acorn electron

Database

CONTENTS

1 GETTING STARTED

Introduction

Acornsoft's Database enables you to use your Acorn Electron to store, retrieve and analyse large quantities of information quickly and easily.

Information handling is a task to which the computer is ideally suited, but the programming involved can be tedious and time-consuming. A database relieves you of this burden, acting as an intermediary between you and the computer and making the tasks involved in handling information simple and straight forward.

No previous knowledge of computing is required to use Database. It can store names, addresses, descriptions, quantities, dates, measurements or any other type of information. It can be used in schools, factories, doctors' surgeries, estate agents, offices, libraries, clubs, pubs or at home. Wherever information has to be stored, sorted or searched, Database is the ideal tool.

Some definitions

In Acornsoft's Database, the space that the computer makes available for each item of information is described as a field. Each field is given a name which describes the type of information it holds; this can be an address, telephone number, date of birth, or anything else which can be thought of as a single unit of information.

In most cases, a group of different fields will be related in some way; for example because they each contain information about the same person. A group of related fields is described as a record. A record can be thought of as a blank form containing spaces (that is, fields) where information can be written. If a particular record is too long for all its fields to be shown on your TV or monitor screen at one time, the computer will split it into two or more pages – just as a long form will have additional pages.

A collection of similar records is described as a file. Each record in a file must have an identical structure; that is, the same number of fields with the same names and of the same length.

3

File ━━━▶

Record ━━━▶

Field ━━━▶
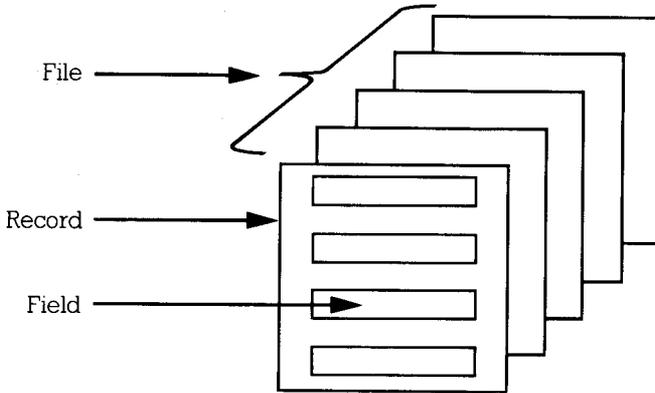
Loading the program

Data discs

To use Database, transfer the Database program on to a blank, formatted disc.

Then place this data disc in the disc drive. Type *** O P T 4 3**.

Database can then be loaded by means of an 'AUTO-BOOT' and this is executed as follows:

- Press **SHIFT**.
- While holding down **SHIFT**, press and release **BREAK**.
- Release **SHIFT**.

Alternatively, type **C H A I N "M E N U"** and press **RETURN**. After a few moments you will see the Database Main Menu:

```
           Database Main Menu

    1 : Create a new file

    2 : Examine/Update an existing file

    3 : Add a field to a file

    4 : Change field sizes/names

    5 : Convert a file to text

    6 : Make a View macro file

           Your selection :
```
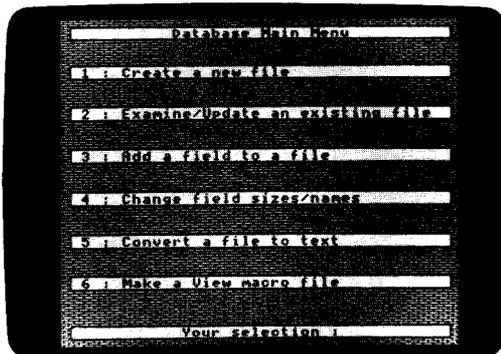
You will be able to use this disc to store your data as well as running the program from it.

4

2 SETTING UP YOUR RECORD STRUCTURE

Choosing your structure

Before you can start using Database you must tell the program about the structure you require for each record; that is, how many fields each record will have (the maximum is 32), how long each field will be (the minimum is two characters and the maximum is 240 characters), and the name that you want to give to each field (each name can be up to seven characters long).

These are important decisions which will affect the usefulness of your finished database and you should give them careful thought; it is even worth planning your record structure on paper before telling the program what it is to be. Alterations can be made later if it is necessary but it is far more efficient to get the structure right at the start. As you use Database you will rapidly learn how to design an efficient structure. Appendix B lists some points to bear in mind.

The example below shows the record structure that could be used for a simple address book database. Some other examples are given in Appendix C.

Field number	Field name	Field length
1	SURNAME	20
2	NAME	20
3	ADDRESS	84
4	TEL NO	12
5	BTHDAY	6
6	CARD?	2

Defining your record structure

This section explains how to give the program details of the record structure of a new file.

After loading the Main Menu as described in chapter 1, type **1** and press **RETURN** to show that you want to create a new file

The `Database: Create new file` heading will be displayed. Press the Space Bar to continue.

The program asks you for the number of fields in each record, and then for the name and size of each field. To see how this works, try entering the address book information shown above. Note that you must press **RETURN** after typing each entry.

After you have defined the name and length of each field, the program will ask you to enter a name for the new file. The name can be up to ten characters long; press **RETURN** to show that it is complete.

The program will create a file on the disc with the name and record structure that you have entered; although the file will not of course contain any records yet. On pressing the Space Bar the program returns you to the Main Menu.

Changing your mind

If you make a mistake whilst entering the details of the record structure you can correct it by pressing the **DELETE** key, provided you do this before pressing **RETURN**. If you notice the mistake after pressing **RETURN** you can either press **ESCAPE** to go back to the Main Menu and start again, or enter the remaining details in the normal way and then alter them later.

To alter the details of the record structure (including adding extra fields to each record) see chapter 11, 'Changing your record structure'.

