ATLAST PLUS

USER MANUAL

ATLAST PLUS

DATABASE MANAGER

Amstrad PCW8256, PCW8512, PCW9512

and

Amstrad CPC6128

Issue 1, 1987

This manual was written using PROTEXT and printed from camera-ready copy produced by PROTEXT on a KYOCERA F1010 laser printer.

Manual written by David Foster.

Disclaimer:

Although every reasonable effort has been made to ensure that this program functions as described in this manual, it is not guaranteed to do so, nor can it be. The user is expected to exercise caution when using it and to make regular and adequate back-up copies of databases. No liability can be accepted for any loss of data, or other damage that may occur to the user's business, or other affairs, as a consequence of using it.

Terms of Licence:

The Copyright Protection Act makes it an offence to make copies of a computer program without the consent of the copyright owner. Permission is given to the registered purchaser to make such copies of the program disc as are necessary for back-up or greater convenience of use, provided that the program name and disc serial number are written onto the disc label and that no more than one copy of the program is in use at any one time and by only one person. If you wish to make multiple use of AtLast, you should contact Rational Solutions Ltd, for details of multi-user licenses, which are available at a considerable saving in cost.

We expect users to appreciate that these licensing arrangements are advantageous to the purchaser whilst protecting the investment in labour of the program's author and that you will co-operate in respecting them. If you make copies of the program or documentation for which permission has not been obtained then you could be liable to prosecution and/or to pay considerable compensation to the copyright owner.

Data Protection Act 1984

If you keep records of the names and addresses of people, together with other details, on a database, it is probable that you should register under the Data Protection Act. This applies, whether the database is being used for business purposes, or not and there are few exceptions to this ruling. If you have any doubts about whether you are liable, it is recommended that you contact the Data Protection Registrar for full details.

CONTENTS

1. INTRODUCTION

This is the latest version of the well known AtLast database program that has been available for use on a variety of computers for some time. It includes a number of enhancements, to make it more powerful and also easier to use. Anyone who has used the original version should find that they will quickly learn this version. In spite of this, it is still recommended that the manual is read thoroughly, otherwise some of the added features may be missed.

Compatibility with Earlier Versions

One of the main enhancements is that AtLast now saves data as variable length records. One side effect of this is that it is not possible to directly load data files from earlier versions. Full details are given in the chapter called 'Advanced Use', on how to transfer data to the new version.

How to proceed

The best way of learning about any program is to read through the manual to get an idea of what the program is supposed to do, and what it is capable of. Then it is important to try the program with the manual on your lap. First you will need to set up a working disc. Then try working through the tutorial chapters, to feel your way round the program and gain confidence, referring back to the appropriate chapters in the manual when you need help. If you ever start to get exasperated or irritated, then you may be trying to take short cuts to the learning process. Don't. It is worth taking the business of mastering a powerful program like AtLast slowly and carefully.

AtLast is a sophisticated program. It requires some effort to master all of its features, but that effort is well worth while. We have tried, as far as possible to avoid the use of 'jargon' in the manual, but inevitably there are some terms that cannot be avoided when talking about a specialised program and we have tried to explain these as they arise. There may also be terms referring to the use of your computer's operating system, that you do not understand. If so, you should be able to find the required information in your computer Operating System Manual.

About the Manual

The manual is broken down into a number of chapters covering different subjects. After the initial description of databases and AtLast in particular, there is a chapter that describes how to get AtLast running on your computer. This is followed by chapters describing the various parts of AtLast in detail. Following these chapters are two Tutorial chapters that provide a step by step example of defining and then using a database. This is followed by a chapter on more advanced techniques.

There may also be some example databases on the supplied Master disc and if so, these should be loaded and experimented with.

Enhancements are periodically made to AtLast and, inevitably, it is possible that the manual may not detail the latest information. If there is a file called 'READ.ME' on the Master disc, it will detail any changes/enhancements. You can read this file, either by loading it into a word processor, or by using the CP/M command 'TYPE'. Use:-

TYPE READ.ME and press RETURN

Conventions Used in the Manual

Certain conventions are used throughout the manual. In the tutorial chapters, the keystrokes that are required to carry out commands are always shown in upper case. In use, either upper or lower case may be used. Similarly, in the examples, field names, file names and index names are shown starting with an upper case letter and sometimes with another upper case letter in the middle where a space would normally be (spaces are not allowed in these names). They could have been in either upper or lower case. We just happen to feel that they are more legible this way. For example, 'MemberList' could have been shown as 'memberlist' or 'MEMBERLIST'. As far as AtLast is concerned, these are all the same.

There are certain 'special keys' on most computer keyboards. Unfortunately these differ from computer to computer, though in many cases they have the same uses. Wherever possible, we have used the most commonly used terms, though it is impossible for it to be correct for every computer.

This version of AtLast can be used on both the Amstrad PCW and CPC range of computers. Throughout the manual the 'special' keys that are used at various points in the program are referred to firstly by their PCW name, then by the CPC name in brackets.

Where two keys have to be pressed at the same time, they are shown separated by a hyphen (-). For example:-

ALT-DEL→ (CTRL-CLR) means 'Hold down the ALT (CTRL) key and press the DEL→ (CLR) key'.

Note: CPC users should note that when they use AtLast, the program refers to EXIT in the menus. Whenever this command is shown, the ESC key should be used.

2. WHAT IS A DATABASE

What is a "Database"?

'Database' is the word given to a type of computer program that is used to store information. This information can be just about anything, from membership or address lists, to library catalogues, client lists or academic references.

The form that this information takes may differ widely, from highly organised records of names and addresses, all containing similarly sized and ordered information, to seemingly un-connected sections of text. Some examples will give some idea of the two extremes:-

a) A collection of names, addresses and probably other information, such as telephone numbers, membership numbers, or the date that a subscription was paid.
b) A collection of articles from magazines.

The first of these is called a structured database, as the information in the record for one person takes a similar form, or structure, to the information in other records. The second example would appear to have no structure, as the text may be of differing lengths and seemingly un-connected. In reality, there may well be some connection between each piece of text, whether it is a reference to an author in the various bits of text, or perhaps they are all about the same subject. This is sometimes known as a 'free-form' database.

In order to make the best use of these different types of information, different sorts of database have evolved to cater for the different requirements.

AtLast has been developed to deal with structured information in particular and in some ways may be compared with the traditional 'Card Index' systems in use in offices everywhere. This is not really a fair comparison though, as AtLast is capable of much more than was ever possible with a box containing index cards full of information.

Most people will have come across card index files being used in offices for names and addresses, or for keeping a record of spare parts etc. Whilst these can work well, they do have a number of disadvantages and limitations.

With a traditional card index system, every time changes are made to records, either the information has to be crossed out and altered, which looks messy, or else the whole record has to be typed/written out again. Another disadvantage is that it is only possible to store the cards in one order. Name order, for example, in a name and address card index. This might be sufficient as long as you only want to look for cards by searching under the person's name, but if you need to search for details in another order, then you either need another complete index file containing all the details in the original index file, but stored in a different order, or else you have to go through each card, check the contents, remove the card if appropriate and finally sort them into the order you require.

AtLast is designed to allow you to not only store information easily (as easily as you would enter it onto a card), but so that you can display the information in a number of different ways (equivalent to having further sets of index cards with the information written on them in a different order) and in different orders (equivalent to having a number of sets of cards stored in different orders), in a tiny fraction of the time it would take to do it manually.

The structure of an AtLast database

The simplest way to consider the structure of an AtLast database is to look at a fairly simple example and for this purpose we shall imagine that we wish to store all the information about the members of a club in our database.

The complete structure is known as a **'Database'**. An AtLast database may contain a number of **'Files'**, with one file containing details of the members and another file containing details of competition results, for example. The file of members will contain similar sets of information for each member and is called a **'Record'**. In our example, the 'member file' would contain a record for each member, consisting of his name, address, telephone number, membership number and details of when he paid his subscription. Each of these bits of information is called a **'Field'** in database jargon. In AtLast, these fields can be further broken down into smaller units and the 'address field' could consist of four or five lines of address and each of these are known as **'Elements'** of a Field.

It may be useful to picture the database as being a filing cabinet, with each file being equivalent to an individual drawer of the filing cabinet. A record could be likened to a folder and you would have a separate folder for each member in your drawer. Each folder would contain a sheet of paper for each separate piece of information. If any of these pieces of paper were connected with each other, they would be clipped together. Each individual sheet, or clipped together bundle of sheets, would be equivalent to a field, with each individual piece of paper in the clipped together groups being an element of that field.

With this arrangement, you could open a drawer and select any folder, open it and read all of its pieces of paper, or only those that you were interested in. You could remove unwanted sheets, alter existing, or add new. If you wanted to look at something else, you would put the papers back and select another folder from the same drawer, or you could close that drawer, open another and select another folder to look at. Where AtLast scores is that it is very much faster and it can search for all sorts of information in different ways, whereas you could only select a folder by the order in which it is positioned in the drawer.

There are two separate stages to using an AtLast database. Firstly, you must tell AtLast exactly what you want to store in the database, in what order you want to view the information and how you want this information to be displayed. This is known as 'Defining' the database.

Secondly, you would use the database, once it has been defined, to enter and view this information. These stages are fully described in subsequent chapters of the manual, but first of all it is important to make 'working' copies of the supplied Master disc, containing the necessary files. The process of creating a 'working copy' is described fully in the next chapter.

3. GETTING STARTED

1. Creating a 'Working copy'

Contents of the AtLast Master disc

The AtLast program is supplied on one 3" disc, which may be read by all Amstrad CP/M Plus computers and contains all the required program files and some example database definition files for you to examine.

Side one of the disc contains the following files:-

AL.COM	The main PCW program file that integrates all other files.
ALKEYS.PCW	Key definition file used in conjunction with CP/M SETKEYS.
DBUSE.CHN DBUSE.000 to DBUSE.005	The main Database Access program and overlay files.
SINSTALL.SUB	SUBMIT file used for single drive Installation - PCW.
TINSTALL.SUB	SUBMIT file used for twin drive Installation - PCW.
CINSTALL.SUB	SUBMIT file used during installation - CPC6128.
S?.???	Used during single drive installation - PCW.
T?.???	Used during twin drive installation - PCW.
C?.???	Used during installation - CPC6128.
TIDYCPCA.SUB	Used during installation - CPC6128.
READ.ME	A text file giving details of any enhancements/changes to AtLast, together with details of the use of the example Definition files.

Side two of the disc contains the following files:-

```
ALCPC.COM       The main CPC program file  that integrates all other files.
                This file is renamed during installation to AL.COM.
ALKEYS.CPC      Key definition file used in conjunction with CP/M SETKEYS.

DBDEF.CHN       The main Database Definition program and overlay files.
DBDEF.000
DBDEF.001

TIDYCPCD.SUB    Used during installation - CPC6128.
```

In addition, the following files may be found on either, or both, side(s) of the disc.

```
????????.DEF  Example Database Definition files
????????.FRM
????????.FIX
```

Note: '????????' in the above files means that they may have any name. Each Example Database will have one of each file 'type' with the same filename.

Details of the various examples will be found on the READ.ME file on the Master program disc. Use:-

TYPE A:READ.ME and press RETURN

or load it into a word processor, to examine the READ.ME file.

Making your Working Copy of AtLast

Warning: Make sure that the AtLast Master disc is 'write-protected' with the write protect tab before starting installation.

Note: You should enter anything that is printed in **bold** when instructed to enter commands.

Installation - All models of PCW computer

AtLast may be used with single or twin floppy drive computers, or with a hard disc (Any valid CP/M driver letter is acceptable).

Reset the computer by switching it off and then on again. Load CP/M by inserting the CP/M Plus System Disc and once the 'A>' prompt appears, you should enter:-

DISCKIT and press RETURN

Note: From now on we shall not tell you to press RETURN. Unless told otherwise, press RETURN at the end of each line that you enter.

Follow the instructions to format a disc. This should be a 'CF2 disc in A:' on the PCW8256 and PCW8512. When finished, format the second side of the disc. PCW9512 users should not format the second side, as DISCKIT will format both sides at once on the PCW9512. Mark clearly on one side of the disc, **AtLast - Access** and on the other side of the disc, **AtLast - Define.**

Note: PCW9512 users should mark the side of the disc that was uppermost when formatted, **'AtLast - Access and Define'** as both parts of the program are stored without the need to turn the disc over.

PCW9512

The supplied Master disc is in a format that may be used on all versions of the PCW. Each side of the Master disc contains program files, unlike the standard PCW9512 disc, where all programs are stored on one side. This disc may be read on the PCW9512, but you will have to insert the appropriate side of the disc, as instructed during installation. The files will be transferred onto a normal PCW9512 disc in the installation process.

Note: During installation, you will receive messages to insert the 'AtLast - Access' and 'AtLast - Define' sides of your newly formatted disc. **You should ignore these messages** and leave the 'AtLast - Access and Define' disc the same way up, but still turn over the AtLast Master disc when requested.

Follow the installation procedure for the PCW8256 (Un-expanded), remembering the points mentioned above.

PCW8256 (Un-expanded).

At the CP/M prompt (A>), select drive M: by typing:-

M: and pressing RETURN

Note: We shall not tell you to press RETURN from now on. Unless told otherwise, you should press RETURN at the end of each line you enter.

Insert the CP/M System disc into the disc drive.

B:SUBMIT A:SINSTALL

When you are instructed to insert the disc for B:, press RETURN. When instructed to insert the disc for A:, insert Side 1 of the AtLast Master disc.

When the CP/M command prompt returns, enter:-

S2

Insert the newly formatted disc marked, 'AtLast - Access', when prompted for disc for B: and insert the CP/M System disc when prompted for A:.

Note: PCW9512 users should insert the same side of the disc, marked 'AtLast - Access and Define', on all occasions.

When the command prompt returns, enter:-

S3

The 'AtLast - Access' disc should be in the drive.

When the command prompt returns, enter:-

S4

Insert the disc marked 'AtLast - Define' when prompted for B: and Side 2 of the AtLast Master disc when prompted for A:.

Your working copy of the AtLast program has now been created.

PCW8512 (and PCW8256 with expanded memory AND a second disc drive)

At the CP/M prompt (A>), select drive M: by typing:-

M: and pressing RETURN

Note: We shall not tell you to press RETURN from now on. Unless told otherwise, you should press RETURN at the end of each line you enter.

Insert the CP/M System disc into drive A: and Side 1 of the AtLast Master disc into drive B:. Enter:-

A:SUBMIT B:TINSTALL

When the CP/M command prompt returns, insert the disc marked, 'AtLast - Access' into drive A: and enter:-

T2

When the command prompt returns, insert Side 2 of the AtLast Master disc into drive B: and Side 2 of the disc marked, 'AtLast - Define' into drive A:, then enter:-

T3

Your working copy of AtLast has now been installed.

CPC6128 Installation

Note: the supplied AtLast Master disc is a Data Format disc. If you wish to attempt to make a copy to both load CP/M and AtLast, you will have to copy the files on to a System format disc, after following the installation procedure described below.

Use the DISCKIT3 program to 'Copy' the supplied Master disc on to a new blank disc, having made sure that the AtLast Master disc is write protected before you start. You must copy both sides of the disc. Mark the side of the disc copied from Side 1 of the Master disc 'AtLast - Access' and the reverse side, 'AtLast - Define'.

Once you have copied both sides of the disc with DISCKIT3, re-insert the CP/M System disc into drive A: and enter:-

B: and press RETURN

Note: We shall not tell you to press RETURN from now on. Unless told otherwise, you should press RETURN at the end of each line you enter.

With the CP/M System disc in drive A: and the the disc marked 'AtLast ~~- Access~~ DEFINE' in drive B: (single drive users should insert the disc for B: when prompted and re-insert the CP/M System disc when prompted for A:), enter:-

A:SUBMIT B:CINSTALL

When the 'PIP' asterisk (*) command prompt returns, insert the CP/M System disc into drive A: and the 'AtLast - Define' disc into B: and enter:-

B:=A:PIP.COM

On the return of the prompt, insert Side 2 of the supplied AtLast Master disc in drive A: and the new disc marked 'AtLast- Access' into drive B: (Single drive owners should insert the disc when prompted by CP/M) and enter:-

B:AL.COM=A:ALCPC.COM
B:=A:ALKEYS.CPC

Once the PIP prompt returns, press RETURN once more, to leave the PIP program.

Insert the 'AtLast - Access' disc into drive B: and enter:-

B:SUBMIT B:TIDYCPCA

When the command prompt returns, insert the CP/M System disc into drive A: and the disc marked 'AtLast - Define' in drive B: (single drive users, insert when prompted) and enter:-

A:SUBMIT B:TIDYCPCD

AtLast is now installed on your discs.

All models

You should now put the AtLast Master disc away in a safe place.

Note: You should **never** use the Master disc, other than to create a working copy. If you do and you damage it, you will not be able to create a further working copy.

Before going any further, it is recommended that you now make a back up copy of the new working version of AtLast, using DISCKIT (DISCKIT3 on the CPC6128), so that you do not have to go through the installation procedure again, in the event of a disaster.

Example files

When installed on a floppy disc, the Example files are not copied onto the disc and it is recommended that you copy them on to your data disc when you want to use them.

2. Loading and Using AtLast

PCW - All models

Switch on the computer and insert the 'AtLast - Access' side of the disc into drive A:. CP/M will automatically load followed by AtLast.

