

3

DTP with Word Processors

Between simple word processing and ‘real’ DTP there is a grey area where the two meet. In this chapter we shall look at three word processors, *1st Word Plus*, *PipeDream* and *EasiWriter*, that can make use of computer-based graphics and can therefore give DTP-like results, ranging from simple to quite complex. Each is better at some things than others for the purpose, though all are designed for word processing rather than DTP by the definition I give it in Chapter 6.

As you might expect in a book about DTP I shall spend comparatively little time on the word processing angle and concentrate on the marriage of word processing and graphics capabilities. In looking at *EasiWriter* I shall also consider the concepts it embodies that bridge the divide between word processing and DTP.

1st Word Plus

This is basically a straightforward word processor with a limited capacity to include pictures within a document. Only fairly basic text effects (italics, underline, condensed, enlarged, etc.) provided by a dot-matrix printer are available with *1st Word Plus*; it can’t enhance your document’s appearance by using the Archimedes’ outline fonts. Pictures can be printed out as long as your printer is set up correctly and has a graphics mode. This may seem rather primitive alongside other programs to be described in this book, but it is a beginning.

Page Layout

Leaving aside the mechanics of typing in text, inserting an illustration in a *1st Word Plus* document is a relatively simple matter of working out where you want it to go, dragging the icon for the picture file (a sprite, not a *Draw* file) into the document window, fine-tuning its position and reformatting the text around it if necessary.

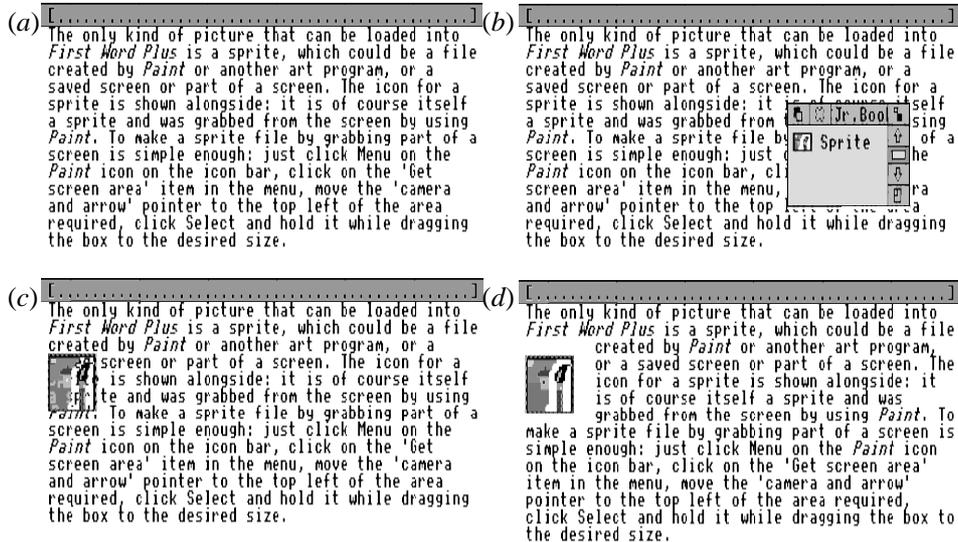


Figure 3.1 Stages in incorporating a picture into a *1st Word Plus* document

The sequence of operations for a slightly unusual layout is shown in Figure 3.1. In (a) is the plain text containing a reference to a picture, as it would appear in a *1st Word Plus* document window. Panel (b) shows a directory window containing the sprite file called 'Sprite', opened on top of the document window for the purposes of demonstration. Dragging the file from the directory window into the document switches *1st Word Plus* into graphics mode automatically and gives the result shown in (c), where the picture has been overlaid on the text, partly obscuring it. After that it is merely a matter of reformatting the text to flow round the picture, as in (d).

The picture is loaded into the document at the position of the cursor. If it turns out not to be in exactly the right place, just drag it around with the mouse. Positioning is a little coarse because a picture can only be moved in units of whole text characters, both horizontally and vertically. In fact this can be an advantage, as it makes it easier to line up pictures and text neatly.

1st Word Plus scores with its simple controls for reformatting the text, even though two stages were needed for getting a clear space around the picture in Figure 3.1(d) by adjusting lines three to seven. First, lines three to the end of the paragraph were shortened by putting the cursor at the start of the third line of text, dragging the left margin on the ruler rightwards to what looked like a good place and then pressing F7 to reformat the paragraph. Then lines eight onwards were lengthened again by putting the cursor at the start of line eight, dragging the left margin back to its original position and pressing F7 again.

You can only see the pictures in the document while *1st Word Plus* is in graphics mode: turning graphics mode off makes the pictures disappear, although their positions are marked by outlines. Doing this saves memory and makes scrolling round the document quicker.

Deleting pictures can be done in one of two ways. One is simply to click on the picture to move the cursor to the start of a line containing it, then click on 'Delete picture' in the Graphics submenu. The other, which is useful if you want to remove text as well, is to mark the piece of text where it is embedded, then use 'Cut' in the Select submenu.

Remembering that *1st Word Plus* is a word processor, not a DTP package, will help you not to forget that pictures in a document are not held as part of the document itself, but only by reference to whereabouts the picture files can be found on disk relative to the document file. This means that it is best to put the document and any picture files in the same directory, and if you want to move or copy the document you must always move or copy the picture files as well.

Another consequence of *1st Word Plus*'s being a word processor is that you cannot alter the size of pictures in a document, making it unlike all the other programs discussed in this book. Simplicity has its merits; it also imposes limitations.

Printing

1st Word Plus doesn't make use of the RISC OS printer drivers; to make up for this it comes with a good number of printer drivers, giving access to a wide range of printers. A printer driver is loaded every time you start up the *1st Word Plus* application: it is the one called `1wp_print` in the `!1stWord+` application directory (you can open it to look at the files inside by double-clicking on the icon while holding down one of the Shift keys).

If the default printer driver doesn't match your printer there are two ways of loading a different one. The first is useful for checking out all the drivers to see which of them, if any, is suitable: just open the directory `1WP_Print` and then its subdirectory `Config`, and drag one of the drivers shown on to the *1st Word Plus* keypad.

The second method, which ensures that the correct driver is loaded every time you run *1st Word Plus*, involves deleting the existing `1wp_print` file from the `!1stWord+` directory, copying the correct driver from the `1WP_Print.Config` directory to the `!1stWord+` directory to replace it, and renaming it `1wp_print`. Be sure to distinguish between the main directory `1WP_Print` and the file `1wp_print`: the confusing nomenclature is unfortunate.

If none of the drivers is suitable, you will have to alter the source file of a similar driver so that it matches your printer's commands. This is more than a little complicated (as well as being unlikely) and is dealt with well in the manual, so it needs no explanation here.

Unusually for a RISC OS application, *1st Word Plus* cannot print a document that is held in memory: it must first be saved to disk and printed from the saved file. Setting up the 'Print file' dialogue box is also well explained in the manual: comment here would be superfluous.

