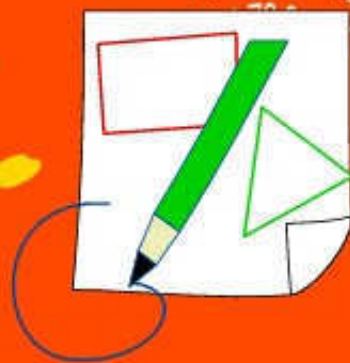


# The Book of Draw stuff.

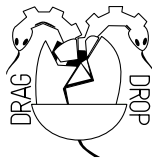
Sample

44 72 61 77  
C9 00 00 00  
00 00 00 00



for RISC OS computers  
Chris Dewhurst

# The Book of Draw Stuff



**DRAG'NDROP**

[www.dragdrop.co.uk](http://www.dragdrop.co.uk)

## **Acknowledgements**

The author is indebted to the writers of the many fine applications developed for RISC OS computers, without which this book would have been very difficult to produce.

Produced on RISC OS computers using EasiWriter, Draw, PostScript3 printer driver, and the OKI-C321 colour laserjet printer.

© 2015 (with the exception of Appendix 3) Chris Dewhurst

ISBN 978-1-78280-501-4

First published 2015 by  
Independent Publishing Network (IPN)  
[www.ipubnet.co.uk](http://www.ipubnet.co.uk)

Printed and bound in Great Britain by  
Drag 'N Drop,  
Burrsville, Essex UK

[www.dragdrop.co.uk](http://www.dragdrop.co.uk) [dragdrop@dragdrop.co.uk](mailto:dragdrop@dragdrop.co.uk)

Typeset in 12/13 Plato.


# Contents

Contents	3
Introduction	4
Know your Mouse	5
<b>1 Using Draw</b>	
The Unit Square	6
Projects (1)	7
Scaling (1)	8
Projects (2)	10
Scaling (2)	11
Position and Alignment	12
Projects (3)	16
Grids	18
Cross Hatching	24
Format Painter	25
Projects (4)	26
Designing Fonts	27
Projects (5)	30
<b>2 Programming Draw</b>	
Draw file Structure	36
Notes on the listings	40
Grid Generator	41
Text Effects	44
Rotated text	49
Curved text paths	53
Table Maker	56
Working with curves	60
Easier Circles	62
Concentric Circles	63
Colour Graded circles	66
Curved Boxes	67
'Gons, Stars and Cogs	72
More Draw Colours	79
More Dash Patterns	84
Converting Foreign Files	88
Path Merger	94
Appendix 1 - !EasyCSD	A1-1
Appendix 2 - !PathMerger	A2-1
Appendix 3 - Draw reference	A3-1


# The Unit Square

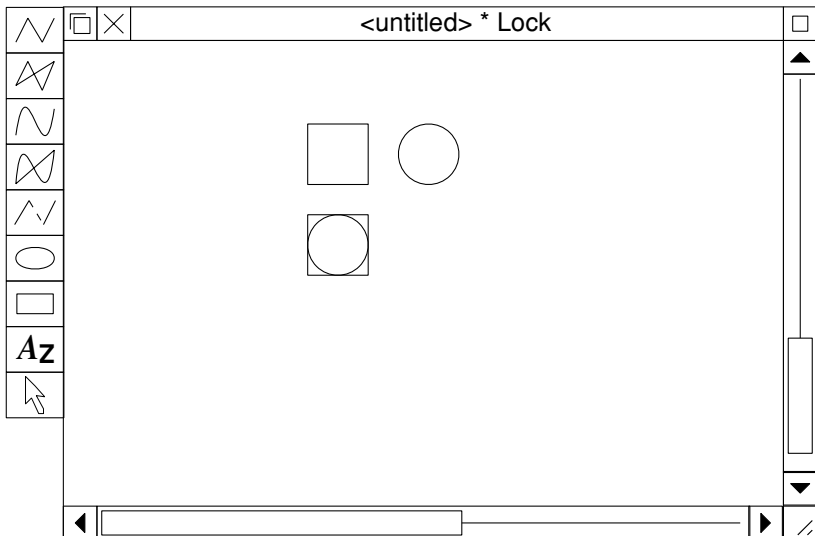
For accurate diagrams the *unit square* is the most important element. It may not be used by itself but it's an important building block.

Open a Draw document. Click menu > Grid > Lock or press Ctrl+Shift+F1. This locks to the default grid of 1 centimetre with 2 subdivisions.

 Select the rectangle tool and select-click on the paper. We said above the grid is 1cm with 2 subdivisions so one move of the mouse corresponds to  $1\text{cm} \div 2 = 0.5\text{cm}$ .

