6

Essentials of DTP

Now this, as they used to say on radio, is where the story really starts. The previous five chapters have outlined DTP in general terms, described the equipment in the Archimedes world that makes it possible, and hopefully have shown that achieving DTP-like results is not confined to any sort of elite with pots of money and expensive equipment. In the next three chapters I shall be setting out what can be done with a ‘proper’ DTP package, with reference to those available for the Archimedes, and the effects that can be achieved with them. That doesn’t mean that there is nothing here for those who are using the software described in Chapters 3 to 5: some of the concepts, practices and hints are just as valid for users of *EasiWriter*, *Edit*-with-*Draw*, or whatever.

What, then, is the distinguishing mark of DTP software? How is it different from a word processor? Word processors have various edit and search facilities; so do many DTP packages. Some word processors offer a wide range of typographic and text layout controls. Many word processors permit, with various degrees of flexibility, the integration of text and graphics. The boundaries seem to be becoming blurred with the introduction of the term *document processor* to signify DTP software with full word processing capabilities – or is it vice versa?

Yet it does seem to me that a distinction can be made that is not merely artificial. The key difference is that word processors are still conceptually linked to the old-fashioned manual typewriter, with margins, tabs and indents all bearing a notional relationship to the bar at the back on which you could slide the metal bits around. The page is then essentially a series of *lines* of typing. DTP, on the other hand, treats the page as a set of *areas*, of which some contain text and some graphics. Each of these areas, when on screen, is bounded by a *frame*, and the frame is, to my mind anyway, one of the hallmarks of DTP. Another is the general way in which DTP work is structured, so first we should look at words commonly used to describe this.

DTP Structures

In this section I shall explain some of the broad terms used in describing what goes on in DTP. Other more detailed words will be defined as we go along.

Document. I have been using this term quite a lot already throughout the early chapters, so unless you are either skimming or haven't been paying attention you will probably have picked up the sense from the context. It is in any case difficult to discuss DTP at all without using this word, so it is fortunate that the meaning is pretty obvious. I have been unable to track down a formal definition, but if one is needed it is something like 'a file or group of files that, when loaded into a DTP package, is treated as a single entity'. But that sounds pedantic: just think of it as a set of pages that you want to work on, and print, together.

Story. This is simply a continuous piece of text within a DTP document. It may occupy a single frame or it may thread through a set of frames and across several (or many) pages. Each of the chapters in this book was written as a separate story, as was each of the figure captions.

Chapter. This is a self-contained part of an *Ovation* or *Impression* document; all chapters are treated separately in that they can have different master pages (letting you have a different style for ordinary pages and, say, the index pages). Each chapter is normally a separate story. Adding one or more pages to a chapter causes the start of the chapter following it to be shuffled along the document. The page numbering at the start of each chapter can follow on directly from the previous chapter, or it can be defined to start at a completely different number.

Master Page. This is a page much like any other but one which lets you define a layout that will be repeated throughout many pages in a document (or all of them, if you want). Master pages are especially useful for giving a document a consistent appearance or layout; you can also put text or pictures on a master page, which will then appear on each page of the document. You can arrange for the page number to be shown on each page, usually in a header or footer area. If the left- and right-hand pages are to be differently designed (for example, in a book), you can define different master pages for left and right pages that will be inserted in pairs automatically.

Headers and Footers. These are special areas of the page that are set aside for items such as page headings, page numbers or document details, which are to be treated consistently throughout a document. In *Acorn DTP* you can have text in up to three parts of a header or a footer: at the left, the centre or the right. In *Ovation* you can position the elements of headers and footers by applying text styles and tab stops. Headers and footers are really a hangover from word processing days, and are helpful to the extent that you don't have to worry about creating frames for them in exactly the right place (the software takes care of that). However, they are also less flexible and run counter to the freedom and control over the whole page area that DTP is intended to provide.

Paragraph. Put simply, as far as DTP software is concerned, a paragraph is a piece of text that ends with a Return character. This general definition covers not only what you would normally recognise as a paragraph (a chunk of text running over several lines) but also anything such as a single-word heading. We shall see later why this is important.

Properties of Frames

We saw in Chapter 4 how 'frames' are used in *Draw* to assemble pages on screen, so now let us examine how DTP extends and develops this.

Appearance

Frame Handles. The software writers of the three DTP packages on the Archimedes have elected to give text frames very similar appearances. Basically, of course, a frame is just a rectangular box that you can type in or drop text or graphics into, but the software usually wants to know which of the frames in a document you want to work in, especially for text. After all, you too want to be sure that when you press a key the letter you typed goes where you expected. There are therefore two states that a frame can take, *selected* (i.e. active) and *deselected* (i.e. all those frames that aren't selected). Figure 6.1 shows a selected frame: it has a box at each corner and one in the middle of each side. The boxes are called *handles* because they are what you 'grab hold of' with the mouse pointer to change the shape or size of the frame. The colours used vary between packages, but no matter. Another way of identifying

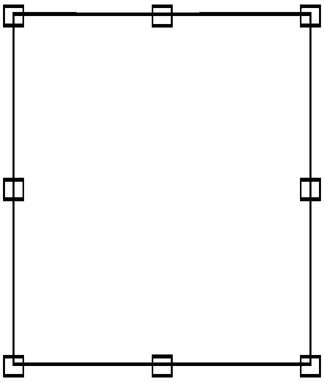


Figure 6.1 *What a selected frame looks like*

where text will go when you type is by the *caret*, a sort of red I shape; if you have used *Edit* or a word processor on the Archimedes you will already be familiar with this. A frame that contains graphics won't have a caret in it, for obvious reasons.