Using data discs

You are strongly advised to use a separate disc for each file of Database information that you create. You can store more than one file on the same disc – but this may lead to problems when you try to enter additional information.

If you create many datafiles it may not be practical to store each file on a separate disc. In this case you may prefer to keep all your files on a 'storage' disc. Then when you want to use one of these files with Database, copy the file on to a 'working' disc and use this working disc as your data-base disc. At the end of the session, copy the file back from the working disc to the storage disc (unless it is unchanged); and delete the file from the working disc before using it with another file. *Note* that you should copy the Database program on to each working disc.

3 ENTERING INFORMATION

Getting started

To enter information into a new file that you have created, display the Main Menu and then type **2** and press **RETURN** to select the **Examine/ Update an existing file** option.

In response to the **Load File** heading type in the name of the datafile and press **RETURN**.

(If you are not sure of the name you used for the file you can enter the normal ***.** command at this point to display the files on your disc.)

After the file has been loaded the program displays the first record card. If this is a newly created file the record will be blank; if you have loaded a file for which information has already been entered the record will display the first set of this information.

The message at the top of the screen shows the filename, and the record and page number currently displayed. A record has more than one page if its structure contains more fields than can be displayed on a single screen. In this case the figure after the **/** shows the number of pages in the record: **Page 2/3** means that the screen currently shows page 2 of the 3 pages for this record. If all the fields can be displayed on a single screen **Page** is always **1/1**.

Giving commands

After selecting option 2 from the Main Menu (that is, **Examine/ Update an existing file**) you give commands to Database by using single key commands.

The command given by each key is shown on the screen by pressing **H**.

In the rest of this manual the keys will usually be referred to by the commands they give to the program. For example, 'press ADD RECORDS' means 'press **A**' and 'press REPLACE FILE' means 'press **R**'.

7

It is important to remember that the keys can only be used to give these commands whilst the basic filename/record number/page number message is shown at the top of the screen (eg **ADDRESS Rec. 1 Page 1/1**).

Entering information into a file

To enter information in the file that you have loaded you must first press either ADD RECORDS or REPLACE FILE.

ADD RECORDS instructs the program to add information at the end of the existing file. REPLACE FILE instructs the program to delete the existing file and replace it by a new file, with the same name and structure, but containing the new information that you enter. (*Caution:* If you want to retain the existing file in addition to setting up a new file, copy the old file on to a different disc before pressing REPLACE FILE.) If your file is a new one which does not yet contain any information it does not matter which of these commands you use.

After pressing the appropriate key the program displays a blank record card with the structure you have set up for the file. The **Add** message at the top of the screen reminds you that you are entering information and shows you the number of the record you are creating. After REPLACE FILE or for a new file this will be **Rec. 1**; after ADD RECORDS it will be the previous number of records plus one.

You can now enter the information you want to store by typing it on the keyboard in the normal way.

At the beginning of the first field you will see a flashing black line. This is the cursor; it shows you where the next character you type will be placed. Pressing **RETURN** moves the cursor to the start of the next line, or if it is on the last line of a field, to the start of the next field. Pressing the ↓ key moves the cursor to the start of the next field.

Pressing ↓ or **RETURN** when the cursor is at the bottom of the screen makes the program display the next page of the record (if there is one), or a new blank record card for you to enter more information.

The program will accept any information you type up to the specified length of the field; if you try to type more characters than are allowed the computer will bleep at you. However, for the information to be most useful you should, of course, enter the appropriate type of information in each field, and try to use consistent formats and abbreviations. A particularly useful format to use for dates is **YYMMDD**, so that 12 June 1984, for example, would be written as **840612**. This ensures that date order is also numerical order, which is very convenient if you wish to sort records in order of date.

8

If you have loaded the address book record structure suggested in chapter 2, 'Setting up your record structure', practise entering information about some of your friends and relatives.

Whilst you are entering information the disc drive will occasionally come into operation. This is caused by the program storing information on the disc, and is a normal part of the operation of the program.

Changing your mind

If you decide to change what you have typed before you move on to the next field, use the **DELETE** key to delete the character to the left of the cursor. Alternatively, move the cursor to the character you want to change using the ← and → keys and type the correct character.

If you want to change what you have typed in an earlier field you can return to it using the ↑ key.

If you decide that you want to change information after you have moved on to another record you must edit the record that needs changing as described in chapter 7, 'Editing the information'.

If you press ADD RECORDS by mistake, press **ESCAPE** (see the next section). If you press REPLACE FILE by mistake, press **ESCAPE** before pressing the Space Bar. After pressing the Space Bar the information in the computer's memory is lost and cannot be retrieved, unless it has first been saved on a disc.

Ending information entry

You do not have to tell the computer how many records there will be in the file; you can simply carry on adding records until all the space on your disc is full.

When you have entered all the records you want press **ESCAPE**. The program returns you to the record that was on your screen when you pressed ADD RECORDS or REPLACE FILE and awaits your next command.

Note that information entered in the record that is displayed on the screen when you press **ESCAPE** is ignored by the program and lost. For this reason you should always move the cursor to the top of the next (blank) record before pressing **ESCAPE**; unless the information on the screen is incorrect.
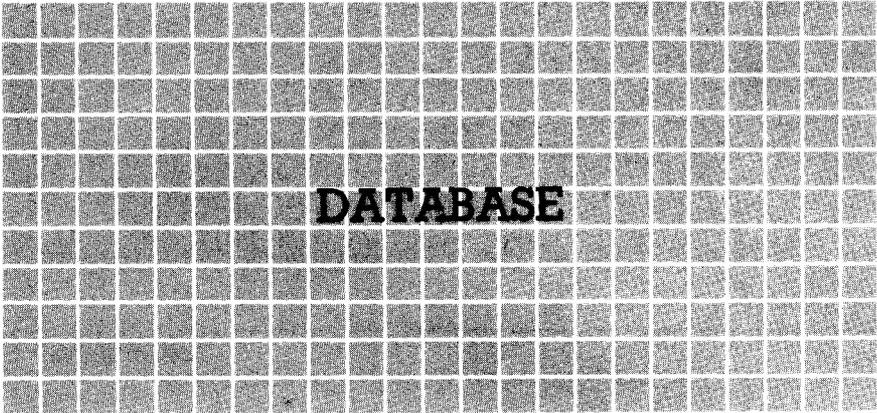
DATABASE

This manual was written by Information Transfer Limited.

