))
1757 SELECT ON k
1758   =32:EXIT Dev_Ip
1759   = 9:dr%=dr%+1:IF dr%>8:dr%=1 :REMark Storage Drive
1760   =208,216:Set_Date 1:STRIP#ch%,bc%
1761   = 80,112:Page_BkGnd :REMark Page BkGnd Colours
1762   = 76,108:OrgDFile:ct%=65:Org_Contents :REMark (L)oad Data Files
1763 END SELECT
1764 END REPEAT Dev_Ip
1765 bc%=bgc%(1):BLOCK#ch%,218,160,260,46,bc%:STRIP#ch%,bc%:Page_Set 4,'Contents'
1766 END DEFINE
```

Note: Select page pg% and then background bg%, Set background colours bc% for each page

1768 DEFINE PROCEDURE Page_BkGnd

```
1769 REPEAT bg_Ip
1770   BLOCK#ch%,12,10,208,32+pg%*20,0:BLOCK#ch%,10,8,209,33+pg%*20,bgc%(pg%)
1771   k=CODE(INKEY$(-1)):BLOCK#ch%,12,10,208,32+pg%*20,bgc%(pg%)
1772   SELECT ON k
1773     =192:bg%=bg%-1:IF bg%<2:bg%=7:bgc%(pg%)=bg%:ELSE bgc%(pg%)=bg%
1774     =200:bg%=bg%+1:IF bg%>7:bg%=2:bgc%(pg%)=bg%:ELSE bgc%(pg%)=bg%
1775     =208:pg%=pg%-1:IF pg%<1:pg%=6
1776     =216:pg%=pg%+1:IF pg%>6:pg%=1
1777     =10:BLOCK#ch%,16,116,206,48,bc%:EXIT bg_Ip
1778   END SELECT
1779 END REPEAT bg_Ip
1780 OrgSave 5,'Org&yr%&'_PL'
1781 IF bgc%(1)=bc%:RETURN :ELSE ct%=65:Org_Contents
1782 END DEFINE
```

Note: LOAD's DATA Files at Start of program or in Settings if requested. Uses default Storage Device.

```

1786 DEFINE PROCEDURE OrgDFile
1787 OrgFChk 'Org'&yr%&'_PL':IF ck%=1:OrgLoad 5,'Org'&yr%&'_PL'
1788 OrgFChk 'Org'&yr%&'_ML':IF ck%=1:OrgLoad 4,'Org'&yr%&'_ML'
1789 OrgFChk 'Org'&yr%&'_D_'&mth$(m%):IF ck%=1:OrgLoad 3,'Org'&yr%&'_D_'&mth$(m%)
1790 OrgFChk 'Org'&yr%&'_B_'&mth$(m%):IF ck%=1:OrgLoad 2,'Org'&yr%&'_B_'&mth$(m%)
1791 OrgFChk 'Org'&yr%&'_A_'&mth$(m%):IF ck%=1:OrgLoad 1,'Org'&yr%&'_A_'&mth$(m%)
1792 CURSOR#ch%,260,196:PRINT#ch%, '(S)ave': FILL$(' ',24)
1793 END DEFINE

```

```

1795 DEFINE PROCEDURE OrgFChk(SFile$) Check File exists in DIR List
1796 CURSOR#ch%,260,196:PRINT#ch%, 'Searching...':DELETE dev$(dr%)&'FList'
1797 OPEN _NEW#9,dev$(dr%)&'FList':DIR#9,dev$(dr%):CLOSE#9
1798 OPEN _IN#9,dev$(dr%)&'FList':ck%=0:STRIP#ch%,bc%
1799 REPEAT dir_lp
1800   IF EOF(#9)
1801     CLOSE#9:CURSOR#ch%,260,196:PRINT#ch%, 'File NOT Found...'
1802     PAUSE 20: EXIT dir_lp
1803   END IF
1804   INPUT#9,DFile$:IF DFile$==SFile$:CLOSE#9:ck%=1:EXIT dir_lp
1805 END REPEAT dir_lp
1806 END DEFINE

```

