

ACORNSOFT

A database manager
for the BBC Microcomputer

User Guide

VIEWSTORE

ViewStore

A database manager
for the BBC Microcomputer

User Guide

ACORNSOFT

This user guide was written by Information Transfer Limited using the VIEW word processor.

Copyright , Acornsoft Limited 1985

All rights reserved

First published in 1985 by Acornsoft Limited

British Broadcasting Corporation has been abbreviated to BBC In this publication.

No part of this book may be reproduced by any means without the prior consent of the copyright holder. The only exceptions are as provided for by the Copyright (photocopying) Act or for the purpose of review or in order for the software herein to be entered into a computer for the sole use of the owner of this book.

FIRST EDITION

Printed in the United Kingdom

ISBN 0 907876 45 5

Published by Acornsoft Limited, Betjeman House , 104 Hills Road,
Cambridge CB2 1 LQ, England. Telephone (0223) 316039

Contents

Introduction		1
Using this manual		1
Your DFS		2
Some definitions		2
1 Getting started		5
Installing the ROM		5
Copying the ViewStore disc		5
Using the function key card		5
Using the reference card		5
Conventions used in this manual		6
2 Using ViewStore		7
Entering ViewStore	7	7
Leaving ViewStore		8
Loading a database File		8
Examining a database file		10
Another way of looking at data		15
Examining the record format		17
Examining the database header		20
Alternative formats		21
Changing the index		23
Wildcards		23
Locating a record in a database File		24
Editing data		25
Adding new data		26
Making more space available		27
Finishing a ViewStore session		27
3 Extracting information from a database		29
Preparing for printing		29
Using utilities		31
Selecting and sorting records	32	32
VIEW macro parameters		39
ViewSheet linking files		41

Labels		42
Reports		45
Registers		51
Import		57
Using SPOOL with utilities		63
Passing text into View		64
Using exec files with utilities	65	
4 Creating a database		68
Planning your database		68
Defining a new database		73
Entering data		80
Preserving your data		81
5 Altering a database		82
Changing the card layout		82
Using different formats		84
Rebuilding a database		85
Creating a new index file		87
6 Filing systems		89
CFS		89
DFS		89
Hierarchical filing systems;ADFS, NFS		89
Filenames		90
Use of directories in Viewstore		91
Prefixes		92
Running out of room		93
7 Error messages		95
8 Notes		103
Screen colours		103
*FX Calls		104
Faster data entry		105
BREAK		106
Extra memory		107
Exec files	107	
The View range		108

9 Specification	110
Summary	110
Specification	110
Viewstore file format	113
Index	116

Introduction

ViewStore is a sophisticated database manager for use with the BBC Microcomputer.

ViewStore enables a large quantity of data, far larger than can be accommodated in the computer's memory, to be stored in a single disc file. This information can be displayed in virtually any order, and the system can search rapidly for an individual item. The information can be selected, sorted and summarised in reports designed by the user and it can be printed out as labels. Information can be passed to Acornsoft's VIEW word processor, and summaries of numerical information can be passed to the ViewSheet spreadsheet program.

ViewStore can be used for invoice registers, mailing lists, fixed asset registers, personnel records, appointment diaries; in fact, for any application where large amounts of data need to be stored simply and accessed rapidly.

Using this manual

This manual has nine chapters.

Getting started deals with the installation of ViewStore, and the things you should do first before starting to use it.

Using ViewStore introduces you to the ways in which a database can be displayed, manipulated and added to. It refers to an example database, called Cars, which is provided with the system.

ViewStore Extracting information from a database explains the facilities that ViewStore provides for extracting and reporting on the information in a database, including passing information to VIEW or ViewSheet.

The next two chapters are intended for those who want to use ViewStore to build databases of their own.

Creating a database takes you through the steps involved in building a database of your own from scratch

Altering a database shows you how your database can be adapted after it has been built.

Filing systems points out the ways in which your filing system affects your use of ViewStore.

Error messages lists the ViewStore error messages and gives a brief

explanation of each, suggesting remedial action.

Notes details facilities of the BBC Microcomputer which you may find useful

Specification provides details of the internal format of ViewStore files.

Each chapter contains many suggestions for things to try out, often using the example database. We strongly suggest that you read this manual whilst sitting at the keyboard of your microcomputer, and that you try each of the suggestions as you come to them.

Your DFS

ViewStore can be used with a BBC Microcomputer that is connected to a floppy disc drive, a Winchester disc drive or the Acorn Econet; it cannot be used with cassettes.

The storage of information on a disc, and retrieval of the information from the disc, is controlled by a program known as the disc filing system which forms part of your microcomputer's permanent memory.

You are most likely to be using a floppy disc drive with the DFS. From now on, this manual assumes that you are. If you are not using the DFS, you will need to refer to Chapter 6 Filing systems.

Filenames, including the directories and drives to which files belong, are very important in ViewStore. ViewStore keeps different types of file in different directories as follows:

- Databases are in directory D.
- Format files are in directory F.
- Indexes are in directory I.
- Utilities are in directory U
- Sort files are in directory S.
- Report files are in directory R.

If you are unsure about directories you might find it helpful to spend some time reading about them in your filing system user guide before reading further in this manual. When you start to build databases of your own, this knowledge will be essential.

Some definitions

The rest of this introduction explains some of the special terms used in connection with a database. Those familiar with using databases can go straight on to chapter 1.

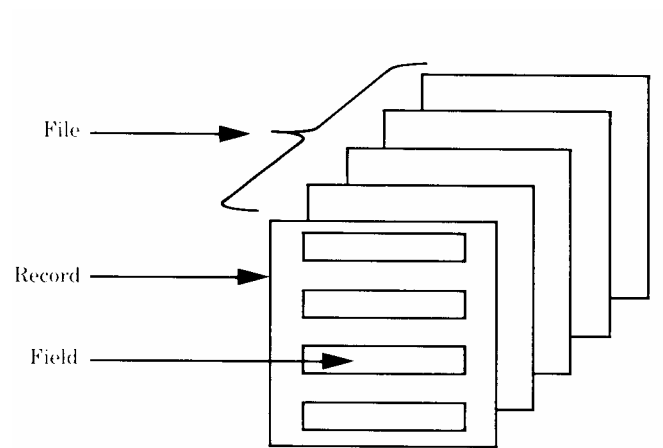
In a ViewStore database, the space that the computer makes available for each

item of information is described as a field. Each field holds a particular type of information; this can be an address, a telephone number, a date of birth, a quantity, a value, or anything else which can be thought of as a single unit of information.

Within the database, a group of different fields will be related in some way, for example because they each contain information about the same person, or the same piece of equipment, or the same invoice. A group of related fields is described as a record. A record can be thought of as a blank form containing areas (that is, fields) where information can be written.

A collection of similar records is described as a file. Each record in a file must have the same fields. However, it is not necessary for every field of every record to contain information.

It can be helpful to envisage a database like this:



Intentionally blank

1 Getting started

Your ViewStore package comprises :

- The ViewStore ROM
- The ViewStore User Guide.
- The ViewStore disc containing the ViewStore utility programs and the files for the example database.
- The ViewStore function key card.
- The ViewStore reference card.
- A ROM installation leaflet.
- A disc configuration leaflet.

Installing the ROM

Follow the instructions in the ROM installation leaflet.

Copying the ViewStore disc

Before starting to use ViewStore, you will need to take at least one copy of your ViewStore disc and then to put the original in a very safe place. Use a copy or working disc in all subsequent operations. Then if any of the files are inadvertently deleted or corrupted you can retrieve them from the original.

The way in which you copy your ViewStore disc will depend upon your disc configuration. This is described in detail in the disc configuration leaflet enclosed in this package.

Remember to label the working discs carefully, showing that they contain ViewStore files.

Using the function key card

The function keys are the red keys at the top of the keyboard. The function key card is a rectangular cardboard strip which shows you the special commands that certain function keys invoke when using ViewStore. You should slide the card under the clear plastic strip just above the keyboard so that DATA lines up with the key marked f0.

Using the reference card

Once familiar with ViewStore you will find almost all the information that you need on the reference card.

Conventions used in this manual

In this manual, the commands shown on the function key card are referred to simply by their names; for example, DELETE RECORD. Consult the card to see which key or keys must be pressed to invoke this command. In the case of DELETE RECORD, for example, hold down SHIFT and press f9.

Commands which you type yourself are shown in computer typeface, for example

PREFIX

Text which you type yourself or which you see on the screen is also shown in computer typeface, for example

PREFIX U:1. **RETURN**

In the example above **RETURN** is shown in bold type to indicate that you should press the key marked **RETURN**. **DELETE** would mean that you should press the key marked **DELETE**, and so on.

CTRL → means that you should press the key marked **CTRL** whilst, at the same time, pressing and releasing the right arrow key.

If you see a word in computer typeface italics the word is not to be typed, type what the word represents. In the example below literally.

SF *filename* **RETURN**

you would not type the word filename but the name of your file.

You are now ready to Start using ViewStore.

2 Using ViewStore

Entering ViewStore

The message you see when you first switch on your computer depends on the way in which your ViewStore ROM has been installed. If you see a message like this:



you are already in ViewStore. If you see any other message you must first enter ViewStore by typing:

***STORE RETURN**

That is, type ***STORE** and press the **RETURN** key.

The message at the top left hand corner of the screen confirms that you are in ViewStore and shows you which screen mode you are currently using.

E d i t i n g N o F i l e means that there is no data in the computer's memory; Bytes free tells you how much space there is available in the memory . The arrowhead => below the message is the prompt which tells you that ViewStore is waiting for you to give it a command.

You might like to investigate the effect of each of the different screen modes, 0 to 7, on the memory space available for data. To change screen mode type **MODE number RETURN**, for example **MODE 3 RETURN**. You will see that the smallest space is provided by screen modes 0 , 1 and 2. Nevertheless screen mode 0 is recommended for using ViewStore because, as you will see, it allows the maximum amount of data to be displayed on the screen at one time. The efficient way in which ViewStore makes use of memory means that, even in screen mode 0, memory constraints are not usually a problem. If you have shadow memory or a 6502 Second Processor you will have a lot of memory in all screen modes.

If you are using ViewStore with a television rather than a monitor you may find screen mode 0 rather hard to read. In this case try screen mode 4. If the screen is flickering type :

***TV0,I RETURN**

before changing the screen mode.

Leaving ViewStore

To transfer from ViewStore to BASIC, type:

***BASIC RETURN**

If you have other ROM programs installed in your computer you can transfer to them in a similar way. For example, to transfer to VIEW type:

***WORD RETURN**

and to transfer to ViewSheet type:

***SHEET RETURN**

Loading a database file

The rest of this chapter shows the facilities that ViewStore provides for examining a database, using as an example the database called Cars provided on your ViewStore disc.

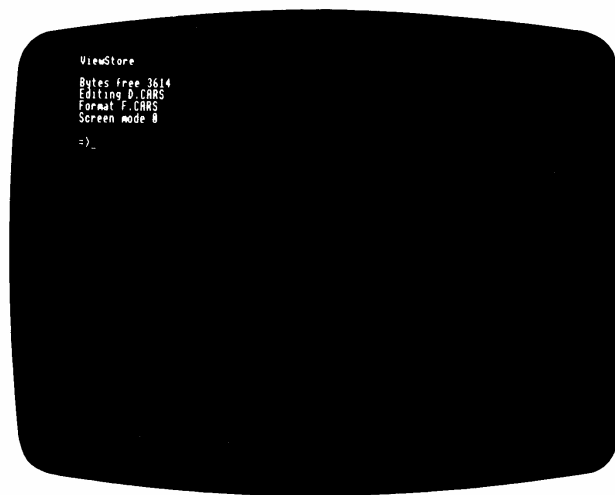
To load the database, first place the working disc that you copied from the original into your disc drive.

Then type :

LOAD CARS RETURN

Most ViewStore commands can be abbreviated; for example, this command could alternatively have been typed as **L CARS RETURN**. The valid abbreviation for each command is shown on your reference card. Commands can be typed in either upper case or lower case or a combination of the two.

Almost immediately, the message in the top left-hand corner of the screen changes, to something like this:



The message shows you that you are now editing the file DCARS. Editing here includes examining; you can look at the file, and make changes to it if you wish.

Note that ViewStore has loaded the file D.CARS, although you only gave it the name CARS. ViewStore assumes that all database files are in directory D. There is an additional line in the message: Format F.CARS. The format file contains details of how the data is to be displayed on the screen, and a format

file must always be loaded with each database file. Unless you tell it otherwise, ViewStore assumes that:

- The format file has the same name as the database file.
- The format file is in directory F.

Finally, you may have noticed that the `Bytes free` figure is now smaller. This is because some of the computer's memory is taken up by the format file.

Examining a database file

You are currently in ViewStore's command screen where you give commands such as `LOAD` after the prompt `= >` To examine the database you must switch to the data screen.

ESCAPE is used to switch between the command screen and the data screen. When working in either screen, **ESCAPE** can be used to end whatever you are currently doing. **ESCAPE** will always return you eventually to the command screen.

When you enter the data screen you first see the titles that have been set up for the database. and then the message `Reading` as ViewStore reads the first screenful of records from the disc. Each dot after the message represents a record which has been read.

When a screenful of records has been read in, ViewStore displays it; you should see something like this:

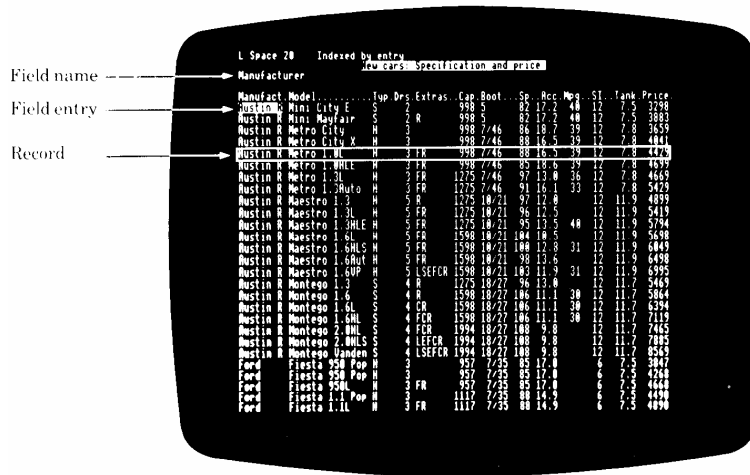
```

L Space 28 Indexed by entry
Manufacturer New cars: Specification and price

```

Manufact.	Model	Typ.	Drs.	Extras.	Cap.	Boot	Sp.	Acc.	Mpg.	SI	Tank.	Price
Austin	Mini City E	S	2		998	5	82	17.2	40	12	7.5	3298
Austin	Mini Mayfair	S	2	R	998	5	82	17.2	40	12	7.5	3883
Austin	Metro City	H	3		998	7/46	86	18.7	39	12	7.8	3659
Austin	Metro City X	H	3		998	7/46	88	18.5	39	12	7.8	4841
Austin	Metro 1.8L	H	3	FR	998	7/46	88	18.5	39	12	7.8	4429
Austin	Metro 1.8HLE	H	3	FR	998	7/46	85	18.6	39	12	7.8	4699
Austin	Metro 1.3L	H	3	FR	1275	7/46	97	13.0	36	12	7.8	4669
Austin	Metro 1.3Auto	H	3	FR	1275	7/46	91	16.1	33	12	7.8	5429
Austin	Maestro 1.3	H	3	R	1275	10/21	97	12.0		12	11.9	4899
Austin	Maestro 1.3L	H	3	FR	1275	10/21	96	12.5		12	11.9	5419
Austin	Maestro 1.3HLE	H	3	FR	1275	10/21	95	13.5	40	12	11.9	5794
Austin	Maestro 1.6L	H	3	FR	1598	10/21	104	10.5		12	11.9	5698
Austin	Maestro 1.6HLS	H	3	FR	1598	10/21	100	12.8	31	12	11.9	6049
Austin	Maestro 1.6Aut	H	3	FR	1598	10/21	98	13.6		12	11.9	6498
Austin	Maestro 1.6UP	H	3	LSEFCR	1598	10/21	103	11.9	31	12	11.9	6995
Austin	Montego 1.3	S	4	R	1275	18/27	96	13.0		12	11.7	5469
Austin	Montego 1.6	S	4	R	1598	18/27	106	11.1	30	12	11.7	5864
Austin	Montego 1.6L	S	4	CR	1598	18/27	106	11.1	30	12	11.7	6394
Austin	Montego 1.6HL	S	4	FCR	1598	18/27	106	11.1	30	12	11.7	7119
Austin	Montego 2.0HLS	S	4	FCR	1994	18/27	108	9.8		12	11.7	7465
Austin	Montego 2.0HLS	S	4	LEFCR	1994	18/27	108	9.8		12	11.7	7885
Austin	Montego Vanden	S	4	LSEFCR	1994	18/27	108	9.8		12	11.7	8569
Ford	Fiesta 950 Pop	H	3		957	7/35	85	17.0		6	7.5	3847
Ford	Fiesta 950 Pop	H	3		957	7/35	85	17.0		6	7.5	4268
Ford	Fiesta 950L	H	3	FR	957	7/35	85	17.0		6	7.5	4668
Ford	Fiesta 1.1 Pop	H	3		1117	7/35	88	14.9		6	7.5	4498
Ford	Fiesta 1.1L	H	3	FR	1117	7/35	88	14.9		6	7.5	4898

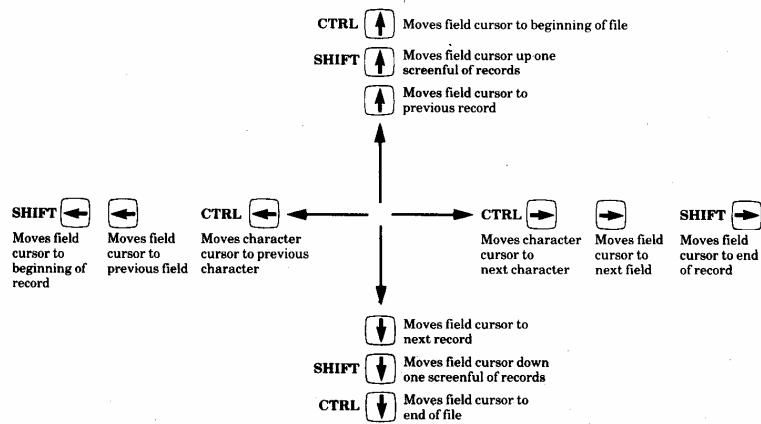
If you have experience of a spreadsheet program (such as Acornsoft's ViewSheet) this display will be familiar to you; for this reason it is referred to as the spreadsheet display. You can think of the screen as a window looking on to the database which you can move around to examine different parts. Each line on the screen displays a single record in the database; each box represents a single field within a record. The titles above the data are the field names.



In the spreadsheet display there are two types of cursor

The field cursor

This is shown as a block of reverse video and when you first switch to the spreadsheet display. It will be in the top left hand corner of the database. You move the field cursor using the arrow keys as shown in the diagram below. Practise moving the field cursor around the database. Notice that if you hold the arrow keys down the field cursor keeps moving until you release the key. This is a facility known as auto repeat which applies to most of the keys on the keyboard.



The character cursor

This is shown as a flashing underline character and shows you where the next character you type on the keyboard will appear on the screen. It can be moved left or right by pressing **CTRL ←** or **CTRL →**. The function key commands DELETE END OF FIELD, BEGINNING OF FIELD, END OF FIELD, INSERT CHARACTER and DELETE CHARACTER are also used to move the character cursor.

Here are some points that you should particularly notice:

1. Certain fields appear to contain abbreviations. For example, the top left-hand field in which the field cursor is originally placed reads **A u s t i n R**. →Move the character cursor to the right by pressing **CTRL →**. You will see that, when the character cursor reaches the end of the field, the text scrolls to the left to reveal that the full field entry is **A u s t i n R o v e r**. As you see, ViewStore does not restrict the size of an entry in a field to the number of characters that can be shown on the screen. A quicker way to display the remainder of an entry that is wider than the field can be to press **END OF FIELD**

2. **RETURN** moves the field cursor to the next field There may be a short pause whilst ViewStore accesses the disc.

3 When you move the field cursor a message appears on the third line of the screen This is either information about the field or the field name if no information has been given.

4 If you press **RETURN** or → when the field cursor is in the field at the right-hand side of the screen, the screen moves across the database to the right to display the next field. Eventually you will see that the fields no longer have names but numbers; these are spare fields for you to extend the database if you wish Pressing **RETURN** with the field cursor in the last field of a record moves the field cursor to the first field of the next record.

5. When you press ↑ or ↓ , the field cursor moves vertically up and down the screen In most cases this will be what you want, particularly when examining data. However, there may be occasions when you want the field cursor to move to the first field of the next or previous record, for example when entering data In this case, press **CURSOR LOCK** If you want to change back to locked movement press **CURSOR LOCK** again Note that the letter **L** is shown at the top left-hand corner of the screen when the field cursor is locked to movement in the same field, and **U** when it is unlocked.

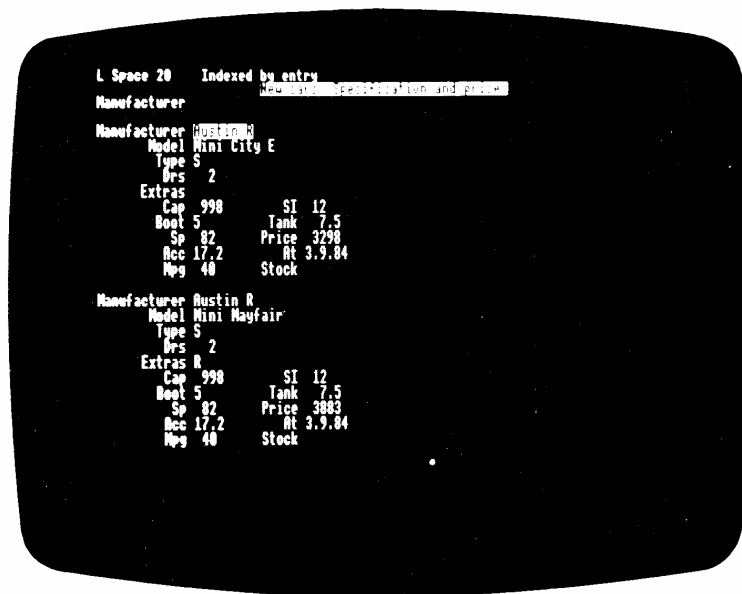
6 Pressing **SHIFT ↓** makes the field cursor move down the database by a screenful of records **SHIFT ↑** moves up the database in a similar way There is a short wait whilst ViewStore reads in the next screenful of records: the **Reading** message and the dots confirm that this is happening.

