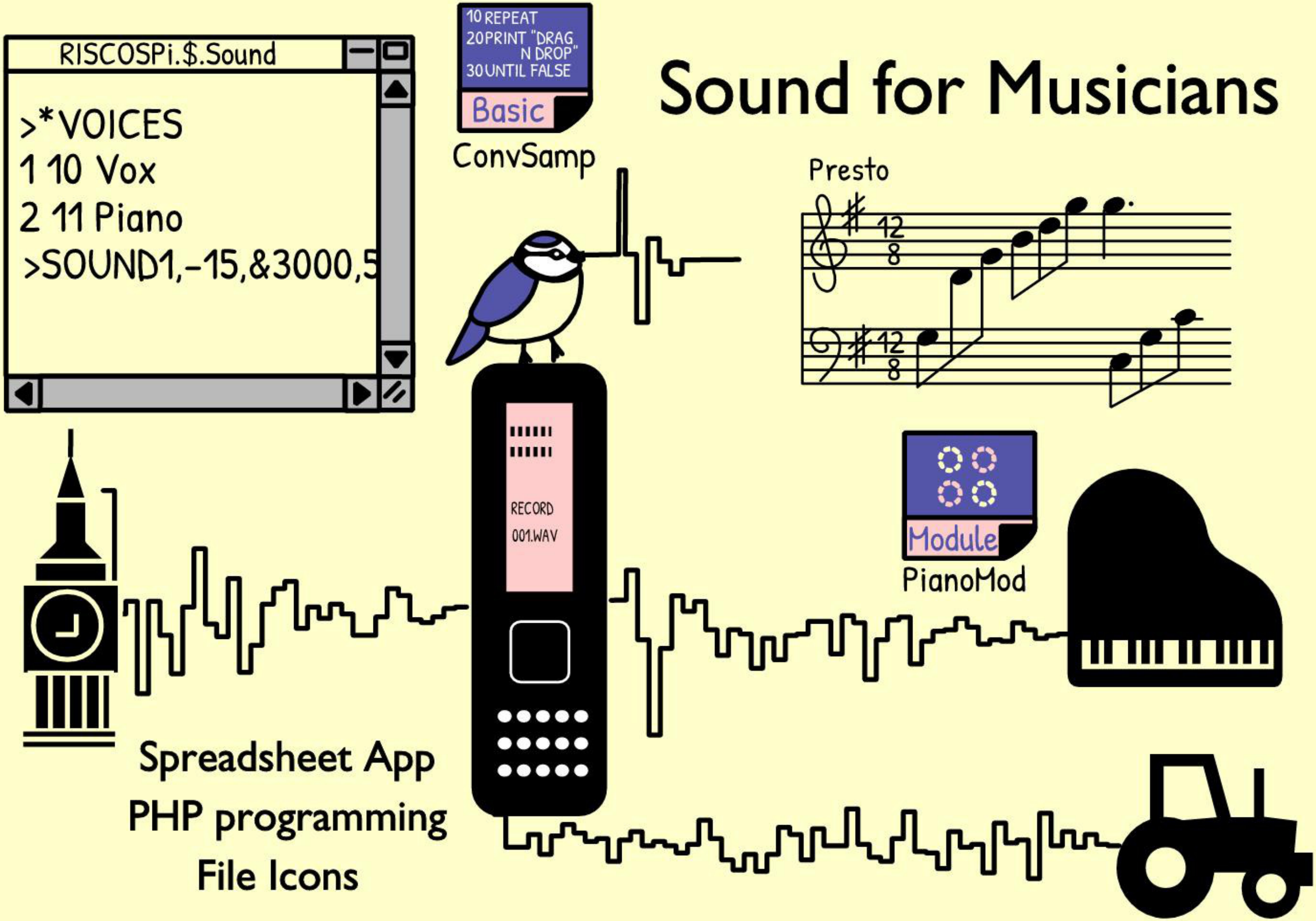
DRAG'N DROP

Volume 11 Issue 3
Spring 2022
£4.25

RISC OS **Pi** and all RISC OS 5 machines

Sound for Musicians



```
>*VOICES
1 10 Vox
2 11 Piano
>SOUND1,-15,&3000,5
```

```
10 REPEAT
20 PRINT "DRAG
    N DROP"
30 UNTIL FALSE
```

Spreadsheet App

PHP programming

File Icons

New Location!