```

1808 DEFINE PROCEDURE OrgLoad(FChk,SFile$)
1809 STRIP#ch%,bc%:OPEN _IN#9,dev$(dr%)&'SFile$
1810 CURSOR#ch%,260,196:PRINT#ch%, 'Loading...': FILL$(' ',20):PAUSE 20
1811 IF FChk=5
1812   FOR a=1 TO 6: INPUT#9,bgc%(a)
1813   FOR a=1 TO 12: INPUT#9,Event$(a)\col%(a)
1814   FOR a=1 TO 12: FOR b=1 TO 31: INPUT#9,pln%(a,b):END FOR b:END FOR a
1815 END IF
1816 IF FChk=4
1817   FOR a=1 TO 50: INPUT#9,email$(a)\addr$(a)
1818   FOR a=1 TO 50: FOR b=1 TO 5: INPUT#9,name$(a,b):END FOR b:END FOR a
1819   FOR a=1 TO 50: FOR b=1 TO 6: INPUT#9,mail$(a,b) :END FOR b:END FOR a
1820 END IF
1821 IF FChk=3
1822   FOR a=1 TO 31: FOR b=1 TO 4: INPUT#9,Wth%(a,b):END FOR b:END FOR a
1823   FOR a=1 TO 31: INPUT#9,Jrnl$(a)
1824   FOR a=1 TO 31: INPUT#9,Remd$(a)
1825 END IF
1826 IF FChk=2
1827   FOR a=1 TO 31
1828     FOR b=1 TO 8:FOR c=1 TO 3:INPUT#9,Tran$(a,b,c):END FOR c:END FOR b
1829   END FOR a
1830 END IF
1831 IF FChk=1
1832   FOR a=1 TO 31: FOR b=1 TO 6: INPUT#9,Apt$(a,b):END FOR b:END FOR a
1833   FOR a=1 TO 31: INPUT#9,Note$(a):END FOR a
1834 END IF
1835 CLOSE#9: CURSOR#ch%,260,196:PRINT#ch%, '(S)ave': FILL$(' ',24)
1836 END DEFINE

```

```

1838 DEFINE PROCEDURE OrgSave(FChk,SFile$)
1839 STRIP#ch%.bc%:DELETE dev$(dr%)&SFile$:OPEN_NEW#9,dev$(dr%)&SFile$
1840 CURSOR#ch%,260,196:PRINT#ch%, 'Saving...'; FILL$(' ',20):PAUSE 20
1841 IF FChk=5
1842   FOR a=1 TO 6:PRINT#9,bgc%(a)
1843   FOR a=1 TO 12:PRINT#9,Event$(a)\col%(a)
1844   FOR a=1 TO 12: FOR b=1 TO 31:PRINT#9,pln%(a,b):END FOR b:END FOR a
1845 END IF
1846 IF FChk=4
1847   FOR a=1 TO 50:PRINT#9,email$(a)\addr$(a)
1848   FOR a=1 TO 50:FOR b=1 TO 5:PRINT#9,name$(a,b):END FOR b:END FOR a
1849   FOR a=1 TO 50:FOR b=1 TO 6:PRINT#9,mail$(a,b) :END FOR b:END FOR a
1850 END IF
1851 IF FChk=3
1852   FOR a=1 TO 31: FOR b=1 TO 4: PRINT#9,With%(a,b):END FOR b:END FOR a
1853   FOR a=1 TO 31: PRINT#9,Jml$(a)
1854   FOR a=1 TO 31: PRINT#9,Remd$(a)
1855 END IF
1856 IF FChk=2
1857   FOR a=1 TO 31
1858     FOR b=1 TO 8: FOR c=1 TO 3:PRINT#9,Tran$(a,b,c):END FOR c:END FOR b
1859   END FOR a
1860 END IF
1861 IF FChk=1
1862   FOR a=1 TO 31 :FOR b=1 TO 6:PRINT#9,Apt$(a,b):END FOR b:END FOR a
1863   FOR a=1 TO 31:PRINT#9,Note$(a):END FOR a
1864 END IF
1865 CLOSE#9:CURSOR#ch%,260,196:PRINT#ch%, '(S)ave'; FILL$(' ',24)
1866 END DEFINE

```

Note: When changing months/years the Page may require a change of DATA File. The OrgDChk checks if this is required and if so SAVE's the current DATA File with any updates and then if found LOAD's the requested DATA file. If Not found then a new DATA File will be created.