Note: If you have been using another program before AtLast, make sure that there is sufficient space on the M: drive for all the AtLast program files that will be copied onto M: during the loading process and, if necessary, erase those that are no longer required.

PCW9512

The 'Access' program files will be copied onto drive M: and you can then remove the disc from the disc drive and insert a data disc instead.

PCW8512 (and fully expanded PCW8256)

The program files will be copied onto drive M:. You should then turn the disc over and type:-

LOADREST

The 'Define' program files will also be copied onto drive M: and then AtLast will be loaded. You should insert a data disc into drive B:.

PCW8256 (Un-expanded)

Drive M: is not large enough to hold all the program files, so when you insert the 'AtLast - Access' disc, only the 'Access' part of the program is copied onto M:. When you select the '(Re-)Define' option from the menu (See below), you should insert the 'Define' side of the disc. You will then be prompted to enter the drive letter of the drive containing the 'Define' files. Enter 'A:' and the 'Define' part of the program will load.

CPC6128

Switch on the computer and load CP/M in the usual way, by inserting the CP/M Plus System disc into drive A: and entering **|CPM.**

Once the command prompt appears, you should remove the CP/M System disc and insert the side of your working copy of AtLast with 'AtLast - Access' uppermost. Enter:-

ATLAST <drive> <filename>
and press RETURN.

where <filename> is optional and is the name of an existing database and <drive> (also optional) is the drive where the database will be found. If specified, the database will be loaded as well.

Where to store the database files

Note: The PCW computers have a 'RAM' drive (drive M:). This should not normally be used to store data files or database definitions. If you forgot to copy the files onto a normal disc before switching off the computer, everything would be lost.

The drive on which you will store the database files will differ according to which computer you are using and the various options are listed below:-

PCW9512

Once the program files have been copied onto drive M: as part of the loading procedure, the 'AtLast - Define and Access' disc may be removed from the drive and a data disc inserted, as the program will be used from drive M:. Alternatively, there may be sufficient spare space on the AtLast - Define and Access disc for you to store the data files on it as well. When you specify the drive for your database, you should specify A:.

PCW8512

Once the program files have been copied onto drive M: as part of the loading procedure, the 'AtLast - Define/Access' disc may be removed from the drive, as the program will be used from drive M:. Another disc should be inserted into the drive that will hold the data. You may choose whether you wish to store your data on a disc in drive A:, or one in drive B:. Generally, you would specify B:, as it is capable of holding approximately four times as much data, but with only a small database, you might prefer to use drive A:, which provides marginally faster disc access and enables you to keep two distinctly separate databases on different sides of the same disc.

PCW8256 (Un-expanded)

The M: drive on an un-expanded PCW8256 is not large enough to hold all the program files, so a slightly different procedure has to be followed.

Note: You might consider the relatively small cost of upgrading the memory, to 512K, a worthwhile expenditure, not only for use with AtLast, but with many other programs as well.

The procedure to follow is to store the data files on the 'AtLast - Define' side of the disc. When you initially select the 'Database Definition' option (described below and in the next chapter), you should specify drive A:, both for the drive on which to find the definition program files and also for the drive for your new database. You can then define your database.

Finally, once you have tested out the database and if your database will be large, you may copy the definition program files onto another disc for use when, or if, you need to make any further changes to the definition and delete the definition program files from the data disc. All files, except for the database files, may be deleted.

From CP/M, you can enter **ERA A:DB*.*** (assuming you have not named any of your database files 'DB' something!), in order to delete the program files.

CPC6128 - Twin drives

With two drives, you would normally leave the program disc in drive A: and the database data on a disc in drive B:.

CPC6128 - Single drive

With a single drive, the situation is not quite so simple, but normally, you would DEFINE your database, specifying that the drive to store the database on is drive A:. When you have completed defining the database and saved it (onto the same side of the disc that contains the 'Define' part of the program), you should return to the CP/M prompt and copy the database files onto the 'AtLast - Access' disc. A SUBMIT file is provided on the disc, called 'TRANSFER.SUB'. Insert the 'AtLast - Define' disc and with A: as the selected drive, enter:-

SUBMIT TRANSFER filename

where 'filename' is the name of the database to be copied. When prompted for drive B:, you should insert the 'AtLast - Define' disc and when prompted for A:, the 'AtLast - Access' disc.

To ACCESS the database, you would insert the 'AtLast - Access' disc and when asked for the drive, enter 'A:'. You only need to copy the database definition files across from the 'Define' disc when you have made any alterations to the definition files on the 'Define' disc.

All models

The program will load and you will be presented with the following menu:-

```
AtLast Plus: Database Manager
-----------------------------
        (Version 2.xx)
Copyright (c) Rational Solutions Ltd., 1987
(Program devised and created by Mike York)

 D ... (Re-)Define Database

 A ... Access Database

 Q ... Quit


Enter selection
```

This is the opening menu that will always greet you when you first load AtLast and it is this menu that you will NORMALLY return to having finished either Defining or Accessing a database. This menu is referred to as the Database Manager Menu throughout the manual. It is not essential that you leave AtLast by using the 'Quit' option on this menu, as long as you leave it from either the Database Access Menu or the Database Define Menu (described in the following two chapters), although it is preferable.

Note: It is important that on all occasions you leave AtLast by following the correct procedure, as it ensures that all data is correctly saved and all files are correctly closed before you leave the program. You should NEVER leave the program by just removing the disc and switching off.

There are two distinct parts to using AtLast and these are reflected in the two choices available (plus the Quit option) in the Database Manager Menu. The first option '(Re-)Define Database' must be used before you can access the database and this is where you define (specify the dimensions of) the database.

The second choice, 'Access Database', is the one you use to enter records, view existing data and create reports and listings.

Note: If AtLast is unable to find the required 'DBDEF' (Definition program) files or the 'DBUSE' (Access program) files, as appropriate, on the selected drive, you will be prompted to 'Enter drive identifier'. If the wrong drive is currently selected, you should specify the correct drive. If the drive is correct, then the disc containing the required program files should be inserted into the selected drive and RETURN pressed.

The following two chapters describe these two options in detail and we recommend that you read through these chapters thoroughly and then study the Tutorial chapters, so that by the time you start to create your own database, you are fully conversant with the operation of AtLast.

It is also recommended that you take the trouble to design your database fully before attempting to define it. The short amount of time that is required to think out the format of a database thoroughly beforehand, is more than compensated for by the time saved by not having to keep going back and re-structuring it when you find you have got it wrong.

Before describing the methods of Defining and Accessing a database, there are a number of special 'Editing' keys that you should know about. The remainder of this chapter describes these and the standard methods used throughout AtLast to respond to menu prompts.

3. Special Editing keys in AtLast

When AtLast has been installed according to the methods described in the previous section, it will be found that certain of the 'special' keys and Function keys on the keyboard have been defined to carry out special editing tasks.

You will see in the following list that as well as the special keys, there is a column headed 'Key(s)'. This lists equivalent keys that may be used in conjunction with ALT (CTRL) to perform the same tasks. These ALT (CTRL) key sequences are the same as those used by a number of word processors and may be used if preferred.

Note: Some other programs re-define keys to suit their own purposes. If you have used another program before loading AtLast and you find that the 'special' keys don't do what you expect them to do, this may well be the cause. This situation may also arise if AtLast has not been loaded according to the recommended procedures. You should still be able to use AtLast by using the keys listed under 'Key(s)'. Alternatively, reset the computer and re-load AtLast according to the instructions.

PCW	CPC	Key(s)	Description
EXIT	ESC	^[	Escape
STOP		^[	Escape (PCW only)
RETURN	RETURN	^M	Enter. ENTER key serves same purpose.
←	←	^S	Move Cursor Left
→	→	^D	Move Cursor Right
↑	↑	^E	Move Cursor Up
↓	↓	^X	Move Cursor Down
TAB	TAB	^I	Tab cursor right
ALT TAB	CTRL TAB	^B	Tab cursor left
ALT EOL	F7	^Q S	Move cursor to start of line
EOL	F8	^Q D	Move cursor to end of line
←DEL	DEL	^H	Delete character to left
[+] or [-]	F9	^V	Toggle Insert/overwrite mode
DEL→	CLR	^G	Delete character under cursor
ALT DEL→	CTRL CLR	^Q Y	Delete from cursor to end of line
LINE	F4	^N	Insert new line/Split line
ALT LINE	F5	^Y	Delete complete line/Join lines
CUT		^Y	Delete complete line/Join lines (PCW only)
RELAY	F1	^F	Re-margin text
PASTE	F0	^W	Insert field marker

Note: 'LINE' is obtained by pressing SHIFT and the LINE/EOL key. The '^' before a character means that you should hold down the ALT (CTRL) key

Text may be entered as though you were using a typewriter or a word processing program. When the required text is entered you should press the **RETURN** key, which tells the program that the entry is complete.

If you type an invalid character then a beep will sound. Valid characters depend on the circumstances. When entering names for files, fields, indices or forms, the valid characters are all alphabetic characters, upper or lower case, digits 0 to 9 and the hyphen '-', but when entering data into records, the valid characters will differ, depending on the type of field. For example, you may use punctuation marks as well as the above characters in a text field, but letters will not be allowed in a numeric field.

If we were all perfect typists, there would be less need for editing keys, but for the less accurate typist, there are plenty of ways of altering what is typed in. In addition to the standard editing keys there are certain special keys. One such is **EXIT (ESC)**. This takes you from the current stage of the program and either returns you to an earlier stage, or advances you to the next stage, depending on the situation at the time. EXIT (ESC) is usually a good key to press if you want to abandon what you have just been doing.

The **Left, Right, Up** and **Down Cursor keys** will generally do what you would expect and move the cursor around. Up and Down will also often have the same effect as pressing RETURN. This means that you can usually finish entering data by pressing the Up or Down Cursor keys to move to the previous or next entry field.

With AtLast, you can either overwrite characters or insert characters at the current cursor position. The **[+]/[-] (F9)** key toggles between the twc possibilities.

The **DEL→ (CLR)** key deletes the character under the cursor, whereas the **←DEL (DEL)** key moves the cursor back one position, then deletes the character under the cursor.

ALT-DEL→ (CTRL-CLR) will delete all characters from the current position up to the end of the character string being edited. It is particularly useful for removing part, or all, of any existing values in fields.

The **ALT-EOL (F7)** key moves the cursor as far as possible to the left hand side of the line or character string being edited. The opposite of this key is the **EOL (F8)** key which moves the cursor as far as is possible to the right hand end of the character string being edited.

The **TAB** key means 'go to the next tab stop setting'. Tab positions are assumed to be set every eight columns. **ALT-TAB (CTRL-TAB)** moves you to the previous tab position.

The **LINE (F4)** key is used when editing a form or a definition table. In a form, it inserts a new line at the current position, splitting the current line in two, if the cursor is not at one end or the other of the line. In a definition table, it inserts a new empty line ready for a new definition.

When editing a form, the **ALT-LINE (F5)** key will delete from the current position up to and including the end-of-line marker. If used at the end of a line it will cause the next line to be joined to it. The ALT-LINE (F5) key will delete a line from a definition table.

RELAY (F1) is only used when editing forms. It will re-lay text between left and right margins, from the current position up to the next field marker or the end of the paragraph, whichever is soonest. When editing a form, normal word-wrap, as in a word-processor, operates to keep words or field markers together.

The **PASTE (F0)** key will insert a field marker at the current cursor position. The following chapters will explain the meaning of terms such as 'field marker' and 'forms'.

4. Responding to Menu Prompts

Prompts for information take four different forms in AtLast.

The main menus clear the screen and display a list of available options centrally. Selection is made by pressing a single letter key, which is the first character of each option listed in the menu (also the first character of the description).

The second type is where you are asked to fill in information between square brackets. Items such as requests for file, form and index names, are of this type. The required information is typed in and confirmed with RETURN. If the information that is required is something that AtLast could be expected to already know (such as form and index names, after they have been defined), a list of available names will usually be provided in the lower part of the screen and you may select the name by typing the minimum number of characters that make its meaning un-ambiguous and pressing RETURN. Having selected the name, you should then confirm it by pressing RETURN a second time.

The third kind is where a list of options is displayed on the bottom row of your screen. In this case, pressing the key for the first letter of the option you require will select that option, except for EXIT which is selected by pressing the EXIT (ESC) key.

The fourth type is where you are asked to confirm something. In this case you press 'Y' for yes, 'N' for no and EXIT (ESC) to escape from the prompt.

4. DEFINING THE DATABASE

Before you can enter any records into a database you must use the 'Define Database option to tell AtLast exactly what sort of information you want the database to hold. This option is also used when you wish to alter, or add further Forms to, an existing database, or modify an existing database definition (See chapter on 'Advanced Use').

When you first load AtLast, as described in the previous chapter, you will be presented with the Database Manager Menu and should select the '(Re-)Define Database' option by pressing the 'D' Key.

You will then be asked to specify the drive. You should specify the drive that contains the database you wish to use.

You will also be asked to specify the name of the database that you wish to work on. If you want to define a new database, this should be the name that you want to call the database. Any filename containing only alphabetic or numeric characters, or a hyphen (-), that is also a valid CP/M filename, may be used, but you should not specify any filename extension, as AtLast will provide a suitable extension.

Note: If you wish to make alterations to the definition of an existing database, you should specify the existing name.

AtLast will check on the specified drive to see whether a database of that name already exists. If the database is found then it will be loaded into memory and you will progress to the 'Database Definition Menu'.

If it is not found, then the program will inform you that it is unable to open the database and ask whether you want to open a new database. By implication, you either want to define a new database or else you have made a typing error, not specified the correct drive or the wrong disc is in the drive. Type 'Y' to confirm that you wish to define a new database of that name.

Defining a new Database

If you are defining a new database, the program does not take you to the Database Definition Menu because the first thing that always needs to be done is to define the files and you therefore have no alternative. On subsequent occasions, when you choose to re-define an existing database, you will be presented with the Definition Menu, but for now you should pass over the next part of this chapter and continue reading the manual from the section headed '1. D ... Data Definitions'.

Re-defining an existing Database

If you are editing an existing database, you will be presented with the following menu:-

```
AtLast : Database Definition Menu
---------------------------------

       Database: A:CLUB

D ... Data Definitions

F ... Form Definitions


C ... Change Database


Q ... Quit Definitions


Which Option?
```

You should select the required option from the menu by selecting the appropriate letter.

1. D ... Data Definitions

This is the part of the program that you automatically come to when defining a new database and also the option that you would choose if you wished to alter the definition of a file.

Note: for the remainder of this chapter, we shall assume that you are defining a new database, rather than editing one that already exists. The only difference is that when editing an existing database, all existing definitions will be displayed and may be edited in the normal way.

Warning: if you change any field or index definitions this may have other consequences which you might not necessarily foresee. See the section on reorganising the database in chapter 8 for details.

File Definition

As soon as you select this option, the menu will be replaced with the following screen:-

```
Database:  A:CLUB

FileName  Size  Can Edit  Can Delete  Can Add
[SYS-CLUB]    0     Y         N          N
```

Note: The 'A:CLUB' in the above example will be the name you have given to the database, preceded by the drive you specified. In the remainder of this chapter, the file and fields shown at the various stages are those that will be used in the Tutorial section of the manual.

For the time being, you can ignore the various headings, apart from FileName, as they are not used until the file has been defined and they are described fully later in the chapter.

Even with a new database there is one file already named. This is the system file, which is called SYS-????, where ???? are the first four characters of your database name. There may be up to ten files in a database, in addition to the special system file.

Whilst in this screen, you can choose to edit a particular file by moving the cursor (the square brackets) down to its name, using the Up and Down cursor keys and pressing RETURN at the appropriate place. By moving down to the first blank line, you can input the name of a new file, which may be any combination of alphabetic or numeric characters, or a hyphen (-) that is also a legal CP/M filename, but without a filename extension.

Note: Care should always be taken in selecting names for files, to ensure that they do not duplicate the name of any existing file, either in the same database, or any other database on the same disc. If they do, the new file will replace the original with the loss of any data in the original file.

The system file has a special role which must be borne in mind when defining files. It will only ever contain one record, which will hold all the Constant (pre-determined lists) data that the database will use and the next values of any Serial number fields. This will be explained in more detail later.

Note: If you have already planned your database and know what Constant or Serial fields you will need , you will find it convenient to define the system file immediately. Please note, however, that the actual data values of any system fields, such as the choices that will be allowed in a Constant field (as opposed to their type definitions, etc.) are NOT specified at this time.

Field Definition

Note: Although AtLast does not have any special facilities for linking related files, care in defining fields can be very useful in using AtLast's facilities to relate records to each other. For instance, if two files have the same field defined, this can be very useful in selected listings of all records of one type which refer to another and if the record referred to is indexed on that common field in the other file, then rapid switching from the referencing record is possible in Scan/Update mode. This will be explained more fully in the chapter on 'Advanced use'.

Having named a new file to edit, the screen will clear and be replaced with the Field Definition Screen. This is where you enter details of all the fields in a record and the type of data that they are to contain. Up to 20 fields may be defined for each file (See 'Elements', below).