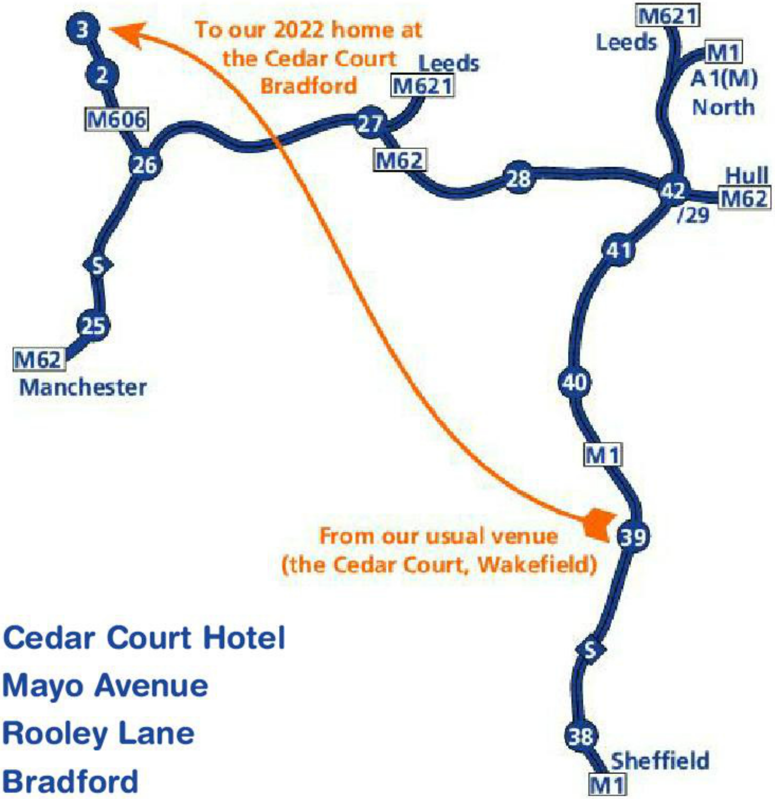
Wakefield

Acorn & RISC OS Computer Show

Saturday 21st May

2022

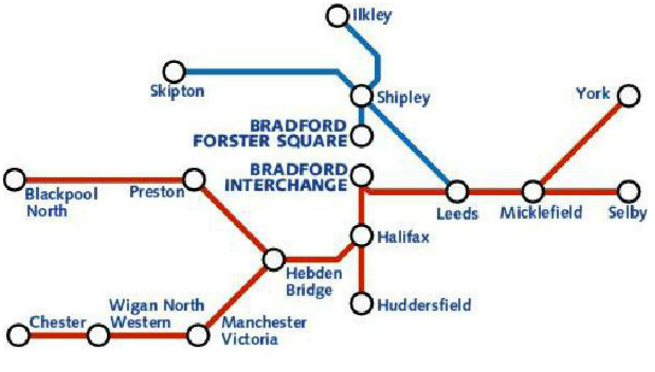
The North's premier RISC OS show

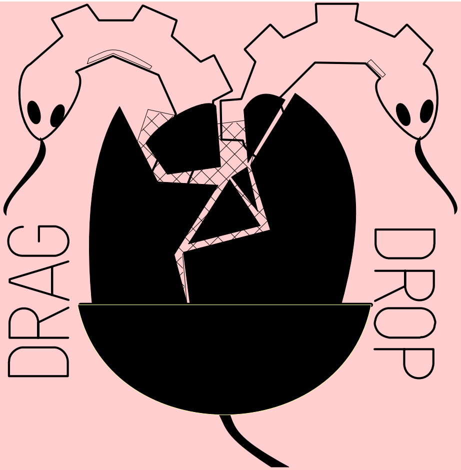


- **Big Names and Small Developers**
- **Raspberry Pi**
- **Show Theatre**
- **On-site Catering**
- **Charity Stall** raising funds for Wakefield Hospice

