4

Edit and Draw: Budget DTP

When looking for DTP software it is perhaps inevitable to dwell on the features that the competing packages offer for the prices. Indeed, Chapter 2 listed the various virtues of Archimedes DTP software and by implication their suitability for certain tasks. Yet it is quite possible to achieve results almost identical to ‘proper’ DTP without any expense at all, though rather less conveniently: the applications *Edit* and *Draw* that come free with every Archimedes can be used together to good effect and can give you a first taste of DTP.

Don’t be misled by the price: with this combination you can produce pages that make use of the outline fonts in any size, in multiple columns, and with graphics wherever and however large or small you want them. What you don’t get is very easy control over how text and graphics fit together. You have to plan your document fairly carefully before you start, in terms of the page size, the number of columns to a page, what font(s) you will use and in what sizes. However, because RISC OS allows different applications to work interactively, it is relatively easy to make changes to the text in *Edit* and then see their effect almost at once on the page by ‘saving’ the text into *Draw*.

Text handling

Code Control

Fairly obviously you need to enter some text before you can put it into *Draw*; in fact it isn’t possible to edit text once it has been inserted into a *Draw* file. Each text file for importing into *Draw* can be thought of as a separate *story*, and indeed this term is widely used in DTP.

Edit, although not a word processor as such, is perfectly adequate for text preparation. One of the main features it lacks is 'word wrap', whereby words are prevented from being split over two lines; however, after using *Edit* for a while you readily adapt to seeing bits of words at both ends of lines.

‰*Draw* needs much more information than *Edit* to make a success of putting text on the page. DTP programs use menus to give you a choice of text fonts, sizes and arrangements. In contrast, *Draw* asks for all this information to be placed in coded instructions in the text file itself, so the first part of any text to be put into *Draw* should be a 'header' containing a set of these instructions; others can be scattered through the text as the occasion arises.

The full list of codes is shown in Figure 4.1. Don't be intimidated by them; we shall take them in easy stages. Notice that they all begin with a backslash, \ , to distinguish them from the text itself, and each one should normally appear on a line of its own. The capital letters used are significant: lower-case letters won't work. Where parts of the command are in italic, these are variable elements that you should replace with a value (number or font name).

`\!1`	'this is text'	`\M`*left right*	left and right margins
`\AC`	centred text	`\P`*n*	space between paragraphs (points)
`\AD`	justified text		
`\AL`	left aligned text	`\U`*p t*	underline position and thickness
`\AR`	right aligned text		
`\B`*r g b*	background colour	`\U.`	turn off underlining
`\C`*r g b*	text colour	`\V`*move*	vertical move
`\D`*n*	number of columns	`\-`	soft hyphen
`\F`*n font s w*	specify font, size and width	`\\`	insert backslash
		`\<newline>`	force new paragraph
`\`*n*	use font/size no. *n*	`\;`	comment line
`\L`*n*	line spacing (points)		

Figure 4.1 *Codes for formatting text imported into Draw*

The first thing is to tell *Draw* that the item you are importing is to be treated as text. This is done with the introductory command `\!1` (the third character is a numeral one, not a letter 'ell'). Next you will need to alert *Draw* to the fonts you will want to use in the whole text, and at what sizes. For now we shall use just one as an example, Homerton.Medium, and specify it as being at a size of 12pt by the command `\F0 Homerton.Medium 12` (yes, it really is that simple). Because no number is given for width in the example, *Draw* takes it to be the same as the height. If you give different numbers for height and width, the letters will appear stretched or squashed.

The zero after the F in the font specification serves to identify that font at that particular size when you want to use it, and you would summon up the example font with `\0`. You can have up to 99 font/size specifications in one text file. To complete the basics, we shall use `\L14` as an example of how to set the line spacing at 14pt.

That completes the codes for text specification; now for the layout. As illustrated in Chapter 7 (page 129), there are four sorts of justification, and *Draw* gives access to all of them. So `\AL` gives the text a smooth left margin (and a ‘ragged’ right margin), `\AR` gives the opposite effect (a ragged left margin), `\AD` makes both margins even by justifying lines to their full length (like the text in this book), and `\AC` centres lines between the two margins – a sort of ragged left and right.