The screen will be largely clear, except for a column of numbers down the left hand side of the screen and a line across the top, showing the following headings:-

```
No FieldName Type     Elements Length Sub-length Sign Field[El]    Size
```

Once an entry has been made under a heading, the cursor will move across to the next relevant heading. Not all headings are relevant to all types of fields and AtLast will only offer those that are, according to the type specified. For example, it will not ask if an Alpha field should be signed. It will also carry out a certain amount of checking that the definitions make sense.

Note: There are circumstances where AtLast may become confused, particularly if data types are changed in an already defined field, so you should always check for yourself that the definitions you have made are consistent.

a). FieldName

The cursor, which takes the form of a pair of square brackets, will be positioned under 'FieldName', in the first row, waiting for you to give the name of the first field you wish to define. You should type in a suitable name. Always try to choose a name that is easily remembered and relevant to the data that it will hold. It is worth choosing names carefully, as there are advantages to trying to give each field a name that starts with a different letter from other fields, or if this is not possible, the minimum number of characters that are duplicated in other fields.

The reason for this is that once AtLast has been given information such as the field names in a file, it will always attempt to find the correct entry, once it has been given the minimum required information to enable it to differentiate it from other entries. For example, if we type in the letter 'N' and press RETURN when a field name is required, AtLast will attempt to find one beginning with the letter 'N'. If there is more than one field commencing with 'N', then the first one found will be selected.

b). Type

Fields may contain information of various types. Some fields may be used to hold only numeric values, whilst others may only contain dates, or only a range of words may be considered acceptable. Fields may be defined to hold one of nine different types of data. There are several advantages to having this number of different types of data.

* Numbers of different sorts (Integer, fixed decimal point etc) may be stored more economically in the file.
* AtLast can check the validity of data when it is entered. Dates are checked for validity, numeric fields are checked to see that they are within the permitted range and are valid numbers and do not contain alpha characters and so on.
* Lists of acceptable values can be specified (Constants).

The nine types of field are:-

Alpha: Any string of letters, numbers and punctuation marks.

Upper: As Alpha, except that any characters in the range 'a' to 'z' will automatically be converted to upper case when typed in.

Integer: A number between -32768 and +32767

Fixed: A number with a fixed decimal point and up to 11 digits

Real: Any real number up to 11 digits and in the range of 10^{-38} to 10^{+38}. (Note this can suffer some rounding errors). Exponents are written with an 'E' followed by a sign and the power of ten. Examples are .1234E-20 and -.5678901E+02.

Date: A date in day/month/year format, (DD/MM/YY) or (DD/MM/YYYY).

HMS: A numeric field to hold hours:minutes:seconds (or even degrees:minutes:seconds)

Constant: An item selected from a list contained in a specified field in the System record.

Serial: A numeric field whose value is taken from a specified field element, of type Integer or Fixed, in the System record, which is subsequently incremented by one very time a new record is added.

The last two types are not allowed in the System record itself, for obvious reasons.

The Type is selected by entering the first character of the required type name and pressing RETURN.

c). Elements

Whilst it is only possible to define 20 fields in a file, it is possible to sub-divide each field into a further 99 elements and you must specify the number of elements you require at this point. The only exception to this is the Serial type field, which is only permitted one element.

Each field may be a collection of a defined type. For example a person's address may actually be one field - but would have say four or five elements - one for each line.

Using a number of elements to gather together similar information, rather than separate fields, has a number of advantages, apart from the obvious one of allowing you to have what amounts to more than 20 fields in a file. It makes it easier to remember the name of a field when a number of items of similar data are being stored, as you only need to remember the one name and then specify the element number.

In the system file (SYS-????), each field that is used to store values used by a Constant field in another file will need as many elements as there are different choices for the 'constant' field in the database (for example 12 for the months of the year). The actual choices are not entered at the definition stage, but when the database is used for the first time to add records. (The Constant choices and Serial numbers are initialised using the Edit option in the Scan/Update mode to fill in the otherwise blank fields in the System record before any other records are added. See next chapter.)

Note: When specifying the number of elements, it is desirable to allow a few extra entries for possible future expansion. The cost, in terms of disc space used, is usually minimal, but the saving in effort, should it become necessary at a later date, is considerable.

d). Length

Having specified the number of elements, this is the next item that requires entering, except in the case of Constant, Serial and Date types, where the lengths of Constant and Serial fields are automatically assigned by their associated System field and the length of a Date field is automatically assigned according to its Sub-Length (see below).

The Length of a field is always its maximum character width on the screen or printer. The range of values it may be assigned is displayed at the bottom of the screen when the cursor is in this column. In the case of numeric fields it includes the space for sign, decimal point and exponent (four characters) if these are allowed. In the case of Alpha and Upper fields, the maximum length is 79 characters.

Note that when a field contains more than one element, the length of each element will be the same and different lengths cannot be specified for each element of a field.

AtLast stores all records in the file using the minimum amount of space required, so there is no disadvantage, as far as disc space is concerned, to specifying long fields. There are, however, a number of reasons why their length should be considered.

* The length specified for fields determines the maximum length that may be entered. This means that it is possible to restrict the contents to a specific maximum size on forms. This solves any possible problems with overlapping fields, or over long lines.
* With numeric fields it allows you to restrict the maximum number of digits that may be entered.
* Some other programs require data to be input in fixed lengths. It is therefore possible to export data from AtLast in such a way that all fields are 'padded out' with spaces to the length determined by this option.
* Whilst AtLast will only use the minimum required amount of space to store a record on disc, the specified maximum length of field is used internally, before the record is written to disc. In extreme circumstances, it could be possible to specify so many fields and elements, each of 79 characters length, that the computer had insufficient memory to store records in memory, before they were stored to disc.

e). Sub-length

The Sub-Length is only relevant for Fixed, Date and HMS fields and is used to specify the format that will be used.

For Fixed fields it determines the number of decimal places.

For HMS type fields, it determines whether minutes and seconds, or just minutes are displayed.
If 3 - only hours and minutes are displayed.
If 6 - hours, minutes and seconds are displayed.

For Date fields it is the number of digits allocated to the year.
If 2 - the date will always be taken to mean this century.
If 4 - the date may be any time in the period 0000 - 9999 AD.

In the latter case, a date entered with only two digits for the year will still be assumed to be this century. Dates in the first century must be entered with preceding zeros. For example, the year 80 AD would be entered as 0080.

f). Sign

This column is only relevant to Integer, Fixed or Real fields and requires the answer 'Y' or 'N'.

If 'Y' - the numeric field can be either positive or negative.
If 'N' - the numeric field is automatically treated as positive and the plus or minus sign will not be allowed.

g). Field[El]

This column is only relevant to Serial or Constant fields and is used to link them with their associated field in the System record. A list of the available System fields is displayed at the bottom of the screen when this piece of information is required. Again you need type only sufficient characters for AtLast to identify which field you mean.

Note: The list of available System fields includes a field called '????????'. If you select this by entering '?' and pressing RETURN, then confirming the choice with RETURN again, the current screen will be replaced by a screen containing the System file definition. You may then enter a new field in the System file definition.

Alternatively, you may move the cursor to the required 'FieldName'. If you then press EXIT (ESC), AtLast will return you to the file definition you were editing previously and the field name that the cursor was on when you left the System definition will have been inserted into the 'Field[El]' space for you. This is most useful when you are defining a new field and have omitted to enter its related field into the System definition beforehand.

h). Size

This column is filled in by AtLast when all the information for a field has been entered and indicates how much space will be used internally by AtLast for that field.

Note: This figure is the maximum size that the field element will occupy when stored on disc. It will be less for null fields, or Alpha/Upper fields that do not use the maximum allocation of space.

Once all entries have been made for a field, the cursor will move down onto the following line, ready for you to enter up the name of the next field to define.

Whilst in the FieldName column, pressing the Up and Down Cursor keys will allow you to move up and down the column, to edit other fields. Pressing LINE (F4) will allow you to insert a new field, or ALT-LINE (F5) will allow you to delete the whole field.

Whilst editing the field attributes (That is, any column apart from the field name), pressing EXIT (ESC) will move the cursor back to the FieldName column. Alternatively, pressing the Up Cursor key will move the cursor back to the previous column if you need to 'back up' to correct an error.

Pressing EXIT (ESC) whilst in the FieldName column will end field definition and move on to the Index Definition stage (except with the System file which has no index).

An example screen, having entered up the field definitions would look something like:-

```
No FieldName Type     Elements Length Sub-length Sign Field[El]    Size
 1 Name      Alpha         2     20                                  42
 2 Address   Alpha         4     30                                 124
 3 PostCode  Upper         1      9                                  10
 4 TelPhone  Upper         2     18                                  38
 5 MemNum    Serial        1      5                    Number         3
 6 SubPaid   Fixed         1      6          2      N                 7
 7 DatePaid  Date          1      8          2                        5
 8 Interest  Constant      7     15                    Sports         7
 9[_______]
10
..
..
20
File:  Members
```

Once you have entered up all the fields, together with their various parameters, you can move on to the next stage, where Index definitions are created, by positioning the cursor in the FieldName column and pressing EXIT (ESC).

What is an Index?

An index has two purposes. Firstly, it provides a means to search for and obtain rapid access to an individual record, according to the fields it is indexed on. Secondly, it provides an order to be used when scanning or listing records.

A 'pseudo' Index is automatically present for every data file in an AtLast database. It is not stored in the same way as any other Index, but is created internally by AtLast. It may be used for 'ordering' purposes, but can not be used to select individual records. This pseudo index is called the 'Physical' Index and displays records in the order in which they are physically stored in the file.

AtLast stores each true Index as a separate file on disc. Care should be taken to see that index names are not duplicated in any other data files in the database, or in any other database on the same disc, or the original file, with the same name, will be overwritten and its contents lost.

You may create up to five indices, other than the Physical Index, for each data file.

Note: It is always best to have all your indexing systems worked out before you even approach the keyboard, in much the same way as you work the structure out beforehand. Further indexing systems can be added later, but it is more conveniently done at this stage, rather than later.

An indexing system keeps the database in a pre-defined order, say alphabetical order, or numerical order. With AtLast these index files are maintained up-to-date all the time, in contrast to many other database programs which require a file to be re-sorted whenever the index needs updating.

An Index consists of entries containing details from a specified field (or fields) in a record, together with the position of that record in the file. This data may be modified in certain ways before it is stored in the Index and the reason for this is described later in the chapter.

An Index is constructed in such a way that it is much more amenable to rapid searching than the sequential data file and permits the location of the record to be extracted, so that it can then be immediately and directly read from the data file.

Defining an Index consists of specifying a 'key' to be used to construct the index entry. In AtLast, each key may consist of up to three parts, each constructed from a field element. The field element making up each key part may also be 'processed', to improve efficiency (See later). The resulting Index will be ordered according to the first key part. Then, if two keys have the same values for the first part, they will be re-ordered according to the second part. If there are still equal values, then they will be re-ordered again, according to the last part of the key.

An example of this would be where you have a field called 'Name' which consists of two elements, one for forename and the other for surname. When defining an Index to select in name order, you would specify 'Name[2]' (the surname) for the first part, followed by 'Name[1]' (forename) for the second, so that the Index was initially ordered on the surname and then, for equal surnames, on the forename.

Index Definition

Once you have finished defining the fields, the right half of the screen will be cleared, leaving the left half showing the fields, their type and the number of elements. New headings will be provided on the right hand side. The headings will now appear as follows:-

```
No FieldName Type   Elements | IndexName Duplicates Field[El] Length Func
```

There will be a dividing line down the middle of the screen.

a). IndexName

The cursor will be positioned in the first column of the new part, under the heading, 'IndexName' and you should enter the name that you wish the Index to be known by.

As an index is saved as a separate file, as well as only containing characters that AtLast will accept, the name must be a legal CP/M name. Care should be taken to ensure that the name does not duplicate any other index names used in other files in the database, or any other database on the same disc.

b). Duplicates

Some index files may contain duplicate entries. Even after sorting on three levels you may still have two or more identical entries. There are some circumstances where this is acceptable and others where it is not. If two people have the same name, they would have the same index entry for an index composed purely from a name. If we wanted a unique record for every index entry, we would have to specify a third, discriminating, field in the key definition. If we don't mind duplicate entries, then we enter 'Y' in this column. On the other hand if we insist on every entry specifying a record uniquely, then we enter 'N', in which case AtLast will ensure that every record we add or edit is checked for a duplicate entry in the index, before allowing it to be saved.

c). Field[El]

This column determines the fields to be used for the Index. Simply specify the field by sufficient characters for AtLast to recognise it and then press RETURN, as usual. If the field has more than one element, you will be prompted to specify which element to use. If you forget the field names, or how many elements you gave to them, you need only look at the left hand side of the screen to see the complete list.

Note: If you select the wrong field you can go back, using the Up Cursor key, but if the incorrect field contained more than one element, you must enter a valid element number for it first.

When constructing a key part, Real type fields may not be used. Integer, Fixed, Date, HMS and Serial fields are all converted to numerical character strings which preserve their numeric or chronological order. Constant fields are converted to the element number in the system file corresponding to their value.

d). Length

This column is used to indicate how many characters in the key part, starting at the left of the string, are to be used for the key. The maximum permitted length for this part of the key is displayed at the bottom left.

The entire index entry must be restricted to no more than thirty characters in length. If you specify 20 characters for the first key part, then only 10 characters will be available for the remaining two parts.

With Integer, Fixed, Date and HMS fields, the length is the number of significant digits in the key part.
Note: Normally this function is only used to truncate Alpha or Upper fields, as AtLast both processes and right justifies other types when constructing the key part, before putting them in the string. If numeric key parts are truncated, the least significant digits are always stripped.

e). Func

This is the function that determines how, or whether, the contents of the Index are to be 'processed' before they are added to the full Index entry. The function takes one of three values, which are listed at the bottom of the screen to remind you and apply only to Alpha fields.

0 Unchanged. This means that sorting is case sensitive. i.e. capital letters are considered to be before lower case letters. This uses the ASCII character set to determine the order of the characters.

1 Convert to upper. This converts the field to upper case letters before it is sorted (though it doesn't affect the case of the letters on-screen, or in the data file). This gives case insensitive sorting and is particularly useful for dealing with problems like sorting names, where 'van Dyke', starting with the letter 'v' in lower case, would otherwise be demoted to the bottom of a list, beneath 'X', 'Y', or 'Z'. Similarly, 'MacLaren' would come before 'Mackeson', if this function was not used, due to the upper case 'L' in 'MacLaren' having a lower ASCII value than the lower case 'k' in 'Mackeson'.

2 Abbreviate. This removes all the characters not in the range 'A' to 'Z'. This is useful for picking out initials.

After the first key part (the first field element) has been defined, the cursor moves down to the next row in the 'Field[El]' column and you can define the second and third key parts in a similar manner.

Whenever the cursor is NOT in the IndexName column, the Up and Down Cursor keys may be used to move backwards and forwards through the other columns and elements for that index entry. If the cursor is in the IndexName column, the cursor keys will move the cursor up and down through the index names so that you can select any index name for further editing.

Pressing EXIT (ESC) whilst in the 'Field[El]' column brings the cursor back to the IndexName column, but pressing EXIT (ESC) whilst in other columns will indicate that you have finished and return you to the opening File Definition screen.

The following is an example of how a typical Index definition might appear on screen:-

```
No FieldName Type     Elements | IndexName Duplicates Field[El] Length Func
 1 Name      Alpha         2   | MemName       Y      Name[2]     20     1
 2 Address   Alpha         4   |                      Name[1]     10     1
 3 PostCode  Upper         1   |
 4 TelPhone  Upper         2   | NumMemb       N      MemNum       5
 5 MemNum    Serial        1   |
 6 SubPaid   Fixed         1   |
 7 DatePaid  Date          1   |[_______]
 8 Interest  Constant      7   |
 9                             |
10                             |
..                             |
..                             |
20                             |
File:  Members
```

Once all the required Index definitions have been created, pressing EXIT (ESC) will return you to the opening definition screen.

Note: The record size (in bytes) of the newly defined file will be displayed in column 2. This is the amount of space that AtLast will allocate to the record internally and not the size of a record when it is stored in the data file. The size in the file will normally be considerably less, as only the minimum space is used, plus a small overhead of seven bytes per record.

File Protection

This part of the definition procedure is where you can specify whether files may be edited, deleted, or added to, when they are subsequently used. Further information on the use of this facility is given in the chapter on 'Advanced use'.

AtLast provides three data protection methods. Any data file can be protected from having its records edited, deleted or added. This is specified in the three columns headed 'Can Edit', 'Can Delete' and 'Can Add'. By default they are all set to 'Y', which means that the file may be edited, deleted or new records may be added. The exception to this is the System file which cannot be deleted or added to. You must leave the 'Can Edit' flag set to 'Y' for the System record until after you have initialised your constants and serial numbers the first time that you use the database to enter records. After you have done this it is a good idea to return to this screen and set this flag to 'N', to prevent any corruption of, or alteration to, your constants.

Having set these flags, by pressing Y or N, the cursor will move on to the next line, ready for a further file to be defined. AtLast allows up to ten files, beside the system file, in each database. Although each file is logically independent, having several files in the same database means that it is easy to switch between them when the database is in use. This is particularly useful if different data files contain related information. Further information on related files is available in the chapter on 'Advanced use'.

The following is an example of how the screen might appear at this point.

```
Database:  A:CLUB
FileName  Size  Can Edit  Can Delete  Can Add
SYS-CLUB   115     Y         N          N
Members    236     Y         N          Y
[_______]
```

You may elect to define a further file, in which case you repeat the procedures described above, or you may choose to leave Database Definition, in which case you should press EXIT (ESC).

