DRAG'NDROP

for RISC OS computers

Volume 13 Issue 1
Autumn 2023
£4.25

Graphics tablets

Type-in apps:
FilesOpen
BBCFonCat

8-bits:
Pixel scrolling
in BBC Basic

RISC OS . . .at a stroke

A graphics pen tablet controls the pointer like the mouse does, but for some uses like freehand drawing and painting a stylus (pen) is a natural and intuitive way of interacting with the graphics software.

Paint, which has come free with every version of RISC OS, is an example. Once you try the tablet you probably won't want to go back to using the mouse!

Two sizes of tablet are available with bundled RISC OS software. Along with the hardware you get a printed manual giving hints and tips on using the tablet with standard RISC OS applications and a guide to the included RISC OS application, OmbrArt.

Prices start at just £30.00 + P&P worldwide.

System requirements: The product has been tested with a number of computers running RISC OS 5.27 including the Raspberry Pi series of machines and the Pinebook.

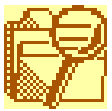
To purchase yours just go to www.dragdrop.co.uk. and follow the link to the tablets page. Payment is via Paypal - you do not have to have a Paypal account to pay by debit or credit card).

DRAG 'N DROP

Produced on RISC OS computers
Programs in this magazine are tested on Raspberry Pis running RISC OS 5. Compatibility with other RISC OS machines and platforms is not guaranteed.
This issue has been blessed with contributions from the following people:
Richard Baraniak (RISC OZ)
Christopher Dewhurst (everything else)

Contact Information
Editor: Christopher Dewhurst
Email: editor@dragdrop.co.uk
www.dragdrop.co.uk

Editorial



I hope you enjoy this issue of *Drag 'N Drop*. There's a range of RISC OS articles planned for Volume 13 but, to quote the editor's old adage, we can't publish what we don't get! If you want a big fat magazine each time then please think about contributing – you don't have write technical, you can just write a story about what you are doing with your RISC OS computer.

The RISC OS Midlands User Group is breaking new ground with their Christmas market on December 9th. This is a fantastic addition to the RISC OS calendar, particularly in view of the London show not happening. Hope to see you there, meanwhile enjoy the read.

Entering listings

You can type in programs in three ways:

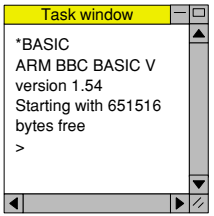
1. Use a text editor like Edit. Press the Menu mouse button over Edit on the icon bar and choose Create>BASIC. For other languages type in at the bottom. Save it by pressing F3.

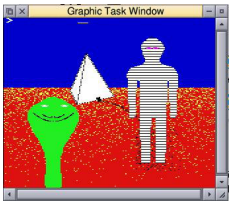
2. Press F12 and type BASIC and press return. Change the screen mode with MODE n where n is a number e.g. MODE 7.

Type AUTO for automatic line numbering. Press Escape to stop and type SAVE "myprog" followed by Return to store *myprog* on hard disc.

Type RUN to execute the program. To return to the desktop type *QUIT. Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated.

3. You can also type programs in a *task window*, Menu over the Raspberry icon on the right side of the icon bar and choose Task window, or press Ctrl+F12. You won't be able to change screen mode or use the cursor keys, though. However, GraphTask from armclub.org.uk/free/ allows you to type in and run Basic programs that use simple graphics (not sprites) in a window on the desktop.





Program lines aren't numbered. Each line starts with space so you should press Return before typing the line. To find out the line number you are on press f5 in Edit.

```
 PROCWRICONF(W1,6,10<<28,15<<28): PROCWR
ICONT(W1,6,"Process")
```

Start of line has a space
Continuation of line (no space)

To run Basic programs from the desktop, double-clicking with select on the filer icon runs it. Holding down Shift and double clicking loads it back into !Edit.

More memory for applications

You may need to reserve more memory for a program. Adjust-click on the Raspberry icon and under *Application tasks* click and drag the *Next* slide bar out to the right.

What does 'currently selected directory' mean?

Articles may tell you to set the CSD (currently selected directory). Click menu over filer window and choose *Set directory ^W*. It's where the computer stores the file when you type SAVE "myprog".

How do I open an Applcation Directory?