(

Examining the information

To examine the information currently held by the computer, whether you have entered it at the keyboard or loaded it from disc, you use the **N** (next) and **B** (back) keys.

Pressing one of these keys instructs the program to display either the next record or the previous record (unless you are at the end or the beginning of the file). If a record has more than one page of information then you can move between pages, using the ↑ and ↓ keys. Pressing **N** or **B** when displaying page 2 of a record will move you directly to page 2 of the next or previous record.

To move to the first or last record of the file, hold down **FUNC** while pressing **N** or **B**. To move ten records forward or back through the file, hold down **CTRL** while pressing the appropriate key. If there are less than ten records after or before the current record, the program moves you to the last or first record in the file.

To examine a particular record press GO TO RECORD and then enter the number of the record you want. If there is no record with that number the program shows you the closest existing record.

Error messages

If when you press ADD RECORDS the program gives the message **Can't extend** it means that there is another file after this file on the data disc. Leave Database by pressing QUIT or **BREAK**, copy the file that you want to extend to a database disc on its own and restart the Database program from the new disc.

If the program gives the message **Disc full** it means that there is no more space on this disc. If the file you are extending is the only file on the database disc there is nothing you can do about this; you have reached the maximum size of the file. However, if there are other files on the disc leave Database by pressing QUIT or **BREAK**, copy the file you want to extend to a database disc on its own and continue as above.

If the program gives the message **Not a Datafile**, this typically means that you have attempted to apply Database functions to a file which is not a Database file.

The great value of a database is that it enables you to look up information quickly and easily. Database does this by asking you to give it a pattern describing the information that you want it to look for in a particular field. Then it searches through the records it holds looking for items that match the pattern. In this way you can use a surname to look up a telephone number, a first name to look up an address, or any one item to look up any other.

Designing a pattern

The success of a search depends on giving the program the correct pattern to match.

The program carries out thé matching process in a very literal way, comparing the pattern with each item of information in the appropriate field character by character. For example, it will recognise that the pattern:

BROWN   matches **BROWN**
          but not   **BROWNING**
          or   **BROW**
          or   **ROWN**
          or   **NUTBROWN**

Note particularly that blanks in or before words are significant characters, so that

BROWN   does not match   **BR OWN**
          or   **BROWN**

To simplify designing a pattern, the program makes available a number of special characters; that is, characters that the program interprets in a special way.

11

The most useful of these is the asterisk (*), which matches any number of any characters (including no characters at all). For example, the program will recognise that:

BROWN* matches **BROWN**
or **BROWNS**
or **BROWNING**
or **BROWN AND POULSON**

and that

*BROWN matches **BROWN**
or **H.A.BROWN**
or **GEORGE BROWN**

and that

BROWN matches **BROWN**
or **BROWNING**
or **H.A.BROWN**
or **SMITH, BROWN AND JONES**

In the majority of cases it is advisable to enclose the word that you want the program to find in asterisks; this ensures that the program will recognise the word whether it is on its own or included with other information.

A number of other special characters are available which may be useful in certain circumstances, and which enable you to construct very specific patterns. These characters are described in Appendix A.

Carrying out a search

To ask the program to carry out a search, press FIND SUBSET.

The program will show you a blank record card which matches the structure of the data in the current file. The message **Find subset** and the filename are shown at the top of the screen to remind you of what you are doing.

Against each field name enter the pattern that you want the program to look for. You can enter a pattern against as many fields as you want. Leaving a field blank is the same as saying that you do not mind what information is held in that field.

For example, if you were using the address book information structure described earlier and you wanted to look up the address of a friend called Jane Brown, you might type:

```
SURNAME        *BROWN*
NAME           *JANE*
ADDRESS
TEL NO
BTHDAY
CARD?
```

As when entering new records, you can move the cursor to the next line using **RETURN**, and to the next field using the ↓ key. If you make a mistake you can correct it using the **DELETE**, → and ← keys before moving to another line. If you notice a mistake on an earlier line you can return to it using the ↑ key.

When you have entered all the patterns that you want the program to look for, start the search by moving the cursor to the bottom of the record card. When the program has completed the search it will tell you how many records it found that matched the patterns you entered.

To examine the records that the program has found, press the Space Bar and then use the **N** and **B** keys in the normal way.

The current subset

The records that the program finds during a search are referred to as the current subset.

Most of the key commands such as FIND, SAVE, **N** and **B** only operate on the current subset of records. For example, when you use **N** to examine the records you will only see those found in the last search.

If you want to be able to examine every record you must restore the current subset to the whole file by pressing WHOLE FILE. The program also restores the current subset to the whole file if you carry out a search which finds no records, if you sort the file (see chapter 5) and if you use ADD RECORDS. When you load a file from disc the current subset is automatically set to the whole file.

You can also alter the current subset by pressing INVERT SUBSET. This includes in the current subset every record which was not previously included (that is, those not found in the previous search), and excludes those that were. Inverting the subset is particularly useful if you want to switch between two clearly defined groups within your datafile, for example men and women.

Note: Your cannot use INVERT SUBSET whilst the current subset comprises the whole file.