7 Pressing **CTRL ↓** takes you to the last record in the database. ViewStore displays a single record which is the last record in the database The field cursor is positioned on a blank line after the last record to enable you to enter another record if you wish. You can move back to earlier records using ↑ and **SHIFT ↑** in the normal way. However if you press v whilst on the final blank line (or ↑ whilst at the beginning of the line) ViewStore will display the **E n d** message

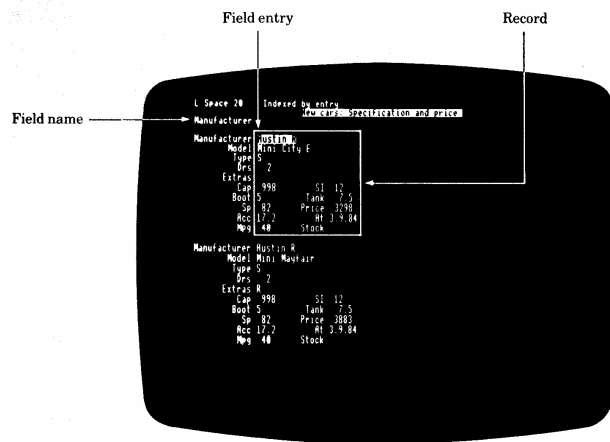
Another way of looking at data

The spreadsheet display that you have been looking at so far enables the maximum number of records to be displayed on the screen at one time. However, depending on the size of each record it may not be possible to see all the fields at the same time, and the field names may not fit into the space provided for the fields.

To overcome these problems ViewStore provides an alternative to the spreadsheet display in which each record appears as if its information had been written out on a reference card. To switch to the card display, press **CHANGE DISPLAY**. Your screen should look like this:



In the card display every field of a record is displayed on the screen with its full name; see for example **Manufacturer** which was shown as **Manufact** in the spreadsheet display. The space allowed for the field entry itself is the same as in the spreadsheet display. ViewStore fits as many complete records on to the screen as it can.



You move the cursors in the card display using exactly the same keys as for the spreadsheet display. As you will see later, you can also edit and enter data in exactly the same way.

Note that **RETURN** and \rightarrow move to the next field in the order shown in the spreadsheet display. Depending on how the card display has been organised this may mean that the field cursor jumps between fields in an apparently illogical manner. Pressing **RETURN** in the last field of a record in the card display moves you to the first field of the next record.

The arrow keys work in the same way in the card and spreadsheet displays. However, because the records and fields are displayed differently the field cursor does not follow the direction of the arrows So \rightarrow and **RETURN** move to the next field which may not be to the right \downarrow moves to the next record which may be several lines or a whole screen down.

Sharp-eyed readers may have noticed that an extra field name **S t o c k** appears in the card display, but that it is not possible to move the field cursor into this field This is explained later.

To switch back to the spreadsheet display, just press **CHANGE DISPLAY** again.

Examining the record format

The next step is to examine the contents of the format file, the file that determines the way in which the data you have just been looking at is displayed.

The format file contains three types of information:

- Record format. This describes the structure of each record in the database.
- Database header. This information relates to the database as a whole.
- Card layout This describes the layout used in the card display.

To examine the record format, press RECORD FORMAT. Your screen should look something like this:

```
L Space 1009
Record format
Field name
Field name.....Mid.T.S.D.Low limit..High limit..I.Key Index name.....Prom
Manufacturer 8 a y R 2 MANUF
Model 14 a y R 5 MODEL
Type 3 n
Drs 6 n 0 S-Sa
Extras 4 n 500 4000 Num
Cap 4 n 500 4000 B-Re
Boot 3 n 0 Ensj
Sp 4 n 0 1 y 3 SPEED Boot
Rcc 4 n 0 1 y 3 Mpg Top
Mug 3 n 0 1 y 3 Mpg Fuel
SI 3 n 0 1 y 3 Mpg Serv
Tank 3 n 0 1 y 3 Mpg Fuel
Price 8 n 0 1 y 5 Price List
Rt 8 n 0 1 y 5 Price Date
Stock 8 n 0 1 y 5 Price Quan
```

This information is presented in the spreadsheet display. Each line of the record format contains the information for one field of the main database. You can move around the record format using the arrow keys in the normal way.

Briefly, this is the meaning of each of the items of.
Information held for each field. A fuller description is given in chapter 4
Creating a database.

Field name

The name to be displayed at the top of the field.

Wid(th)

The number of characters of the field entry to be displayed.

T(ype)

The type of data to be entered in the field

N Numbers, for example the field Drs.

D English-style dates in the form Day Month Year, for example 31/12/85.

M American-style dates in the form Month Day Year, for example 12/31/86.

A Alphanumerics: letters, numbers and punctuation characters.

T Text: letters, numbers and punctuation characters.

T is used for words and sentences; A is used for names

S(croll)

Contains Y if the field entry is allowed to scroll across its column; this means that the entry can be longer than the display width. N specifies that the field entry does not scroll. The default settings (those which VewStore gives you if you make no entry) are Y for alphanumeric, textual and date entries and N for numbers.

D(ecimal places)

The number of decimal places to which numbers are displayed in this field; only applies if the field type is N.

Low limit

The lowest value that can be accepted as an entry in this field (eg in the field Capacity, capacities below 500cc are not allowed).

High limit

The highest value that can be accepted as an entry in this field (in Capacity, capacities above 4000cc are not allowed)

I(ndex)

Contains Y or R if there will be an index for this field. An index enables the database to be displayed in the order of the entries in this field; Y and R refer to different types of index, as explained later.

Key (width)

The number of characters to be used to distinguish between field entries either in the index file or for sorting; this is also explained later.

Index name

If there is an index for this field (see **Index** above), contains the name to be given to the index file.

Prompt

The message displayed on line 3 of the screen when the field cursor enters this field; if left blank, the field name is displayed.

Value list

A list of the valid entries in the field.

Note that, in some cases, the prompt shown on line 3 in the record format includes a list of the valid entries in the column: for example, N , Y , R for **Index**. In each of these cases the first item in the list is the default. Thus if **Index** is left blank ViewStore will assume that entries in that field are not to be indexed. This convention applies to the prompts in the database header and the record format.

The record format information helps to explain the mysterious appearance of **Stock** in the card display. As you can see, the field **Stock** has been given a field width of zero. It can still contain data, but in the card display the field name alone appears, and in the spreadsheet display nothing at all is shown. Setting field width to zero is a good way of hiding data. This is useful if you want to present two or more displays of the same data using different format files.

To investigate the effect of the record format on the database, alter some information (particularly in **Field name**, **Width** or **Prompt**) and then return to the database by pressing **DATA** to see how the display has changed To alter record format information, position the field cursor in the appropriate field and then:

- Use **CTRL←** and **CTRL →** to position the character cursor at the character to be changed.
- Overtyping the old entry by typing on the keyboard in the normal way. each character you type replaces the one underlined by the character cursor.
- Use **INSERT CHARACTER** to make space for an additional character before the one underlined by the character cursor
- Use **DELETE CHARACTER** to delete the character underlined by the character cursor.

- Use the **DELETE** key to delete the character to the left of the character cursor.
- Use **DELETE END OF FIELD** to delete all the characters from the character cursor to the end of the field.

When you are satisfied with the record format, press **DATA** to return to the data screen to see the effect of your changes.

Note that if you blank out the display width for a field, that field will be shown 18 characters wide in the spreadsheet display, but in the card display it will not be shown at all.

Examining the database header

The second part of the format file to be looked at here is the header information relating to the database as a whole. To examine this information, press **DATABASE HEADER**. Your screen should look like this:



This information is record, presented in the card display. It also has just a single record.

Briefly, this is the meaning of each item of information:

Title

The title displayed at the top of the database

Display

Contains S if the database is to be shown in the spreadsheet display when it is first loaded; C if it is to be displayed in the card display.

Record size

This determines the space that is made available for each record as it is entered. It is described in detail in Chapter 4 Creating a database.

Capacity

This determines the largest number of records that ViewStore will try to fit into memory. It can be used to speed up movement around the database and to recover from the Record too big error message. This item is described in more detail in Chapter 4 Creating a database.

Index field

If a field name is entered here, the database is indexed on this field when it is first loaded. If left blank the records are shown in the order in which they were typed in.

Screen mode

The screen mode in which the database is displayed. If there is an entry in this field, ViewStore will change screen mode when switching to the data display if the screen mode you have specified is not the same as the current screen mode.

Experiment by changing the database description and screen mode. and then returning to the data screen (press DATA) to see the effect of your changes.

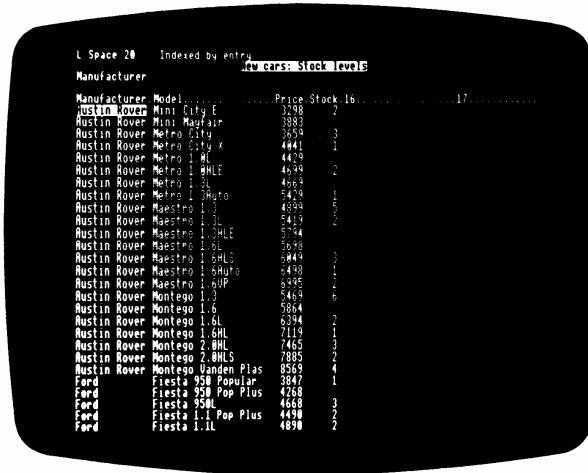
Alternative formats

As you saw earlier, when you load a database file, ViewStore looks for and then loads a format file with the same name. However, it is possible to have more than one format file which refers to the same database. Different formats enable the same data to be displayed rapidly in different ways for different purposes.

As an example of this, switch to the cOMmand screen and type:

LF stock RETURN

LF stands for load format- As you can see, it changes the format file being used with the database to F.stock Now if you switch to the data screen you will see something like this:



```
l Space 20 Indexed by entry
LF cars Stock levels
Manufacturer Model Price Stock
Austin Rover Mini City E 3298 2
Austin Rover Mini Mayfair 3883 1
Austin Rover Metro City 3659 3
Austin Rover Metro City X 4441 1
Austin Rover Metro 1.6L 4428 1
Austin Rover Metro 1.8HLS 4698 2
Austin Rover Metro 1.6L 4663 1
Austin Rover Metro 1.8Auto 5470 1
Austin Rover Maestro 1.3 4999 1
Austin Rover Maestro 1.6L 5419 2
Austin Rover Maestro 1.8HLS 5794 1
Austin Rover Maestro 1.6L 5698 1
Austin Rover Maestro 1.6HLS 6449 3
Austin Rover Maestro 1.8Auto 6490 1
Austin Rover Maestro 1.6UP 6995 1
Austin Rover Montego 1.3 5463 4
Austin Rover Montego 1.6 5864 2
Austin Rover Montego 1.6L 6394 1
Austin Rover Montego 1.6L 7119 1
Austin Rover Montego 2.0HLS 7465 3
Austin Rover Montego 2.0HLS 7885 2
Austin Rover Montego Vanden Plas 8569 4
Ford Fiesta 958 Popular 3847 1
Ford Fiesta 958 Pop Plus 4268 1
Ford Fiesta 958L 4668 3
Ford Fiesta 1.1 Pop Plus 4498 2
Ford Fiesta 1.1L 4698 2
```

This presents the Cars database in the form that might be required by a car dealer, as opposed to the customer's requirement for full details which is met by the original Cars format If you switch to the record format (press **RECORD FORMAT**), you will see that the effect has been achieved by setting the field width of all fields except Manufacturer, Model, Price and Stock to zero. If you want to return to the original format, type:

LF cars RETURN

You can also load a format file with a different name at the same time as you load the database, by typing:

LOAD *databasefilename formatfilename* **RETURN**

For example:

LOAD cars stock **RETURN**

Changing the index

Whilst you are displaying data (either in the spreadsheet display or the card display) you may have noticed the message **Indexed by entry** at the top of the screen This means that records are displayed in the order in which they were originally typed in.

This section explains how you can instruct ViewStore to display records in a different order.

Firstly, look at the record format (press **RECORD FORMAT**) Notice that three of the fields have a Y under I showing that an index file is being maintained for each of them **Indexname** contains the name of the appropriate file.

An index file is a special file which contains just the first few characters of each entry in the corresponding field; the number of characters is shown under **Key width** ViewStore uses the index file to arrange records in the order of the entries in that field.

If you go back to the command screen and catalogue the disc (see your disc filing system user guide) you should see files in directory I. These are the index files for the fields. Notice that they have the same names as shown in the **Index name** column of the record format.

You can tell ViewStore to display the Cars database in the order of the entries in any of the three fields for which it has an index file Go to the data screen and press **INDEX FIELD**. On the third line of the screen ViewStore displays the names of the fields for which it has an index. Type the name of the field you want ViewStore to use, and press **RETURN**.

Notice that you must give the field name, not the name of the index file In most cases the names will be the same. To save typing the full field name you can use the * wildcard character.

Wildcards

Wildcards are characters which have a special meaning when specifying field names or the values of field entries.

The characters are :

? which matches any single character.

* which matches any number of any characters (including none).

For example

?ill matches any four letter word ending in ill.

ill* matches any word beginning with ill, including ill.

Wildcards are most used when specifying field names and when locating data entries. For example `ma*` will match the field name `manufacturer` in the Cars database because no other field name begins with `ma`.

Now , try displaying the data in the order of each indexed field.

As when the file is first loaded, you see the message `R e a d i n g` and dots as the records are read from disc. But now when the data is displayed it will be in the order of the field you have selected, and the message at the top of the screen should read `Indexed by fieldname`. Note that records which have a blank entry in the index field are displayed at the beginning of the database.

If you get an error message refer to Chapter 7 Error messages

If you want to go back to Indexed by entry, press **INDEX FIELD** and then press **RETURN**.

If you find when you index your data that it does not fully follow alphabetical or numerical order , the probable reason is that the index file does not contain sufficient characters to enable ViewStore to distinguish between them. For example suppose an index file for the field `Manufacturer` held just four characters for each entry. Then ViewStore could not distinguish between `Austin Rover` and the `Australian Motor Company`. It would display all manufacturers beginning `Aust` together , but in the order in which they were entered.

Choosing the number of characters to be held by an index file is an important decision, which is described in Chapter 4 Creating a database.

Locating a record in a database file

ViewStore enables you to locate a particular entry in a particular field. However, before you can do that you must display the data in the order of that field. (If you do not have an index file for a field you cannot index on that field, and so you cannot locate an entry in that field. You may find a record using the `SELECT` utility; see Chapter 3 Extracting information from a database).

Suppose you are only interested in cars which have a top speed of at least 95

mph. First, index the data on the Speed field as described above. Then press LOCATE, and in response to the v a l u e ? prompt type in 95 and press **RETURN**.

ViewStore then redisplay the data, with the first record in which Speed is 95 shown at the top of the screen. If there is no record for which the field entry exactly matches your value, ViewStore displays the next one.

Note that ViewStore uses the index file for the field when locating records. If you enter a value which has more characters than are held in the index file, ViewStore will ignore the extra characters.

Again, take the case where the index file for M a n u f a c t u r e r contains the first four characters of each entry. If you enter A u s t i n R o v e r as your value, ViewStore will just look for A u s t , and it might come up with the AustralIan Motor Corporation.

Editing data

You can edit the data held by ViewStore at any time. Firstly, display the data in the normal way: you can use either the spreadsheet or the card display.

Then position the field cursor in the field and record you want to change. You can make changes in three main ways.

1 You can just type in the new entry. Each character you type replaces (or overtypes) the character underlined by the character cursor. If the new field entry is shorter than the old one, a convenient way to get rid of the left over characters is to press DELETE END OF FIELD: this deletes all characters from the character cursor to the end of the field. If you do not need to replace the whole of the old entry, move the character cursor (using **CTRL←** or **→**) to underneath the first character to be changed before starting to type.

2 You can add an extra character. Place the character cursor beneath the character before which you want the new character. Then press INSERT CHARACTER: ViewStore makes a space in which you can type the new character you want. For example, suppose you want to add an s to:

Autin Rover

Position the character cursor under the t :

Autin Rover

press INSERT CHARACTER:

Au_ tin Rover

and type the s :

Austin Rover

3. You can also delete surplus characters. Position the character cursor underneath the character to be deleted and press DELETE CHARACTER. ViewStore moves all the characters to the right of the character cursor one position to the left. You can also use the DELETE key to get rid of unwanted characters, but note that this deletes the character to the left of the character cursor DELETE is particularly useful when you are typing in an entry and notice that you have made a mistake in the last character typed.

Notice that it is possible for the character cursor to move into the gap which separates a field from the next field to the right Both INSERT CHARACTER and DELETE CHARACTER can work in this position; they operate on the character (if any) which is hidden behind the gap. Also, if an entry has scrolled to the left and the character cursor has returned to the beginning of the field, the DELETE key operates on the character hidden behind the gap to the left of the field.

Two other keys which can be useful for moving the character cursor quickly to the required position are BEGINNING OF FIELD and END OF FIELD.

To see how editing works, index the Cars data by entry (see Changing the index) and then try making these changes to the first record:

- Abbreviate Austin Rover to Aust Rov
- Change Mini City E to Mini City 1.0E
- Change the price to £ 3449
- Change the At date to 31.12.84

In the top left-hand corner of the screen you have noticed the message Space. followed by a number This is the number of bytes of space remaining on the disc for this record. Each character (including a space) takes up one byte of space If you enter an extra character the number goes down by one; if you delete a character the number increases by one.

If Space is reduced to zero, ViewStore will not allow you to enter any extra characters in the record you are currently editing; it will beep to warn you that you are out of space If the extra characters are essential you must make space by deleting characters from one of the other fields in the record.

If lack of space is a recurrent problem (perhaps because you have decided to add an extra field to each record) you can increase the space available for each record This is described in Chapter 5 Altering a database.

Adding new data

Before adding additional records to a database it is advisable first to index it by entry (see Changing the index) Then use CTRL ↓ to position the cursor on the blank line after the last record of the file.

If you are using the Cars file the space available for this new record should be very large; in fact, the full unused memory capacity of your computer.

You enter the information for the new record by typing it in the normal way. If you make mistakes you can edit them as you go along, using INSERT CHARACTER, DELETE CHARACTER and the **DELETE** key.

Practise entering information for a new car model; you can make up the details. As you type, the disc will probably whirr and the cursor will temporarily disappear from the screen. This is caused by ViewStore creating space for the new record on the disc. You can just carry on typing, and ViewStore will catch up with you when it has finished with the disc.

As you type each character the space available decreases by one byte, in the same way as when you were adding extra characters during editing. You may also notice that the space decreases when you first type a character in a new field.

When you have entered all the information you want for the new model, check the space available and remember it. Then move the cursor down to the next blank record, type a few characters and move back up again to the record you have just created. Check the space now available for this record. It should be greatly reduced, to about twenty bytes.

Remember that it is better not to begin typing a new record before you have completed the previous one; when you come back to complete the previous record you will find that you have very little space left for the additional information. If you need more than this, you must delete information from elsewhere in the record.

The rule is: enter all the information in a record before starting a new record.

Making more space available

When adding new data you may receive the `D i s c f u l l` or `Can't extend` message. In this case refer to Chapter 6 Filing systems.

Finishing a ViewStore session

As you have seen, ViewStore continually writes information to disc as you enter or edit it. When you have made all the additions or changes you want to your data, just return to the command screen. Then you can switch off your computer, or change to another program such as VIEW or ViewSheet or to a language such as BASIC. All your data will be safely stored on the disc.

If you are familiar with VIEW or ViewSheet you will notice that this is a major difference in ViewStore, unlike these other two programs, you do not have to

tell the computer to save your information; ViewStore does this for you automatically.

The only precaution that you must take is to return to the command screen before leaving ViewStore. Pressing **ESCAPE** once or twice will always take you back to the command screen.

If you switch off or press **CTRL BREAK** whilst in the data screen your files will be irretrievably damaged. As in all work with a computer, you are strongly urged to take backup copies of your files at regular intervals to ensure that, if damage occurs to the files on which you are currently working, you do not lose everything.

3 Extracting Information from a database

This chapter describes the ways in which information can be extracted from a database for different purposes. These purposes are:

- To provide macro parameters for use with the VIEW word processor.
- To pass numerical information to the ViewSheet spreadsheet program.
- To print labels containing information from the database.
- To print reports combining other text with information from the database.

Each of these functions is carried out using a ViewStore utility. Utilities are special programs provided with the system; they are supplied on your ViewStore disc in directory U.

In addition to these functions it is possible to pass text from your database for use with the VIEW word processor.

Several of the utilities allow you to print out information and so you should set up your printer. The following section describes how to do this. If you are confident about setting up your printer or you do not want to print anything, move on to the section Using utilities on page 31.

Preparing for printing

In order to print you must do three things:

- Connect your printer to your computer using the appropriate leads.
- Load the appropriate printer driver (if necessary).
- Give the relevant printing and baud rate codes.

The connecting leads for your printer should be supplied by your dealer.

A printer driver is a program which controls printing. In particular, it translates any highlight codes which you have included in a report specification into signals which are understood by the printer and which instruct it, for example, to start or end bold or underlined printing.

ViewStore contains a printer driver which operates by default if no other is loaded. The default printer driver enables most printers to be used, but without any highlighting. If your printer is capable of highlighting and you want to make use of it, you must obtain or create the appropriate printer driver.

program. Consult your Acornsoft dealer. (Printer drivers used with VIEW or ViewSheet will work with ViewStore, and vice versa.)

To load a printer driver , insert the disc containing the program in your disc drive and type:

PRINTER *name* RETURN

where *n a m e* is the name of the printer driver; usually the name of the printer itself. For example:

PRINTER RICOH RETURN

The **PRINTER** command without a name resets the printer driver to the default.

