

2

Equipping Yourself

This chapter looks at how the Archimedes' operating system, RISC OS, makes computing easier, what else you need to obtain in order to begin desktop publishing for yourself, and how to organise things to get the best out of your system. There is also a short section explaining the words used in working with RISC OS.

The RISC OS Advantage

Archimedes owners have the benefit of a computer which is ideal for DTP work. Apart from the sheer speed of the machine's central microprocessor, the Arm chip, and the capacity for memory expansion up to 4Mb (in fact, more recent Acorn machines can go as high as 16Mb), Acorn have written a very powerful operating system called RISC OS to make use of windows, icons, menus and a pointer on screen (a so-called *Wimp* system) that harnesses this power and makes it easy to use.

If you already have an Archimedes you will probably have encountered the RISC OS Wimp system in action. In case you haven't, or are just browsing to find out more about DTP, here is a brief description of what it will do.

Consistency

One of the problems with learning new software has always been the depressingly long time that it often takes to learn all the new commands. This more than anything has led to an amazing amount of wasted time in offices, for instance when a new word processor has been introduced, and also to a marked reluctance, if not outright hostility, to change.

Happily, this situation is a thing of the past with the Archimedes, because the Wimp system that RISC OS uses, and the guidelines that Acorn has written, require

software writers to work in ways that will ensure that certain actions have predictable results. This predictability makes it very much easier to learn how to use new programs. Although it will never mean that you can do away with reading a manual, it reduces the dependence on the manual and shortens that frustrating time during which the learner must work the computer while balancing the manual on one knee.

Multitasking

Every program that uses the Wimp system of RISC OS is called an *application*, and every application runs itself, and interacts with you, through one or more *windows*. The hallmark of RISC OS is that (as long as you have enough memory) you can run several different applications simultaneously, each with its own window, jumping from one to another with ease.

So, to take a personal example, when I was stuck for a word while typing this book I would consult a computerised thesaurus called *KeyWord* (Really Good Software Company), which let me look for synonyms without having to thumb through a book or close down the DTP program first. This *multitasking* ability is a tremendous benefit and comes as a real, though delightful, surprise to someone used to single-tasking computers such as the original BBC Micro series. It can be hard to get accustomed to, simply because of the difficulty of shaking off all the old restrictions.

However, there is more to multitasking than running several programs at once. Where it really scores is in the ability to transfer data from one application to another. What is more, it does this in a way that is completely unnoticeable to the user because of the consistency of RISC OS: all properly written programs use the same method of saving their data to disk (for instance), and transferring data is done simply by 'saving' it to another application.

Another personal example may make this clear. Producing Figure 1.1(b), the anti-aliased letter, was a potentially complex job which in fact took no more than a minute. In the DTP program I typed a letter g at 18pt, which was anti-aliased on screen. I then loaded the *Paint* application from Applications disk 1 and cut out the letter with the 'Get screen area' option, saving it back into *Paint* itself by dragging the Save icon to the Paint icon on the icon bar.

Next I opened the sprite window containing the grabbed letter and zoomed to a magnification of 8:1, then used *Paint* to cut out the enlarged letter (with the enlarged pixels), saving once more directly to *Paint*. Finally I created a frame in the DTP document ready to hold the picture and just 'saved' to the frame by dragging the Save icon over it. At no time was there any need to save to disk, and the whole process virtually worked itself. Perhaps it sounds complicated, but it wasn't really.

The overlaid outline was created separately and put over the top of the pixels in its own frame in the DTP program. In principle, however, I could have done the whole job in only one more stage by transferring the final picture of large pixels to *Draw*, adding the outline to it (involving yet another application) and transferring the whole thing direct to a waiting frame in DTP.

By reducing operations to simple and easily understood steps, RISC OS makes using a computer a pleasant and satisfying experience. The complicated things that RISC OS does (which neither you nor I need to know about) shield you as far as possible from the inner complexities of computing.

RISC OS terms

Although all RISC OS conventions are explained in the User Guide that comes with your Archimedes, that may be more than you need to know to understand this book. With that in mind, I list below all that you require for now.

Windows. All applications run in windows, which look more or less like Figure 2.1. The central *work area*, which you can visualise as a window onto a larger sheet of paper, is surrounded by all kinds of active bits, many of which operate in an obvious way. The most important for the present are the *scroll bars* and *sliders*, with which you move around the whole underlying sheet, the *adjust size icon*, with which you alter the size of the window, and the *title bar*, which you use to move the window around the screen.