13

Further searching

The search may come up with a large number of records that would take a long time to examine. In this case, it may be quicker to carry out a further search using different patterns to locate the particular record or records you want.

To do this, follow the normal searching procedure as described above. This time, however, the program will only examine the records in the current subset when looking for matches for your pattern.

To carry out a different kind of search covering all the records in the file, first restore the file as described in the previous section.

Some other points

Searching Database is much more complex to describe than to carry out. The best way to learn about it is to try it, in particular examining the effect of the various special symbols.

If the program cannot understand your pattern it will print an error message. This will usually be because there are unequal numbers of right and left brackets. (See Appendix A). Note that when reporting pattern errors the symbol * will be displayed as # ? and a # will appear at the end of the pattern.

Database also enables you to rearrange or sort the records in a file according to the order of the information held in any fields. The sort can be either alphabetical or numerical.

Carrying out a sort

To ask the program to sort the data, press ORDER FILE.

The program shows you a blank record card which matches the structure of the data in the computer's memory. The message **Order** and the filename are shown at the top of the screen to remind you of what you are doing.

You type **1**, **2**, **3** and so on against the field names to show the priority that should be given to each in carrying out the sort. For example, if you wished to sort the address book information described earlier and you typed:

```
SURNAME          2
NAME             1
ADDRESS
TEL NO
BTHDAY           3
CARD?
```

the program would sort the records in order of your friends' first names. If it found two records with the same **NAME** the program would arrange them in order of surnames. If both **NAME** and **SURNAME** were identical the program would organise them in order of birthdate.

As when entering new information, you move the cursor to the next line using **RETURN**, and to the next field using the ↓ key. If you make a mistake you can correct it using the **DELETE**, ← or → keys before moving to another line. If you notice a mistake on an earlier line you can return to the line using the ↑ key.

When you have entered all the priorities, start the sort by moving the

15

cursor to the bottom of the record card. Note that, with a large file, sorting can take several minutes.

When the sort is complete you can examine the sorted records in the normal way. Note that it is only possible to sort the whole file; subsets of the file cannot be sorted. If you want a sorted subset, sort the whole file first and then search for the subset you require.

If you decide not to go ahead with a sort you can press **ESCAPE** while you are entering priorities on the blank record card. However do not press **ESCAPE** while the sort is in progress; this may result in some data being lost.

Numerical sorting

In the example above, all the sorting is carried out alphabetically; that is, entries are arranged as in a dictionary. This may result in numerical entries being sorted incorrectly; for example:

12345678

will be put before

9

since the nine will be compared with one, rather than twelve million, three hundred and forty-five thousand, six hundred and seventy-eight.

To ensure that the entries in a particular field are sorted in correct numerical order, type **N** after the number showing the priority to be given to that field. For example to sort the address book information in order of telephone number you would type:

```
SURNAME
NAME
ADDRESS
TEL  NO     1N
BTHDAY
CARD?
```

You can carry out a numerical sort on entries which are mixtures of numbers and letters. To see how the program calculates the numerical values of entries of this type, see the 'Searching for numbers' section of Appendix A.

Saving the sorted information

It is important to remember that sorting only affects the order of the records in the current file; it does not alter the order of the records held in the file on your disc.

If you want to save the records in their new order you must tell the program to do so, as described in chapter 8.

Some other points

You do not have to allocate a priority to every field of a record; fields without a priority are ignored in carrying out the sort.

The priorities you use do not have to be consecutive, but you must not use a number greater than 32 or than the number of fields you have. The priorities must all be different.

You can mix alphabetical and numerical sorts in any way you wish.

In alphabetical sorts, numbers are given priority over letters and blanks are given priority over either.

Since only the whole file can be sorted, the current subset is restored to the whole file after a sort.

Database can calculate the sum, mean and standard deviation of numerical information held in a file.

Selecting records for analysis

Each statistical analysis is carried out on the current subset of the file (see chapter 4).

The records that will be analysed are those that you see when you examine the file using the **N** or **B** keys. If necessary, first carry out a search to select the records you want to analyse or restore the file.

Carrying out an analysis

After you have selected the records you want to analyse press CALC.

The program shows you a blank record card which matches the information structure of the data in the computer's memory. The message **C a l c** and the filename are shown at the top of the screen to remind you of what you are doing.

To request a statistical analysis of the data in a particular field, enter any character against the field name. If you do not want the data in a field to be analysed, leave the box next to the field name blank.

For example, if you wanted to analyse dates of birth in the address book information you might type:

```
SURNAME
NAME
ADDRESS
TEL NO
BTHDAY    A
CARD?
```

As when entering new information, you move the cursor to the next line using **RETURN**, and to the next field using the ↓ key. If you make a mistake you can correct it using the **DELETE**, → or ← keys before moving to

another line. If you notice a mistake on an earlier line you can return to it using the ↑ key.

When you have indicated all the fields you want analysed, start the analysis by moving the cursor out of the bottom of the record card.

After the analysis has been completed the program displays the sum, standard deviation and mean of the data in each field against which you entered a character. After the last result, pressing the Space Bar returns you to the record that was displayed when you pressed CALC.

You can return to this record at any time by pressing **ESCAPE**, whether you are requesting the analyses or examining the results.

Some other points

A field for which you request a statistical analysis can contain entries which are a mixture of numbers and letters. To see how the program calculates the numerical value of entries of this type, see the 'Searching for numbers' section of Appendix A.

Note particularly that the program assumes that entries beginning with a non-numerical character have the value zero. These entries are not ignored in calculating the mean and standard deviation.

7 EDITING THE INFORMATION

Database's editing facility allows you to alter any of the information held in the record currently displayed on the screen.

Editing a record

To edit the record shown on the screen, press EDIT RECORD. The message **E d i t** is shown at the top of the screen to remind you of what you are doing, with the numbers of the record and field that you are currently editing.