If a printer driver has been loaded, the **Printer** message is shown in the command screen followed by the name of the current printer driver. If no **Printer** message is shown, the default printer driver is in use.

Printing and baud rate codes tell the computer what type of printer it is sending information to, and the baud rate (transmission speed) at which the information should be sent.

Printing codes are:

- *FX 5,0 to print to the screen
- *FX 5,1 to print to a parallel (Centronics (R) type) printer
- *FX 5,2 to print to a serial (RS423) printer
- *FX 5,4 to print to an Econet print-server

Baud rate codes (necessary for serial printers only) are:

- *FX8,1 75 baud (characters per second)
- *FX8,2 150 baud
- *FX8,3 300 baud
- *FX8,4 1200 baud
- *FX8,5 2400 baud
- *FX8,6 4800 baud
- *FX8,7 9600 baud
- *FX8,8 19200baud

You give these commands after loading the printer driver. For example:

***FX5,2 RETURN**
***FX8,5 RETURN**

If you are uncertain about the codes to give, consult your dealer.

Using utilities

All utilities except SETUP operate on an existing database, and you must load the appropriate database in the normal way before starting the utility.

When using a utility you will usually need to use the names of fields in the database. It is advisable to type

LIST RETURN

to list the field names and numbers before starting the utility.

To use a utility you type:

UTILITY *utilityname* RETURN

ViewStore then examines the disc to see if it can find a file with the utility name that you entered, in directory U. If it finds the file, ViewStore starts the utility.

If it does not find the file, the system prompts you to Insert utility disc and hit a key. When you have done this it loads the utility into its memory, and then prompts you to Insert data disc and hit a key. You should insert the disc containing the database on which the utility is to operate. The utility procedure then starts in the normal way.

This enables you to keep the utility files on a different disc from your database and other files, which can be useful if you have a number of different databases on different discs.

If you type the utility name incorrectly ViewStore will not, of course be able to find a file with that name. It will assume that the file is on another disc and will give you the Insert utility disc and hit a key prompt. If you get this prompt unexpectedly check the name of the file that ViewStore cannot find; if you have made a typing mistake, press **ESCAPE** and start again.

Each utility (except SELECT and SETUP) starts with the question:

Use select file (N,Y)?

A select file is a database created using the SELECT utility (explained below) consisting of some or all of the records in your database which may have been sorted into a different order.

If you want the utility to operate on the full current database, type N and press **RETURN**. Note that you can just press **RETURN** to select N as it is the default value. Wherever ViewStore suggests a choice of values, the first value is the default value and can be specified by pressing **RETURN**. If you type Y and press **RETURN**, ViewStore looks for a file called

S . database, where da t a ba s e is the name of the currently loaded database and S.database is a select file. ViewStore then examines this file to find out on which records, and in which order, the utility should operate. If it cannot find S.database ViewStore displays an error message and ends the utility.

If you choose LABEL, REPORT or SELECT , ViewStore asks you whether output produced by the utility should be displayed on the screen or sent to the printer. Type S or P as appropriate, and press **RETURN**. Remember that the first value given, S, is the default value which can be chosen by pressing just RETURN.

Selecting and sorting records

ViewStore's utilities can be used with a whole database, or with a subset of the records in a database which have been selected and perhaps sorted in a particular way.

Selection of records is useful if you are only interested in certain categories of record; for example, only those cars manufactured by Austin Rover. The criteria you specify can be very complex. You could, for example, select all cars manufactured by Austin Rover which have a top speed over 100 mph, and Ford cars which are not hatchbacks, and Vauxhall car's fitted with a radio which cost less than £4000.

Sorting is useful if for example, you want to print addresses on labels and you want the addresses to be in alphabetical order of county. The sorting utility is more powerful than the indexing that you have met so far because it can take account of more than one field at a time. For example, your labels could be printed in alphabetical order of county; but those with the same county could be printed in order of town; and those with the same town could be printed in order of street.

Selected and sorted records are usually written to a special file on your disc. Then, when you run a utility you can instruct ViewStore to use this select file. The utility will only examine the selected records in the sorted order.

If you wish you can select certain records but not sort them; in this case they will be considered by the utility in the order in which they were originally entered. Or you can operate the utility on the whole database but sorted in the way you want. Utilities can also be used with the whole database in the order that it was entered, without either selecting or sorting the records.

This section explains how to carry out the selection and sorting procedure.

When specifying the selection and sorting conditions it is helpful to know the exact full names of the fields in the databases. If you are not completely sure

what they are, after loading the database give the command:

LIST RETURN

to list the field names and field numbers.

It may also be advisable to switch to screen mode 6 or 7 before starting the selection and sorting procedure so as to make available the maximum amount of computer memory.

Selection: an example

Before reading about the selection utility you may find it helpful to see it working. The file e.exe supplied on your ViewStore disc contains the instructions to select and list on the screen all cars manufactured by Austin Rover or Ford which have a petrol consumption of at least 40 mpg. To execute the file, type:

*EXEC e.exe **RETURN**

Starting selection

As the example shows, the utility gives you the opportunity to list the selected records instead of sending them to a file. They can be listed either on your screen or on your printer; if you will want to print the records, first set up your printer.

To use the selection utility, load the database Then in the command screen type:

UTILITY SELECT RETURN

ViewStore replies with the following prompts:

List or create select file (L,F)?

Enter L or F and press **RETURN**. L lists the records on the screen, F creates a select file containing the records. When specifying complex selection criteria it can often be a good idea to list the selected records first to ensure that the selection works as you expect. Then you can repeat the utility to create the select file. Note however, that if the selected records are listed they cannot be sorted. If you type L, you will see the following prompt:

Screen or Printer (S,P)?

Answer S or P and press **RETURN**.

If you type P, you will be asked:

Printer width (80)?

If you are listing the records on a printer ViewStore asks you for the printer width. If you just press **RETURN** it will assume the default width of 80 characters.

Select criteria?

If you do not want to carry out any selection (ie you want utilities to operate on every record in the current database), just press **RETURN** in response to the first Select criteria? prompt.

You must specify the records you want to select by giving the criteria that records must satisfy. A criterion takes the form :

field operator value

For example:

Mpg >= 40

means "select all records for which the entry in the field Mpg has a value greater than or equal to 40".

Field must be the name of one of the fields in the database. It can be entered using any case for the characters; in the example above the name could have been typed mpg or MPG or mPg. Note that you can use the wildcard characters either to abbreviate the field name or to specify several fields for which the criterion may be satisfied. You can also use the field number. for example, 10 in place of mpg.

For example if you specify

address* = Weymouth

ViewStore will search all the fields, which have a field name beginning with a d d r e s s , for W e y m o u t h . If the field found is a text field, ViewStore searches for the word W e y m o u t h anywhere in the field. If the field is an alphanumeric field, ViewStore expects the whole field to equal W e y m o u t h . If any of the fields searched match these criteria, the record is selected.

Operator can be:

= meaning equal to

<> not equal to

> greater than

< less than

>= greater than or equal to

<= less than or equal to

value can be anything which might match the contents of the field: for example the value 1000 will match 1000 in a numeric field, a text field and an alphanumeric field. Note that:

- The wildcards ? and * can be used in the value to match any characters. When comparing letters in a value ViewStore ignores their case, ie it assumes that A is the same as a.

Combining selection conditions

You can enter as many conditions as you want, using AND and OR to show how the conditions should be combined In the file e.ex1, for example, the conditions are:

```
Select criteria? manufacturer="Austin Rover"  
Select criteria? OR  
Select criteria? manufacturer=Ford  
Select criteria? AND Select criteria? mpg>=40
```

This instructs ViewStore to select all records for which manufacturer is Austin Rover or Ford, and for which mpg is greater than or equal to 40.

Note that the words Austin Rover are enclosed in double inverted commas. These are delimiters which specify to ViewStore where the value begins and ends. The first character after the delimiter is taken as the first character of the value and the last character of the value is the character immediately before the next occurrence of the same delimiter.

Delimiters can also be used with field names. They should always be used when the field name or value contains characters which are neither letters or numbers. Most punctuation characters can be used but " and ' are recommended to avoid confusion with characters special to ViewStore

If you have a field called £ you will need to enclose £ within delimiters:
' £ ' > 4000 for example.

Each time you press **RETURN** ViewStore repeats the **Select Criteria?** prompt. The conditions can be split on to as many lines as you like. For example, the conditions shown above could also have been entered:

```
Select criteria? manufacturer="Austin Rover" OR RETURN  
Select criteria? manufacturer=Ford AND mpg>=40 RETURN
```

However, an individual comparison (for example, manufacturer=Ford) cannot be split over two lines.

When examining a record to see if it meets the criteria, ViewStore considers

each comparison in turn, starting from the left. However, if the criteria contain both an **OR** and an **AND** operator it can often be difficult to work out exactly how the criteria will be applied, and ViewStore may not select records in the way you intended.

In this case it is advisable to use brackets to make your intentions absolutely clear. ViewStore carries out comparisons within brackets first, regardless of their order.

For example:

`(manufacturer=Ford AND sp>100) OR (price<6000 AND mpg>40)`

would select Ford cars with a top speed over 100 mph as well as cars costing less than £6000 with an mpg greater than 40. Alternatively:

`manufacturer=Ford AND (sp>100 OR price<6000) AND mpg>40`

would select Ford cars with an mpg greater than 40 which have either a top speed over 100 mph or cost less than £6000.

Brackets can also be nested. For example:

`(manufacturer=Ford AND (sp>100 OR price<6000)) OR mpg>40`

would select any cars with an mpg greater than 40 or Ford cars which have either a top speed over 100 mph or cost less than £6000.

If you are uncertain about how your selection criteria will be applied, it is advisable to list the selected records on your screen or printer before creating the select file.

Comparing selection values

The way in which ViewStore compares the value that you specify for an alphanumeric field with the entries in that field depends on the type of the field (see Examining the record format).

If the field is alphanumeric, the value is compared with the whole entry in the field.

If the field is textual, the value is compared with each word in each entry (ViewStore regards a word as being text which is preceded and/or followed by a space). For example:

Select criteria? `manufacturer=austin`

and

Select? `manufacturer=rover`

would both select all records for which manufacturer is Austin Rover (provided manufacturer is a T type field). Similarly:

Select? title=the

would select books called *Winnie the Pooh* and *The Guns of Navarone*; but not *Winnie-the-Pooh*.

If you use the less than (<) or greater than (>) operators with alphanumeric fields or textual fields, ViewStore compares items in the order in which they would appear in a dictionary; those that would appear earlier are smaller and those that would appear later are larger. For example:

Select? manufacturer > for

would include Ford and Vauxhall but exclude Fiat

Select? manufacturer < f

would exclude Fiat, Ford and Vauxhall. Notice that, if manufacturer is a textual field for which the comparison is made with each word in an entry, both these selections would include Austin Rover; Rover is greater than For, and Austin is less than F.

Carrying out the selection

A single selection criteria line can be up to 255 characters long. You can have as many lines as you want, subject only to the available computer memory.

When you have entered all the conditions you want, press **RETURN** in response to the Select criteria? prompt.

Then, if you asked ViewStore to list the selected records, it displays or prints those that meet your criteria in the order in which they were entered. Records are shown as they appear in the spreadsheet display, but truncated to the screen or printer width.

Sorting

When you have finished entering your selection conditions, if you are not listing the records, ViewStore will prompt you to enter the sorting conditions by displaying the message :

Sort field?

Remember that if you asked for the selected records to be listed they cannot be sorted, and so this prompt does not appear.

To enter a sorting condition, type the field to be sorted, followed if you wish by a comma and a number. For example:

Sort field? Mpg

or:

Sort field? Manufacturer,8

The field must be the name of one of the fields in the database. You can either specify it in full, use wildcards or use the field number; for example, instead of manufacturer. Either upper or lower case can be used; in the example above the name could have been typed mpg or MPG or mPg.

The number tells ViewStore how many characters to take into account when comparing entries in the field; if no number is given ViewStore uses the key width of the field as specified in the format file, if any. ViewStore always uses a width of 3 for date fields (because dates are stored using just three characters) and the display width for other fields. To give maximum speed of sorting and minimum use of disc space, the number of characters should be as small as is required to distinguish adequately between entries.

Any number of sorting conditions can be given, provided the total number of characters being used in the comparisons is not greater than 250. However you may find that there is insufficient memory when the number of characters being used is fewer than 250, especially in mode 0. This problem can usually be overcome by switching to a screen mode which makes more memory available in order to carry out the sort.

Where there is more than one condition ViewStore considers them in the order in which they were given.

For example, if you entered:

Sort field? Sp **RETURN**
Sort field? Manufacturer,5 **RETURN**
Sort field? Model **RETURN**
Sort field? **RETURN**

ViewStore would sort the records first in order of speed. Records for which the speed entry was the same would then be sorted in order of the first five characters of Manufacturer. Records for which both speed and manufacturer were the same would be finally sorted in order of Model.

When you have entered all the conditions you want, press **RETURN** in response to the Sort field? prompt.

Finally, tell ViewStore whether the sorts are to be carried out in ascending or descending order.

If you do not want to carry out any sorting (ie you want utilities to operate on records in the same order as in the current database), just press **RETURN** in response to the first Sort field ? prompt.

Extracting the records

When you have entered all the selection and sorting conditions you want, ViewStore starts to examine each record in the database in turn to see whether it meets the selection conditions.

As ViewStore examines each record a dot appears on the screen to give you an indication of its progress. When it has examined all the records ViewStore displays a message showing the number that met the selection conditions and the total number of records.

The program then sorts the selected records as you have requested, again printing out dots so that you can monitor its progress (You may see the **Memory full** message at this stage, showing that there is insufficient memory available to carry out the sort. In this case change to a screen mode that takes less memory and repeat the utility.)

When both processes are complete, the selected, sorted records are written to a file called *s.database* where *database* is the name of the original database file from which the records have been taken; for example, *S.cars*.

If you decide at a later date to select and sort the records from the same database in a different way, these records will also be written to *S.database* and the file containing your original selection will be deleted. If you want to retain your original selection, rename the file before carrying out the new select and sort.

S . database is not a complete database and you cannot load it. To edit the records it contains you must first create a new database (using **CONVERT**) which incorporates the selected and sorted records.

Depending on the amount of sorting that is required, ViewStore may have to create a working file called *S.SRTINT* during the sorting process. Once the process is completed this file is no longer required and is deleted.

VIEW macro parameters

One of the most powerful features of Acornsoft's **VIEW** word processor is its ability to use macros.

A macro is a block of text, such as a letter, which contains gaps. You provide the information to go in these gaps as a set of parameters; when the document is printed **VIEW** automatically inserts the correct information in each gap. By

providing many sets of parameters you can use a single macro to print many copies of the document, with the information in the gaps being different in each one. (For a full explanation of the use of macros, see the VIEW documentation.)

ViewStore's MACRO utility enables you to extract information from your database and to convert it into parameters which can be used in conjunction with a VIEW macro. The entries in one record are used to create one set of parameters.

First, load the database from which you want to prepare the macro parameters.

Type :
UTILITY MACRO **RETURN**

You will see the following prompts:
Use select file (N,Y)?

Tell the system whether you want the utility to be applied to the full database or only to the records in the select file.

Field 1?

Enter the name of the field from which the first parameter in the macro call should be taken.

ViewStore then prompts you for the other field names which are to form the parameters. You can enter up to ten field names. If you require fewer than ten, just press RETURN in response to a prompt. For example:

Field 1? Model
Field 2? Manufacturer **RETURN**
Field 3? Price **RETURN**
Field 4? At **RETURN**
Field 5? **RETURN**
Two letter macro?

Enter the two letter name of the macro to be used with these parameters; for example, AB or MA. The name must not be the same as a VIEW stored command; see the VIEW documentation.

Output filename?

Enter the name of the file which is to contain the parameters. It should be different from any existing file on your disc. ViewStore creates the new file in the current directory, unless you specify otherwise.

ViewStore then creates the file containing the parameters you have requested. If a field entry contains commas (for example, in an address) the corresponding

macro parameter is surrounded by angle brackets to ensure that the macro works correctly. If a single parameter set would be more than 132 characters long, ViewStore will notify you that it is splitting the call on to two lines. In this case you will need to abbreviate the macro in VIEW to make it fit on one line.

Using the fields in the example above, ViewStore might create macro parameters like this:

```
AB Mini City E,Austin Rover,3298,3.9.84
AB Mini Mayfair,Austin Rover,3883,3.9.84
AB Metro City,Austin Rover,3659,3.9.84
AB Metro City X,Austin Rover,4041,3.9.84
```

ViewSheet linking files

Unlike most spreadsheet programs, Acornsoft's ViewSheet can be used with linking files of numerical information. ViewSheet provides commands enabling these files to be created within a spreadsheet using data from the spreadsheet, and also enabling the data to be read back into a spreadsheet. Linking files allow numerical information to be transferred from one spreadsheet to another without laborious retyping. Full details of linking files and their use are given in the ViewSheet documentation.

ViewStore's LINK utility enables you to create linking files containing numerical information extracted from your database, in a format that can be accepted by ViewSheet. This information can then be read into a spreadsheet in the normal way, allowing transfer of data from ViewStore to ViewSheet. Each record from your database forms one row of the linking file; each field that is selected forms one column of the file.

To create a ViewSheet linking file, first load the database from which you want to extract the information. Type:

UTILITY LINK RETURN

You will see the following prompts:

Use select file (N,Y)?

Tell the system whether you want the utility to be applied to the full database or only to the records in the select file. Note that the maximum number of records from which information can be included in a linking file is 255.

Field 1?

Enter the field name for which the first entry in each row of the linking file should be taken.

Remember that the linking file can only contain numerical information. If you

enter the name of a field which is not numeric, ViewStore will print an error message and repeat the prompt.

ViewStore then prompts you for the names of the other fields from which information is to be extracted. When you have entered all the fields you require, just press **RETURN** in response to a prompt. For example:

```
Field 1? sp RETURN
Field 2? Mpg RETURN
Field 3? Tank RETURN
Field 4? Price RETURN
Field 5? RETURN
```

The maximum number of fields that can be included in the linking file is given by the formula $(10710/\text{number of records})$. This means that if your file will contain the maximum number of 255 records you can use up to 42 fields.

Output filename?

Enter the name of the linking file to contain the information, which should be different from any existing file on your disc.

Note that ViewSheet requires linking files to have names of the form V.V Sn, where n is a number from 1 to 255. You do not have to use a name of this type, but if you do not you will have to rename the file before it can be read by ViewSheet.

ViewStore finally creates the file containing the information you have requested and tells you the size (in rows and columns) of the file. In most cases you will then want to copy the file to the disc containing the spreadsheet which will use it. If you want to switch to ViewSheet, remember that you must give the command:

***SHEET RETURN**

ViewSheet linking files containing totals and subtotals of numerical data can also be created using the REPORT utility which is described later in this chapter.

Note that a linking file created using either ViewStore or ViewSheet cannot be used as a ViewStore database.

Labels

The LABEL utility enables you to extract information from your database for printing on to labels. One label is printed per record.

First, load the database from which you want to print the labels and then

prepare the computer for printing unless you will just want to list the labels on the screen. Type :

UTILITY LABEL **RETURN**

You will see the following prompts:

Use select file (N,Y)?

Tell the system whether you want the utility to be applied to the whole database or only the records in the select file.

Screen or Printer (S,P)?

Instead of printing the labels, you can if you wish display them on the screen; this can be helpful if you want to check that they contain the correct information before printing them In this case answer **S** in response to the **Screen or Printer** prompt. If you want to print the labels type **P** and press **RETURN**.

Label height (Lines)?

Enter the number of printed lines on each label.

Lines between each label?

Enter the number of blank lines between the last printed line on a label and the first printed line on the label below it.

Width of label?

Enter the maximum number of characters on each printed line If a particular line is longer than this number, ViewStore will truncate it.

Characters between each label?

Enter the number of blank character spaces between the end of a printed line on one label and the first character of the line on the label next to it (only relevant where there is more than one label per row).

Number of labels across the page?

Enter the number of labels on each row.

Line 1?

Enter the field name (or names) from which information is to be extracted for printing on the first line of each label.

If there is more than one field on a line the field names must be separated by commas. The entries from the fields are printed with one space between them. ViewStore then prompts you for the names of the fields to be printed on the

other lines of each label. When you have entered details of all the lines you want, just press **RETURN** in response to a prompt.

If ViewStore does not recognise a field name it prints an error message and repeats the prompt.

Alignment print (N,Y)?

Specify whether ViewStore should pause after printing (or displaying) two rows of labels. This enables you to check and, if necessary, to alter the positioning of the labels on your printer.

Reprint first Labels again (N,Y)?

Enter N if you have altered the positioning of the labels on your printer. It will not print the remaining labels until you tell it that the first two rows have been printed correctly (ie do not need reprinting). If you realise after printing the first two rows that the layout of the labels is incorrect you can press **ESCAPE** to abandon the print.

If the field (or fields) on a line is (or are) empty for a particular record, ViewStore closes up the other lines on the label.

If ViewStore finds that the information on a particular line would be wider than the label width it will truncate it (ie print only the first label width characters of the information). At the end ViewStore tells you how many labels it has printed and how many fields were blank.

For example, you could enter:

Label height (lines)?5
Lines between each label?3
Width of label?20
Characters between each label?5
Number of labels across the page?2
Line 1? Model,Drs,Type
Line 2? Manufacturer
Line 3? Sp,Acc,Mpg
Line 4? Price,At
Line 5?

pressing **RETURN** after each entry.

These instructions applied to the Cars database might produce labels like this:

Mini City E 2 S	Mini Mayfair 2 S
Austin Rover	Austin Rover
82 17.2 40	82 17.2 40
3298 3.9.84	3883 3.9.84

Metro City 3 H	Metro City X 3 H
Austin Rover	Austin Rover
88 18.7 39	88 16.5 39
3659 3.9.84	4041 3.9.84
Metro 1.0L 3 H	Metro 1.0HLE 3 H
Austin Rover	Austin Rover
88 16.5 39	85 18.6 39
4429 3.9.84	4699 3.9.84

Reports

The ViewStore REPORT utility is particularly useful for printing tables containing information extracted from your database. A report can incorporate text which you specify, and it can include totals and subtotals which are automatically calculated by the system and printed in the appropriate positions. Totals and subtotals can also be passed to linking files for use with ViewSheet.