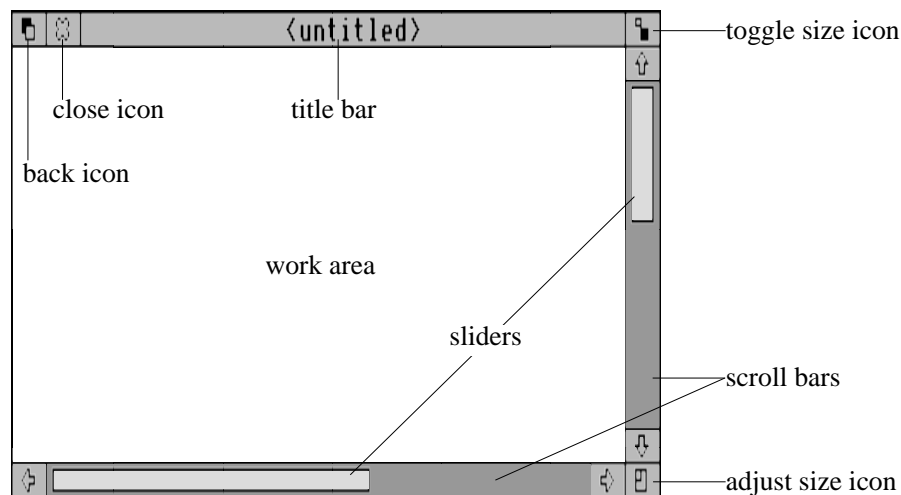


Figure 2.1 A RISC OS window, with labels

Icon. This is a rather fancy word for a picture that represents an object or an action. Icons are the parts of the screen that perform some action when you click the mouse on them.

Mouse buttons. The three buttons on the mouse are, in order from left to right, *Select*, *Menu* and *Adjust*. Whereas the use of keys on the keyboard is described as *pressing*, the mouse buttons are *clicked*. Clicking *Select* will normally be assumed if no button is mentioned; clicking *Adjust* usually has a slightly different effect. Clicking twice in quick succession is referred to as *double-clicking*.

Menus. Clicking *Menu* normally brings up a RISC OS *menu* of the sort shown at the left of Figure 2.2, and sliding the on-screen pointer over one of the arrows to the right of an entry calls up a *submenu*, which may have further sub-submenus leading off them or, as at the right of Figure 2.2, a *dialogue box* giving information or requiring an action.

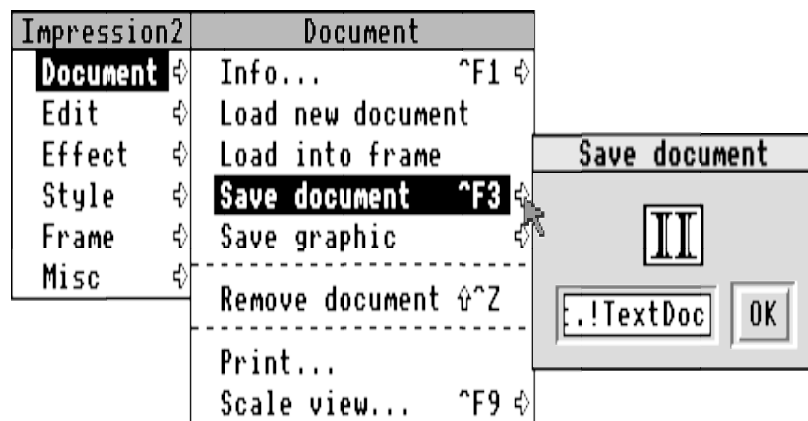


Figure 2.2 A RISC OS menu, submenu and dialogue box

Dragging. The term *dragging* is used to describe pressing a mouse button, holding it down and then moving the mouse as required.

Saving. At the right of Figure 2.2 there is a dialogue box for saving a document, which is used by dragging the square 'II' icon (my interpretation of the *Impression* 'I' icon) to an open directory display. Other files can be saved direct to suitable applications: the icons tell you which files will work with which programs.

Key shortcuts. Once you get to know a program well it can be tedious to have to go through a menu structure to do something simple like saving your work, and if you have both hands on the keyboard it can be distracting to take one off to operate the

mouse. As a result, most applications have built-in *key shortcuts* to save you time. If one exists it is usually shown to the right of the menu item. There are a few in Figure 2.2: the symbol \hat{t} means press Shift at the same time, and the symbol \wedge represents the Ctrl key.