Application directories begin with a ! called "pling", for example "pling boot". To open (without running) it, hold down the shift key and double click select to open the directory.

Blank screen when running games listings

Check you have the Anymode module installed, download it from www.pi-star.co.uk/anymode. It goes in !Boot.Choices.Boot. Predesk.

Open the !Boot application directory, in the root directory of the SD Card, that is SDFS::RISCOSPi. $.!Boot. Locate the *Loader* file and with Shift held down double click it to open it. Create a text file in Edit with the following line (press Return at the end):

```
disable_mode_changes
```

Save it inside Loader as CMDLINE/TXT and restart your machine.

'Screen mode not available'

Check you have Anymode (see above). The other issue may be Aemulor which can interfere with screen modes. Menu over iconbar > Quit > Emulator too.

Sounds are strange

Some listings need the free RDSP module installed. Download it from www.amcog-games.co.uk/rdsp.htm where you'll find instructions on how to install it.

WIMP library

Many programs in *Drag 'N Drop* are multi-tasking (running in a window on the desktop). They use a set of standard procedures to create and deal with windows, icons and menus.

Rather than publish them with every listing they are collected here. Most of them are taken from *The Application Tutorial and Listings Book* available from Drag 'N Drop Publications. If you're interested in writing desktop applications then you should consider buying this book.

```
DEF FNMKWINDOW
READ $T,X,Y,W,H
FOR I=0 TO 84 STEP 4
READ A$
I!B=EVALA$
NEXT
T+= LEN $T+1
SYS "Wimp_CreateWindow",,B TO X
=X
```

Basic's DATA pointer is assumed to be at a line giving window title, position, size, colours, flags comes before this is called. Memory blocks T and B must have been set up. Makes a window returning handle in X.

```
DEF PROCMKICON(H,X,Y,W,D,F,A$,V)
$U=A$ : RESTORE +1
DATA H,X,Y,X+W,Y+D,F, U,V,LEN A$+1
FOR I=0 TO 32 STEP 4
READ B$ : I!B = EVAL B$
NEXT : U+= LEN A$+1
SYS "Wimp_CreateIcon",,B TO I
ENDPROC
```

Make an icon, handle is returned in I. H=window handle, X,Y=bottom left, W,D=dimensions, F=flags, A$=text and V validation string (1 if none)

```
DEF PROCRDICON(W,H)
!B=W : B!4=H
SYS "Wimp_GetIconState",,B
A$=$(B!28) : X = B!24
ENDPROC
```

Read icon W in window H text in A$ and flags in X.

```
DEF PROCWRICONT(W,H,B$)
```

```
PROCRDICON(W,H)
B!8=0 : B!12=0
$(B!28)=B$
SYS "Wimp_SetIconState",,B
ENDPROC
```

Updates text B$ in icon H in window W.

```
DEF PROCWRICONF(W,H,X,Y)
!B=W : B!4=H : B!8=X : B!12=Y
SYS "Wimp_SetIconState",,B
ENDPROC
```

Updates icon W's flag in Window H, X is the EOR word and Y the clear word, ie flag = (flag AND NOT X) EOR Y.

```
DEF PROCMKMENU(A)
READ $T
FOR I=0 TO 24 STEP 4
READ A$ : I!A=EVAL A$
NEXT : T+= LEN $T+1
ENDPROC
```

DATA line before this is called with menu header details. Sets up menu header at memory address A.

```
DEF PROCMKENTRY(W,X,Y,H,F,A$,V)
$U=A$
RESTORE+1
DATA Y,H,F,U,V,LEN $U
FOR I=0 TO 20 STEP 4
READ B$
I!(W+X) = EVAL B$
NEXT : U+= LEN $U + 1
ENDPROC
```

Basic DATA line before this is called giving menu item details. W=header address, X=offset (multiple of 24), Y=work flags, H=submenu pointer (–1 if none). F, A$ and V as for PROCMKICON. Add entry for menu already set up.

```
DEF PROCMKSPRITE
READ A$,W,H,MD,PW,NC
SYS "OS_SpriteOp",&10F,S,A$,-(NC()0),W,
H,MD
SYS "OS_SpriteOp",&125,S,A$,-1 TO ,,,,C
IF NC FOR X=0 TO NC*8 STEP 8:READ C!X:
NEXT
READ J$: J=EVAL("&"+J$): K=PW
FOR Y=H-1 TO 0 STEP -1: FOR X=0 TO W-1
SYS "OS_SpriteOp",&12A,S,A$,X,Y,J
J=J))(32 DIV PW): K-=1 : IF K=0 K=PW: R
EAD J$: J=EVAL("&"+J$)
NEXT: NEXT
ENDPROC
```

