Günter Kreidl

Introduction

to

DA'S VECTOR

- Graphics -

- Presentions -

- Animations -

- Tools -

DIGITAL ARTS

Kurztitel: "Einführung in DA'S VEKTOR"

Copyright 1992 by DIGITAL ARTS AG, CH–5503 Schafisheim

Gesamtherstellung und weltweiter Vertrieb:

H3 SYSTEMSmbH,–6900 Heidelberg

AUTHOR: Gunter Kreidl

ENGLISH TRANSLATION: CGS–ComputerBild

TYPESETTING: CGS–ComputerBild

THE PROGRAM "DA'S VECTOR" IS COPYRIGHT PROPERTY OF DIGITAL ARTS AG

NO PART OF THIS MANUAL MAY BE REPRODUCED IN ANY MANNER WITHOUT THE PRIOR WRITTEN PERMISSION OF CGS–COMPUTERBILD

An Introduction

to

DA'S VECTOR

Graphics, Presentations, and Animation Tools

Contents

Preface 5

Of Pixels and Vectors 9

Concept and Use 17

Vectorise Images 35

Drawing and Flexible Shapes 41

Colour, Text, Complex Transformations 47

Generate Graphics from Numbers 59

Graphics in Motion 67

Services and Know–How 79

Preface

Do you want to generate creative or technical illustrations in colour? Do you want to design complex logos with refined typographical features such as shaped text using all available fonts? Or do you require colourful presentations such as slides or overhead projections? Do you need to represent numerical data in pleasing graphics like diagrams, bar charts, pie charts or graphs of your own design? Or do you even want your own "computer cinema" to create comic films or computer animations which can then be played on the computer or copied to video tape? Or perhaps you are looking for a program to produce video titles which, using a genlock interface, can be overlaid onto, or mixed with, film sequences? All this – and much more – you can do with DA'S VECTOR, the first multimedia vector graphics program in the world.

The results of your creative work can be further processed in all kinds of ways: using the relevant export formats, graphics can be imported into practically all DTP applications. You can, of course, also print directly (to dot matrix, laser and colour printers). A special imagesetter version enables you to output directly to high–resolution imagesetters and slide printers. But you can also generate a (colour) picture from the finished graphic and use that for further work.

*Thanks to its highly–developed rendering technology DA'S VECTOR runs in all colour resolutions from monochrome to TrueColour and on all graphics cards featuring a VDI driver. Of course it also makes use of the new colour resolutions of the Falcon 030 computer. The computer's colour resolution, however, is only relevant for the display, as internally DA'S VECTOR always works in 16.7 million colours. Working with DA'S VECTOR requires a minimum screen resolution of 640*400 screen pixels. The animations can be generated for any target resolution (colour and size), independent of the current screen resolution. They can then, of* *course, only be played on computers featuring the required graphics resolution. If the film format is smaller than the screen resolution, the animation will be displayed in a window. The animation player can be copied freely and used by other programs. Thus DA'S VECTOR becomes the first building block for a multimedia system for Atari computers. Further products in this family will follow: they can be identified by their common family name "DA'S".*

A few more words on this manual. You may ask why such a powerful program comes with such a small manual. The answer is very simple: we intend to take seriously the "paperless office" promised us for years by Information Technology. The reality so far looks quite different, since never before has there been so much printed paper produced. In our daily customer support we have noticed that manuals are the more likely to remain unread if they are bulky and detailed. The computer offers far more elegant means of making instructions available "online", that is, within the computer and while you work. The core of this system is the program "DA'S KNOW–HOW" which will be bundled with all products of the DA family (and continuously developed further). DA'S KNOW–HOW can be installed as an accessory or as a parallel application (under MultiTOS, for example) and accessed either on its own or alongside the main application. DA'S KNOW–HOW communicates with the programs and offers its help. All you need to do is to point with the mouse at an icon and press the HELP key – the relevant chapter from the reference manual will appear on the monitor screen and explain the function to you. You will find further key words and a simple mouseclick is enough to branch out to the relevant chapters. Intensive learning is also supported by a "context display": if you click on a key word using the right mouse button, you are shown a list of all passages (chapters) making further references to that key word. A further mouseclick will move you to the chapter of your choice. Of course, DA'S KNOW–HOW offers not only text but also illustrations – and even these can contain further cross–indexing references, "key pictures" as well as key words.

In addition to the reference manual DA'S KNOW–HOW also contains a *tutorial for DA'S VECTOR, taking you through some simple exercises step–by–step and thus making for an easy start even for complete novices. The tutorial, of course, features cross references to the main manual so that both work hand–in–hand. Just have a go and see how elegant and useful this manual within the computer is. You will soon find that this is not just a "cheap replacement" for a printed manual (and tutorial), but a new solution to the problem of contextual learning.*

There are several other advantages over a printed manual. To begin with, almost every computer software manual is already dated when it is printed: while it is still at the printers, the programmers will often have made last–minute modifications or thought up a new feature to be included in the program – you have seen the README files containing the corrections to the manual which are found on program disks. Our programs, too, are of course continually being developed. New functions are added, old ones modified in response to user requests, and so on. Any manual is thus outdated in a very short time and needs to be reprinted or at least supplemented. All this is no problem for the electronic version of the manual. Each version of the program is supplied with an updated version of the manual on disk. Adding further documentation is no problem either, since DA'S KNOW–HOW can manage any number of documents simultaneously. If someone was prepared to make the effort, you could even place the user manual for your Atari computer in there. Other software houses will also be able to use our system under licence – put an end to the wastage of paper, and make a reality of the computer as a multimedia machine!

Why then a printed manual at all, you may ask. Well, we didn't want to break with old habits altogether: a book which you can quietly sit in a corner (or on the loo) with has certain indisputable advantages, too. And before tackling a new program, it can be useful to learn a bit about its background, its concepts and its general usage – I know that most people do not behave like this: you start the program and click around its icons and menus, but at some point you should still take the time to find out what it is that you have just acquired. That is precisely what this manual *is there for. It is an introduction to working with DA'S VECTOR, presents the basic fundamentals, gives an overview of the possibilities and shows you the essentials of using the software. Above all, however, it wants to demonstrate to you what you can do with the software, because in the final analysis a computer program is not an end in itself but a tool. We wish you success in using this tool, as well as pleasure which is just as important. If you enjoy this program then why not look out for its brothers and sisters from the DA family – perhaps you will find even more pleasure with them.*

Of Pixels and Vectors

Short Introduction to Computer Graphics

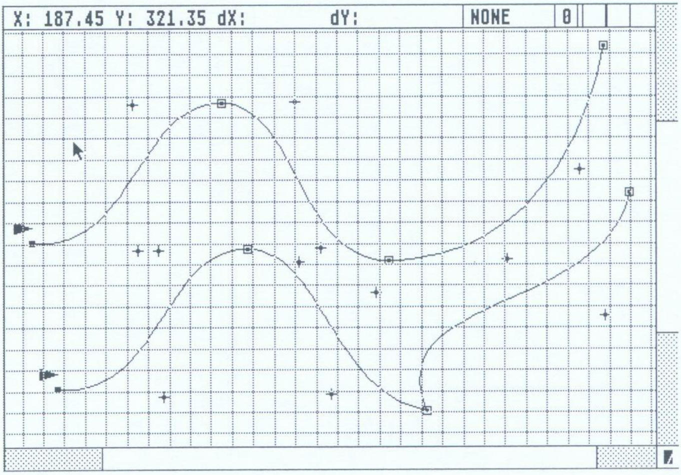
You can skip this chapter if you have already worked with bezier–oriented vector graphics programs and are familiar with the basic terminology.

Many people are fascinated by computer graphics, owners of home computers as well as the DTP professional doing his daily design work on the computer. The processing power, working memory and colour displays of today's desktop computers make it possible to create and edit even highly complex graphics and pictures at a quality which, only a few years ago, was either unthinkable or for expensive mainframes only. To use the computer as a graphic tool, however, it is necessary to learn a bit about the computer as a medium and its possibilities and limitations. In this context, that means primarily the way in which graphics are represented inside a computer.

In colloquial terms we make a distinction between "painting" and "drawing", colour being the main concern of the former and shapes of the latter. Of course the distinction is a fluid one since drawings can contain any number of colours and painting, too, has shapes as an important element. The commonly understood difference between painting and drawing is helpful, however, in understanding how pictures and graphics are handled by the computer. Let us begin with the equivalent of the painted picture in the computer. Due to its binary system, the computer needs to disassemble all data into discrete elements: in the case of pictures, these are pixels, individual picture dots with a clearly defined colour. Just imagine that you overlay a pictures with a screen of small square areas: exactly one colour value is stored for each square element. This principle can be applied to different depths of colour: monochrome (single– colour) picture require 1 bit (the elementary unit of data used by the computer), halftone (grey scale) pictures are normally stored at 8 bit=1 byte per pixel; this is the equivalent of 256 levels of brightness. Hence halftone pictures require eight times as much storage space than monochrome pictures of the same resolution (sharpness). To bring colour into play, one can take advantage of the fact that all colours can (theoretically) be composed from three basic colours (Red, Green, Blue or Cyan, Magenta, and Yellow). Each pixel now requires 3 bytes to obtain the same resolution per colour as with halftone pictures. The number of pixels per centimetre (on screen or when printing) defines the resolution or image sharpness. Computer graphics appear coarse if the resolution is insufficient. For printed halftones or colour photos you should use resolutions of 150 to 300 dpi (dots per inch), about 60 to 120 dots per centimetre. Scanners, which read halftone and colour pictures, work at precisely these resolutions. A colour image of 20*20cm therefore needs about five and a half megabytes for its representation inside the computer. To obtain text and graphics (lines, for example) at high quality, this resolution is no longer sufficient. 600 to 1200 dpi are required in order to achieve a satisfactory definition of the contours. With the computers available today it is almost impossible to place pictures of such high resolution on screen as pixels in 16.7 million colours and to edit them. For this reason, other methods have been thought of to represent text and graphics in the computer, and these involve mathematical descriptions for which the "calculation machine" computer is very well suited. These "vectorised" descriptions lead us into the territory of "vector graphics", the speciality of DA'S VECTOR. But please do not forget what you have learned about pixel images; we shall have to return to that repeatedly later on.

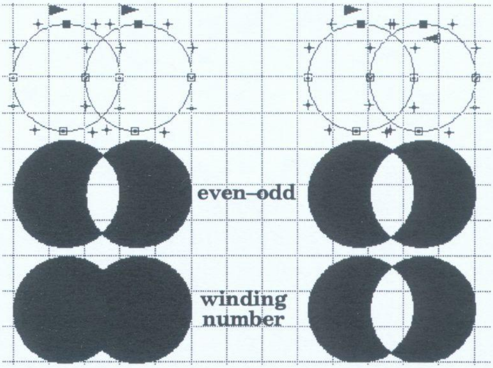
Graphics always contain a constructive element even if a graphic artist is not generally able to describe his creations and designs in mathematical form – or he would presumably have become a mathematician instead. Each graphic shape, however, can be described in mathematical terms more or less accurately, provided you take the trouble. A "shape", in the final analysis, is always an outline or contour. This is how graphic shapes are handled by the computer, as contour outlines. Since "lines" can be linked to other attributes (such as colour, thickness etc), we shall from now on speak of "paths" which are nothing other than (infinitesimally thin) outlines. A mathematician could thus describe such "paths" by means of formulae and let the computer generate graphics from them – which a graphic artist would throw up his hand in horror if he was asked to adopt this method. But this is what the mathematical genius, the computer, is there for – and all the necessary maths are hidden inside the computer, without the need for the user to deal with them. (Now this is not entirely true because DA'S VECTOR also features a "calculator" for complex mathematical transformations, requiring a certain amount of mathematical understanding despite the supplied collection of formulae; but do not worry because even users without much mathematical talent will find plenty of other transformation tools). So how can we reconcile with each other mathematical ("vectorised") descriptions and interactive, creative design and drawing?

This question was asked many years ago by a Mr. Bezier when he was trying to display and model the surfaces of cars on the computer. The method developed by and named after him, of generating graphic objects interactively through so– called "Bezier curves", has found world–wide acceptance. It provides the foundation today of all leading vector graphics programs as well as of the PostScript "printer language". The entire vector font technology, too, is based on this principle. Previous attempts had still used simple geometrical shapes such as polygons, circles and ellipses, but these allow you only to generate very "geometric" objects. Bezier curves (third degree, as widely used) are doubly bent curves which are defined and controlled by exactly four points: the beginning and end points of the curved line and two control points which allow you to regulate the curve in a simple fashion. These control points are always located on the tangents belonging to the end points of the curve. Moving these points (with the mouse) has the effect of an intuitively understandable change in the curve, the mathematically simple structure of the Bezier curve being capable of manipulation in real time on the screen. In DA'S VECTOR, by the way, it is possible to manipulate not only the four Bezier points, but also to grab and reshape the curve at any point using the mouse. Add simple straight lines as the second basic shape, and you can generate virtually any shape (as a path!) from these two basic shapes. Thus Bezier curves and straight lines are the constituent elements of vector graphics, from which all other elements can be freely constructed and reshaped.



So far, however, we have nothing but "paths", the description of outlines, but no graphic as yet. Each path can therefore be linked to attributes, whereby we consider the path on one hand as a line with the attributes line colour, line thickness, line shape (continuous, dotted etc) and line ends (cornered, round etc), and on the other hand as the outline of an area to which a fill colour (and a fill algorithm, see below) is ascribed. Lines and areas can also be transparent, so that it is possible to create graphics consisting purely of lines or areas. All these attributes therefore define how the path is to be drawn. Some explanation is required of the fill algorithm: here a distinction is made between "even–odd" and "winding number" fills: the difference is best demonstrated by an example graphic. With a "winding number" fill, the rotation direction of the path is of great importance, as shown again in the illustration. This is why the program allows you to set the rotation direction of each path.

Every path can have exactly one set of attributes: so if you need, for instance, different line thicknesses, you will need to create several paths. These different "path objects" can then be grouped together. An important factor is the way the individual objects are drawn if overlapping each other: here, too, an agreed paradigm is used world-wide, including the PostScript page description language, and is known as "Painting Model"; this means that all colours completely cover each other just as in painting with opaque (as opposed to translucent) colours. The individual objects are therefore overlaid on top of each other like transparencies. This is why the sequence of their arrangement (as layers) is important and each vector graphics program offers a function to sort the objects (bring forward=to the top, push back=to the bottom, etc).