Drag out a square shape, 2 grid steps horizontally and vertically. This is your *unit square*, measuring  $1\text{cm} \times 1\text{cm}$ .

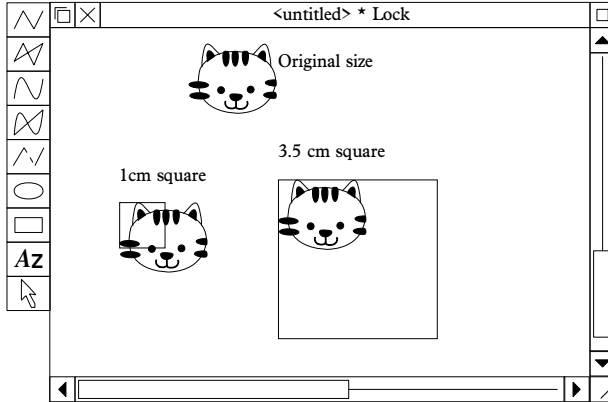
 Select the circle tool. This time drag out a circle 1 step horizontally and vertically. This is a *unit circle* with diameter 1cm. It should fit inside the unit square.



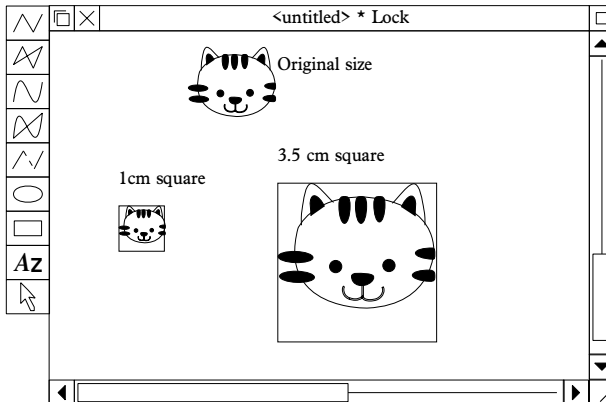
*Unit squares and unit circles (viewed at default 1:1 magnification).*

## Scaling (2)

To reduce or enlarge a drawing (object) to a known size, scale a unit square to the known size. Select the square and the object, group, and justify to Top and Left.

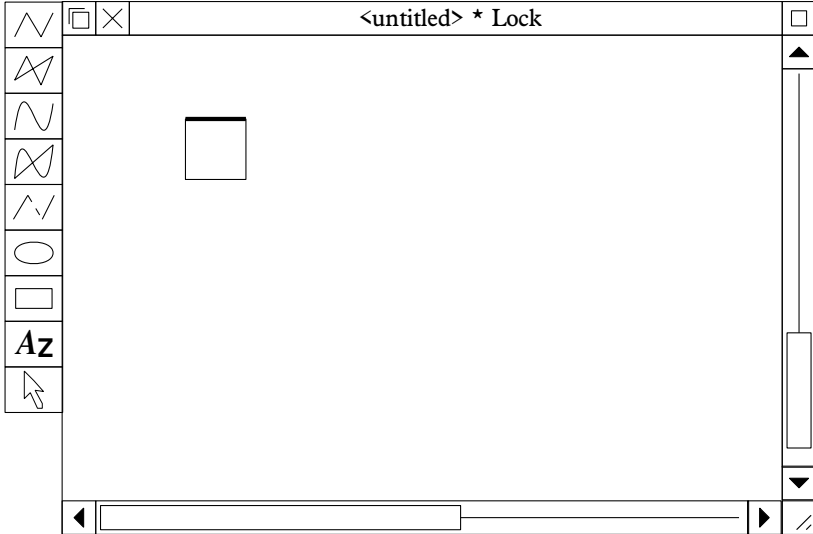


Ungroup the object and square and select the object. Choose Menu > Transform > Magnify. For enlargements type 1.1 in the box and repeatedly *select click* to enlarge by 10% until the bounding box of the object co-incides with the square. To reduce enter 0.9 in the box and repeatedly *select click* to reduce by 10%. (For fine tuning use 1.01 for 1% enlargement, 0.99 for 1% reduction.)



# Grids

Draw a unit square. Use the line tool to draw a horizontal line following the top (or bottom) of the square so that you have a line 1cm long. (As it is in exactly the same position as one side of the square, you may wish to change the line width and/or colour to make it stand out from the side of the square.)

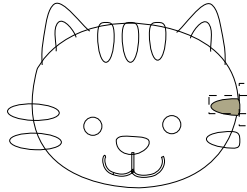


Delete the unit square to leave the 1cm line. Decide how many squares are to go along the top side of your grid, at 0.5cm intervals. Multiply them, for example 20 squares  $\times$  0.5cm = 10cm. Lengthen the unit line to 10cm with the Transform > Xscale option.