Listing data

The simplest form of report that can be produced is a straightforward listing of the information in your database. You can select and/or sort the records to be included in the report using the SELECT utility. The listing can be produced on your screen or be sent to your printer.

First, load the database from which you want a report. If you want a printed report, prepare the computer for printing. Type:

UTILITY REPORT **RETURN**

Note that producing a report requires a substantial amount of memory. If ViewStore gives you the **Memory full** message when it tries to load the utility, change to a screen mode that makes more memory available and start again.

You will see the following prompts:

Use select file (N,Y)?

Tell ViewStore whether you want the report to include the full database or only the records in the select file.

Screen or Printer (S,P)?

Tell ViewStore whether you want your report listed on the screen or the printer.

Use report format file (N,Y)?

Enter N (or just press **RETURN**) if you want to produce a simple listing of your database. If you enter Y ViewStore will ask you for the name of the report definition file which gives details of the structure of the report; see the next section on More complicated reports.

If the report is being displayed on the screen ViewStore skips the remaining questions and lists the records immediately.

Printer width (80)?

Enter the number of characters that your printer can fit on one line. If you just press **RETURN** ViewStore will assume a width of 80 characters.

Page Length (66)?

Enter the number of lines that your printer can fit on one sheet of paper. If you just press **RETURN** ViewStore will assume a length of 66 lines.

Single sheets (N,Y)?

Enter Y if you want ViewStore to stop printing after each page of information.

This enables you to put a new sheet of paper in the printer for each of the report. If you are using continuous stationery or know that the report will fit on a single sheet, enter N or just press **RETURN**.

If you enter Y against **Single sheets**, ViewStore displays the Page.. prompt. Make sure that the paper is positioned correctly in the printer and then press any key to start printing. Otherwise, printing starts immediately. If you press M ViewStore will miss out that page. Pressing **ESCAPE** will abandon the printing.

ViewStore lists the information using the same field widths as for the spreadsheet screen display.

One line is produced per record as this is a simple listing. ViewStore lists as many whole fields as it can fit within printer width or the screen width; it does not list part fields.

On printed reports, field names are printed at the top of each new page.

ViewStore prints a blank line after the field names and two blank lines at the bottom of the page; so if your page length is 70, 66 record lines will be printed on a page.

If you entered Y against Single sheets ViewStore pauses after printing a page and redisplay the Page . . prompt. It does not continue printing until you press a key, enabling you to insert the next sheet of paper. Otherwise, printing continues until all records (or all selected records) have been printed.

More complicated reports: an example

If you want to produce reports which are more sophisticated than a simple print-out you must first create a report format definition file or report file for short. This file is in the form of a specialised database which gives details of the layout of the report and of the fields to be printed in it.

As an example of the type of report that can be produced in this way, try executing the file e.ex2 which is supplied with your ViewStore disc. This first uses the SELECT utility to select all models from the Cars database for which stock is greater than 0, and then uses the REPORT utility to list on the screen details of these models and of the stock value. To produce the report, execute the file by typing:

```
*EXEC e.ex2 RETURN
```

Notice that the file switches to screen mode 6 before starting the REPORT utility. This is to ensure that there is sufficient computer memory available. You may have noticed that the REPORT utility used a report file called example. The following sections explain how to create report files.

Setting up a report file

Report files are created using the SETUP utility. Type:

```
UTILITY SETUP RETURN
```

You will see the following prompts:

Set up database or report (D,R)?

Enter R to tell ViewStore that you want to set up a report file.

Name of report?

Enter the name to be given to the file ViewStore creates a blank file with this name in directory R.

Structure of a report file

Having set up a report file you can enter the information which defines the corresponding report. You do this by typing in the information as if the file were a normal database.

First, load the file by typing:

LOAD R.filename REPORT RETURN

where *filename* is the name of your report file and F.REPORT is a special format file provided with the system Try loading the example report file provided with the system by typing:

LOAD R.example REPORT RETURN

When you switch to the data screen you will see the database that you use to define the layout of a report. Each record in this database defines one or more lines in the report, and each record has four fields. Move the cursor into each of these fields in turn.

Type is a single character code which tells ViewStore how to interpret the information in this record. The valid codes are shown at the top of the screen and will be explained later.

Half1 and Half2 are used for details of the text to be printed in the report The reason for splitting the text into two fields is to enable you to align your reports on the screen.

Field list is used for details of the database fields from which information will be extracted for printing in the report.

Defining a text line

Each text line consists of:

- Characters which are printed on the report exactly as they appear on the text line and/or
- Formats, which are replaced with information when ViewStore prints the report. Usually, the information is taken from a field in your database.

A format is indicated by the special at character (@). The @ character always has this special meaning in the report definition file.

You show ViewStore the size of each format by the number of @ characters that you enter. Then when it is replacing a format, ViewStore truncates the information (if necessary) to the size of the format; or if the information is shorter than the format, ViewStore adds blanks. For example, if you had a text line like this:

Model: @@@@@@@@@@@@@@@@@ Price at @@@@@@@@@@ is
£@@@@@.@@ (exc VAT)

it might look like this when it was printed:

Model: Metro 1.0L Price at 15th Jan is £ 3851.30 (exc VAT)

A format which is to be replaced by a number can include a decimal point. You do not have to include the decimal point in a number format, but if you do the numbers filled in in that format will always be printed with that number of decimal places.

If a number has more decimal places than the format it will be rounded. If it is too large to fit in the format it will be shown in exponential (E) notation with the required number of decimal places. If it cannot be fitted in the format even in exponential notation, ViewStore prints a single % character. For example, this is how certain numbers would appear when printed in different formats:

Number	Printed in @@@ . @@	Printed in @@.@	Printed in in @@@@	Printed @@
0	0.00	0.0	0	
1	1.00	1.0	1	
1.565	1.57	1.6	2	
65.2	65.20	65.2	65	
12345.67	1.23E4	%	12346	

As you see in the example report file, a number format can also use the character b after the initial @. Then if the number to replace that format is zero, ViewStore will print blanks instead of zeros. This can improve the appearance of a report. For example, if a text line was like this:

Acceleration (0-60 in secs): @.@ Top speed (mph): @bb Mpg: @b.b

and each format was to be replaced by zeros, the line would be printed like this:

Acceleration (0-60 in secs): 00.0 Top speed (mph): Mpg:

The field list

As explained already, when it prints a report ViewStore replaces each format by an item of information. To find the information that it is to use for each format, ViewStore consults the field list; that is, the fourth field in the record of the special report file database.

The field list can contain five types of entry

- Field names.
- Arithmetic expressions.
- Register operations (see the next section).
- Page ejects.
- Comments.

A field name must be the name of a field in the database to which the report applies. For each line of the report, ViewStore prints the entry in this field of

the record on which it is currently reporting. It may be followed by $\backslash n$ which indicates that printing should start at the n th character of the field.

Arithmetic expressions can include :

- Numbers.
- The names of numeric fields
- The following operators (which are shown in their order of precedence):
 - () brackets
 - * multiply)
 - / divide)
 - + plus)
 - minus)

Arithmetic expressions can also include registers (see the next section)

ViewStore evaluates expressions using, in place of the field names, the value of the entry in that field of the record on which it is currently reporting. For example, valid expressions taken from the Cars database could be:

$\text{mpg} * \text{tank}$ (to calculate average distance between fill-ups)
 $454 - 6 / (\text{m pg} * 1 - 609)$ (to calculate fuel consumption in litres/100km)

Page ejects are indicated by the hat character \wedge followed by P. You can put this in a total or subtotal line.

Comments are indicated by the hat character \wedge followed by C followed by a number from 1 to 9 Comments are explained in detail on page 54.

To show ViewStore which information it should use in place of a particular format, enter the appropriate field name, arithmetic expression or hat value (or register) in the Field list column on the same line as the format.

Field names can be enclosed within delimiters according to the usual rules and wildcards may also be used within the delimiters.

There must be a field list entry, separated from the other entries by commas, corresponding to each format on a line. ViewStore replaces the first format on the line with information from the first field list entry, and so on. If a format contains $\backslash s$ or a decimal point (ie it is a numeric format), the corresponding entry must be the name of a numeric field or an arithmetic expression.

See for example the R type line of the example report file. For each record in the database, the first format is replaced by the entry in the field **Model** , the second format by the entry in the field **s t o c k** and the fourth format by the value of the expression $\text{stock} * \text{price} * 100 / 115$. Note that, although this text line only has four formats, there are six items in the field list. The last two items are register operations; they are described in the following sections.

Registers

Registers are similar to numeric variables in BBC BASIC. They are used to calculate values during the production of a report.

Each register has a single letter name from A to Z; ViewStore treats upper and lower case letters as referring to the same register.

Page and record registers

Two registers have special functions. These are P (the page register) and R (the record register). The values of these registers are reset to one at the beginning of a report. Then the value of R increases by one after each record for which information is printed. The value of P increases by one after each page of the report.

The current values of the registers can be printed at any point in a report. To print the value of a register in a report, you must include it at the appropriate position in a field list as follows:

IP to print the current value of the page register
IR to print the current value of the record register

See for example line 1 of the example report file, which prints the page number at the head of each new page of the report.

Using other registers

All the other registers can be used by the person creating the report file to calculate any values that are required.

Each register (except P and R) in fact holds two values: a total value and a subtotal value. Both values of these registers are set to zero at the beginning of a report (unlike P and R, which are set to one). The subtotal values of all registers are set to zero after a set of subtotal lines has been printed.

To tell ViewStore to add a value to a register, you must include in the field list a register operation of the form:

r:arithmetic expression

where r is the single letter name of the register, and the arithmetic expression can include anything which can be included in a field list expression (see above). The operation instructs ViewStore to add the value of the expression to the current value of the register.

For example :

A:1

adds 1 to register A, and:

V:stock*price*100/115

adds the stock value of the model in the current record (with V AT deducted) to register V.

Register operations do not cause anything to be printed and do not need a corresponding format in the text line. Operations should be separated from other entries and from each other by commas.

For example, each time line 6 of the example report file is printed, the value of Stock is added to register A and the value of stock*Price*100 /115 is added to register V .

Note that each value is added to both parts of the appropriate register, ie both total and subtotal Even if a value is rounded for printing, the exact value (to 9 significant figures) is added to the register.

Printing register values

To print the value of a register in a report, you must include it at the appropriate position in a field list as follows:

| S n to print the subtotal value of register n (eg |S A)
| T n to print the total value of register n (eg |T V)

The value of a register can be included in an arithmetic expression, either to specify a number to be printed or to increase the value of another register.

As examples of the ways in which register values can be used, look at these lines of the example report file:

- Line 8, where the subtotal values of registers A and V are printed.
- Line 10, where the total values of registers A and V are printed.
- Line 11, which calculates and prints the average stock value per model by dividing the total value of register V by the value of register R (that is, the number of records processed by the report).

Line types

The remaining entry that is required on each line of the report file is the line type, which is indicated by a single character code in the **Type** column.

There are four types of printed line:

R	record line
H	header line
S	subtotal line
T	total line

Record lines are printed once for each record in the database to which the report applies. Each set of record lines has the same layout. ViewStore replaces the formats on each line with the information shown in the field list using, where a field name is given, the entry in that field of the record. There can be as many record lines as you want. All the lines are printed for the first record that is processed: then all the lines are printed for the second record, and so on.

Header lines are printed at the top of each page of the report. There can be as many header lines as you wish, and they are printed in the order in which they appear.

Subtotal lines are printed whenever the value of the entry in a particular field changes. You specify the relevant field (or fields) when you initiate the report. (As you may have noticed, the file e.ex2 specifies Manufacturer as the subtotal field. As a result the three subtotal lines in the example report file are printed whenever Manufacturer changes: for example, from Austin Rover to Ford or from Ford to V auxhall.)

You can specify more than one subtotal line; all the lines will be printed at each change of the value of a selected field, in the order in which they appear in the report file. (Note that the example includes blank subtotal lines to improve the readability of the report.) You cannot specify different subtotal lines for different fields. Note that the printing of a subtotal line resets to zero the subtotal values of all subtotal registers.

Total lines are printed at the end of the report, ie after the last record in the database has been processed. There can be as many total lines as you wish, and they are printed in the order in which they appear.

In addition to the printed lines there are three line types which are not printed. The codes for these lines are:

P	page length
M	margin
C	comment

P and M are followed, not by the usual text line and field list, but by just a number in Half1. C is followed by text in Half1 which will be a prompt for a comment.

Page length gives the number of lines on the page. If more than one page length line is included, ViewStore ignores all except the first. If no page length line is specified, ViewStore uses a length of 66 lines.

Margin gives the number of lines normally left blank at the bottom of each page. These lines are left blank unless the next line to be printed is a subtotal or total line; in this case these lines are printed in the margin, up to the end of the page. Margin should always be at least as large as the number of subtotal and total lines to prevent totals being printed on their own at the top of a new page. If more than one margin is included, ViewStore ignores all except the first. If no margin is specified, ViewStore uses a margin of 4 lines.

A *comment* is an entry in a report which has a field format in the report definition file but text is supplied by the user when printing out the report rather than coming from the fields specified in the field list. Therefore the comment can be different each time the report is printed.

At the beginning of a report, before anything is printed, REPORT will step through each report line with a Type of C, output the text specified in Half1 and wait for a comment to be typed in. As it prints the report, when the utility reaches ^C1 in the field list, the response to the first prompt will be printed. When it reaches ^C2, a response to the second prompt will be printed, and so on.

In the example file the date of the stock summary is entered as a comment. Note the C line with the prompt Date? and ^C1 where the date is to be printed.

Highlight characters

Your printer may be capable of highlighting text in various ways; for example, underlining or boldening. To make use of these facilities in a ViewStore report you must include highlight codes in the field list which show where the highlighting is to start and end.

A highlight code is a number from 1 to 9 preceded by the hat ^ character. For example, ^1 represents highlight 1. Highlight codes must be separated from other items in the field list by commas. For example, suppose that in the example report file you wanted to print the stock value for the model highlighted in the way specified by highlight 2. Then the field list would look like this:

```
model,stock,price,^2,stock*price*100/115,^2,A:stock,v:stock*price*100/115
```

In this example the first occurrence of a highlight code turns that highlighting on; the next occurrence turns it off.

Note that, for the printer to understand what each highlight code represents, you must load a printer driver.

Requesting the report

When you are satisfied with the layout of the report as specified in the report file, return to the command screen.

Load the database from which you want to print a report.

If your report includes subtotal lines, it is advisable to sort the database in order of the field (or fields) for which changes in value will cause subtotals to be printed (This is the field entered in response to the **Subtotal?** prompt; see below) Otherwise the values in the field are likely to be jumbled up, so that subtotals will be printed after almost every record and will be meaningless.

Production of a report can take up a considerable amount of memory , and it is advisable to switch to a screen mode which makes available the maximum amount of memory before starting the utility. If you see the **Memory full** message during production of a report you will have to change screen mode and start again.

If the report is to be printed, prepare the computer for printing Then type:

UTILITY REPORT

You will see the following prompts:

Use select file (N,Y)?

Tell the system whether you want the utility to be applied to the whole database or only to the records in the select file.

Screen or Printer (S,P)?

Tell ViewStore whether you want the report to be printed or displayed on the screen.

Use report format file (N,Y)?

Enter Y. If you enter N (or just press **RETURN**) you will get a simple print-out of your data; see the earlier section on Listing your data.

Report filename?

Enter the name of. the report definition tile that you want ViewStore to use. This is the file you created using **SETUP** and then edited. The file must be in directory R.

Send totals to linking file?

Enter Y if you want ViewStore to write total and subtotal values calculated in the report to a ViewSheet linking file. If you enter N or just press **RETURN** this will not happen. If you enter Y, ViewStore then asks you:

How many totals across?

Enter the largest number of values printed on a total or subtotal line in your report.

Linking filename?

Enter the name you want to be given to the linking file.

Subtotal field?

If you want subtotal lines to be printed, enter the name of the field for which changes in the value will cause printing of the lines. The field must be in the database to which the report refers, but it need not be a field which is being printed in the report. After you enter a field name ViewStore repeats the prompt. If more than one field is to cause printing of subtotals, enter the name of the next field. If no more fields are required, just press **RETURN**. For example:

Subtotal field? Manufacturer **RETURN**

Subtotal field? **RETURN**

If you do not want subtotal lines printed, press **RETURN** in response to the first Subtotal field? prompt. In this case any subtotal lines in your report definition will be ignored, but the subtotal values of any registers will still be calculated and can be used elsewhere in the report.

Single sheets (N,Y)?

If you enter Y ViewStore stops printing after each page of the report; if you enter N or just press **RETURN** it produces the report continuously.

Prompt?

ViewStore prompts you for text for any comment entries. The prompts come from C lines in the report definition file and the responses are printed where specified by the ^ C n entries. Enter the appropriate response.

If you entered Y in response to the Single sheets? prompt ViewStore displays the Page1 . . prompt. If the report is to be printed make sure that the paper is positioned correctly in the printer. Then press any key to start the report. ViewStore will stop at the end of each page to enable you to insert a new sheet of paper in your printer; to restart printing hit any key. If you press M ViewStore will miss out the page. Pressing **ESCAPE** abandons the print. If you entered N the report will be printed continuously; your printer should be supplied with continuous stationery.

If you asked for a ViewSheet linking file to be created, ViewStore will write the amounts printed on each total or subtotal line of the report to a separate line of the linking file, in the order in which the amounts appear on the report line. The total line of the report forms the first line of the linking file ; the subtotal lines form the subsequent lines of the file, in the order in which they appear in the report.

For example, the example report file might produce a report looking something like this:

Model	Stock	Selling price	Value (excVAT)
Continental Co	1	7B10B	67920
Bentley Eight Turbo	2	49497	B60B2
Mulsanne	1	61743	53690
		55240	
Vehicles: 4		Value: 207691	Subtotalline
Silver Spirit		55240	
Silver Spur		1	62778 54590
Continental Co	1	76108	66181
Camargue		1	83122 72280
Vehicles: 3		Value: 193050	Subtotalline
Total vehicles: 7 Total value. 400742)			
verage value/model:		50093)Total lines
Average value/vehicle;		57249)

Then if a linking file had been requested for this report it would contain these entries:

7	400742	39105 7153	Total line
4	207691)
3	193050) Subtotal lines

IMPORT

IMPORT is designed to enable data from other databases to be converted into ViewStore format files.

IMPORT will not read discs in any format other than the formats normally read by the disc filing system in use. If the data is from another disc filing system such as CP/M, then the data must first be transferred to the BBC Microcomputer disc filing system. There are conversion programs available to do this and you should consult your dealer if you require one.

Starting IMPORT

Unlike the other utilities IMPORT is written in BBC BASIC so you should switch to BASIC by typing:

```
*BASIC RETURN
```

and then type:

```
CHAIN "IMPORT" RETURN
```

Using IMPORT

IMPORT will only handle data stored in ASCII format; it has no facilities to handle floating point numbers or other data formats. Data can often be output from a database system in a "transfer" format which is generally ASCII. If so, this is the format to use when transferring your data to ViewStore. It will sometimes be possible to operate on the database files directly as it is for Acornsoft's Database, but detailed information is required about the file format in order to do this.

If information about the format of a particular database's files is unavailable, it is often possible to use an operating system facility to send all screen output to a file - equivalent to the BBC Microcomputer's *SPOOL facility. If so, set this operating system facility in operation, and then issue a command to the database to list all fields of all records on the screen. This will generally create a large file full of data and spaces which can then be read by IMPORT.

IMPORT does not create a format file for the ViewStore database. The format file must be created using ViewStore's SETUP utility. Use ViewStore to set up the details of the information in the database once the data has been converted. The set of questions that IMPORT then asks you will vary depending upon the answers that you give; certain questions will not be asked after particular answers. If there is a record separator for example, IMPORT will not ask you for the length of the last field.

The questions

Source file

Give the name of the file on the current filing system in which resides the data to be converted.

Destination file

Give the name of the file into which you wish the converted data to go. It is advisable to ensure that this file does not exist at first, and use *COMPACT to compact the disc. If the file does exist, you will be asked if you want to delete it,

but this will leave a certain amount of space on the disc which may not be large enough to contain the converted data.

Position in file where data starts

Give the number of bytes between the start of the file and the start of the actual data. This allows any header information at the beginning of the database file to be bypassed. Replying with **RETURN** gives a zero offset.

Record separator

Give the sequence of bytes which the database uses to separate each record. If the database has fixed length records, there may be no record separator, in which case you should reply with **RETURN**. Each byte in the sequence is specified either as a decimal number giving the character's value, or as an ASCII character enclosed in inverted commas " ", or as a hex number prefixed with an ampersand. Each byte is separated by a comma. A question mark ? may be used as a wildcard to allow any byte to occur in that position.

0,?, "+"

for example, specifies a string of 3 bytes, starting with a null or zero byte, then a byte containing anything, and then the plus character. Note that more than one consecutive comma will not be treated as a sequence of null bytes as in some versions of BASIC ; a sequence of consecutive commas is equivalent to one single comma.

Appears before first record (Y,N)?

This question will only be asked if you have given a record separator. Databases may have the record separator either before a record, or at the end of a record. It is also possible for a database to store an extra record separator at the beginning of the database. Answer Y if there is a record separator before the first record. There is no default answer for this question.

Appears after last record (Y,N)?

This question will only be asked if you have given a record separator. Answer Y if there is a record separator after the last record in the database. There is no default answer for this question.

Field separator

Give the sequence of bytes used to separate each field. The database may not use field separators, in which case you should reply by pressing **RETURN**. You will then be asked for the number of fields in the record, and then the length of each individual field. The byte sequence is specified in the same way as for Record separator above.

Appears before first field (Y,N)?

This question will only be asked if you have given a field separator.

Answer Y if there is a field separator before the first field There is no default answer for this question.

Appears after Last field (Y,N)?

This question will only be asked if you have given a field separator.

Answer Y if there is a field separator after the last field.