You may wish to use `\P` to insert some space between paragraphs to break the text up into manageable chunks. It is certainly easier on the eye not to have a solid block of text running down a page. To insert a space of 8pt between paragraphs the command would be `\P8`. If you omit a `\P` command, 10pt will be inserted automatically, so if you don’t want this you must put `\P0` in the header.

You can set up margins within a column of text with `\M`*`left right`*, where *left* and *right* are the distances in points from the edge of the column (not the page). This is not normally very useful if you don’t plan to change margins within a document, because (as we shall see) you can always alter the size of the text area once you are working in *Draw*. However, it is handy in the middle of text if you want to indent a few paragraphs, for example for emphasis or as quotations. So, to give a 2 em indent to both sides of a paragraph in 12pt type, you would use `\M24 24` at its start and `\M0 0` at the start of the next paragraph. You can also produce hanging indents or indent the first line of a paragraph with this command, if you embed it in the text.

The *User Guide* says that the `\D` code is used to tell *Draw* how many columns of text you want. In fact, you will do better to think of it as specifying how many separate ‘frames’ holding text there are to be on the page: they don’t have to end up side by side. It is a good idea to set it to a higher value than you immediately expect to use, say `\D6` for a double-column page, to give you maximum flexibility as you work in *Draw*. You can, of course, increase the number later if you find that you haven’t made enough frames, but that is less convenient.

Other commands (`\B` and `\C`) give control over the background and text colours respectively. The three numbers *r*, *g* and *b* are the red, green and blue components of the colour you want, each of which can vary between 0 and 255. Think of them in much the same way as the slider bars in the Palette window; the only difference is that you can't alter the palette with these commands, but only choose one of the palette colours. For instance, the command `\B0 0 255` will define the background colour as blue, or the nearest colour available in the palette. Unless otherwise defined, the background is, rather sensibly, set to white (equivalent to `\B255 255`

`255`) and the text colour to black (i.e. `\C0 0 0`). This is one of those commands that you learn best in detail by experimentation.

Apart from changing font, a way of adding emphasis is by using underlining with `\Up t`, where *p* is the position of the underline and *t* is its thickness; both are measured in units of 1/256 of the font size. The underline position runs from −127 to 128, with negative values being below the baseline and positive ones above it, so a position of 0 would run through the letter bases. The thickness ranges from 0 (invisible) to 255 (the whole font height). So `\U-50 25` would produce a reasonably thick underline well below the baseline but still passing through the descenders of lower-case letters.

Turning underlining off is easy: the separate instruction `\U.` does this, as would `\U0 0` by setting the line thickness to zero.

Occasionally you may want to print subscripts or superscripts. Strictly, these should be in smaller type (for which you can use a font change), but they also have to be moved up or down from the baseline. This is done with `\Vmove`, where *move* is measured in points up or down relative to the current character position and can range from −9 to 9. If `\V9` or `\V-9` don't provide enough of a shift in either direction you can use two or more commands in succession, but remember to cancel them with an opposite movement of the same amount afterwards.

Next, a few specialised commands: first, `\-` inserts a ‘soft’, or optional, hyphen in a word to enable it to be split and hyphenated if necessary. If the soft hyphen is not used to break a word, no hyphen is printed, as opposed to a ‘hard’, or ordinary, hyphen. Secondly, the sequence `\\` is interpreted by *Draw* as a single backslash in the text, in case you need one. Thirdly, your *Edit* text can contain comments, notes or asides to yourself that will not be printed: the `\;` command is used for this. All the text after it up to the next newline (i.e. what you get when you press Return in *Edit*) will effectively be ignored on importing into *Draw*.