Create a sprite. DATA pointer must be be at a line giving sprite's name, width, height, mode, pixels per word (eg 32 for 2-colour) and number of colours before PROCMKSPRITE called. If number of colours >0 the following words give RGB palette entries. Shape DATA then given as words (without &) in compacted format. Intended for use with 2-, 4- and 16-colour sprites.

```
DEF PROCPROGINFO
RESTORE +1
DATA About this program,0,0,500,200,X,Y
,X+W,Y+H,0,0,-1,&84001012,&1000207,&C010
3,0,-H,W,0,&13D,0,1,0,T,0,0,0
W0=FNMKWINDOW: $V="R2"
DATA Name,Purpose,Author,Version
DATA Application Demo,Drag N Drop,n.m (
dd-mm-yyyy)
FOR IC=0 TO 7
READ A$:LOCAL DATA
IF IC<4 PROCMKICON(X,0,-50-IC*50,130,50
,&97000301,A$,0)
IF IC>3 PROCMKICON(X,140,-50-(IC MOD4)*
50,350,50,&9700010D,A$,V)
RESTORE DATA
NEXT
ENDPROC
```

Create 'About this program' window giving it handle W0. Adust 3rd line of data as needed. Needs FNMKWINDOW and FNMKICON.

Application Directories

Type-in applications for RISC OS in

the magazine are usually presented as single listings in BBC Basic for convenience and space reasons. The standard 640k of memory allocated under Next in the Tasks window is more than enough for most *Drag 'N Drop* applications.

The programs can be put into proper RISC OS application directories as follows. Create a new directory whose name begins with a pling (!) followed by the application name eg !Txt2Draw. Hold down shift and select double click to open it.

Copy the application typed in from *Drag 'N Drop* into the application directory. It is conventional to rename this main program !RunImage but it isn't compulsory.

Either run the program below or create your own application sprite, it should be no more than 68×68 pixels big. Ensure the sprite name is identical to the application directory name (but in lower case) and save it inside the application directory.

```
DIM S 210
!S=210:S!8=&10:SYS "OS_SpriteOp",&109,S
DATA !myapp,32,32,25,32,1, &BFE3F200,&A
8540000
 DATA 0,0,BC383FBF,FE3C7F7F,EF7C7776,C76
E77E6,76E3FE6,F7FF3FE6,EECF7776,FFCFF77F
,7DE7EFBF,0,0,F7C00,66C00,6E800,7E400,76
000,66000,6F000,0,7F493FBF,FE3E7F7F,EE63
7776,EE4177E6,FEC1FFE6,7E413FE6,1E637776
,1E3EF77F
 DATA 3F49EFBF,0,0,0
 PROCMKSPRITE
 SYS "OS_SpriteOp",&10C,S,"!Sprites"
 END
```

Next create two Obey files in the application directory, one called !Boot with the following line

```
iconsprites <obey$dir>.!sprites
```

and a second Obey file called !Run with the following lines:

```
wimpslot -min 32K -max 32K
run <obey$dir>.!RunImage
```

the K (kilobyte) numbers after wimpslot command may need to be adjusted depending on how much memory the application needs.

Now simply double click the application to run it.

!MyApp

Have you a story to tell about your RISC OS computer or even written a program for it?

Send it to dragdrop@dragdrop.co.uk. Payment available for quality RISC OS journalism.

News and Apps

Graphics Tablets

After an extended period of research and development we now have released a graphics pen tablet for RISC OS. If you are an artist then you may find a graphcis tablet easier to use than the mouse for writing and painting with.

There are two sizes of tablet and they have been tested with Raspberry Pi-based RISC OS machines plus the Pinebook. We would be interested if anyone has tried it with other hardware for example R-Comp range of machines.