Changing the size or shape of a frame is merely a matter of dragging a handle with the mouse: the handles on the sides are for lengthening or shortening the frame in that direction, and the corner handles let you alter height and width simultaneously.

Deselected frames appear without handles, variously in plain red (*Acorn DTP*) or grey (*Ovation*), or as dotted black outlines (*Impression*).

Moving a selected frame can be done in one of two ways, either directly, by dragging it (*Ovation* uses Adjust for this purpose), or indirectly and more accurately, by altering the X and Y positions in the relevant dialogue box.

Input Focus. Some DTP software will let you work on only one document at a time, but some let you have more than one on screen simultaneously. Just as you need to know which frame on a page is currently selected and will receive input, with *Impression* and *Ovation* you also have to bear in mind which document is active, or, to use the technical term, which window has the *input focus*. Thanks to the unified approach of RISC OS, you can tell this by the colour of the title bar: it is pale yellow when the window has input focus, and grey at other times.

It is especially important to be aware of this when you are working on several documents at once: if you are careless you can do something unexpected to the wrong one. Sometimes, too, a document can lose the input focus during dialogue box work and not get it back afterwards. The only effect of this is that none of the key shortcuts work and no text can be entered, and the cure is to click in the required frame.

Stacking. Because it is possible to have many frames on a page at the same time, it is worth knowing at the outset what happens when two or more frames overlap. The normal state of affairs is that all the frames are in a stack, with those created last on the top. This is true whether or not they overlap at the time of their creation; all DTP software needs to treat frames this way so that it acts consistently when one is moved to a space occupied by another. Looking at the frames on the left of Figure 6.2, from *Ovation*, you can easily see which frame is on top; however, it is not so immediately obvious on the right, from *Impression*, that the selected frame is under the other one in the stack. It takes a moment's reflection to work this out. There will be many more complicated situations than this, and failing to visualise the stacking order can cause puzzling problems such as disappearing text or graphics.

It is therefore a good thing that you can change the stacking order of frames. The

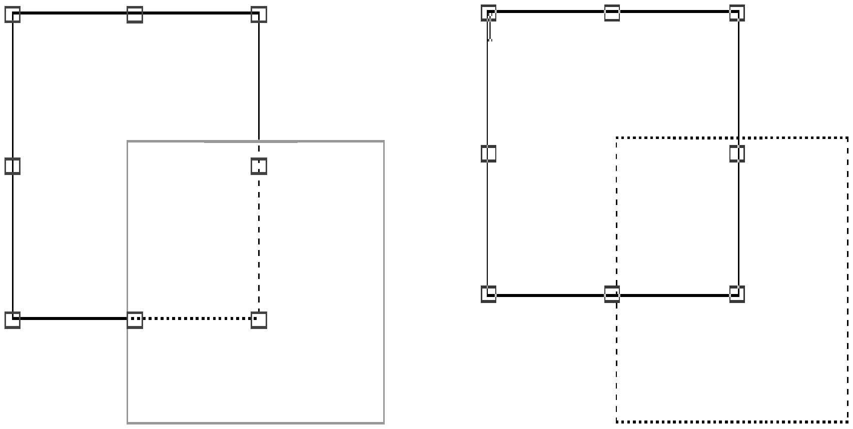


Figure 6.2 *Stacked frames in Ovation (left) and Impression (right); in each the selected frame is underneath the other in the stack*

only way this can normally be done is via the ‘Bring to front’ and ‘Send to back’ options, which put the selected frame at the top or bottom level of the stack. These are rather coarse controls and with a complex stack it may be quite hard to juggle frames to the stacking order you want. *Impression* lets you select frames successively lower in a stack by holding down Alt while clicking Select.

Frame Types

The Text/Graphics Divide. Up to now we have been looking at frames in general, but of course DTP software has to cater for both text and graphics. In *Acorn DTP* and *Impression* when you create a frame it is of a ‘null’, or undefined, variety. It is only when you type in it or drop a sprite or *Draw* file into it that it becomes committed one way or the other: you can’t afterwards alter its type and load text into a graphic frame, or vice versa.

On a different level from the others is *Ovation*: the two types of frame are defined at the outset by the use of the appropriate frame tool. In Figure 6.3 you can see on the left a text frame and on the right a graphic frame, both empty. Each is arrowed to its respective tool. Incidentally, the arrows were created by using the next tool along to the right, but more about that in Chapter 8.

Local Frames. The lowest level of frame is one that you make for a specific purpose, use for that purpose and then forget about. Whatever effect it has on other things on the page, once it is satisfactorily placed and does its job, be it textual or graphical, it is disregarded unless something happens later to bring it to your notice.