It is only when printing graphics with this program that you are likely to discover that your printer isn't configured correctly as regards extra line feeds: for nearly two years I had been happily printing graphics on my dot-matrix printer with all manner of Archimedes software, with never a hitch. It was only when I came to use *1st Word Plus* that I found blank lines appearing between successive lines of printed graphics. So take note, and set the 'Auto linefeed after carriage return' option on your printer to OFF.

PipeDream 3

In 1987 Cambridge Computer Ltd brought out a revolutionary laptop computer, the Z88. Its main 'workhorse' program was a powerful custom-built spreadsheet/database/word processor called *PipeDream* – all in a machine the size of an A4 sheet of paper. This is a curious mixture of functions for a software package, you might think, but it proved surprisingly popular and after an incarnation on the 8-bit BBC Micro series a much faster and more powerful version was developed for the Archimedes.

For these historical reasons, *PipeDream 3* has inherited all the functions of its ancestors, and more have been added; but running on the Archimedes with RISC OS menus makes it very much faster and easier to use. As a word processor it could be described as 'quirky but powerful'; if you have *PipeDream 3* for its spreadsheet capabilities you probably have no need for a separate word processor. In addition, you can include graphics, which is why it is described here.

Page Layout

As with *1st Word Plus*, this is controlled in ruler fashion by setting left and right margins and the number of lines to a page. The concept of 'slots', unique to *PipeDream* as a word processor, will probably only concern you in the sense of different columns of text if you want to prepare tables. Figure 3.2 shows the unique document window of *PipeDream*, with the cursor in the first slot.

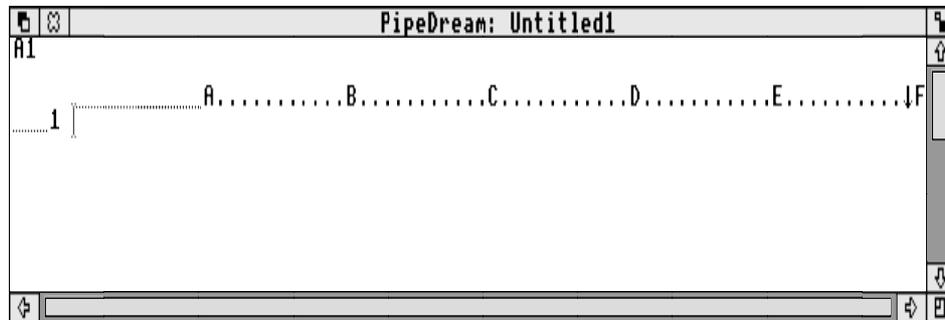


Figure 3.2 The default document window of *PipeDream*

PipeDream as a word processor mostly makes use of column A; in fact, if you don't intend to use tabulation you will probably find it safer to remove all the other columns with the 'Delete column' command in the Edit submenu, for which the key shortcut is Ctrl-EDC. The extensive range of key shortcuts, a cross between the idea of a 'hot key' for different functions and a mnemonic for each command, is another feature that has been carried over and extended from the program's origins.

It is perhaps fair to say that *PipeDream* is easy to use but not easy to learn: it takes quite a bit of work to adjust to at first, and there are a number of conceptual hurdles to overcome before using it becomes effortless. It also imposes a few restrictions for DTP work, though not as many as the manual would have you believe. For instance, it does allow you to use outline fonts on screen as well as in printing: it therefore falls into the category of WYSIWYG software, although only the widths of letters (not the heights) may be correctly shown.

One limitation, not too severe once you are aware of it, is that the line spacing is constant throughout a document, so there is less flexibility when using different type sizes than with programs designed solely for DTP.

New users may be confused by the fact that pressing Return in the middle of your text doesn't usually make a new line (unlike other word processors), it just moves you to the start of the next line. If you want to change this to the more conventional mode of operation you can click on 'Insert on Return' in the Options dialogue box (see Figure 3.3 for the way to reach it by menus, or use Ctrl-O). This will also let you delete backwards from the start of one line to the end of the line before.

Indents and hanging indents. Because of the way in which *PipeDream* uses Tab to jump from one column to another, you can't use tabs *à la* normal word processor for indenting the first line of a paragraph: you will have to insert the indent yourself by adding spaces. However, the spaces you need aren't the normal sort, because if you

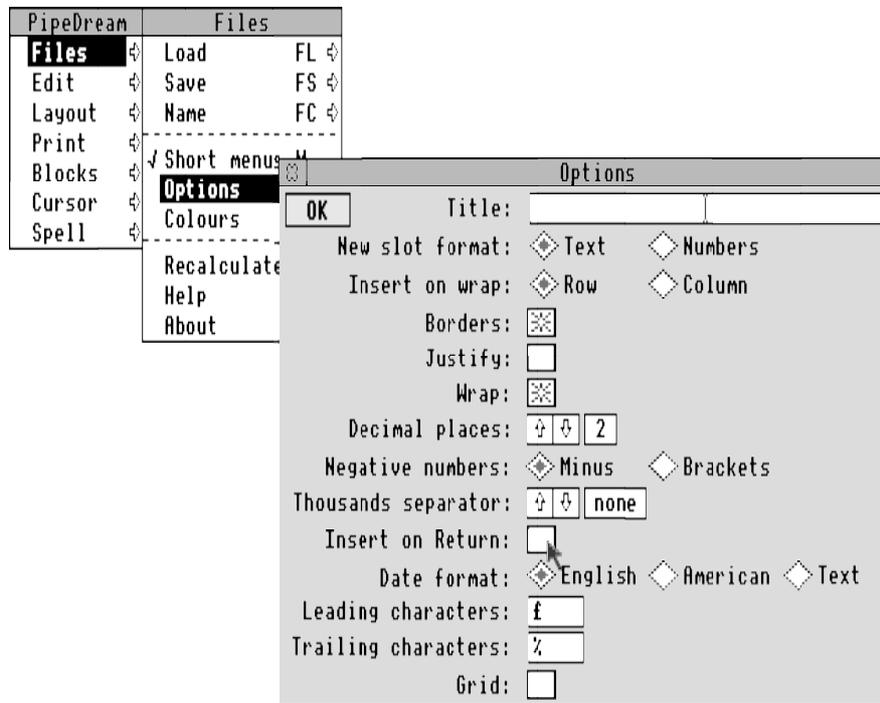


Figure 3.3 The 'Options' dialogue box in PipeDream

justify the text (see the Options dialogue box in Figure 3.3) all such spaces are stretched as the line is expanded to fill the distance between the left and right margins. Instead what you need is something that looks like a space but which will be treated as a letter and will therefore not be stretched: this is called a 'hard space' and is obtained by pressing Alt-Space. As the width of a hard space is normally one quarter of an em (see page 8), for every em of indent that you want at the start of a paragraph you should type four hard spaces.

Indenting a whole paragraph, for example to indicate a quotation, is done very simply by using the 'Set column width' option in the Layout submenu (or by the key shortcut Ctrl-W) to set the start of the next column to the position you want, and pressing Tab to move to the next column. For example, suppose your normal text column is A and you want to indent one or more paragraphs by half an inch, then you should put the cursor in column A, press Ctrl-W and enter the number 5 (for 5 tenths of an inch) so that column B will now start five units in from column A.