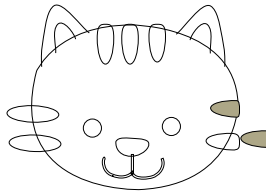
Leaving the line selected press Ctrl+C. Draw makes a copy of the line at an offset determined by the grid division (0.5cm)

## Format Painter

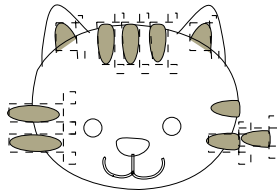
Draw sets the line style, fill colour etc the same as the last object on its 'stack'. For example, set the required fill colour for one part of the drawing (Menu > Style > Fill colour)



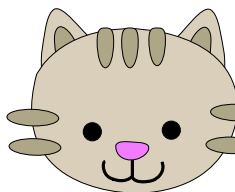
Make a copy of the filled shape to bring it to the top of the 'stack'.



Adjust-select the other shapes to be filled in the same colour and choose Style > Fill colour > OK



Delete the extraneous copy. Format paint other parts of the cartoon if required

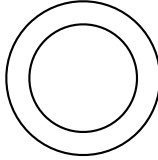




# Designing Fonts

Fonts sizes are measured in points and there are 72 points to the inch so it makes sense to work in inches. Choose Grid > Inch > Spacing > 1 and Subdivision 2. Draw a 1 inch unit circle and set fill to black. Make a copy, magnify 0.7 and set fill to white. Group the two circles and justify to centre middle.

This is a basic 'O' shape at 72 points (1 inch).



Make a copy and set the outline colour to light grey. This is the guideline for other letters based on the O (C, G, O and Q).

Draw a unit square. Set Grid > Inch > Spacing > 72. Each grid position is 1 point apart which is good enough for most purposes at this scale (letters can be shrunk later on making any slight unevenness less noticable). Within the unit square draw some rectangles which are 0.2 which will also act as guidelines.



Group a rectangle and the O shape and justify to centre bottom.



Select the curved path tool and choose a thicker line width in black. Trace around the guidelines to make the Q. It can be

## Draw file Structure

We will look at a word-by-word memory dump of a Draw file consisting of just a unit square. The left hand column shows the offset from the start of the file in hex, the four bytes making up the word, and an Ascii representation if appropriate.

The first word of a Drawfile spells out “Draw” in Ascii characters. In computing, when talk about a *word* we mean a collection of a certain number of bits, rather like a word in English is a collection of letters. A byte is a word of 8 bits but a convention has been established in RISC OS where *word* means a group of four bytes (32 bits).

0000	44	72	61	77	Draw
------	----	----	----	----	------

In hexadecimal the Ascii characters for “Draw” are &44, &72, &61 and &77 in hex. So every Drawfile begins with these four numbers. The next two words are:

0004	C9	00	00	00	É...
0008	00	00	00	00	....

Words at offset &0004 and &0008 concern which version of Draw which created the file. They aren’t the same as the numbers which appear on the “About this program” window. They are currently fixed at 201 (&C9) and zero and you should use the same version in your programs.

In RISC OS bytes are stored *low byte to high byte* or *low byte first* so the 201 (&C9) appears at position &0004. (In other computing worlds data is stored the other way round, high byte to low, so the &C9 would appear at position &0007.)

Words three to five contain the name of the program which created the Draw file. This could be Draw itself or the name of the application which output the file, which is up to twelve characters in length padded with spaces (&20).

000C	44	72	61	77	“Draw”
0010	20	20	20	20	
0014	20	20	20	20	

Finally a SYStem call is made to save the Draw file. OS\_File with R0=0 saves the data and a \*SetType ensures the file is correctly filetypeped so you can double click on it to open it in Draw.

Notice that the entire grid is just one object and it would take much longer to create by hand in Draw as in the first section of this book.

```
10 REM Grid Generator
20 REM The Book of Draw Stuff
30 REM (c) Drag 'N Drop 2015
40
50 DIM mem% &1000
60 ptr%=0
70 unitsperinch% = 46080
80 xlength=4
90 ylength=4
100 size=0.1
110 PROCheader
120 ptr1%=ptr%:REM record current pointer position so can record
object size later
130 PROCwrite(2):REM object 2, path object
140 PROCwrite(0):REM object size - dummy value for now to be
filled in later
150
160 REM object boundary box
170 FOR x=0 TO 3:PROCwrite(0):NEXT
180
190 PROCwrite(-1):REM fill=none
200 PROCwrite(0):REM outline colour=black
210 PROCwrite(0):REM line width=thin
220 PROCwrite(0):REM path style options (see PRM4-468)
230
240 FOR gx=0 TO xlength STEP size
250 PROCwrite(2)
260 REM move to (gx,0)
270 PROCwrite(gx*unitsperinch%)
280 PROCwrite(0)
290
300 PROCwrite(8)
310 REM draw to (gx,ylength)
320 PROCwrite(gx*unitsperinch%)
330 PROCwrite(unitsperinch%*ylength)
340 NEXT
350
```