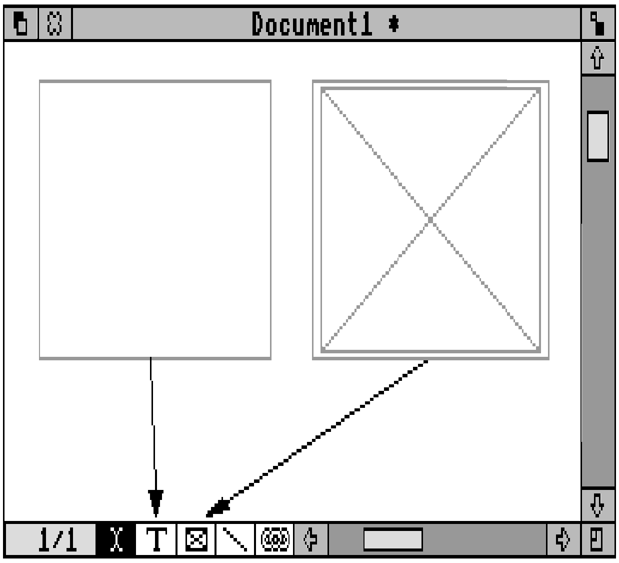


Figure 6.3 *Ovation's two frame types: text (left) and graphic (right)*

At its most basic, that is the essence of a *local frame*. There is no mystery, no fuss; it is just a workhorse. If you overfill it with text you have to make a new frame and link the two together manually so that the text can flow between them.

Master Frames. At the top of the frame pecking order we encounter *master frames*, which you design carefully with a whole document or series of publications in mind, and that you store somewhere safe. These are what you use to achieve consistency, an air of uniformity. All the Archimedes DTP packages except *Impression Junior* let you define a master page for a document (with *Impression* you can define many); all will store this special page (or, in the case of *Impression,* the set of them) as part of the document, and all except *Impression* can also save it separately with other defined style material in a *stylesheet*. We shall see later how master pages and stylesheets are used.

Master frames normally have the useful property of automatic text flow from page to page: once the frame (or frames) on one page is full a new page is created with a new set of master frames on it. *Acorn DTP* is the exception, because stories must be flowed from page to page manually.

Default Frames. It can be useful to be able to treat your DTP software as a word processor, ready for you to type into as soon as you start it up. All the Archimedes DTP packages except *Acorn DTP* let you do this in principle by providing a *default* *frame* (or *principal frame*) that is available as soon as the document window is opened. Default frames are really just another aspect of master frames because they can usually be redesigned on the master page, then saved as a document or stylesheet and either stored in a special place on disk so that it will be loaded automatically with the application, or else kept handy so that you can load the one you want straight away. For example, *Acorn DTP* opens with a blank page with no frame, but a style sheet containing your favourite master frames can be saved in the directory !DTPWork for customised beginnings.

Repeat Frames. *Impression* and *Impression Junior* have an interesting frame type, the *repeat frame*, which has the special property of propagating itself on ordinary pages through to the end of a chapter. This is very handy for inserting material of header or footer nature but which is to vary from chapter to chapter. For example, the left-hand page headings in this book were inserted on the left-hand master page because they are all the same, but I used repeat frames for the right-hand ones because the text is different in each chapter. Deleting a repeat frame on a page also removes all the copies of it from that page right up to the end of the chapter.

Hierarchical Frames. If you have created several frames on a page and then, after arranging them carefully, find that they need to be somewhere else, the last thing you want is to have to move them one by one and set them all up again. For this reason *Ovation* has a special system of hierarchical frames: any frame created inside another is treated as a ‘child’ of the first one. The child frame can be dragged around inside the parent frame but can’t be dragged outside it; conversely, if you move the parent frame the child moves with it. The nested sequence parent–child–grandchild, and so on, can be extended as far as you feel is useful. Although this is convenient most of the time it can be a bit restricting not to be able to drag a child frame outside its parent (although you can, of course, cut it to the clipboard and paste it elsewhere).

Multiple Frame Selection. Similar to, but rather more flexible than, *Ovation*’s built-in parent–child system is the way in which *Impression* enables you to select more than one frame at once and treat them as a unit. Multiple frames are selected by holding down Shift while clicking on them one at a time; all the selected frames are given green outlines to show that they are part of the multiple selection but only one, the first to be selected, has handles, as Figure 6.4(*a*) shows.

Any moving or resizing of the frame with handles is echoed in all the frames; notice that frames are resized in proportion to their original size (the edge of a large frame moves faster than that of a small one), as demonstrated in Figure 6.4(*b*). To make the grouping more permanent the ‘Group frames’ option in the Frame submenu can be used to draw a single frame around all the selected frames; Figure 6.4(*c*) shows the final assemblage of two text frames and a graphic frame, looking for all the world like just one frame.