You can then move the cursor to any part of the record, using the → and ← keys to move along fields, **RETURN** to move to the next line and ↓ and ↑ to move between fields. You can type in new information at any point. The character that you type replaces the character above the cursor.

When you are happy with the information in the record, press **ESCAPE** or move the cursor to the bottom of the record card. The amended record is displayed with the filename at the top of the screen, indicating that you can enter commands in the normal way.

Note that the program preserves the original entry in a field and ignores any changes, unless you move the cursor to another field before pressing **ESCAPE**. If you make a mistake in the changes you are making to a field you may find it quicker to press **ESCAPE** and then restart the editing process, rather than trying to remember and retype the original entry.

Deleting records

In many cases it is not necessary to delete records from a file. If you no longer require the information in a record you can replace it with fresh information using the normal editing procedure described above.

However, if it is desirable to exclude certain records from your file, first carry out a search (see chapter 4) to select those records that you no longer require. You will often have a field in the record structure which enables these records to be easily identified: for example, items with a stock level equal to zero, or invoices dated before a certain date. Alternatively you

might specifically set up a **D E L E T E ?** field in the structure which you can set to **Y** when the record is no longer required.

After completing the search you should examine all the records found to ensure that you really do not require them. When you are satisfied press INVERT SUBSET, leaving you with a current subset containing only the records that you do want. Finally, save the subset as described in the next chapter.

Editing more than one record

If you wish to edit several records it is possible to move from one to the other within the editor, updating the information as you go.

To move to the next record hold down **CTRL** and press **N**. To move to the previous record hold down **CTRL** and press **B**. The cursor will remain on the current field in each case. This is particularly useful when updating one field in each record belonging to a subset.

8 SAVING YOUR INFORMATION

Any changes that you make to a file by using ADD RECORDS, REPLACE FILE or EDIT RECORD are automatically copied to the file on the disc. You do not need to tell the program to save them.

However, if you have carried out a search or a sort of the file and wish to keep the result you must tell the program to do so.

Saving information

The records saved by the program are those in the current subset. The current subset can of course be the whole file (see chapter 4). Remember particularly that the current subset is reset to the whole file after a sort. You can check which records are in the current subset, and which will therefore be saved, by using **N** and **B** in the normal way.

To save the current subset, press SAVE SUBSET. The program then asks you to type in the name that you want to give to the saved file. The name you choose can be any valid filename, but it must be different from any existing file on the data disc. If you wish you can give the ***** . command to display the files on the disc before typing the filename.

When you have typed the name, press **RETURN**. After the new file has been saved, pressing the Space Bar returns you to the record it was displaying when you pressed SAVE SUBSET.

Notice that the disc containing the original file must be in your disc drive whilst the program is saving the subset. Unless you have a dual drive this means that the new file must be saved on the same disc as the original file. After saving, you should transfer the new file to its own database disc or to your storage disc.

Loading information

You can load your saved information at any time during the use of the Database program, or when you restart the program subsequently.

If you want to load the information from a different file during use of the

program, first ensure that the database disc in your drive is the one that contains the file. (The new file does not have to have the same record structure as the existing file in the computer's memory.)

Then press LOAD FILE. The program asks you for the name of the file that you want to load. Type in the name exactly as you typed it when saving the file and press **RETURN**. If you wish you can give the * . command to catalogue the disc before typing the filename.

When the new file has been fully loaded the program displays the first record of the file in the normal way.

9 PRINTING INFORMATION

Database enables you to print information from the records in the current subset. You must tell the program which fields from these records you want printed, and in what order.

Operating system commands

In order to print from Database you must first give the computer the appropriate *F X commands required by your printer (see the manual accompanying your Electron).

The Acorn Electron regards all commands beginning with * as operating system commands. To give an operating system command whilst using Database press O.S.* COMMAND. The program displays the initial *. You type in the rest of the command in the normal way and press **RETURN.**

Remember that, as with all the function commands, O.S.* COMMAND will only work whilst a record card is shown on the screen with the filename, record number and page number at the top. As you have seen elsewhere in this manual, the program will allow you to use operating system commands at other times, particularly when it is waiting for you to type in a filename. In these cases you must type the whole command (including the initial *) in the normal way.

Note that operating system commands which cause information to be loaded into the computer's memory (such as *COMPACT, *COPY) may cause the Database program to be corrupted. You should leave the program before giving a command of this type.

Printing

After giving the appropriate *F X commands, select the subset of records (if necessary) from which you want to print.

Then press PRINT SUBSET. The program displays a blank record card matching the structure of the data in the computer's memory. The message

Print and the filename are shown at the top of the screen to remind you of what you are doing.

To tell the program the order in which fields are to be printed, type **1 , 2 , 3** etc against the appropriate field names. The numbers do not have to be consecutive but must not be greater than 32. The field with the lowest number is printed first; fields with no number entered against them are not printed. All the requested fields are printed for the first record in the current subset, then the fields for the next record, and so on.

Fields for the same record are printed consecutively, separated by a single space. Each record starts on a new line. If you want the printer to do a carriage return after printing a field, put **R** after the number entered against that field.

For example, if you wanted to use the address book structure described earlier to print out a personal telephone directory you might type:

```
SURNAME    2R
NAME       1
ADDRESS
TEL NO     3R
BTHDAY
CARD?
```

As when entering new records, you move the cursor to the next line using **RETURN**, and to the next field using the ↓ key. If you make a mistake you can correct it using the **DELETE**, ← and → keys before moving to another line. If you notice a mistake on an earlier line you can return to it using the ↑ key.