## Curved text paths

So far our text has followed straight lines. We can generate curved effects e.g. where the text follows an arc. We have to write out the text letter by letter, using some basic trigonometry to adjust the position and angle of each character, as the following program demonstrates.

```
10 REM Text on arcuate paths
20 REM The Book of Draw Stuff
30 REM (c) Drag 'N Drop 2015
40
50 DIM mem% &1000
60 ptr%=0
70 unitspercm%=18140
80 unitsperpoint% = 640
90 xcentre%=10*unitspercm%
100 ycentre%=10*unitspercm%
110
120 angle=0
130 scale%=1*&10000
140 fontsize%=12
150 radius=5*unitspercm%
160 start=0
170 end=180
180 text$="THE QUICK BROWN FOX JUMPS"
190 step=(start-end)/LENtext$
200 angle=start
210
220 PROCheader
230 ptr1%=ptr%:REM record object size later
240 PROCwrite(0):REM Object 0 font table
250 PROCwrite(0):REM object size dummy value to fill in later
260 PROCwritestring(CHR$1+"Homerton.Medium"+CHR$0,0):REM ID 1 +
font name
270 ptr1%!(mem%/4) = ptr%-ptr1%:REM fill in object size
280
290 FOR let%=1 TO LEN text$
300 IF MID$(text$,let%,1)=CHR$32 THEN angle=angle+step:NEXT
310 ptr1%=ptr%
320 PROCwrite(12):REM object 12, transformed text
330 PROCwrite(0):REM object size dummy value to fill in later
340 FOR x=0 TO 3:PROCwrite(0):NEXT
350 REM transformation matrix
360 letangle=angle
370 PROCwrite(COSRADletangle*scale%)
```

# Table Maker

This program is a simple table generator which combines our skill and knowledge from the preceding two chapters (text and lines). You are prompted for the name of a CSV file (Comma Separated Value, filetype &DFE) file in the current directory. The CSV should have the standard RISC OS line separator, CHR\$10.

The file is formatted into a Draw file called *Table*. CSV files are exported by almost all modern spreadsheet applications. An example is:

```
Name,Draw colour,HTML code,%Red,%Green,%Blue
Black,&00000000,#000000,0,0,0
White,&FFFFFFF00,#FFFFFF,100,100,100
Red,&0000FF00,#FF0000,100,0,0
...
Navy,&80000000,#000080,0,0,50
```

which results in

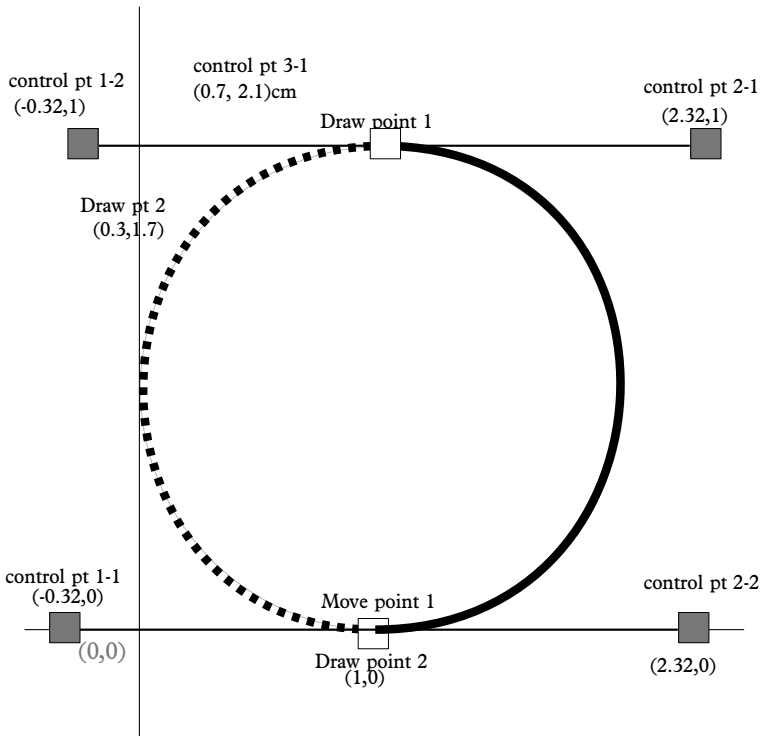
Name	Draw colour	HTML code	%Red	%Green	%Blue
Black	&00000000	#000000	0	0	0
White	&FFFFFFF00	#FFFFFF	100	100	100
Red	&0000FF00	#FF0000	100	0	0
Lime	&00FF0000	#00FF00	0	100	0
Blue	&FF000000	#0000FF	0	0	100
Yellow	&00FFFF00	#FFFF00	100	100	0
Cyan	&FFFF0000	#00FFFF	0	100	100
Magenta	&FF00FF00	#FF00FF	100	0	100
Silver	&C0C0C000	#C0C0C0	75	75	75
Gray	&80808000	#808080	50	50	50
Maroon	&00008000	#800000	50	0	0
Olive	&00808000	#808000	50	50	0
Green	&00800000	#008000	0	50	0
Purple	&80008000	#800080	50	0	50
Teal	&80800000	#008080	0	50	50
Navy	&80000000	#000080	0	0	50