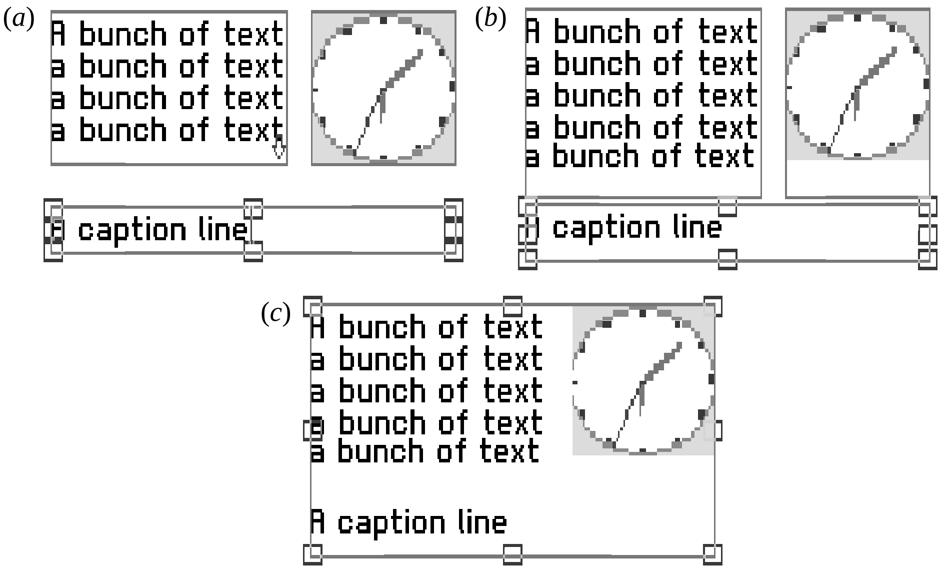


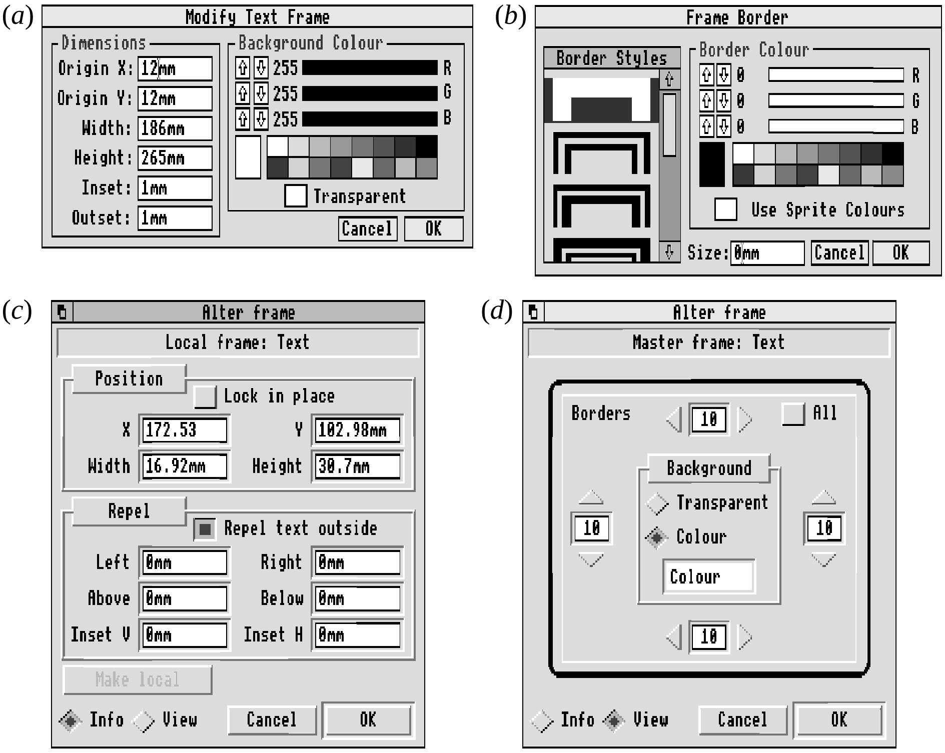
Figure 6.4 *Multiple frame selection in Impression: (a) two text frames and a graphic frame selected together; (b) after resizing the bottom frame; (c) after grouping*

Frame Attributes and Operations

The term *frame attributes* is just a fancy way of describing the properties of a frame: its appearance and how it interacts with other items on the page. *Acorn DTP* uses four dialogue boxes to define a frame's attributes, whereas *Impression* uses three and *Ovation* only two. I shall concentrate on the *Ovation* and *Impression* dialogue boxes shown in Figures 5.5 and 5.6; those for *Acorn DTP* have similar functions despite being different in detail. The two *Ovation* dialogue boxes are reached via different menu options in the Object submenu, whereas the *Impression* ones are called up from just the one menu option in the Frame submenu and you switch from one to the other by clicking on the appropriate radio button, Info or View, at the bottom left of the dialogue box.

Position and Size. These controls are an alternative to dragging entire frames or their handles and are useful in placing a frame accurately according to measurements made on a rough design or on a pre-existing page that you want to imitate. The X and Y values are measured from the top left of the frame.

Background colour/transparency. A common (and often overworked) feature of documents produced by DTP is the panel of text printed on a tinted background. In



(a) Origin X: 12mm
Origin Y: 12mm
Width: 186mm
Height: 265mm
Inset: 1mm
Outset: 1mm
255 R
255 G
255 B
□ Transparent

(b) 0 R
0 G
0 B
□ Use Sprite Colours
Size: 0mm

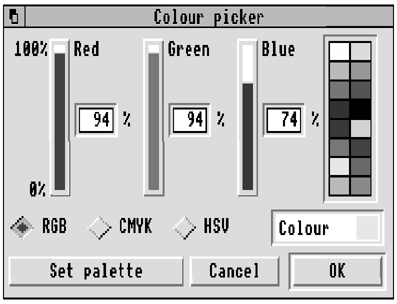
(c) Local frame: Text
□ Lock in place
X 172.53 Y 102.98mm
Width 16.92mm Height 30.7mm
☒ Repel text outside
Left 0mm Right 0mm
Above 0mm Below 0mm
Inset V 0mm Inset H 0mm
● Info ○ View

(d) Master frame: Text
Borders 10 □ All
10 10 10
○ Transparent
● Colour
○ Info ● View

Figure 6.5 *Dialogue boxes for controlling frame attributes: (a) and (b) are from Ovation; (c) and (d) are from Impression (Info and View respectively)*

Ovation the background colour is chosen with the dialogue box shown in Figure 6.5(*a*); in *Impression* you have to get to a separate one called a ‘colour picker’ by clicking on the Colour square in the View dialogue box. *Impression*’s colour picker is used in several different situations, and is shown in Figure 6.6. Using it or the similar one in *Ovation* is easy: you can either click on one of the 16 standard colour boxes or drag the sliders to set the colour square or rectangle to whatever colour you want. In 16-colour modes such as mode 12, however, the final colour will merely be the nearest match available from the Desktop palette. In order to use the background colour facility at all, the ‘Transparent’ button must not be selected. To try to have both a coloured and a transparent background is impossible and is anyway not sensible.