When you have indicated all the fields that you want printed, start the printing by moving the cursor to the bottom of the record card. In the example given above the print-out might look like this:

JANE BROWN
01-234 5678

NIGEL JACKSON
CAM. 436987

FIONA ALLCHURCH
0468 769854

10 USING ESCAPE

If, while using Database, you decide that you do not want to continue with the present procedure, you should normally press **ESCAPE**. In most cases this is perfectly safe. The program will stop what it is doing and display the record that was on the screen when you selected the procedure. Then you can select another procedure in the normal way.

However there are a few circumstances in which pressing **ESCAPE** is not safe. These are:

- While a file is being loaded.
- While a file is being saved. The file will be incomplete and may be left open, preventing further use of that file or filename.
- While a file is being sorted. One record may be lost, being replaced by a duplicate of another record.

26

If you wish, you can change the record structure of a file of Database information. You can add an extra field to each record, or alter the name and size of individual fields. (If you want to change the actual information held in the file, edit it as described in chapter 7.)

To change a record structure you must first select the appropriate option from the Main Menu.

Returning to the Main Menu

The Main Menu is displayed when you first start up the Database program (see chapter 1).

Alternatively, if you are using the **Examine/Update an existing file** part of the program (that is, you have selected option 2 from the Main Menu and details of a record are displayed on the screen), press OUIT. Then insert the program disc, if necessary, and press the Space Bar. The Main Menu should be displayed; if it is not, and the **Insert Program Disc** message is repeated, it is because the computer cannot find the program files on the disc it is searching.

Adding an extra field

From the Main Menu, select option 3 (**Add a field to a file**).

After a few moments you will see the **Add field** title. Insert the data disc containing the file you want to alter and then enter its name.

The program then asks you for a new name for the amended file, and for the name and size of the new field. Type these in the normal way, remembering to press **RETURN** after each.

The program then saves the altered file under the new name that you have given.

Each record on the new file is given an additional empty field with the name and size you requested. To enter information in these fields, follow

the instructions for inserting the program disc. When the Main Menu is displayed select option 2, load the amended file in the normal way and then edit the records one by one.

Changing field details

From the Main Menu, select option 4 (**Change field sizes/ names**).

After a few moments you will see the **Database: Change Fields** title and you will be asked to enter the name of the file you want to alter.

The program asks you for a new name for the amended file. Then it lists the existing name and size of the first field on each record and asks you to supply the new name and size. Type these in the normal way, remembering to press **RETURN** after each. If you wish to keep the existing name or size of the field just press **RETURN**.

The program repeats this procedure for each field in the record structure. Then it saves the altered file under the new name that you gave.

Note that if you reduce the size of a field, then for each record the existing information in that field is truncated (if necessary) to fit the new field size.

12 USING DATABASE FILES WITH A WORD PROCESSOR

Converting to text

Database enables you to convert your files of information into the form of text which can be read by most word processing packages, including Acornsoft's VIEW. Then you can use the word processor to manipulate the information, or to combine it with other text, or to print it out.

From the Main Menu, select option 5 (Convert a file to text).

After a few moments you will see the Database: Convert to text title and you will be asked to enter the name of the file you want to convert.

The program then asks you for the name that it should give to the new (or destination) file that will contain the text form of your information. (The names that you use for your text files should enable you to distinguish them easily from the Database files. A convenient method is to precede the filename with the directory T; for example, T. ADDRESS.) This assumes of course that directory T has been created using * CDIR..

The program then carries out the conversion process and saves the text form of the information under the name of the destination file. When it has finished press the Space Bar to return to the Main Menu.

Note that text files cannot be loaded into Database.

Converting to 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 manual.)

Database enables you to convert your files of information into the form of parameters which can be used in conjunction with a VIEW macro. Each record becomes a separate set of parameters; each field within the record becomes a separate parameter.

From the Main Menu, select option 6 (**Make a View macro file**).

After a few moments you will see the **Macro Converter** title. Insert the data disc containing the file you want to convert and then enter its name.

The program then asks you for the name that it should give to the new (or destination) file that will contain the macro parameter form of your information. (The names that you use for your macro parameter files should enable you to distinguish them easily from the Database files. A convenient method is to precede the filename with the directory M; for example, M.ADDRESS.)

You must also give the program the name of the macro that you will be using with these parameters. A macro name must consist of two letters, for example AB or MA. The name must not be the same as a VIEW sorted command; see the VIEW manual.

The program then carries out the conversion process and saves the macro parameter form of the information under the name of the destination file. Insert the program disc and press the Space Bar to return to the Main Menu.

Note that a single set of macro parameters cannot be more than 132 characters long (including the commas used to separate parameters and any angle brackets surrounding parameters which include commas). If a parameter set produced by the conversion is longer than 132 characters Database warns you with the message **Line(s) too long!**. The conversion is completed in the normal way, but the additional characters in the parameter set are placed on an additional line without the macro name in the margin. You cannot use these additional characters in the macro unless you delete characters so as to get the whole parameter set on a single line.

If you have a field in your record structure which contains more than one line, blanks at the end of each line (except the last line) are included in the parameters. For example, information such as this:

```
SURNAME        JENKINS
NAME           JANE
ADDRESS        15, THE GROVE
               WITHINGTON
               MANCHESTER M34 8JH

TEL NO         061-111 2345
```

would be converted to a macro like this:

```
AB JENKINS,JANE,<15, THE GROVE WITHINGTON MANCHESTER M34
   8JH>,061-111 2345
```

In most cases, you will simply need to edit the parameters by deleting the extra blanks.

Note also that you cannot use more than ten parameters with a macro. If your record structure contains more than ten fields you will not be able to use the information from the eleventh and subsequent fields, unless you first edit each parameter set so as to end up with no more than ten parameters.

Creating searching patterns

This appendix describes the special characters that can be included in the patterns used by the program when searching the database; see chapter 4, 'Searching the information'.