10.30am – 4.00pm

Cedar Court Hotel
Mayo Avenue
Rooley Lane
Bradford
West Yorkshire BD5 8HW





Copyright ©2022 Drag ‘N Drop
Produced on RISC OS computers

This issue has been blessed with contributions from the following people:
Norman Lawrence (Schema2)
Jeroen Vermeulen (Coding the Classics)
Gavin Wraith (The Uses of Literacy)
Christopher Dewhurst (everything else)

The views expressed in this magazine are not necessarily those of the editor. Alternative views are always welcome and can be expressed by either writing an article or a short editorial. All articles and advertisements are published in good faith. No materials in this publication are meant to be offensive or misleading. If you come across something you believe is either of the above please contact the editor using the details below.

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

EDITORIAL

Whilst carrying out research for the Sound for Musicians article, I had to go through a purchase and returns process a total of eight times owing to being given inaccurate information before the sale. Making pre-order enquiries to check seems to be a waste of time since vendors quote back what you already know reading the product description. Such a lot of trial and error (and patience) is required for internet shopping these days!

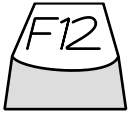
Please support people who ‘know their onions’ including RISC OS developers. Their knowledge and advice is priceless.

And with that rant over, enjoy this issue of the magazine and the RISC OS shows coming up over the next month or two.

Chris.

How do I get the BBC Basic prompt?

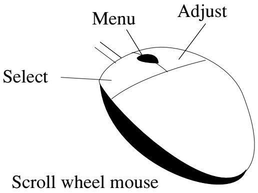
Press F12 and type *BASIC and press Return. You can change the screen mode with MODE n where n is a number e.g. MODE 7 or MODE 0. Type AUTO for automatic line numbering. Press Escape to stop and type SAVE "myprog" followed by Return to store *myprog* on hard disc. To return to the desktop type *QUIT.

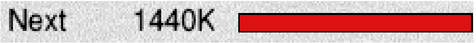


Programs listed in *Drag 'N Drop* are assumed to work on all machines with RISC OS 5 e.g. Raspberry Pi, unless otherwise stated.

How do I open a Task window?

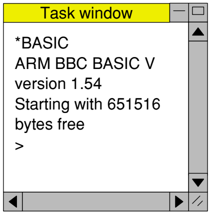
Menu click over the Raspberry icon on the right side of the iconbar and select click on Task window. Or press Ctrl+F12.



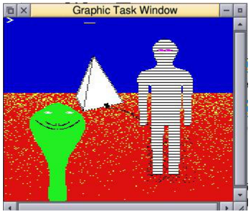


You may need to reserve more memory for the task. Adjust-click on the Raspberry icon and under *Application tasks* click and drag the *Next* slide bar out to the right.

You can also type programs in a *task window*, press Ctrl and F12. You can't use the cursor editing facility or change MODE so you might like GraphTask from armclub.org.uk/free/. It allows you to type in and run Basic programs that use simple graphics (not sprites) in a window on the desktop.



```
*BASIC
ARM BBC BASIC V
version 1.54
Starting with 651516
bytes free
>
```



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 into a text editor like !Edit.

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'. Hold down the shift key and double click select to open the directory. The main !Boot file is an application 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'