Transparent frame backgrounds have many uses. It is easy to imagine at some time wanting to put textual labels on a picture, or preparing fancy effects by superimposing text over a large picture printed in a pale colour. To avoid the picture



Colour picker

Red 94 %

Green 94 %

Blue 74 %

● RGB ◇ CMYK ◇ HSV

Figure 6.6 *The 'colour picker' dialogue box from Impression*

being obscured by text the background colour of the frame should be set to transparent by clicking on the Transparent button. The (*a*) and (*b*) in Figure 1.7 were added in this way. Note that sprites and *Draw* files will blot out anything underneath them regardless of the transparency of the frame, except where you used the mask colour in the sprite or the fill colour 'None' in the *Draw* file.

Text repulsion. As we saw in Figure 6.2 it is possible to stack frames so that one is in front of another. It is important to prevent text from disappearing behind a picture or any other text, so by making a frame repel any text in a frame beneath it in a stack you can ensure that it is not encroached upon. In *Ovation* this is closely linked with the Transparent button: if the frame is set to have a transparent background it cannot repel text. On the other hand, in *Impression* text repulsion and transparency are entirely independent options.

In both *Impression* and *Ovation*, text can be repelled in either direction from the frame's edge, although the one most often considered is outside because that is the most noticeable. *Impression* lets you set the repulsion distance outside independently for each edge, whereas in *Ovation* one value for 'Outset' has to suffice for all round. Internal repulsion (or *inset*) is useful for a frame with a border, to stop the text getting tangled up with it. The effect of a frame repelling text is shown in Figure 6.7: in (*a*) the first frame has been drawn and then filled with text, and in (*b*) an overlying frame has been created. In (*c*) the repelling frame has been sent behind the first, proving that the effect depends on the frames' relative positions in the stack.

Using a combination of transparent, tinted and repelling frames you can quickly build up complex arrangements – but beware: it is very easy to create a new frame over the top of (or too close to) another one containing perhaps a single letter, forgetting the effect of repulsion. All of a sudden you will find something missing, or (worse) you will overlook the fact that something has disappeared. This is something you really will have to get to grips with because, believe me, frames

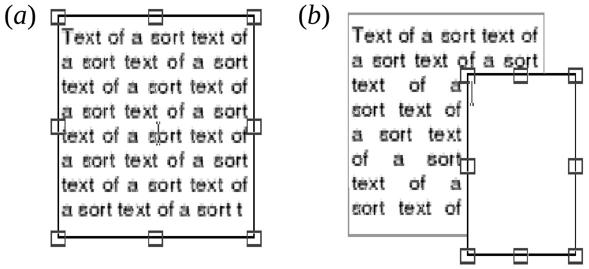


Figure 6.7 *Repelling text with an overlying frame: (a) the first frame, containing text; (b) text repelled by a second frame; (c) text repulsion prevented by altering the stack*

repelling text caused me more headaches than anything else in my early DTP days, and they still catch me out on occasion.

Figure 6.8 is an example of how to use what we have learnt so far to flow text around an irregularly shaped picture. Expensive programs on long-established machines such as the Apple Macintosh can do this genuinely (that is, the frame holding the picture is not rectangular but shaped close to the graphic), but it involves learning complicated new techniques. In contrast, the advantage of the software on the Archimedes (*Impression* in this particular case) is that you can use straight-forward techniques in different ways to get results that are just as effective.

The illustration shown in Figure 6.8(*a*) is a sprite that has just been scanned, dropped into *Paint* and given a transparent mask all around (the area with the vertical stripes), so that when it is placed in a transparent frame any text underneath will be visible in the mask area. If you are unfamiliar with *Paint*, to use a mask you must ensure that both ‘Palette’ and ‘Mask’ in the Edit submenu are selected, then click on ‘show colours’ in the Paint submenu to bring up the colour window; clicking on the last colour (so that it shows a capital T) chooses the mask colour. Then you just use it to paint over the area around the picture; this need only be done fairly roughly as you will probably want to leave a white space between the picture edge and the text.

In Figure 6.8(*b*) the picture has been dropped into an *Impression* frame over a two-column text layout and the frame made transparent and non-repelling; one small frame, the same height as a text line, has been created on top and because it still has a white background it is covering up the top of the picture as well as repelling text. Everywhere else the text is running underneath the picture and disappearing, except in the mask area. In (*c*) the small overlying frame has been made transparent, copied quite a few times and each copy extended to cover the width of the picture at that level: the text is now therefore being repelled, apparently by the picture itself. The last step, shown in (*d*), is to group all the repelling frames and the graphic frame