End of file marker

If there is a marker which delimits the end of the data within the file (rather than the end of the file itself being the end of the data), then enter the byte sequence here as for the separators. Answer **RETURN** if there is no end of file marker.

Number of fields

This question will only be asked if **IMPORT** cannot work out how many fields there are in the record from the format of the data: if there are both record and field separators, then this question is superfluous. Reply with the number of fields in each record. This must be a whole number greater than 0.

Length of field n

This series of questions will only be asked if **IMPORT** cannot work out the size of each field from the format of the data: if there are no field separators, then **IMPORT** must know the length of each field to work out where each field ends. If there is a record separator then **IMPORT** will never ask for the length of the last field since this information is implicit in the data structure.

Is the data reversed (N,Y)?

Data which has come from BBC BASIC may well have the bytes within the fields stored in reverse order. Answer Y if the data is reversed. The default answer is N.

Leading character to skip

Data from another database may have leading spaces or another character in the field. These are undesirable in ViewStore databases since they waste a lot of space. You may enter up to one character here using the conventions described above for separators. Press **RETURN** if you do not wish to skip a leading character.

Trailing character to skip

Data from another database may have trailing spaces or another character in

each field to pad the field out to the field length. This is especially likely if an ASCII intermediate file has been created. You may enter up to one character here using the conventions described above for separators Press **RETURN** if you do not want to skip a trailing character.

ViewStore record size

Reply to this question with the number giving the size that you wish the ViewStore record to be created You can reply with a number prefixed with + , as in ViewStore, which will reserve the current record size plus so many bytes.This will be the most frequently used option. The default is + 20.

Once all the questions have been answered, IMPORT will switch to MODE 7 and start converting, giving a field and record count at the top of the screen.This process can be time-consuming.

Deleted records

IMPORT contains no facilities for ignoring deleted records These must either be removed by using the database's "remove deleted records" function, if there is one, or transferred and then deleted from the final database with ViewStore. If the deleted records are removed completely by the database, then there is of course no problem, or if the deleted records are completely overwritten with a deleted record character (as is done by ViewStore), then these will be ignored, or blank records will be created.

Specific examples

Here are example replies for two databases that are likely to be encountered, Acornsoft's Database, and Ashton-Tate's dBASE II.

Database

The conversion can be performed directly on the Database file without intermediate conversion.

Position in file where data starts

Type 817 Database files have a data header of 817 bytes

Record separator

Press **RETURN**. There are no record separators.

Field separator

Type 0,?. Database field separators are a null followed by a byte giving the field length.

Appears before first field (Y,N)?

Type Y . Database fields have the separator before, rather than after, them.

Appears after Last field (Y,N)?

Type N.

End of file marker

Press **RETURN**. Database has no end of file marker.

Number of fields

Enter the number of fields in each Database record.

Is the data reversed (N,Y)?

Type Y . Since Database is written in BBC BASIC , the field contents are stored in reverse order.

Leading character to skip

Press **RETURN**. Database fields are left aligned.

Trailing character to skip

Type &20. Database pads fields with trailing spaces. These are not desirable in a ViewStore database since they waste a lot of space.

ViewStore record size

Enter the number of bytes that you wish to reserve for each record. This is a trade off between flexibility and disc space. + 50 is recommended.

dBASE II

Firstly, you must produce an intermediate ASCII file using the COPY command:

```
.USE <filename1>.DBF  
.COPY TO <filename2>.TXT SDF DELIMITED WITH ,
```

F i l e n a m e 1 is the filename of the database you are converting.

F i l e n a m e 2 is the filename of the intermediate file. Once the intermediate file has been created, you must transfer it from CP/M (or other) format to BBC Microcomputer format Then you can use IMPORT , answering the questions as follows:

Position in file where data starts

Press **RETURN**. The data starts at the beginning.

Record separator

Type &D , &A . This specifies a carriage return linefeed which is what dBASE II has output.

Appears before first record (Y,N)?

Type N.

Appears after last record (Y,N)?

Type Y.

Field separator

Type " , " . dBASE II has output commas to separate the fields.

Appears before first field (Y,N)?

Type N.

Appears after last field (Y,N)?

Type N.

End of file marker

Press **RETURN**. There is no end of file marker.

Is the data reversed (N,Y)?

Press **RETURN**. The data is not reversed.

Leading character to skip

Press **RETURN**.

Trailing character to skip

Press **RETURN**. dBASE II has already stripped spaces.

ViewStore record size

Enter the number of bytes that you wish to reserve for each record. This is a trade off between flexibility and disc space +50 is recommended.

Using *SPOOL with utilities

The ***SPOOL** command enables you to send the output from a utility to a disc file which can then be used with the **VIEW** word processor. This is particularly useful with the **REPORT** and **LABEL** utilities. It enables you to print the same output a number of times without having to repeat the utility commands on each occasion. It allows you to edit the output before it is printed and it enables you to incorporate the output in another document.

First, type:

***SPOOL *filename* RETURN**

where *filename* is the name of the file which is to contain the output; for example:

***SPOOL text RETURN**

Then give the utility commands in the normal way, sending the output to the screen.

As the utility output appears on the screen it is also written to the named file. At the end of the output type :

***SPOOL RETURN**

to close the file.

The file can then be read into VIEW using the command:

READ *filename* RETURN

and can be amended and printed in the normal way. The file will contain the commands you gave to request the utility in addition to the actual output, but this unnecessary information can be easily deleted.

Full details of the ***SPOOL** command are given in your filing system user guide.

Passing text to VIEW

Any ViewStore database file can be read in to VIEW and can then be edited or amended using the normal VIEW commands. VIEW files cannot be edited in ViewStore even if they were originally created using ViewStore. However VIEW files can be converted to ViewStore format using IMPORT.

If you are using ViewStore, remember that you can switch to VIEW by typing:

***WORD RETURN**

In VIEW give the **NEW** command if you want to read in the ViewStore file on its own, or load a VIEW file to which you want the ViewStore file to be added. Then type:

READ *filename* RETURN

where *filename* is the full name of the ViewStore file; for example, to read in the Cars file you would type:

READ D.cars RETURN

VIEW reads in the full entry in each field. Fields are shown across the screen, as in the ViewStore spreadsheet display, with a tab character separating field entries. Entries in the same field can be lined up or given more space by adjusting the tab stops (ie the asterisks) in the VIEW ruler. For further details of the VIEW commands, see the VIEW documentation.

Each record is shown on a separate line, which may well be longer than the ruler. Note that text lines in VIEW are limited to 132 characters.

If you wish to save the information you can use the normal VIEW SAVE command. However, it is important that you give the saved VIEW file a different name to the original ViewStore file (even if it is only placed in a different directory), since the VIEW file cannot be read back in to ViewStore.

Using exec files with utilities.

As you have seen, most ViewStore utilities require you to enter answers to a number of questions. You may find yourself frequently using the same utility in the same way (for example, to produce a similar report or to sort your database in a particular way), and hence having to enter the same set of answers.

You can speed up this process by typing the command to start the utility and the answers it requires in to an 'exec file'. Then when you execute the file your computer will read in the answers as if you were typing them at the keyboard, and will carry out the utility automatically.

Full details of exec files are given in your filing system user guide. Briefly, you can create the file using the command:

***BUILD *filename* RETURN**

where *filename* is the name you want to give to the exec file. On each line of the file you type a separate command or answer. Then to execute the file you give the command:

***EXEC *filename* RETURN**

If you have VIEW you can also use it to create exec files and, more importantly, to edit them. For example, you could create a file which displayed a report on your screen using the REPORT utility. You would use the exec file to run and re-run the report until it was as you wanted it; then you would change the answer to the screen or Printer prompt from S to P, and execute the file again to produce a printed report.

As an example of how this works, the lines of the exec file e.ex2 are shown below, with the question being answered by each line alongside. The questions

do not, of course, form part of the exec file. Remember that this file selects all records in the Cars database for which Stock is greater than zero, sorts them in order of manufacturer, and then produces a screen report using the report definition file R.example.

L CARS	Loads the Cars database
UTILITY SELECT	Starts the select utility
	Screen or Printer (S , P) ? - blank response
F	List or create select file (L,F)?
Stock > 0	Select criteria?
Manufacturer	Select criteria? blank response ends entries Sort field?
	Sortfield? - blank response ends entries
A	Ascend or Descend (A,D)?
MODE 6	Ensures sufficient memory for the report
UTILITY REPORT	Starts the report utility
S	Screen or Printer (S,P)?
Y	Use select file (N,Y)?
Y	Use report format file (N,Y)?
example	Report filename?
N	Send totals to linking file (N,Y)?
Manufacturer	Subtotal field?
	Subtotalfield? blank response ends entries
N	Single sheets (N,Y)?
1 Jan 1985	Date?

Note that if you exclude the last line from the file ViewStore will stop at the Date? prompt, enabling you to enter a different date each time you execute the file.

To produce the report by executing this file, type:

***EXEC e.ex2 RETURN**

To simplify repeated production of the report even further you could program a function key to give the ***EXEC** command; for example:
`*keyO *EXEC e.ex2IM`

You can also program function keys to carry out utilities. For example, the **SELECT** part of the exec file shown above could be carried out by function key 1 which had been programmed by typing, in the command screen,

```
*key1 L CARSIMUTILITY  
SELECTIMIMFIMStock>0IMIMManufacturerIMIMAIM
```

and pressing **RETURN**

Note that, unlike in an exec file, the **RETURN**s must be specified using the code IM. Blank responses are given by repeating the IM The main disadvantage of function keys compared with exec files is that they cannot be edited.

Programming function keys in this way has no effect on the function key commands.

4 Creating a database

Planning your database

Building a new database can be a time-consuming business ; though the effort involved is repaid many times over by the speed and simplicity with which you can access the information once the database is built.

As you will see in the next chapter, ViewStore enables you to alter the structure of a database after it has been built, but changes to structure are best avoided if possible.

For these reasons, time spent planning your database before you start to create it is well worthwhile This section describes the points you should consider and the decisions you should take in order to build a database which most effectively and efficiently meets your requirements.

What data ?

Clearly, the first step is to decide what data you are going to include in your database. Specifically, you must decide what types of information you are going to store.

You may find it helpful to decide first of all what your database is for; that is, the ways in which you will use it. Then write down all the types of information that might be relevant to these functions, in any order.

The temptation when creating a database is to include everything that might possibly be of use. However, entering large amounts of marginally useful information is wasteful of time, computer memory and disc space. Your next step should therefore be to go through your list, ruthlessly crossing out all the categories which are not of real relevance. Should you build a second database to accommodate these other categories?

Finally, think about the way in which the information types will relate to the fields in your database In general, you will probably use one field for each type However, if you will want to index a list of people by surname you will need to split the information type Name into two fields: Surname and First names. Similarly you could have a single field for addresses; but if you will want to index the data by town you will need separate fields for Street, Town, County and so on.

How is the data to be recorded?

An item of information can be recorded in two basic ways; in full, or as a code. Where information is very variable it must be recorded in full: for example, names, addresses or telephone numbers. However, where each item must be selected from a fixed list it can save time and space to use a code: for example, Y for yes and N for no or, as in the Type field of the Cars database, S, E, H or C for saloon, estate, hatchback or coupe respectively.

Where a code is the best way to record one of your types of information, decide what the permitted entries are to be and the code to be used for each.

What limits are to be placed on entries ?

ViewStore enables you to place limits on the entries that it will accept in each field of your database. It is advisable to specify limits wherever relevant. They help to ensure consistency, and to prevent the user from inadvertently entering incorrect data, or data in the wrong field.

The limits can take two forms.

Firstly, you can restrict entries to those on a specific list; in the case of the Type field in the Cars database, the list consists of S, E, H and C only.

Secondly, you can specify the minimum and/or maximum value of an entry. In the Cars database, engine capacity must be between a minimum value of 500cc and a maximum value of 4000c. However, you can also specify minimum and maximum values for alphabetic entries; for example, that the first letter of an entry must be between D and G inclusive.

In what order are fields to be displayed on the screen ?

In the spreadsheet display, it is usually most convenient to have the information which is the principal identifier of each record in the first or left-hand field; for example, the name of the person or company, or the invoice number, and so on. Then arrange fields in the order which will simplify entry of data. If data will be copied from existing forms, for example, display fields in the same order as on the form.

As you will see, the order in which fields are shown in the card display does not have to be the same as in the spreadsheet display, and can be easily changed. However, whatever the arrangement of fields, data is stored in the same order as in the spreadsheet display.

What is to be the width of each field ?

Remember that ViewStore allows each entry to be as long as you wish. Field width therefore refers to the number of characters of each entry that can be

seen on the screen. Where an entry is longer than the field width the remaining characters can be seen by scrolling the entry across the column. It can simplify the use of the database if all the fields can be seen on the screen at the same time. This means that, in screen modes 0 and 3, the total column width must be 80 characters or less, and in screen modes 1, 4, 6 and 7 it must be 40 characters or less. (In calculating total column width, remember that ViewStore requires an extra blank character between each pair of fields and does not print in the last column of the screen). However, there is no point in having all the fields on the screen at the same time if they are so small that the contents cannot be easily read; so some compromise must be made. In some fields, such as a name, date or telephone number, it is important to be able to see the whole entry; for other fields such as addresses a window on to the entry five or six characters wide may be perfectly adequate.

Another point which you should bear in mind is that the visible field title can be no longer than the column width. For fields where the entry will never be more than a single character code it may be worth using a column width of three or four characters to enable a meaningful title to be displayed. Remember that the full field name is given on the prompt line if no prompt is supplied.

Finally , you should decide whether the entries in a field are to be restricted to the width of the displayed column. Entries in this field do not then scroll. It is often advisable to specify this for fields containing dates.

What name is to be given to each field ?

It is generally a good idea to choose a name for each field since this makes them easier to remember. However ViewStore automatically numbers unnamed fields and you can always refer to your fields by number.

Each field name used in the same database must be different. A name can be longer than the field width, but in the spreadsheet display the name is truncated to the field width. In some cases an abbreviation that fits into the field may be easier to understand than the truncated full name; for example `Extn` rather than `Exte` for `Extension`. In the card display the full field name is shown. A field name cannot be more than 15 characters long.

For which fields are indexes to be kept?

Any field for which an index is kept can be used to order the records in your database. For example if you want to see your records in order of price, you must create and maintain an index for the price field.

Indexes can be of two types, updateable and read only. Updateable indexes are amended as records are entered and are always correct. Read-only files are

created after records have been entered and if records are added or amended the index will no longer be correct.

ViewStore can keep read-only indexes for any number of fields. It can keep updateable indexes for up to 4 fields using the DFS. Note that the fewer updateable indexes you have, the faster you can move around the database.

Each index is kept in a file in directory I on your disc.

For each index you must decide two things:

1. The filename. If the field name is seven characters or less and does not contain punctuation characters, the file can have the same name as the field; this is usually the best solution. However, if the field name is longer than seven characters you must think of a different name for the file; usually, an abbreviation of the field name.

- 2 The key width for alphanumeric or textual fields. You should choose a key width which is just long enough to distinguish between entries. For example, a key width of 4 characters would not distinguish between *Austin Rover* and the *Australian Motor Company*, a key width of 5 would. The longer the key width, the slower the operation of the system and the bigger the index file. Key widths can be up to 100 characters, but should not usually need to be more than 10.

How much space will be reserved on the disc for each record in the database?

You can tell ViewStore to reserve space on the disc for each record in two ways.

Firstly, you can tell it to reserve the amount of space needed for the record when it is first entered, plus a fixed amount of extra space. This extra space might be required if an entry (such as an address) has to be replaced by one which is substantially , longer (for example, if 3 New Rd is replaced by 459A Northumberland Avenue); or if, you decide to add an extra field to every record.

The amount of extra space you request will depend on the extent of the changes you expect to have to make; a typical figure is 20-30 characters. You can specify zero extra characters, which would be a very efficient use of disc space but which would mean that, if additional characters needed to be added subsequently , other characters would have to be deleted from elsewhere in the record. The maximum extra space you can reserve is 512 bytes.

The second method is to tell ViewStore to reserve the same fixed amount of space for each record. This method is a less efficient use of disc space, except where all records have very similar lengths; for example, in databases consisting almost entirely of numbers. However, it prevents problems which

can occur with long records where there may be little available computer memory . It is recommended if records are longer than 512 bytes.

ViewStore gives you twenty extra characters for each record by default.

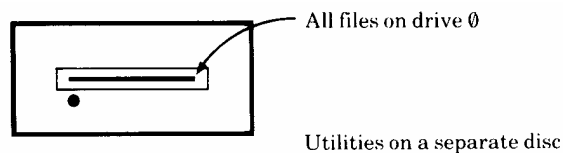
How will files be organised on your disc?

As you have seen, ViewStore makes use of a number of different types of file. ViewStore expects all files of the same type to be in the same directory; for example, all database files to be in directory D. It also expects all files of the same type to be on the same drive.

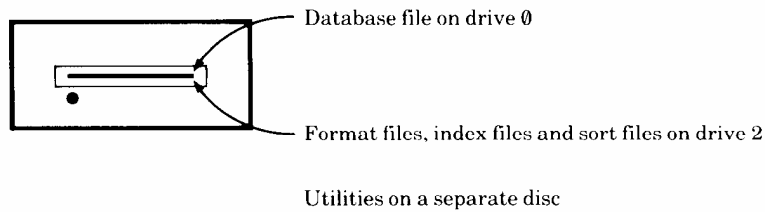
As you add records to your database the database file on the disc will grow. If the file bumps into another file the DFS will give you the Can't Extend message. Therefore careful file organisation is important.

The diagram below presents some recommended arrangements depending on the number of drives you have available.

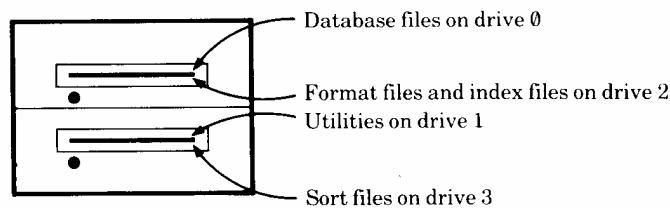
Single sided drive



Double sided drive



Dual disc drive



What is the database's title?

You can if you wish choose a title for the database which will be shown at the top of the screen when you are examining the data. This title can include any characters, but for obvious reasons it should not be longer than the width in the

screen mode you will be using (ie 80 characters in modes 0 and 3; 40 characters in modes 1, 4, 6 and 7).

ViewStore will work perfectly if you decide not to choose a title.

What is the database's filename?

You must choose a filename for your database. The name must be different from any other file on your ViewStore disc (or discs), and it must follow the normal rules for filenames.

Defining a new database

Setting up prefixes

This section explains how to tell ViewStore about the way in which your files are arranged on your working disc or discs; that is, on which drives they are to be found. This section can be ignored if you intend to keep all your files on the same drive.

In the command screen, type:

PREFIX RETURN

You should see a message like this:

```
D)ata  
F)ormat  
I)ndex  
S)ort  
U)tility
```

This lists the five principal types of file used by ViewStore and shows you that no prefix is currently applied to them. In other words, if ViewStore is looking for a database file called `c a t` it just looks for `D . c a t`; if it is looking for a format file it looks for `F . c a t`, and so on.

Now type:

PREFIX D :2.

This tells ViewStore to attach the prefix `: 2 .` to files of type `D`, that is, database files. Now if ViewStore is looking for a database file called `c a t` it will actually look for `: 2 . D . c a t`; in other words, for `D . c a t` on drive 2.

The command to change a prefix is:

PREFIX *letter drive* RETURN

where letter is `D`, `F`, `I`, `S` or `U`, depending on the type of files to which the prefix is to apply, and drive is the actual prefix. As you see, when you change a

prefix. ViewStore shows you all the current prefixes which it applies when looking for a file. Note that the appropriate prefixes are also applied when ViewStore creates files.

To reset a prefix to blank type:

PREFIX *letter* RETURN

Note that if the prefix is blank ViewStore will look for the file on the current drive. For example, if you change the current drive to drive I (using the ***DRIVE** command), ViewStore will look for an index file called `file name` under the name `. I . I . f i l e n a m e`, even though the prefix for I files is blank.

To avoid confusion it can be a good idea to specify the prefix for each file type. Following the recommended file arrangement shown above, you would give these commands for a double sided drive using DFS:

PREFIX D :0. RETURN
PREFIX F :2. RETURN
PREFIX 1 :2. RETURN
PREFIX S :2. RETURN
PREFIX U :0. RETURN

and these commands for a dual drive using DFS:

PREFIX D :0. RETURN
PREFIX F :2. RETURN
PREFIX 1 :2. RETURN
PREFIX S :3. RETURN
PREFIX U .1. RETURN

Chapter 8 Notes describes how to do this automatically

Creating the files

Having planned your database you are ready to start creating the required files using the SETUP utility.

Insert in your disc drive the disc (or discs) on which you want the files to be created It is advisable to have no more than one database file on the same side of a disc.

Start ViewStore in the normal way and in the command screen, type:

UTILITY SETUP RETURN

You will see the following prompts:

Set up database or report (D,R)?

Enter D to show that you are setting up a database. R is used to create a report definition file; see the Reports section of Chapter 3.

File name of database?

Enter the name to be given to your new database file. On the DFS this should not be longer than seven characters. ViewStore automatically creates the file in directory D.

Format filename (database filename)?

Enter the format filename. If this is the same as your database filename just press **RETURN**.

How many bytes to reserve?

Enter the amount of space that you want to be reserved on disc for your database, in bytes. Note that the larger the database file the longer it takes to load. If your database is the final file on your disc or if you are using the ADFS there should be no **C a n ' t e x t e n d** problems; in this case enter zero. Otherwise you can estimate the space required using the formula:

Average characters per record X Expected no of records x 1.1

Your database can take up more space than this if it proves necessary, but this may cause **C a n ' t e x t e n d** problems. It is therefore advisable to overestimate rather than underestimate; but remember that this will leave you with unused space on your disc and the database will take longer to load.

Number of indexes?

Enter the number of updateable index files that you will want to use with the database. The maximum on the DFS is 4; if you also want to create read-only indexes the maximum number of updateable indexes is 3. The number of indexes may be zero. If so, the following three prompts are not displayed.