Xmas Fayre

The first ever Christmas RISC OS Market hosted by the Midlands User Group takes place on 9th December 2023 near Droitwich, Worcestershire. Entrance is free. Full details including exhibitor list and handy travel information cane be found at mug.riscos.org/xmasmarket/MUGshow.html.

64 Bit Raspberry Pi 5

Picture: YouTube

It doesn't seem that long ago in this publication that we were talking about 64-bit processors becoming the norm. The Pi5 was released for pre-order in September 2023 and has a 64-bit Arm A76 at its heart, the first major revision to the Pi family since 2019 when the Pi4 came on the scene. Unfortunately RISC OS has a long way to go before it can be ported to the new machine. The good news is that the Pi4 and friends will be in production for a long time yet.

Solid State RISC OS

Picture: RISCOSbits

Nottinghamshire-based RISC OS developer RISCOSbits has released a new RISC OS computer, bringing their

Pi4-based-range to a total of 30 options across seven machine types. Housed in an aluminium case it features the ARM8 processor running at speeds up to 2.28GHz with embedded multimedia card (eMMC) storage which means RISC OS software will run faster than ever before. Prices start at £199. More details at pihard.co.uk/shop.co.uk.

!CLFiler 1.44

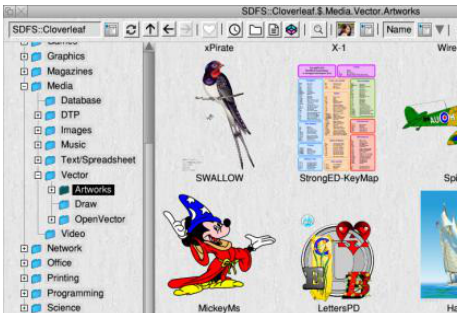


Image:: RISC OS Cloverleaf

CLFiler is an alternative to the standard RISC OS filer and version 1.44 is now available (for die-hards it complements rather than replaces the default filer) and offers a more modern GUI experience – thumbnails, image conversion, tree views, timelines, favourites, export file meta information as CSV and more. CLFiler costs €15.97 with a 21-page manual. Visit riscoscloverleaf.com/product/new-filer-gui-predorder/ to purchase.

Draw to SVG

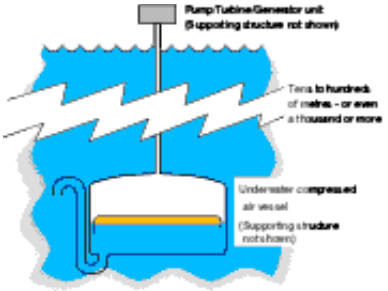


Image: Clive Semmens

In the last *Drag'N Drop* we covered SVG (Scalable Vector Graphics) and exporting to Draw from Netsurf; you can do the reverse with Clive Semmens' app. Clive prefixes all his RISC OS applications with XP and if you're tired of converting Draw to bitmaps for your webpages then !XPDr2SVG is actually very good. Best of all it's free. Visit clivesemmens.org.uk/RISCOS.

Partition Manager 1.02

The application to manage paritions of various media on your RISC OS machine is available from forums.jsapp.org.uk, buried in the discussion threads so search for parttion manager and 'jump' to the latest thread.

YouTube

Don't forget to check out the latest offerings for your favoueirw OS on YouTube. Follow the *RISC OS Community on GitHub* channel where. episode 19 covers creating a desktop front end using the 'FrontEnd' module.

Files Open

Have you been in the middle of something and tried opening a file on the desktop only to get the 'This file is already open' error message?

This may be because you have forgotten to CLOSE# a file in Basic or sometimes a process behind the scenes just bails out without attending to its files.

The application presented here allows you to view details of all files currently open with the option of shutting each of them down.