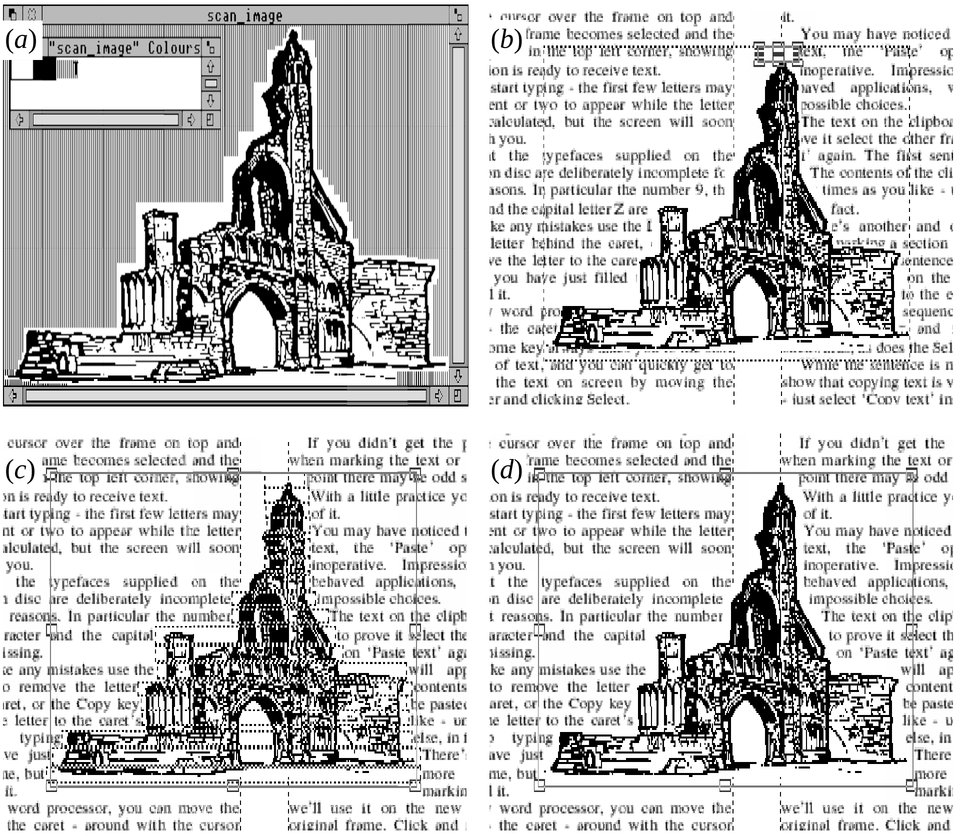
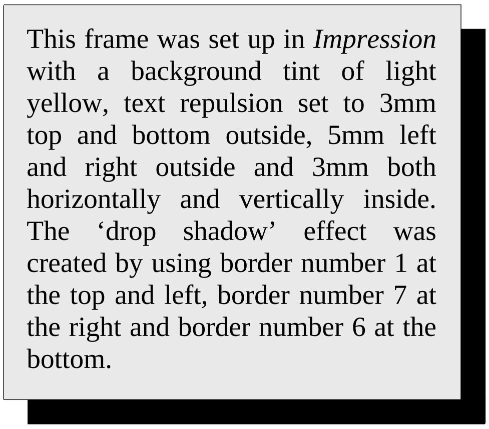


Figure 6.8 *Stages in preparing text running around a picture: (a) the basic picture with mask, in Paint; (b) text running behind; (c) repelling frames added; (d) all picture frames grouped*

together so that they can be treated as one frame, which could now be copied and used elsewhere with a minimum of fuss.

Borders. Normally when you print a page the frame edges are invisible: frames are for holding text or graphics, not for show as entities in themselves. Sometimes, though, you will want to make a display of something, and one way of doing this is to give the frame a border. Borders can be plain or fancy (though only *Ovation* and *Impression* let you add borders of your own design, and only *Impression* uses *Draw* files as borders, to avoid the jagged appearance of sprites when enlarged) but should be used sparingly – if you try to emphasise everything you end up with nothing being emphasised.

Impression can set a different border for each side of the frame (see Figure 6.9), whereas the others apply the same border to all four sides. However, *Ovation* lets you alter the thickness of the border, so turning it on and off involves only altering a value rather than selecting a button.



This frame was set up in *Impression* with a background tint of light yellow, text repulsion set to 3mm top and bottom outside, 5mm left and right outside and 3mm both horizontally and vertically inside. The ‘drop shadow’ effect was created by using border number 1 at the top and left, border number 7 at the right and border number 6 at the bottom.

Figure 6.9 *Making use of frame background colour, frame borders and text repulsion – and very ‘DTP’ it looks, too*

Frame Operations. Apart from being moved and resized by means of the handles or the dialogue box, a selected frame can be removed, or *cut*, from the page to a part of the computer’s memory known as the *clipboard*, which is used in much the same way you would expect if you were working with scissors and paste. In other words, after a frame has been cut out and placed on the clipboard it can be *pasted* back onto the same page or another one (or even into a different document if the software will let you work on more than one at a time). Normally only one item can be stored on the clipboard: a newly cut item will replace anything already there. However, you can paste the same item from the clipboard as many times as you like. *Ovation* lets you look at what is on the clipboard, which is a nice touch.

If you want to cut a frame without removing it from the page, you can *copy* it: this creates a duplicate of the selected frame and puts it onto the clipboard, ready for pasting back. It is normally also possible to *delete* a frame. The difference between this and cutting is that deletion doesn’t put it on the clipboard; it is simply discarded.

Master Pages: the Consistency Enforcer

Earlier in the chapter I explained briefly what master pages are and what they are for, but a little more detail is called for because they are so important and powerful. Conceptually, a master page is like a template or prototype document in word processing: every new page that the software generates automatically is based on a master page. When you are producing a multi-page document it really is very liberating, as you complete one page, not to have to worry about adding a new page by hand and positioning the necessary frames afresh. Master pages therefore provide several facilities that are worth finding out about.