Number of records?

Enter the number of records that you expect to have in your database. ViewStore uses this number in reserving disc space for index files. As with space for the database, you can exceed this number, but this may cause problems. Therefore, overestimate rather than underestimate. If you are not using the DFS enter 0.

Index:n
Filename?

Enter the filename of each updateable index. If you are using the DFS this should not be longer than seven characters. As discussed earlier, an index filename will often be the same as the name of the field for which the index is

being maintained. Enter the filename. ViewStore automatically creates each file in directory I.

Keysize?

Enter the keysize to be used in each index, that is, the number of characters from each entry to be stored in the file. If the field is to contain dates, the keysize should be 3. If the field is to contain numbers, the keysize should be the same as the field width. If the field is to contain alphanumeric entries, the keysize should be the minimum needed to distinguish sufficiently between entries.

When you have entered all the information it requires, ViewStore creates the appropriate files. If you catalogue your disc you should find:

- A file in directory D with the name you gave to the database (ie the database file).
- A file in directory F with the name you gave to the database (ie the format file).
- Files in directory I with the names you gave for the index files, if you created any indexes.

All these files will be initially empty .

Entering field formats

Firstly, load your new (empty) database by typing (in the command screen):

```
LOAD filename RETURN
```

where *filename* is the name you chose for your database.

As when experimenting with the Cars database, the command screen message should change to something like this :

```
Bytes free n  
Editing D.filename  
Format F.filename  
Screen mode 0
```

Now press **ESCAPE** to switch to the spreadsheet display. All fields are 18 characters wide, numbers are shown in place of field names and there is no data; but you can move the cursor from field to field in the normal way.

Before entering any data it is advisable to enter the field details. Press **RECORD FORMAT** to display the blank field details for the database, and then enter the details. Remember that there is one record in the record format for each field of your database. You should enter details in the order in which you want fields to be shown in the spreadsheet display.

The valid entries are summarised below.

Valid record format entries

Field name

Enter the name to be used for this field The name must not be more than 15 characters long.

Width

Enter the column width, ie the number of characters of each entry in this field to be displayed on the screen. The default value is eighteen. Width can be zero, enabling data to be hidden. Fields wider than the screen are truncated in the spreadsheet display and wrap around in the card display.

Type

Enter:

- A if entries in this field are to be alphanumeric; ViewStore allows any characters to be entered.
- T if entries in this field are to be textual; this is like alphanumeric data except that words and phrases can be searched for more easily
- N if entries in this field are to be numeric; ViewStore only allows numbers to be entered, including negative and decimal numbers. Numbers cannot be entered in exponential format, although ViewStore will display them in this format in reports if necessary.
- D if entries in this field are to be European dates; ViewStore only allows entries in the form D.M.Y , eg 10.4.85 = 10th April 1985, or 1.12.85 = 1st December 1985.
- M if entries in this field are to be American dates; ViewStore only allows entries in the form M.D.Y eg 4.10.85 = April 10th 1985 or 12.185 = December 1st 1985.

If **T y p e** is left blank, ViewStore expects entries to be alphanumeric.

Scroll

Enter **Y** if entries in this field can scroll Each entry in this field can then be up to 239 characters long. If **S c r o l l** is left blank, entries cannot scroll; ie they can be no longer than the column width. This should always be the case for numeric fields Dates, alphanumeric and textual data scroll by default, numbers do not.

Decimal places

If **T y p e** is **N** (ie entries in this field are numeric), enter the number of digits to be displayed after the decimal point. ViewStore will then accept entries in

the field with fewer digits, but not more, eg if `D e c i m a l p l a c e s` is 2 , ViewStore will accept 34 . 56 and 98237 but not 6 . 451. If left blank, ViewStore assumes zero decimal places.

Low Limit

If you wish, enter the lowest value or the earliest date to be accepted in this field. An alphabetic value can be given, eg if the value is `A N` , ViewStore will accept `A n t a r c t i c a` but not `A m e r i c a` . Case is ignored in comparing alphabetic values. If an entry contains more than one word, each word is compared with the limit; the entry will be accepted if any word is lower. If left blank, no lower limit is placed on entries.

High Limit

If you wish, enter the highest value or latest date to be accepted in this field. When looking at alphabetic values, ViewStore compares them as if they were arranged in a dictionary. Those that would come later in the dictionary are considered to be of higher value. For example, if `H i g h l i m i t` is `Z A N` , ViewStore will accept `z a m b i a` but not `z a n z i b a r` . If `H i g h L i m i t` is `Z A` , ViewStore will not accept either of these. Case is ignored in comparing alphabetic values. If an entry contains more than one word, each word is compared with the limit; the entry will be accepted if any word is higher. If left blank, no upper limit is placed on entries.

Index

Enter `Y` if an updateable index is to be maintained for this field; `R` if a read-only index can be created for this field. If `I n d e x` is left blank, no index can be used with this field.

Key width

If Index is `Y` or `R` , enter the number of characters of each entry in this field to be stored in the index or the number of characters to be used as the default when sorting.

Index name

If `I n d e x` is `Y` or `R`, enter the name of the file to be used to hold the index for this field. If `I n d e x` is `Y` , the filename should be the same as that given when using the `SETUP` utility.

Prompt

If you wish, enter a message which will be displayed on the third line of the data screen when the cursor is moved into this field. The message might give an expanded title or description of the field, or details of limits on entries, or a list of valid entries. The message can be up to 79 characters long, but in 40

column screen modes only the first 39 will be shown. If no message is entered, the field name will be displayed when the cursor is moved into the field.

Value List

If you wish, enter a list of the entries to be allowed in this field. If you give a list, only these entries (and no other entry) will be allowed. The entries in the list should be separated by commas. The wildcards ? (for a single character) and * (for any number of characters) can be used.

When you have entered all the necessary details for each field, return to the spreadsheet display to check that it is as you want. Move the cursor through each field to check the prompt messages. If you find anything that needs changing, return to the field details and edit them in the normal way.

Entering the database header details

When you are satisfied with the field details, press DATABASE HEADER to display the blank database header. To move from line to line press the arrow keys or the **RETURN** key.

Enter the details you require in the normal way . Details of each entry are given below.

When you have completed the details, return to the spreadsheet display to check that the database description is as you want. If it is not, return to the database header and edit the description.

Valid database header entries

Title

If you wish, enter a title or description to appear at the head of the data screen. Note that the number of characters shown is limited by the screen width.

Display

Enter C if the database should be presented in the card display when it is first loaded. If you enter S or leave the entry blank the database will be presented in the spreadsheet display.

Record size

This entry determines how much space ViewStore makes available on the disc for each record. If you enter + n (eg + 30), ViewStore makes available the space required for the record when it is first entered with n extra bytes (in the example, 30 extra bytes). n cannot be more than 512. If you enter just n (eg 600), ViewStore makes available n bytes for each record (in the example, 600 bytes). If **R e c o r d s i z e** is left blank, ViewStore makes

available 20 extra bytes for each record (ie this is equivalent to an entry of + 20). Remember that each character takes up one byte, but that extra bytes are required for each new field.

Capacity

If you wish, enter the largest number of records that ViewStore will load in to memory. If ViewStore tries to load more records than will fit in to memory, the computer will beep and display the `R e c o r d t o o b i g` error message. This problem can be solved by reducing the number in the `C a p a c i t y` field after which you should reload the database before returning to the data display. Alternatively, you can switch to a screen mode which gives you more memory. Note that reducing the capacity increases the number of disc operations but makes each one faster. The default setting is 50 records which is also the largest setting.

Index field

If you wish, enter the name of the field which should be used to index the display when the database is first loaded. If left blank, the data will be displayed in the order in which it was originally entered.

Screen mode

If you wish, enter the number of the screen mode that you want to use for your data display. If left blank, the current screen mode will be used.

Entering data

You are now ready to start building your database by entering records. For suggestions on how to speed up the entering of similar data, see Faster data entry in Chapter 8 Notes.

Note that updateable index files are created automatically as you enter data, but that read-only index files are not; you must give ViewStore the command to create them after the data has been entered (see Creating a new index file in the next chapter).

The creation of an updateable index involves ViewStore writing to disc after each entry in the corresponding field. This can substantially slow down the entry of data. If you are entering a large number of records and/or have a large number of updateable indexes, you may find it quicker to switch the indexes off (by setting `I n d e x` to `N`) whilst you are entering the data. When data entry is complete switch the indexes back on again (by setting `I n d e x` to `Y`) and then create the index files as described in the next chapter.

Preserving your data

As explained previously, ViewStore saves most of the data on to your disc automatically as it is entered and updates the disc completely when you press **ESCAPE** to return to the command screen. You do not have to tell it to save the data.

However, you are strongly advised to make copies of your database, format and index files on other discs, so that you can retrieve the information if your original files should for any reason be deleted or corrupted.

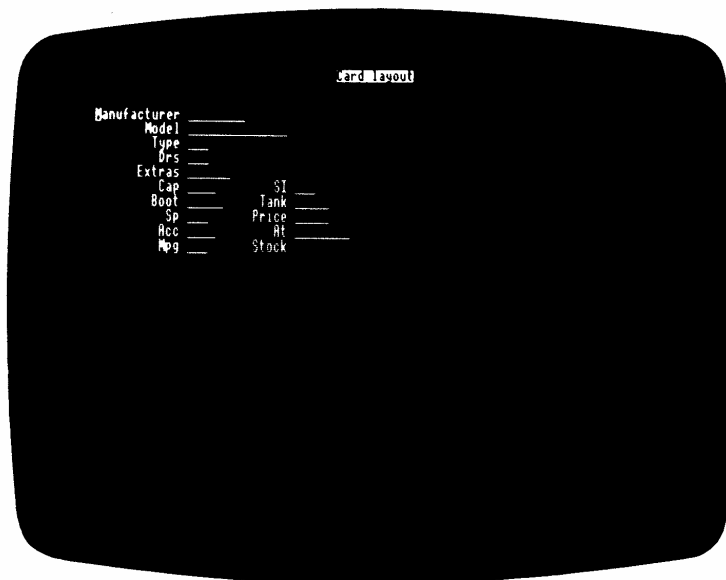
5 Altering a database

This chapter explains the ways in which a database can be altered once it has been created. The methods available to edit data, including the record format and database header, have been described earlier in this manual.

Changing the card layout

When you first switch to the card display for a new database, ViewStore lays out the fields in the order in which they were specified in the record format. In all probability this layout will not be very tidy or convenient, and this section shows you how you can change it.

To change the layout of the card display, press CARD LAYOUT.



ViewStore shows you a single record in its current card display layout, but with the area where each entry would normally be shown indicated by underline characters (or dashes in mode 7); the number of underline characters being equal to the field width. As in normal card display, field names are shown in full. If a field has zero width its name is shown without underlines. If a field has been given a width but no name, just the underlines are shown. Unused fields with no name or width are not shown.

To change the display you must first tell ViewStore which field you want to move. Using the arrow keys, move the cursor to a character within the field, which can be part of the field name or one of the associated underlines. Note that the cursor wraps around; that is, if you move it to the right when it is at the right-hand side of the screen it appears at the left hand side; if it is moved to the left of the screen it appears at the right-hand side. If you move the cursor to the bottom of the screen it will reappear at the top and vice versa.

When the cursor is positioned in the field you want to move, press **COPY**. The first character of the field name changes to an asterisk to show that the field is marked.

Then move the cursor to the new position where you want the field, and press **COPY** again to move the field. ViewStore moves the field to this new position, with the first character of the field name at the cursor position.

When a field is moved it must not overlap any existing field; there must be at least one blank space between it and the previous and next fields. Otherwise, ViewStore does not put the field where you have requested, but instead moves the field to the right and, if necessary, to the following lines until it finds a position where the field will fit; this may have to be at the very end of the display. Check that there is space for the field before moving it, and remember that the cursor position indicates the start of the field name. (The field can overlap its own existing position, enabling it to move just a few characters along the same line.)

When you are satisfied with the layout, press **DATA** to return to the card display to see what it looks like with data. Remember that, in card display, ViewStore fits as many whole records on to the screen as it can. If your new layout takes up more lines than before, there may be fewer records on the screen.

In some circumstances, it may be helpful to include an end of record marker in your card layout. This can be done by switching to the record format and adding, after all the other fields, an extra field with the name #, for example, and zero width. Then switch to the card layout, where the character will appear on its own, and move it to the required position.

Pressing **RETURN** when the cursor is in the last field of a record in the card display moves the cursor to the first field of the next record.

You should note that, if you set up your card layout in an 80 character wide screen mode but then switch to the card display whilst in a 40 character wide screen mode, ViewStore will carry out its own reorganisation of the card display layout in order to get all the information on the screen. Your layout will be lost and is not recovered when you return to the 80 character wide screen mode. To prevent this happening accidentally, make sure that you have specified the screen mode in which your data is to be displayed in the database header.

Using different formats

You will remember that, in ViewStore, data never has to be explicitly saved. Any additions and changes to the data are automatically saved to disc.

The same is true of the database format; that is, the record format, database header and card display layout. Any changes that are made are automatically saved to the format file shown in the command screen message.

In some cases, however, you may want to make changes to the format (such as adjusting column widths or the card layout), whilst not losing the current format. In this case, before you make the changes, tell ViewStore to save the current format.

To do this, return to the command screen and give the SF (for Save Format) command. Type:

SF *filename* RETURN

where *filename* is the name you want to give to the file which will contain the current format; for example:

S F FORMAT1 RETURN

The filename must be different to that of the current format file.

ViewStore saves all the current format details, including record format, database header and card layout, into this file. The file is saved in directory F using, where appropriate, the prefix for format files.

The new file that you have just created is an exact copy of the current format file. However, when you now alter the format of the database, ViewStore will change the format file correspondingly, whilst your new file contains an unchanged snapshot of the format that you want to preserve.

In this way you can build up many different formats applying to the same database, each of which has a different function. For example, you might prefer

to edit your data using very wide columns in which you can see the whole entry , whilst for normal scanning of the database you use narrow columns which enable all the fields to be on the screen at the same time. Another format might include a widely spaced card layout which allows only a couple of records on the screen at the same time but is easy to understand; another might use zero column widths to hide sensitive data. As you have seen already , the Cars database is provided with one format (also called Cars) for use by those interested in the details of the various models, and another format (Stock) for use by dealers.

To apply one of these different formats to your current database, return to the command screen and give the L F (for Load Format) command. Type:

LF *filename* RETURN

where *filename* is the name of the format file that you want to use. ViewStore looks for the file in directory F using, where appropriate, the prefix for format files.

Note that, as the command screen message shows, this file has now become your current format file. If you make any changes to the format it is this file that will be automatically altered.

As you have seen, when you initially load a database file ViewStore automatically loads the format file with the same name. If you want to use a different format file, include it in the LOAD command like this:

LOAD *databasefilename formatfilename* RETURN

For example :

LOAD cars stock RETURN

Note that ViewStore will allow you to use a format file created for one database with a completely different database. This can lead to very curious results. If you are using more than one database on the same disc, it can be helpful to include in the name of the format file an indication of the database to which it belongs.

Rebuilding a database

ViewStore enables you to rebuild a database using the CONVERT utility. This in fact creates a new database from the information in the existing database. In creating the new database you can:

- Select and/or re-order the fields to be used in the new database.
- Select and/or sort the records to be included in the new database.
- Make available more (or less) disc space for each record.
- Make available the disc space used by deleted records.

Firstly , load the existing database from which the new database is to be created.

If you want to select and/or sort the records to be included in the new database, do this next (see Chapter 3).

Then in the command screen, type:

```
UTILITY CONVERT RETURN
```

You will see the following prompts:

Use select file (N,Y)?

Tell ViewStore if you want it to use a select file of records.

Field *n*?

Type in the names of the fields that you want to include in the new database, in the order in which you want them to be shown.

When you have entered the names of all the fields you want, press **RETURN** in response to the *F i e l d n ?* prompt.

CONVERT will accept a wildcard specification to the *F i e l d n ?* prompt. For example, if you reply to the *F i e l d 1 ?* prompt with ***RETURN** and to the *F i e l d 2 ?* prompt with **RETURN** , then CONVERT will convert all the fields.

Then tell ViewStore the size of the space to be reserved for each record of the new database on the disc. You do this in the same way as when entering the database details. If you enter a number preceded by a + character it reserves the space needed for each record plus that number of bytes. If you enter a number on its own, ViewStore reserves exactly that number of bytes for each record; however , if a record requires more than that number of bytes it will be saved in full, but with no extra space.

Finally, enter the name you want to be given to the new database. It should be different from the current database or any other database on the disc. Unless you specify otherwise, the new database will be created in directory *D* using, where appropriate, the prefix for database files.

ViewStore then builds the new database in the way that you have requested, and saves it in a file with the name you specified. This can be a lengthy process, particularly for a large database. At the end it tells you how many records it has processed and the length of the longest record.

ViewStore does not create a format file for the new database. You can load the database using the format for the old database, but if the fields are not identical

you will then need to edit the format file accordingly . In particular if you have changed the order of the fields you must reorder the fields in the format file.

Creating a new index file

In most cases, you tell ViewStore which indexes you require when you are first setting up your database (see p.70), and the updateable index files are then created automatically by the system as you enter the data. However, read only index files must be created after the data has been entered.

You may also want to build or rebuild an updateable index after the data has been entered. This may be because you realise that you want to be able to index the data using a different field; or because the number of characters of each entry stored in the existing index does not distinguish sufficiently between them; or because you want to select the records to be included in the index; or because you want an index for a new expanded database (see the previous section) ; or because an existing index has become incorrect for some reason (see for example the section on Preserving your data).

Whatever the reason, the INDEX utility enables you to create a new index. The index can be either updateable or read-only .

First, load the database for which you want the index.

If you want to select the records to be included in the index, do this next (see Chapter 3). Only the selected records will be displayed when you CHANGE INDEX using the new index. It is inadvisable to build an index on a sorted field. If you wish to build an index on a field which has been sorted, sort the records on a different field.

Then in the command screen, type:

UTILITY INDEX RETURN

You will see the following prompts :

Use Select File (N,Y)?

Enter N . This will select all the records in the database. If you enter Y you will be asked for the name of the select file.

Field name?

Enter the name of the field for which you want the index. The `I n d e x` column of the report format for the field must contain Y or R , and the `I n d e x n a m e` column must contain a filename.

INDEX will accept a wildcard specification to the `Fieldname?` prompt. For example, if you reply to `Fieldname?` with `* RETURN`, then INDEX will

build indexes for all the fields with Y or R in the `I n d e x` column.

ViewStore then builds the new index as you have specified and saves it in a file with the name in `I n d e x n a m e`. The file will be created in directory `I` using, where appropriate, the prefix for index files. A dot is displayed as each record is processed, enabling you to monitor progress. Building a large index can be very timeconsuming.

If the new index is updateable (ie `I n d e x` in the record format is set to `y`), any further records that you add to the database subsequently will be included in the new index. If the index is `read.only` new records will not be added to the index; to include them you must rebuild the index as described above.

Note that it is possible to switch off an index temporarily by setting `I n d e x` to `N` or `R`, and then to switch it on again by returning `I n d e x` to `Y`. Any records entered whilst the index was switched off will not be displayed if the data is then displayed in the order of the index. If any records are deleted whilst the index is switched off, problems may occur when the index is used subsequently to order data.

6 Filing systems

The purpose of this chapter is not to give a complete guide to your filing system. This is provided in your filing system user guide. This chapter describes the major differences between the filing systems and points out how the differences affect ViewStore.

Acorn supply several filing systems as standard components for the BBC Microcomputer. They can be divided into three classes:

CFS

The Cassette Filing System provides a slow but inexpensive way of saving and loading data. To find information the cassette has to be wound to the correct place. This makes it unsuitable for large database programs and it cannot be used with ViewStore.

DFS

The standard Disc Filing System provides a fast and straightforward way of saving and accessing data. One advantage it has over the CFS is random access which enables information to be found very quickly wherever it is on the disc. The DFS can handle up to four disc surfaces which will be typically on two double-sided discs. There may be up to 31 files on each drive surface. These files can be organized into groups called directories and operations can be performed on whole directories.

When accessing a disc file, the filing system opens the file and stores information about it in its workspace in the computer. ViewStore requires the database file and all active updatable index files to be open when the database is being edited. The DFS allows only five files to be open at any time and so there is a limit of four indexes which can be updated at any time.

Hierarchical filing systems: ADFS, NFS

These have the advantages of the DFS and also a more developed directory structure. On the DFS a directory can contain only files but in a hierarchical filing system a directory can contain both files and directories. This nesting of directories can go to any level.

The ADFS allows ten files to be open at any time. Thus you may have nine updatable active indexes. The NFS allows five files to be open at any time and so you may have four updatable active indexes.

Filenames

DFS

A full filename on the DFS consists of the drive number followed by the directory name followed by the filename For example

.1.r.example

: 1. is the drive number. This can be between 0 and 3.

r . is the directory name. This can be a single letter, number or a punctuation character. It is best to use letters.

e x a m p l e is the filename. Filenames can be up to seven characters long and can consist of most of the characters on the keyboard. It is best to use just letters and numbers.

The default drive is 0 and the default directory is \$. Thus if you have not changed these, typing

*delete freda

will delete :0.\$freda. In other words the file named f r e d a in directory \$ on drive 0.

The default drive and directory can be changed with the ***DRIVE** and ***DIR** commands. Refer to the Disc Filing System User Guide.

ADFS

Filenames on the ADFS consist of the drive identifier followed by the pathname of directories followed by the filename For example:

:1.\$book.ViewStore.latest.info

:1. is the drive number. This can be 0,1,4,5,A,B,E,F depending on whether you are using floppy discs or a Winchester disc or a mixture of these.

\$.book.V i e w S t o r e.l a t e s t. is the pathname The highest level or root directory on each drive is \$. book is a directory within directory \$, ViewStore is a directory within book , latest is a directory within Viewstore Directory names can be up to ten characters and can consist of letters, numbers and some punctuation characters. It is best to use only letters and numbers.

info is the filename. Filenames can be up to ten characters and can consist of letters, numbers and some punctuation characters. It is best to use just letters and numbers.