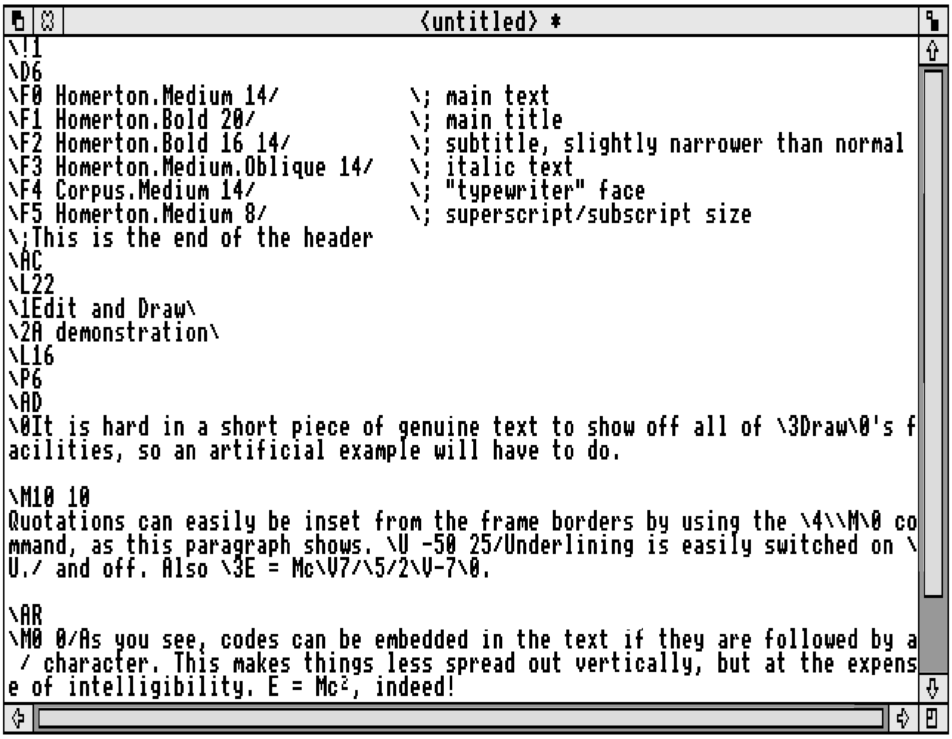
Finally, a few words on text formatting and how *Draw* understands paragraphs: pressing Return once only at the end of a line to make a new paragraph is all right in *Edit* but when *Draw* imports the text it converts each single newline into a space character. This is necessary because some word processors (though not *Edit*) insert a newline at the end of every line and not just when you press Return. There are two ways of dealing with this. The first is to press Return a second time at the end of each paragraph (and each heading line) so that there is a blank line between each: *Draw* will interpret this as meaning that a new line is required, and each further consecutive Return in the imported text will count as one more blank line in *Draw*. Alternatively, you can end each paragraph with a `\` character just before pressing Return, to force *Draw* to start a new line.

Although, in the header, it is convenient to put all the codes on separate lines, when it comes to putting them in running text you may want to embed them in your text instead. This avoids having to break up a paragraph into what looks like lots of pieces. If you follow a code with text, put a `/` (slash) character after the code to tell *Draw* that it is concluded.

One thing you will almost certainly forget to do at first is to end your text by pressing Return: if your text doesn't finish with a newline, *Draw* will complain when you try to import it and will abort the attempt. No, I don't know why, if it knows that there isn't a newline, it doesn't simply add one; that is just the way it is.

Using the Codes

After that brisk run through the codes, let us see how they are used by looking at a piece of demonstration text. Figure 4.2 shows some text from an *Edit* window, with a variety of commands. Notice that the font changes, such as `\0`, are always followed immediately by a `/` character or the text to be set in the designated font: any space after the number would be treated as part of the text. For clarity, all the commands in



```
\!1
\D6
\F0 Homerton.Medium 14/          \; main text
\F1 Homerton.Bold 20/            \; main title
\F2 Homerton.Bold 16 14/         \; subtitle, slightly narrower than normal
\F3 Homerton.Medium.Oblique 14/  \; italic text
\F4 Corpus.Medium 14/            \; "typewriter" face
\F5 Homerton.Medium 8/           \; superscript/subscript size
\;This is the end of the header
\AC
\L22
\1Edit and Draw\
\2A demonstration\
\L16
\P6
\AD
\0It is hard in a short piece of genuine text to show off all of \3Draw\0's f
acilities, so an artificial example will have to do.

\M10 10
Quotations can easily be inset from the frame borders by using the \4\\M\0 co
mmand, as this paragraph shows. \U -50 25/Underlining is easily switched on \
U./ and off. Also \3E = Mc\V7/\5/2\V-7\0.

\AR
\M0 0/As you see, codes can be embedded in the text if they are followed by a
 / character. This makes things less spread out vertically, but at the expens
e of intelligibility. E = Mc², indeed!
```

Figure 4.2 *A sample of text for export from Edit into Draw*

the header, as well as the commands before the start of the text, are shown on separate lines.