You will then be offered the option to 'Save, Edit, List, EXIT (ESC)?'. Pressing 'S' to save the definitions will result in the definitions being saved and a return to the above message so that you can continue editing if you wish. Selecting 'Edit' will return you to the file list so that you can continue editing.

The 'List' option is provided so that you may print out a complete listing of the structure, either to a printer, the screen or a disc file. The section on 'Print', in the Scan/Update in the next chapter, should be studied for details of how to change the default print settings, if necessary. By default they are suitable for use with standard 11" continuous stationery and if the 'Single Sheet Paper' option is set to 'Y', single sheet A4 paper.

Selecting EXIT (ESC) will take you to the Database Definition Menu. This is the menu that you would have encountered first, when editing an existing database structure (rather than defining a new one). If you attempt to use EXIT (ESC) without first saving the definitions, AtLast will ask whether you wish to 'Save new definitions (Y/N)?' before taking you to the Definition Menu.

NOTE: Please note that any changes you make to existing file definitions may also imply that changes need to be made to any associated form definitions. It is also possible that your new definitions may be incompatible with any existing data. If you are making changes of this sort, but wish to preserve your existing data, then you will need to follow special procedures. See the chapter on 'Advanced use' for details.

2. F ... Form Definitions

Forms determine the way that the data is presented, either on screen, in a disc file or when printed. It is not really practical to have a database that merely lists out the raw data. In real life, we often want the data displayed with accompanying descriptions, or headings, or laid out in a particular way on the screen. Naturally, if you are using a database heavily, you will need to have a number of forms for various purposes, such as creating standard letters, producing checklists or generating labels. AtLast calls all these things 'Forms' and any file can have a number of them.

When File definitions are saved, AtLast checks to see whether any forms already exist for the defined files and, if not, it automatically generates a form for every file in the database for which a form does not already exist. The one possible exception to this is that a form will only be created for the SYS-???? (System) file if any fields have been defined in it. These default forms will always be given the name of the file to which they are related. For example, the form created for the SYS-???? file will also be called SYS-???? (where '????' is the first four characters of the database name).

Each of these forms consists of a 'simple' form, containing all the fields and elements that have been defined in the file, preceded by text descriptions consisting of the field name. This form may be used for entering and displaying data, but usually you will want to create your own forms, tailored exactly to your requirements.

AtLast creates these forms to ensure that it is possible to enter and display data in any file in the database, even if you have forgotten to define a form for any of them. This is particularly important with the SYS-???? file which will normally only be edited the first time that the database is used, in order to specify any Serial or Constant values. Generally this file will be quite adequate for this purpose, as it will not normally be viewed or used to display data. The next section describes how you may create your own forms.

Defining a Form

A Form may, and usually will, be a mixture of text and 'field markers'. The field markers mark the places on the form where data will be entered or displayed and the surrounding text usually has the purpose of explaining what this data is. When using the database, this text will be constant for all records displayed, or entered, on the form, but the data values occupying the places marked by the field markers will vary from record to record.

Each form is divided into three sections, called Head, Body and Tail. When Adding, Scanning and Editing data, only the Body of the form is used.

The Head and Tail sections are only used when listing records. The Head section is listed at the beginning of each page of the listing but the Tail section is only listed once at the end. The main use of the Head section is in providing column headings for tabulated listings. No field markers may appear in the Head section. The Tail section is often used to report totals of numeric fields, or the number of records listed. Totals will be calculated for any field element for which there is a field marker in the Tail section.

You can have as many Forms as you like for any File, assuming you have the disc capacity for the resulting FRM file. The maximum size of a form is about 1,000 characters (excluding continuous 'spaces' and blank lines) or 140 field markers (each field marker uses 6 characters) or a compromise between these.

There are two ways that you may go about creating new forms. You may either copy an existing form and then modify it to suit your needs better, or else you can start with a clean sheet and completely design your own. Which method you choose to use can only be determined by trying both methods, but often it will be found convenient to copy and amend forms that will require little modification, but to create completely new layouts from a clean sheet.

When you first select the Define Forms option you will be offered the alternatives of 'Get (a form), Create (a new form), EXIT (ESC)?'. The 'Get' option would be selected if you wished to alter an existing form, but if you want to create a new form you should select 'Create'.

If 'Create' is selected, you are asked to specify a name for the form, which may be up to 12 characters long, followed by the name of the file to which it is to be related (available files are listed in the lower part of the screen). You are then presented with a blank form and the cursor in the top left corner, ready for text and field marker entry, which is described under 'Editing a Form', below. Pressing EXIT (ESC) will display the available commands, as described under 'Get'.

When 'Get' is selected, a list of available forms is displayed at the bottom of the screen for you to select the form you want. Typing sufficient characters for AtLast to recognise the form, then pressing RETURN will result in the form being selected. The screen will clear and be replaced by the form.

The lower part of the screen will contain the Status line and details of the available commands:-

```
Form: Members        File: Members   Section: Body

Edit, Head, Body, Tail, Save, Get, Delete, Rename, Create, EXIT?
```

The 'Section' heading on the Status line will always show which part of the form is currently selected for editing. By default it will be the 'Body'.

All of the options, with the exception of EXIT, for which the EXIT (ESC) key is pressed, are selected by keying the first letter of the command.

a). Edit

Selecting this option will move the cursor to the top of the screen, above the 'End of Section' line, so that you may add text or field markers to the form. Full details of how to edit a form are given later in this chapter under the heading 'Editing a form'.

b). Head

Selects the 'Head' part of the Form. The head may only contain text and it is not possible to insert field markers. The main use for the Head is so that when Listing is in use, the contents of the Head will be printed at the top of each page. It will frequently be used to provide headings at the top of a page for tabulated listings.

c). Body

This will re-select the Body of the form for further editing, if either Head or Tail has been used.

d). Tail

Selects the 'Tail' of the form for editing. The Tail will be printed only once, at the end of the form and its principal use is to enable you to print out total figures for any numeric fields. During the course of the listing, any fields that have been marked in the Tail will have the figures for that field totalled for each record and at the end of the listing, the totals for each field marked in the Tail will be printed.

e). Save

Self explanatory. It enables you to save the current form without leaving the Form Editing screen.

f). Get

This provides the facility to 'Get' another form for editing. When selected, the cursor moves to the 'Form:' heading on the Status line, so that you can enter the name of the new form you want to edit.

g). Delete

Allows you to delete the currently selected Form. You will be asked to confirm the deletion.

h). Rename

Allows you to rename the currently selected form. When selected, the cursor will move to the 'Form:' part of the Status line, so that you may change the name of the Form. The original form will be replaced by the newly named form in the definitions.

i). Create

Allows you to create a new form. When selected, the cursor will move to the 'Form:' part of the status line so that you can name the new form you wish to create. Once you have named the form, you will be asked whether you wish to 'Copy current form (Y/N)?'. If you select 'Y', then the existing form will remain on screen, but the original will not be removed from the database definition. This is one method by which you can copy one of the AtLast generated forms (or any other form) for modification and then saving with a new name. See 'Editing a Form' later in the chapter for details of how to alter an existing form. If you select 'N', then the screen will clear and you may create a new form from the beginning.

j). EXIT (ESC)

Enables you to exit from Form Definition. If you have not saved the Form, then AtLast will ask whether you wish to save it or not. Selecting 'N' will result in the form being abandoned.

Editing a Form

There are a number of reasons for wanting to edit a form. Some examples are:-

* You want to make the field headings more explanatory
* You want to provide extra help for someone who will enter the data (e.g. a list of acceptable values for a Constant field)
* You want to put headings into the Head section for a tabulated listing
* You want to insert total markers in the Tail section
* You want to include special directions for another program that might take its input from an output listing from AtLast (e.g. 'dot' commands for a word-processor).

Editing a form is in many ways similar to using a simple word processor. Word wrap will operate as in a normal word processor, whether in insert mode or not.

There are a number of 'special' keys that may be used to carry out certain tasks and these are listed in the previous chapter and also in the appendix.

Forms are not limited to the size of the screen and if you reach the bottom of the screen, the form will scroll up by half a screen. The width of a form is limited to the width of the screen for viewing purposes, but facilities are provided to allow greater widths to be used when listing to a printer or a disc file. Details will be found at the end of this section.

You may freely move around the screen entering and altering text. Pressing RETURN or the Down Cursor key when at the 'End of section' line will result in the bottom of the form section being moved down to make room for more text and field markers. In addition to the normal deletion of characters, parts of, or whole lines may be deleted and blank lines inserted, by using the appropriate keys described in the previous chapter. Insert and overwrite mode of text entry may be toggled by using the [+]/[-] (F9) key.

While in the text part of the form, pressing the LINE (F4) key will result in the insertion of an 'end-of-line' character, causing any text on the line to be split at the cursor position while ALT-LINE (F5) will delete everything up to and including the end-of-line marker, thus joining the next line. If you position the cursor at the end of a line and press ALT-LINE (F5), then the following line will be pulled up and joined to the end of the line that the cursor is on.

Note: When LINE (F4) is used to delete the last line of a form, the cursor will be re-positioned at the end of the 'new' last line.

Pressing RELAY (F1) will cause text to be re-distributed, one space between each word, between the left and right margins up to the end of the paragraph or the next field marker, whichever comes first. This provides a convenient way to tidy up text that has been messed up by editing.

Field Markers

You can indicate where a field element is to be displayed on the page by the use of the PASTE (F0) key. This will insert the current default field marker at the position of the cursor. Initially, this will be undefined (shown by a '?'), unless the cursor has 'passed through' an existing field marker. If a field marker has just been deleted, then this will become the current default, otherwise it will be the next element in order from the last field marker passed through. (Either the next element in the same field array or the first element of the next field array.) Whenever the cursor is in an area occupied by a field marker, it is highlighted and the following options are displayed at the bottom of the screen:-

```
Field Marker: Edit, Delete, Nudge, <Cursor>
```

a). Edit

This will allow you to specify the name of the field (and, if necessary, the element number) that you want displayed at that point. A list of available fields will be displayed in the lower right corner of the screen.

Having named the field, you will then be asked if you wish to omit leading and trailing spaces when listing and printing. This can be set to 'Y' to ensure no unwanted gaps occur between fields when they are printed. An example of where you might use this would be in a label form, to prevent large gaps between forename and surname. It would also be used on most occasions where you wish to export data to a file for use by another program.

Note: Some other programs will only accept 'constant length fields' as input, in which case, if the form was going to be used for 'data export', you would make sure that this option was set to 'N', so that each field would be padded out with spaces to the length specified in the field definition.

When used in the Tail section of a form, the field markers become 'total markers'. When the file is listed, the tail section is listed at the end and any field marked in the Tail will have been totalled over all records listed and the total will be displayed at this point. An extra field marker, '#####', is available and, if used in the Tail, then the total number of records listed will be displayed.

b). Delete

Allows you to remove an existing field marker from the form. When a field marker is deleted, the deleted field will become the next one used by AtLast when a new field marker is inserted. This is particularly convenient as a means of moving a field marker around a form.

c). Nudge

This command has the effect of 'nudging' the field marker one space to the right. This is mainly provided so that you can move an existing field marker away from the left margin, or away from an adjacent field marker, so that text may be inserted. Once a space has been opened up using this command, text may be entered and as long as 'Insert Mode' is selected, the field marker to the right will be pushed across as text is entered. In 'Overwrite' mode, the cursor will move back into the field marker.

d). <Cursor>

Any of the Cursor keys may be used when in a field marker, in order to move out of it and back into the normal part of the form.

Wider Listings

Whilst AtLast will not permit you to display any forms wider than the width of the screen when displayed on screen, it is possible to send output to a printer or disc file in such a way that wider lines are created. This means that the full width of wide carriage printers may be taken advantage of. Alternatively, using a different typestyle (such as Elite or even Condensed mode) will allow you to print more than the number of characters across the screen on one line. The same form can still be used for screen listings without losing the display off the right hand edge of the screen.

This is done by marking the end of a line with a backslash '\', to indicate that the following line is to be treated as an extension of the current line. When a listing is made and AtLast finds a backslash followed by an 'end-of-line' sequence, it will ignore both the backslash and the 'end-of-line', with the result that the following line will be printed on the same line. (On a PCW computer, the '\' character is obtained by the EXTRA-½ combination.

Another occasion when this facility will be found useful is when creating a form for the export of data in 'comma-delimited' format, which requires all the fields of a record to be on one line. This is covered more fully in the chapter on 'Advanced use'.

3. C ... Change Database

This option is provided to enable you to load different databases for re-definition, without the need to leave AtLast and start from the beginning again. If this option is selected, all files in the existing database definition will be closed before the new database is opened. The screen will clear and you will be presented with the same screen that you encountered when you first entered the database definition. The drive and the new filename should be specified in exactly the same way as before.

Note: When this option is used, all previous settings in AtLast will revert to their default values and no data of any sort will be carried over from one database to the next.

4. Q ... Quit Definitions

Selecting this option will normally return you to the Database Manager Menu, from which you may either choose to Access your database, or select 'Q' to return to the CP/M prompt. If you want to leave AtLast altogether, you should then select 'Q' again, from the Database Manager menu.

If, however, the disc containing the 'AL.COM' file is not in the expected drive, then this option will return you straight to the CP/M prompt without passing through the Database Manager Menu.

5. USING THE DATABASE

Once you have defined your database, as described in the previous chapter, you are nearly ready to start entering records.

If you have just returned to the 'Database Manager' Menu, having completed the Database Definition, there is no need to reload AtLast as you can switch between Defining and Accessing a database without leaving the main program. If you have just switched on your computer, you should load AtLast in the same way as previously.

Select the 'A ... Access Database' option from the Database Manager Menu by pressing 'A'.

Note: If AtLast is unable to find the necessary program files on the specified drive, then it will prompt you to specify the correct drive. If the drive is correct, then insert the disc containing the program files into the drive and press RETURN.

If you have just loaded AtLast, you will then be asked to specify the drive. You should specify the drive that contains the database you wish to use.

You will also be asked to specify the name of the database that you wish to use and this should be the name that you initially gave when you started to define the database. AtLast will then load the appropriate data files and you will be presented with a new menu.

If you have come to this point directly from defining the database, then AtLast will remember both the drive and the name of the database and it will be loaded automatically.

Note: If this is the first time that the database has been accessed and the System file contains any defined fields, a reminder to initialise the System record, using the 'Scan/Update' option, will be displayed. See the following section for full details of what this means.

When the 'Access Database' option is selected, the following menu is displayed.

```
AtLast: Database Access Menu
----------------------------

     Database: A:CLUB

A ... Add Records in Batch

S ... Scan/Update Data

L ... List Data

C ... Change Database

T ... Tidy/Repair Database


Q ... Quit Database Access

Which Option?
```

Using a Database for the first time

If you are using a new database for the first time and when you defined it you defined any fields in the System file (called SYS-???? unless renamed) to hold the values of any Constant or Serial fields, you must now specify the starting values of any Serial fields and also the acceptable values of any Constant fields. To do this, you should select the 'Scan/Update' option from the menu by pressing 'S'. The screen will clear and you will then be asked what Form you want to use. At least two Forms will be listed as being available and you should select the form called SYS-???? ('????' will be the first four characters of the database name unless you have Renamed it during Definition). Do this by typing in sufficient of the letters of the form name to enable AtLast to recognise it from any others. If you followed the recommendation to choose names that start with different characters for each form, then just typing 'S' will be sufficient. Press RETURN and the full form name will appear under the cursor. If correct, then pressing RETURN once more will select the form.

The SYS-???? form was created automatically by AtLast at definition time and will be displayed on screen, showing the names of any Constant or Serial fields that you defined. Near the bottom of the screen will be a line of text containing details of the form selected, the file being used and other information that is of no importance at the moment. This is the Status line. Beneath this line is a further line showing the message, 'Form, Edit, Print, EXIT?'. At this point, we wish to enter the appropriate information, so we should select 'Edit' by pressing the 'E' key. The cursor will now move to the top of the screen and position itself on the first field requiring an entry and we can start to enter the details.

If the field is associated with a Serial field in another file, we should enter the number that we wish the first record to start with. Remember that a Serial field is a numeric field that will be incremented by one every time a new record is entered.

If the field is associated with one to hold acceptable Constant values, then it will have a number of elements and we should enter the first word (or words) that we wish to be in the Constant list. When complete, pressing RETURN will move the cursor on to the next element, which should also be filled in and so on, until all the full list of Constant data has been entered. Full details of how to edit and amend entries are given in the following section of this chapter.

Once any Constant and Serial data has been entered, EXIT (ESC) should be pressed and the record saved with 'Save' option, before returning to the Database Access Menu by pressing EXIT (ESC) again.

Note: The System file will only ever contain one record, containing details of Constant and Serial values. You cannot add further records, but you may change the values and add new Constant values (assuming there are spare elements) in this record by using the Scan/Update option

It is not possible to make further use of the Scan/Update option for any other files, until such time as a minimum of one record has been entered. The next step is then to enter some records using the 'Add Records in batch' option from the menu.

Using a Database at all times

1. A ... Add Records in Batch

This option is normally selected when new records are to be entered into the database. The screen will clear and a list of available forms will appear at the bottom of the screen, together with the message 'Use what form?'.