For every item in the CSV file a cell (rectangle) is generated

The move points are white squares and the associated control points are grey (point 1-1 and 1-2 being the control points for the first curve, 2-1 and 2-2 for the second and so on.)

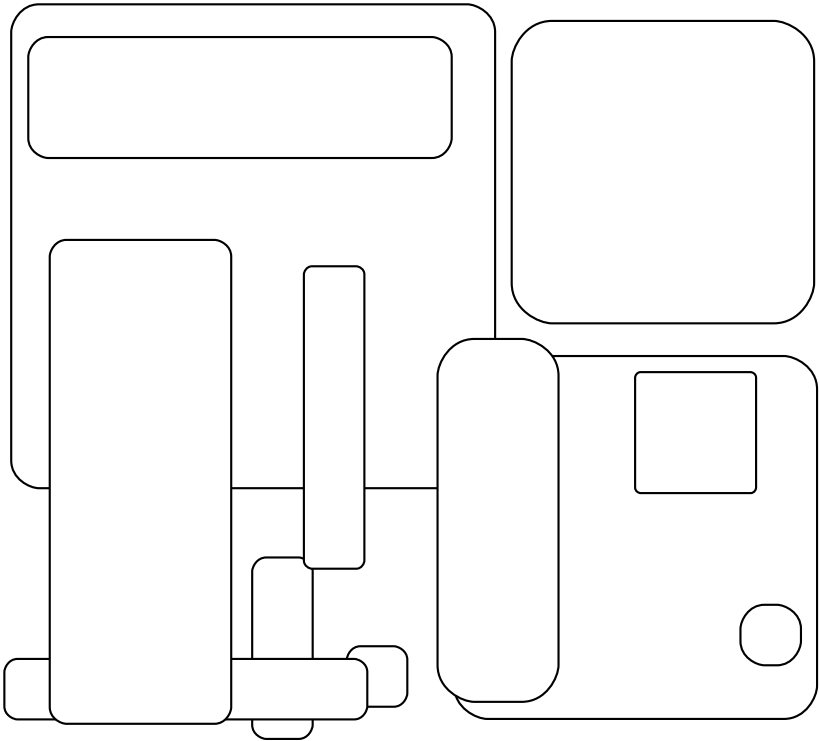
## Easier Circles

Four curves an accurate way of making a circle but it can be done perfectly with just two curves as shown: Control points are always at a distance of the radius plus a third away from the top and bottom of the circle.



*A circle made of two curves*

Put another way, a third of the radius inward from the edges of the boundary box which encloses the circle. There is probably



*Some boxes with nicely rounded corners (top) and used with path merging and masking to give a photo rounded corners.*

# 'Gons, Stars and Cogs

Polygons, stars and cogs are three examples of shapes which would be nearly impossible (and certainly take a very long time) to construct accurately by hand in Draw.

But with a little trigonometry and using our knowledge of the Drawfile format they are easy to do in BBC Basic. In all cases we 'sweep' round a circle (from 0 to 360 degrees) joining several points on the circumference of the circle. In the case of stars and cogs some points are joined to the circumference of an imaginary inner circle.

The following diagrams show pictorially how this is done. Angles are in degrees because this is what we are all familiar with. But as we discussed in a previous chapter they have to be converted to RADians when calculating SINES and COSines in BBC Basic.