By referring to the previous section you should be able to work out what will happen when you import this text into *Draw*. In fact, to get the full effect several stages are needed, as we shall now see.

Importing Text into *Draw*

With a 1Mb Archimedes it is just possible to run both *Edit* and *Draw* together, so if you want you can transfer text directly from one to the other without saving to disk first, although for safety's sake it is impossible to save too often.

Before importing text into *Draw*, whether from disk or direct from *Edit*, you should first set up a page to receive it. Click on the Draw icon to open a window, then click Menu in the window to call up the main *Draw* menu; on the Misc submenu there is an item 'Paper limits', as shown in Figure 4.3. Choose from the range of A paper sizes shown and also between portrait and landscape orientation (see pages 10 and 11 if these terms baffle you). If you have a RISC OS printer driver loaded, then choosing 'Show' will cause a grey border to appear within the page area. Anything inside the border is the printable area corresponding to the printer driver loaded, so don't put anything that you want to be printed into the grey area.

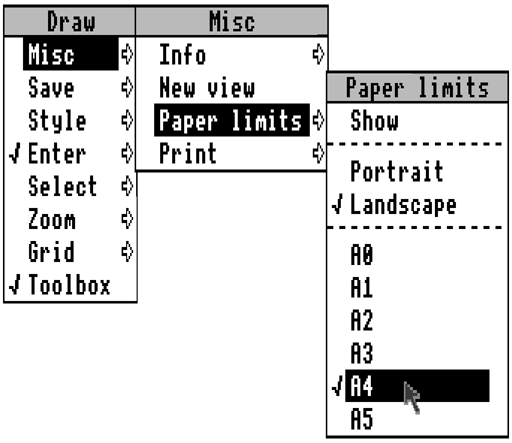
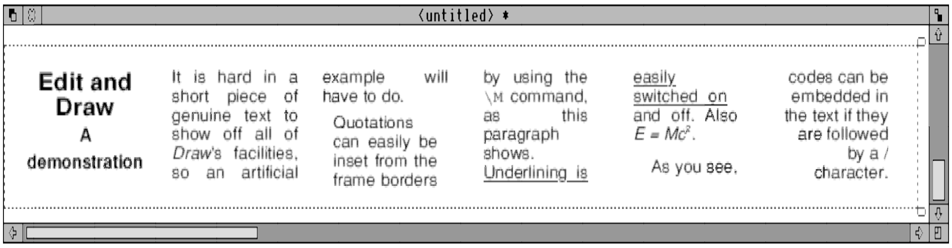


Figure 4.3 *Setting paper limits in Draw*

The actual transfer of text into *Draw* is, as you might expect, as simple as dragging the file's icon from disk, or the 'Save' icon from *Edit* or another application, into the *Draw* window. The initial result, shown in Figure 4.4 for the demonstration text,



Edit and Draw

A

demonstration

It is hard in a short piece of genuine text to show off all of *Draw*'s facilities, so an artificial example will have to do.

Quotations can easily be inset from the frame borders by using the \M command, as this paragraph shows.

Underlining is easily switched on and off. Also *E* = *Mc*^{2}.

As you see, codes can be embedded in the text if they are followed by a / character.