```

1868 DEFINE PROCEDURE OrgDChk(FChk)
1869 IF oy%=yr% AND om%=m% OR FChk=5 AND yr%=y5%: RETURN
1870 IF FChk=5: SFile$='Org'&y5%&'_PL':LFile$='Org'&yr%&'_PL':y5%=yr%
1871 IF FChk=3
1872   SFile$='Org'&y3%&'_D_'&mth$(m3%)
1873   LFile$='Org'&yr%&'_D_'&mth$(m%) :y3%=yr%:m3%=m%
1874 END IF
1875 IF FChk=2
1876   SFile$='Org'&y2%&'_B_'&mth$(m2%)
1877   LFile$='Org'&yr%&'_B_'&mth$(m%) :y2%=yr%:m2%=m%
1878 END IF
1879 IF FChk=1
1880   SFile$='Org'&y1%&'_A_'&mth$(m1%)
1881   LFile$='Org'&yr%&'_A_'&mth$(m%) :y1%=yr%:m1%=m%
1882 END IF
1883 OrgSave FChk,SFile$:OrgFChk LFile$:IF ck%=1:OrgLoad FChk,LFile$
1884 CURSOR#ch%,260,196:PRINT#ch%, '(S)ave'; FILL$(' ',24)
1885 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 idea for an image is sketched, this is then redrawn over a grid to map out the coordinates. The global x, y is used as a central point for **Vectors** and **Graphics coordinate** system and top left for the **Bitmaps** images using the **Pixel coordinates** system. The offset values for start and end locations can be then worked out.

1887 REMark QBITS Graphics

1889 DEFine PROCEDURE Weather(wn)

```
1890 ch%=1:INK#ch%,0:wx%=14:wy%=16:wr%=2:IF wn>8:wn=0
1891 STRIP#ch%,bc%:BLOCK#ch%,46,30,20,174,7:Wth%(d%,4)=wn
1892 SElect ON wn
1893 =1: Sunny      ch%,wx%,wy%,wr%
1894 =2: Sunny      ch%,wx%+4,wy%+1,wr%*.7:Cloud ch%,wx%,wy%-1,wr%
1895 =3: Windy      ch%,wx%,wy%,wr%
1896 =4: Showers   ch%,wx%,wy%,wr%
1897 =5: Overcast  ch%,wx%,wy%,wr%
1898 =6: Downpour  ch%,wx%,wy%,wr%
1899 =7: Stormy    ch%,wx%,wy%,wr%
1900 =8: Snow      ch%,wx%,wy%,wr%
1901 END SElect
1902 CURSOR#ch%,20,196:PRINT#ch%,Sky$(wn)&FILL$(' ',10-LEN(Sky$(wn)))
1903 END DEFine
```

The Weather Patters

```
1905 DEFine PROCEDURE Sunny(ch%,wx%,wy%,wr%)
1906 CIRCLE#ch%,wx%,wy%,wr% :r1%=wr%*1.5:r2%=wr%*2
1907 FOR i=1 TO 9
1908   LINE#ch%,wx%+r1%*COS(i*60),wy%+r1%*SIN(i*60)
1909   LINE#ch% TO wx%+r2%*COS(i*60),wy%+r2%*SIN(i*60)
1910 END FOR i
1911 END DEFine
```



```
1913 DEFine PROCEDURE Cloud(ch%,wx%,wy%,wr%)
1914 ARC#ch%,wx%-wr%*1.2,wy%-wr% TO wx%-wr%,wy%+wr%,-PI
1915 ARC#ch% TO wx%+wr%,wy%+wr%,-PI*.8 TO wx%+wr%*1.2,wy%-wr%,-PI
1916 END DEFine
```



```
1918 DEFine PROCEDURE Windy(ch%,wx%,wy%,wr%)
1919 LINE#ch%,wx%-wr%*2,wy%+wr%/2 TO wx%,wy%+wr%*2,PI
1920 LINE#ch%,wx%-wr%*2,wy%-wr%/4 TO wx%+wr%*.6,wy%+wr%/4
1921 ARC#ch% TO wx%+wr%*.6,wy%+wr%*1.4,PI
1922 LINE#ch%,wx%-wr%,wy%-wr% TO wx%+wr%,wy%-wr%:ARC#ch% TO wx%+wr%*1.5,wy%,PI
1923 END DEFine
```