Typing in sufficient characters to enable AtLast to recognise the required form and then pressing the RETURN key will display the complete name of the form. If this is correct, pressing RETURN a second time will select it. If it is not the correct form, either the entry may be edited, or else the complete entry may be deleted with ALT-DEL→ (CTRL-CLR) and a new entry made.

Once selected, the form will appear on the screen and you will be prompted 'Keyboard or File input?'. New data may either be entered by typing it in at the keyboard, or it may be entered automatically from a data file on disc, as long as it is in a suitable format. You would usually choose Keyboard, so that the details may be typed in.

a). Keyboard input

As soon as this option is selected, the cursor moves to the upper part of the screen and is positioned on the first field that requires input. The entry may then be typed in and then RETURN should be pressed to indicate that the entry is complete. The cursor will then move to the next field and the process may be repeated. If there is no entry required for a field, it may be passed over, either by pressing RETURN, or the Down Cursor key.

When entering data into a Constant field, it is only necessary to enter sufficient of the word for AtLast to recognise it, before pressing RETURN and the program will fill in the rest for you. Any Serial fields will be passed over and filled in by AtLast automatically.

When you have no more data to enter for a record, you may press EXIT (ESC), even if you have not reached the end of the form. A message will appear at the bottom of the screen requesting, 'Save, Re-do, <Up>, EXIT?'. If the end of the form has been reached, this prompt will appear automatically.

Note: Any attempt to enter non numeric characters into a numeric field, or alphabetic characters into a date field will be rejected at the time, but when RETURN is pressed, or Down Cursor is used to move on to the next field, AtLast will check the entry and reject any invalid data by refusing to move to the next field until it has been corrected. AtLast checks that any data entered in Constant fields matches one of the values in the System record. It also checks the validity of dates. Note that any spaces in a Constant field entry are taken into account when checking is carried out, so that putting two spaces between words, when the Constant had only one, will result in the entry being rejected.

If all the information is correct, then pressing 'S' will save it to the data file. If you wish to abandon the record altogether, then pressing EXIT (ESC) at this point will return you to the Access Database Menu without saving the

current record. If, however, you notice that either there are some mistakes, or you have missed something out, you may press 'R' to Re-do, in which case the cursor will be positioned at the start of the first field in the form. Alternatively, you may press the Up Cursor key and the cursor will be positioned on the last field in the form.

You may then move around the form, correcting entries as you go, using the Up or Down Cursor keys to move the cursor up and down through the fields in the form. Once all corrections have been made, you can finish editing the form by pressing the EXIT (ESC) key and the 'Save, Re-do, <Up>, EXIT?' message will re-appear.

Once the Save option has been selected, the record will be saved, the form cleared and the cursor positioned at the first field in the form, ready for another record to be entered. When you have no more records to enter, pressing EXIT (ESC) twice will return you to the Database Access Menu.

b). File Input

Entry of data from a disc file is a very simple process. Firstly you will be asked to enter the name of the ASCII file containing the data text. The contents of this file are read one byte at a time, as though it was being typed in from the keyboard. Just as you press RETURN between field entries when typing records from the keyboard, so the text read from the file must simulate this and this is represented by the carriage return/linefeed sequence at the end of each line of data in the file.

The program does not distinguish between what is keyed from the keyboard and what it takes from a file, so the text in the file must be valid data only, with no extra explanatory material. No extra 'carriage return/linefeed sequence' (no blank line) is necessary between each set of field elements (i.e. each record) as the program knows when all the fields on the form have been filled.

All this may sound complicated, but all it really means is that each field, or field element (item of data) within a record must be on a line by itself in the text file (even if blank) and in the correct order required by the field markers on the form. If the program suspects that you have made a mistake, (if a validation fails, or there is an invalid duplicate index entry in a file where no duplicate index is permitted) then entry will pause and you will be given the option to continue or abandon the transfer of data into your database. Depending on the situation, you may choose to make a note of any records that contain such data and then correct them when the transfer is complete, or you may decide to abandon the import of data and correct the file before trying again.

Note: It is important that you make a backup of the database files before attempting to import data. If anything goes wrong with the import, the back up files may then be used to return the database to the state it was in before the import was attempted, before having a further attempt.

Just as in keyboard entry, all Serial fields are entered by the program and should not, therefore, be present in your data input file. Particular care is necessary with Constant fields, as they must contain exactly the same text as that defined in the System record and care should be taken with the use of spaces, as they are also taken into account as part of the field.

2. S ... Scan/Update Data

This is the option to use when you want to browse backwards and forwards through the records in the database, or search for individual records and it allows you to view them in a number of different ways. You may specify 'Conditions' that will determine whether records are selected or not, which Form is used to display them and in what order they will be displayed. In addition, you may add new records and alter or delete existing records.

When the option is first selected, a list of available Forms is displayed, together with the message, 'Use what Form?'. The required form should be selected as described earlier in the chapter and will then be displayed on screen. Beneath the Status line will be a list of available Index files and the message, 'Which Index order?' at the bottom of the screen. The Index names displayed are those that you specified when defining the database and it is these Index files that govern the order in which records will be displayed. The required Index is selected in exactly the same way as the form was selected.

There is always one special Index. This is the Physical Index, which is a 'pseudo' Index created by AtLast and is the order in which the records are physically stored in the data file. If the Physical Index is chosen, it is not possible to specify a starting point, so AtLast will automatically select the first record in the data file.

If you are using a real Index, rather than the Physical index, the cursor will then move into the form and will be positioned on the first field used in the chosen Index. The message, 'Enter Key at or BEFORE record to find' will be displayed at the bottom of the screen. The object of this option is to establish the starting point in the Index from which the records will be displayed. If you wish to start at the beginning of the Index, then just press RETURN. If you only want to view certain records, it will usually be possible to specify either a starting record, or a point just before the ones you wish to view.

If the index you selected was based on names, for example and you wanted to look at the records for anyone whose name began with 'S', it would be a waste of time starting at the beginning of the Index, so you would just specify 'S' at this point and the first record with a name starting with 'S' would be displayed. If the Index consists of more than one field element, then the cursor will move on to the next Index field element, as well.

If the selected Form does not contain one, or more, of the field elements that make up the selected Index, then any elements of the Index that are displayed on the form will prompt for values and the other elements will assume a null value (equivalent to pressing RETURN).

Having specified the starting point of the records to be displayed, the first record will now appear in the form. The Status line at the bottom of the form will display the currently selected Index, Form and File, the number of records in the file and the Physical Position of the record in the data file.

Note: The Physical Position of a record may change if a record is altered, as the figure shown is not tied to a record. It is merely the position of that record in the data file **at that time.**

A number of options will then be displayed at the bottom of the screen;-

```
'Form, Index, Back, Next, Conditions, Search, Add, Edit, Delete, Print, EXIT?'
```

Each of these options may be selected by typing the first letter of the word, except for EXIT which you select by pressing the EXIT (ESC) key. If the selected Index is the 'Physical' index, then the 'Search' option in the menu is replaced by the word, 'Start', as it is not possible to search on the Physical Index.

Note: Not all these options are available for every file. For instance, you cannot change the Index for files that have no Index defined. Nor can you Scan or Search for a record in the System file as there is only one. Nor can you edit, delete or add records that are marked to prevent this. This marking is carried out during Definition of the database and consists of the options 'Can Edit', 'Can Delete' and 'Can Add', each of which may be set to 'Y' or 'N'. See the section in the chapter on Defining a Database for full details.

a). Form

This option lets you select a different Form to use and therefore display the current record in a different Form. This is particularly useful if you have a Form that only shows the main details of a record, which may be more convenient for scanning purposes, but once you have found the record you want, you can switch to another form, giving more detailed information.

Another example might be where you find a record and then decide that you wish to print out an address label for it. You would not want all the information displayed on the form to be printed on the envelope, so you would switch to a Form designed as an address label layout and then select the Print option (see below).

Because Forms are always linked to a specific File, you may also switch to a different file by using this option. When you select a Form that is linked to a different file, the existing file will be closed and the new file opened.

An example of this would be where you have an entry in a name-and-address database that contains a membership code. Using the Form option, you can switch to a Form used by the 'members' file, select the entry with an index based on the membership code, by entering the membership number and the membership details for that individual will then be displayed. You can, with this option, leap nimbly from file to file within a database, pursuing a line of enquiry.

Note: Every time you change files, any selection Conditions (See below) are reset so that no filter is applied and it will be necessary to re-specify them if they are required. It will also be necessary to re-specify an Index to scan on (See Index command below).

b). Index

With the 'Index' option, you can change the index order in which the file is scanned. It is not necessary for the Index elements to be displayed on the Form for AtLast to find the correct records.

Note: If for any reason AtLast cannot find the current record in the new index, due to some inconsistency in the database (This can happen - See the section on repairing the database), the form will clear and you will be prompted to give a new key for searching, as in the Search option (below), to find a new start location for scanning.

c). Next

This allows you to display the next record in the database. The 'next record' is the one that is next in the current scanning order unless a set of selection conditions have been defined, in which case it will be the next record that meets those conditions. When the end of an Index is reached, a blank record is displayed, with the message, 'End/Start of file'. Selecting this option again will re-start the scan from the beginning again.

All records in the file are displayed as it is scanned and the scan stops when a record that meets the conditions is found. Alternatively, the scan can be stopped at any point by pressing any key and you will be asked whether or not you want the scan to continue or not. A great deal of time can be saved by using the right index and selection conditions to scan on and making a good entry in the key fields for your initial search.

d). Back

This allows you to display the previous record in the Index scanning order. The previous record is 'previous' in the current scanning order and according to any selection conditions that are in operation. It is the opposite of the 'Next' option and otherwise identical in its use.

e). Conditions

This enables you to specify selection conditions which act as a filter when scanning using the 'Next' and 'Back' options. Any records which do not match the conditions will be passed over. If the current record does not meet the conditions, a message will be displayed, saying that the 'Record does not match conditions'. Selecting Next or Back will continue the scan until a matching record is found, or the start or end of the file is reached, if no matching record is found. When more than one condition has been specified, only those records that match all of the conditions will be selected.

Note: Any conditions will remain in force until either the conditions are changed, or the data file is closed and another file is opened. Conditions are always cleared when a new file is opened.

When you first select this option, the screen will clear and the new screen will contain details of the currently selected Form, File and Index at the top of the screen.

Beneath this will be the heading, 'Field Selection Condition' and beneath this, the cursor, positioned under the 'Field' heading. At the bottom of the screen will be a list of all the available fields in the current file and you should enter the field on which you wish the condition to be checked. You do this in the normal way, by entering sufficient characters for AtLast to recognise it and pressing RETURN to fill it in, followed by RETURN again to select it.

If the field selected is one that contains more than one element, you will then be prompted for the element that is to be used. If you specify '0' for the element number, all elements will be checked. The condition only has to be met by one of the field elements, to be considered 'true'. In the case of testing a multi-element field for 'Not Equal' or 'Not Containing' (See below), the result will only be 'true' if none of the elements are equal to or contain the comparison value.

Once the field is selected, the cursor will move across and position itself under the 'Selection' heading.

The Selection operators that are available will be displayed at the bottom of the screen and are as follows:-

BF : Before, or lower, in the sequence (before in alphabetic order if a string of characters, lesser if a number, beforehand in date or time, or lower in element order within a constant field).

AF : After, or higher, in the sequence (after in alphabetic order if a string of characters, greater if a number, afterwards in date or time, or higher in element order within a constant field).

EQ : Equal to, or matching the comparison value.

EB : Equal to, or before, in the sequence (the same as a string or before in alphabetic order if a string of characters, less than, or equal to, a specified value if a number, beforehand or the same in date or time, or lower or equal in element order within a constant field).

EA : Equal to or after, in the sequence (the same or lying later in alphabetic order if a string of characters, greater or equal if a number, afterwards or the same in either date or time, or the same or higher in element order within a constant field).

NE : Not equal to, or not matching the 'comparison value'.

CN : Contains the 'comparison string'. This is only relevant for Alpha or Upper fields. If, for example, you were to explore a surname field in a name and address list, using the CN operator with 'bl' in the 'comparison value', it would select all those names containing a 'bl', including not only 'Bloggs' 'Blackshaw' and 'Blythe' but also 'Ablett', or 'Hubble' (all contain a 'b' followed by an 'l' within the field).

NC : Does not contain the 'comparison string'. This is only relevant for alpha or upper fields and is less useful than the CN operator. If the specified field with in the record contains the string entered in the 'comparison value', then the record is omitted from the listing.

Note: Comparisons are case-insensitive (no distinction is made between capital letters and lower-case letters) for string fields but otherwise they are in 'ASCII' order.

Once the comparison operator has been chosen, the next task is to put in the comparison value. Obviously, it must have the same data type (eg. Alpha, Upper, Integer, Date etc) as the field it will be compared with. Constants can be specified in the usual way (i.e. just typing enough characters to define the constant element uniquely). Dates and HMS fields must be entered with their proper delimiters.

A number of comparisons may be made on fields and they are entered into the list one at a time and when one has been entered, the cursor will move down to the next line, ready for the next one. Once the list is complete, pressing EXIT (ESC) will return to the original form that was being used for scanning.

Note: The same field may be used more than once, in order to restrict the acceptable range to be selected. For example;-

debt EA 100 - Debt equal to or greater than 100
debt EB 1000 - Debt equal to or smaller than 1000

will result in all records where 'debt' is between 100 and 1000 being selected.

Conditions remain in force until they are either cancelled by new conditions, or by removing them from the Condition list. Positioning the cursor on the condition that is no longer required and pressing ALT-LINE (F5). Closing the file and opening another file will reset the conditions.

If the current record does not match the conditions when you first return to the Scan option, a message, 'Record does not match conditions', will be displayed. The file may then be scanned by using the Next and Back options, or scanned from a new starting point in the Index by using the Search option.

f). Search (Start - If Physical Index is selected)

This initiates a new scan through the database, using a value that you will be asked to enter for the current Index. This means that, having done one search, you can initiate another.

Note: If the current Index is the 'Physical' index, which is created by AtLast, the Search option is not available and in its place is 'Start', which has the effect of starting a scan from the first physical record in the data file. If your file does not contain an Index, then 'Physical' will be the only Index available and simply displays each record in the order in which it is stored in the file. This need not necessarily be the same order as the order in which the record was entered.

If, for example, you are doing a series of address updates on a name-and-address database, you can find each name in turn, make the alteration, and then search for the next one. The same rules apply as for the initial search. This search activity finds the record corresponding to the first entry in the index on or after the value that you typed in.

If you simply want to search for the first record in the file then press EXIT (ESC) or enter blank values in the key fields by pressing RETURN.

If you see a message **'End/Start of File'** then this means you have reached the end of the file. The message **'Free Space'** means that the record has been deleted and is available for re-use. This should only occur when using the Physical Index. The message **Record does not match conditions** will be displayed if the record found does not meet the current selection conditions. The message, **Error in Index file**, will occur if there is an irregularity in the index. See 'Repairing a Database' for more details.

g). Add

With this option you can add a new record to the database. This is done in exactly the same way as adding batch records, described in the previous section, except that data entry is from the keyboard only and that you are returned to the Scan/Update mode after saving. The saved record becomes the current record and the start point for any further scanning.

Note: If you abandon the record without saving it, you will have to specify a new starting point for any further scanning

h). Edit

Editing a record allows you to change or alter a record's contents. You can only alter those components, or fields, of a database that are within the current form. This is actually exactly the same as adding a record as already described. Obviously, if one needs to update the information contained within a particular record, this is the way to do it.

If you press EXIT (ESC), instead of Saving the record when you have finished editing a record, the original data will be restored. If you Save the amended record and any changes were made to any fields that are part of an Index, AtLast will automatically update any index file that is affected.

Note: If the changed entry affects the current Index, further scanning will take place from the new index position. The physical position of the record (as shown by 'Pos:' on the Status line) in the data file might also change.

This option must be used when you first use a new database, in order to fill in the Constant values and initialise the Serial fields in the system record. Remember also that Constant values are assigned by their element number. If you subsequently change the order around then the values that appear in the data records that use them will change. See earlier in the chapter for details of entering Constant and Serial field values.

i). Delete

You may delete the current record from the file if you want, by using this option. Deleting a record can be a saddening experience if it is done accidentally. To prevent this, the program will ask you to confirm deletion before going ahead.

j). Print

Selecting the Print option will clear the screen and the current 'Print Parameters' will be displayed, together with a message, 'Change Parameters (Y/N)?'. Selecting 'Y' will move the cursor up to the list of parameters so that you may change them. The default settings, which are ideal for use with standard 11" continuous stationery:-

```
Destination        : Printer
Single Sheet Paper : N
Use Form Feeds     : N
Page Length        : 66
Printed Lines      : 58
Page Each Record   : N
```

Allowable values are:-

Destination	: Printer, Screen or File. Selected with P, S or F. If File is selected, then the program asks for a Filename. If the name specified already exists on the disc, you will be asked whether you wish to over-write it with the new data. See warning below about the use of the printer.
Single Sheet Paper	: Y/N. This option is only available if Printer is selected and determines whether listing pauses at each page end, so that new paper may be inserted.
Use Form Feeds	: Y/N. If Y is selected, a form feed code will be sent to the printer at the end of each page. This should only be used when using single paper is selected.
Page Length	: 0-255. The default setting of 66 suits standard 11 inch continuous stationery (at 6 lines per inch). A4 continuous paper should be set to 70.
Printed Lines	: 0-255. The difference between the value of this setting and Page Length determines how many blank lines are printed between each record. With continuous stationery, it provides the necessary blank lines to skip over the perforation.
Page Each Record	: Y/N. If Y is selected, then a new page is started for each record, otherwise one record will follow another, according to Page Length and Printed Line settings.