Guides. All three Archimedes DTP packages allow the placing of *guide frames*, *page guidelines* or *column guides* on the master page. Where these are frames they have several interesting properties: (1) you can’t use them for holding text or graphics; (2) they can’t be given a border or a background colour; and (3) like the other guide objects they are invaluable for accurate repetitive drawing of local frames. The main benefit of such guides is that when drawing or moving a frame over the top of it the frame edges will tend to ‘stick’, or *snap*, to the guide, making it easy to line things up correctly. On the other hand the inability to get close to a guide without the frame being sucked onto it can be a nuisance, so there is always an option to turn this ‘snap’ function off.

Impression has a special frame type, the guide frame, one or many of which can be placed on the master page (they can be put on an ordinary page too, but why this should be useful will be considered a little later) so that every new page starts with the same basic scaffolding, on which you place your own layout of frames according to what the design for that page requires. This is a very flexible way of working, but requires some care and skill for best results.

Ovation uses margin guide lines rather than frames: these are four lines (two horizontal, two vertical) placed on the master page to help you position the principal frame or, for that matter, any other frame you want to align with them.

Acorn DTP works on the column guide principle, where you specify how many columns you want on a page and how large the side, top and bottom margins and the gaps between columns should be. From all these the software calculates the width of the columns and places the guides on the master page in the right places. The slight disadvantage of this method (as with the *Ovation* columns) is that you can’t generally create page layouts with columns that are irregular or unequally spaced (unless you are particularly good at mathematics), but for straightforward and regular layouts it can be much easier and is inherently more accurate than the *Impression* way.

Columns within Frames. This is where the early simplicity of the concept of a text frame as a mere container for text starts to come unstuck, though fortunately not too badly. In general you can quite happily get along on the basis of the ‘one frame – one column of text’ theory, but you may sometimes want an easy way to put equally spaced columns of equal widths on the page. One way is to use column guides, as in *Acorn DTP*, and draw your frames over them.

Alternatively, the principal frame on *Ovation*’s master page can be split into columns, with defined spaces between them. In contrast, *Impression* lets you set up guide frames on the master page, over which you can draw master frames and then link them as needed, or (for something like a magazine with lots of small stories to a page) you can just leave the master page containing only guide frames and draw and link together frames as you need them on the page. In the end it depends on what you want to do and is really just a trade-off between convenience and adaptability.

Double Pages. If you look at the open pages of a book you will observe some differences between left-hand pages and right-hand pages. The margins closer to the spine may be different from those at the cut edges, the page numbers will probably be placed at opposite sides, and the page headings may alternate. Very often the design of facing pages seems to be as a mirror pair. With master pages you can devise a pattern so that your document will have pages of alternating kinds, left and right, to let you prepare book-like publications with printing on both sides of the paper. Naturally enough, I did that for this book.

Dynamic Updating. Here is a problem for those who often have second thoughts: after you have been working on a document for some time you may realise that the master page has to be changed, and in that case you are likely to want this change to be reflected in the pages you have already laid out. If you are using *Impression* you will find that after modifying the master page and returning to the document the pages designed around it have been changed. You have the further advantage of being able to choose from a number of master pages in any document, and you can change your mind about which master page you want to use at any time. This opens the possibility of completely redesigning a publication at the last minute; all that has to be done by hand is the repositioning and possible resizing of any local frames containing pictures, captions or separate stories.

Acorn DTP and *Ovation*, however, don’t have such dynamic updating of existing pages when master pages are altered; only pages added after the change will be affected. The only way around this in *Ovation* is to insert the required number of pages (these will be in the new format) at the start of the chapter, which will cause the text to flow back from the pages in the old format; any blank pages at the end of the chapter will be deleted, but any local frames containing illustrations and so on will have to be moved manually.

That is at least relatively easy, but *Acorn DTP* limits you to working one page at a time, which makes the process a lot slower and quite inconvenient. The only practical answer is to export the text and re-import it into a document with a new master page (we shall look at working with text in the next chapter), but even then you could still have local frames to transfer from one document to another – and *Acorn DTP* only lets you work on one document at a time. Your best bet is, I fear, to get it right first time.

On the other hand, you may *want* to change the design of your pages part of the way through a document. In that case, *Acorn DTP* and *Ovation* are ideal. Only

Impression is slightly awkward here, but there is no real problem: just create a completely new master page and start a new chapter. Remember to base the new chapter on the new master page, of course.

Impression Junior. So far in this section we have been ignoring *Impression Junior*. The reason for this is simple: it doesn't use master pages. Instead, every time you reach the end of a page a new one is automatically created, containing a copy of the frames on the previous one. So, the guide frames placed on a page in *Impression Junior* serve the same function as they would on a master page in *Impression*. This 'non-master page' master page facility is also available in *Impression*, though it is not really needed unless it be for compatibility with *Junior* and for easing the transition from one to the other after an upgrade.

Modal and Non-modal Operation

Acorn DTP inherited from its progenitor, *Timeworks*, a way of operating in four completely separate *modes*, in which it is possible to perform only some of the operations available; to work on frames you have to be in frame mode, for example. In this section we shall take a brief look at this aspect of the different DTP software, because some work at least partly in modes like *Acorn DTP* and some, like *Impression*, are non-modal, or *context-sensitive*. In a way, you should be familiar with the concept of modes if you have worked with *Draw*: each of the tool icons on the toolbox panel at the left represents a different mode of operation: you can't enter text while in ellipse-drawing mode, for instance.