Weather Patterns cont...

```
1925 DEFine PROCEDURE Overcast(ch%,wx%,wy%,wr%)
1926 FILL#ch%,1:Cloud ch%,wx%,wy%,wr%
1927 LINE#ch%,wx%-wr%,wy%-wr% TO wx%+wr%,wy%-wr%:FILL#ch%,0
1928 END DEFine
```



```
1930 DEFine PROCEDURE Showers(ch%,wx%,wy%,wr%)
1931 Cloud ch%,wx%,wy%,wr%
1932 FOR i=-1 TO 1:LINE#ch%,wx%+wr%*i,wy%-wr%*0.8 TO wx%+wr%*i,wy%-wr%*1.2
1933 FOR i=-1 TO 1:LINE#ch%,wx%+wr%*i,wy%-wr%*1.5 TO wx%+wr%*i,wy%-wr%*2
1934 END DEFine
```



The Text with this is given as Heavy Rain

```
1936 DEFine PROCEDURE Downpour(ch%,wx%,wy%,wr%)
1937 Overcast ch%,wx%,wy%,wr%
1938 FOR i=-1 TO 1 STEP .4:LINE#ch%,wx%+wr%*i,wy%-wr% TO wx%+wr%*i,wy%-wr%*2
1939 END DEFine
```



```
1941 DEFine PROCEDURE Stormy(ch%,wx%,wy%,wr%)
1942 Downpour ch%,wx%,wy%,wr%:INK#ch%,7:FILL#ch%,1
1943 LINE#ch%,wx%,wy% TO wx%-wr%,wy% TO wx%,wy%+wr% TO wx%+wr%/2,wy%+wr%
1944 LINE#ch% TO wx%,wy%+wr%/2 TO wx%+wr%,wy%+wr%/2 TO wx%-wr%/4,wy%-wr%
1945 LINE#ch% TO wx%,wy%:FILL#ch%,0:INK#ch%,0
1946 END DEFine
```

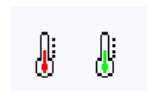


```
1948 DEFine PROCEDURE Snow(ch%,wx%,wy%,wr%)
1949 Cloud ch%,wx%,wy%,wr%
1950 CIRCLE#ch%,wx%-wr%*0.8,wy%-wr%*1.5,wr%/6:CIRCLE#ch%,wx%,wy%-wr%,wr%/6
1951 CIRCLE#ch%,wx%+wr%*0.8,wy%-wr%*1.5,wr%/6:CIRCLE#ch%,wx%,wy%-wr%*2,wr%/6
1952 END DEFine
```



Bitmap Thermostat symbols with colours shown for Hight & Low now independent of background colour.

```
1954 DEFine PROCEDURE Therm(wx,wy,wc)
1955 RESTORE 1963: FOR i=1 TO 16:READ a,b,x,y,c: BLOCK#ch%,a,b,wx+x,wy+y,c
1956 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
1957 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
1958 DATA 3,13,3,1,7, 7,3,1,10,7, 1,8,4,6,wc, 3,2,3,11,wc,
1959 DATA 2,1,8,2,0,2,1,8,4,0,2,1,8,6,0,2,1,8,8,0
1960 END DEFine
```



Vector image to depict Wind speed

```
1961 DEFine PROCEDURE Windmill(wx,wy)
1962 CIRCLE#ch%,wx+2,wy+4,2,.4,-PI/3:CIRCLE#ch%,wx+2,wy+7,2,.4,PI/3
1963 CIRCLE#ch%,wx+6,wy+7,2,.4,-PI/3:CIRCLE#ch%,wx+6,wy+4,2,.4,PI/3
1964 LINE#ch%,wx+3,wy+.5 TO wx+4,wy+5 TO wx+5,wy+.5 TO wx+3,wy+.5
1965 LINE#ch%,wx+2,wy+.3 TO wx+6,wy+.3 TO wx+6,wy TO wx+2,wy TO wx+2,wy+.3
1967 END DEFine
```



The Diary Page (R)eminders

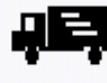
1969 **DEFine PROCEDURE Health(wx,wy,wc)**
1970 **ARC#**ch%,wx,wy-1 TO wx+3.6,wy,-PI TO wx+7.4,wy-1,-PI
1971 **LINE#**ch% TO wx+5.6,wy-4.2:**ARC#**ch% TO wx+2,wy-4.2,-PI/3:**LINE#**ch% TO wx,wy-1
1972 **INK#**ch%,wc:**LINE#**ch%,wx+1.2,wy-1 TO wx+1.8,wy-1 TO wx+2.2,wy
1973 **LINE#**ch% TO wx+3.2,wy-3.5 TO wx+4,wy-1 TO wx+6,wy-1:**INK#**ch%,0
1974 **END DEFine**