If you also want to indent the right hand edge of the quotation, press Tab to place the

cursor in column B (say), call up the 'Set right margin' dialogue box from the Layout submenu (or by pressing Ctrl-H) and enter the value of the margin you require. The downward-pointing arrow on the ruler will move to indicate the new right margin for column B. To get back to column A just press the left cursor key.

Thereafter, every time you want to type indented text all you do is press Tab, type your paragraph(s) and after the last Return use the left cursor key to put the cursor back in the original column.

Hanging indents, where the first line of text sticks out to the left of the rest of the paragraph, are not sensibly possible in *PipeDream*. However, the sort of numbered or bulleted list where the number or bullet stands clear of a rectangular block of text can be achieved in much the same way as an indented paragraph: type the number or bullet character • (hold down Alt and type 143 on the keypad at the right of the keyboard) in the normal text column before pressing Tab.

Line Spacing. To alter the line spacing as defined on page 7, use the 'Printer line spacing' command from the Print submenu to call up a dialogue box into which you should type the intended value in points. If you alter it from 12pt, *PipeDream* will automatically recalculate the number of lines per page so that the page length remains the same.

The Initialisation File. This is where you can store your own preferences so that *PipeDream* always starts up in a way that suits your way of working. All you have to do is set up the layouts, colours and other options as you prefer them in a document, then press Ctrl-FI, which is the command to bring up the 'Save initialisation file' dialogue box, then press Return or click on the OK box. In this way you can, for instance, make 'Insert on Return' the normal state for the program if you wish.

Form Documents. If you are preparing a series of documents to a common pattern, you may want to save time by setting up a form document that has the necessary page and column layout already specified, as well as any other options (such as the printer font). You can then save this blank document under a name like 'Template' or 'MemoForm'. For letters on headed notepaper or work on a regular newsletter, for instance, you can even add constant items such as a logo or title line and save them as part of the form document.

Use of Fonts

Printer Font. For DTP work you will probably want to use outline fonts rather than the ordinary 'System' font that you normally see in *Edit*. First, open a document window by clicking on the PD icon installed on the icon bar. Next, decide in what

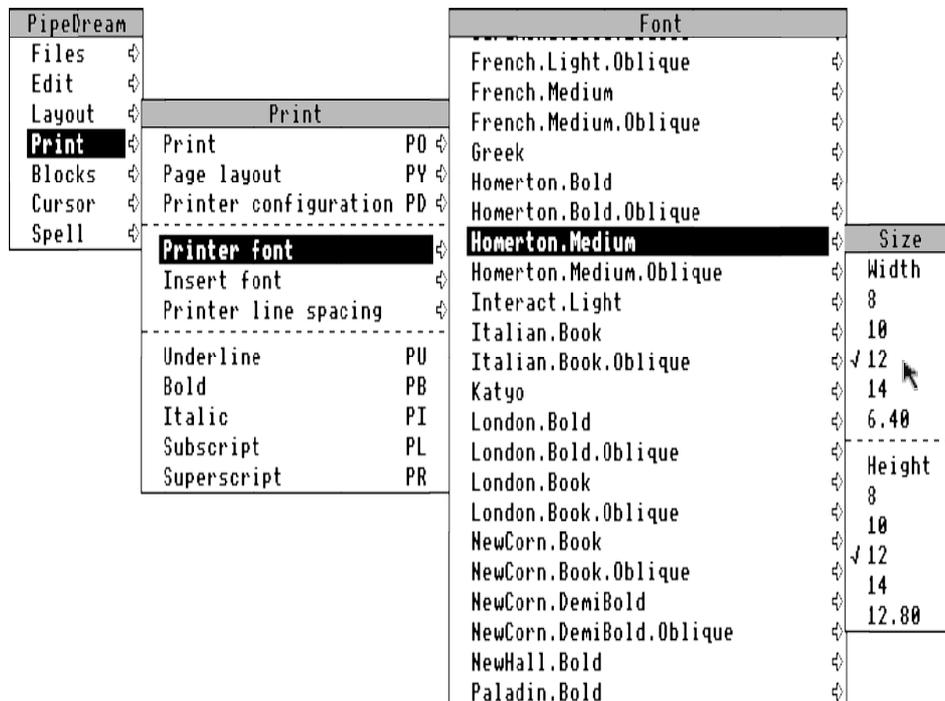


Figure 3.4 Choosing a printer font for the main text

font and at what size the main text of your document should appear, then find 'Printer font' in the Print submenu and slide the pointer across the arrow.

As long as you have some outline fonts and RISC OS has 'seen' the !Fonts directory, after a short delay a list of fonts available will appear, with arrows alongside; following one of these will present you with a further submenu offering widths and heights, as shown in Figure 3.4. Simply choose the font and sizes you want.

If your preferred width or height is not in the range of sizes shown, click on the bottom slot of each of the Width and Height lists and type in the value needed. Note that in *PipeDream* you can easily set the width and height of letters independently.

Local Font Changes. Once you have chosen an outline font as the printer font, you can also pick out headings or sections of text by putting them in a font and size different from the main text. This is done with the @F command, for font change.

The structure of the command is @F:font name,height,width@, where *font name* is the name including full stops (for instance, Trinity.Medium.Italic), and *height* and *width* are measured in points. If no width is specified, it is taken to be the same as

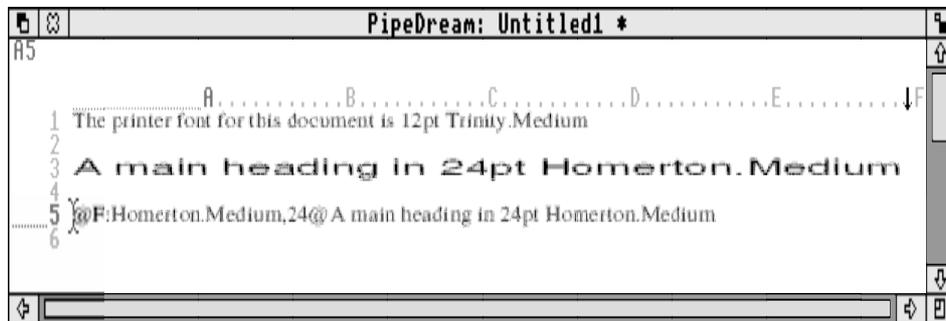


Figure 3.5 Changing font and size: line 1, the printer font; line 3, the heading; line 5, the command

the height. If the width is different from the height the type will look squashed or stretched.

Suppose you have already typed a heading in the printer font and you want to make it appear in Homerton.Medium at 24pt (assuming that the font is in your !Fonts directory): just move the cursor to the beginning of that line and type

```
@F:Homerton.Medium,24@
```

While the cursor is on that line you will see no difference, as is normal with *PipeDream* commands, but as soon as you move the cursor to another line you should see your heading in larger type. Figure 3.5 shows an example of this.

Note that setting the height of a font to more than 12pt with a width to match may make the text look stretched on screen, as in Figure 3.5. This is because each line is being forced to fit into the space allocated to it, which is in effect 1 pica (see page 7) or 12 points high, but the width is still being correctly shown, as in Figure 3.5. When such lines are printed, the character heights will be calculated correctly but successive lines will be printed only 12pt apart. You should therefore make your own arrangements for spacing the lines appropriately if you don't want 24pt text printed '24 on 12', i.e. overlapping.