Figure 4.4 *The start of the text from Figure 4.2, as first seen in Draw*

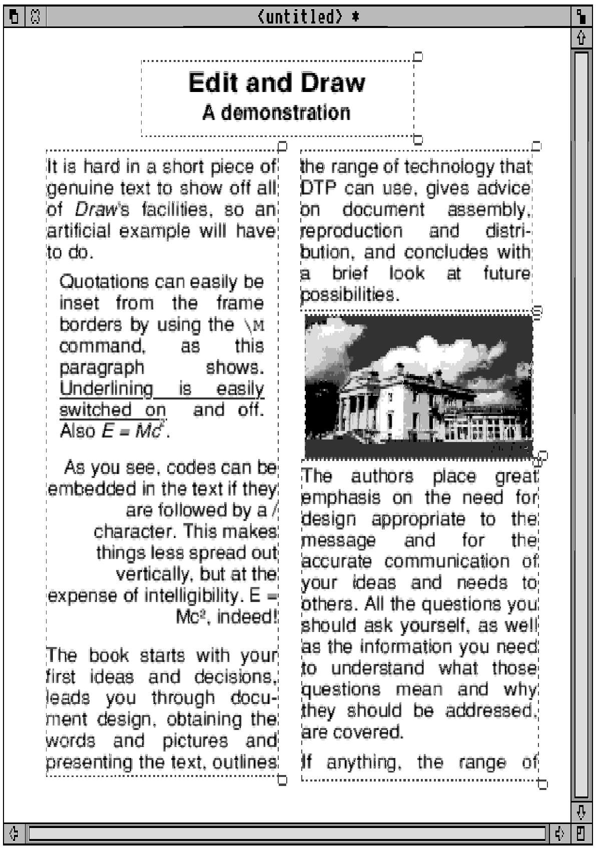
may not impress you: only a small part of the text is likely to appear, although in the correct number of columns as specified by `\D`. However, all the text is now available for manipulation by *Draw*; it merely needs to be brought into view.

To do this, click on the bottom *Draw* tool (the pointer, for select mode) and then click once on the text. A dotted outline will appear, showing the full extent of the text area. You can move the text around by dragging on it. If there is only one column you can also alter the size and shape of the text area by dragging the ‘ear’, or ‘handle’, at the bottom right of the area, but if there is more than one column this won’t work; if you try it you will find that it seems to work while you are dragging, but the effect is ignored once you let go. Incidentally, the ‘rotation’ handle at the top right of the outline doesn’t work at all with text, though it does no harm to try.

So, there has to be another way to change the size and shape of the columns of text if there are more than one: it is done by double-clicking on a column to select it, and then using the bottom handle to resize it. Note that you can drag a selected column anywhere on the page, even getting them out of their original order, but *Draw* will still remember what the order was to begin with and flow the text between columns accordingly.

You can also have more than one column selected at the same time by double-clicking in a slightly unusual way: the first click on the column you want to add to the selection should be with Adjust and the second should be with Select. This is useful for selecting several columns so that they can be moved as one, or for keeping them equal widths or heights during resizing.

Figure 4.5 shows the text after it has been arranged, by resizing and dragging the columns, into a simple double-column A5 page, with a full-width column for the title and a gap in the second column for a picture to be inserted. Therefore, only four of the six columns originally specified have actually been used; the other two have disappeared because there was no text in them. In this way you can see how ‘columns’ become ‘frames’ holding the text, and for the rest of this chapter I shall refer to them as frames.



Edit and Draw

A demonstration

It is hard in a short piece of genuine text to show off all of *Draw*'s facilities, so an artificial example will have to do.

Quotations can easily be inset from the frame borders by using the \M command, as this paragraph shows. Underlining is easily switched on and off. Also *$E = Mc^2$*.

As you see, codes can be embedded in the text if they are followed by a / character. This makes things less spread out vertically, but at the expense of intelligibility. E = Mc^2, indeed!

The book starts with your first ideas and decisions, leads you through document design, obtaining the words and pictures and presenting the text, outlines the range of technology that DTP can use, gives advice on document assembly, reproduction and distribution, and concludes with a brief look at future possibilities.

The authors place great emphasis on the need for design appropriate to the message and for the accurate communication of your ideas and needs to others. All the questions you should ask yourself, as well as the information you need to understand what those questions mean and why they should be addressed, are covered.