Sprites and Draw files. On the Archimedes there are two types of file used for holding *graphics* information. The simpler is the *sprite*, which is basically a file containing information about the pixels (picture elements) that can be drawn on screen. Sprites are things like saved screens or output from any of the commercially available art software or from *Paint* on Applications disk 1. The other graphic type, the *Draw* file, is mainly produced by the object-oriented drawing application of the same name, though other applications exist that will generate or manipulate them. All Archimedes DTP software will happily work with both types.

What Software?

This is not the place for recommendations of one piece of software over another: all the programs listed are well written and properly supported by their respective software houses, and each would be preferred for some purposes. The available programs fall into three groups according to the needs of the user: word processors with graphics capabilities, educational DTP and 'straight' DTP. Later chapters go into more detail of what can be done and how; here I merely summarise some of the main features of each, and their comparative strengths and drawbacks for DTP work.

An alternative avenue, and one that does not commit you to buying any new software, is to use two programs which are supplied on Applications disk 1: the text processor *Edit* and the much underrated graphic application *Draw*. Between them they provide all the facilities needed for simple DTP, even if this can lead to rather less flexibility and speed, so there is a separate chapter on using them together.

Word Processors with Graphics

There are three contenders: *1st Word Plus* from Acorn, *PipeDream* from Colton Software and *EasiWriter* from Icon Technology.

1st Word Plus, release 2 (Acorn). The first version was the original word processor for the Archimedes; it has now reached version 2. The software is part of the Acorn Learning Curve systems.

Strengths:

- easy to learn and use;
- spelling checker;

- multiple documents on screen; *and*
- automatic hyphenation.

Drawbacks:

- not WYSIWYG;
- doesn't use outline fonts;
- some text formatting problems;
- space in text for graphics has to be inserted manually;
- graphics cannot be resized;
- uses sprites only, not *Draw* files; *and*
- document size limited by memory.

PipeDream 3 (Colton Software). This has been developed from a word processor/database/spreadsheet originally supplied on the Cambridge Computer Z88 portable, via *View Professional* on the BBC micro, and is now in a RISC OS version.

Strengths:

- mostly WYSIWYG operation;
- uses outline fonts;
- is also a spreadsheet and database;
- several text import/export options;
- multiple documents on screen; *and*
- spelling checker.

Drawbacks:

- not always intuitive in operation;
- complex instructions, so can be hard to learn;
- space in text for graphics has to be inserted manually;
- non-variable space between lines;
- really needs more than 1Mb of memory; *and*
- document size limited by memory.

EasiWriter (Icon Technology). Developed by a firm with experience in producing Apple Macintosh software (e.g. *MacAuthor*, *Formulator*), this package has many DTP-like features even though marketed as a word processor.

Strengths:

- complete word processor;
- WYSIWYG operation;
- uses outline fonts;
- multiple documents on screen;
- uses disk as virtual memory: can have documents larger than memory;
- spelling checker and automatic hyphenation;

- comprehensive scaling on screen;
- text structures for applying fonts, layouts, etc.;
- easy table building;
- 'undo' and 'redo' facilities;
- imports both sprites and *Draw* files;
- graphics can be embedded in text or placed between paragraphs;
- 'stationery' can be saved as a template; *and*
- several printing options.

Drawbacks:

- needs more than 1Mb of memory to run; *and*
- graphics cannot be placed so that text runs around them.

Educational DTP Software

Archimedes PenDown (Longman Logotron). This software is based on *Folio* for the BBC Micro and is intended as a word processor with desktop publishing facilities.

Strengths:

- WYSIWYG operation;
- easy for children to learn and use;
- can be configured according to user's experience;
- several text import/export options;
- uses outline fonts;
- spelling checker and word count facility;
- screen view can be scaled up or down; *and*
- several printing options.

Drawbacks:

- only one document at a time;
- space in text for graphics has to be inserted manually; *and*
- uses sprites only, not *Draw* files.

Desktop Folio (ESM). Developed from *BBC Folio* on the BBC Micro, this is a multifaceted package that can be used for simple word processing, for DTP work and for interactive publishing (where the user's clicking on 'action' areas of the page on screen causes things to happen).

Strengths:

- WYSIWYG operation;
- easy for children to learn and use;
- flexible operation in the learning environment;

- uses outline fonts;
- frame-based;
- graphics can be placed within text or in their own frames;
- more than one document on screen at a time;
- screen view can be scaled up and down;
- 'stationery' can be saved as a template; *and*
- excellent documentation, with practical ideas and activity cards.

Drawbacks:

- no spelling checker; *and*
- early versions limited to one style of text per frame, and no text flow between frames.