All these tools enable you to work very nicely in a constructive sense, but for the creative designer, too, vector graphics are today an indispensable tool because of the versatile reshaping offered, especially when such functions as "freehand drawing" are also available. In this, the computer begins by following the drawing mouse pointer and automatically generates a number of short line segments. DA'S VECTOR has a particularly clever way with this, as the sequence of short line segments can be automatically converted into appropriate Bezier curves and thus rounded and made "fluid".

It is also possible, however, to begin by sketching something onto paper and then read the sketch into the computer via a scanner and "trace" it: the computer will determine the outlines and generate them as lines and Bezier curves (DA'S VECTOR can do this even with colour originals). Hence you can transfer everything which you already have on paper to the computer: an important application for this is the vectorisation of already existing logos which are required in the computer for page layouts. You can also trace "manually", by displaying the original image within the path editor and "redrawing" the outlines.

What are the advantages of such a "vector graphic" as compared to a painted or scanned picture, i.e. a pixel graphic (apart from the saving in storage space)? Two important points must enter our considerations here: since the entire graphic is stored as a mathematical structure, it can also be transformed very easily by mathematical and hence geometrical means – beginning with simple enlargement or reduction, via rotation, skewing, proportional and disproportional distortions, to the projection of the graphic object onto a three–dimensionally curved area. All this can be done without any loss in quality, which would be impossible with a pixel graphic: when a pixel graphic is enlarged, nothing is added to the image but the existing pixels are simply enlarged.

And this leads us straight to the second consideration, that of "resolution independence". But first one clarification: as soon as a vector graphic is drawn, it is also a pixel graphic. The reason for this is that the computer does not store the drawing (the pixel image) but the drawing instructions in mathematical form. For each possible resolution the picture is then freshly generated (drawn). Thus you can not only zoom into any level of detail, but generate any desired output resolution. Think, for example, of the production of print masters: to ensure that the graphic is printed smoothly, a resolution of 600 to 2400 dpi is required – about 250 to 1000 dots per centimetre. To store such a graphic in the computer as a pixel image would require around 16 megabytes for an A4 page at 1200dpi (monochrome, and 24 times as much for colour!). This can be done these days without too many problems, but just imagine yourself having to draw a picture in that sort of resolution; the monitor screen would only ever display a very small section of the drawing area (about 4000*6000 pixels) and you would find it virtually impossible to draw a line across the entire working area; painting by hand would also be an impossible task as up to 24 million pixels would need to be edited. A vector graphic, on the other hand, can be calculated to such a resolution without difficulties – a screen display (at about 72 – 92 dpi) with a rather coarse and blocky appearance would emerge perfectly smooth on an imagesetter. Only this makes it possible to render graphics in perfect quality at print time and it is the precondition for the creation of camera–ready artwork by computer.

The entire font technology, too, depends on this approach. "Vector fonts" are typefaces in which each character is in effect a small vector graphic (to be precise, a path object). Thousands of different typefaces are available in PostScript format – and can, of course, all be used with DA'S VECTOR (in addition to the CFN fonts which are found only on Atari computers). Other questions related to this technology, such as the display of halftones and colours through raster screens, cannot be discussed here. A piece of software which transforms vector graphics (including typefaces) into an image is known as a RIP (Raster Image Processor). Each PostScript interpreter contains such a RIP. DA'S VECTOR, too, contains its own RIP (and can therefore output directly to an imagesetter without PostScript).

For all the reasons given here, vector graphics have become an indispensable part of graphic design work. But there are also disadvantages, of course. Interesting as such graphic creations may appear, they also always seem somewhat "technical" and cannot be anything like "photo realistic". The reason for this is to do with the structure of natural objects and our senses of perception: natural objects always have a surface structure which contributes decisively to their colour effect. No natural objects is smoothly painted. Vector graphics, by contrast, know only "pure colours" without any structure and this is why vector objects appear so artificial. A surface structure such as that of natural objects can only be created by pixel graphic programs or with a scanner. DA'S VECTOR removes this obstacle by making possible a new kind of combination between vector and pixel graphics: areas can not only be filled with a colour, but also with a halftone or colour image of any size, and you can even set the size, position and perhaps the "repeat rate" of such a "fill pattern" to your requirements. Thus it is now possible, for example, to generate vector areas with a wooden or metallic structure or to integrate colour pictures with arbitrary outlines (paths) into vector graphics. The applications of this are virtually unlimited and greatly expand the possibilities of vector graphics, as you will soon find in your own experience.

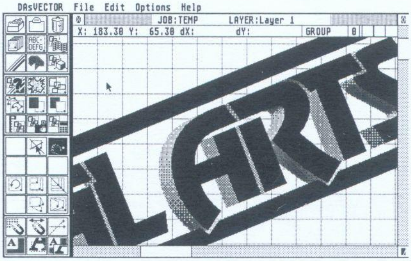
DA'S VECTOR is not a program for editing pixel graphics (but there is also DA'S PICTURE!), but it makes multiple uses of such halftone and colour images: you can use these pictures for automatic or manual tracing or as a "fill pattern/filled area" in a vector graphic, and, finally, you can generate pixel graphics by outputting the entire vector graphic as a halftone or colour picture.

Concept and Use

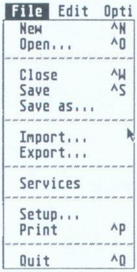
After all that theory let us now go straight to the heart of the matter and turn to DA'S VECTOR itself. This chapter will introduce you to the concept of the program, i.e. the functions of the various parts of the program, how they are used and the standard features of the user interface. Subsequent chapters will then show you the main parts of the program with their most important features. We assume in all this that you have mastered the basic usage of your computer and that you are therefore familiar with the meaning of "dropdown menu", clicking, double clicking and so on.

The Menu Bar

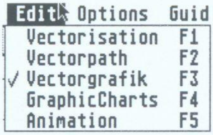
You will find three different elements on your desktop after starting DA'S VECTOR: the dropdown menus with the global functions under the four headings "File", "Edit", "Options" and "Help", the toolbox with many graphic symbols ("icons") on the left, and the window containing the work area.



Under the menu heading "File" you find, as expected, functions to load and save, import and export, print or quit the program. You can load and save "job files" with the "HTJ" extender, which save the entire system state. What you can import or export depends on which program part you happen to be in at any time.

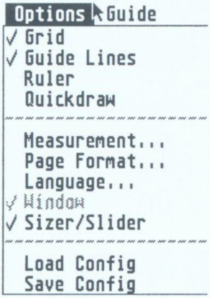


The DA'S VECTOR program consists of five more or less independent program parts which can be selected in the "Edit" dropdown menu: Vectorisation, Vector path, Vec-

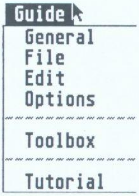


tor graphic, Presentation graphic and Animation. Each program part has its own toolbox on the left. This toolbox contains the actual tools and functions.

Under "Options" you find a number of global settings. Here you can enable or disable the visibility of the guide grid, the guide lines, and the ruler. If you want to work particularly fast, you can switch on the "Quickdraw" mode in which only outlines (paths) are drawn. You can select the units of measurement (Millimetres, 1/8th inch, 1/10th inch, Pica, Cicero) and even the language in which the program will communicate with you (normally English or German). "Page format" gives you a dialog box to make a number of important settings: you can define an "active format" to be used for printing and the generation of animations; and also set the origin and the unit width of the grid. In addition, you can enable or disable certain window elements (slider and size box) – this is important for multitasking environments such as MultiTOS – or save and load the entire configuration.



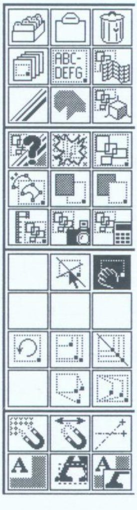
The last menu heading is for calling up the Help system. This Help system employs an external program, DA'S KNOW–HOW, which is normally installed as a desk accessory. Under MultiTOS it can also be used as a parallel application.



DA'S VECTOR sends messages to DA'S KNOW–HOW and asks for help. This help, consisting of a complete online manual as well as a tutorial, can be requested in different ways. One is to branch into the different main chapters of the electronic manual or the tutorial via the menu bar and from there go elsewhere. But you can also request explanations of the different toolbox functions directly by positioning the mouse pointer over one of the icons and pressing the HELP key.

The Toolbox

The structure of the toolbox is the same in all program parts and a number of functions or tools appear in several toolboxes. A basic understanding of the structure of these toolboxes will greatly simplify your work with DA'S VECTOR, and for this reason we shall describe this structure right at the beginning. The example used is the toolbox in the key program part "Vector graphic" – a key part because here you can edit the graphic objects produced with all the other program parts (except the animations).



Each toolbox consists of four parts: the top part contains a maximum of 9 symbols; each of these symbols calls up another submenu when clicked on, i.e. another toolbox with specialised tools. All these additional submenus can be left again by clicking on the RETURN symbol in the top left corner; you are then back in the main menu (toolbox) from which you have called the submenu. The submenus themselves can be very different depending on their function, hence they do not necessarily have the same structure as the five main toolboxes. In the "Vector graphic" toolbox, all nine symbols for submenus have been used up: from here, you can access the clipboard, the library and the trashcan (which is also a kind of clipboard from which the deleted objects can be retrieved); in addition you can get from here to the Layer menu, the Vector Text menu, the menus for setting the line and area attributes, the Bezier area transformer and the 3D transformer.

The next section of each toolbox contains global or local functions which have a certain effect on either all objects or just the currently selected one; in the Vector graphics menu these include, for example, such elementary functions as "Group or Ungroup", "Move to foreground" or

"Move to Background", but also complex functions such as the calculator (for mathematical transformations of graphic objects) in which case a dialog box will appear to allow you to make the relevant settings first.

The third section of each toolbox contains the actual interactive tools. When one of these icons is selected, you will then be able to carry out a special function with the mouse in the working window. In the Vector graphic box, for example, these include the following tools: Protect object, Move object, Rotate object, Free or proportional scaling, skewing or perspective distortion.

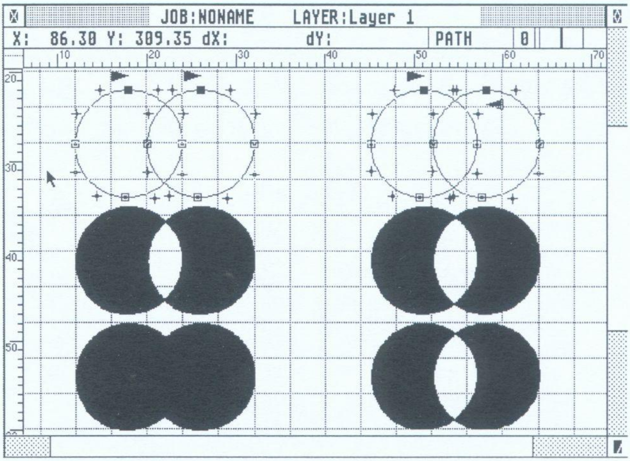
The fourth section is the same in all five toolboxes and contains six important functions that are required in all program parts. The functions can, in turn, be subdivided into two groups (upper and lower row): the upper row makes available the construction tools: magnetic snap to grid, lines etc; restricting the direction of movements (horizontal, vertical or any angle); draw guide lines. All three icons have a pop–up menu which can be opened with a doubleclick or SHIFT+mouse click and contains the "fine tuning" for these functions. These very refined construction aids will be further described below.

The bottom three icons are for selecting the screen area. i.e. that section of the working area which you can see on the monitor. The icon on the left makes the current "page format" or the entire working area visible (with ALT+mouseclick). The one in the middle lets you select a new section for the working window: once you have clicked on this icon, you can draw up a frame in the working area which will then appear in the window at the corresponding magnification (Zoom function). The icon on the right is for switching between the current zoom mode and the previous one and back, so that you always have at least two zoom levels available without first needing to define a new section.

All the toolbox icons can, of course, also be selected from the keyboard.

The relevant keys (key combinations) are listed in the online reference manual.

The Working Area



JOB:NONAME LAYER:Layer 1

X: 86.30 Y: 309.35 dX: dY: PATH 0

To begin with, imagine your working area as a sheet of the size 81.92*81.92 centimetres. This working area can be shown in its entirety, and therefore greatly reduced, in the working window. Within this working area you can define a special active region, called the "page format". It is marked by a frame consisting of a broken line. This is the area which will be printed (hence "Page format") or, with animations, is used as the picture area. You can select any section of the working area using the zoom function, and this area will then be displayed in the window in the required magnification. This section can be moved all over the working area in a number of ways: if you have enabled the standard GEM window mode, it can be moved about in the usual way using the window sliders, but irrespective of this there is a second method of defining the visible section under full visual control: with the right mouse button depressed, either the current page format or, with the ALTERNATE key pressed in addition, the entire working area is displayed in a special "Quickdraw" mode which draws only the paths. You also see a frame which you can move over the working area with the mouse. As soon as you release the mouse button, the area marked by the frame will be displayed in the working window.

The working window contains a number of additional control and display elements. The title line indicates the name of the current "job" and the active "layer" (see below). Underneath the title line you find a coordinates display. The coordinates (position and size of an object) can be entered manually in some functions (move, scale) to enable you to define a precise position and size. Next to this there is a display of the object type (path, group, vector text) and of the "group depth" (see below), as well as arrows to switch between different "layers" (see below). You can, optionally, also display horizontal and vertical rulers.

There are some basic rules for working within the window: each object must first be selected by clicking before you can edit it. It will then be displayed with a frame around it featuring "control points". With many manual tools (you will remember the division of the tools in the boxes) you can simply leave the mouse button depressed; the object will then first be selected and you will then directly enter into "action mode" (e.g. "Move") which will be indicated by a different–shape mouse pointer. Each manual editing action is carried out with the mouse button down; if, for example, you want to proportionately enlarge or reduce an object, you begin by selecting the required tool in the toolbox. Then you select the object (if it is not already selected), "grab" the object by one of the control points using the mouse and then drag the object, with the left button depressed, to its larger or smaller size. A special advantage of DA'S VECTOR is that you can always see what you are doing. Thus, in our example, it is not only the frame that is enlarged or reduced, but the object itself is directly adjusted to the mouse movements. For reasons of speed, the object is drawn in Quickdraw mode, i.e. only in its outlines, just like when you select the visible screen section. Upon release of the mouse button, the manipulation of the object is completed and it is drawn again complete with fills. You can abort any manipulation by pressing the right mouse button when releasing the left one. The object will then be redrawn in its original position, shape and size.