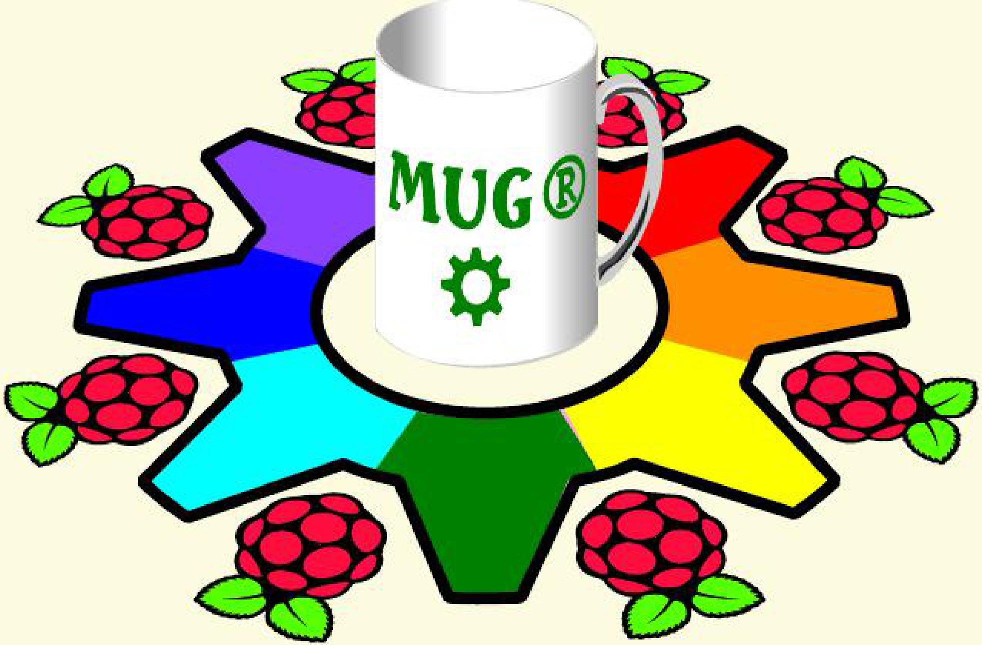
Check you have Anymode (see above). Aemulor can interfere, menu over iconbar > Quit > Emulator too.

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.

Midlands User Group
Virtual Midsummer Show

Saturday 2nd July 11.00 – 4.00



Entirely free but for your time!

RISCOS and Raspberry Pi™

On Line via Zoom in Virtual rooms that include
Getting started with RISCOS, How to demo's , Fixit workshops,
Details on our website

http://mug.riscos.org

http://raspberrypi.org

News and Application Updates

New RISC OS Show Dates

Owing to more overrunning refurbishment hotel works, both the date and the venue of the Wakefield show have been moved. It will now take place at the Bradford Cedar Court hotel on 21st May 2022 from 10.30am to 4pm.

RISC OS Users on the continent will be pleased that the Big Ben Club show is back, on 14th May in central Amsterdam. Visit www.bigbenclub.nl for details.

Don't forget the Midlands User Group virtual show on 2nd July.

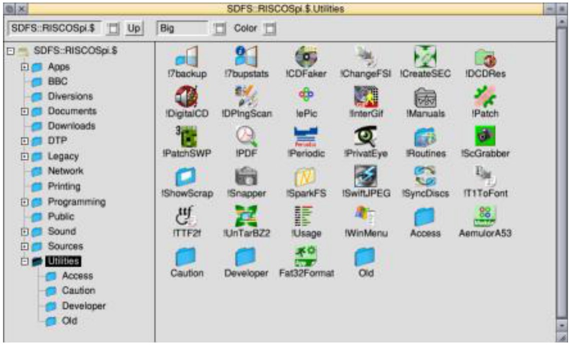
CloverLeaf

The alternative filer, CLFiler has received some bug fixes taking it up to version 0.76. Also you can now drag and drop files between the it and the standard RISC OS filer.

CLFiler installs itself on the left of the iconbar and when clicked a tree-style filer display is shown which will feel more familiar to new users of RISC OS who probably are used to File Explorer on Windows.

Patrons of the CloverLeaf scheme will have received a copy of archive of the updated CLFiler by email.

Due to technical problems with the download process we haven't been able to review CloverLeaf's RISC OS distro.


SDFS::RISCOSpi.$.Utilities

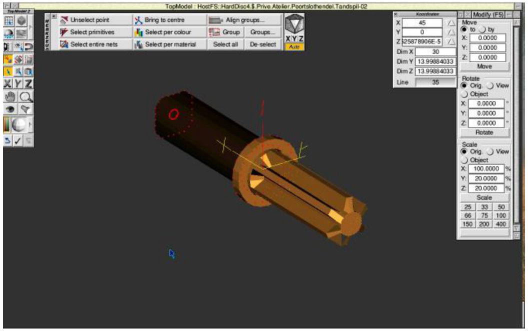
SDFS::RISCOSpi.$ Up Big Color

Open source TCP/IP stack

TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of standardised rules allowing computers to communicate on a network like the internet.