If you want the effect of a local font change to extend over more than one line you have to type the command at the beginning of all of them, as it is turned off automatically at the end of a line.

Now for the good news about changing fonts: you can save yourself a lot of typing by using the 'Insert font' item in the Print submenu. This operates in the same way as the printer font described above, and all it does is insert a font change command in the text at the cursor position.

An even easier way of ringing the typographical changes is to use *PipeDream's* highlight facility, which lets you choose bold or italic variants (if they exist) of the font currently in use. This is also reached by way of the Print submenu: Figure 3.4 gives the options available. As well as bold and italic type you can choose from underlined text, superscript or subscript; alternatively you can use the Ctrl- shortcuts shown alongside to speed things up. You still have to remember that the effect is only valid for the line it is on, and has to be repeated on subsequent lines if needed.

Function Keys. You may want to prepare several documents with similar typographic styles, with titles and headings in known fonts and sizes. *PipeDream* lets you define function keys to generate any text you like, so they can be used to apply changes of font very easily. Clicking on the option 'Define function key' in the Cursor submenu calls up the dialogue box shown in Figure 3.6. This option, incidentally, appears only on the long menus: if you can't find it you may be working with short menus. If the 'Short menus' option in the Files submenu is ticked, click on it to deselect it. Alternatively, and perhaps preferably, you can reach the 'Define function key' dialogue box with the shortcut Ctrl-CDF.

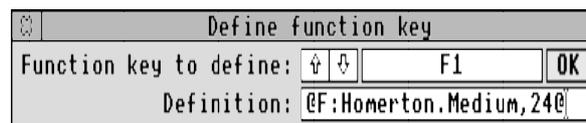


Figure 3.6 Defining a function key to apply a typographic style

The up and down arrows cycle through the function keys and any definitions. When you have reached a suitable one, type in the font change command you want to assign to that key and click on the OK box. In this way you can set up a whole series of convenient commands that can be invoked simply by pressing the function keys.

Your best bet is to use the Shift-Ctrl function keys for font changes, to avoid overwriting any of the built-in function key commands. To help you remember which keys do what, prepare a keystrip showing the definitions, preferably giving them meaningful names such as 'Title', 'Heading', 'Sub-heading', and so on.

You will not, of course, want to type in all your definitions from scratch every time you start a DTP session, so a more generally useful technique is to set up what is called a *macro file* so that *PipeDream* always starts up with the keys defined. It is possible to write your own with commands like

```
\CDF |i "F1" |i "@F:Homerton.Medium,24@" |m
```

which would set F1 to the definition shown in Figure 3.6, but as you can see it is a

complex task. It is far better to let the computer do the work for you, and the powerful 'Record macro file' command was designed with exactly this in mind.

As the name suggests, this is simply a way of making a record of a set of keystrokes and storing them in a file. Then, at any later time you can drag the file onto the open window of a file and the instructions will be executed exactly as if you had issued them from the keyboard then and there.

The best way to start recording is to type Ctrl-FY. The dialogue box shown in Figure 3.7 will appear, into which you should type the name of the file you want to create; call it Key. If you don't specify a full file path (showing all the directories from the root directory), *PipeDream* will put the file in the directory the last document came from, or into the !PipeDream directory if you haven't loaded one. Then all you do is keep calling up the 'Define function key' dialogue box with Ctrl-CDF and entering your definitions.

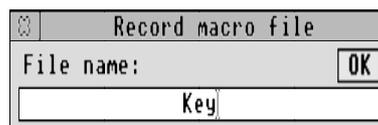


Figure 3.7 Preparing to record a macro containing function key definitions

When you have typed in all your definitions you should type Ctrl-FY again to stop the recording process. At this point you have a file called Key on disk, which you can use at any time. However, if you will always want *PipeDream* to start up with these key definitions you can copy the file to the !PipeDream directory (so it is then called !PipeDream.Key). If *PipeDream* finds a file with that name it executes it automatically at the start of a session.

You may well want to add to your key definitions as time goes by. The best way of doing this is to use 'Record macro file' for all the new definitions, saving them under a different name, and then load them and the originals into *Edit*, saving them back again to disk as !PipeDream.Key when you have checked them over.

Using a key file in this way, together with form documents will make your life easier and keep your printed works looking uniform.

Including Graphics

PipeDream can incorporate both sprites and *Draw* files within a document. There are two methods of inserting such pictures: by the @G command and by RISC OS

dragging. The first of these is similar to changing font, and uses a command such as:

```
@G:picfile,size@
```

in an empty slot. The entry you put in for *picfile* should be either a file path (such as \$.PipeDream.Pictures.Cars.Rover) or a file name (such as Rover); if you give only a file name, *PipeDream* will look for the picture file in the same directory as the file you are editing. The optional *size* value is a percentage of the full size, and can of course be adjusted later; if no size is given it will be taken as 100, i.e. full size. Once you move the cursor off this line the picture will appear, as long as the slot containing the graphic command is still in the document window.

Loading a picture by dragging is probably a more intuitive way of doing it, though. The file reference will be inserted in the document automatically, so you don't have to worry about spelling it correctly. Just put the cursor in an empty slot and drag the icon for the graphic file from its directory window into the document window and move the cursor to another line so that you can see the picture. You can then alter its size by moving the cursor back into the correct slot and changing the value for *size*.

Whichever method of importing graphics you choose, remember that the text in lines immediately after the picture will need moving down to make space for it. If you have 'Insert on Return' selected you can do this with the Return key to add lines, and Delete to remove them, as necessary; otherwise, use Ctrl-N and Ctrl-Y respectively. The example files in the Documents.Examples directory on the *PipeDream* disk should give you some ideas about formatting text around pictures.

As with *1st Word Plus*, *PipeDream* doesn't store the graphics themselves in the document, only a note of where they can be found on disk. This means that you should ideally keep a document and its graphics together in the same directory, and that you should never move a document without also moving the graphics (and vice versa), because otherwise *PipeDream* will get upset.

Printing

When it comes to printing your document, *PipeDream* offers you two alternatives: its own printer drivers and the RISC OS ones. However, as the *PipeDream* drivers are limited to printing text only in your printer's native typefaces you will probably prefer to use the RISC OS drivers, which let you print text in the outline fonts and also any pictures you have included, either or both of which are likely to be needed if you are using *PipeDream* for DTP.

Setting up for printing with a RISC OS driver is straightforward, involving only two steps. The first is obvious (even though I sometimes forget it!): you should load the

printer driver application suited to your printer (!PrinterDM, !PrinterIx, or whatever) by double-clicking on it. Then make sure that *PipeDream* directs its output appropriately by giving the 'Printer configuration' command from the Print submenu; the key shortcut is Ctrl-PD. The dialogue box is shown in Figure 3.8; choosing 'RISC OS' avoids the need to consider any of the other settings.