You often need to select more than one object, to group together several objects into one for example. If you click on an object with the SHIFT key pressed, it will be selected in addition to any already selected objects, or, if already selected, deselected. Another possibility is to draw up a frame with the ALTERNATE key pressed: all objects which are completely located within that frame will then be selected (after you release the left mouse button).

When there are many objects on top of one another, it is often difficult to retrieve one particular object for selection. Here, too, DA'S VECTOR offers a clever tool: with the CONTROL key down, you can cycle through several objects in turn – in other words, each click selects another object. To select one object firmly, you need to press the SHIFT key in addition and click again: using this method, several objects can be picked from one large heap.

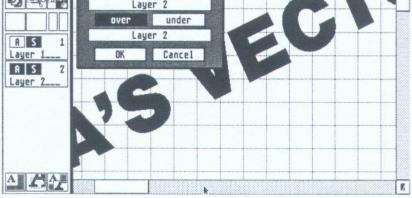
The doubleclick, too, has a special function: you can use it to open objects "temporarily". This requires a more detailed explanation: Basic objects are "paths" which are normally created in the path editor. Such path objects can be grouped together with others; such groups can in turn be grouped together again, etc. You can therefore produce a nesting of object groups to any depth. Now imagine that, for example, you want to change the colour of a path object within a group which is itself part of a group. Normally you would now have to ungroup the top group, then the subgroup, change the colour, then group the subgroup again (select all objects and call up the Group function) and finally group the top group again. But there is a simpler method: Doubleclicking on an object ungroups a group temporarily, and you can do this at any number of levels. The current "object depth" is indicated at the top right of the working window by a number. Also, only the open objects are drawn; all other objects disappear. In our example, you would select the required subgroup with your first doubleclick, and the required path object with the second doubleclick. You can now change the colour. But how do you "close" the object again? To do this, click on the object depth display figure at the top of the window. Each click on this closes one level. In our example, therefore, you need to click twice and are then back to your original object. If, by the way, you carry out another doubleclick on a path object, you will find yourself in the vector path editor and will be able to edit the path. In this event, you can only exit the path editor via the Return icon, as explained above, and not via the menu bar.

Thus all work within the window is done with the mouse, with the SHIFT, ALTERNATE and CONTROL keys modifying the mouse functions in a multitude of ways. In principle, the left mouse button controls all editing functions, the right mouse button selecting the image section displayed in the window or aborting a left mouse button function. Most of you will control the mouse with your right hand and will therefore have the left hand free for the control keys or command selection (see below). There may be additional options with certain functions; sometimes, for example, the right SHIFT key is also used. You will find details under the relevant functions in your electronic reference manual.

Thousands of transparencies:

The Layer system

For the sake of simplicity, we have compared the working area with a large sheet of paper. But that is not quite right because in reality this working area consists of an almost unlimited number (65,536) transparencies, called "layers". The purpose of this layer structure, familiar from professional CAD software, is to simplify work and the combination of individual objects into one complex object. Each of



these "transparencies" can be edited separately and contain its own objects. Only one layer can be active at any time, so that you can only edit the objects in that layer. The inactive layers can be made invisible if needed in order to speed up the screen redraws.

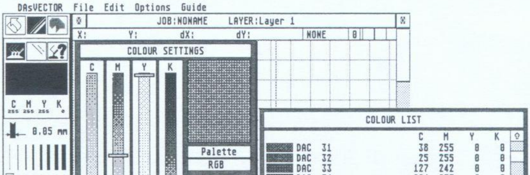
The sequential order of these layers is important for the screen display (they cover each other – remember the "Painting Model"). The bottom layer is drawn first, followed by the second from bottom, etc. For this reason it is possible to re–sort the layers. It is also, of course, possible to copy or move graphics objects from one layer to another. All these functions are found in the Layer menu which can be accessed as a submenu from several toolboxes.

The layers are of special importance in the animation part, since a "film" is made up of such transparencies. Here, however, you can have only one object (however complex) per layer. The order of the layers again determines the sequence in which they are drawn, as in background and foreground or the way they overlap when two moving objects "meet".

The ability to separate the layers at printing also offers a kind of spot colour separation facility.

Colour systems, attributes, fill patterns

As already described in the first chapter, DA'S VECTOR always works internally with 16.7 million colours. You can use two different colour systems: the RGB and the CMYK systems. In principle, both systems can be converted into each other, especially if you



		C	M	Y	K
DAC	31	38	255	0	0
DAC	32	25	255	0	0
DAC	33	127	242	0	0
DAC	34	204	255	0	0
DAC	35	255	255	0	0
DAC	36	216	255	0	0
DAC	37	229	255	0	0
DAC	38	255	204	76	20
DAC	39	255	127	0	0
DAC	40	255	76	25	0

omit the K (black) part from the CMYK system. But since the colour definitions are interpreted differently when exporting and printing, and since the black component can also be used to generate pure halftones, you should be very familiar with the subject. It is therefore necessary to describe both systems in a little more detail.

The RGB system displays all colours through the additive combination or mixing of the three basic colours Red, Green and Blue. This method is applied in colour monitors and colour televisions, where each screen pixel consists of three dots in the basic colours at various intensities. The effect is integrated by the eye into a composite colour. Naturally the background colour is black, i.e. zero intensity or the absence of all colours.

Of much greater importance is the CMY or CMYK system, since virtually all methods of reproduction (print, photography etc) make use of the subtractive basic colours Cyan, Magenta and Yellow. They are called subtractive because they create colour by filtering one component from white light. This also shows the link with the RGB system: Cyan filters red light, Magenta green, Yellow blue. The "background colour" is, of course, always white. The additive basic colours Red, Green and Blue can each be generated by combining two subtractive basic colours: Cyan + Magenta = Blue, Cyan + Yellow = Green, Magenta + Yellow = Red. Black is produced by combining all three subtractive basic colours, but this only works in theory in most cases, since the "black" tends to turn out a dark brown. This is because of the impurity of print colours which cannot be produced at better quality and at reasonable cost. For this reason, black is usually added as the fourth basic print colour, since this colour is normally involved anyway for printing text. Thus the CMY system is extended into the CMYK system: you should, however, be aware that this "four–dimensional" colour system is ambiguous and over–dimensioned and cannot therefore be simply converted into the RGB system unless you know the separation process (the conversion of a 3–colour into a 4–colour system, the so–called process colours) very well and know how to reverse it. Process colour technology is very complicated in general because it is so dependent on many detailed conditions of the process of reproduction. With colour pictures, especially, which you can use as fill patterns, the problem of colour correction comes into consideration because without such correction the colours will be reproduced inaccurately. With DA'S VECTOR, however, you need not bother with all these complicated problems, as all these things are automatically optimised by the printer driver wherever they are actually required. The only thing you need to know and keep in mind is the way in which DA'S VECTOR handles colours defined by you in the RGB or the CMYK systems. The following rules apply:

1. For RGB output, e.g. in animations or colour pictures:

RGB colours are output unchanged. CMYK colours are converted to RGB by adding the black component to all three colours, for example: 50% C, 60% M, 70% Y, 10% K (black) results in 60% C, 70% M, 80% Y and this, in turn, produces 40% Red, 30% Green and 20% Blue in the RGB system. We therefore recommend that for such applications you work in the RGB system from the start.

2. For CMY output, e.g. a 3–colour printer:

RGB colours are corrected, where necessary, and output as CMY. With CMYK colours, the K component is added to the colours (as described above) and the resulting CMY values are output directly without modification.

3. For CMYK output, e.g. 4–colour printers:

RGB colours are separated, corrected and output as CMYK values; CMYK colours are output unchanged. This gives you the choice between two alternative methods of defining your colours: on the one hand, you can fix the composition of the four process colours in absolute terms for printing, by using a colour scale for example. But then the on–screen colours will greatly differ from that in appearance. Alternatively, you can define the colours in the RGB system, with DA'S VECTOR automatically converting them into the "correct" or at least the best possible colours for printed output. But please remember that very brilliant, (almost) pure colours like those on the screen are rarely possible in print, so that you must always take into account a certain loss of brilliance and colour purity.

4. For Black–and–White printers, e.g. laser printers:

Here, of course, you can only produce different degrees of brightness (through raster screening). All RGB and CMYK colours are converted into brightness values, with the K component again added to the colours as described above.

You can, of course, also generate pure halftone graphics: select CMYK as the colour system, leave C, M and Y at 0 and simply select the required grey value on the K scale.

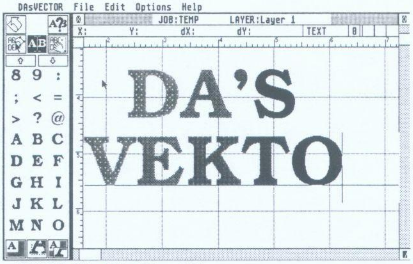
You find the colour definition in several places in DA'S VECTOR, always using the same submenu and the same "slider" system: you select a line colour under line attributes, an area colour under area attributes (submenus for lines and areas); you can also define a start and an end colour for vector text (see below) and define the colours of many single elements in presentation graphics. Using the multicopy function, you can define two colours and thus generate colours gradients very simply. For animations, you can make the colour of objects (lines, areas) change during run– time.

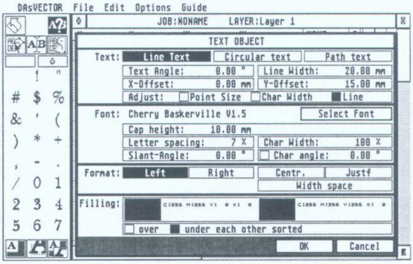
Vector areas (but not lines) can, in addition to a colour, also be filled with a halftone or colour picture as a "fill pattern". The position and the (display) size of the fill pattern can be freely selected in absolute terms as well as in relation to the position or size of the object. Halftone pictures can also be assigned a colour. The possibilities stretch from the simple incorporation of a picture (inside a rectangular frame) through vector backgrounds to the classical fill pattern effects, in which graphic patterns are repeated at selectable intervals. In contrast to the familiar fill patterns in the Atari operating system, these fill patterns are resolution–independent (within certain limits). The colour or halftone pictures can be of any size, since they are handled in virtual memory (as separate files on the hard drive) and loaded into the computer only when required (sometimes scaled down). They must be available in the TIFF block formats (TIH=TIFF halftone, TIP=TIFF palette picture, TIC=TIFF colour picture). The bundled graphics converter allows you to convert many graphics formats into the required picture formats. If you require natural structures as fill patterns, you can scan these with, for example, an Epson colour scanner and our GT LOOK II software, and save them directly in the appropriate formats. The same formats (except TIP) are also used by the tracer.

In addition to the colours, you can also set a number of other parameters for all path objects: the fill algorithm for areas, the line thickness, line pattern and the shape of the line ends and corners for lines. All these settings are found in the relevant submenus for lines and areas.

Vector text and vector fonts

DA'S VECTOR features many kinds of text display: Line text, circular text, path text (text on a curved vector path); the text can be formatted left and right justified, centred as well as fully justified; it can also consist of multiple lines or follow multiple paths. This allows text settings of virtually all shapes and forms. Despite these abilities, DA'S VECTOR is not a text setting program as it only handles graphics text which can be freely manipulated like all other graphics objects. You should keep one important aspect in mind: Text is a special kind of graphics group and cannot be freely transformed; in many instances you need to convert it into a "normal" graphics object first. The program carries out this conversion automatically whenever required. But it gives you a warning first, because once a text object has been converted into a normal graphics group, it can no longer be edited in normal text mode. It is therefore no longer possible to alter the text. The reason for this is straightforward: with text objects, the program obtains all path information from the vector font used. Hence a frequently occurring letter needs to be present only once, in the font itself. That saves a great deal of storage space. But if you want to project such a text onto a three–dimensional Bezier grid, all graphics paths must be available separately since a particular letter may be transformed differently depending on its position.





TEXT OBJECT

Text: Line Text | Circular text | Path text
Text Angle: 0.00 ° | Line Width: 20.00 mm
X-Offset: 0.00 mm | Y-Offset: 15.00 mm
Adjust: □ Point Size □ Char Width ■ Line

Font: Cherry Baskerville V1.5 | Select Font
Cap height: 10.00 mm
Letter spacing: 7 % | Char Width: 100 %
Slant-Angle: 0.00 ° | □ Char angle: 0.00 °

Format: Left | Right | Centr. | Justf
Width space

Filling: C:255 M:255 Y: 0 K: 0 | C:255 M:255 Y:255 K: 0
□ over ■ under each other sorted

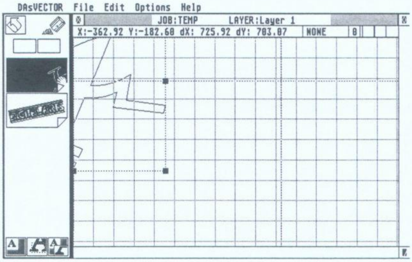
OK | Cancel

Which fonts can be used? DA'S VECTOR uses two font formats directly: CFN (Calamus Font Notation) and DFN (Didot Font Notation). Fonts in the CFN format are available direct from the vendor (DMC GmbH, Walluf, publishers of the Calamus DTP software) and are those used by the Calamus desktop publishing program. DFN fonts are not commercially available but the bundled font converter allows you to convert virtually any industry– standard PostScript Type 1 font into the internal DFN format. The conversion even supports the so–called "hinting", which provides a much improved output quality at low resolutions (screen, laser printers). Converting a text object into a graphics object, however, will also lose the hinting. Through these two font formats, you have thousands of professional typefaces available to you. The "Speedo" format due to be used by Atari was not available yet when this manual went to the printers, but will also be supported by future versions.

Like pictures used as fill patterns, fonts must also be available as separate files on the hard disk and are not included in the documents. The program simply remembers the font name and loads the font when required.

Library, clipboard, trashcan

To conclude this chapter, we would like to introduce you to three further basic components of DA'S VECTOR. The first one is the library system. You can, of course, save and load your graphic creations individually (and in different formats). Alternatively, however, you can store these graphic objects (especially the more frequently required ones) in a library. Such a library is similar to a graphic database integrated into the program and using a card index system; the objects are given a name under which they can be found; but the most important things is that they are always available from within the system and can be selected "on view". Using the library is very simple: to archive an object, you just move it into a free slot in the library (the program will prompt you for a name); whenever you need an object from the library, you "pull" it onto the working area. Each "card" shows the graphics object in scaled–down form; a doubleclick produces a full–colour preview. You can, of course, keep multiple libraries (for customers or particular topics) and freely switch between them. Libraries are also available in the vector path menu and the supplied library already includes many ready–made primitives such as circle, rectangle, simple polygons etc. If you should miss such primitives as circles and rectangles in DA'S VECTOR, have a look at the (path) library. You can, of course, expand this library as you wish.