RISC OS Developments have been working on a new TCP/IP stack for RISC OS and now released version 1.0. The new stack improves security and also prepares RISC OS for proper Wi-Fi.

ROD are inviting users to download and try it from the projects page at www.riscosdev.com.


TopModel : HostFS::HardDisc4.$.Prive.Atelier.Poortslothendel.Tandspil-02

X 45
Y 0
Z 525878906E-5
Dim X 30
Dim Y 13.99884033
Dim Z 13.99884033
Line 35

Move ● to ○ by X: 0.0000 Y: 0.0000 Z: 0.0000
Rotate ● Orig. ○ View ○ Object X: 0.0000 Y: 0.0000 Z: 0.0000
Scale ● Orig. ○ View ○ Object X: 100.0000 % Y: 20.0000 % Z: 20.0000 %

TopModel not forgotten

TopModel is a 3D graphics program last developed 20 years ago which has among many uses designing functional parts for 3D printing RISC OS user Steve Royd Marker has restored contact with the author (Paola Castagno) and is garnering support for Paola to update the application to run on modern hardware like the Raspberry Pi. Visit www.markerdesign.de and click on the TopModel graphic at the bottom to see progress.

Catching your bus

RISC OS developer KevSoft has updated BusTimes. Input a postcode, click a button to get a list bus stop codes which are then fed into the application to display times of the next buses. Download from kevsoft.co.uk/.

All the back issues...and more

Apps, programs, PDFs, original DTP, Artworks
from every Drag 'N Drop all on one USB stick.

£20.00
Order at dragdrop.co.uk

In the last issue of *Drag ’N Drop* we explored the WaveSynth module that is built into RISC OS. It comes with a sine wave which produces the standard but rather wimpy beep.

We saw how easy it is to add new and interesting waveforms. Using some simple trigonometry we made a square wave to simulate a clarinet plus entirely new electronic instruments that we can use with the BBC Basic SOUND statement and also in Maestro.

There is one drawback to using WaveSynth, however. Any new wavetable (waveform) can only be 256 bytes, or samples, long. The computer repeats this sample over and over again according to the pitch (frequency) required and for as long as required as specified in the SOUND statement.

That’s fine for sound *synthesis* – making sounds from waves generated by maths – but it isn’t suitable for playing sound *samples*.

To get the computer to play a sound sample, we have to write a *voice generator*. WaveSynth is a voice generator but geared up for 256-byte samples only.

By writing our own we have control over the sample size and other things. Once installed, you can play notes with them just like we did with our WaveSynth instruments.

Incidentally when we say 'voice' generator we mean ‘voice’ in the sense that it can speak a sample, like your hi-fi or phone speakers can play a song.

Now, voice generators have be written in machine code (ARM assembly language) but before you throw your arms up in horror, as long as you get the principles you don't actually have to really understand the machine code. There's a crash course in ARM code at the end of this article in case you're interested.

Before we even start writing a single line of code, we have to understand a bit about sound samples and the way they work on RISC OS.

Let's look at sampling first. A sound sample is just a short recording of a sound. You’ll have heard computer musicians talking about sampling rates. We learned in the last article that the *amplitude* of a sound is a number between –127 and 127 and the sample rate is how often, in the space of one second, this amplitude number’s measured.

A common sample rate is 44.1 kilohertz (44.1hKz) which means that the recording sampled what the sound was doing 44,100 times in the space of a second.

Why that particular number, 44,100 ? Well, the upper limit of human hearing is about 20,000 Hz, and it was found in the early days of digital sampling that you have to sample at least twice the maximum frequency of the sound being captured. So 2 × 20,000 is 40,000 with an extra 100 added for good measure.

If there is one byte per sample (it could be more than one byte but let's keep things simple) then one second of sound takes 44,100 bytes or 43k (1024 bytes to a kilobyte, so 44,100 ÷ 1024 = 43 approximately). If the sound was two seconds that would double to 86k and so on. So sound samples are quite big beasties.