If anything, the range of

Figure 4.5 *A simple (nay uninspired) layout of the demonstration text; notice the large spaces between words in justified text, caused by the excessive type size*

Before we move on to the next section, on assembling a whole document, it is worth mentioning a couple of limitations of *Draw* with text. They arise from the way that the software positions text within a frame, and from the fact that the edge of a text frame is treated as the boundary between what is printed and what is not, as far as text is concerned. Put more simply, if for any reason some text or part of a letter goes outside the frame, then either part or all of it will not be printed. There are two ways in which this can happen.

The first, which can be a source of great bewilderment until you realise what is going on, is when the line spacing is somewhat smaller than the font height: the top line of every frame will be missing. The second comes about because some characters, particularly in italic fonts, are wider than the character width and consequently overhang onto the previous or next character. The software assembles a line of type by adding together all the character widths on the line and quite correctly ignores any overhangs (they are called *kerns*) for this purpose. It is therefore possible for a letter to overhang beyond a frame, causing part of it to be clipped off, as Figure 4.6 illustrates.

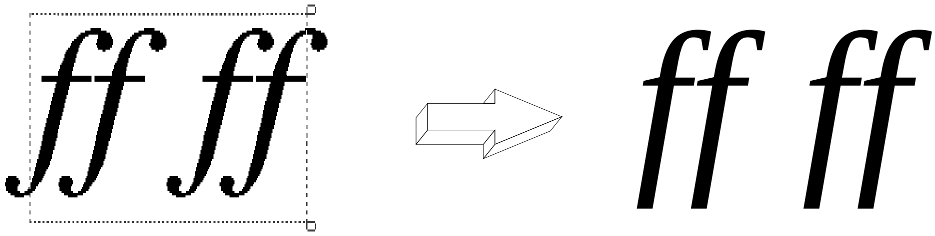


Figure 4.6 *The ‘clipping’ effect of overhanging a text frame in Draw*

The solution to the first problem is obvious: just set the line spacing appropriately for any top lines affected. The second situation can be cured, if it happens, by creating margins between the text and frame edges with the `\M` command.

Assembling a Page in *Draw*

The problem with the example page in Figure 4.5 is that it was prepared *ad hoc*: if I had wanted to put together a newsletter or booklet on a two-column design, it would have been hard to achieve any sort of uniformity or consistency through the pages. So, the first task on any document, especially if it is repetitive over time or has several pages, should be to devise a template for a typical page. If it is going to be unique, a one-off, you may not feel like following all the pointers here but, as Chapter 9 points out, effective design doesn’t just happen, it is carefully planned. Spend a few minutes putting down your visual thoughts with a pencil and a sheet of paper if you do nothing else.

The Page Template

Once you have decided what sort of publication you are going to produce, how often, with how many pages and with what degree of consistency, you will be getting ideas about what each page should look like. The best way to achieve consistency is to borrow a technique from commercial DTP software: *guide frames*. These are frames that will not be printed but which help you position your text frames easily. Unfortunately, *Draw* doesn’t have a ‘guide frame’ object that is visible on screen but won’t print, so we must improvise.

Having chosen your paper size and orientation, go into the Grid submenu, make sure that ‘Show’ and ‘Lock’ are selected, and choose the rectangular grid size that suits you. The options are inches or centimetres as the basic units, and you can specify how many subdivisions of either are shown. Now you can measure your way across the page and set up guide frames in the knowledge that they will snap onto the grid markers, making accurate positioning easy. To make the guide frames distinctive, they can simply be white rectangles with thin orange borders: choose the rectangle tool and then go to the Style submenu to set the line thickness to thin, the line colour to orange and the fill colour to white.

Then draw your rectangles to show where you will want to place your text areas, not forgetting positions for titles, page numbers and page headings; you can also indicate locations for logos, illustrations and anything else you will often be needing. If you feel adventurous you can use other border colours for these different items.

To alter the sizes of items you have drawn you need to be in select mode, then select the object and drag the handle at the bottom right of the frame. Once you are fairly happy with your layout, save it to disk. Now you can print it out (see the page opposite) and check whether your ideas on positioning look right on paper, refining them if necessary. Don’t forget to keep saving the latest version. When all is well, click Adjust on ‘Select all’ in the Select submenu, then click on ‘Group’. This will combine all the guide frames into one object.