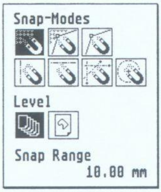

DAsVECTOR File Edit Options Help
JOB:TEMP
LAYER:Layer 1
X:-362.92 Y:-182.60 dX: 725.92 dY: 703.87
NONE
8

The clipboard is very similar in use to the library. You may know this function from other programs, but DA'S VECTOR offers a few special features here: you can place a virtually unlimited number of objects on the clipboard and retrieve them again; the clipboard is saved with the "job", therefore not lost on quitting the program (provided you have saved a job file). In addition, as with libraries, the objects are displayed in Quickdraw mode with a full–colour preview upon a doubleclick. In use it is just as simple: Grab object with the mouse and push it into the clipboard, or pull it from the clipboard onto the working area. A copy will be made if you have activated the copy icon, otherwise the original is placed in the clipboard; this applies in both directions. If you press the CONTROL key when retrieving an object, it will be placed on the working area in precisely the same position it was copied to the clipboard from.

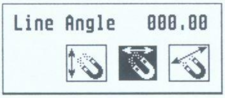
The trashcan is a close relation to the clipboard, you will notice this only in exceptional cases, as graphics objects are "disposed of" by simply dragging them onto the trashcan icon. However, if you doubleclick on this icon, the trashcan is opened and all the trashed objects are once again available. The contents of the trashcan is not, however, saved and whenever the program begins to run short of memory, it will gradually empty the trashcan; you can also empty the trashcan yourself (press RIGHT SHIFT key and click on trashcan).

Construction aids

DA'S VECTOR features a complex system of construction aids, such as a freely configurable grid, guide lines and guide circles. All these construction elements (and much more) can be "snapped to" in all interactive functions, and the snap radius is also freely configurable. In other words, as soon as you get within a precisely definable distance of these "magnetic" elements when drawing, moving, scaling etc, the mouse pointer will "jump" towards them. This "snap to" function can be configured in a variety of ways: Snap to grid, horizontal, vertical or other lines, circles and even any curve control points of another graphics object. In addition to the snap radius you can also select whether or not



Snap Range 10.00 mm


Line Angle 000.00

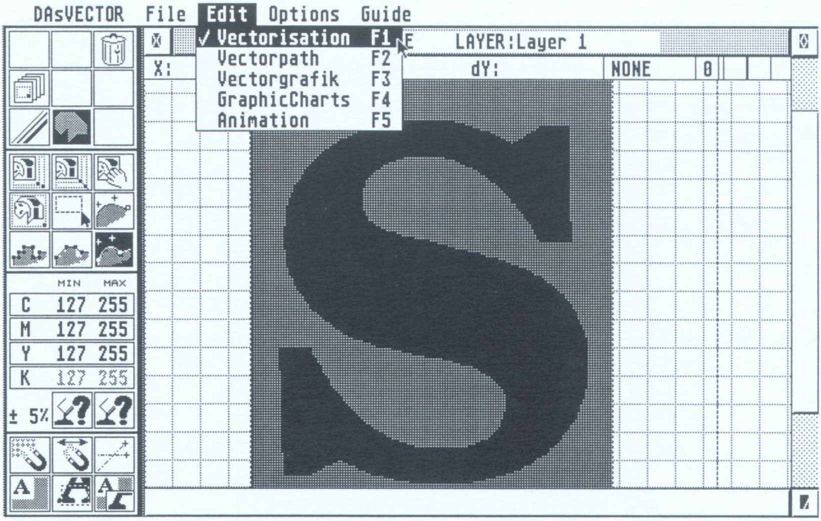
object. In addition to the snap radius you can also select whether or not only objects or construction aids on the currently active layers or on all layers are to be active. This "snap to" is a lot more complex than you will gather even from these descriptions, as can be seen from the following example: let us assume that you move a graphic object using magnetic guide lines; if you grab the object by the top left corner of its "bounding box", it will be precisely this corner which snaps to the guide lines when you come near them. But if you have grabbed the object by a curve control point, this control point will snap to the guide line when you are within the snap radius of a line. The guide lines can be drawn at any angle; alternatively you can set horizontal or vertical guide lines with a mouseclick on the working area, or draw up guide circles.

Another important construction aid is the limitation of the direction of mouse movement. You can limit the mouse movement to vertical or horizontal directions or even to a freely definable angle. This applies to all activities inside the working window, including the drawing of guide lines. So if, for example, you require an angle of 60 degrees, you can set this accurately while drawing or begin by setting a guide line at that angle.

All these functions are globally available; as already discussed above, the required tools can be found in every toolbox. The necessary settings can be made via pop–up menus which are activated by a doubleclick (or SHIFT+click) on the relevant icons.

Vectorisation of Images

The Tracer



	MIN	MAX
C	127	255
M	127	255
Y	127	255
K	127	255

± 5%

Imagine the following case: a new customer wants you to create a layout for a letterhead; he already has a business logo which, of course, he wants included in it. He gives you a coloured business card containing the logo to work with. If you have a scanner, you can scan this card. Using appropriate image processing software you will probably be able to cut the logo out from its background and adjust the colours. And yet this image is only usable with strict limitations, above all because you require it in various sizes. In other words: you need the logo as a freely scalable, resolution–independent vector graphic.

A tracer or "vectoriser" is designed to filter certain parts out of a pixel image and to generate their outlines, as accurately as possible, as a vector graphic. But what is the best possible accuracy? What is required for a computer program to achieve the best results when automatically determining the outlines? This depends on the one hand on the quality of the original, of course, with multicoloured originals presenting particularly difficult problems, and on the other hand on the built–in tracing algorithms. There is probably no such thing as an ideal recognition algorithm capable of producing optimal results automatically in every case. One method of improving the result is to give the user the ability to set the tracer parameters, i.e. configure it for the requirements of a particular job. But that is not all that easy because we are dealing here with complex mathematical procedures. One cannot, after all, expect the user to understand the recognition algorithms. DA'S VECTOR follows a completely new approach here by always working with halftone or colour pictures, while vectorisation normally uses monochrome pixel graphics. Apart from the additional advantage of being able to read colour originals, this provides an easy method of setting the parameters via colour and brightness thresholds. To understand this, think of how a scanner works. When generating a monochrome picture, it needs a threshold value to determine at what point a pixel is considered as white or black. Now imagine that one element of the scanner meets precisely with a borderline which is then assigned a medium grey value such as 40%. If the threshold is preset to 50%, the dot will appear white even though it encroaches on the pixel to an extent of 40%. Some data is lost which will later be missing when it comes to defining the contour. If the same image were to be scanned in grey scales (as must be done for DA'S VECTOR), you would obtain a grey value of 40% for the pixel in question. The data is preserved in this case and can therefore be used by the tracer. The interesting thing about this, however, is that it makes it easy to parametrise the result of the vectorisation very simply through selection of a colour threshold value. Hence no more complicated mathematical parameters to painstakingly try to improve the results of the tracer, without really understanding what it is that you are doing! You determine the threshold for the tracer yourself, and the program offers you useful aids in this. You can, for example, simply pick the two colour thresholds (upper and lower) from the picture itself, or pick one single colour from which the program will calculate the thresholds to your own specifications (tolerance in %). Due to the scanner "noise" there are no completely uniform colour areas, so that you always need to define a "colour space" (a section of the colour range) or a "grey space" to be vectorised in order to compensate for the colour variations of the original and of the scanner ("noise"). Using the threshold values, you can therefore determine the object colour (and hence the object to be vectorised) as well as influence the quality of the vectorisation.

Since the result will probably not always be fully satisfactory, even with the best tracer, DA'S VECTOR (in the path editor) offers an option to edit the vector graphic manually "against the original", with a visible background picture. Since you can work at an extremely large magnification, such corrective editing is very easily and very accurately done.

This approach does, however, have one drawback: the image files can become very large. For this reason, DA'S VECTOR works with "virtual memory" and this is therefore not at all limited to a picture which can fit entirely into memory. This has the advantage that you can work at any scanner resolution, in itself a contribution to improved vectorisation quality. The required hard disk space should not be a problem since the pictures will not normally be required after vectorisation and can therefore be deleted. You can, of course, also vectorise monochrome images; you only need to convert them temporarily into halftone format first. In this case the threshold method cannot produce any quality improvements.

The tracing function can, incidentally, also be used for other purposes: in the path editor to convert a line into Bezier curves; and similarly when drawing freehand the Bezier tracer can be used to automatically smoothen the lines.

To conclude this chapter, a few hints on using the "vectorisation" part of the program: The "Import" function (in the dropdown menu) only loads pictures in the "TIH" (TIFF halftone) or "TIC" (TIFF colour) formats. All other graphics formats can be converted into these formats with the conversion program.

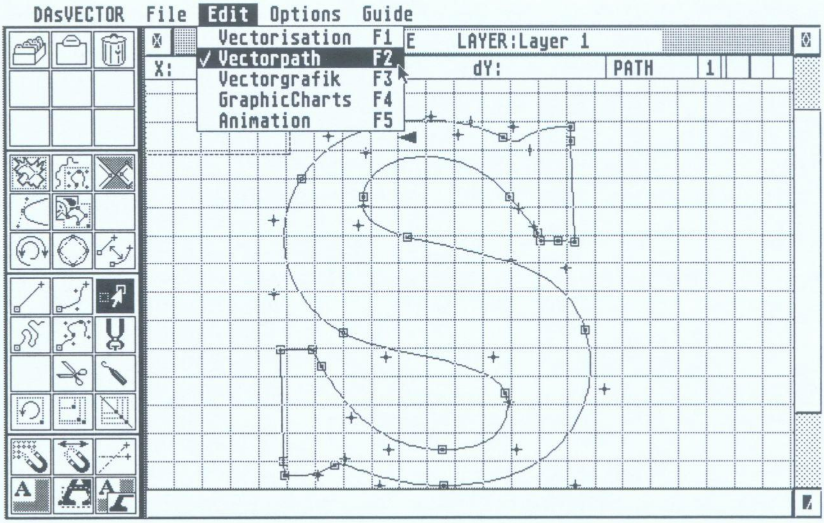
The toolbox offers the following functions:

Submenus, Layer (to select the layer in which the graphic is to be placed) line attributes, area attributes (the vectorised graphic acquires the attributes which you have selected here; of course this can be changed subsequently but this can be more complicated if you have a complex graphic with many paths) – trashcan (to delete, of course). Picture functions: the background picture (to be traced) can be modified in its position, size and proportions (distortions). Picture size and proportion also determine the size of the graphic generated. You can make the picture as large as you like. The distortions are of course normally unwelcome, but can on occasion be very useful.

Generate picture: this is a special function which generates a vector graphic object with the picture as a fill pattern. Very useful if all you want to do is to include pictures (such as photos) in your design, perhaps as a background for animations. Select trace area: you "cut out" a section of the picture; only this section will be displayed and vectorised. Trace: starts the vectoriser. There are three icons to select the type of vectorisation: lines, optimised lines, Bezier curves. Threshold values: here you find four pairs of numbers for the subtractive basic colours Cyan, Magenta, Yellow and Black (or K= contrast, all the same thing). With halftone pictures you can only enter the K value for the lower and upper threshold, with colour pictures for C, M and Y. Once again on the meaning of the threshold values: all colours (halftones) between the lower and the upper threshold (including the threshold values) are interpreted as ONE colour, the outline of which is vectorised. To define these colours, use the Pick function for the lower and upper threshold: select the icon click somewhere in the picture and the values will be entered. If you click on one of the two icons with the SHIFT button depressed, you can pick a "medium colour" from the picture; the threshold values will then be set according to the selected tolerance value (to the left of the Pick icons). So if, for example, you want to vectorise a yellow text item from a colour advertisement, you simply click on the text using this function; a tolerance of 5–10% will normally compensate for the scanner noise; depending on the background colour and picture quality the tolerance value can also be chosen larger or smaller.

Drawing and Flexible Shapes

The Path Editor



The path editor is the "drawing part" of DA'S VECTOR. If you do not construct a graphic from predefined primitives or obtain the outlines by tracing an original, you will begin by "drawing" in the path editor. This "drawing", however, is quite different from drawing on paper; it is more of a construction process, akin to technical drawing, although functions such as "freehand drawing" are somewhat more reminiscent of manual work on a piece of paper. We do, however, think that the computer (at least for the time being) cannot replace rough sketching on paper. This will only be possible when you can use a sort of "pen" on a (flat) high–resolution, colour screen with absolute fluency; today's technology is still several years away from this.

"Drawing" or constructing vector graphics requires just as much manual dexterity as drawing on paper; the computer and the software will never completly be able to replace basic skills even if the possibility of permanent correction makes work a great deal simpler. Creative abilities will always be reserved for humans; the computer being nothing but a tool just like the drawing pen, albeit a much more complex one. Since vector "drawing" differs from the familiar manual drawing, you will need to develop a special technique for it, as for any other tool. This technique can only be acquired through practice; even at the computer nothing comes easy.

As already explained in the first chapter, each vector graphic is initially defined by outlines (paths); these outlines are later assigned "attributes" such as line thickness, line colour, area fill colour etc which can be changed again at any time. The setting and alteration of these attributes is done in the vector graphic part which we are going to examine in detail in the next chapter. For now, we are concerned only with the drawing or construction of the outlines. The basic elements for these outlines are straight lines and Bezier curves. For this reason, there are two different tools to "draw" lines and Bezier curves. This "drawing" is done by continuously setting points with the mouse, with the program following the mouse movement and displaying the appropriate lines, so that you are always in visual control of the result. Initially, the two tools do not appear different from each other; even the Bezier curve tool creates straight lines first which are only distinguished from ordinary lines by two control points (visible as small crosses on the line). It is only when you switch on the "permanence" of the Bezier curves that curved lines are immediately generated, so that you can "draw" organic–flexible curves straight away in this mode. This is difficult to describe in words; just try it out for a while and you will soon understand how this works. As in all other parts of DA'S VECTOR, you can always see what you are doing. Before you set the new end point of a line or Bezier curve, you can always judge what the final outcome would be.

A few more hints on the drawing technique: you will continuously generate lines and curves; if you want a break in that continuity, simply press the CONTROL key before setting the next point (the start point of a new line). You can switch between straight lines and Bezier curves at any time while drawing, by clicking on the relevant icon in the toolbox and then drawing on.