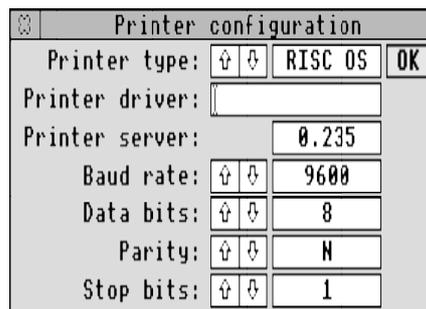


Figure 3.8 Telling *PipeDream* you wish to use a RISC OS printer driver

Printing a single copy of a document is no more complicated than dragging its icon from the directory to the printer driver's icon on the icon bar, after making sure your printer is turned on and has paper in it. Printing more than one copy or with special requirements is started by pressing the Print key or typing Ctrl-PO or choosing the 'Print' option from the Print submenu. Any of these call up the Print dialogue box shown in Figure 3.9. Your likeliest choices will be between portrait and landscape,

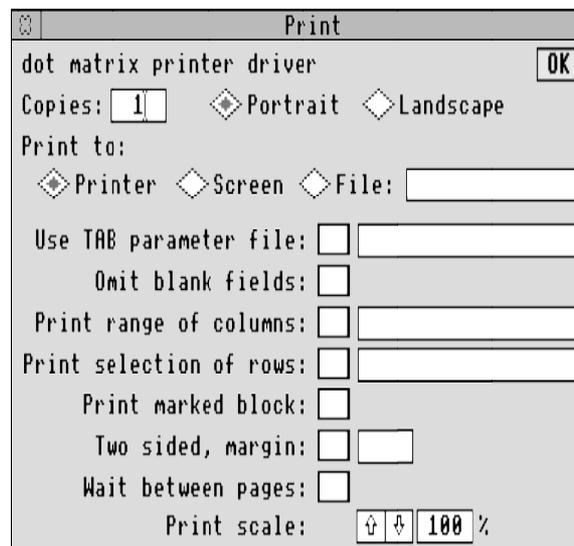


Figure 3.9 Getting ready to print a *PipeDream* document

whether or not to wait between pages, and what scale you want the printed image to be. The OK box sets everything in motion.

EasiWriter

The arrival of Icon Technology's word processor *EasiWriter* put me in a quandary when deciding the final arrangement of this book. It is described as a word processor, and indeed it is one (though cheaper word processing software is available, not least *Ovation* and *Impression Junior*), but in addition it lets you prepare documents with many of the outward appearances of DTP. It could argue for a chapter to itself, but I include it here because it is *text-based* as distinct from *frame-based* (see Chapter 6 for a discussion of why this is significant): what you end up with may look similar, but the concepts behind achieving the result are different to a degree. That makes it hard to discuss it alongside page design software like *Acorn DTP*, *Ovation* and *Impression*. And, to be fair, there are still things that those packages will do that aren't easy with *EasiWriter*, such as preparing a newspaper-like document with multi-column stories piled on each other and rules all over the place. For that you need a graphic rather than a textual approach to the page.

However, *EasiWriter* has some very nice advanced features which merit attention, as well as some basic ones like those in *1st Word Plus* and *PipeDream*.

Basic Concepts

Before starting, you should realise that if your Archimedes has only 1Mb of memory you will need to upgrade it to at least 2Mb, because *EasiWriter* claims just over 700K to begin with, and more as you use other facilities such as editing and spell checking (the multilingual spelling checker is worth a mention).

Unlike either *1st Word Plus* or *Pipedream*, *EasiWriter* can incorporate graphics (both sprites and *Draw* files) as part of your text, even within a paragraph, and such inserted graphics will move forward and backward through the document if text before them is added or deleted. In this respect they behave much as embedded graphics in *Impression*. *EasiWriter* does, however have a further graphic type, the *figure*, which (like the figures in this book) consists of both a graphic part and a caption. Whereas in 'traditional' DTP programs the graphic and caption occupy separate and independent entities on the page, in *Easiwriter* they are part of the same element and are handled together.

So how does this relate to the philosophy behind *EasiWriter*? The key to understanding how the program works is to realise that text is more than just a set of words: it consists of *structures* such as paragraphs, section headings, chapter titles,

or even address lines on a letter. Each of these will have its own special properties, such as type face and size, position on the line, and the space above and below it, as well as things like the number of columns of text that there should be, or whether it should start a new page and/or have a frame around it. The beauty of the system is that you don't have to worry about remembering what size type to use for main headings, or whether they should be centred or start at the left margin: once you have defined all the properties of a structure called Heading, for instance, you can just tell the program that you want a heading and it will apply all the necessary characteristics together. Pictures and figures are just two other sorts of standard structures; other structures you can make use of are lists and tables (the table-building capabilities are particularly impressive).

As with *1st Word Plus* and *PipeDream*, I shall gloss over the standard word processing facilities in *EasiWriter* and concentrate on the layout techniques for pages containing both text and graphics.

Page Layout

Opening a new document by clicking on the *EasiWriter* icon on the icon bar gives you a window looking like Figure 3.10, consisting of a set of icons at the top left, two information boxes describing what is happening at the caret or in a highlighted region (see Figure 7.2 for a highlighted region if you haven't come across one before) and what structure/style is currently active; under these comes the page ruler, and beneath that the marker area for placing margins and tabs; then comes the page area itself.

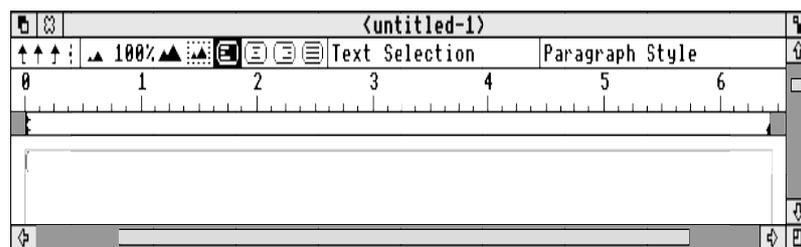


Figure 3.10 The default document window of *EasiWriter*

The icons at the top left fall into three types: tab stops, scale controls and alignment (justification) icons. The *tab stops*, which look rather like three umbrellas and a dot in a vertical line, can be dragged onto the marker area or placed there by double-clicking on the appropriate icon (see Figure 7.4 for the four different tab types) and entering the required position numerically. Similarly, you can drag an existing margin or tab sideways underneath the ruler, or you can double-click on it and

change the value shown in the resulting dialogue box. The *scale controls* change the size at which the page is drawn on screen: the 'distant mountains' icon halves the current scale; the next indicator shows the current scale (100% in the illustration; clicking on it allows you to alter the value directly via the Zoom dialogue box); the 'close-up' icon doubles the current scale; and the 'viewfinder' icon magnifies the view so that a specified area fills the window. The *alignment icons* give you a choice of four text formats: left aligned, centred, right aligned and fully justified (Figure 7.8 has details).