For example If you are working on drive 1 and the current directory is:

`$.book.ViewStore` then typing

`*delete latest.info`

will delete `.1.$.book.ViewStore.latest.info`

NFS

Filenames on the NFS consist of the drive identifier followed by the pathname of directories followed by the filename. For example:

`:disc1.$.book.ViewStore.latest.info`

`: d i s c 1` is the name of the disc containing the fileserver.

`$` is the root directory of the fileserver.

`b o o k` is a user directory within `$`.

`V i e w S t o r e` is a directory within book's filespace. `l a t e s t` is a directory within `V i e w S t o r e`. Directory names can be up to ten characters long and can consist of letters, numbers and some punctuation characters It is best to use only letters and numbers.

`i n f o` is a file within `l a t e s t`

If you have signed on by typing `* l a m b o o k`, typing

`*delete ViewStore. latest.info`

will delete `:disc1.$.book.ViewStore. latest.info`

Use of directories in ViewStore

ViewStore uses different types of files to manage databases. To avoid confusion and to leave the database directory free of other file types, ViewStore stores each type of file in a separate directory.

Directories are allocated as follows:

D - database file

F - format file

I - index files.

S - sort files

U - utilities

R - report files

Therefore, if you type

load cars

ViewStore loads the database file `D . c a r s` using the format file `F . c a r s`. Report files in directory `R` use the `D` prefix.

Prefixes

The use of prefixes enables you to keep the different types of files on different disc surfaces and, with a hierarchical filing system, in different parts of your filesystem. On the DFS you can allocate maximum space to the database file without the need to change discs or supply drive numbers in all your commands. On the hierarchical filing systems you can perhaps keep the files which you never want to refer to directly in a sub-directory. On the NFS, rather than every user having his own set of utilities, they can be accessed in a common library.

ViewStore applies prefixes before its own directory settings. So if you have prefix `D` set to `d a t a b a s e`, and type:

Load cars

ViewStore tries to load `d a t a b a s e . d . c a r s` as the database file.

Prefixes are most conveniently set from exec files which can be run automatically when entering ViewStore. Once you have created the directories, built the files and written the exec file you need never worry about where the files are kept or use long pathnames to get to them.

DFS

A typical set of prefixes for the DFS is:

```
PREFIX d :0.  
PREFIX f :2.  
PREFIX i :2.  
PREFIX u : 1.  
PREFIX s :3.
```

This would be used on a double-sided dual disc system. Drive 0 has only the database on it, to allow maximum room for the database to grow. Drive 2, the other side of the database disc, has all the other files you want to be associated with this particular database. The utility files which you use for all your databases are on drive 1, the second disc. The sort files, which are temporary and which need not concern you when you are editing your database, are on drive 3 which can be used simply as a scratch disc for all your databases.

ADFS

A typical set of prefixes for the ADFS might be:

PREFIX d ViewStore.
PREFIX f ViewStore.
PREFIX i ViewStore.
PREFIX s ViewStore.
PREFIX u :1.\$lib

Note that on the ADFS the sub-directories must be created before use. This is done with the * C D I R command. If you have a dual drive system you might want to put the database on a different drive to the other files to give it maximum room.

NFS

A typical set of prefixes for the NFS is :

PREFIX d ViewStore.
PREFIX f ViewStore.
PREFIX i ViewStore.
PREFIX s ViewStore.
PREFIX u \$.lib.Store.

This has all the files connected with the database in sub-directories D , F , I and S of the directory V i e w S t o r e The utilities are in the library for everyone to use. Note that on the NFS the sub-directories must be created before use. This is done with the * C D I R command.

Running out of room

The following messages indicate that you have run out of room :

Can't extend
Disc full

together with the name of the file for which the error occurred. These errors occur because database and index files expand as new records are entered into the database. C a n ' t e x t e n d indicates that the expansion has caused one file to bump into another file on the disc; D i s c f u l l indicates that the expansion has caused the file to bump into the end of the disc.

DFS

The ideal arrangement is to store the database on a surface by itself allowing it maximum room for growth. Store the indexes on a different surface and rebuild them when necessary. On a single drive system you must store the database

and indexes on the same surface. Keep the utilities on a separate disc. Make sure that the database is the last file on the disc.

To expand an index:

- * D E L E T E the index.
- * C O M P A C T the disc surface.
- Rebuild the index using the INDEX utility.

To place a file at the end of the disc:

- * C O P Y the file to another disc
- * D E L E T E the file.
- * C O M P A C T the disc.
- * C O P Y the file back to the disc

Can't extend

Database you have a file on the disc at the end of the database. Place the database file at the end of the disc using the method described above.
Index expand the index using the method described above. If you have the database on the same surface as the index replace the database at the end of the disc, after expanding the index.

Disc full

Database the database has bumped up against the end of the disc. Delete as many files as you can from the disc and * C O M P A C T it. If you cannot delete any files, try deleting records from the database and use CONVERT to reclaim the disc space. If this does not work you will need a bigger disc system to store your database.

Index expand the index using the method described above. If you have the database on the same surface as the index replace the database at the end of the disc, after expanding the index.

ADFS

You should not get the C a n ' t e x t e n d message. If you do, follow the same procedure as with the DFS. On getting the D i s c f u l l message try to free some space on the disc by removing any unnecessary files and then * C O M P A C T the disc. If this does not work, keep the database on a separate disc to everything else if you have a dual drive system.

NFS

You should not get the C a n ' t e x t e n d message. If you do, follow the same procedure as with the DFS. On getting the D i s c f u l l message ask your Econet manager to free some space on the disc.

7 Error messages

This chapter contains details of ViewStore error messages, or error messages which have a special meaning in ViewStore. If an error message is displayed which is not described here, consult your computer's user guide or the user guide for your filing system.

Bad date

This is a date field, but the entry you have just made is not a valid date or is not of the correct type, eg you have entered an American date in a European date field or the wrong number of days for a month.

Bad directory

The directory you have specified in the filename is not allowed for this type of file; for example, you have tried to give a name for an index file using a directory other than I, or for a format file using a directory other than F. Check the settings of your prefixes.

Bad drive

ViewStore has tried to look for a file using an invalid drive number. for the DFS this must be in the range 0 to 3; for the ADFS it must be 0/A, 1/B, 4/E or 5/F The reason for this could be that you have included an invalid drive number in the filename (eg : 6 . D . C A R s). or that you have specified an invalid drive number in a prefix (eg : 6.), or that the prefix you have specified is incomplete (eg :2, causing ViewStore to look for : 2 D . C A R S).

Bad expression

Your report definition file contains an invalid expression Make sure your field names are valid and that you have delimiters around any ambiguous filenames.

Bad field

The currently loaded database has no field with this name. Have you specified the field name correctly?

Bad file

The file you are trying to load is of the wrong type. It may be a BASIC program, or a file created by another program such as VIEW or ViewSheet, or you may be trying to load a format file as data, or the file may be corrupted.

Bad FS

ViewStore cannot be used with cassettes. Type * **D I S C** to enter the DFS.

Bad macro

You have entered an invalid macro name; it must consist of two letters.

Bad mode

You must enter a positive number or zero as a parameter to **MODE**

Bad name

There is something wrong with the filename you have specified; for example, it has too many characters or is blank. Check the prefixes are what you intend. Remember you need a dot at the end of a prefix.

Bad pointer

The index or select file that you are trying to use has either become corrupted or refers to a different database. Rebuild the file, using **INDEX** for an index or **SELECT** for a select file.

Bad prefix

The letter you have used in a **P R E F I X** command is not valid or has been omitted. The valid letters are D , F, I, S, and U.

Bad record size

Record size must be a number, optionally preceded by +

Bad register

Your report definition file contains an invalid register, ie one that is not a single letter.

Bad selection

Your response to a **s e l e c t ?** prompt is incorrect. It must be in the form:

fieldname operator value

for example

manufacturer=Austin Rover

Bad string

You have tried to index your database using a particular field, but the **I n d e x n a m e** column of the record format contains no filename (or an invalid name) for that field.

Brackets

An expression in your report definition file has an unequal number of right and left brackets.

Can't extend

See Chapter 6 Filing systems

Channel

The filing system has detected an error. Return to the command screen, type N E W and reload the database.

Data screen onLy

You have tried a function key command which only works in the spreadsheet or card display.

Disc full

See Chapter 6 Filing systems

Divide by 0

In printing a report ViewStore has attempted to divide by zero.

Editing no file

You have tried to switch to the data screen but you have no database loaded in the computer's memory.

End

You are at either the beginning or end of your database and cannot move any further.

Escape

You have ended an operation by pressing **ESCAPE**

Field is not numeric

ViewSheet linking files can only be created using fields containing numeric data.

Field not found

ViewStore cannot find a field with the name you have given in the current database. This may also occur when you press **ESCAPE** after loading the database if the database header information is incorrect.

File exists

You have tried to create a file with the same name as an existing file. Either choose a different name for the new file, or delete the existing file (both the data and the format files).

File more than 10710 slots

You have tried to create a ViewSheet linking file which would have more than 10710 entries.

File not wide enough

Your ViewSheet linking file does not have enough columns to contain all the totals or subtotals on a single line.

Filename not found

ViewStore cannot find the file called `filename`. The problem may be that the file is not on the drive or in the directory indicated by the prefix for that file type. Use the `PREFIX` command to check which drive or directory this is. (If the file is in directory `U` ViewStore assumes that the file is a utility which you are keeping on a separate disc. It prompts you to insert the other disc. If you do not want to do this, press **ESCAPE**.) Note that ViewStore needs to access more files than the ones you have explicitly specified. For example, if you get the `Filename not found` message when loading a database file this may be because ViewStore cannot find the corresponding format file.

File not open

An error occurred when trying to open the file and it is now not open. Try pressing **ESCAPE** twice. If this does not work try reloading the database.

File open

The file you are trying to load was left open when you were last editing it, perhaps because you switched off or pressed **BREAK** whilst the data display was on the screen. Give the `NEW` command to close the file and then reload the database. This can also happen if you try to use `REPORT` whilst the report definition file is still loaded. Load the database on which the report will be based and repeat `REPORT`.

Fixed format

You cannot edit the format files supplied with ViewStore for creating report definition files.

Index: Bad filename

You have tried to order your data using a field for which the name in

the `I n d e x n a m e` column of the record format is invalid, perhaps because it is too long. Check your prefixes.

Index *fieldname*: File not open

You have tried to order your data using a field for which the index has not yet been created; or for which the name in the `I n d e x n a m e` column of the record format is invalid, perhaps because it is too long.

Index *filename*: Can't extend

This means that no more records can be included in the index file named because it is bumping up against the next file on your disc. This message is usually only obtained if you are using ViewStore with the DFS.

The record you have just entered will not be included in the index. To include this record (and any additional records) you must move the index file so that it is the last file on your disc.

See Chapter 6 Filing systems

Index: *filename* not found

The index is switched on for this field (ie in the field format details for this field there is a Y against `I n d e x`), but ViewStore cannot find a corresponding index file (ie a file in directory I with the same name as the field or , if given, with the name entered against `I n d e x n a m e` in the field format details for the field)

Remember that ViewStore looks for the file on the default Index drive ; it will still give the message if the correct file is on your disc but on a different drive.

Key too long

The total sort key you have specified is more than 250 characters.

Keysize should be ≥ 1 and ≤ 105

The number of characters of each entry that you have asked ViewStore to use in creating an index for a field is greater than 105 or less than 1.

Limit error

The data you have just tried to enter is greater than the `H i g h l i m i t` value or less than the `L o w l i m i t` value specified for this field in the record format, as shown on the third line of the screen.

Maximum is 4/9 indexes

You have tried to set up more indexes than are allowed with your disc filing system.

Memory full

The computer does not have enough memory for the operation you are trying to carry out. In most cases you will be able to make more memory available by switching to a different screen mode (eg 6 or 7) using the `MODE` command.

Then after carrying out the operation you can switch back to your required screen mode.

Mistake

ViewStore does not recognise the command that you have just typed. Have you typed it correctly? Is it a ViewStore command?

No data

Something has destroyed the data in the computer's memory (for example, copying or compacting a disc). Reload the database.

No database Loaded

You have attempted to use a utility when no database is loaded

No end marker

The file you are trying to load does not include an end marker. It may be a file of an incorrect type (see `Badfile`), or something may have gone wrong when the file was last being edited. You will need to use your backup copy of the file.

No fields

You have tried to change to the card display or to card layout display when, if you did, no fields would be displayed on the screen. This is because neither a name nor a width has been given to any field in the database; edit the record format accordingly and try again.

No fields found

The printer width you have specified in `REPORT` is less than the width of the first field, so that no fields will be printed. Give a larger printer width, or make the field widths smaller.

No index

You have tried to create an index for, or order your database using, a field for which `Index` in the record format is not `Y` or `R`.

No index field

You have pressed `CHANGE INDEX` and there is no index field.

No records found

No records have been found which match the criteria you have entered.

Not enough fields

There are more formats than fields in the field list in the REPORT utility.

Not numeric

This is a numeric field, and you have tried to enter non-numeric data (eg text or a date)

Overflow

The number you have just entered (or that has been calculated when evaluating an expression) is outside the range that ViewStore can handle.

Read error

ViewStore cannot read the file you have named. You will need to use your backup copy of the file.

Record too big

The next record in the file is too big to fit into the computer's remaining memory. Either switch to a screen mode which gives you more room or reduce the number of records in the `C a p a c i t y` field in the database header. After doing this, you should reload the database before switching to the data screen.

Sheet display

have tried to edit the card layout when the spreadsheet display is currently selected. Switch to the card display before pressing `CARD LAYOUT`.

Stack overflow

The expression currently being evaluated is too complicated for ViewStore to handle and must be simplified. This can also occur with a syntax error in a REPORT expression, or if you try to build an index on a large sorted field.

Too Long

Either the total number of characters you have asked ViewStore to use in an operation is greater than 250 or the filename is more than 12 characters.

Too many files

You have exceeded the number of updateable index files which can be in operation at one time, and must switch one (or more) off by setting `I n d e x` in

the field format to N or R ViewStore can operate up to 4 updateable indexes at one time if you are using the DFS, 9 if you are using other filing systems.

Too many places

You have tried to enter a number with more decimal places than are specified in the `D e c i m a l p l a c e s` column of the record format.

Type mismatch

In the report utility you have attempted to perform a numeric operation on a non-numeric field.

Value not in List

The data you have just tried to enter does not match any of the values in the `V a l u e l i s t` specified for this field in the record format, as shown on the third line of the screen.

8 Notes

This chapter describes some of the facilities of the BBC Microcomputer and shows how to use them with ViewStore. For further information read the BBC Microcomputer User Guide and the Advanced User Guide for the BBC Microcomputer.

Screen colours

If you use a colour monitor or television set you may wish to take advantage of alternative screen colours.

Both text and background colours can be changed in screen modes 0 to 6. The colours available are:

- 0 - black
- 1 - red
- 2 - green
- 3 - yellow
- 4 - blue
- 5 - magenta
- 6 - cyan
- 7 - white

These may be used as follows in the command screen.

For green text:

Hold down **CTRL** and press s Press: 7 2 0 0 0

For white text on a blue background:

Hold down **CTRL** and press S Press 0 4 0 0 0

For black text on a white background:

Hold down **CTRL** and press S Press 0 7 0 0 0

Hold down **CTRL** and press S Press 7 0 0 0 0

The first digit indicates whether it is text or background colour that is to be changed. 0 means background and 7 means text. The second digit

indicates the colour, as in the table above.

To return to the standard white text on a black background hold down **CTRL** and press **T**.

***FX calls**

We have mentioned some of these already. For comprehensive lists of *FX calls refer to the BBC Microcomputer System User Guide and to the Advanced User Guide for the BBC Microcomputer. You may find the following calls useful with ViewStore.

Throughout this section, *n* is used to represent a number which you type in.

- *FX5,*n* To specify the printer type.
- n* = 0 Direct all print output to the screen.
 n = 1 Send output to a parallel printer.
 n = 2 Send output to a serial printer.
 n = 4 Send output to the Econet printer server
- *FX6,*n* To determine the character that will be suppressed by the printer driver routines. The default setting is * F X 6 , 10 which supresses linefeed characters. Ifyou wish linefeeds to reach the printer you should use * F X 6 to suppress another character you do not need * F X6, 255 for example.
- *FX8,*n* Sets the transmission rate to a serial printer.
- n*=1 75 baud
 n=2 150 baud
 n=3 300 baud
 n=4 1200 baud
 n=5 2400 baud
 n=6 4800 baud
 n=7 9600 baud
 n=8 19200 baud
- *FX11, *n* Sets the auto-repeat delay period in hundredths of a second. For example ifyou type * F X 1 1 , 100 then keys will start to repeat after being pressed for a second.

- *FX12,*n* Sets the auto-repeat period. For example if you type
* F X 12,3 keys will repeat every three-hundredths of a
second once the delay period has expired.
- *FX12,0 Resets the default values of both FX 11 and FX 12.
- *FX 202,48 Switches CAPS LOCK off. This can be useful when
initializing ViewStore from an exec file.

Faster data entry

If you frequently make the same entry in ViewStore, such as
A u s t i n R o v e r in the M a n u f a c t u r e r field, there is a way of entering
the same data without having to type it over and over again.

First, program a function key with the text that you want to enter frequently.
To do this return to the command screen and type, for example:

*KEY0 Austin Rover **RETURN**

which will program key f 0 with the text A u s t i n R o v e r . Then, still in the
command screen, type:

*FX228,1 **RETURN**

Now return to the data screen and move the cursor to the M a n u f a c t u r e r
field of a new record. Hold down **CTRL** and **SHIFT** together. Whilst holding
down **CTRL** and **SHIFT** press and release **f0**. A u s t i n R o v e r will appear on
the screen at the cursor position.

When programming function keys you can include the characters |M . This pair
of characters is equivalent to pressing the **RETURN** key, which in the
ViewStore data screen moves the cursor to the next field. So a key programmed
like this:

*KEY0 Austin Rover|M **RETURN**

will enter A u s t i n R o v e r and then move the cursor to the next field. A key
programmed like this :

*KEY0 Austin Rover|MMini |MS|M2|M|M5|M **RETURN**

will make entries in a series of fields, including two blank entries, like this:



BREAK

Pressing the **BREAK** key when in ViewStore's data screen can corrupt your database. You should never need to press **BREAK** as pressing **ESCAPE** will always take you back to the command screen.

If you have pressed **BREAK** by accident when in the data screen your database may be in an indeterminate state. If you do the following the database should be recovered and be written safely to disc.

- Make sure you are in the correct filing system. For example if you are using ViewStore on the DFS type ***DISC RETURN**. You should do this before entering the data screen.
- Press **ESCAPE** to enter the data screen.
- Press **ESCAPE** to return to the command screen.

This procedure should write the whole database from memory to the filing system and close all the files. Ensure that it has done this by reloading the database. If you cannot access records in your database or you get irrecoverable error messages then you will have to use your backup.

Pressing **CTRL BREAK**, turning the computer off, or a power failure when in the data screen will almost certainly corrupt your database. Neglect backups at your peril.

Extra memory

The BBC Microcomputer Model B has eight screen modes which take up varying amounts of memory. Generally the most useful screen modes in ViewStore are 0 and 3 which provide eighty columns of data. However these screen modes only leave up to about 10000 bytes for holding ViewStore records depending on which filing systems and service ROMs you have in your computer. Whilst this is generally all you will need because of ViewStore's efficient use of memory you may consider using shadow memory or a 6502 Second Processor. Shadow memory boards offer about 25000 bytes in any screen mode and the 6502 Second Processor offers about 30000 bytes in any screen mode. On the 6502 Second Processor, the utilities do not use up record space. ViewStore will also run faster on the 6502 Second Processor.

Extra memory has two advantages:

- It increases the maximum record size since this is limited to either memory size or about 60000 bytes.
- It speeds up movement around the database and sorting records since ViewStore can hold more records in memory and will need to access the disc less frequently.

Exec files

When entering ViewStore you may find that you always use the same commands to initialise the system before editing your database. Exec files allow you to enter the commands automatically.

An exec file is simply a file of commands stored on disc which can be called from the command screen at any time. The commands are then read from disc and entered as if you were typing them in.

To create an exec file you can either use the VIEW word processor or the *BUILD command on the DFS. In the command screen type *BUILD !BOOT **RETURN** and enter your standard sequence of commands in response to the numbers supplied. An example exec file might be

- 1 *STORE enter ViewStore
- 2 *TV255 1 turn interlace off and move display down
- 3 MODE 3 change to screen mode 3
- 4 *FX202 48 turn off CAPS LOCK
- 5 PREFIX D :0. data on drive 0
- 6 PREFIX I :2. indexes on drive 2
- 7 PREFIX U :1. utility disc on drive 1
- 8 PREFIX S :2. select files on drive 2
- 9 PREFIX F :2. format files on drive 2

Pressing **ESCAPE** closes the file and saves it on to your disc with the name **!BOOT**. You can run the exec file by entering ***EXEC !BOOT** in the command screen.

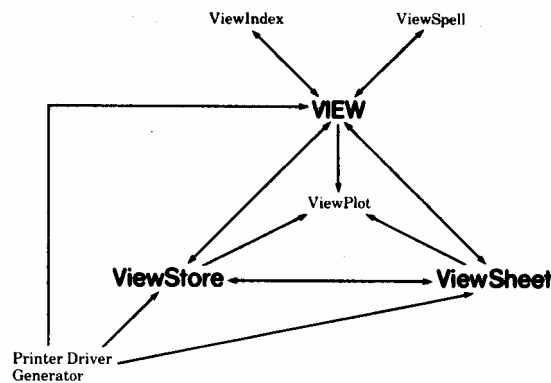
The **!BOOT** file is a special file on the DFS When pressing **SHIFT BREAK** (ie an auto-boot) you can make the filing system automatically select the **!BOOT** file. This is done as follows:

- Put the disc containing the **!BOOT** file into the disc drive and type ***OPT 4,3 RETURN**. This tells the disc that the **!BOOT** file is an exec file.
- When you want to run the exec file put the disc in the drive, hold down **SHIFT** and, while holding down **SHIFT**, press and release **BREAK**.

This will enable you to print standard reports by simply pressing a function key.