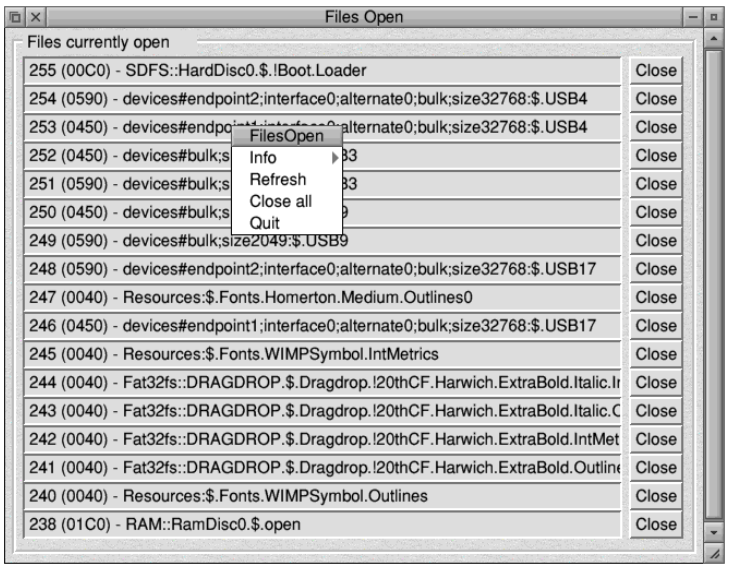
Type in the listing and, ensuring there are no errors, save it. Don't forget to add the relevant procedures from the Wimp library on pages 6-8 of this magazine. Double click it and an icon will appear on the iconbar and a window will open as shown. When a new file is started such OPENOUT in Basic, or a font is 'opened' for use, RISC OS allocates a file handle starting at 255 and counting down.

The file handle is shown on the left with a 16-bit word indicating the file's status and the pathname of the file. For example, if the third digit of the word is 'C' it means the program which opened it has read and write access.

To close a file just click the Close button on the right. Beware that if you close a font file with an application still using it an error is likely and your machine will probably crash if you try closing anything like devices#. You have been warned!

The meaning of each bit in the status word can be found in the *Programmer's Reference Manual* under OS_Args 254 which is how the program finds out about each file. The file handle is passed in R1 and OS_Args returns the status word in R0. If bit 11 is unset it means that file handle is being used and the program then employs OS_Args 7 to get the pathname of the file.

This has to be done in a roundabout way. The file handle is given in R0 and a pre-dimensioned memory buffer whose address is in R1 to put the characters of the pathname in. The size of the buffer is given over in R4. That is

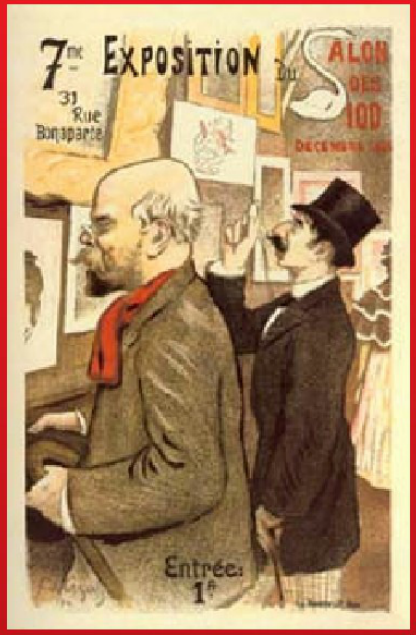


The MUG RISC OS Xmas Market

CHRISTMAS wouldn't be CHRISTMAS without RISC OS *

All the essential RISC OS stands

The chance to see and purchase all your essential RISC OS Xmas goodies



* it would be HTMA

Sponsored by Soft Rock Software and RISCOSitory

9th December 2023 11:30am - 4pm

Dodderhill Parish Community Hall , Wychbold ,WR9 7PU

Free to Entry

For more details see :- www.mug.riscos.org

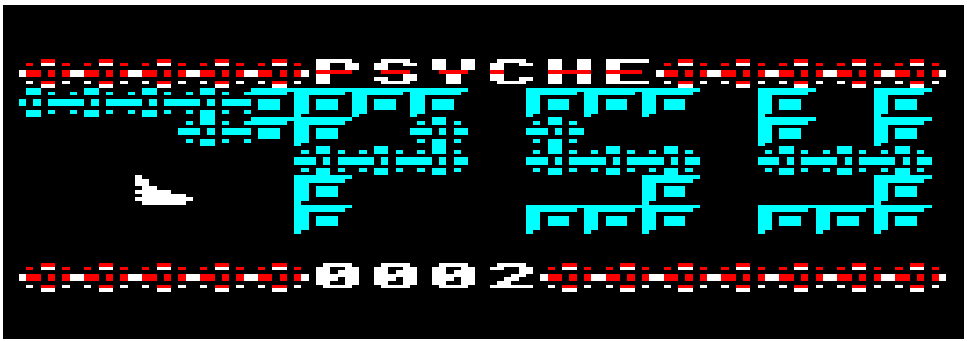
We kick off this season with a BBC Basic game from Eric Richards of Earlscombe, Glos.