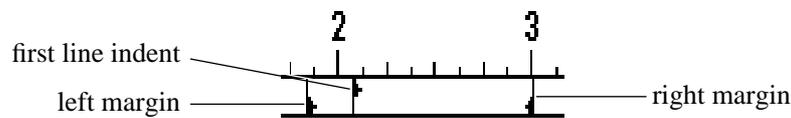


Figure 3.11 The margin markers in EasiWriter

As you type, note that you can make changes to the layout of the text by moving the margin markers shown in Figure 3.11 (just drag them with the mouse pointer). Any change you make affects only the paragraph that the caret is in: they are *local* to that paragraph, though if you then press Return to start typing a new paragraph it too will inherit the same margins. This is also true of type face, size and colour (choose them from the Text submenu) and also the text alignments as set by the icons along the top. If you mark a region by multiple clicks, or by dragging, you can localise the effect still further to individual words or letters, or extend it beyond a paragraph (though text alignment always operates at the paragraph level, whether or not a region is selected).

However, rather than simply controlling the appearance of text by changing options one at a time, you can let the software do much of the hard work in laying out pages by combining the options into structures that are *global*, or valid throughout the document. To get the best from them you should think about what you are doing before you start, because although you can easily change the properties that make up a structure it is not as easy to include completely new structures in a document.

It is useful to think of structures in terms of a hierarchy, a 'pecking order' where one item (such as a chapter) contains other items (a chapter heading and the text of the chapter) which themselves contain still others (the text of the chapter may contain sections, each containing subsections, and so on). Within this structured document you can insert other items virtually anywhere: lists, tables, figures and pictures are sort of 'freelance' structures that aren't hierarchical. Of course, if you have a document that is only one 'chapter' long you can use what the program calls chapter headings as section headings, and consider the chapter structure as a section. It

doesn't matter what the software calls a structure as long as you know what you are using it for.

It seems to me to be important to differentiate between marking and selecting (my distinction, not *EasiWriter's*). Marking is done by dragging, or clicking with Select then Adjust as normal, and results in a highlighted region without regard to any structures it contains (it can include half a paragraph plus a following heading plus a table). Use this technique for the editing options (cut, copy and delete). Selecting is done via the Select submenu or with F11 or Shift-F11, and selects particular structures. Remember to look at the indicators at the top of the window to see what structure is selected and the style currently in force.

Any structure, such as a chapter or a section (or a subsection...), can have a heading in a style and position quite different from the text that makes up the rest of the structure (the *body text*). An explanation of how structures work in *EasiWriter* is best given by means of an example, so let us examine setting up a chapter containing a section.

Start with the caret in a new document window. Before doing anything to the page, take a look at how to change the text area on the page by choosing 'Page set up...' from the Misc submenu. The dialogue box displays two pages, representing left and right page pairs, but you can choose whether or not they look different by clicking on the 'Reflect margins' icon. Try dragging the rectangle in the right-hand page around: as it moves, the rectangle in the other page changes as well. The same happens if you drag an edge or corner of the rectangle to change its shape or size. You therefore only have to set up one of the two pages and this will automatically define the other. This also applies to setting up header or footer areas. You can choose landscape (sideways) format if you wish. If you want to use your newly defined page layout, click on OK; otherwise click on Cancel.

Open a new chapter by choosing 'Chapter' from the Structure submenu, then select the whole (currently empty) chapter by choosing 'Chapter' from the Select submenu. Notice that two things in particular happen: (1) the top line of the text area goes black because it has been selected, and (2) the marker area is white for the whole of the width of the text area and only the left and right margin markers show (compare Figure 3.12 with Figure 3.10). You need to know that this *always* happens when you select a whole structure (as opposed to the heading or the body of one) via the Select submenu, because this is how you set the margins for it. I mention this because it



Figure 3.12 A whole chapter structure selected in an empty *EasiWriter* document

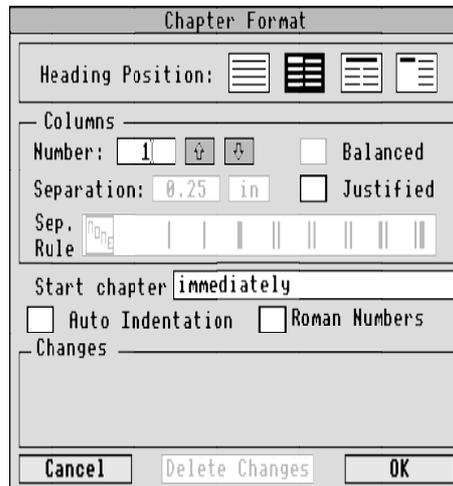


Figure 3.13 The 'Change format' dialogue box

took me a while to realise how to stop subsections from being automatically indented more than the section containing them.

Only while the chapter structure is selected can you alter its format: logically enough, this is done via the 'Change...' option in the Format submenu, which leads to the dialogue box shown in Figure 3.13. Here is where you begin to see the strength of *EasiWriter*: for instance, from the 'Heading position' section you can choose to have no heading at all, a heading confined to one column, a heading across all columns and a heading in a column of its own on the left of the page. A single option allows you this flexibility, and remember that chapters, sections, subsections, sub-subsections and so on can each have their own format. The 'Columns' section lets you choose the number of text columns associated with the structure, and if you have more than one you can choose from various types of rule between them and whether or not they will be balanced. Unbalanced columns are where text in the first column goes to the bottom of the page as it is added, before the next column starts to fill; with balanced columns the software tries to fill each column equally as text is added, keeping the bottoms of all the columns about equal. An enhancement on this is provided by the Justified option, which tries even harder to balance the columns by adjusting the space between paragraphs: note that this is *vertical justification*, as distinct from the justification of text on a line (in *EasiWriter* that is controlled either by the icons at the top of the window or from the 'Alignment' option in the Format submenu). *EasiWriter* is the first software for the Archimedes to provide such an easy and powerful way of keeping multiple-column pages looking tidy.

The next option allows you to start a new chapter immediately after (on the same page as) the previous one, or on the next new page, or on a new left or right page. A new left page for a chapter start is rare indeed, but a new right one is quite common, as this book exemplifies. The 'Auto indentation' option is rather cunning, though possibly misnamed; if activated it suppresses any indent that would normally be applied to the first line of a paragraph, when it immediately follows a heading or comes at the start of a chapter. This is a very desirable thing for documents that use indentation rather than spacing to indicate paragraphs (this book uses the latter convention).

For the purposes of illustration, set up a chapter style with the heading to the left of the text (click on the rightmost icon at the top of the dialogue box), two columns, balanced and justified, a thin rule between columns, starting chapters on the next page and with auto indentation. After you click on OK, something like Figure 3.14 should appear at the top of your page. There are a few things to notice: (1) 'Chapter' and 'Chapter Style' are showing right at the top; (2) there is an asterisk after 'Chapter Style' to show that changes have been made to it; (3) there are now three columns separated by two vertical rules; (4) the first of these is for the heading (which is still selected) and the other two are for the text; and (5) two extra symbols,] and [, have appeared in the marker area below the ruler. These define the right edge of the heading column and the left edge of the text columns, respectively: you can drag them around to alter the proportion of the page given over to heading and text.