Actually collecting this data is done by a sound sampler, or voice recorder. They're little machines that

This short article is aimed at people who may be familiar with machine code on older computers (e.g. 6502 on the BBC Micro). BBC Basic has always featured an assembler and an programs which assemble machine code will all look something like:

```
DIM code 1000
FOR pass=0 TO 2 STEP 2
P%=code
[ OPT pass
\  assembler here
.label
MOV PC,R14  \like RTS
]
NEXT
```

An area of memory is DIMensioned (reserved), 1000 bytes large in the example.The assembler is entered with square brackets, with P% pointing to the reserved memory.

Two passes are needed through the assembly. The first is when the interpreter is on a sort of 'reconaissance' pass to record the addresses of labels and branches.

On the second pass it actually sets the machine code up.

The ARM has 16 registers, called R0...R15 however R13, R14 and R15 are special,and usually referred to as SP (Stack Pointer), LR (Link Register) and PC (Program Counter) respectively.

Mnemonic	Byte/ Word	Condition	To Register (load) or From Register (store)	Operand 1	Operand 2	Shift

There's no 'Return from Subroutine' instruction in Arm code, the MOV PC,R14 instruction is broadly equivalent.

Machine code instructions are always 4 bytes (one word) in length, not like 6502 where instructions could be one, two or three bytes. Operations can only be carried in the registers only, you can't operate directly on memory (as in the 6502 for instance). The memory must be loaded into a register, operated on, then stored back in the memory.

Generally speaking instructions consist of seven parts, which can be visualised as the colours of the rainbow aide memoire.

Richard the **Mnemonic** (Red) – what the instruction is doing.

Of (Orange) **Size** – Whether the instruction is operating on a byte or four bytes (a word)

York (Yellow) the **Condition** code. If the operation is to affect the flags an 'S' is added. If not (the default) the flags aren't affected.

Gave (Green) the To or From **Register**. If storing to memory, the register to be stored, if loading from memory the register the memory byte or word is stored in.

Battle (Blue) **Operand** number one. This can be a constant (number preceeded by the hash #) or a register.

In (Indigo) Operand number two. Again can be a constant or a register.

Vain (Violet) the Shift. Numbers in registers can be shifted left (multiplied by two) or shifted right (divided by two) before the operation with the other operand

Turning on

my Raspberry Pi one day I noticed that the file icon for PDF files had disappeared and PDF (Adobe Acrobat) files were either showing as blank icons or ones with a question mark.

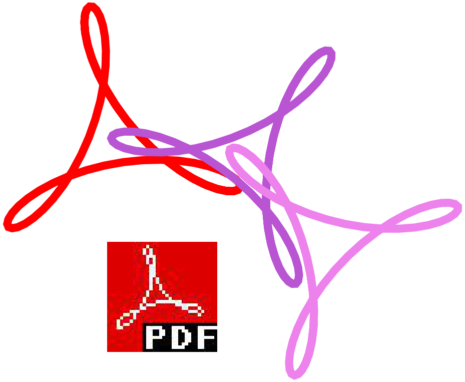
Some time afterwards I discovered that in !Boot > Configuration > Theme the 'Load extra theme resources on startup' had become unchecked so RISC OS wasn't loading a set of icons which included one called file_adf. &ADF is the file type for PDF.

However this wasn't after some effort spent in designing my own icon for PDF files and I thought I would share what I learned here. The program creates an application directory, a sprite called file_adf and a short Obey file in the application directory to merge the sprite with the pool sprites with the *IconSprites command.

The distinctive three point curve used on the Adobe Acrobat logo is actually a mathematical pattern called a parametric curve which I'll go into in a little more detail at the end of the article for anyone interested.

The main business of the program is done in the PROCSPRITES procedure.

Firstly an application directory is created with OS_File 8. Several calls are made to the operating system with SYS "Sprite_Op". Firstly an area of memory, the sprite area (called S here) is dimensioned to store the sprite in.



Sprite_Op number 9 initialises a sprite area. The actual number sent to RISC OS is &109 and not 9, you add &100 to let it know you are talking about a private sprite area and not the system or Wimp sprites.

Sprite_Op number &0F creates a sprite of the specified dimension in a particular mode, here it's 34×34 pixels in Mode 27.

Sprite_Op &3C is a powerful call which redirects graphics output to a sprite, in effect making it a tiny screen inside which you can draw lines, plot VDU characters, even other sprites.