It's more of a prototype for a bigger game and demonstrates pixel scrolling at the text cursor in BBC Basic.

Readers will recall last time in 8 Bits how we achieved this by printing pre-defined frames of user-defined graphics (UDG) characters, each one a scrolled version of the last one. Eric has come up with a clever routine to scroll a landscape two pixels at a time, something normally only found in machine code games. The scrolling code can be used in any mode including the text-only ones.

Psyche is a sideways scrolling game. Use the A and Z keys to move your plane up and down to steer through the landscape. The game ends if you crash into the scenery or make it out of the tunnel intact. The scenery rolls past two pixels at a time. (Not quite pixel accuracy then but a vast improvement on whole character jumps usually done in Basic.)

It transpires that as well as a base tile (a tile being a UDG which tessellates), nine rotated and/or partial variations are required to achieve the scroll effect. Quite a lot of setup has to be done whilst 'please wait' is shown on the screen. The first thing the program does is to define two tiles using DATA in lines 1200, then use Osword 10 and lookup tables for the number of binary places to rotate and/or mask each row in the UDG.



I'll take this opportunity to reinterate that GraphTask is recommended for running 8-bit BBC Micro programs (in Basic only) on RISC OS. This is mainly so the low colour depth modes of the BBC don't go squiffy on the 256+ colour modes on the Raspberry Pi. Provided 'legal' calls are used this should be no problem, whilst benefitting from the huge speed increase on RISC OS.

There seems to be a bug when using Osword 10 from within a program, however. At least there is on Pi – the memory block where the bit pattern is put by Osword 10 gets blanked, which doesn't happen in regular, single tasking mode. Run the program, pressing Escape, type PROCchars <return> then re-run.

The next job is to set up the map with PROCmaps. The number of characters across and down the map is READ in from DATA in lines at the end in which 0 represents a space and 1 and 2 one of the two scenery blocks.

Many years ago, when I first taught senior physics and mathematics, I was asked to teach a hobby of mine: computer programming.

In those days, I had just purchased a BBC Master 128 computer. That was after I had done some thorough 'research'. In my humble opinion, that machine beat other all machines available at the time, hands down.

Whilst teaching I had taught myself BBC Basic programming – on the Master 128 this was BBC Basic IV, so how to teach it to younger students with a variety of abilities? After all, that was really new stuff at the time. I still reckon BBC Basic is the best high level language!

When my Archimedes arrived, I decided that I wanted to learn RISC OS. All Archimedes and RISC OS machines are equipped with BBC Basic V, the latest version of BBC Basic.

Alas, there wasn't not enough time. Years later, plus help from *The Application Tutorial and Listings Book* (TATALB) the opportunity has arisen. What an incredibly compact information and knowledge filled book this is!

After reading through it, I needed to go back to some earlier learning and experimentation: variables, array, algorithms, programming constructs of all types, yadda, yadda... Better re-build those foundations again.

That long ago understanding but not-quite-forgotten understanding of variables, array structures and the like required a re-foundation. So, now to continue.

Let's construct a simple array, and something a little different, populate that array with hexadecimal numbers and display the array contents. Program Arrays01 plus its output is the result.

Arrays01

```
 REM Create and display a linear array
   of specific hex values:
CLS
Size% = 3
DIM Hex(Size%)
Hex(0) = &A3
Hex(1) = &D
Hex(2) = &4C
Hex(3) = &61:FOR Z = 0 TO Size%
PRINT Hex(Z)
NEXT:END
```

Oh dear, you can do better. So now try this next effort:

Arrays02

```
REM Create, populate and display a
   linear array of randomly generated
   hex values
CLS
Size% = 3
DIM Hex(Size%)
DemoKount% = 0
REPEAT
DemoKount% += 1
```