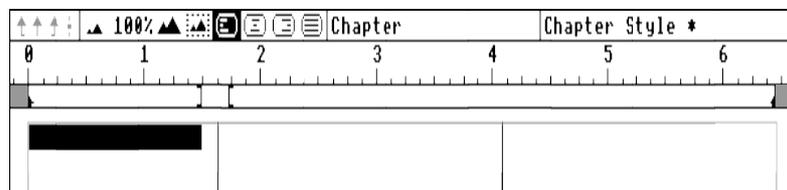


Figure 3.14 A modified chapter style: a heading column and two text columns

Having made changes to the chapter style, it is worth making them permanent. Again, this can only be done while it is selected: go via the 'Save style changes' option in the Format submenu to call up the dialogue box shown in Figure 3.15. If you now click on OK without altering the style name the definition of the style on which it was based will be changed (note that if you do this when you are well into a document you will also change the appearance of all other parts of the document that use this style); to avoid this you can create a whole new style by changing the name. For now you can use the same name.

You haven't yet defined font sizes for the heading and the text, nor the text margins and alignment. First, the heading: click in the highlighted space in the heading area

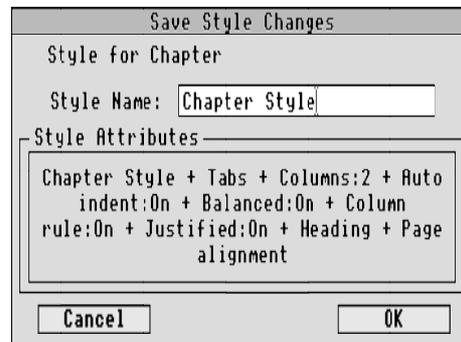


Figure 3.15 Getting ready to save the changes to the chapter style

to bring back the caret (note the change to the marker area: it now shows only the width allowed for the heading), then choose 'Chapter heading' from the Select submenu to select the heading only. Look in the 'Font' list in the Text submenu and check that Trinity.Bold font is ticked; moving to the 'Font size' list, choose 24 point. If you had wanted to, you could have changed both together (and also set the width of the font to a different value from its height) by clicking on 'Type style...' instead. Finally for the chapter head, click on the third icon in the alignment group (☐), to select right alignment. Then save the style as before.

Now to work on the chapter body text, but this time you will need to have entered at least the chapter heading before you can work on the text style. So, type the word Chapter and press Return: the word appears in the chapter heading position and style that you have just defined, and the Return tells *EasiWriter* that the heading style is finished with. The marker area under the ruler has changed, and now you can alter the margins. To make any changes apply to the whole Chapter structure you will need to choose 'Chapter Body' from the Select submenu before moving a marker; do so, and then drag the 'first line indent' marker to the right by about ¼ inch to set up paragraph indents. While the chapter body is still highlighted, apply full justification to it by clicking on the fourth alignment icon (☐). Then save the changes to the style as you did earlier.

Exactly the same procedures apply to defining styles for sections and subsections: these use either the same text width or successively smaller text widths. To continue the example you will need to enter a fair amount of text onto the page: any rubbish will do, as long as it has spaces every seven or fewer characters and a return after every few lines (note that the first line of the chapter isn't indented but all the other paragraphs start with an indent). To save time, just enter a few paragraphs, mark them by clicking Select at the text start and Adjust at the end, then either use the

‘Copy’ option in the Edit submenu or type Ctrl-C, click again at the end of the text to regain the caret and choose ‘Paste’ or (for speed) type Ctrl-V. Then type Ctrl-V as many times as necessary to make the page about two-thirds full. As you type, notice how the program tries all the time to keep the lengths of the two columns equal. Part of this is done by expanding the small inter-paragraph space which strictly shouldn’t be there for indented paragraphs; you can just see this effect in Figure 3.16.

Now for something a little silly (the page layout is already rather unlikely for various



Figure 3.16 A page with a chapter heading and two-thirds full of text

reasons) but illustrative. So far the chapter is all one structure; normally you would expect to divide it into sections for clarity. Each section would usually have its own distinctive heading style, even if the text itself were laid out in exactly the same way as the chapter style. If you look at the chapters in this book, for instance, there is a small preamble before the first section heading in each chapter: this would be part of the Chapter structure, going on to a Section structure for the first heading.

Click to put the cursor anywhere in the middle of the first column of text, then choose ‘Section’ from the Structure submenu, or type Shift-Ctrl-F2 (the new structure will be inserted at the end of the paragraph nearest to where the cursor was). Then select the whole section from the Select submenu because we want to change the style: note the selection and style indicators above the ruler and also the change to the marker area, which should now be similar to Figure 3.17; the white area is limited to the text width of a single column of text in the Chapter style.

Now call up the ‘Change format’ dialogue box and choose the following options: heading left aligned (the first icon), two columns balanced with no rule between, auto indented. After clicking on OK, make sure that the selection indicator still reads

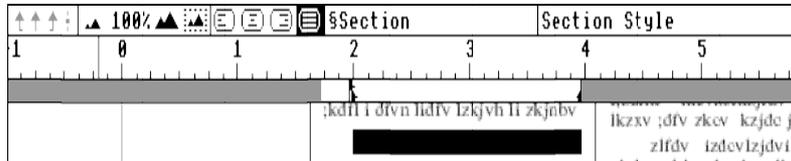


Figure 3.17 A new section inserted and highlighted

§Section; the left markers can be moved back to the left edge of the white area, but for a change you can choose to have the text ranged left. Now save the style changes before proceeding. Place the cursor at the start of the section and type in the section heading ('Section heading' will do), then press Return. The marker area changes yet again, occupying only half the column width now: you can finish the example by typing more text to discover how easy it is to have double-column text within double columns. You can alter the gap between the columns by selecting the §Section structure and changing the format. As I said, a rather silly example, but rather interesting too; Figure 3.18 shows most of the final page.

A rather simpler example would be a letter. In that instance the address lines could be aligned right or centred at the top of the page after you type them, by highlighting and applying the appropriate local style. Then the rest of the letter could be a single section, with the salutation line ('Dear...') as the heading, but in the same type face and size as the main text (leave a space below it by selecting the section heading and altering the 'Space after' value in the Spacing dialogue box from the Format

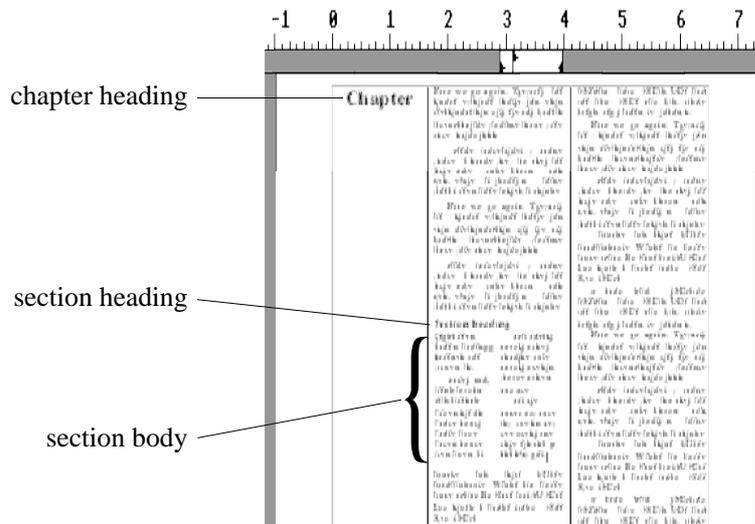


Figure 3.18 A page of double-column text with embedded double-column text

submenu). By setting the heading indent to zero and using Auto indentation you can have indented paragraphs except under the salutation line.