If Page Length is set to an exact fraction of the length of the paper and Printed Lines to the same number, selecting this option will cause a number of records to be printed on each page with the necessary blank lines between. Positioning will be maintained from page to page. For example, using 11" continuous stationery, if you wanted to print three records to a page (assuming that 3 will fit on a page!), you could set 'Page Length' to 22 (66 divided by 3), 'Printed Lines' to the same value and 'Page Each Record to 'Y' and each time 'Print' is used, it will equally space each record on that page and subsequent pages.

'Destination' is selected by typing the first character of the destination device. Page lengths are set by entering the number. Press EXIT (ESC) and you will once more be asked whether you want to change the parameters. Answering 'N' will cause the current form to be printed.

Warning: If the destination is 'Printer', it is important to make sure that the printer is switched on and 'On-line' before printing is started. On some computers, the effects of trying to print when the printer is not 'On-line' can be unpredictable and may even result in loss of data that has been updated during the current session.

Whereas the most obvious use of this function is to get a printed copy of the currently selected record, it is also useful for any purpose that requires the extraction of small amounts of information from the database. If, for example, you wanted to send a 'standard' letter to selected customers whose details are kept in a database, but you needed to inspect the record before deciding on who should be sent the letter, then you could prepare a special form beforehand, consisting of the whole or part of the letter. Once a record is selected, you select your 'letter' form, using the 'Form' option and then print the form. If the letter is reasonably short and uniform, it can be printed straight to the printer, but for longer letters it may be more convenient to print it to disc, for subsequent printing using a word processor.

Printing to screen is less useful, but allows you to see what a complete form looks like if it is longer than the screen area.

k). EXIT

Selecting EXIT (ESC) will return you to the Database Access Menu.

3. L ... List Data

The ability to produce lists is one of the most important assets of a database. Lists can be sent straight to the printer, to the screen or can be written to a disc file. At times, most database users will require to select lists based on a number of conditions. For example, the possessor of a database of plants may need to produce a list of all plants with red flowers that thrive in shade on an acid soil. At other times, a simple list of all, or part, of the records in their alphabetic order is all that is required.

With AtLast, all these tasks are possible and are carried out by using the 'List Data' option. Having selected this option, the Form that is to be used for the listing should be selected. This could be a letter, a label, a 'Card Index' type of form, or a simple list with each field spaced out, in columns, across the screen. Next, you will be asked for an Index to use for listing. Selection of the Form and Index is carried out in exactly the same way as for the previous two options.

The screen will be replaced with the Listing Screen and Menu. The Status line is at the top of the screen and gives details of the selected Form, File and Index. In addition, it shows whether the Index is to be used in 'Ascending' or 'Descending' order. With a numeric Index, this means that the lowest value will be listed first and subsequent records will be in increasing value order. If the Index is an Alpha, Upper or Constant field, the listing will start with the record with the lowest ASCII value. 'Descending' is the reverse and in effect means that the listing will be from the end of the Index to the beginning.

The bottom of the screen will list the available options:-

```
'Form, Index, Range, Conditions, Ascending, Descending, List, EXIT?'
```

Several of these options are the same as those in Scan/Update and are only briefly covered here. Full details may be found in the 'Scan/Update' part of this chapter.

a). Form

Allows selection of a different form to be used for listing and if the form is linked to a different file, the current file will be closed and the new one opened.

b). Index

Allows re-selection of the order in which the listing is to be made.

c). Range

Selecting this option allows you to specify a range in the Index for which selection will be valid. You will be asked to enter a 'Lowest' Index value, from which the listing will commence and also a value for the 'Highest', which will be the point at which listing will stop. You will be asked to give values for each field element that is part of the Index 'key'. The default settings are the first (Lowest) Index entry up to the last (Highest) Index entry.

'Lowest' means the actual value, or the highest value that precedes, the value of the first record that you want to list. 'Highest' means the actual value of the last record that you want listing or the lowest value above the last record you want printing. If you want to list all names beginning with 'F', you could specify a lowest of 'F' and a highest of 'G'.

AtLast will display the nearest index entries to those that you specified, so that you can check you selected correctly. You may change the selection or confirm it by pressing RETURN. When both lowest and highest have been chosen, you can still change your mind by answering 'N' to the 'Is this correct (Y/N)?' question.

If the selected index is the Physical index, then you may specify the starting and finishing number, but remember that the physical index number is not fixed and may change if records are subsequently edited.

Note: Specifying a range is more efficient than using the 'Conditions' option with the equivalent selection settings.

d). Conditions

Permits Conditions to be applied, so that only certain records will be listed. This is the same as the option of the same name in 'Scan/Update' and you should see the section on Scan/Update for details.

e). Ascending

This option lets you specify that the listing will be made from the lowest record in the index to the highest record. This is actually the default setting, but this command lets you reset it, if it has been changed to 'Descending' previously.

f). Descending

This enables you to list records from the highest in the index order, to the lowest. This can sometimes be useful if you wish to list dates in order of the most recent first, for instance. (For birth dates this would be youngest first.) Outstanding debts might usefully be listed in this order, with the largest at the top of the list!

g). List

This option is selected to set the listing in motion and is used when all the other options have been specified. Having selected it, you are asked whether you wish to 'Change Parameters (Y/N)?'. The default settings and the methods used to change them are exactly the same as those in the 'Print' option in 'Scan/Update', which should be studied for full details.

Once 'N' has been selected in answer to whether you wish to change the parameters, you will be asked whether you wish to 'Print selection conditions (Y/N)'. If you select 'Y', then any conditions that have been set with the 'Conditions' option will be listed before the main listing of records. This can be useful, in that it provides an indication of exactly which records have been listed when the listing is studied at a later date.

When listing forms, any 'Head' or 'Tail' parts of the form will also be listed. The Head part will be listed at the top of each new page that starts with a new record. The Tail will be printed only once, at the end of the listing, together with the totals for any field elements that are marked in the tail part of the form.

h). EXIT

Pressing EXIT (ESC) returns you to the Database Access Menu when no further listings are required.

4. C ... Change Database

This option is provided to enable you to select a completely different database to work on, without the need to leave AtLast and reload it and is exactly the same as the option of the same name in the Database Definition Menu and its use is fully described in the previous chapter on Database Definition.

5. T ... Tidy/Repair Database

Note: Before using this option, it is important to make a back up of all the data files, so that you have copies to fall back on in the event that the repair is not successful the first time.

There are a number of reasons why you might want to use the 'Tidy/Repair Database' option:-

* The amount of free space remaining on the disc has become inadequate.
* You have made alterations to an existing Index Definition, which means that the Index file must be rebuilt.
* You have added a new Index to the database, which means that existing records need indexing on the new Index.
* The database has become damaged in some way.

Reasons for Collecting Free Space

Unlike many databases, which store records as fixed length blocks in a file, no matter how short the contents of the record, AtLast stores records in such a way that only the minimum amount of space is used for each record. This has obvious benefits in terms of the amount of data that may be stored on a disc. When a database is first started, each new record is stored immediately after the previous record, but, if you subsequently delete a record, the space that it occupied will be freed for later use.

Similarly, if existing records are changed by adding extra data, so that they are larger, there will be insufficient space available in the original location, so AtLast saves the amended record at the first available place in the file that is sufficiently large to hold it. Initially, this will probably mean that the record is added to the end of the file, but as existing records are deleted or altered, the space that they used to occupy may be used instead, if it is large enough.

The chances of the record being exactly the same length as the old record that has been deleted are very small and this will usually result in a small amount of space remaining unused. After a considerable amount of changing and deleting files, there may be a number of these small, unused, areas throughout the file.

When AtLast saves records, it maintains a linked list of them within the file, keeping track of the records, as it goes. When a file is closed, not only is all the data you typed in, saved, but the program saves special information about what and where the data should be.

In the course of re-building a database, AtLast will automatically gather together any contiguous areas of unused space, into larger areas, but these may still not be large enough to hold a record. AtLast has an option to gather together all these areas into one.

Reasons for Re-building an Index

If a new order of searching or listing a file becomes necessary, which was not predicted when the database was originally designed, then you can add it by using this option, in conjunction with the (Re-)Define Data facility. Full details of how to do this are given in the chapter on 'Advanced Use'.

If you have either altered an existing Index Definition, or created a new Index, it is necessary to re-index the existing data file to which it relates, so that the file may be ordered according to the newly (re-)defined index.

Reasons for Repairing a Database

In addition to tidying the database, there are a number of reasons why a database might need repair:-

* You may have attempted to enter information into a database when there is insufficient room on the disc.
* The disc may be faulty.
* You may have suffered a power cut before alterations have been saved onto disc.
* You may have switched the computer off without leaving the program properly by the 'Quit' option from the menu, or you might have taken the disc out before leaving the program.

If something does go wrong, you may notice 'odd' things happening.

* Any changes you made might not be recorded properly.
* Records may begin to disappear.
* Unfamiliar error messages may appear.

Whatever may have occurred, it is possible to rebuild the index files, which constitute an important part of the structure of a database, from the original data **as long as the original data file is intact and readable** and, in some circumstances, it is possible to recover data which would otherwise have been lost.

Using Tidy/Repair

When you first select the 'Tidy/Repair Database' option you will be confronted with a display of all the possible Data and Index files that might need tidying or repair. By default, all selections are set to 'N'. If you wish to carry out any processing on a file, you must select it by moving the cursor with the Up and Down Cursor keys and changing it to a 'Y'.

Once a file is selected, you are offered the option to 'Collect free space at end of file (Y/N)?'.

You would not normally use the 'Collect free space' option, unless you are working with a large database and find that you are running short of disc space on which to save records. If this situation does arise, then selecting it will result in the file being re-arranged so that all these small blocks of unused space, contiguous or not, are collected together into one large area at the end of the file and the records are all pushed to the front of the file.

Note: AtLast has to do a considerable amount of work in order to gather all these blocks into one, at the end of the file and the process may take some time to complete, which is why you would normally only use it when it became absolutely necessary. The 'Repair' part of this option carries out a process of gathering together contiguous areas of free space in a file and this is faster and may well provide the space that is required, without having to use the 'Collect free space' option.

If you have decided not to select the Collect free space option, the cursor will move to the first Index name for that file. You should select any Index files you wish to re-build by selecting 'Y'.

Note: All the index files for a data file will automatically be selected for re-building when 'Collect Free Space' is selected, as the records themselves may have been moved in the process.

When you have selected which files you wish to 'Collect Free Space' for, or want to 'Re-build', or 'Re-Index', you may press EXIT (ESC) to indicate that you have finished and the program will display the message, 'Do Tidy/Repair (Y/N/EXIT or <Up>)?'. If you respond with a 'Y' then AtLast will start processing the files and finally mark the end of each file that has been repaired.

WARNING: If you have any records with illegal duplicate keys (this could arise from having made changes to the index key definitions after having saved them) then only the first of these records (in file order) will be entered into the rebuilt index. Any illegal records will still exist in the data file and can still be accessed for editing (to make them legal again) from another index.

Errors during tidying/re-building

If AtLast finds a place in the data file, during tidying/re-building, that it cannot understand, it will halt, giving the message, 'Un-recoverable error'. You are then given the option to mark the data file at that point as the 'End of File'. If you do, then all data prior to the 'End of File' will be intact, but all other data after that point will be lost. This may be alright, as the succeeding data may be either non-existent, or redundant. Alternatively, if you elect not to mark it, you may be able to recover the data if you are experienced in the use of programs such as SID.COM, but this is not for the faint-hearted.

Note: The above situation is rare and will only normally occur in situations where a disc fails, or something similar. If you have kept regular back ups of all your files, which you should always do, then the situation is rarely disastrous and only a small amount of data will be lost, at worst.

After the process has been completed, things should work a lot better. Occasionally, data may be irretrievable and may need to be re-entered by hand, or some of the records may be corrupted and need deleting and re-entering, but the basic integrity of the database will have been restored.

6. Q ... Quit Database Access

Using this option, will return you to the Database Manager Menu.

6. 'DEFINING' TUTORIAL

In this chapter we shall go through the procedure involved in creating a database and, having created it, we shall then use it to enter in a few records. Finally, we shall use this data to experiment with the various methods that are available to display the information in a variety of ways.

We could have picked a very simple 'Name and Address' example, but as AtLast possesses so many features, we have chosen to pick a slightly more involved example. There is nothing particularly hard or complicated about it and you will soon find that once you get used to the way in which AtLast works, you will probably be getting ahead of the Tutorial!

We recommend that you load AtLast and then read through this tutorial, entering text as instructed. If you find a particular part that you don't understand, then stop and read the section in the appropriate chapter in the manual again, before continuing with the tutorial.

Throughout the tutorial, all input that you are required to type in is shown in *Italics*, whereas program prompts are shown in **bold** typeface. The ordinary text is our commentary and instructions.

Planning the Database

Probably the single most important thing to do is to plan your database. Whilst it is easy enough to just rush in and define a database and then use it (you might even find it does what you want!), there will ultimately come a time, sooner or later (probably sooner) when you will wish you had done it differently. AtLast is quite capable of allowing you to re-define your database, but it is never as convenient as getting it right first time and will take considerably longer to do than it will take you to sit down and plan it properly from the start.

The first thing is to decide exactly what you want the database to achieve. This may sound obvious, but you will often find that when you actually get down to planning it, things that seemed unimportant suddenly assume far greater importance and vice versa.

In this Tutorial, we shall create a database for keeping a record of members of a Club. We shall assume it is a general sports club, though it could just as easily be any sort of club. The first thing to do is decide exactly what information we must store. Obviously we must have a record of each member's name and address and in this case, we also want to have a record of their telephone number. We also want a record of when subscriptions were paid and the amount they paid. As it is a general club, we also want to know what their interests are.

So far, our list of requirements looks something like this:-

```
Name
Address
Telephone
Amount paid for subscription
Date subscription paid
Sporting interest
```

Perhaps the next thing we ought to do is consider what we are going to do with all this information once we have entered it, so we shall now consider the ways that we might want to look at this information and list or print it out. We will obviously want a form to enable us to enter the information when the person first joins. To save us work when we send out our newsletters, we will want to be able to produce address labels. We will also want to be able to produce a simple list of all members, showing their names and telephone numbers. We also have a word processor with powerful mail merging facilities and we want to be able to use the data from the database to send 'personal' letters to each member on occasions, so some method of creating files that can be read by our word processor will be required. Our list of ways that we might like to display or list the information will now look something like:-

```
Entry form for initial information about a member
Address label facilities
The ability to list members' names and telephone numbers
The ability to 'export' data
```

We should consider the ways that we might want to choose which member records are selected. We would obviously want to be able to select them by name and it would also be useful to be able to select them by their membership number (That is a good point. We haven't made any provision for a membership number in our list of what we want to store in the database, so we must add that to the list). Our list of items we would like to search with now looks like:-

```
Name
Membership number
```

At this time, we do not want to keep any records of each member's sporting achievements, but if we did, then we would find it most convenient to keep them in separate files for each sport, so, in case we ever do want to, have we got any feature in our database that would be useful to provide a link with any new databases? Yes, we have. We could use the membership number or the name to provide a link (it is always useful if our link is also something that we would normally select a record with and both name and number qualify). If there hadn't been any sort of logical link, then we would have had to consider whether we ought to have some further detail, in order to provide this link.

We have now got an outline of what our database should look like and after a check to make sure that we haven't missed anything out, it is time to start to fill in the details, so we shall start by deciding what type and size each field ought to be, in order to hold out information.

Name - Normally when we look for someone's name, we will look down the list for their surname and then, if there is more than one person of that name, then we would look at their initials. We would also sort our list into surname order. This means that we need to be able to look at the surname first. We could always enter them into the database as 'Smith J' or 'Bloggs David', but it wouldn't look very good on our address labels if they were sent out to 'Bloggs David' etc., so the sensible thing is to separate them into different 'pigeonholes' (fields). AtLast has the ability to have a number of different elements or 'pigeonholes' for the same field, so we shall make use of this facility and call our field 'Name' and give it two elements, one for the surname and the other for the initials or forename. If we give them a length of 20, it should be more than adequate. As names nearly always start with capitals, but the rest of the name is in lower case letters, we shall have to use an Alpha type field.

Address - Most addresses consist of three or four lines and a post code. It would be a good idea to make use of a number of elements, as this information is all of a similar type, but it might also be useful to keep the postcode separate from the rest, as it might prove useful as something to search on at a later date (postcodes can provide a useful means of sorting or searching as they are unique to given areas of the country). A length of 30 ought to be adequate for the addresses, but the maximum length of a post code is only nine characters, so we shall settle for 9. The address field ought to be of type Alpha, as we would normally use an address as a mixture of upper and lower case characters, but the post code field would benefit from being of type, Upper, as the letters in a post code are always shown in upper case, so even if we forget when typing it in, AtLast will do it for us.