However it must be used with extreme caution. Any mistake and the computer will try to report an error whilse still 'inside' the sprite and cause your Raspberry Pi to hang. The statement SYS "OS_SpriteOp",&13C,S,,1 (where S is the sprite area, note the two commas in the statement) causes output to return to the screen so really you should put this inside any error trapping routine in your programs.

The curve is drawn in the sprite, output redirected back to the screen. Sprite_Op &1B copies a sprite. This is done because there needs to be a sprite whose name matches the application directory's name, although you could design a different one in Paint.

There are now two sprites which are saved in the application directory as !Sprites with OS_Sprite &0C.

The program creates a !Boot file with the *IconSprites command using Basic's BPUT statement and finally the !Boot has it's filetype set to &FEB, the type for Obey.

All being well you will have an application directory called !ADF which whenever is 'seen' by the filer will cause your Raspberry Pi to show this icon for all PDF files.

SpSheet is

a very simple multi-tasking spreadsheet application.

It can be used to create tables with numbers, text and formulas ready for export to Draw. Tab Separated Value (TSV) text files can be dropped onto SpSheet's window. The table can then be dragging in to other applications like Impression.

Once you've typed in the listing and debugged it, double clicking will result in a setup window where you type in the number of columns and rows needed. This is currently limited to 9×9 but can easily be expanded.

Click the Create button and a grid appears. Click once in a cell to reveal the formula, number or text in there. Numbers and formulas begin with an equals sign, text with an apostrophe.

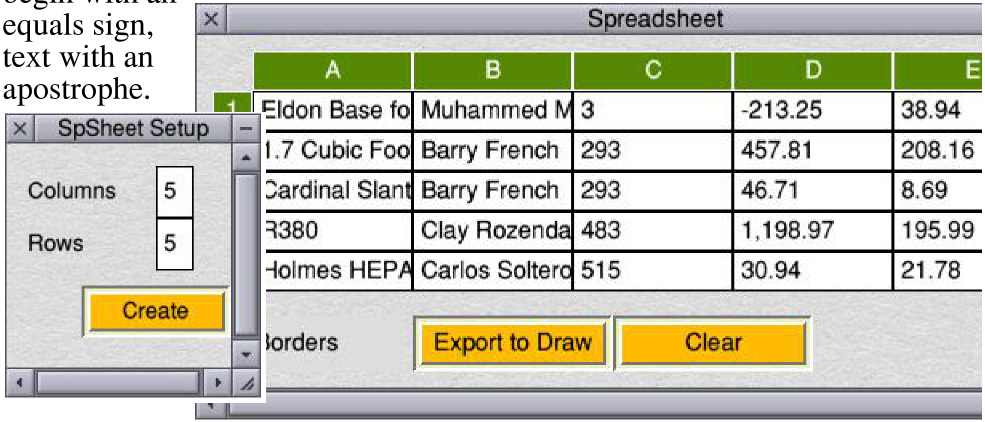
Click twice to edit the data. Press Return or click anywhere else in the window to confirm your entry. If the application detects alphabetical characters it assumes text and prefixes with an apostrophe. Similarly, if your entry starts with a digit or minus sign SpSheet will prefix it with an equals. To force digits into a textual entry, manually type in an apostrophe.

The program demonstrates several aspects of Wimp programming such as responding to files dragged to it, changing the attributes of an icon (cell) when clicked and constructing Draw files for export.

The application doesn't need an application directory but you could set one up.

Many of the procedures are taken from *The Application Tutorial and Listings Book*, available from Drag 'N Drop for £20.00 plus postage.

SpSheet could benefit from plenty of improvements including saving and loading in the application's internal, native format, cell references, colours and option of choosing cell sizes.