Alongside these more "constructivist" drawing methods there are also two different methods of "freehand drawing". These are not so different from one another during the actual drawing; you simply draw on the working area with the left mouse button depressed – the program will continuously generate short line segments. The difference between the two tools only becomes visible when you release the mouse button; in Bezier mode the line segments are now automatically transformed into appropriate Bezier curves so that you obtain flexibly rounded shapes. Another function also lets you "trace" a line consisting of straight segments into Bezier curves. For really fluent drawing you need a fast computer such as a TT. You can greatly increase the drawing speed on slower models, however, by working in monochrome mode.

In addition to the pure drawing functions, you will also find in the path editor many functions to edit already created vector paths. To simplify the description of these functions, we would like to first clarify some of the terminology used in what follows: by path segment we mean a linked sequence of lines consisting of straights and/or Bezier curves; a path can consist of any number of such path segments (subpaths or path sections); start and end points are distinguished and displayed differently; curve control points are the start and end points in straight lines; in Bezier curves you also have the two control points which determine the curvature of the curve; each path segment has a rotation direction indicated by a small arrow next to the start point; this rotation direction determines how the fill is drawn when two paths intersect with each other (more on that in the next chapter).

Next to the drawing tools, the most important editing tool is that for moving and reshaping. The possibilities are enormously varied and can hardly be described comprehensively – you will just have to play around with it. But of course you also need to know something about it before you start, so here is a brief description of its use: Firstly, you can simply grab a curve and reshape it with this instrument; the effect depends on where you grab it: if it is the start or end point of a curve or line, then only that point will be moved; if you grab a line at its middle, the entire line will be moved; if you grab a Bezier curve at its middle, the curvature will follow the mouse movement; if you grab a control point of a Bezier curve, only that control point will be moved, which will also change the curvature; the effect is always dependent on whether the "permanence" of the curve has been switched on or not: in the former case the neighbouring Bezier curves will also be affected. But you can also first select any number of curve point (including the control points) (as always when multiple objects are selected, with SHIFT+click) and then move them together. To this end, the program offers some further aids: CONTROL+click selects all points simultaneously, ALTERNATE+click all points of a path segment.

There is a further plethora of tools and functions which can only be listed briefly here: please find further details in the electronic online manual.

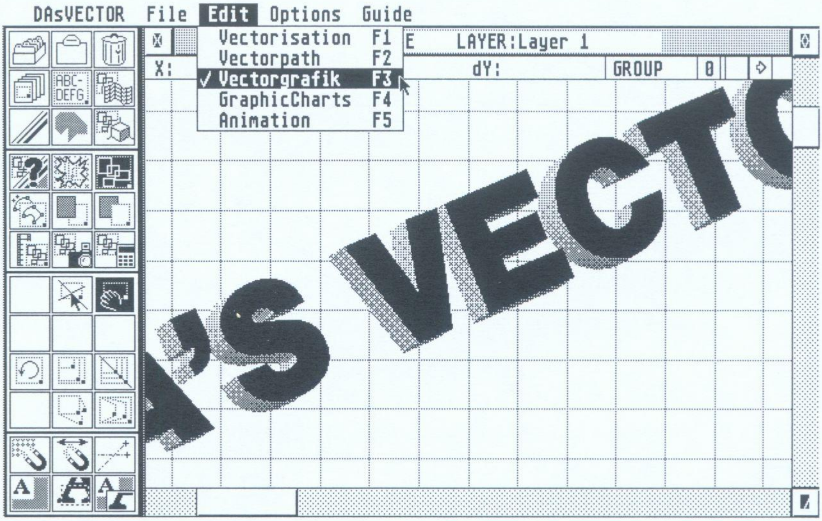
The "pincers" are used to remove single curves or lines from a path segment, while still leaving the line sequence closed or linked. The "scissors" are to cut curves, i.e. separate them into two subpaths, and the "crochet hook" joins them together again (but only at the start and end points). Additionally, you can rotate path segments or the entire path (depending on what you have selected) and scale them proportionately (enlarge, reduce) or freely (distort). The rotation tool has been designed particularly flexibly; you can freely position the rotation centre, which is indicated by a small symbol, and also choose the rotation axis since it is possible to grab every single curve control point as well as the corners of the "bounding box". Of course you can also enter the rotation angle from the keyboard, as you can control the positioning (Move) or scaling via keyboard entry of the coordinates.

There are some functions in addition to these instruments. You can change the rotation direction of a path, "round and close" a path, to convert a subpath into a full object (it will then disappear from the path editor because it becomes a separate path object) and to change the type of a path element: Bezier curve to line and vice versa. Another function joins to intersecting paths. If you have imported a picture for the tracer, this picture can be displayed in the path editor to be able to touch up the path created by vectorisation with the original in the background.

Three already familiar submenus are available in the vector path menu: the clipboard can store and copy paths or path segments; in the library there are a number of simple basic shapes which you can build on yourself; the trashcan from which deleted elements can be retrieved again, is already known to you. All these functions work exclusively with paths in the path menu, other objects remaining invisible.

Colour, Text, Complex Transformations

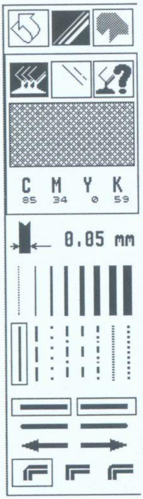
The Vector Graphic Editor



In terms of functionality, the vector graphic editor is the most comprehensive part of the program. The basic objects created in the vector tracer or the path menu can be assigned attributes (line attributes, fills) and combined into more complex objects in a variety of ways. These complex objects, including those created in the presentation graphics menu (see the next chapter), can be further modified and transformed and the "generator functions" can generate very complex objects from more simple ones (for example all kinds of colour gradients). With the vector text functions you can create "text objects" with all kinds of possibilities for setting text in various shapes and forms. Vector graphics objects can be imported or exported in different formats, using the various "drivers" offered by the program.

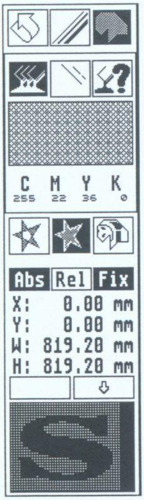
Attribute functions

To begin with, paths are "nothing" in graphical terms; they are mere outline descriptions which are only given substance by the attributes. A distinction is made here between line and area attributes; for both there is, as already mentioned, a separate submenu, each of which will be discussed now in some detail. You can, incidentally, switch between the two submenus at any time without having to first return to the main menu.



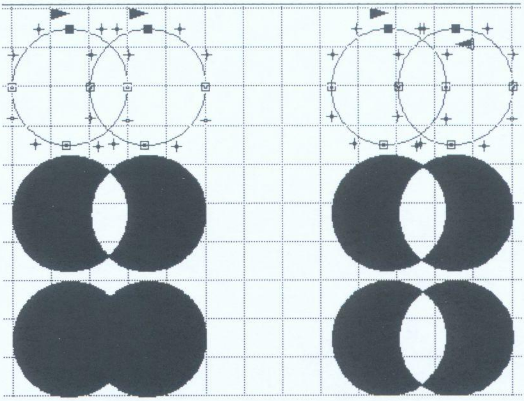
For a path object to be assigned attributes, it must first be selected; if the path is already part of a group the group must either be ungrouped or opened as has already been described. A group object can contain any number of path objects with their own attributes so that it would make no sense to assign attributes to a group.

A line can be transparent (that is, absent) or coloured, and you can select the colour in the RGB or CMYK system. To generate a pure halftone graphic (with grey levels only), you simply use the K component of the CMYK system only. You can also "pick" the colour from another object and thus copy colours from one object to another in a simple fashion.



The line thickness can be selected from 8 predefined values or freely set via the keyboard. In addition, you can select between 8 predefined line patterns. Both line ends can be separately assigned a shape (cornered, round, arrow), and the shape of the corners (joints between two straight lines) can be selected: corner, mitre, or round.

The colour selection for the fill areas is done in exactly the same way as for the lines. Of course an area can also be transparent if you want an object made up of lines. The setting of the fill algorithm is particularly important and not all that easy to understand. DA'S VECTOR, like PostScript, uses two different fill algorithms with the exotic names "Winding number" and "Even–odd"; this is, of course, only meaningful to the mathematician. In many cases, as with "simple" areas, there is no visible difference. The difference is only visible if two path sections cross each other. If you tend to understand concepts better visually, have a look at the adjacent illustration and experiment a little with this and other examples; you will soon acquire an intuitive understanding of the two different methods. For more analytically inclined users, however, we can also give an explanation. Imagine two circles which half overlap. Then imagine a horizontal line cutting through the entire object and crossing the two circular paths at four points; with the "even–odd" algorithm the fill changes at each of these points, in our example: filled, transparent (the area where the two paths overlap), and once again filled. With the "winding number" algorithm, the middle section can be either filled or transparent, depending on the rotation direction of the two subpaths (which is irrelevant with "even–odd"): if the rotation directions are the same, it appears transparent. You should prefer the "winding number" method as it is more versatile.



In addition to the simple area fill with colours DA'S VECTOR, as already mentioned, also offers the unique option of filling areas with halftone or colour pictures. These can be of any size since they are kept in "virtual memory" on the hard disk. Halftone pictures can also be assigned a freely selectable "base colour" to achieve incredible effects: vector areas can be given natural structures, pictures can be masked with any shape and the fill patterns familiar to you from many paint programs can now, at long last, be used in print or with other high–resolution output media because of their resolution–independence. You can, of course, also simply place the picture into a rectangular frame (the tracer offers a special function to do this) and use it in your graphic like a normal photo. Before you can use a picture as a fill pattern, you have to load it. At the bottom of the menu box there is a picture field similar to the clipboard entries. If you click on this, the file selector box appears and you can select a picture in one of the TIFF block formats (TIH, TIP, TIC) which will then be displayed in reduced form. Of course you can have more than one picture; the number is virtually unlimited and you can use the scroll arrows to click through all the fill patterns loaded. To activate a fill pattern, you need to select the bar next to the picture. To assign this fill pattern to a path object, simply select the relevant icon. There are also a number of refined additional settings which offer a variety of effects: you can anchor the picture on the working area (it will keep its position when you move the object) or in relation to the object (top left corner of the bounding box, in this case it will move with the object). The size can be fixed (select "fix") or remain variable with the object size. If the object area is larger than the size of the picture, the picture will be repeated cyclically, which is also why we speak of "fill patterns" even though a lot of other effects are possible with this function.

A fair number of fill patterns are supplied, but of course only a few colour images can be fitted onto floppy disks, and certainly not at high resolutions. You can create your own fill patterns with practically any paint or image processing program and convert these into the required formats using the supplied conversion utility. If you own a colour or greyscale scanner, you can of course also use it to generate suitable picture material (surface structures, for instance). Our scanner software GTLOOK II for all Epson colour scanners will provide the "correct" image formats directly.

One more word about the picture formats, because we already know the most frequently asked question on this topic (from experience!): Why are only these "exotic" picture formats supported? The answer: to begin with, they are far from exotic because they are simple implementations of the TIFF format, the most important colour image format in the world. Secondly there is an important technical reason: the pictures are handled "virtually", i.e. they are stored on the hard disk and are only loaded into memory by the program when they are actually needed; if the pictures are large, a reduced copy is loaded (into the so–called "cache") and this is sufficient for work on the monitor. The original picture in its full resolution is only used for output (print, picture, animation). This is not possible with any old picture format, at any rate not with acceptable speeds (or do you want to go and make a coffee every time a picture is required?) and so we have chosen an appropriate image format which has already been tried and tested in many other applications.

Alongside the two submenus for setting the attributes you will find another important function in the main menu: you can search–and–replace the attributes fill colour, line colour and line thickness in a group, a layer or globally. This offers an extremely simple way of altering even a very complex graphic.

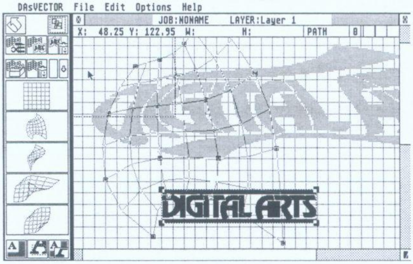
Montage functions

Now you already know how to create path objects and how to assign attributes (rendering instructions) to them. A path can be a very complex structure, but since it can only have one set of attributes (line and fill attributes), the creative possibilities are still rather limited apart from the fill patterns. For this reason, a vector graphic usually consists of many different path objects overlaid on top of each other (remember the "Painting Model" with its opaque colours!). To determine the sequential order of their overlapping, you have two basic functions available: "Place object in foreground" and "Place object in background". Since, however, you would not wish to mess up an already created order, you will group the different path objects into one group which can then be treated as one object and modified in various ways. This is what the functions "Group" and "Ungroup" are there for. To alter a partial object in this group without ungrouping it, it is possible to temporarily open it with a doubleclick as already described elsewhere. Not only path objects can be grouped, but also groups at any depth of nesting. A further function allows you to combine a number of path objects into one single path; this is particularly useful with vectorisations which often produce a considerable number of different path objects. These are combined into a group, and if you ungroup this, all individual objects will initially remain selected and can then be merged into one large path object with a single mouseclick.

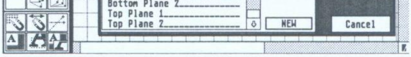
The vector graphic menu offers a total of seven further tools for the manual editing of objects (paths or groups): you can move an object, protect it (against any change), rotate it (with freely definable rotation axis as in the path editor), scale freely and proportionately, skew and distort in perspective. All these functions are executed under full visual control in Quickdraw mode, so that you can always see what you are doing and abort the operation (right mousebutton) if you are not happy with the result.

Transformation functions

In addition to the manual instruments for the manipulation of objects described in the previous section, there are two very powerful tools for the transformation of objects. The first is the projection onto flexibly reshapable Bezier areas, which has its own submenu. In this submenu you will find, in addition to functions for manipulating these Bezier grids, a library of predefined grids which you can add to yourself. In its basic form, such a Bezier grid consists of a "grid" with 4 times 4 "active" points. The horizontal as well as the vertical rows of points (4 each) are linked to each other by a Bezier function. The reshaping tool lets you grab each of these four points and move them; since all the points in the grid structure are linked, the whole grid will be flexibly reshaped. Now you can "project" a graphic object onto this grid using the calculation function and thus realise virtually any kind of flexible reshaping of objects. This is quite difficult to describe in words; you should simply try it out for yourself and examine the predefined grids in the library. You will find this an intuitive method of flexible reshaping which allows you to forget about all the mathematical background and rules.