Telephone - The most efficient way to store telephone numbers is by making use of the dialling code, instead of using the exchange name, not only from AtLast's point of view, but for convenience of use as well. We really ought to make provision for two telephone numbers, as we might want to contact the member at work as well as at home, so again, we shall make use of elements. The maximum length required for a telephone number and dialling code, including allowing for the space between the parts is 14, but as one of the numbers is a work number, maybe we ought to allow a few more characters for an extension number, so we shall settle on 18. We shall also make the field of type Upper, as it will look neater if any text in the number is consistent.

Amount paid - This is always going to be a number and it will also have to show the pence columns as well, so looking at our list of available field types, we can see that we should use a 'Fixed' type, as that allows us to specify the number of decimal points to be used. When deciding on the length, we must allow for the two figures after the decimal point AND the decimal point when deciding on what length it needs to be, so, assuming that our subscriptions are going to be less than £1000.00 per annum, we can settle for a length of 6.

Date - This is easy to decide on. We should obviously make use of the 'Date' type and the length will be decided for us by AtLast.

Sporting interest - We only provide a certain number of sporting facilities in our club, in our case, four and the entry will always be one or more of these, so we can take advantage of the 'Constant' facility and let AtLast do some of the work for us when we come to enter in the data. There is always the possibility that the club might introduce more sports, so we should also allow for a few extra interests, to take account of this, so we shall settle for seven possible elements, as some members may have more than one interest. We will also have to specify the length and 15 characters is long enough for the sport with the longest name.

We have now considered all our entries, so our list now looks something like:-

FieldName	Type	Elements	Length
Name	Alpha	2	20
Address	Alpha	4	30
PostCode	Upper	1	9
Telphone	Upper	2	18
MemNum	Serial	1	5
SubPaid	Fixed	1	6
DatePaid	Date	1	
Interest	Constant	7	15

You will see that we have selected our names for the FieldName with some care. All names have to be no more than 8 characters long and we aren't allowed any spaces. We have also chosen names that start with different characters, as this makes life easier when we enter up information at a later stage, but have still managed to make them meaningful. We could have used words starting with the same characters, but it would have just made a little more work later.

You will also see that we have now included a field called 'MemNum' and that this is of type Serial. This means that AtLast will automatically create our membership numbers for us, incrementing the number by one for each new record.

Next, we ought to consider how to select/sort our information in more detail.

Name - We have already decided that our name is in two elements with the initials/forename in the first element and the surname in the second, but we want to sort it with the surname first, so we shall specify Name[2] before Name[1] in our Index list. Another thing we should consider is whether we want to 'modify' the information that will be stored in our name index, so that it is more efficient. One possibility is that we store the names in the index in upper case (this does not affect the way it is stored in the data file). The advantage of doing this is that it avoids the problems you come across when sorting names that may contain upper case letters in the middle (such as MacLaren), which would otherwise be sorted out of order, so we shall select this option.

Membership Number - This is straightforward, as there is only one element and numeric fields may not be 'modified' in any way. We shall store the next Serial number in a field in the System record called 'NextNum'.

Our list of items we would like to sort/index on now looks like:-

```
IndexName   Field[El]     Func

MemName     Name[2]       1
            Name[1]       1

NumMemb     NextNum
```

At this stage, we are now ready to start entering information into AtLast, as we shall sort out how the information will be displayed/listed after we have defined our database.

Entering the Definitions

We shall assume that you have already created your working copy of AtLast, as described in 'Getting Started'. If not, then you should not continue with the tutorial until such time as you have done, as you should never use the original disc for a working database.

Load AtLast as described in 'Getting Started' and select *D* from the Database Manager Menu. This will take you to a screen where you will be asked to enter the drive that you require. You should now enter the drive that you want to store the database on and press the RETURN key.

Note: You should **NEVER** select a RAM drive (drive M: on a PCW) to be the drive on which the definitions/data will be stored. If you forget to copy the files back onto a proper disc afterwards, all your definitions will be lost.

Under the heading **Database:** enter *CLUB*, as that is the name that we are going to give our database and press RETURN once more. After a second or two,

whilst AtLast searches the drive to see whether the database already exists, the message, **Unable to open CLUB.DEF ... Define New Database (Y/N)?** will appear. this means that it hasn't found a database of that name. As we do want to define a new database, press *Y*.

The screen will clear once more and the File Definition screen will appear. Beneath the heading **FileName** will be the word **SYS-CLUB** with the cursor round it. The cursor in most AtLast screens consists of a pair of square brackets, and it might also appear in inverse, depending on the model of computer being used. 'SYS-CLUB' is the name of the System file that will eventually hold our Constant and Serial data and before doing anything else we ought to define the fields that will eventually hold these values. Press RETURN to select SYS-CLUB.

The Field Definition screen will appear. This will be largely empty, apart from a column of numbers down the left hand side and a heading across the top. The cursor will be in the **FieldName** column and row 1. Enter *Sports* and press RETURN. This is the field that will contain the names of our sports and we shall always display them in upper case, so type in *U* and press RETURN. The rest of the word 'Upper' will then fill the space and the cursor will move to the next column. This is the first example you have seen of AtLast's attempts to help you. Because it knew what sort of information it should receive, it could select the correct entry, once it had sufficient characters.

We have already decided that there are four possible sports, but this might grow to seven, so we shall enter *7* under **Elements**. Enter *15* under **Length** as we decided that was sufficient and press RETURN. From now on we won't bother telling you to press RETURN every time, only when necessary, or if some other keys has to be pressed.

The cursor will now have moved down into column 1 of the next line, ready for the next field entry. Note that the right hand column **Size** has been filled in for you by AtLast.

You should now enter the details for the associated field that will hold the next number to be used as a membership number, so enter *NextNum*, *I*, *1* and *5*, to name of the field and define it as being of type Integer, having one element and being five digits long. The cursor will have moved across to the **Sign** column this time. Press RETURN to confirm the default setting of 'N', as we don't want negative membership numbers!

As we have no more entries to make in the SYS-CLUB file we can now press EXIT (ESC) to indicate that we have finished. We will now return to the original screen, but with the cursor beneath the **Can Edit** column. We needn't concern ourselves with these entries at the moment, so just press RETURN and the cursor will move down into the first column on the next line.

We have now got to define the main file that we will use for the storage of all our data and we shall call this 'Members', so type in *Members*. The screen will be replaced with another Field Definition screen and you can now start to enter the details that we decided on when designing the database. Start by entering *Name, A, 2* and *20*. You can then do the same for 'Address', 'PostCode' and 'Telphone' using the values we decided on earlier. When you enter up the details for 'MemNum', you will find that once you have specified 'Serial', it will automatically fill in the **Elements** column and move to the **Field[El]** column. AtLast is now asking what field in the System file we wish to relate the Serial field to. We named the field in the System file 'NextNum' and if you look at the bottom of the screen, you will see that the available names are displayed for us, in case we can't remember. Type *N* and press RETURN and AtLast will fill in the rest of the word for us. Press RETURN once more, to confirm the choice and the cursor will move down for the next field to be entered.

You should now enter the details for 'SubPaid' and this time you will find that the cursor moves to another different column, the **Sub-length** column. AtLast is now waiting for you to specify how many decimal points you want, so enter *2* and accept the default value 'N' for the sign, as we don't want any negative value subscriptions either!

Fill in the entry for 'DatePaid', using the 'Date' type, *1* Element and AtLast will pass over the **Length** and go straight to the **Sub-length** column. This time it wants to know how many digits you want in the year. The bottom of the screen will be prompting you for '2' or '4'. Select *2*.

You can now enter in the last field, which is the Constant field and you will find that yet again AtLast will pass over the invalid columns and stop in the **Field[El]** column. This time, if you look at the bottom of the screen, you will see that the System field name, 'Sports' that we entered earlier, is displayed. Type *S* and press RETURN twice, to select it and confirm it.

You have now finished defining the fields, which wasn't too painful, so now press EXIT (ESC) to say you have finished.

Index Definition

The right hand side of the screen will have cleared when you pressed EXIT (ESC) and will have been replaced with new headings and the cursor in the first column, **IndexName**. Enter *MemName*. the next column is **Duplicates** and AtLast is asking whether duplicates should be permitted. As it will be quite acceptable in this case for there to be two or more members with the same name, you should type *Y* to change the setting. Next you are asked for the **Field[El]**, which is the field that you want the index to be created on. Type *N* and press RETURN.

Now you know why we chose to give all the field names different starting letters, as it has filled in the remainder for you. Press RETURN once more, to confirm the choice and the cursor will move to the **[El]** column. The program is now waiting for you to say which element of 'Name' you want, so press *2* to select the second element. Press RETURN to select the offered length and then select *1* to tell AtLast that you want it to save the index as upper case.

The cursor will now have moved down to beneath the 'Field[El]' column again, waiting for the next part of the index to be given. Press *N* again and confirm it and then *1* to specify that you want the first (forename) element of the 'Name' field. This time, it will tell you that you can have a length of '5'. this is because the maximum length of an index entry is 30 and we have already used 25. In this case, it is alright, but if it hadn't been, we could have backed up to the previous level, using the Up Cursor and altered the value of the first entry, so that they were more balanced. Select *1* again to make the entry upper case.

Once more, the cursor will have moved down beneath the **Field[El]** column, but this time we don't want to enter a third index element, so we can just press the Down Cursor key to move on. This time, the cursor will move back to the **IndexName** column so that we can enter another Index name. You can now type in the entry for the 'NumMemb' index entry, but select *N* in answer to **Duplicates** as we don't want to have more than one member with the same number. This time we want to index on the 'MemNum' field, so type that in now and confirm it.

As we don't have any more Index entries to make we can simply press EXIT (ESC) twice to indicate that we have finished and once more we shall be taken back to the File Definition screen. Once again we shall just accept the default entries for **Can Edit, Can Delete** etc. As we may want to alter data later on. The 'Advanced Use' chapter gives details on how to make use of these options to protect data.

Finally, press EXIT (ESC) again and this time a menu at the bottom of the screen will prompt you to **Save, Edit, List, EXIT?**. Type *S* to save the database definitions, then EXIT (ESC) to indicate you have finished with this part of the program.

You will now find yourself in a new menu, the Database Definition Menu. This is the menu that will normally greet you when you load an existing database and want to change it in some way, but which was bypassed when you first started this session, because there was no existing database.

You have now finished defining your first database, but there is still one more thing that you have to do before you can make full use of it. You have still got to create the forms that we decided are required and to do this, you should type *F*, to select the **Form Definitions** option.

Defining Forms

Creating forms is a bit like using a word processor in many respects, but with the difference that whereas a word processor will normally only accept text that you type in, when creating a form we can also insert 'field markers'. Field markers, which are inserted by pressing the PASTE (F0) key, have the effect of inserting the current contents of the field into the text when we later display it.

We have already decided that we will need four different forms, so it is time to start on creating the first form. If you haven't already chosen the **Form Definition** option by pressing *F* from the Database Definition Menu, then you should do so now. The screen will clear and at the bottom will appear the request, **Get (a form), Create (a new form), EXIT?**. For now, select 'Get' by typing *G*.

You will be informed that there are **Forms available:** and beneath it will be the names of the available forms. In this case, **MEMBERS** and **SYS-CLUB**. These are the 'default' forms that AtLast automatically creates when we define a database. It will create one form for every file in the database, with the exception that it will only create a 'SYS-????' form if there are any fields actually defined in the 'SYS-????' file.

If you select the **MEMBERS** form, by typing *M* and pressing RETURN twice (to select the full name and then confirm the choice), you will be presented with a screen appearing much like the following example:-

```
Name     : Name[1]_____________ | Name[2]_____________
Address  : Address[1]_____________________ | Address[2]_____________________
           Address[3]_____________________ | Address[4]_____________________
PostCode : PostCode_
TelPhone : TelPhone[1]_______ | TelPhone[2]_______
MemNum   : MemNu
SubPaid  : SubPai
DatePaid : DatePaid
Interest : Interest[1]____ | Interest[2]____ | Interest[3]____
           Interest[4]____ | Interest[5]____ | Interest[6]____
           Interest[7]____
_____________________________ End Of Section _____________________________

Form: Members       File: Members   Section: Body

Edit, Head, Body, Tail, Save, Get, Delete, Rename, Create, EXIT?
```

This is the default screen created by AtLast for the members file and if you study it, you will see that the left hand side of the screen consists of a list of the fields that we defined. This is a text description. To the right of the colon (:) following the description, you will notice that the field name is repeated and in the case of those fields that have more than one element, it is repeated once for each element, followed by its element number in square brackets. These are field markers and if there are more than one on a line, they are separated by a short vertical line.

You will also notice that some of the field markers are abbreviated, whilst others are followed by an 'underline'. The reason for this is that the field markers are used to show the length of the fields, so small fields are abbreviated, whilst longer ones are padded out with underlines.

Looking at the form, it is obvious that it would make a good starting point for our entry form that we will need for entry of data, but the layout, whilst being functional, could be made neater for our use (AtLast doesn't know how we want it to look, so it always creates forms in this layout).

This is the 'Body' of the form, as indicated by the **Section:** heading on the Status line. This is the main part of a form and is always displayed when we choose to Add or Scan information. The other two parts of a form, the 'Head' and 'Tail' are only used when we List. As this form is primarily intended for use when Adding or Scanning records, we shall ignore them for now.

Type *E* to select the **Edit** option that is displayed at the bottom of the screen. This will move the cursor to the top left hand corner of the screen and it will now appear as the normal cursor that we come across in most programs, rather than the 'square bracket' cursor we have become used to in AtLast. If you now use the Right Cursor key to move the cursor across the screen, you will find that as soon as it comes to the first field marker, the cursor will change to a highlight of the size of the field. If you now look at the bottom of the screen, you will see that the bottom line menu now says **Field Marker : Edit, Nudge, Delete, <Cursor>?.** These are the options that are available when you are in a field marker. For now, just use the Right Cursor key again and you will find that the original small cursor reappears immediately after the field marker. This is because, as far as AtLast is concerned, a field marker is a single character.

How can we improve the appearance of the form? Well, the name might look better if it was on two lines. The first element should have its description changed to 'Forename' and the second element to 'Surname'. Similarly, the address field elements would look better on separate lines. You might also think that the 'Telphone' field would look better on two lines with 'Home' and 'Business' descriptions. For now we shall sort out the name and address fields and you can then experiment with any further changes you want to make.

Press the ALT-EOL (F7) key now and the cursor will return to its starting position at the top left of the screen. You can now type in *Forename*. You will notice that it overwrites the existing description. By default, AtLast is in overwrite mode. Carry on moving the cursor across the screen with the right Cursor key, through the first field marker, until it is positioned over the short vertical line between the field markers, then press DEL→ (CLR) until the cursor moves up to, but not into the **Name[2]** field marker. Press LINE (F4) now and you will find that the line has been split at the cursor position. If you now press RETURN you will find that the cursor moves down to the start of the next line, but without inserting a new line.

Try typing in *Surname* and you will find that the cursor will soon move into the field marker, whereas we really want to move the marker across so that it lines up with the first element. Press *N* and you will find the field marker moves across one space, so that we are once more out of the field marker. The *N* is one of the Field marker editing commands and is for just that purpose (to move the field marker one column to the right). We could keep doing this for each character we still need to enter, but it is more convenient to toggle into 'Insert' mode by pressing *[+]/[-] (F9)*, so do that now. You can now type in the rest of your 'Surname' description, followed by the colon and spaces until the field marker lines up with the first one. You can toggle in and out of insert mode as you require, with the [+]/[-] (F9) key. The current mode is always displayed on the Status line.

You can now modify the address fields in the same way and also the 'Telphone' fields. Don't forget to alter the description to 'Telephone' whilst you are doing it and you can also change any other descriptions you want.

The form might look a little better if there was a blank line between 'Telphone' and 'MemNum', so move the cursor down to the 'MemNum' line and position it in column 1, then press *LINE (F4)* and the whole line will move down, leaving a blank line. Do the same with 'Interest', so that the interests are also separated from the details of the member by a blank line.

Once you have finished altering the form, you can press EXIT (ESC) to say you have finished and the original editing menu at the bottom of the screen will re-appear. You should now type *S* to tell AtLast to 'Save' the form, then, once the **Saved** message appears, press EXIT (ESC) again, to say you are finished with this form. You will now be back at the Database Definition Menu.

We shall now create the 'list of member names and telephone numbers' form, but this time we shall use the 'Create' option instead. Select *F* for 'Form Definitions' and this time *C* to create a new form. We are asked to **Enter Form Name**, so we should think of a suitable name for the form. Type in *ListMembers* and press RETURN (you will notice that with Form names we are not restricted to only eight characters) and AtLast then asks us to **Enter File Name**. We should specify

the name of the file from which we want AtLast to take its information (or for that matter, what file we want it to put new information into!) and in this case, as we only have one file, apart from SYS-CLUB, you can type in *M* and press RETURN twice. If you look above the status line, you will see a list of available file names.

This time, an empty form will appear on screen, with the cursor in the top left corner. We don't want any descriptions in the body of the form this time, as we are creating a form to make a list of details. Instead, we shall eventually put the headings for the columns into the 'Head' of the form.

Press the *PASTE (F0)* key and immediately a question mark (?) will appear and at the same time, the 'Field Marker' menu will appear at the bottom of the screen. The '?' is the default field marker that AtLast uses when it doesn't know which field you wish to use. Normally, it assumes you want the next field in the list, after the one you last used, or else the last field you deleted (if you had deleted one), but because you haven't yet used one, it does not know.