## Polygons

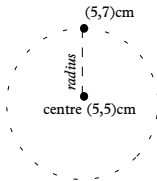
Pentagon sides=5

xcentre=5, ycentre=5cm  
radius=2cm

step=360÷sides =360÷5=72°

### Step 1.

Move to (xcentre,ycentre+radius)

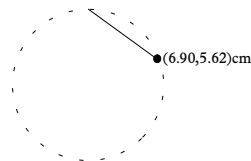


### Step 2.

set angle = step

### Step 3.

Draw to (xcentre+radius\*SINangle,  
ycentre+radius\*COSangle)

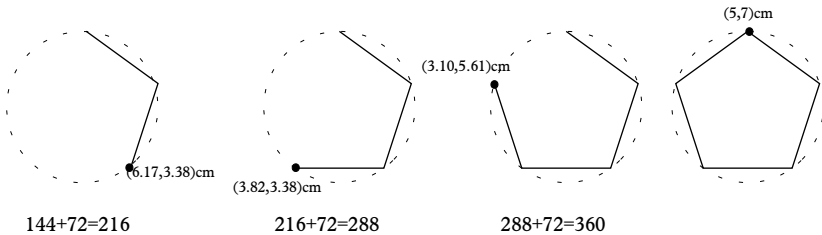


### Step 4.

Add step to angle  
72+72=144

### Step 5.

Repeat steps 3 and 4 until all sides drawn.



*Constructing Polygons*



## Converting Foreign Files

You may have found drawings on the internet which aren't compatible with Draw. They are stored in a format used by other computers and a common example is the Windows Meta File (WMF).

The quality of these files isn't high quality as Draw files because there are no curves. Circles and the like are actually made up of lots of little straight line sections. But they are arguably more efficient because only *half words* (2 bytes, 16 bits) are needed to store co-ordinates. WMFs are used in Microsoft Word and Powerpoint, as well in numerous clip-art collections on CD-Rom so it's useful to be able to translate them into Draw format.

This chapter lists a simple WMF-to-Draw converter. It can convert polygons, correctly translating line width, line and fill colours. This is sufficient for most purposes. Sometimes, though, overlaid polygons will be in the wrong colour but you can easily edit the resultant Drawing, called *WMFDrawing*.

Like Draw, WMFs have a *header* followed by a number of *objects*. The header begins with a 'magic number' (&9AC6CDD7) identifying it as a WMF, rather like Draw begins with &77617244 except less memorable: the first word of Draw are the Ascii characters D r a w.

The only useful piece of information in the header is at offset &0E. It gives the number of Metafile units per inch, for example 1440. Thereafter coordinates are quoted in these units, and to translate to Draw we divide every co-ordinate by the Metafile units per inch and multiply by Draw units per inch (46,080).

The program knows about six types of WMF object. The main loop scans through the file picking up the object type and size. The size is the number of halfwords the object occupies in memory. If the type is not recognised the program skips forward to the beginning of the next object.

Object types &20B concerns the origin and PROCorigin extracts and stores in *ox%* and *oy%* in the program. Object &20C specifies the dimensions of the drawing (*ex%* and *ey%*) rather like the boundary box in Draw.