'Straight' DTP Packages

Acorn DTP (Acorn). This was the first true DTP software to become available for the Archimedes, and is a conversion of *Timeworks*, a package originally developed for DOS computers. It was revolutionary when it first appeared, but as a pathfinder it gave other software developers something to improve upon. It is included as part of the Archimedes A420/1 Learning Curve system.

Strengths:

- pre-tagging of styles for imported text;
- can import some styles from *1st Word Plus*;
- can set maximum/minimum word space for automatic hyphenation;
- uses disk as virtual memory: can have documents larger than memory;
- simple graphics facilities; *and*
- paragraph styles and local styles.

Drawbacks:

- no spelling checker;
- limited scaling options on screen; no print scaling;
- awkward to add more fonts;
- large program, really needs more than 1Mb of memory;
- slow in operation;
- no print margins shown;
- limited text editor;
- no way to condense or expand text width;
- only one document at a time; *and*
- can only work on one page at a time.

Impression, release 2 (Computer Concepts). Version 1 of *Impression* was the second DTP program to appear, followed about nine months later by version 2.

Impression Junior, a simpler and cheaper version, appeared at about the same time as version 2; features that do not appear in *Junior* are shown in parentheses.

Strengths:

- complete word processor;
- loader modules for text import from other word processors;
- spelling checker and automatic hyphenation;
- pre-tagging of styles for imported text;
- abbreviation expansion;
- automatic text flow from page to page;
- several documents can be open simultaneously;
- multiple views of a document;
- (uses disk as virtual memory: can have documents larger than memory);
- (instant rotation of sprites and *Draw* files);
- (global styles);
- local effects;
- grouped frames;
- repeating frames;
- embedded frames;
- comprehensive scaling on screen and in printing;
- 'thumbnail' printing;
- draft printing in printer's own fonts;
- optional flashing cursor;
- (designable frame borders); *and*
- (contents and index generation).

Drawbacks:

- no graphics drawing facilities.

Ovation (Beebug). This was the third DTP package to arrive, and it introduced some new features. It was released in two phases: an initial release without the spelling checker and hyphenation elements, followed shortly after by the complete version.

Strengths:

- word processing;
- spelling checker and automatic hyphenation (only with 2Mb or more of memory);
- paragraph styles;
- simple local style changes;
- tracking control;
- visible clipboard;
- line-drawing facility;
- object repeat facility;

- automatic text flow from page to page;
- hierarchical frames;
- multiple columns in a frame;
- definable frame borders;
- comprehensive scaling on screen and in printing;
- several documents can be open simultaneously; *and*
- automatic document saving.

Drawbacks:

- no pre-tagging of text for style changes; *and*
- document size limited by memory.

What Hardware?

There are many different levels at which DTP can be approached with the Archimedes series, ranging from the home-based hobbyist or enthusiast to the small business using DTP as an integral part of the company's work. Your hardware requirements (like your software ones) will be determined partly by what use you intend to put it to – it may have to double as a games machine! – but principally by the depth of your purse.

All other things being equal, if you haven't yet plumped for a system and know that you will be making more than just the occasional foray into DTP, my advice would be to get the best you can possibly afford at the time, because as your DTP skills develop you are likely to find yourself outgrowing a bottom-of-the-range model. I should make it clear that I have no vested interest in any of the Archimedes software or hardware firms, only some insight into how absorbing DTP can be and how one's expectations of it can grow!

This section covers mainly the computer side of things, as well as some optional extras; it only touches briefly on printers. This is because a separate chapter deals with printers and printing. There is a certain degree of logic behind this: you can in fact do DTP without even having your own printer (and Chapter 11 tells you how). Approximate prices are shown here, rather than in Appendix 2, mainly because, apart from the computers themselves, the choice of other hardware is quite wide and it would be invidious to be too specific.

For the tight budget. It is perfectly possible (though not necessarily convenient or fast) to achieve reasonable DTP results with a basic, unexpanded BBC A3000 (about £700) and a nine-pin dot-matrix printer. If you don't mind the frequent prompts to change floppy disks, the limited memory size, the slow printing and the wear and tear on printer ribbons, then you can produce acceptable pages with minimal outlay:

a cheap dot-matrix printer (about £140) and the software are the only extras you need. Remember to ask yourself what value you place on your own time, though, and include that in the reckoning. Bear in mind that printing just one A4 page may take upwards of ten minutes.