When you first open a fresh *Acorn DTP* document, something like Figure 6.10 is what you see: on the left is the *control panel* and *browser*, and on the right is a separate window for the document. To enter any of the modes you just click on the appropriate icon. Frame mode is a prerequisite for any of the work on frames that we have discussed in this chapter; what goes on in paragraph and text modes will be described in the next chapter, and Chapter 8 discusses graphics.

Ovation uses a sort of cross between modal and non-modal activities: instead of having a separate control panel it has a 'toolkit' along the bottom left corner of the document window (see Figure 6.3). From left to right they are: chapter/page number (not so much a tool as an indicator), text insertion, text frame, picture frame, line, and link. However, most of the time you can work in text insertion mode, which is why it is the default: the two frame tools are needed only for drawing frames, the link tool is needed only occasionally for defining how text should flow between frames (more on that in the next chapter), and the line tool equates to a scaled-down version of *Acorn DTP*'s graphics mode.

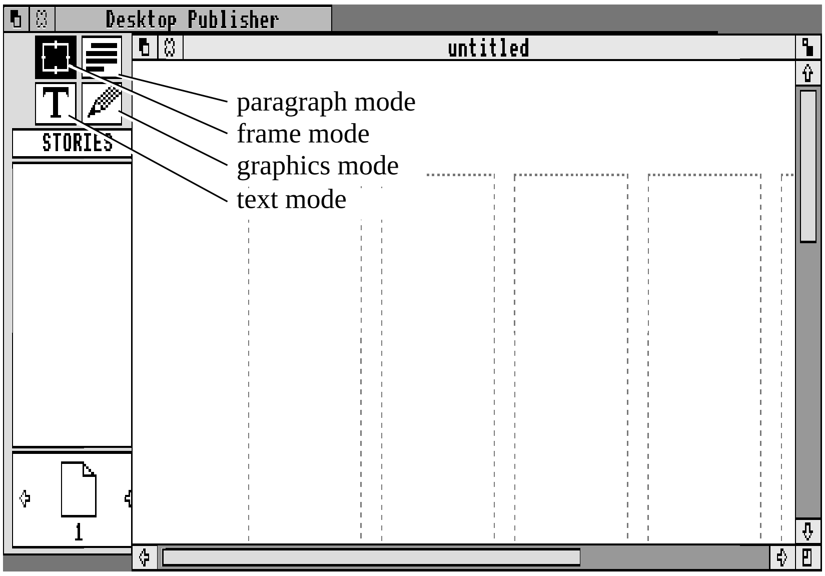


Figure 6.10 *The four modes of Acorn DTP and their icons on the control panel*

Ovation has a slight peculiarity that is worthy of mention at this point: because it has two types of frame (for text and for graphics) and can also be used to draw lines and arrows, *Ovation* groups all these items into a class called *objects*. Depending on which sort of object (frame or line) is currently selected, the ‘Modify frame’ dialogue box shown in Figure 6.5(*a*) may read ‘Modify line’ and look distinctly different. Working on text and working on objects require separate menu items; a longer menu is part of the price you pay in *Ovation* for the freedom of non-modal action.

Impression, unlike the other two packages, is context-sensitive in operation and lacks graphics-drawing capabilities. Thanks to the multi-tasking nature of RISC OS, it is argued, you can always make use of a separate graphics application such as *Draw* for preparing any pictures or lines, and simply transfer them into a frame in the DTP package wherever and as many times as you want. This is one area where I remain less than a complete convert to the ‘total integration’ concept of RISC OS, because from a user’s point of view, if you need only a few arrows or pointers on a diagram, it seems to me that it is far more convenient to draw them *in situ* rather than having to load a whole new application, set it up, load in the picture, fiddle with scales and grids, add the lines or arrows and transfer the finished item back to the document. And, of course, you *may* not have enough computer memory for all this.

As it happens, Figure 6.10 is both a good and a bad example of what I mean: I intended to draw only black lines pointing to the icons, which would have been easier without using *Draw* (if *Impression* could have done it), but I decided to give the lines white backgrounds to make them stand out, which was easy to do in *Draw* (thank goodness for 4Mb) but would have been harder with the less powerful facilities in a DTP program. It all comes back to what you need at any given time – and for all their merits, programmers aren’t psychic.

Colour

I had originally intended to steer clear of a discussion of the use of colour in DTP work because (1) it is expensive and (2) at the time of writing the necessary hardware and software were only just appearing. There is also enough to the subject to allow a whole book to be written about it, and space is limited here. On reflection, this seemed an unhelpful attitude, so on occasion I shall mention the topic, though with some diffidence in view of likely rapid developments in the future.

Preferences

In connection with the non-psychic powers of programmers, it is worth noting that DTP software (*Impression* in particular, but see the next chapter for what you can do with *Acorn DTP* and *Ovation*) lets you customise its operation to your own tastes. In general the choices reflect things like whether frame outlines and print borders are shown, or whether the cursor is flashing or steady. In *Impression* you can also choose, among other things, whether the spelling checker is loaded or the hyphenation is active on start-up, or define independently what the default units are for the page ruler, the style entries and the style ruler. For good measure, you can save your preferences to disk. In this way, just as with the default document or stylesheet (see pages 126 and 132), whenever you start up the program you can feel more in tune with it, rather than fuming at the inflexibility of computers.