You may want to add rules between columns or below page headings: you can do that now. Click on the top tool, for straight line paths, and set the line width appropriately. Once you have drawn the lines, *don’t* group them together with the guide frames before saving the file to disk.

This, then, is your page template: keep it somewhere safe, preferably on more than one disk.

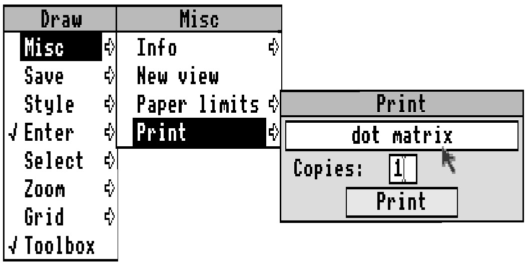
Putting together Text and Pictures

Thanks to *Draw*’s ability to handle both text and pictures, the actual assembly of a real page is fairly straightforward. We have already looked at importing text and moving the text frame, or frames, around; with the help of the guide frames and the grid lock option (you don’t need to be able to see the grid now, if it gets in the way) it shouldn’t take too long to lay the text out as you want. Pictures, like text, are imported by dragging files, and moving or resizing them is also done in select mode. *Draw* files can also be rotated at this stage, but rotating sprites can only be done in *Paint*; this is slow, and large sprites may cause *Paint* to run out of room in the process.

So, eventually, your page is ready for printing – well, not quite, as there are two things still to be done. Before you do anything else with it, save it under a new name (not the name of the page template file). Printing it now would probably print bits of the guide frames too, so these must be disposed of now that they are no longer needed. In select mode, click repeatedly within the page area until you have selected the object that is the group of guide frames (now you see why they were grouped!), then click on 'Delete' in the Select submenu. The guide frames should now disappear, leaving the page ready for printing after saving again.

Printing

The Print dialogue box, which you find via the Misc submenu, is shown in Figure 4.7. Just enter the number of copies you need, check that the dialogue box contains the name of a suitable RISC OS printer driver (which proves that you have loaded one), and click on the OK box. Chapter 11 gives further advice on working with the RISC OS printer drivers.



dot matrix

Copies: 1

Figure 4.7 *The Print dialogue in Draw*

Hints and Tips

Draw can't handle more than one piece of paper in any one file, although you can, for example, put two A5 pages side by side on a landscape A4 sheet so that they will be printed sideways. As a result, a story may overrun the space available on one piece of paper. All is not lost, however: once you have finished arranging one page go back to the text in *Edit* and take out everything on the page you have just completed (leave the header codes at the beginning of the text, though), and then save it under a different name from the original text. In this way you will whittle down the story until it has all been put onto pages.

You may find it easier to put titles and other special elements such as picture captions and footnotes in separate files from the main text: treating each type of material as a different story, each in its own frame or set of frames, will give you more flexibility in placing them on the page than if they are all part of the same story in one set of interlinked text frames.

For both coarse and fine control of positioning, you can work at two magnifications simultaneously by using the ‘New view’ option in the Misc submenu to give you two windows on screen. One of the windows can be made quite small by setting the zoom factor to 1:3 or smaller for an overall view of the page, while the other window can be set to 2:1 or larger for close-up work. Any changes you make in either window will be reflected straight away in the other.

To ensure that multiple-column guide frames are of equal sizes, create one column first and use ‘Copy’ in the Select submenu as many times as you need. The copies will be aligned correctly with the grid if the ‘Lock’ option is on.

If you have laid out a page and only then discover a mistake in text you have imported, you don’t have to start again from scratch: all you need to do is correct the text in *Edit* and re-import it into one of that story’s frames. The new story will replace the old.

The last step in preparing a *Draw* file for importing should be to go into select mode and group all the objects. That way, once you have imported it into your document there will be no risk of its ‘coming apart’ during moving, resizing or rotation.

It is possible to export a story from *Draw* as a text file; just click once on the text to select it all, then go through the Save submenu to the ‘Text area’ dialogue box and save in the normal way. If ‘Text area’ is greyed, you haven’t selected the whole story.