Beyond the tight budget. It may come as a shock to those who, like me, graduated from the restrictions of 32K on a BBC model B, but the problem with 1Mb of memory is that it is never enough: you keep running out of room for documents, especially those with graphics, and you can't take advantage of the full potential of multitasking under RISC OS to any useful extent. So a basic 'serious' DTP configuration has to start with a 2Mb computer (or, better, 4Mb if you can afford it) such as the A420/1 (about £1500). Alternatively, because the price of memory chips is very much less than it once was, upgrading an A3000 or one of the other 1Mb machines such as the Archimedes A410/1 is a cost-effective improvement. This should cost £70–100 for an extra 1Mb or £100–300 for an extra 3Mb, depending on the computer. Many suppliers now sell A3000 computers already upgraded to 2Mb at no extra cost.

If you plan to spend much time at the computer you should consider your eyes and get a high-resolution monitor with a steady picture, or go for a multisync monitor, which will give you access to the higher-quality modes of the computer. Prices range from £400 to over £1000 at the top end of the scale: take a look first.

If you can find one, a second 3½" floppy disk drive (about £130 for an internal drive for an Archimedes 310) is something else you will appreciate: it will eliminate much tedious disk-swapping. Alternatively, if you can afford a hard disk, buy one – but be warned: it will fill at an alarming rate, so go for the largest capacity that your budget will stretch to. Prices of hard disks vary greatly and are still falling as they become more common.

A 24-pin dot-matrix printer is faster and gives a better image than a 9-pin, but ribbon wear and consequent unevenness of printing is still a consideration. A cheaper (but slower) alternative to a laser printer is a bubblejet or inkjet: these cost in the region of £350–600, or £600–700 for colour printers.

The 'professional' set-up. (The use of qualifying quote marks is not because of any doubts on my part about the suitability of the Archimedes for professional DTP, but because of its low profile in this area.) For those with more exacting requirements in terms of speed or quantity (the two are of course related), a larger budget is called for. It is well worth considering a top-of-the-range Acorn computer with a faster processor and more memory: the A540 may be expensive (about £3500) but in a commercial situation there is a trade-off between the cost of time spent doing a job and the financial outlay in getting good equipment.

In other words, a faster machine will cost more initially but may save money in the long run. The surprising thing is that a 'professional' DTP set-up on an Archimedes costs far less than, say, the equivalent on an Apple Macintosh, and is considerably faster. The only limitation at present is in what the software for the Archimedes is capable of – and this situation is improving all the time.

In a work environment it is important to protect employees from occupational hazards. With DTP the biggest of these is eyestrain, so a high-quality monitor is essential; expect to pay up to £1000. The A540 comes with 4Mb of memory (which can be increased, if you really must, to a massive 16Mb) and a 100Mb hard disk; these should suffice for all but the most extensive DTP activity.

For professional results a laser printer is essential: the standard resolution of 300 dots per inch (d.p.i.) has now been made to look a little primitive alongside the 600 d.p.i. of the newer generation, the 'high-res' laser printers. A 300 d.p.i. printer can be bought for £1000–1200, and 600 d.p.i. is available for as little as £1700, which is frankly cheap alongside DTP systems running on other computers.

What Else?

Unless you have already opted for the best of everything you are almost certain to outgrow your equipment. Rather than starting again from scratch, consider using what you have as a basis for something better. You can add a new dimension to your work as well as making your time more productive. Building on an existing system is more than just adding 'go-faster' stripes, though speed is an important element. Print quality and the versatility of your set-up are needs you will become increasingly aware of.

Hardware

My own sequence for upgrading from a system consisting of a 1Mb computer with a single floppy drive would be: (i) a memory upgrade to at least 2Mb, to give the full benefit of RISC OS; (ii) a hard disk to simplify and speed up file handling; (iii) a better printer (24-pin dot matrix, inkjet/bubblejet or laser printer in ascending order of quality and, incidentally, price); (iv) a better monitor to reduce headaches, and (v) an ARM3 processor chip to improve overall performance. Of course, you may have other priorities, particularly if you are building on a different foundation.

Potential add-ons (as opposed to upgrades) are concerned with providing alternative sources of graphic images: scanners and digitisers. Their use is discussed in Chapter 8, which covers graphics.

Software

There is plenty of ancillary software around to add to the power of your system; much of it is for providing special effects with text, and the rest is to do with graphics, so we shall look at this later (Chapter 7 on text; Chapter 8 on graphics).

Fonts

For proper DTP work you need to have the 'new' font manager that Acorn wrote to turn outline fonts into anti-aliased images, as described in Chapter 1. The Applications disks provide a font manager, but it is the 'old' version, designed for the less sophisticated bitmap fonts. All DTP packages for the Archimedes come with a set of outline fonts that will suit straightforward work as well as the appropriate font manager, but neither *1st Word Plus* nor *PipeDream* do. So, although you can use *1st Word Plus* or *PipeDream* in a DTP-like way, you should think about getting some outline fonts.