We need *ox%* and *oy%* to add to any coordinates. Y

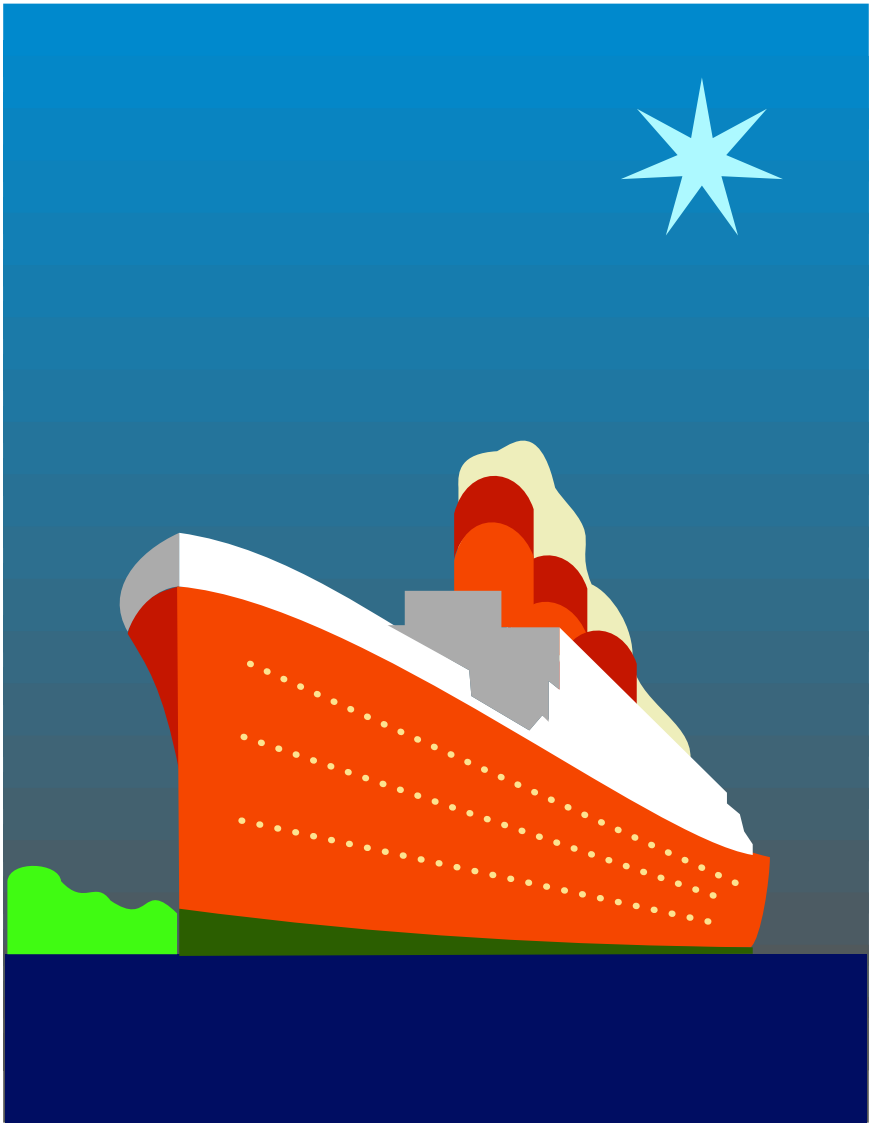
## Path Merger

Sometimes it is desirable to join two or more shapes to form a single path. For example, if you tried to design a typeface (in the *Designing Fonts* chapter) you will have realised that some letters had inner shapes which had their fill colour set to white which then meant the letter could really only be placed on a white background (or of the same colour as the fill).

Of course, the Move tool in Draw can be used to build up a single path of several subpaths but only if you were constructing a shape from scratch.

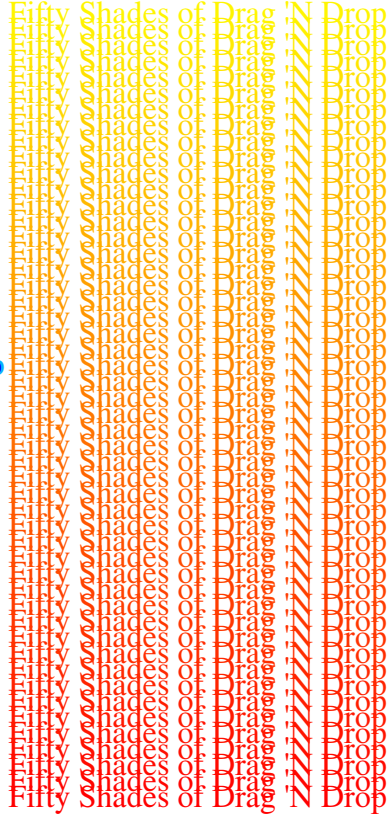
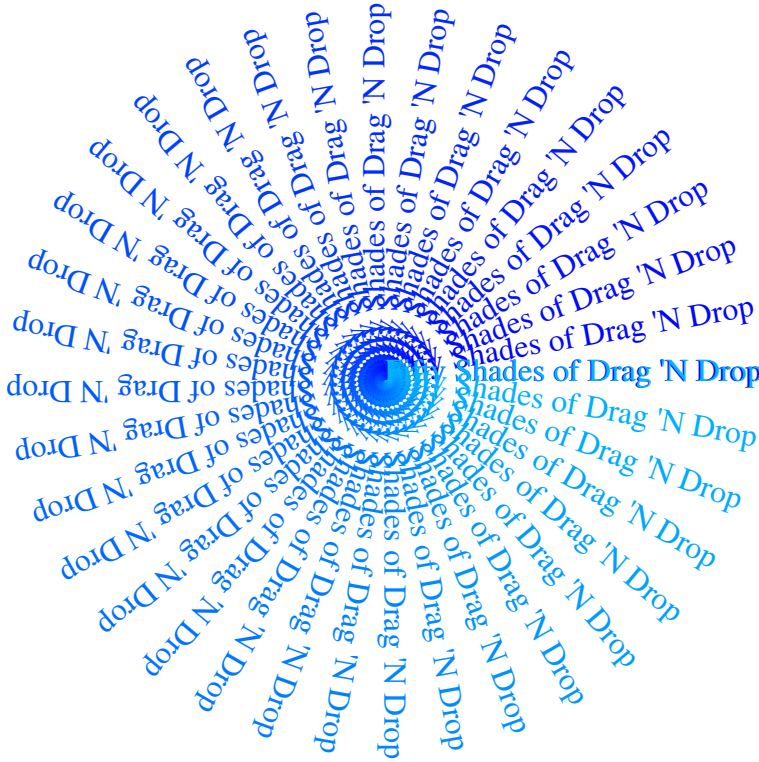
The below dump, beginning at offset &80 into the Draw file, shows two objects - both simple, one-section straight lines.