This is not the case with the second transformation function, the "calculator". Here some mathematical knowledge is required, but in return the possibilities are virtually unlimited. There is probably no transformation which cannot be achieved with this system of formulae. Mathematically knowledgeable users will be quick to recognise the possibilities and, for example, adopt and apply geometric transformations found in specialist literature. But we have also remembered the mathematically less skilled user: the supplied collection of formulae offers many useful functions and examples. You can, of course, expand this collection yourself and load different ones. Perhaps the mathematically advanced users will make the results of their experiments available to other users: free exchange is beneficial to all! So if you have developed new transformations and



```
x'=b+((g*a)/(a-h))
y'=c-((e*a)/(a-h))
a =sqrt(sqr(xmax-xmin)+sqr(ymax-ymin))
b =(xmin+xmax)/2
c =(ymin+ymax)/2
d =pi/180
e =(c-y)*cos(298*d)
f =(c-y)*sin(298*d)
g =((x-b)*cos(315*d))-(f*sin(315*d))
h =((x-b)*sin(315*d))+(f*cos(315*d))
xc = 102.40 mm    yc = 102.40 mm
```

FORMEL:

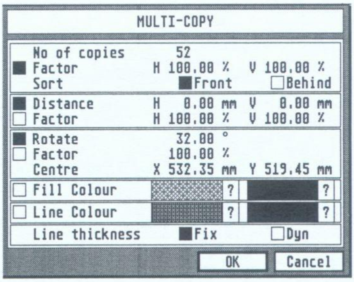
- Horizontal Plane 6
- Bottom Plane 1
- Bottom Plane 2
- Top Plane 1
- Top Plane 2

wish to make them available to the general public, you can send these to us or upload them to the support mailbox. We will then take care of their further distribution.

Generator functions

The functions and tools described so far can be used to transform objects spatially, i.e. to reshape them in some way. The object structure itself remains preserved, if somewhat distorted. The "generator functions" introduced in what follows go beyond that: they can generate a much more complex object from a simple one. DA'S VECTOR features three such generators.

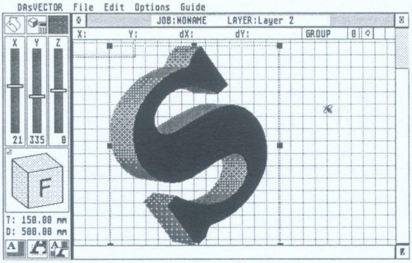
The first generator is the "multicopy function". This can be used not only to generate any number of copies of an object, but every copy can have different properties: the objects can grow or shrink, they can be positioned or rotated with constant or dynamically changing distances and the colour of the lines and areas can change cyclically from a freely definable "start colour" to an equally freely definable "target colour". The tutorial offers some examples which show how this function can be used to create very simply linear and circular gradients and grid structures. But that does not exhaust the possibilities by a long way, as your own experiments will soon show you. The multicopy function creates a number of new objects; you will generally group these together. Of course you can also use this function for such simple purposes as tile montage.



```
MULTI-COPY

  No of copies      52
■ Factor          H 100.00 %   V 100.00 %
  Sort             ■Front       □Behind
■ Distance        H   0.00 mm  V   0.00 mm
□ Factor          H 100.00 %   V 100.00 %
■ Rotate            32.00 °
□ Factor           100.00 %
  Centre          X 532.35 mm  Y 519.45 mm
□ Fill Colour      ▨ ?          ■ ?
□ Line Colour      ▨ ?          ■ ?
  Line thickness   ■Fix         □Dyn
```

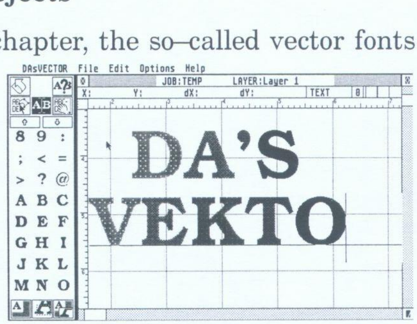
The second generator is the "3D extruder", which has its own submenu. With this function, you can give depth to any shape and position the resulting three–dimensional object freely in all three spatial axes. But that is not all! You can assign a colour to the depth dimension, cast light on the object from a freely positionable light source and thus create



realistic light and shade affects. Despite this, DA'S VECTOR is not a 3D vector program: the extruder function is simply intended to help you generate realistically–looking spatial objects. Upon leaving the extruder, you are back to a normal two–dimensional graphic; it only appears to be genuinely three–dimensional.

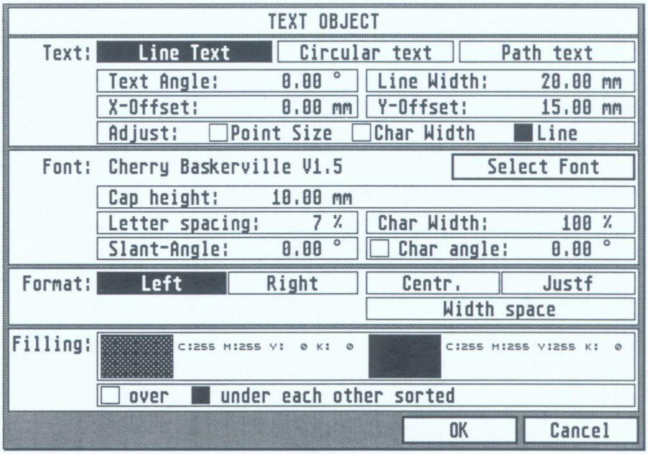
The last generator is the "Time–Space converter". The principle behind this powerful function is almost impossible to describe as long as we have not looked in more detail at DA'S VECTOR's animation part. Just imagine a film strip consisting of a sequence of individual pictures which are now overlaid on top of each other. In other words, a time axis is represented as space – hence the name. Before you can use this function, you must therefore have created or loaded an animation. A mouseclick will then generate a spatial representation of the animation. An important application of this powerful function lies in the generation of colour gradients of any shape, but even such refined effects as realistic shadows with "soft" (blurred) contours are possible. There are some simple examples in the tutorial; these will at least give you some idea of what can be done with this unique function.

Text objects



As already mentioned in the first chapter, the so–called vector fonts consist of the same basic elements (lines, bezier curves) as our vector graphics system. Therefore "text objects" can be integrated into the vector graphic using sophisticated typesetting and design functions. You find all these functions in a submenu consisting of four basic functions and a character selector box in which the characters of the current font are displayed (over several "pages" accessed by the arrow icons). This display is not just for viewing since you can click on the characters in text entry mode and insert them into the text that way; this gives you access to all the special characters. If you have not loaded any particular font (see below), you see the (vectorised) system font.

The first "function" is calling up the complex "Text objects" dialog box. This dialog is subdivided into four parts. The topmost part, "Text", is for selecting between three basic kinds of setting (Line text, Circular text, Path text). Depending on the choice you then get other additional options:



TEXT OBJECT

Text: [✓ Line Text] Circular text Path text

Text Angle: 0.00 ° Line Width: 20.00 mm

X-Offset: 0.00 mm Y-Offset: 15.00 mm

Adjust: □ Point Size □ Char Width ■ Line

Font: Cherry Baskerville V1.5 Select Font

Cap height: 10.00 mm

Letter spacing: 7 % Char Width: 100 %

Slant-Angle: 0.00 ° □ Char angle: 0.00 °

Format: [✓ Left] Right Centr. Justf

Width space

Filling: ▨ C:255 M:255 Y: 0 K: 0 ■ C:255 M:255 Y:255 K: 0

□ over ■ under each other sorted

OK Cancel

With line text, you can enter the text angle and the line width. The X and Y offset determine the distance between individual lines of text as you can also enter multi–line text. The "Adapt" setting determines what happens if you exceed the selected line width: either the font size or the font width can be automatically adapted, or the line is simply made longer. DA'S VECTOR is not a typesetting or DTP program, which is why there is no automatic formatting, but you can still edit text over several lines and manually format it (see below).

With circular text you can enter a start and end angle as well as the radius. You can also choose if the text is to be set on the inside or the outside of the circle. "Adapt" offers the same options as for line text, but of course the radius rather than the line width are enlarged here.

With path text – text running along a path of any shape – you can change the "rotation direction" at any time. "Adapt" offers only font size and font width, an automatic enlargement of the path as in line or circular text is not possible. You must first generate the path for such a text in the path editor. If this path consists of several subpaths (separate path segments) you can also generate "multi–path" text.

The second part of the dialog has the title "Font". If you click on "Select font", a file selector box appears and you can load a font in CFN or DFN format. Other font formats must first be converted. A converter for Postscript Type 1 fonts is included. As soon as it is available, Atari's new Speedo format will also be supported. But selecting a font is not enough. Of course you can also set the point size, but in addition you can add letterspacing, modify the character width, slant the type at any angle or output each character rotated by any angle. Using appropriate combinations of line angle and character angle, for example, it is possible to type vertically from the top downwards.

The third part of the dialog determines the formatting: left aligned, right aligned, centred and fully justified. You can also expand the text to the full line width – this adds not only word spacing as in fully justified, but also inter–character spacing.

The last part concerns colour assignment (fills). In contrast to path objects you can select two colours, a start and an end colour, which may also be identical. That produces a colour gradient within the line, so that each character assumes a different colour. You can also specify in what sequence the characters are to be drawn if, as is the case with negative letterspacing, they overlap each other.

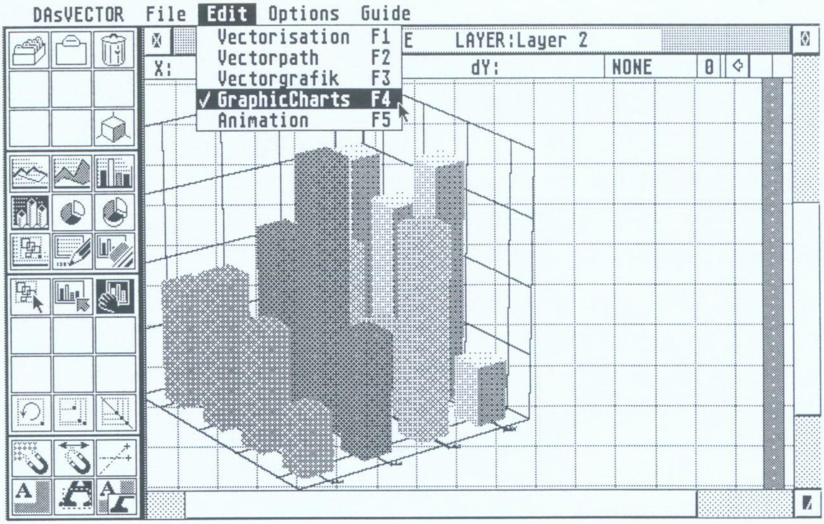
All settings always apply for one line only, or for one circle or path segment. They can be changed subsequently, but several fonts or different settings within one line (or one circle or path segment) are not possible. If you need that, you must put it together from two or more text objects. As we have already said, DA'S VECTOR is not a DTP or typesetting program.

We must now deal with the remaining three functions in the submenu. The most important one is text entry. If you click on this icon, a line or circle will "stick" to the mouse pointer and you can position this object somewhere in the working area and begin to enter text. If, however, you had already previously selected a text object, then you will find yourself directly in edit mode for this object and can alter its contents. You can move the text cursor within the line by using the horizontal arrow keys. You can also jump between different line and path texts using the vertical arrow keys. If you hold down the CONTROL key simultaneously, the text to the right of the cursor is moved to the next line (manual formatting). The RETURN key will take you, as expected, to the next line; if there is none, a new one will be automatically created. The correction keys BACKSPACE and DELETE work as expected within one line. Of course the text is automatically kerned; additional manual kerning is possible using the horizontal arrow keys if the SHIFT (fine tuning) or the CONTROL (coarser) keys are pressed simultaneously.

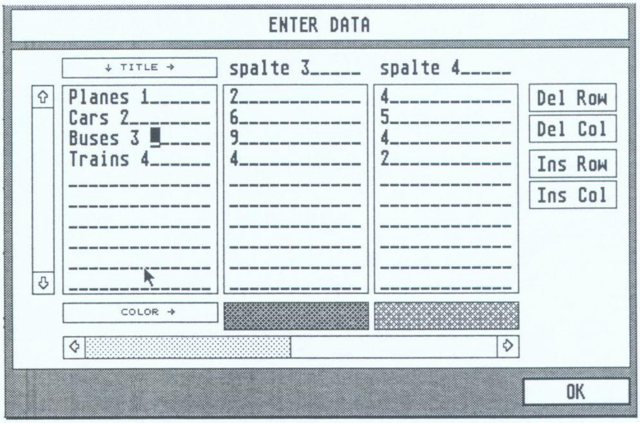
Generate Graphics from Numbers

The Chart Editor

The success of the natural sciences depends on their ability to measure and count the investigated phenomena – the fact that the natural sciences only investigate such quantifiable phenomena and will therefore never find anything else has led many people to the wrong conclusion that therefore there cannot be anything else. Whatever the truth may be, numbers today have an almost mystical power of persuasion; as soon as something exists in numerical form it appears true or at least proven. And where you cannot measure, as in the social sciences, the attempt is made to equal the natural sciences by the use of statistics. Numbers persuade, but they can only be read "sequentially"; it requires more or less effort to interpret figures intellectually. You need to think about them. When (simple) numerical relationships are represented graphically, however, they can be understood "at a glance". The suggestivity of figures is combined with graphical display. Small wonder that politics and marketing have a strong preference for such displays – it is all about "presentation". It should be well known how this can be used to lie and cheat – the possibility of manipulation lies in the selection and combination of the numerical material. The finished result, the statistic, no longer shows how it was achieved at. As long as they are not abused, the so–called "charts" have their justification, but we wanted to at least raise the problem before turning to the presentation graphics part of DA'S VECTOR. This part of the program is designed to meet two requirements: to generate the standard forms of presentation graphics at the push of a button and, secondly, to make possible the realisation of entirely new ideas in tandem with the vector graphics editor.



Presentation graphics or charts are generated from figures. The first question therefore is: where do these figures come from and in what structure do they have to be available? I am not talking of the data file format here – this will be discussed further along – but of the structures which can be represented. Statistics usually contain a comparison, for example the sales figures for a series of products during a given period of time. The most simple form of expressing this is a table. Such a table with a maximum of 8 columns and a maximum of 32 rows is used by DA'S VECTOR as the basis of its presentation graphics.