In any event you are likely to find that you want to expand your repertoire of fonts, and there are plenty of sources to choose from. You can buy font packs from Acorn, Beebug, Computer Concepts, Longman Logotron and some other suppliers, although their ranges are somewhat limited and some (particularly from the smaller firms) are really only suitable for decorative headings rather than straight text. A much larger variety of fonts is available from the Electronic Font Foundry (EFF). All the fonts, except Computer Concepts', come with a copy of the outline font manager with their fonts, though EFF supply the font manager only if asked to do so.

A word of caution about fonts: apart from the matter of look-alikes mentioned on pages 9 and 10, be warned that some fonts are better than others. Most of the above suppliers sell high-quality fonts that are described as 'PostScript compatible'. Why this is important will become apparent in Chapter 11, but don't be tempted to try using cheap fonts for serious work: they are unlikely to have been as carefully constructed and so will probably not give good results at small sizes. They are fine for recreational and casual use, though.

Getting Organised

The rest of this chapter is a bit technical and only of interest if you are actually embarking on the DTP journey, so if you are only reading this for interest you can skip to Chapter 3 – but remember to come back here before you sit down at the computer.

Setting up your computer correctly and organising your files appropriately can make a big difference to the efficiency of both you and your Archimedes. It is all too easy to hamstring yourself or hobble your computer unwittingly if you have only 1Mb or 2Mb at your disposal, or only one floppy disk drive, so a short period of preparation is time well spent.

Memory Management

If you are starting a DTP session after doing something else, make sure you save any files you have been working on and press Ctrl-Reset. This will clear out any modules that have been loaded and restore your computer to a relatively pristine state.

Before starting a DTP session some of the slots on the Task Manager can be set to small values, or even zero. To bring up the Task Manager window, just click Menu on the Archimedes 'A' icon at the bottom right of the screen, then click on 'Task display'. You can drag the 'System sprites' and 'RAM disc' bars to 0K without prejudice, and it will do no harm to drag the 'System heap/stack' and 'Module area' bars as far left as they will go.

To avoid having to play with the Task Manager every time you want to start on DTP, you can set some of these options with the *CONFIGURE command from the command line. Any non-DTP software you may want to use should then either fend for itself by changing the allocations automatically, or complain and make you do it (although under RISC OS this shouldn't really be necessary). To reach the command line, just press F12 from the desktop: you should see a * at the bottom of the screen.

The following sequence of commands can be typed (press Return after each) at the * prompt to ensure a large amount of free memory:

```
CONFIGURE SPRITESIZE 0K
CONFIGURE RMASIZE 32K
CONFIGURE SYSTEMSIZE 16K
CONFIGURE RAMFSSIZE 0K
```

Finish your configuration session by pressing Return to go back to the Desktop screen, then press Ctrl-Break to make the new configuration effective.

Font cache. A font cache is necessary because every time a character is sent to the screen or to the printer the computer has to know which pixels should be light and which dark. The pattern for any letter is different for each size, though the outline stored on disk is the same; calculating it from the outline at any given size is a lengthy process. The font cache is just a part of memory set aside in the Archimedes to store as many as possible of the bit patterns it has already calculated, to save time.

Obviously, the larger the font cache you can spare, the more fonts and sizes can be stored there before the computer (specifically the font manager program) has to decide to throw any away. A bare minimum size is 32K, although 64K is better if you can spare it (not with Acorn DTP running in 1Mb); on a machine with 2Mb of memory you should set it to at least 128K, and with 4Mb of memory or more at your disposal you can afford to go up to the limit of 1Mb if you think it will be needed.

To set up a permanent font cache, at the * prompt type:

```
CONFIGURE FONTSIZE nK
```

where *n* is the size in kilobytes that you have decided on. This gives the font cache a minimum size, but if you want you can allow it to expand to a maximum size as well, in case a session becomes font-intensive. To do this, load the outline font manager (double-click on the !Fonts icon on your font disk) and type:

```
CONFIGURE FONTMAX mK
```

where *m* is larger than *n*.

The ReadMe file supplied with the outline font manager (to find it, double-click on the !Fonts icon while holding down one of the Shift keys) lists a number of other, more specialised, configuration options that you can alter. The most useful of these is FontMax3, which is the maximum size of text (in pixels) that will be cached. Big fonts take up a lot of cache memory, so don't make FontMax3 larger than necessary: it should be set to a value in proportion to both the number of fonts and the range of sizes that you will use, and also the size of cache that you can spare. A good value for normal use on a 1Mb or 2Mb computer is 30 pixels, which is set by typing

```
CONFIGURE FONTMAX3 30
```

Finally, press Return an extra time so that you return to the Desktop screen.