1976 **DEFine PROCEDURE Shop(wx,wy)**
1977 **RESTORE** 1984:**FOR** i=1 TO 20:**READ** a,b,x,y,c:**BLOCK#**ch%,a,b,wx+x,wy+y,c
1978 **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
1979 **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
1980 **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
1981 **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
1982 **END DEFine**



1984 **DEFine PROCEDURE Goods(wx,wy)**
1985 **RESTORE** 1992:**FOR** i=1 TO 13:**READ** a,b,x,y,c:**BLOCK#**ch%,a,b,wx+x,wy+y,c
1986 **DATA** 4,5,0, 6,0, 8,8,3,2,0, 3,3, 4, 3,7, 20,10,12,0,0
1987 **DATA** 5,2,4,11,0, 3,4,5,10,0, 5,2,25,11,0, 3, 4,26,10,0
1988 **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
1989 **END DEFine**



Also used for Postage in Mail List

1991 **DEFine PROCEDURE Post(wx,wy)**
1992 **LINE#**ch%,wx+1,wy+4 TO wx+5,wy+4 TO wx+2.5,wy+3 TO wx+1,wy+4
1993 **LINE#**ch%,wx+1,wy+4 TO wx,wy+1 TO wx+4,wy+1 TO wx+5,wy+4
1994 **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
1995 **LINE#**ch%,wx+6.5,wy TO wx+7.5,wy+2 TO wx+7.5,wy+5 TO wx+6.5,wy+3
1996 **LINE#**ch%,wx+7.5,wy+5 TO wx+4,wy+5 TO wx+2.5,wy+4
1997 **END DEFine**

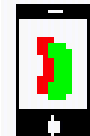


1999 **DEFine PROCEDURE Trash(wx,wy)**
2000 **RESTORE** 2007:**FOR** i=1 TO 8:**READ** a,b,x,y,c:**BLOCK#**ch%,a,b,wx+x,wy+y,c
2001 **DATA** 5,1,5,0,0, 9,1,3,1,0, 15,1,0,2,0, 13,11,1,3,0, 11,1,2,14,0
2002 **DATA** 1,7,3,5,7, 2,8,7,5,7, 1,7,11,5,7
2003 **END DEFine**



For the Mail List Page - Mobile Phone

2005 **DEFine PROCEDURE Phone(wx,wy)**
2006 **RESTORE** 2013: **FOR** i=1 TO 12: **READ** a,b,x,y,c:**BLOCK#**ch%,a,b,wx+x,wy+y,c
2007 **DATA** 19,24,0,0,0, 17,16,1,3,7, 3,2,8,21,7, 1,4,9,20,7
2008 **DATA** 3,10,7,6,2, 2,3,5,6,2, 2,3,5,13,2, 4,1,8,1,7
2009 **DATA** 3,10,10,7,4, 1,8,13,8,4, 2,3,8,7,4, 2,3,8,14,4
2010 **END DEFine**



QBITS Organiser

Wed 1 Jan 2020

Banking

-412.99

(C)alculator

%	C	+	/
7	8	9	*
4	5	6	-
1	2	3	=
N	0	.	

99998.99
+ 100.00
= Overflow!
189.55
189.55
- 47.34
= 142.21
142.21
1200.00
1300.00
100.00

Currency Dollar Euro Pound Yen

Wed 01 Jan 2020

Contents — Set Date ++++
Edit() ++++ (Esc)

(T)ransaction	-/+	Bal
Current Account	0	189.55
Cheque	-47.34	142.21
	0	0
Saving Account	0	1200.00
Deposit	100.00	1300.00
Holiday Deposit	-412.99	0
	0	0
	0	0

(S)ave

QBITS Organiser

Wed 1 Jan 2020

Contents

- Appointments ☐
- Banking ☒
- Calendar ☐
- Diary ☒
- Mail List ☐
- Planner ☐
- Settings ☒
- Quit ☐

Wed 01 Jan 2020

Contents — Set Month +++ Year

(P)age Colour ++++

Organiser Data Files
Org2020_A_Jan
Org2020_B_Jan
Org2020_D_Jan
Org2020_ML
Org2020_PL

Storage Device: dos8_ [Tab]

(L)oad Organiser Data Files