Asterisk (*)

An asterisk matches anything (including nothing), so that the program will recognise that:

```
*BROWN* matches  BROWN
         or      BROWNING
         or      H.A. BROWN
         or      SMITH, BROWN AND JONES
```

Question mark

A question mark matches any single character, so that the program will recognise that:

```
?ROWN    matches  BROWN
         or       FROWN
         or       GROWN
```

and that **?R*** matches any word with the second letter **R**.

Hash (#)

A hash indicates that the program should look for any number (including none) of the character following it. For example, the program will recognise that:

```
#BROWN   matches  ROWN
         or       BROWN
         or       BBBBROWN
```

It can be useful if you think that there may be leading blanks before the word you want:

BROWN matches BROWN
     or   BROWN
     or   BROWN
     but not GEORGE-BROWN

Notice that * is equivalent to #? .

Round brackets and forward slash ((/))

In some cases you may want to find more than one word within a particular item of information. You can show the program this by using round brackets and the forward slash, for example:

(BROWN/BRAUN)	matches	BROWN
	or	BRAUN
(A/B)	matches	any word including an A or a B
M(R/#(,/.))	matches	MR or M, or M. or M
	but not	MRS or MS or MISS or MR.

Percentage (%)

Occasionally you may want to tell the program that part of a pattern should match no characters at all; you can do this using the percentage symbol (%). For example:

MR (%/.) BROWN	matches	MR. BROWN
	or	MR BROWN
	but not	MRS. BROWN

Single quote (')

Sometimes you may want the program to look for one of the special characters, but if you simply include the character in the pattern the program will interpret it in its special way. To overcome this problem, precede the special character by a single quote ('). For example:

'*BROWN	matches	*BROWN
	but not	H.A.BROWN
	or	BROWN

If you wanted the program to find every item of information enclosed in single quotes you would give it the pattern:

'*'' (Note that this is two single quotes, not the double quote ".)

The NOT operator (~)

In some cases, the best way to look for the information you want may be to tell the program to find everything that does not match the pattern you give it. To do this, precede the pattern by the NOT symbol. (~)

For example:

~*B* will match **SMITH,** **JONES,** and anything else not containing **B**.

The case equivalence operator (@)

The program will not normally recognise a match between upper and lower case letters; for example, **B** and **b**. If you want the program to ignore case in carrying out its search, precede the pattern by **@**. For example:

@BROWN will match **BROWN**
- or   **brown**
- or   **Brown**
- or   **bRoWn**

Note that the operators **@** and **~** must be the first characters of the pattern if the program is to interpret them in their special sense. If they are preceded by anything else, even a blank, the program will simply attempt to match them to themselves in the normal way. The only exception is when they are both required; in this **@** must be given first and **~** second.

Searching for numbers

As well as comparing characters, the program can also carry out searches on the basis of the values of numerical information.

To tell the program to search in this way, you must use one of the characters **>**, **<** or **=**. For example:

>10	will match all values greater than 10
<831231	will match all values less than 831231
and **=1**	will match all values equal to 1

When trying to interpret an item of information as a value the program ignores the first non-numeric character and anything that comes after it. If the first character of any item is not a number, the program will interpret the value of the item as zero. For example, it will interpret the value of:

	31st DECEMBER 1983	as 31
and	**DECEMBER 31st 1983**	as 0
and	**83 12 31**	as 83

34

The NOT operator (~) can also be used when searching for numbers: for example:

~>**10**   will match all values less than or equal to 10.
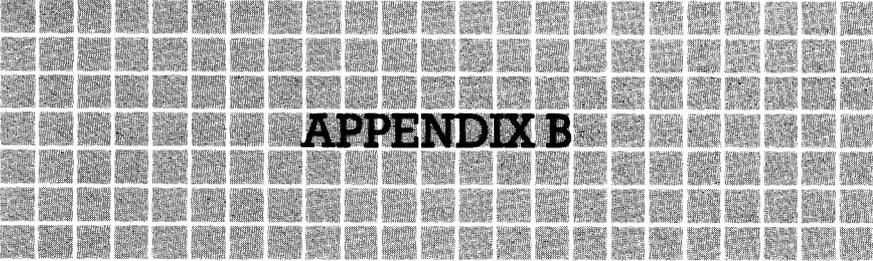
Designing your record structure

The success of your database can be affected by the way in which you design your record structure. This appendix includes some of the points that you should bear in mind.

How much information should be included?

It can be tempting to include everything that you can think of. For example, if you were creating an address book database you might include a friend's name, address, telephone number, date of birth, children's names, colour and make of car, pets, daily newspaper and so on. However, entering large volumes of marginally useful information can be very tedious, and is very wasteful of space. Try to decide the purpose of the database you are constructing, and restrict yourself to information which is relevant to that purpose.

How long should each field be?

Again, it is tempting to make each field 240 characters long 'just in case'. However, this is wasteful of space, both on the screen where it limits the amount of information you can see at one time, and on disc where it limits the number of records that can be held. In general, you should make fields as short as possible and use abbreviations to fit the information into them.

It is worth noting that one line of information on the screen contains 24 characters. If in doubt make your fields 24 characters long, or a multiple of 24 if you want a field to extend over more than one line (for example, for an address).

How should the information be divided into fields?

In some cases, large items of information such as addresses can be organised into fields in more than one way. For example, an address can be included as a single field (perhaps extending over several lines); or it

can be included as a number of separate fields: house number or name, road, town, county, postcode and so on.

The method you choose depends on the way in which you are going to use your information. If you will want to sort your records into order of road, town or postcode, then these items should be given separate fields. If you will never want to do more than look up the address, then a single field for all the components of the address will be sufficient and will use less space.

You should also give some thought to the order of the fields in each record. In particular, the first field should contain the identifier of the record; for example, the name of the person or object that the record refers to.