Printer driver. At some stage you will need to use a printer driver (see Chapter 11), but the whole application doesn't have to be resident in memory once you have set it up to match your printer. Remove its icon from the icon bar by clicking Menu on it and then Select on 'Quit'. This will release the memory that the application occupied but will retain the printer module in memory, which is all that is needed for printing.

In the Picture

Screen mode. Choose an appropriate screen mode, according to how much memory you can spare, what sort of monitor you have and how much of a page you want to

see at once. Mode changes are handled by clicking Menu on the palette icon on the icon bar, then choosing the Mode submenu; there are three preset options (modes 12, 15 and 20) and a space in which you can type another number before clicking on it.

The range of modes available is restricted if you have an ordinary monitor, and more so if memory is tight: use mode 12 for colour (it takes 80K) or mode 8 for monochrome (40K).

The *Impression* and *Ovation* packages contain relocatable modules that can be loaded to provide extra screen modes; the mode numbers they use are, unfortunately, different. For a wider screen that shows more of a page on an ordinary monitor, choose *Impression's* 16-colour mode 66 (120K) or *Ovation's* mode 42 (112K).

With a multisync (multiscan) monitor you can use mode 20 (16 colours) if you can spare 160K; otherwise mode 19 allows you 4 colours and occupies 80K. On an A540 with a VGA monitor you can use mode 28 (300K). The 'extra modes' modules mentioned in the previous paragraph give you the chance to try mode 82 (*Impression*) or 54 (*Ovation*), which use 160K of memory.

If you want the computer always to start up in a particular mode, as long as it isn't one of the extra modes, you can press F12 and type the command

```
CONFIGURE WIMPMODE x
```

where *x* is the chosen mode.

Window effects. You can make it much easier to scroll around documents if you configure your Archimedes so that dragging on the scroll bar is immediately reflected in the window's contents; this is called *solid scrolling*. This is simple to arrange: just double-click on the !Configure icon on Applications disk 1, then click on its icon on the icon bar to open the window showing the computer's current settings. The last of the options is 'Instant effect window drags'; if there is no star in the box by it, click in the box to choose the effect, then close the window and quit from the *Configure* program in the usual way.

Some 'naughty' programs have been known to alter the window scroll state in such a way that *Configure* can't re-enable solid scrolling; if this happens to you (this is rare, but infuriating when it happens), press F12 and type

```
CONFIGURE WIMPFLAGS 15
```

then press Return twice to restore the Desktop screen.

File Organisation

Careful arrangement of your files on disk is vital for efficiency's sake: not only will it make it easier for you to find things when you need them, but you can also help RISC OS find what it needs.

Each of the software packages has a section in its manual giving detailed advice on how to set up your disks for easiest working, so here I shall give only a few general notes and suggestions according to the number and type of disk drives you have.

Single floppy disk. Obviously you want to avoid swapping disks as far as possible, so you should set up two sorts of disk: a program disk and one or more working disks. The program disk will have on it everything needed to get the program running, which is probably just the application itself and a !System directory, with perhaps the appropriate printer driver as well if there is room for it. After you have double-clicked on the application to load it, the program disk can be removed and replaced with the working disk, which should contain your documents and a !Fonts directory into which you have copied all the fonts you need, plus the files !Boot, !Run, !Sprites, Fonts and Super from the original !Fonts directory. Cataloguing this disk will install the font manager and cause the fonts to be recognised. You can then open a document, either by double-clicking on it or by dragging it onto the application's icon on the icon bar.

Twin floppy disks. With two disk drives you can split the working disk into two, one containing the documents and the other your fonts. This lets you have longer documents and more fonts, as well as being a more convenient and flexible arrangement. Be sure to open the directory display on the fonts disk before loading any document.

Hard disk. Things are much simpler here: all you have to do is copy !System and !Fonts to the root directory so that they are always available to RISC OS, and create a new directory (call it DTP or something similar) into which you should copy the application and any extensions and utilities supplied with it. In this way, by the time you have opened the DTP directory, RISC OS will already have 'seen' both the !System directory and the fonts. If you want, you can also copy your choice of printer drivers into the DTP directory, or if you will be using a printer with other applications you can copy them to their own directory, called something like P_Drivers.