0080	02	00	00	00	Object 2 path
0084	44	00	00	00	Size=&44 bytes
0088	E0	46	00	00	Bounding
008C	E0	46	00	00	.
0090	60	62	01	00	box
0094	E0	46	00	00	.
0098	FF	FF	FF	FF	Fill=none
009C	00	00	00	00	Outline=black
00A0	00	00	00	00	Linewidth=thin
00A4	42	00	10	20	Path style
00A8	02	00	00	00	Move to
00AC	E0	46	00	00	X=1cm
00B0	E0	46	00	00	Y=1cm
00B4	08	00	00	00	Draw to
00B8	60	62	01	00	X=5cm
00BC	E0	46	00	00	Y=1cm
00C0	00	00	00	00	End path
00C4	02	00	00	00	Object 2
00C8	44	00	00	00	Size =&44 bytes
00CC	E0	46	00	00	Bounding
00D0	C0	8D	00	00	.
00D4	60	62	01	00	box
00D8	C0	8D	00	00	.
00DC	FF	FF	FF	FF	Fill=none
00E0	00	00	00	00	Outline=black
00E4	00	00	00	00	Linewidth=thin

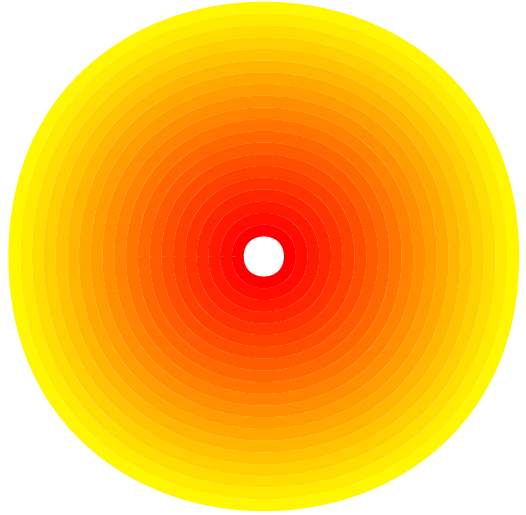
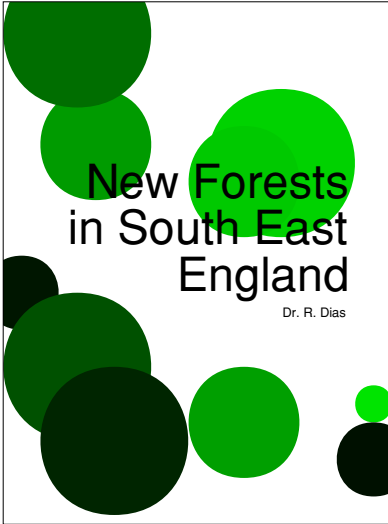


# BLUE STAR LINE

*Vintage travel poster, designed using techniques in the first half of the book.*



*Graded and rotated text from the "Text Effects" chapter*



*Graded and random circles from the Working with Curves chapter.*



*“BEACH” was typed in 72pt Homerton Bold, converted to a path, ungrouped, the arrow drawn around them, and then the Drawing was imported into the path merging program. A graded-colour rectangle (see Projects (4) chapter) was then produced and put behind the letters/arrow shape to create the finished effect.*



*Colour chart generated by the listing in the More Draw Colours chapter. Export the desired coloured squares and apply to your own shapes as described in the Format Painter chapter.*

# The Book of Draw Stuff for RISC OS computers

Whether you are a user or programmer of RISC OS, have an old RiscPC or new Raspberry Pi, this book is for you.

The first section builds on techniques in the *User Guide*: creating grids quickly, accurate positioning of objects, hatching, graded backdrops and format painting, with projects to design CD labels, fonts and a vintage travel poster.

The second section examines the structure of Draw files and shows how you can generate complex diagrams with repetitive or complicated patterns, just using BBC Basic v. There are listings to generate graded text, tables from CSV files, a Windows Metafile (WMF) converter, extra colours and dashed patterns, producing text along curved paths, and a utility to merge paths.

Both Draw and BBC Basic v. are supplied free with every copy of the operating system and this book is for everyone using RISC OS 3.10\* and above.

\*Subject to memory limitations on earlier machines.

ISBN 978-1-78280-501-4



9 781782 805014 >



:0



Discs



:0



16GbPi



Apps

**DRAG'N'DROP**

[www.dragdrop.co.uk](http://www.dragdrop.co.uk)