The View range

ViewStore is one of the View range of products. The whole View range has been designed for use with the 6502 Second Processor and a range of Acorn filing systems More importantly, the range is integrated; data can be passed from one program to the others. as shown in the diagram below.



Other products in the range include:

The VIEW word processor

Facilities include on-screen formatting, adjustable margins, centred headers and footers, automatic page numbers, operation in any screen mode and automatic mail-merge.

The ViewSheet spreadsheet program

Facilities include windows allowing flexible displays and print-outs, column and row headings, automatic bar-charts, operation in any screen mode, trigonometric functions, protection, insert/delete rows and columns and the linking together of spreadsheets.

The ViewSpell spelling checker

Facilities include fast checking of text files, a large standard master dictionary, user dictionaries, and flexible searching of all dictionaries. Essential for crossword and Scrabble addicts!

The ViewPlot graph generator

Facilities include the generation of bar charts, pie charts and line graphs from spreadsheets, databases and keyboard, multi-coloured charts and multiple charts on screen.

The ViewIndex index generator

Enables generation of automatic indexes for books and long reports.

The Printer Driver Generator

Printer drivers enable the use of printer specific facilities such as underlining, bold characters and alternative fonts. The Printer Driver Generator asks for information about your printer and then generates a driver which you can store on cassette or disc for permanent use.

9 Specification

Summary

ViewStore is a general purpose database program for use with the BBC Microcomputer Model B and related microcomputers. Records and fields are of variable size and records can contain a varying number of fields. The file size is limited only by the size of the disc used with the microcomputer. The record size is limited only by the memory available.

ViewStore can access one database file at a time with up to nine indexes. Extensive data entry checks and display formats are provided. Subsets of records can be created by using complex criteria when selecting.

A range of utilities allows comprehensive reports or labels to be produced as well as data exchange with the VIEW word processor and the ViewSheet spreadsheet. The sorting facilities enable small and large files to be sorted efficiently on many fields.

ViewStore operates in any of the BBC Microcomputer's screen modes.

Specification

Maximum file size

4,294,967,296 bytes (4,096 megabytes)

In practice file size is limited by disc size as follows:

40 track DFS disc - 100K
80 track DFS disc - 200K
Double sided ADFS disc - 720K
Winchester disc - 10 megabytes
Winchester disc 30 megabytes and upwards

Fixed file size?

No. File size is variable

Maximum record size

60,706 bytes

In practice the record size is limited by the memory available as follows:

With the BBC Microcomputer Model B :

Mode 0 - 5k
Mode 3 - 9k
Mode 7 - 25k

Other microcomputers may have different memory capacities.

Maximum number of fields
254

Maximum field size
239

Maximum decimal places
9

Maximum significant figures
10

Maximum key length (index)
105

Maximum key length (sort)
250

Number of sort fields
Unlimited

Data types

These are as follows:

- Alphanumeric
- Text Date (D/M/Y)
- American date (M/D/Y)
- Numeric

Maximum number of indexes 9

In practice the maximum number of indexes is limited by the filing system in use as follows:

DFS - 4
ADFS - 9
NFS - 4

Number of indexes updated

Up to 9

Link to ASCII files?

Yes

Record structure

Variable structure; variable or fixed size records; variable size fields. Utility to re-order fields and expand space available for records.

Number of database files

1

Data validation

Low and high limits; list of permissible values including wildcards.

Wildcards

Wildcards for specifying field names and for selecting field values.

Display

80 Or 40 column. Standard spreadsheet-like display or user-definable card display.

Selection

AND and OR operators; wildcards and operators = , < > , > , < , >= , <= Complex expressions using brackets. Wildcards on both sides of a comparison.

Selected records can either be listed on the screen or stored in a file for later use.

Labels

Variable label format.

Report

Simple and user-defined reports. Subtotalling and totalling; pages; headers; calculations in the report. Comments entered at report printing time.

Word processor link

Integration with the VIEW word processor. Can produce macros for standard letters; report output can be transferred; or database files read directly by VIEW.

Spreadsheet link

Integration with the ViewSheet spreadsheet. Can transfer numerical data directly, or subtotals and totals from reports can be transferred.

Tailoring

User-defined screen formats and data checks. User defined soft keys Exec files for extended utility operations.

ViewStore file format

The ViewStore file format has been designed to be as simple as possible to minimise structural problems caused by the operating environment of microcomputers and to enable data to be transferred easily both to and from ViewStore.

Database files

The database file contains records; the ends of the records are delimited by an end-of-data marker. The total size of the data can be less than the size of the whole file; from the end-of-data marker to the end of the file is filled with pad characters.

The records contain fields; each record is delimited with an end-of-record marker. Each record can contain expansion space. This is indicated by a collection of spacer characters; the spacer characters come immediately before the end-of-record marker and after all the other characters in the record, specifically after the last end-of-field marker.

The fields contain data; each field is delimited with an end-of-field marker. The data is represented with ASCII characters in the range & 20 - & 7 F .

Deleted records are represented by a block of deleted markers followed by an end-of-record marker.

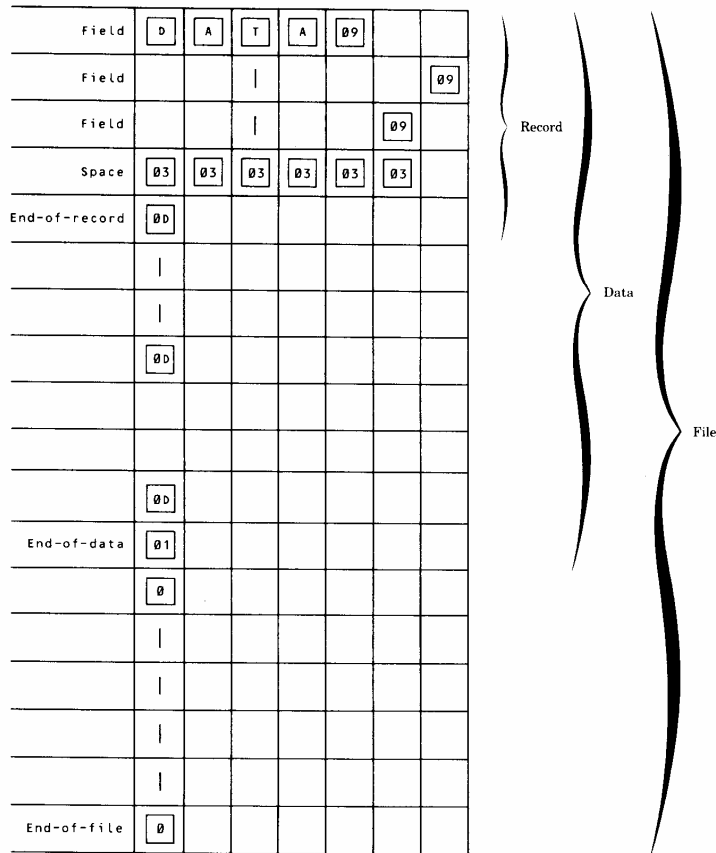
Format files

Format files follow the same rules as database files. Each record contains the information relating to a field in the corresponding database file.

Report files

Report files follow the same rules as database files. Each record contains one line of information for the REPORT utility.

ViewStore file formats



Markers

The value of the marker characters is as follows:

End-of-data &01
End-of-record - & 0D
End-of-field - &09
Spacer character & 03
Deleted character - & 02
Pad character - &00

INDEX

- Adding new data 26-27
- Adding value to register 51-52
- ADFS 75,89,90,93,94
- Alignment print (N Y)? 44
- Alphabetic values, comparing 78
- Alphanumeric fields 36,37 71,77
- Altering a database 82-88
- Altering information 19
- American dates 77
- AND 35,36
- Angle brackets (in macros) 41
- Appears after last field (Y N)? 60
- Appears after last record (Y N)? 59
- Appears before first record (Y N)? 59
- Appears before first field (Y N)? 60
- Arithmetic expressions (in field lists) 50
- Arrow keys 12,16,83
- ASCII files 112
- Auto-boot 108
- Auto-repeat 12
- Auto-repeat delay 104
- Auto-repeat period 105

- b (in report formats) 49,50
- BASIC 8
- *BASIC 8
- Baud rate codes 30
- BEGINNING OF FIELD 13,26
- display width 20
- Blank prefixes 74
- Blanks (in reports) 49
- Bleep 26
- Bold print see Highlight codes
- !BOOT 107,108
- Booting the system 108
- Brackets (in selection conditions) 36

- BREAK 106
- *BUILD 65,107
- Building an index 87
- Bytes free 7,8,10

- C (comment) lines (in report file) 56
- Can't extend 72,75,93,94
- Capacity 21
- Capacity (in database header) 80
- CAPS LOCK, switching off 105
- Card display 15
- Card display, changing 82-84
- Card layout 17
- Card layout, changing 82-84
- CARD LAYOUT 82
- Cars database 8
- Cars database, labels from 44
- Cassette Filing System 89
- CFS 89
- CHANGE DISPLAY 15,16
- CHANGE INDEX 87
- Changing a database 82-88
- Changing the card layout 82-84
- Changing the index 23,24
- Character cursor 13
- Characters, suppressing 104
- Characters between each label? 43

- Colours 103
- Combining selection conditions 35
- Blank
- Command screen 10,28
- Commands, giving 6
- Commands, typing 9
- Comment (in report file) 50,53,54
- *COMPACT 94
- Comparing alphabetic values 78
- Comparing letters 35
- Comparing selection values 36

Conventions 6
 CONVERT utility 39,86,86,94
 Converting data from other databases see IMPORT
 Converting VIEW files to ViewStore format 64
 COPY key 83
 *COPY 94
 CP/M format files 67
 Creating a database 68-81
 Creating a new database from an old one 86-87
 Creating an index file 87-88
 Creating database header 79-80
 Creating exec files 66
 Creating record formats 76-79
 CTRL BREAK 28
 CTRL BREAK 107
 CTRL ↑ 14,26
 CTRL → 13,14,26
 CTRL ← 13,26
 Cursor, character 13
 Cursor, field 12
 CURSOR LOCK 14
 Cursors, moving the 12,14,16

DATA 19,20,83
 Data, listing 46-47
 Data, preserving 81
 Data, transferring from other Deleted records (in IMPORT) 61
 databases see IMPORT
 Data entry, speeding up 80,106
 Data screen 10
 Data types 111
 Data validation 112
 Database, altering 82-88
 Database, creating 68-81
 Database, defining 73-80
 Database, planning 68-73
 Database, rebuilding 85-87
 Database (the Acornsoft program), converting data from 61

Database file 9
 Database file, examining 10
 Database file, loading a 8
 Database files, format of 113
 Database files, number of 112
 Database header 17,20-21
 DATABASE HEADER 20,79
 Database header details, entering 79-80
 Database name, choosing 73
 Database title, choosing 72
 Date fields 38,77
 Date? 64,66
 dBASE II, converting data from 62
 Decimalplaces, maximum 111
 Decimal places (in record format) 18,77
 Decimalpoint (in report format) 49,60
 Default entries (in record format) 19
 Default values (in ViewStore prompts) 31
 Defining a new database 73-80
 Defining a text line 48
 Definition file see Report file
 DELETE key 20,26
 *DELETE 94
 DELETE CHARACTER 13,19,26
 DELETE END OF FIELD 13,20,26
 Deleted character 116
 Deleted records (in IMPORT) 61
 Delimiters 36,60
 Destination file 68
 DFS 2
 DFS 2,89,90,92,93
 *DIR 90
 Directories 2,91
 Directories, suggested organisation of 72
 Disc Filing System see DFS
 Disc full 93,94
 Disc space, reserving 71-72

Disc space, running out of 93-94
 Display (in database header) 21,79
 *DRIVE 74,90

 e.ex1 33,35
 e.ex2 47,65,66
 Editing 9
 Editing data 25-26
 Editing exec files 65
 Editing No File 7 8
 Enboldening see Highlight codes
 End message 14
 END OF FIELD 13,14,26
 End of file marker 60
 End of record marker 83
 End-of.data marker 115
 End-of-field marker 115
 End-of-record marker 115
 Entering data 80
 Entering database header 79-80
 Entering record formats 76-79
 Entering ViewStore 7
 Error messages 95-102
 ESCAPE 10,28
 European dates 77
 Evaluating expressions 50
 Examining a database file 10
 Example files 47
 *EXEC 47,65,66,67,108
 Exec files 107
 Exec files, using with utilities 65-67
 Exponential notation 49
 Extra memory 107

 F REPORT 48
 Faster data entry 105
 Field 3, 1 2
 Field 1? 40,41,86
 Field cursor 12
 Field formats, entering 76-79
 Field list 48-50
 Field name (in record format) 18,77

 Field name, length of 70
 Field name? 87
 Field names 34,35,49
 Field names, choosing 70
 Field names, listing 31
 Field number 10
 Field order 69
 Field separator 59
 Field size, maximum 111
 Field type 36,37
 Field types 77
 Field width, choosing 69-70
 File 3
 File format 113-115
 File organisation 72
 File size, maximum 110
 Filename of database? 75
 Filename? 75
 Filenames 90-91
 Filing systems 89-94
 Finishing a ViewStore session 27
 Format (in report file) 48,49
 Format file 9,10,17,21-23,86
 Format filename (database filename)? 75
 Format files, format of 114
 Format files, loading 85
 Format files, saving 84
 Format of ViewStore files 113-115
 Function key card 5,6
 Function keys 5
 Function keys, programming 67,105
 *FX calls 30,104,105

 Giving commands 6

 Half1 (in report file) 48,53,54
 Half2 (in report file) 48
 Header see Database header
 Header line (in report file) 53
 Hierarcical filing systems 89
 High limit (in record format) 18,78

Highlight codes 54-55
 Highlighting 29
 How many bytes to reserve? 75
 How many totals across? 56

 IMPORT 57-63
 Index, changing 23,24
 Index, switching off 88
 Index (in record format) 18,78,80
 INDEX FIELD 23,24
 Index field (in database header) 21,80
 Index file 23,24
 Index file, creating 87-88
 Index name (in record format) 19,23,78
 INDEX utility 87
 Index1 Filename? 75
 Indexed by 23,24
 Indexes 70-71
 Indexes, maximum number of 111
 Indexes, number updated 112
 Indexing a database 23,24
 INSERT CHARACTER 13,19,25,26
 Insert data disc 31
 Insert utility disc 31
 Installing the ViewStore ROM 5
 Is the data reversed (N,Y)? 60

 *KEY 67
 Key length (index), maximum 111
 Key length (sort), maximum 111
 Key width (in record format) 19,23,78
 Key width, choosing 71
 Keysize? 76

 L (command) see LOAD
 L (on data display) 14
 Label height (lines)? 43
 LABEL utility 32,42,43,63
 Labels 42-25
 Layout, changing 82 84
 Leading character to skip 60

 Leaving ViewStore 8
 Length of field n 60
 LF (command) 22,85
 Limits on entries 69
 Line 1? 43
 Line types (in report file) 52-54
 Linefeed, suppressing 104
 LINK utility 41
 Linkingfilename? 56
 Linking files see ViewSheet
 linkingfiles
 LIST 33
 Listing data 45-47
 LOAD 9,23,76,85
 Loading a database 8,23
 Loading a format file 22,23,85
 LOCATE 25
 Locating a record 24-25
 Locked movement 14
 Low limit (in record format) 18,78

 MACRO utility 40
 Macro parameters see VIEW
 macro parameters
 Margin (in report file) 53,54
 Markers 115
 Maximum decimal places 111
 Maximum field size 111
 Maximum file size 110
 Maximum key length (sort) 111
 Maximum key length (index) 111
 Maximum number of fields 111
 Maximum number of indexes 111
 Maximum record size 110
 Maximum significant figures 111
 Memory 8
 Memory, providing extra 107
 Memory full 39,45,55,100
 Messages, error 95- 102
 Missing pages (in printing) 56
 MODE 8

Name of database 73
 Name of report? 47
 NFS 89,91,93,94
 Number of database files 112
 Number of fields 60
 Number of fields, maximum 111
 Number of indexes 75
 Number of indexes, maximum 111
 Number of indexes updated 112
 Number of labels across the page? 43
 Number of records 75
 Number of sort fields 111
 Numeric fields 77

 Operators (in arithmetic expressions) 50
 Operators, selection 34
 *OPT 108
 OR 35,36
 Order of fields 69
 Organisation of files 72
 Other databases, transferring data from see IMPORT
 Output filename? 40,42

 P (page register) 51
 Pad character 115
 Page.. 46,47
 Page1.. 56
 Page ejects 50
 Page length (66)? 46
 Page length (in report file) 53,54
 Page register 51
 Parameters see VIEW
 macro parameters
 Passing text to VIEW 64,65
 Pathname 90,91
 Planning your database 68-73
 Position in file where data starts 59
 PREFIX 73,74
 Prefixes 73-74,92-93
 Preparing for printing 29

 Preserving your data 81
 PRINTER 30
 Printer, transmission rate to 104
 Printer driver 29,30
 Printer Driver Generator 109
 Printer type 104
 Printer width (80)? 46
 Printer width (80)? 34
 Printing, preparing for 29
 Printing at nth character of field 50
 Printing blanks (in reports) 49
 Printing codes 30
 Printing labels 42-45
 Printing register values 51,52
 Printing reports 45-57
 Printing subtotal lines 53
 Printing total lines 53
 Programming function keys 67 105
 Prompt (in record format) 19,78
 Prompts (when printing reports) 56

 R (record register) 51
 R.filename 48
 READ 64
 Read-only index 70,80,87,88
 Reading 10,14
 Rebuilding a database 85-87
 Rebuilding an index 87
 Record 3,12
 Record format 17-20
 Record format, entering 76-79
 RECORD FORMAT 17,76
 Record line (in report file) 53
 Record register 51
 Record separator 59
 Record size, maximum 110
 Record size (in database header) 21,79
 Record structure 112
 Record too big 80
 Reference card 5,9
 Registers 51-52
 REPORT utility 32,42,45,55,63,65

Report definition file see Report file
 Report file, setting up 47
 Report file, structure of 47
 Report filename? 55
 Report files, format of 114
 Report format file see Report file
 Reports 45-57
 Reprint first labels again (N,Y)? 44
 Requesting a report 55
 Restarting printing 56
 RETURN 14,16
 ROM, installing the ViewStore 5
 ROM programs 8
 Running out of room 93-94

S.database 39
 S.SRTINT 39
 Saving format files 84
 Screen colours 103
 Screen flicker 8
 Screen mode 7,8,21
 Screen mode (in database header) 80
 Screen or Printer (S,P)? 33,43,46,55
 Scroll (in record format) 18,77
 Second Processor 107
 Select criteria? 34,35
 Select file 31,32
 SELECT utility 31,32,33,45
 Selecting records 32-37
 Selection conditions, combining 35
 Selection criteria 34-36,37
 Selection values, comparing 36
 Send totals to linkingfile? 55
 Serial printer, transmission rate to 104
 Set up database or report (D,R)? 47 74
 Setting up a database 73-80
 Setting up prefixes 73-74
 SETUP utility 47,74,78
 SF (command) 84
 Shadow memory 107
 *SHEET 8,42

SHIFT BREAK 108
 SHIFT ↓ 14
 SHIFT ↑ 14
 Significant figures, maximum 111
 Single sheets (N,Y)? 46,56
 Sort field? 37,38,39
 Sort fields, number of 111
 Sorting conditions 38
 Sorting records 32,37-39
 Source file 58
 Space 26,27
 Space, reserving 71-72
 Space, running out of 93-94
 Spacer character 115
 *SPOOL 63,64
 Spreadsheet display 12
 Starting IMPORT 58
 *STORE 7

Structure of report file 47
 Subtotal value (of register) 51
 Subtotal field? 56
 Subtotal line (in report file) 53
 Subtotal values, sending to linkingfile 56,57
 Subtotalling (in reports) 53,56
 Summary 110
 Suppressing characters 104
 Switching off 28
 Switching off an index 88

Television, using ViewStore with 8
 Text line, defining 48
 Textual field 71
 Textual fields 36,37
 Textual fields 77
 Title 21
 Title, choosing 72
 Title (in database header) 79
 Total line (in report file) 53
 Total value (of register) 51
 Total values sending to linking file 56,57

Trailing character to skip 60
 Transmission rate 104
 *TV0,1 8
 Two letter macro? 40
 Type (in record format) 18,77
 Type (in report file) 48,52,54

 U (on data display) 14
 Underlining see Highlight codes
 Updateable index 70,80,87,88
 Use of directories in ViewStore 91
 Use report format file (N,Y)? 46,55
 Use select file (N,Y)? 31,40,41,
 43,45,55,86,87
 Using * SPOOL 63-64
 Using exec files with
 utilities 65-67
 Using IMPORT 58-61
 Using utilities 31-32
 Utilities 29,31

 V.VSn 42
 Value list (in record format) 19,79
 Value? 25
 VIEW 109
 VIEW, building exec files using 107
 VIEW, passing text to 64-65
 VIEW, transferring to 8
 VIEW, using utility output with 63
 VIEW, using with exec file 65
 VIEW files, converting to ViewStore
 format 64
 VIEW macro parameters 39-41
 ViewIndex 109
 ViewPlot 109
 ViewSheet 109
 ViewSheet, transferring to 8
 ViewSheet linking files 41-42
 ViewSheet linking files (from report) 56,57
 ViewSpell 109
 ViewStore, entering 7

 ViewStore, finishing a session 27
 ViewStore, leaving 8
 ViewStore file format 113-115
 ViewStore record size 61

 Width (in record format) 18,77
 Width of fields, choosing 69-70
 Width of label? 43
 Wildcard (in IMPORT) 59
 Wildcards 23-24,34,35,50,
 79,86,87,112
 Winchester discs 90
 *WORD 8
 Working disc 5,8

 Zero width fields 19

 \n 50
 ^ C (comments in report
 files) 50,54,56
 ^ number (highlights in
 report files) 54
 ^ P (page ejects in report files) 50
 \$ (directory) 90,91
 % (in reports) 49
 * (wildcard) 24
 ? (wildcard) 24
 @ (format character) 48,49
 → 14,16
 ← 14,16
 ↑ 14,16
 ↓ 14,16
 => 7,8
 6502 Second Processor 107