APPENDIX C

Some sample record structures

Personnel records

Field number	Field name	Field length	Comments
1	SURNAME	22	
2	NAMES	22	First names
3	TITLE	4	Mr, Mrs etc
4	STATUS	10	Single, married etc
5	ADDRESS	70	
6	TEL NO	12	
7	DOB	6	Date of birth
8	QUALS	22	Qualifications
9	JOB	15	
10	£SAL	5	By including £ in field name, salary can be entered as a number, enabling searches above or below particular figures
11	DAT JND	6	Date joined
12	RVW DAT	6	Review date

This is the end of page 1 of the record

13	OTHER	46	

Estate agent's house list

Field number	Field name	Field length	Comments
1	ROAD	22	
2	TOWN	22	
3	AREA	10	North, south, town, village, suburb etc
4	TYPE	10	Detached, semi-detached, bungalow etc
5	B/ROOMS	2	Number of bedrooms
6	£PRICE	6	
7	C/H	7	Central heating?
8	GARAGE	7	Garage?
9	GARDEN	22	Size and details of garden
10	DATE	4	Date built
11	CONTACT	22	Name, telephone number of owner

This is the end of page 1 of the record

Field number	Field name	Field length	Comments
12	OTHER	94	Other information
13	HALL	22	
14	LIV RM	22	
15	DIN RM	22	
16	KITCHEN	22	
17	B/RM 1	22	
18	B/RM 2	22	Size of each room
19	B/RM 3	22	
20	B/RM 4	22	
21	BTHRM	22	
22	SEP WC	22	
23	OTH RM	22	

This is the end of page 2 of the record

Appointments

Field number	Field name	Field length	Comments
1	DATE	20	
2	FOR:	20	Name of person to be seen, room to be used etc
3	9.00	20	
4	9.30	20	
5	10.00	20	
6	10.30	20	
etc			

Bills outstanding

Field number	Field name	Field length	Comments
1	PAY TO	22	
2	£AMNT	8	
3	RCVD	6	Date bill received
4	DUE	6	Date bill due
5	DSCNT	6	Date by which bill must be paid to get discount

Slide collection

Field number	Field name	Field length	Comments
1	BOX	3	
2	NUMBER	3	
3	SUBJECT	68	Subjects should be described using key words which can be searched for later
4	DATE	6	
5	BW/COL	3	
6	FILM	10	
7	EXPO	20	Exposure
8	PROC	20	Processing

APPENDIX D

Using Database with a dual disc drive

When using the Database program with a dual disc drive the program disc must always be inserted in drive 0. You may find it advisable to keep the program disc in drive 0 and use the lower drive (drive 1) for the data discs. In this case the program disc is always in position making copying the programs on to the data discs unnecessary.

When giving a filename to the program, remember to precede the name by a colon and the number of the drive on which the file is (or will be) stored; for example, **:$. ADDRESS** for a file called ADDRESS on drive 1 in directory $.

APPENDIX E

Internal format of Database files

This section is included for anyone who wishes to write his own file handling utilities, for example to merge files together, change the order of fields or merge fields together.

Each datafile starts with a large block of information which relates to the format of the file. The names and lengths of the fields are contained in this block which is called the control block. A file which contains no records, ie one which has just been created, will consist of just this control block and will be 817 bytes long.

The information in the control block is as follows:

1. The first section of the control block contains the names of all the fields. It is made up of 33 blocks of nine bytes. The first nine bytes refer to field 0 which is not used currently but may be used to mark files of different types if so wished. The other 32 blocks correspond to the 32 different fields which it is possible to create.

Each set of nine bytes begins with the two bytes 00 and 07. If the field has been used then the seven characters which were given as its name then follow, in reverse order. If the field was not used then these contain spaces.

2. The second section contains the lengths of each field. Each field from 0 to 32 is dealt with in turn, five bytes being used for each. The five bytes are 64 followed by the four bytes making up the length, the least significant byte being given first.

3. This third section is of exactly the same format as section 2. The numbers stored in here, however, are not important. The section is used during operation of the program for storing either the lengths of the extended fields set up for searching or the priorities assigned to each field during sorting and printing.

4. This section is also in the same format as the previous two. The numbers stored indicate which page of the display each field will appear on. The numbers are for reference only since the program calculates them for itself each time, hence setting these numbers differently will have no effect on the display.

5. Next comes a single integer (five bytes) which represents the maximum number of records available on the cassette version of the program. Again, this is for reference only since it is calculated each time the program runs to take into account any variations in the value of PAGE.

6. This is a single integer which is the number of records contained in the file. This is very important and must be preserved or data will become corrupted.

7. Another single integer which is the number of fields per record. It is crucial that this is correct as well or the program will not be able to function correctly.

8. A single integer which is used during operation of the program to hold various information and is not important.

9. The final five bytes contain a single integer which is the length of each record (in bytes).

In files which contain records the control block is followed by each record in turn. A record is stored as a series of fields one after another. These fields have the following format:

– The start byte 00.
– A byte containing the length of the field.
– 2 – 240 bytes of data in reverse order.

As an example consider the following record:

Initial   **P . M . F**

Number   **3 7**

The record would be stored:

00 05 70 46 77 46 80 00 02 55 51
         F   .   M   .   P   7   3

Note that if the fields were set up to contain more characters than are used, they are filled up with spaces, for example:

Initial   **P . M . F**

Number   **3 7**

Would be stored as

00 06 32 70 46 77 46 80 00 03 32 55 51
            F   .   M   .   P   7   3

This format is used because it is the same as is used by BASIC's **PRINT#F,A\$** and **INPUT#F,A\$**. Thus it is a simple matter to produce routines which can read and write the fields.

INDEX

ACORNSOFT