Columns 5

Rows 5

	A	B	C	D	E
1	Eldon Base fo	Muhammed M	3	-213.25	38.94
	1.7 Cubic Foo	Barry French	293	457.81	208.16
	Cardinal Slant	Barry French	293	46.71	8.69
	R380	Clay Rozenda	483	1,198.97	195.99
	Holmes HEPA	Carlos Soltero	515	30.94	21.78

Borders

SPSHEET listing

```
REM SpSheet Simple Spreadsheet
REM (c) Drag N Drop April 2022
ON ERROR PROCERROR
APP$="Simple Spreadsheet"
DIM B 2000,  T 1000, U 3000,V 50
Q=FALSE
SYS "Wimp_Initialise",200,&4B534154,APP$
PROCWINDOWS1 : PROCICONS1
!B=W1
SYS "Wimp_GetWindowState",,B
SYS "Wimp_OpenWindow",,B
REPEAT
SYS "Wimp_Poll",,B TO E
CASE E OF
WHEN 2:SYS "Wimp_OpenWindow",,B
WHEN 3:SYS "Wimp_CloseWindow",,B : Q=TRUE
WHEN 6:PROCCLICK
WHEN 8:PROCKEY
WHEN 17,18:PROCMESSAGE
ENDCASE
UNTIL Q OR INKEY-113
SYS "Wimp_CloseDown"
END
:
DEF PROCCLICK
BT= B!8 : WH = B!12 : IC = B!16
IF WH=W1 PROCSETUP:ENDPROC
IF IC>=AI PROCACTION:ENDPROC
IF IC<>OI PROCABANDON
IF IC>-1 AND IC < RC PROCMODIFY
ENDPROC
:
DEF PROCMESSAGE
CASE B!16 OF
```

PHP stands for Hypertext Pre-Processor. It’s a scripting language used to develop static and dynamic webpages.

You can perform file handling operations, send emails, access and modify browser cookies – in fact almost everything that you might need in your web project. Facebook, Yahoo, Wikipedia etc, all use PHP.

Most importantly PHP is open source so anyone can use it.

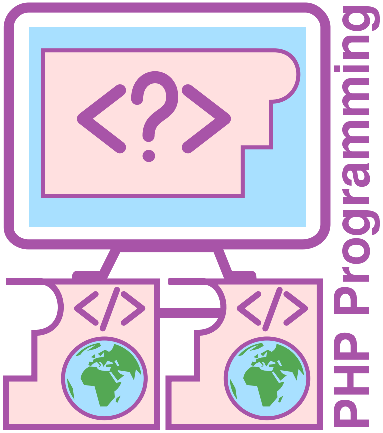
To run and execute PHP code, we need an interpreter to run your PHP, rather like the BBC Basic interpreter runs our Basic programs.

Usually the PHP interpreter resides on a web server, so it’s called a server-side language. All the popular hosting services have PHP installed their server.

The PHP code is executed on the server and the result is sent to the browser such as Netsurf in plain HTML.

You could use your web service's PHP interpreter, assuming you have a website, but a better option is to use !PHP for development. It's a free PHP interpreter for RISC OS that can be downloaded and installed on your machine.

Of course, once you’re confident you can upload your code to your website for fine tuning and incidentally benefit from features in the newer versions of PHP.



Getting !PHP running is a bit of a faff, firstly go to www.cp15.org/php/ and click on ‘Download version 2.23 6-May-07’. Save the php-2/23/zip file to a suitable location on your hard disc (I put it in the Downloads directory). You may have to set the filetype of the downloads to Archive by click menu over the file > Dir ’php-2/23/zip’ > Set Type > and type archive or ddc and press Return.

You now need to patch the program to run on the Pi using a download from another website, www.riscosports.co.uk/vfp/webjames.zip. Note, do not navigate to Downloads on riscosports.co.uk and then try clicking on "WebJames http server..." as that seems to be something different.

Once you have downloaded webjames/zip and opened the archive, copy out !PHP over the other copy you saved a moment ago. (Incidentally, Webjames is another application you can use in PHP development which we’ll come to later in the series.)

This updates the main application code dating from 2007 to run on modern hardware.

In this article we’ll learn the syntax to write basic PHP code, how we can integrate PHP code in an HTML file and how to run the PHP script and check the output in the browser such as Netsurf.

All the PHP code is written inside tags which are <?php and ?> like this:

```
<?php
// code statements
?>
```

To make RISC OS recognise programs as PHP the file type needs to be set to &18A.

Hence a simple “Hello, World!” program in PHP will be:

```
<?php
echo "Hello, World!";
?>
```

echo is a command used in PHP to display anything on screen. If we want to include this PHP code in an