Other Structures. So far I have ignored tables, lists, pictures and figures as they don't form part of the hierarchical arrangement of a document; the last two have their own section later, but tables and lists deserve a brief mention here.

The table structure in *EasiWriter* is the easiest way I have ever encountered to set up a neat table: just press Tab after entering each column and Return to finish each row, and the software does its best to space everything nicely. Once you have entered the whole table you can highlight columns, rows or blocks within it and change the alignment to suit the material: text may be best left aligned or centred, figures often look best right aligned. You can also change the font and size of the headings, or make a totals column or row stand out by putting it into bold type. Don't forget to tell *EasiWriter* when you have finished the table by choosing 'End Table' from the Structure submenu or pressing Shift-Ctrl-F11.

The list structure automatically begins each new paragraph with a bullet (•) character; if you follow it with a Tab all the text lines will start in the same place. After selecting the list body you can add a new tab to the marker area if you want to alter the space between the bullet and the start of the text; in that case you will also need to move the left margin marker and press Tab twice before starting each entry in the list. By using a font other than Trinity you can choose from other bullet characters: select the list by using the Select submenu, and call up the 'Change format' dialogue box. Click in the 'Typeface' box to obtain a menu of available font families, then choose Selwyn (if you have it). Clicking on the yellow 'Try' button shows what the current character looks like in that font. There is no • character in Selwyn, though, as only letters a to o are defined; other symbol fonts give a wider choice. Click on OK when you are happy with the bullet character. As with tables, remember to tell *EasiWriter* when you want to end the list.

Including Graphics

As already mentioned, there are two graphic structures in *EasiWriter*: pictures and figures. Including a *picture* in a paragraph is easy: just drag its file icon to the cursor.

This will insert the picture into the text, rather like this: . Notice that it disrupts the regular line spacing rather a lot; in *Impression* this only happens on one side of the line whereas in *EasiWriter* the extra space is initially shared equally above and below the line in question, which frankly looks somewhat better (there is also an option to fix the line spacing so that inserted pictures don't change it). To change the size of the picture or to crop pieces off it, double-click on it and alter the values in the dialogue box. You can also move it vertically or crop it by dragging it variously

with the mouse.

A picture can also be inserted between paragraphs with the 'Picture' option in the Structure submenu (or with Shift-Ctrl-F8) followed by dragging the file icon onto the resulting outline with an × in it. The picture can then be moved horizontally within the text width and the vertical space allowed for it can be altered by dragging the red outline up or down; the scale of the picture can be changed as before.

The *figure*, on the other hand, is a more complex structure, consisting of a graphic element (it behaves like a picture) and a caption. In much the same way as a chapter or section heading, a caption can be placed in several different positions with respect to the graphic: above, below, to the right or to the left of it. Figure 3.19 shows the dialogue box for changing the format of a figure: the default is to have the caption below the graphic.

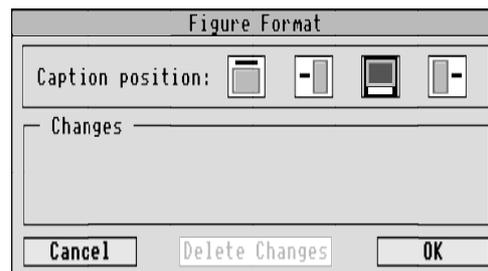


Figure 3.19 The 'Change figure format' dialogue box

Importing a graphic involves selecting the picture area by clicking on it (the outline turns red) and dragging a graphic file's icon to it. The horizontal position of the picture area can be changed by dragging it with the mouse; its vertical position can be changed with the Spacing option in the Format submenu. The horizontal position and alignment of the caption are altered by putting the caret in the caption and choosing 'Figure Caption' from the Select submenu or typing F11, then altering the margin markers under the ruler and clicking on the appropriate alignment icon, respectively. If the caption is alongside the picture, the relative positions of each part are adjusted with the] and [markers after selecting the whole figure. The vertical position of the caption is controlled in the same way as for the picture part.

Because figures are a structure you can alter the style of each one individually. Alternatively, if you want to give them all the same scale and caption position you can change the scale and caption position of one of them, then select the whole figure and save its style under the original name of Figure Style, thus effecting a global change to the figures in the document. In an analogous way you can change

the style definition of the first picture you add, resaving the style under its original name; after that any picture you insert will automatically adopt the same reduction factor and other characteristics. This is a quick way of ensuring consistency of scaling.

Neither pictures nor the graphic part of figures can be stretched or squashed to make them wider or thinner than normal. In version 1 of the software it is also impossible to place illustrations on the page rather than in the text (in other words, they flow with the text rather than allow text to flow round them), making it impossible to have a page layout like that around Figure 6.9 in this book. For the sorts of document for which *EasiWriter* is likely to be used this is no great problem, but in strict page layout terms (for which I use the term DTP) it is a drawback. Later versions may tackle this area.

Printing

Printing an *EasiWriter* document is done via the RISC OS printer drivers, which makes it a completely standard piece of software. The Print dialogue box is reached either from the Misc submenu or, more conveniently, with the Print key. Figure 3.20 shows the options available: essentially you can choose start and finish pages, the number of copies to print, left/right pages (you may want to produce a document printed on both sides of the paper), normal or reverse order printing (convenient for a long document on a printer that produces documents face up), collated multiple copies (the copies are printed as complete sets of pages) and whether or not to pause between pages (useful for sheet-fed printers with manual operation).

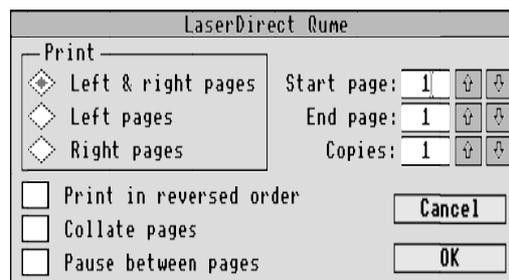


Figure 3.20 The Print dialogue box

A Few Hints

Save your favourite layouts (for instance letterheads) as *Stationery pads*. These are simply documents saved in such a way that they can be loaded as a template for

future documents. They contain all the style definitions, text and graphics of the document at the time you saved it as stationery but the file type is different, which tells *EasiWriter* to load it without remembering its file name. This is useful because when you save it you can't accidentally overwrite the stationery file.

If you have defined several different styles within a structure type the first three are automatically assigned key shortcuts of Ctrl-F9, Ctrl-F10 and Ctrl-F11; Ctrl-F9 always returns you to the basic style for any given structure.

If your favourite style has a space before or after it in the style definition and you want to start a new paragraph without the space being added after the previous one, use the 'new line' facility that is as far as I know unique to *EasiWriter*: typing Shift-Return instead of just Return suppresses the extra vertical space that would normally appear between paragraphs. Although this may seem to be cheating somehow, it can be very useful at times.

Envoi

This lightning tour has necessarily covered only a fraction of the capabilities of *EasiWriter* because far more than 12 pages are necessary to do it justice – I haven't even mentioned emphasis or using different styles within a structure, and have hardly touched upon borders or key shortcuts – but it should have given you something of the flavour of the program.