ENTER DATA

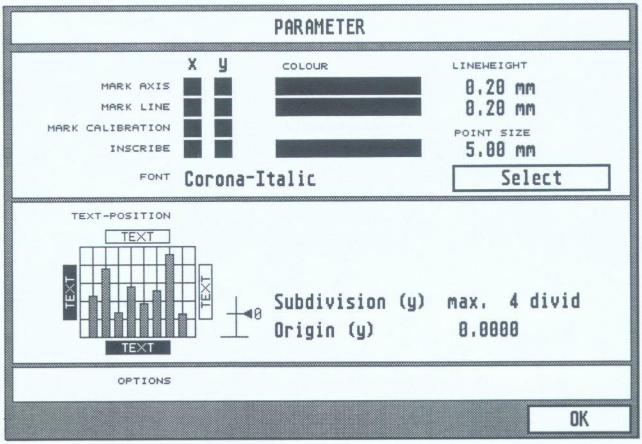
↓ TITLE →	spalte 3_____	spalte 4_____
Planes 1	2	4
Cars 2	6	5
Buses 3	9	4
Trains 4	4	2

This table can be produced in a small editor where you can also enter, in addition to the 8*32 numerical values, 8 column titles and 32 row titles for the legends. In addition, you can select an object colour for each row for the graphical presentation. You will normally work with less data since the viewer's ability to comprehend declines drastically with growing complexity, so that the purpose of the display – the simple and quick comprehension of numerical relationships and statistics – would be missed with too much different data. Psychological research suggests that a human being can have 5 to 7 (abstract or real) things on his mind simultaneously. The "things" (for example products) are represented by the 8 columns in DA'S VECTOR, for example 8 different products in a sales statistic. In this example the rows for each column contain the individual sales figures, perhaps monthly. The names of the months can then be the titles of the rows.

The tables can be imported and exported. A simple ASCII format (SFD) is used, which in a similar form is interpreted or used by virtually all database and mailmerge programs. It consists of a maximum of 33 text lines which in turn can contain up to 9 entries. All entries are marked by commas. In the first line are (optionally) the column titles, all further lines contain first (again optionally) the row title, followed by up to 8 numerical values (as many as there are columns), again comma–delimited. The decimal symbol must be a full stop. This format can be generated with many database systems and any text editor. Of course DA'S VECTOR also saves the tables created by the program in this format ("Export" in the chart editor).

It is important to understand the relationship between the column and row elements with the different presentation forms when you design your own charts. They are therefore explicitly mentioned in the description of the different forms. Before you can generate a presentation graphic you must therefore have some numerical material in the table; a special tool will then let you draw up a frame for the chart graphic in the work area. Either immediately or after a short calculation period a diagram appears in the mode selected. You can choose between 6 predefined forms of presentation and one with freely definable objects.

Before looking at these seven forms in more detail, we should take a brief look at a second settings dialog which is used to define the properties of the graphic. Here you can select if the x and y axes are to be drawn and in which colour and line thickness; the same options are available for the coordinate lines (coordinates grid); the unit markings on the axes can be enabled or disabled; the legends for the two axes (titles) can be enabled or disabled separately and the font, font colour and font size be selected (with some forms, the font size is limited). You can also determine the positioning of the titles and the unit markings and origin of the y axis. Some forms also give you a choice between the "Sum" and "Absolute" options; the effects of this are discussed in the descriptions of the different forms which we shall now turn to:



PARAMETER

	X	Y	COLOUR	LINEWEIGHT
MARK AXIS	■	■	■	0.20 mm
MARK LINE	■	■	■	0.20 mm
MARK CALIBRATION	■	■		POINT SIZE
INSCRIBE	■	■	■	5.00 mm

FONT Corona-Italic

Subdivision (y) max. 4 divid
Origin (y) 0.0000

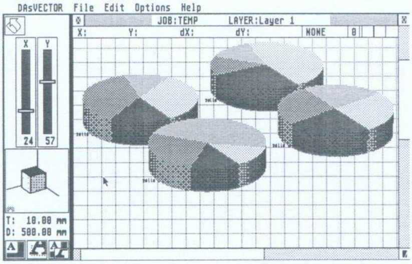
GRAPHS with lines: This is the simplest form of a diagram with lines; this simple form can be made more striking by choosing different colours. The values are always taken as absolute ones. The line thicknesses are preset but can be altered afterwards in the vector graphics editor and assigned a line pattern. You should generally remember that each presentation graphic is a very ordinary group object once generated which can then be further edited and modified using all the instruments available in DA'S VECTOR.

Graphs with filled areas: This does not draw a line but fills the entire area beneath the line. When "absolute" is selected the areas can, of course, cover each other; if you wish to prevent this select the "sum" option in which case the total of the values of all lines is taken as the backmost area (equivalent to the last column), the total of the first to the one before last for the area in front of it etc. The complete diagram thus shows the values only as area components. Of course, the whole thing will only become visible if you select (clearly) different colours (object colours) for the different areas. You can subsequently also assign colour fill patterns to the areas, for example.

Two–dimensional bar chart: probably the most frequently used standard form. The values in one column (absolute, no options) are drawn next to each other as a group of coloured bars; such a group is generated for each line and they are strung along the x axis with small gaps between them. The proportions of the bars are determined by the proportions of the entire frame and of the numerical values. Strong colour differences are advisable here, too. "Filling" these bars subsequently with halftone or colour pictures can produce interesting possibilities.

Three–dimensional bar chart: Here the individual bars are drawn as well as arranged three–dimensionally (on an area). The whole graph can be displayed in perspective using the 3D submenu, rotating it around the x and y axes by up to 90 degrees. You can also place a light source; the basic colours of the bars should not be too dark so that the shadow effect created by the light source can be effective. This form always takes absolute values, with no option.

Two–dimensional pie chart: A "pie" is drawn for each line. The differently coloured pie "slices" represent the relative sizes of the column values. There are, of course, no options here because pie charts cannot show absolute values but only relative ones. Hence they are popular for displaying percentage values. They reduce a message to the very simplest, which is why we recommend this form for use by politicians.



Three–dimensional pie charts: Here the pies are not drawn as "flat circles" but with a depth selectable in the 3D submenu and in perspective with selectable angles. Again you should use fairly bright colours since it is then possible to obtain impressive shadow effects with the freely positionable light source (for discerning politicians).

Freeform diagram: This is the most powerful tool in DA'S VECTOR'S chart editor. Its special appeal lies in the ability to build the presentation graphic using graphics objects defined by yourself. Depending on whether you have selected the "Sum" or the "Absolute" option, very different results are obtained which require different approaches. We are therefore treating them here as two different methods. But before going into the details, a few words on the special tool used to assign your own objects to the charts. After the initial calling up of the diagram montage you see a frame consisting of as many rows of frames as you have columns in the table. Each row (in the x direction) is equivalent to one column in the table. Each row can be assigned a graphics object using the special tool and this will be drawn immediately. To do this, you select the special tool, click first on a row and then on the graphics object. Of course you can replace objects by other ones at any time. When importing a path object into the presentation graphic, it will assume the colour set in the table. Group objects, however, retain their own colour attribute.

Freeform area charts: This form of chart can be obtained when you have enabled the "Sum" option. The graphic consists of groups of arbitrary graphics objects arranged one behind the other, the size of which is determined by the sum of the areas (in one column, from the first column to the last), as in the area graphs. These charts are also similar to pie charts in that they represent quantitative relations through area sizes, but there are no "standard units" here and the absolute sizes are preserved. This form is mainly suited for simple, symmetrical areas (usually simple paths). You will only obtain an exact, easy–to–visualise result with objects of the same area (in terms of the "bounding box" rather than the area that is actually drawn). As a general rule, the best results are produced with identical objects for all "rows". Try it out with simple shapes like squares, triangles, semi–circles etc. If you try this with any old object you will rarely get satisfactory results: the added–up areas of different, complicated shapes lose visual clarity and easy perceptibility, the main purpose of presentation graphics, all too easily. But nothing is impossible, so you are free to experiment with very complex objects. But the next form is better suited for that.

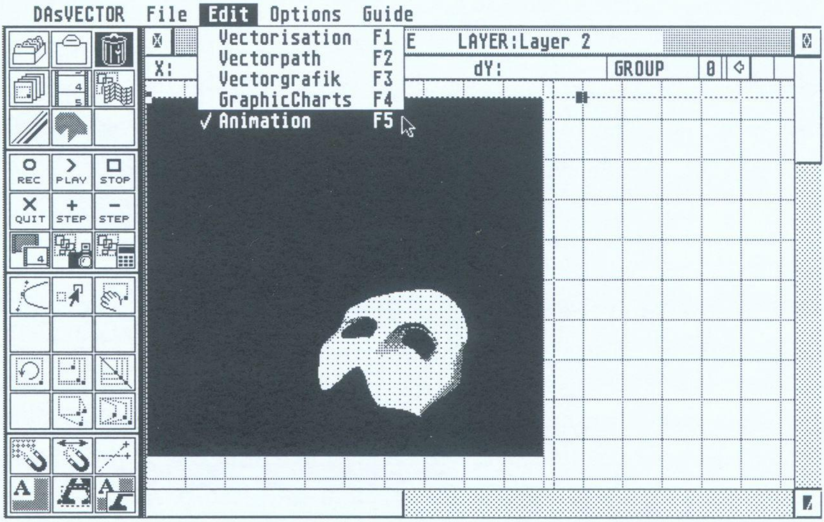
Free object diagrams: Select the "Absolute" option for this approach. Now the objects for the different groups (located next to each other) are drawn in absolute sizes in a kind of spatial perspective. Their relative sizes are defined by the proportions of the individual objects. The angle of the pseudo–perspective (no reduction with increasing distance) is predefined, but can subsequently be altered in the vector path editor by skewing or perspective distortion. The coordinates grid itself is not drawn in perspective but that can easily be changed afterwards. This form is suited for arbitrarily complex group objects, including different shapes in the different "rows". 3D objects from the extruder, complex colour gradients, vector text, objects with fill patterns can all be used here. Given the editing possibilities in the vector graphics editor there are no limits to your imagination in creating charts.

You have now encountered all the special tools except one simple one for moving a presentation graphic. The remaining tools (Rotate, free and proportional scaling) and submenus (Library, clipboard, trashcan) are exclusively for editing the "normal" vector graphics objects required for the freeform charts and are used as described in the chapter about the vector graphics editor.

For as long as you are in the chart editor, you only have a "prototype" version of your presentation graphic for experimenting with to your heart's content. It is only when you quit the charts editor that a complex vector graphic object in the currently selected form is actually generated, and you can then edit it further in the vector graphics editor if required. You can also, of course, place it in the trashcan – you cannot do this in the chart editor itself.

Graphics in Motion

The Animations Editor

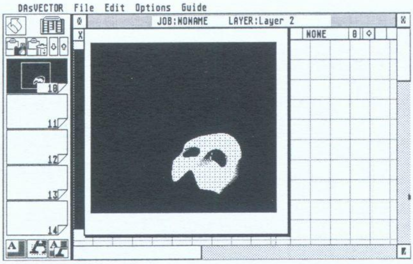


Computer animations created with enormous processing power are taking cinema audiences into surreal or deceptively lifelike worlds of illusion. DA'S VECTOR does not pretend to be able to compete with such productions. To make this clear from the outset: DA'S VECTOR is no 3D animation program. Instead, it takes its inspiration from the classical, two–dimensional cartoon film and it also offers a variety of ways to produce video titles. In this area, however, it breaks new ground because as a vector graphics program it can solve many a problem mathematically which, in a pixel–oriented animation program, would have to be solved tediously by hand. In addition, all the powerful tools of DA'S VECTOR which you have become acquainted with in the previous chapters can be used to generate such animations. The resolution independence of vector graphics and vector fonts, too, are an enormous advantage.

To begin with, DA'S VECTOR creates a "vector film"; this can then be rendered at any target resolution (width and height, colour depth) in the "recorder" and saved to the hard disk as a compressed animation file. The supplied animation player, which you are free to pass on to others, can play these films in "real time" (in the target resolution). The full screen format (for video recordings) or just a portion thereof can be used; in the latter case the films runs in a desktop window. The player program can be called by other applications and thus be seamlessly integrated into a multimedia environment.

DA'S VECTOR's animation editor offers so many new possibilities that it is difficult today to guess the full variety of applications: it is ideal for animating cartoons, it can generate animated charts, technical instruction films, slideshows, running titles etc. And, last but not least, in conjunction with a genlock interface, it provides a fantastic workshop for video titling and video tricks. Once you have understood the simple principle underlying the program, you will continue to discover new application possibilities. These are not only found in the field of animation; using the "Time–Space Converter" you can also use the animation editor as a generator for highly complex graphics objects.

"Keyframes" and "snapshots"



DA'S VECTOR can be used to generate film sequences of up to 10,000 single pictures. You begin with a blank ("unexposed") film. You do not need, however, to "expose" all pictures. Depending on the type of animation, a few "keyframes" are often sufficient. All intermediate pictures are calculated by the program through (linear) interpolation when a film is recorded or when in simulation mode. This involves transforming all point coordinates including Bezier control points (location and shape) as well as the object attributes (line thickness, line colour, area colour – some attribute changes also lead to abrupt changes, such as switching to transparency, switching line patterns or switching between fill patterns and fill colours).

Before looking at a simple example to illustrate how this works, we need to look at the "camera". It is a simple "snapshot camera" which records individual pictures and consists of nothing but a "viewfinder" and a "shutter". You already know the "viewfinder": it is the current working area, the top left corner of which as well as the height and width are selected in the "Page format" dialog. The absolute format is not so important here, compared to the proportions; the absolute "size" of a film picture is only determined during recording. But if the proportion does not agree with the target resolution, you may get irritating stripes at the edges of the pictures. It is advisable not to select too large a "viewfinder area" at the centre of the working area, because you will then have room for manoeuvre; the important thing is that objects filling the entire picture can still be moved outside the picture area in their entirety: in addition to the recording area, therefore, you should leave the same space again on either side, and the same applies vertically. In some special cases (for example long, running text columns for video titling) it can also be advisable to select a rather small recording area.