Font Management

This section applies mainly to hard disk users, because they are more likely to accumulate large numbers of fonts. The rest of this chapter deals with coping with the problems that this causes.

A note about *Edit*. There is a bug in *Edit*, version 1, that causes it to crash if RISC OS has ‘seen’ more than 12 fonts, a number that is easily exceeded once DTP takes hold of you. There are three possible solutions to this. The first is to copy the !Fonts directory into its own directory called Fonts or into your DTP directory; this does, of course, prevent you from using your DTP application and *Edit* simultaneously.

The second is to have two !Fonts directories, one containing only a few fonts and the other holding all of them (put it in a separate directory). Before running *Edit* you can double-click on the small !Fonts directory, and once *Edit* is running you can double-click on the full one. *Edit* doesn’t mind if more fonts are made available after it has been started up, which is why the scheme works. It just means setting aside a small amount of disk space for duplicate fonts.

The third solution is slightly tortuous, but it works: open the !Edit directory display by double-clicking on its icon while holding down Shift, load the !Run file from the !Edit directory into *Edit* itself (remember that throughout this book I use roman type and a prefatory ! to indicate an application directory, and italic type without the ! to represent the application’s name), and add a new line before the last one:

```
UNSET FONT$PREFIX
```

Then save the !Run file under its old name (you may wish to make a copy of the original first under a different name for safety). This will have the effect of making RISC OS ‘forget’ that it has seen the fonts. But beware: after loading such a doctored *Edit* you **must** double-click on the !Fonts directory to tell RISC OS where they are again.

‘Quiet’ font loading. Once you have built up an impressive collection of fonts you will find it tedious to wait for the list of fonts to scroll by if you have to double click on the !Fonts icon. You can stop this from happening by using *Edit* to amend the !Run file in !Fonts: just put a bar character, |, in front of the four lines

```
Echo Outline Font Manager 2.42 installed  
Echo Fonts now available  
Echo  
FontCat
```

and save it again (the bar character is on the same key as the backslash, \, above the Return key). The number may not be 2.42; it depends on the version you have. For this alteration to be useful you must (i) remember what fonts are in the !Fonts directory, and (ii) be able to trust your double-clicking abilities: watching the disk drive light will be the only proof you have that anything has happened.

Adding new fonts to *Acorn DTP*. With most applications that use fonts all you need to do is copy the new fonts into your !Fonts directory and press Ctrl-Break; the applications will automatically take note of them the next time you load them. However, *Acorn DTP* is a bit subtle and insists on a configuration file being altered so that the program can make use of the new fonts. This, briefly, is how it is done.

Working on a backup copy of the program disk, open the disk's directory display and double-click on the application !DTP while holding down a Shift key. This should open the directory to reveal a good few files. Load the text file dtp_config into *Edit* and look for lists of font names and the equivalents by which *Acorn DTP* knows them, for instance:

```

corpus                = 1
corpus                = corpus.medium
corpus italic         = corpus.medium.oblique
corpus bold           = corpus.bold
corpus bold italic    = corpus.bold.oblique

```

The name in the first line of the group is the font family name that will be used by *Acorn DTP*, and the number is the internal font number. This should be the same as the font number used by a DTP program called Ventura, which you may find hard to discover; the actual number is only really important if you intend either to swap files between different Archimedes machines running different copies of *Acorn DTP*, or to send your document to a PostScript printer, in which case you could ask for help from whoever supplied the printer. If neither of these is true, or if the font is unique to the Archimedes and therefore doesn't have an authentic Ventura number, all you must remember is to use a number that has not already been allocated. The subsequent lines in the group show, on the left, the way in which the program will interpret the font names (which are in fact file names and therefore have full stops in them) on the right. Neither capitalisation nor spacing are significant, except between the parts of the names on the left.

Suppose that we wish to add the font Pembroke; we shall for the sake of illustration let *Acorn DTP* call it by its 'true' name Palatino and use the font number 21, which is the real Ventura font number. Pembroke comes in four faces: Pembroke.Medium,

Pembroke.Medium.Italic, Pembroke.Bold and Pembroke.Bold.Italic, so we merely have to give equivalents the program can handle. Starting a fresh group, we type:

```
palatino           = 21
palatino           = pembroke.medium
palatino italic    = pembroke.medium.italic
palatino bold      = pembroke.bold
palatino bold italic = pembroke.bold.italic
```

After saving the altered file, press Ctrl-Break to reset the computer. If, when you start the application again, you receive the error message 'The fonts listed in your configuration file do not match those available on your system', first examine the styles to find which fonts are not being recognised and then check the dtp_config file carefully for typing errors or missed lines.