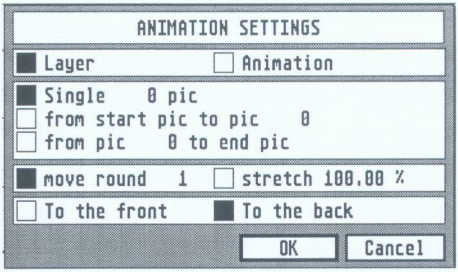
Now we need to find the "shutter". It is located in the "Film clipboard" submenu. This looks rather like the familiar "Clipboard" submenu, but in this case the individual entries are film pictures. Unlike with the ordinary clipboard, however, they are always present, so that you can click through all of the 10,000 (initially blank) pictures of the film or, alternatively, jump to any pictures via the keyboard. To record a picture, first select the target picture. If you now click on the copy icon in the top left, the "viewfinder picture" is recorded: a keyframe has been exposed. Before dealing with further options, now the promised simple example of a "keyframe animation":

Take any vector graphics object and position it in the bottom left corner of the "viewfinder area" or, if you wish, outside of the active work area. Select the film clipboard, select Picture 0, press "shutter". You are returned to the main menu and move the graphics object to the top right corner (perhaps outside the frame). Back to the film clipboard, jump to Picture 59 and "click!", that was it. You have generated your first film. Now click on PLAY in the main menu and the entire film is shown in slow motion (Don't worry, this is not the final speed!). You have generated an animation of 60 pictures in which the object moves from the bottom left to the top right. At a replay rate of 30 pictures/second this movement takes 2 seconds.

This simple example shows the basic principle of the animations editor very clearly: the various "keyframes" contain a "snapshot" each of the same basic object (a group object of any complexity) in various states. All pictures between the keyframes are calculated by linear interpolation of all point coordinates and an identical interpolation of the attributes (shape, line thickness etc). The object in our example could therefore also have changed its colour; all you would have had to do would have been to change its colour before copying it into the target picture (59). Of course you cannot produce a curved trajectory using this method with two keyframes; but for such purposes there are a number of tools which you will get to know. You cannot just generate simple motions; virtually all the tools of DA'S VECTOR are available to you as long as you do not change the basic structure of the object (number of paths) in the process; you can arbitrarily reshape and distort the object or parts thereof, even individual paths. But before we look at these tools, we must have another look at the "camera". Perhaps you have noticed that in the description so far we have spoken of an animated object. Can only one object be animated? (That would be rather boring). Yes and No! Here our layers come into the picture once again. You can indeed only animate one object for each layer, but we have thousands of them! In reality our "film strip" has a much more complex structure than we have revealed so far, since it can consist of any number of superimposed layers and each layer ("track") can contain a different object animation. The sequential order of the layers also governs how the objects cover each other. The different layers can contain a very different number of keyframes and are totally independent of each other. To cope with this multitude of possibilities, the "camera" has a few hidden functions which we are now going to describe. The simple snapshot camera will turn out to be a sophisticated, universal tool which can even function as a "projector".

The simple "snapshot" which we have encountered only makes one shot of the currently active layer, but if you press the SHIFT key when clicking, a picture is recorded for all layers – to put it in different words: a keyframe is generated for each track of the film. The keyframes are displayed in the clipboard items (Quickdraw mode). You can activate a colour preview with a doubleclick, just as in the clipboard and the library functions. This works not only with keyframes but with all "intermediate" pictures/frames. But that does not exhaust the possibilities of the "camera": it turns into a "projector" when you select a film picture (keyframe or other) and then press the copy icon with the CONTROL key down; the picture is now brought to the work area as an object and can be further edited there. This again only applies to the currently active layer, but if you press the SHIFT key in addition to CONTROL, all layers of the film picture are copied to the working area. Since you can do this also with (calculated) intermediate pictures, the animation can be gradually "refined". A further tool can delete keyframes from the active layer and, you may have guessed this, all layers with SHIFT pressed.

The Film clipboard submenu offers you a further powerful function to edit the timing of individual tracks (film layers) or of entire films. This enables you to stretch, compress or move single tracks, parts thereof or entire films and parts thereof IN TIME. Clicking on the topmost icon opens a dialog for this purpose in which you can make the relevant settings. You can, incidentally, also import or export individual tracks in addition to complete animations. This, in conjunction with the "time montage" functions, thus provides you with a digital editing suite to combine many different individual sequences.



ANIMATION SETTINGS

■ Layer □ Animation

■ Single 0 pic
□ from start pic to pic 0
□ from pic 0 to end pic

■ move round 1 □ stretch 100.00 %

□ To the front ■ To the back

Transformations

The basic concept of animation in DA'S VECTOR you now know: within one track one object – which can be a group of any level of complexity – can exist in any number of keyframes in various stages of transformation; the program then interpolates the "empty frames". Any transformation which does not destroy the "physical structure" of the object is allowed; the "physical structure" being the number and grouping of the paths. The "geometrical" (location, shape) and "qualitative" (line thickness, line pattern, colour, fill etc) structures, however, can be changed at will. The tools and functions of the animation editor only allows such "legal" modifications. But we cannot, of course, prevent you from leaving the animation editor and editing an object in other program parts before copying it back into a keyframe; it is your responsibility to effect only "legal" changes: never add any paths or delete them; groups should only be ungrouped temporarily, since an erroneous regrouping changes the "physical" structure and renders the object unusable.

The simplest transformation, of course, is moving an object. You have the normal tool to move it. You also have all the scaling tools from the vector graphics editor at your disposal: free and proportional scaling, skewing, distortion and rotation. Please remember with rotation that a simple linear interpolation of the two rotated states does not result in a rotating motion! For this, you need step–by–step refinement or the multicopy function (see below). A further tool is already familiar from the path editor: you can move and manipulate individual paths and path segments (even inside groups!). A further tool allows you to enable or disable automatic tangent calculation. With all these editing tools it is useful to have a control function which you can enable as an option: this displays the original state of the object as a line structure and you can check your alterations against this.

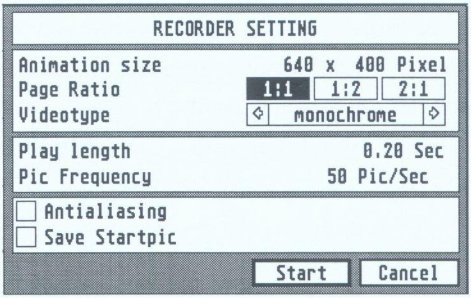
To change the attributes you have the usual submenus for line and area settings. All alterations are allowed, but there are changes in attributes which do not allow interpolation and can therefore result in sudden changes. These including transparent fills, fill pattern changes, line pattern changes, line ends, line corners. Colour changes are particularly effective but require a lot of disk space in an animation. Any type of colour gradient can be generated through a combination of changes in colour, shape and size using the "Time–Space Converter".

The Bezier grid transformation in the submenu already known to you gives you a powerful tool for animations with flexible reshaping. Similar transformations, but also, for example, complex trajectories with iterative methods, are made possible with the calculator also already introduced. The multicopy function has a different meaning here from the vector graphics editor. There it generates further objects but this is not permitted within one track. Instead, you can generate keyframes with it: all copies are stored as keyframes one after the other, beginning with the currently selected film picture. You can then subsequently stretch out this sequence of keyframes, and in this way animate rotations, for example, accurately.

The familiar Layer menu is needed in particular to edit the different "tracks". You can freely exchange objects using the library and clipboard. You must ensure yourself that "alien" objects do not intrude on a track; the program will of course complain but perhaps something totally unexpected will happen.

Simulator and Recorder

Every now and then, of course, you will want to check how the motion sequences and transformations which you have designed for your animations turn out. This is what the "simulator" is there for; it generates the entire film or parts thereof in the working window, but of course not in "real time" but by rendering picture after picture as vector graphics in the current screen resolution. You can stop the sequence, jump backwards or forwards onto single pictures or restart anywhere. Note: The simulator can be left by clicking on "Quit".



RECORDER SETTING

Animation size 640 x 400 Pixel
Page Ratio [✓ 1:1] 1:2 2:1
Videotype ◁ monochrome ▷

Play length 0.20 Sec
Pic Frequency 50 Pic/Sec

□ Antialiasing
□ Save Startpic

Alongside the five "buttons" to control the simulator you find the recorder button, which reveals an extremely powerful program part that actually generates the computer films proper playable in "real time" with the "Player". To do this, the single frames of the film must first be generated in the target resolution (width and height), then rendered in one of nine different colour resolutions and finally be compressed into an animation that can be played in real time. Such an animation can be extremely large, dependent on the picture size, colour depth and the degree to which the pictures are manipulated, certainly larger than the computer's RAM. This is why the Player can play them from virtual memory, constantly loading in sections from the hard disk. The amount of data to be transferred is of crucial importance in this, and to create a seamless animation, the data must be greatly compressed and then again be decompressed and displayed in "real time". A fast computer and a fast hard drive are required, especially with high colour resolutions and large film formats, but even then not everything expected is always possible. In critical cases, slight corrections to the timings can often be helpful. As long as you keep to the technically feasible, you can create amazing video films with DA'S VECTOR's recorder. The high compression rate (from 10 to 30:1 on average) and the virtual memory replay permit film sequences of several minutes length. Using additional software, these sequences can be linked together, mixed with slideshows and given a "soundtrack".

To conclude this chapter, a brief discussion of the recorder settings. Here you enter the size of the animation window in pixels. It should, of course, have the same size as your "viewfinder"; the program may correct the width as this cannot be set arbitrarily depending on the colour depth. With monochrome animations, the width must be a multiple of 32. Films can be generated for screen resolutions with a normal 1:1 pixel ratio, but also for 1:2 (for example, 3208480) or 2:1(640*200). Nine different colour resolutions are possible:

1 bit monochrome

4 bit planes 16 colours

4 bit packed 16 colours

8 bit planes 256 colours

8 bit packed 265 colours

15 bit ET4000 32767 colours

16 bit Falcon 65536 colours

24 bit low 16.7 million colours

24 bit high 16.7 million colours

This covers the colour resolutions of all Atari computers (except the 4–colour resolutions) and all graphics cards known to us at present. You therefore have to select a target resolution when recording a film. A resolution–independent recording which would only be adapted for the monitor on display would, of course, be nice, but would let "movies" come to a standstill at today's processor performances. Nothing can be done about this (as yet), but this is where one of the big advantages of DA'S

VECTOR's vector animation technique shows itself. The actual animation is generated and saved in vector format, so that from such an animation, which also has the advantage of using very little storage space, you can generate new films for any target resolution at any time.

Two further settings concern the timing controls. They are a little bit difficult. Firstly, you enter the frame frequency; this means the "animation frequency", i.e. the number of pictures generated by the recorder for one second's play. It should be equivalent to the screen redraw frequency of the target device, i.e. be identical or (normally) a multiple integer of it. Typical frame frequencies are between 10 and 30 pictures/second. The second timing control concerns the absolute play length in seconds. This allows you to adapt the timing to the technical realities or to generate extreme slow motions and speedups.

Two further options can be enabled or disabled. With "Anti–aliasing" you can smoothen the edges of text and vector graphics. This will be necessary especially with smaller output sizes, but also depends on the contents. The anti–aliasing requires a lot of processing time and the animations will also require a little more disk space.

The "Save start picture" option should always be enabled as a general rule. Disabling it only makes sense in certain special cases, for example when animations are to be chained together and you know for sure that the end picture of one and the start picture of the other are identical. It may then, under certain conditions, be possible to save some disk space by doing without individual start pictures. Only for those who know what they are doing!

In this context we might as well discuss the animation editor's file system: the recorder always creates three files – the time track (a sort of script file) with the extender MOV, the start picture with the extender

MPC and the animation file with the extender MAM. The player requires all three files in its search path.

When the actual recording begins, DA'S VECTOR first saves its current status into a job file and then passes all its available memory to the greedy recorder. When finished, or if aborted with UNDO, the original state is restored again. You will not normally get to see the job file (SWAPFILE.HJT) as it is immdiately deleted, but should the recorder ever crash (which we hope won't happen) then you can always restore the status quo by loading this file.

Services and Knowhow

The program comes with a some useful accessories. Some of these can be used without DA'S VECTOR, but their cooperation with DA'S VECTOR is especially easy. So far we have not mentioned the purpose of the "Services" menu item in the "File" dropdown menu. If you click on this item you see what services are offered by way of external programs. Clicking on one of these, DA'S VECTOR will be quit (temporarily) but not without saving its current status in a job file (the same SWAPFILE. HTJ). Control is passed to the program called and you can take advantage of the required "services". Once finished, you will be returned to DA'S VECTOR and find your original working environment restored. "Services" can only offer programs which follow a certain protocol. At the time of writing, these are only the bundled utilities, but this is certain to change in future. Why shouldn't you be able to call up your scanner software this way? It would save a lot of mouseclicks and mouse movement.

Without aiming to be exhaustive and taking into account future changes we want to briefly characterise the purpose of these programs. Further details can be found in the electronic reference manual which always reflects the latest situation.

The graphics converter allows you to generate fill patterns with almost every program; you will always be able to convert one of the standard formats. The font converter is for converting standard PostScript Type 1 fonts into the internally used DFN format. The gradient generator lets you generate various kinds of halftone and colour gradients as pixel graphics; these have the big advantage of being capable of having "noise" applied to them which eliminates the visible steps that can never be completely eradicated from vector gradients. These gradients can also be used as fill patterns, the "noise" softening the unnatural appearance of vector graphics. All the bundled programs are part of the product and must not be copied and distributed, like the main program.

This does not apply, however, to one program: the animation player. You are free to copy and distribute it: after all, you do not necessarily want to keep your films to yourself and who could do anything with them without the Player?

DA'S KNOWHOW

DA'S KNOWHOW occupies a special position among the accessories. Without this program you cannot read either the reference manual or the tutorial. You can start DA'S KNOWHOW as a program and simply use it to study the documentation. If you want to use it "online" while working with DA'S VECTOR, you need to install it as an accessory (unless you are already using a multitasking system, in which case it could also run as a parallel application). DA'S KNOWHOW can handle several manuals simultaneously so that you can use several manuals at the same time, even for different programs. Even cross references between different documents are possible.

Documentation managed by DA'S KNOWHOW has several advantages over a printed manual:

The multi–dimensional cross referencing system supports context–sensitive learning in a unique way; a single mouseclick gets you from chapter to chapter; in addition to the usual references you always have a context list available which can give you a quick overview over where else you can find something about the current topic.

Reference manuals can be integrated into the main application(s) as online help systems. In DA'S VECTOR, all you need to do is to point at an icon with the mouse and press the HELP key, and the relevant chapter of the manual will appear. From there, you can then explore the further context.

The documentation is available as a file and can always be maintained up– to–date with short–term alterations, upgrades and so on. This is difficult and time–consuming with printed manuals.

Not least, it is easy on the environment and your purse; with five pounds of documentation in the baggage we would have to charge you a lot more for this fantastic program. It is not bulkiness that counts, but DA'S KNOWHOW.