Type *E*, to 'Edit' the field marker and at the bottom right of the screen a list of the available fields will appear. Type *N* to select the 'Name' field and then confirm it, as usual. You will see it displayed beneath the Status line. Now type *2*, to say you want the second element. You are now asked whether you wish to **Omit leading and trailing spaces?**. If you answer 'N' to this, then when the form is listed, all fields will be displayed to their full length, as specified in the field definitions. this would not be very good if we wished to have the forename followed by the surname, say in an address label, as it would appear something like 'John Smith', in which case you would select 'Y' and AtLast would then remove all spaces at the start and end of each field, but in this case, we wish to keep our list all tidily lined up, so press RETURN to select the default **N** value.

You can now move out of the field marker by pressing the Right Cursor key and you will find the '?' marker replaced by the proper field marker. Move the cursor across a space and position another field marker with *PASTE (F0)*. Note that this time AtLast gives you the next field it knows, 'Address[1]', which in this case is not the one we want, so press *E* again, to 'Edit' the marker and clear the 'Address' name with ALT-DEL→ (CTRL-CLR), then type *N* to select the 'Name' field again and this time, select the first element. Repeat the process for the first telephone number element.

We now have a problem, as there is insufficient room for the second telephone number on the same line as the rest, so this is where we must make use of the facility to tell AtLast to treat separate lines as one line when printed. We do this by typing a backslash (\) at the end of the first line, so move the cursor across again and type \. You can now press RETURN to move to the start of the next line. Enter the field marker for the second telephone number now.

Note: on the PCW range of computers, the backslash symbol is obtained by pressing EXTRA-½, next to the normal slash (/) key.

We have now entered all the information that we require in the list, so press EXIT (ESC) to finish editing the body of the form. We now want to create some headings for our form, so select *H* for 'Head', from the bottom line menu and a blank form will appear again. If you look at the Status line, you will find that it now shows **Section:** as **Head.**

All you have to do is type in suitable headings spaced appropriately across the screen and continue on the next line, having used the '\' at the end of the first line. when complete, press EXIT (ESC) then use *S* to 'Save' the form.

That completes the second form and by now you should be quite able to produce the other two forms with little trouble. A suggested layout, with a few notes, is shown for each of them:-

```
"Name[1]_________________","Name[2]_________________",\
"Address[1]_________________","Address[2]_________________",\
"Address[3]_________________","Address[4]_________________",\
"PostCode_"
_____________________________ End Of Section _____________________________
```

This is the *Export* form, for exporting data to other programs that require 'comma-delimited' fields. The following points should be noted. The commas, quotation marks and the backslash are typed in as text, but the remainder consists of field markers. If you want all fields to be a fixed length, then you should make sure the first field marker is set to 'N' for 'Omit leading and trailing spaces'. If you want the minimum size, then select 'Y' to this option. After setting the first field, all others will default to the same setting, so you can quickly select each field marker and accept the default name and setting for each subsequent field marker. You can add as many field markers as you want, making sure to finish each line (except the last line) with a '\'.

The following is a suitable form for labels:-

```
Name[1]_________________ Name[2]_________________
Address[1]_________________
Address[2]_________________
Address[3]_________________
Address[4]_________________
PostCode_

_____________________________ End Of Section _____________________________
```

The 'Labels' form contains nothing but field markers and the 'Omit leading and trailing spaces' option is set to 'Y', so that there will be no unsightly gaps when printed. Note that in this case, we have positioned the field markers part way across the screen. This was done so that the form could also be used for addressing individual envelopes as well. If your printer will not allow you to position the labels across the screen, then you should set the field markers nearer the left hand side.

Create these two forms now and then press EXIT (ESC) after saving, to escape back to the Database Definition Menu and you have now completed defining your first database with AtLast, so press 'Q' to quite the definition program.

In the next chapter, we will look at how we may use the database, now that we have created it.

7. 'USING' TUTORIAL

Now that we have defined our database, we can start to enter data into it, so type *A* from the Database Manager Menu, to access the database.

If we have just returned from defining the database when we select this option, then AtLast will assume that we want to use the database that we have just defined, so it will automatically load it for us. If we have only just loaded AtLast, then we will be prompted to specify the drive and the name of the database we want to use.

The next thing that will happen is that a message will appear, telling us that we must initialise the System record using the Scan/Update option. This is because AtLast knows that we have specified a number of fields in the SYS-CLUB record. If we hadn't defined any, then AtLast would not have bothered to warn us.

Filling in the System Record

We can now select what we want to do, from the Database Access menu and, as we have just been warned, we should select the **Scan/Update** option so that we can initialise the System record, so enter *S* and we will be informed that a number of forms are available (the ones that we have just created), including one called **SYS-CLUB**. This is the default System Form, so press *S* and then RETURN, twice, to select and confirm the choice.

The screen will clear once more and a 'blank' form appear, complete with a Status line towards the bottom of the screen and beneath that a menu of available commands. When we are using 'Scan/Update' normally, the options that are available will change according to the situation and will only show those commands that are valid at that time. At the moment it says, **Form, Edit, Print, EXIT?**. We want to 'Edit' the form, so type *E* and the cursor will move to the first field marker in the form (Note that when we were creating the forms, the field markers were displayed, but when we are using them, only the 'text descriptions' are displayed) and it will appear as two square brackets marking the limits of its length. It may also be highlighted, depending on the model of computer being used.

In this form, the first field is the first element of the 'Sports' field. Enter *Badminton*, *Squash*, *Tennis* and *Swimming*, into the elements, ending each with RETURN. Finally press *Down Cursor* to indicate that you have finished with this field.

The cursor will now have moved down to the field headed **Number**. This is the Serial field that will normally hold the next value for our membership number, so type in *1* as we want our numbers to start at the beginning.

We have now completed entering the data for the System record, so you can type *S* to save the record,or use either 'R' to Re-do the form, or the Up Cursor key to move back up from the bottom of the form, making whatever alterations are necessary, before saving it. Press *EXIT (ESC)*, once the record has been saved and you will return to the Database Access Menu.

Adding Records

We are now ready to enter up a few records into the database, so press *A* to 'Add Records in batch', but this time, when asked to select a form, specify the **MEMBERS** form. As you are going to type in the records at the keyboard, select *K* for keyboard input.

You can now enter up the details for the first member. You will see that the membership number contains the number '1'. When you enter the date, enter *1/1/87*. You will find that AtLast will automatically add the required leading zeroes to single character months or days parts of the date.

When you come to the 'Interest' part of the form, if they don't have any interests, you can simply press *EXIT (ESC)* to indicate that there is no more to enter, or type sufficient letters of the interest to make it identifiable. Type *S* to select 'Swimming' and press RETURN. If you now look at the entry, you will find that AtLast has selected 'Squash'. This is because both 'Squash' and 'Swimming' start with an 'S', but we originally entered 'Squash' first. Press ALT-DEL→ (CTRL-CLR) to clear the field and this time, type in *SW* before pressing RETURN. This time, AtLast will select the entry we want, so press RETURN again to confirm the entry. If the member has more than one interest, then you can now enter them as well. When you have no more to enter, press *EXIT (ESC)* to say you have finished, then type *S* to save the record.

Note: You might have found it useful to have entered the names of the various interests onto the form, as a reminder of the valid entries. If you want to do this when you have finished entering records, you can go back in to the Form Definitions, 'Get' the form and add the required text, before re-saving the modified form. You can alter forms in this way at any time.

The form will clear and you are now ready to enter the next record. Enter up five or six more records now, giving them a variety of different interests. When you have finished, press *EXIT (ESC)* once more and you will return to the Database Access Menu.

Scanning the Database

Now that we have a few records in our database, it would be a good idea to have a look at the different ways in which they may be scanned and altered.

Type *S* to 'Scan/Update' the database and you will be asked which form you want to use. Select the 'Members' form once more, by typing *M* and confirming it. The chosen form will be displayed and you will now be asked **Which Index order?**, with a list of the available Index names on the line above. Type *N*, to select the 'NumMemb' index and confirm the choice. The cursor will now appear in the first field element that was specified in the index. In this case, there is only one and it is the membership number field. Type in *3* and press RETURN.

The form will then fill in with the record for the member with a membership number '2'. Press *N*, 'Next' and the next record will appear. Press *B* for 'Back' and the previous record will appear. You will see that the order in which they are displayed is the membership number order, rather than the order you entered them. This is because they are in the order set by the 'NumMemb' Index.

Type *I*, to select the 'Index' command and this will allow you to select a different index order for the records to be displayed. Select the 'MemName' index. Nothing will appear to have changed, but if you now use 'B' or 'N' to move backwards and forwards through the records, you will see that they are now ordered around the member's name instead of the membership number.

We now decide that we want to look for a member's record, or a group of members and there are two ways that we can do this. If we just want to browse through the members with names beginning with 'S' for example, we first of all make sure that the index is 'MemName' and then press *S*, to 'Search'. The form will clear and The cursor will be positioned in the first element of the 'MemName' index, which is the Forename element. Press RETURN, as we want to search on the surname (RETURN will make AtLast start searching at the beginning of a field), then type *S* in the Surname field and press RETURN. The form will fill with the record for the first person whose name begins with a letter 'S'. You can then use 'N' to browse through the rest of the members beginning with 'S'.

Secondly, we can scan a group of records by setting 'Conditions'. Type *C*, to select 'Conditions' and the screen will clear and be replaced with the 'Conditions Screen'. Setting 'Conditions' lets you specify certain conditions that must be met for a record to be displayed. One advantage of this method of searching is that it is not necessary to use a field that is part of the current index, or even an index at all. We want to see all the members whose interest is 'Swimming' AND whose name begins with 'B' in this example. The cursor is positioned under a column called **Field** and at the bottom of the screen is a list of all the fields in the data file. Type *I* to select 'Interest', then *0* as 'Swimming' could be any entry in the 'Interest field' elements and you are then asked to enter a comparison operator and the available options are shown at the bottom of the screen. Type *EQ*, for 'EQuals' and then you must enter the value that you want it compared with. In this case, we want 'Swimming', so type *Swimming*.

We now want to specify the initial letter of the member's surname, so enter *N* and *2* to specify the second element. We want it to be either equal to or after the letter 'B', so select *EA* and *B* for the condition value. That would select all names from 'B' onwards, so we must also specify where it is to stop, so select 'Name[2]' again and this time specify *BF*, indicating that it must be before 'C'. If you now press *EXIT (ESC)*, you will return to the original scanning screen, complete with the current record displayed. If this record does not meet the condition, a message will tell you. You may then either start the search from the beginning by using 'Search', or continue using 'B' and 'N' to browse through.

You will now find that AtLast will run through all the records until it finds one that matches the Conditions that you set. Pressing 'N' or 'B' again will continue the search. These conditions will stay in force until either you cancel them, or you leave the file you are using. To cancel a set of conditions, or alter existing ones, you simply press *C* again and then edit the existing conditions. If you want to remove all, or some, of them, then position the cursor on the unwanted entry and press the ALT-LINE (F5) key. When finished, EXIT (ESC) back to the scanning screen.

Whilst we are in the Scan/Update screen, we can also Add new records, Edit existing records, or Delete records (depending on the settings specified in the definitions, regarding adding, altering and deleting). Typing *A* will give you a clear form in which you can enter the new record, whilst *E* will move the cursor into the existing form and you can then move around it using the Up and Down Cursor keys, altering entries at will. Selecting *D* will result in you being asked to confirm that you want to delete the form before it is erased from the file forever.

There are two further commands available when using the Scan/Update option. The first of these is 'Form'. Type *F* and the available forms will be listed. Type *L* and confirm. The form on screen will change to the 'Labels' form, containing the details from the current record. This provides an ideal way to switch between different views of the same record, frequently with some of the forms only showing selected data, as in the case of 'Labels'.

As we now have a form on screen suitable for printing an address on an envelope, it is time to try the last command. Type *P*, to select 'Print' and the form will be replaced with a list of **Print Parameters**. If the default settings are ideal for printing on 11" continuous stationery, but if we want to address one or two envelopes, we should elect to **Change Parameters** by typing *Y*. The cursor will move into the parameter list and you can then move it down to the second line and change the **Single Sheet Paper, Use Form Feeds** and **Page Each Record** settings to *Y*. This has the effect of ensuring that when an address is printed, the envelope will be ejected.

You may also select between sending the output to the Screen or a disc File. the default setting is 'Printer', but may be changed by typing either 'S' or 'F'. The only time that printing to the screen may be found of any great use is when you are using a form that is too long to be displayed on the screen at one time. You could view the full screen by using the 'Edit' option and scrolling the form with the Cursor keys, but it may also be viewed by printing it to the screen.

That completes a run through most of the facilities available in the Scan/Update option and we recommend that you should experiment with all the options until you are happy that you understand them.

There is one more way that we can view and print the information in the database and that is by making use of the 'List Data' option from the 'Database Access Menu, which will be described next. For now, when you have finished experimenting with the database, press EXIT (ESC), to finish Scanning the database.

Listing Data

This facility may be used for many purposes, but the main use is to enable you to produce listings of your data.In the previous section, we looked at records as individual items. Listing looks on them as being part of a group and allows you to print all records, or only selected records in the form of lists.

Type *L* from the Database Access Menu and you will be asked to specify the form you want to use. Select the 'Memberlist' form by typing *ListMembers* form in the usual way and select the *MemName* index, as we want our listing in alphabetical order.

We are now presented with a screen containing a Status line at the top and a list of available commands at the bottom. Many of the commands will be familiar and serve the same purpose as in the Scan/Update option, but some are new. The 'Ascending' and 'Descending' commands let you select whether the listing is in the normal Index Order, or in reverse order. As the default setting is 'Ascending', we don't need to change the order because we want the list in alphabetical order. We don't need to set any conditions, as we want to list all members. Neither do we want to specify a 'Range', for the same reason (Range allows us to specify the starting point in the Index from which records will be listed and also the ending point).

All we need to do now is 'List' the file. This is virtually the same as the print option in Scan/update and we are offered the current print settings and the opportunity to alter them. We want to send the listing to the printer, so, unless any of the defaults have been changed since AtLast was loaded, you can press *N* to indicate you don't want to change the parameters. In this case, we don't want to print the 'Selection conditions', so you can answer *N* to that also.

At this point, you should make sure that the printer contains suitable paper and that it is 'On-line', before pressing a key to list the data to the printer. As each record is sent to the printer, the contents of the index fields will be displayed on screen, so that you can watch progress (This also happens when a listing is sent to a disc file and is very useful as it is the only indication of what is happening). When completed, we are asked to press a key to continue and the Listing Screen will re-appear, so that we can carry out further listings if we want.

We recommend that you now experiment with different settings for Range, Conditions and forms, until you are happy you know how to set the various options.

By this time, you should be feeling at home with the way that AtLast is used and you should be ready to create and use your own database. The only main feature that we have not covered is the 'Tidy/Repair Database' option, but this is beyond the scope of a tutorial and is fully described elsewhere in the manual.

8. ADVANCED USE

1. Converting Data from earlier versions

This section describes how you may re-construct a database created with an earlier version of AtLast. Data files created with earlier versions are not directly compatible with this version of AtLast. The reason for this is that this version makes full use of 'variable length' records, whereas earlier versions used 'fixed length' records.

The procedure required is straightforward and revolves round the process of exporting all the existing data to a special disc file, using the earlier version of the program and then defining a new database with the new version and importing the data into it.

You should load the 'DBDEF' definition program from the earlier version into the computer, select the option to define a form and create a form that contains only the field elements for every field. These should be located on the left hand side of the form, with one field marker to each line. Edit the field marker so that the 'Omit Leading and Trailing spaces' option is selected. Save the form with a suitable name and EXIT (ESC) back to the opening menu.

Note: You should **NOT** enter field markers for any fields that will be 'Serial' in the new database, as they will be inserted by AtLast at the time the data is imported. If it is important that any existing Serial values in records are retained, then they may be exported by inserting the field marker into the form, **BUT** you must make sure that they are inserted into an 'Integer' or 'Fixed' field in the new database, rather than a Serial field. This Integer/Fixed field can be converted to Serial again, once the data has been transferred.

You should also make a note of the names of the various fields and their dimensions. In fact, the simplest way to do this is to select the 'Define database' option and then use EXIT (ESC) from the File Definition screen to bring up the options menu at the bottom of the screen. If you select the 'List' option, it will print all the field and index definitions in the current database.

Having done all the above, you can leave 'DBDEF' by the normal methods and then load 'DBUSE' and the database you wish to transfer. Once loaded, you should select the 'List Records' option, select the new form that you have just created and the Index that you wish to use. You should then press RETURN for both the 'First' and 'Last' requests for the Index, 'Normal' for the Index order and 'All' records.

You should then select 'File' for the destination of the listing and give the file a suitable name, including an optional drive identifier, but not a memory drive. The next thing you will be asked is the 'Page Length'. Specify a length of 1 and the same when asked 'Printed Lines per Page'. Answer 'N' to 'Form Feeds (Y/N)?' and also to 'New Page for each record'. The contents of the file will then be written to disc.

The procedure should be repeated for each file in the database, with the exception of the 'SYS-????' file. You should also obtain a print out, on paper, of a single record for each form, so that you can easily create forms in the new database with a similar layout. Finally, having exported all the data, you should select the 'Scan-Edit' option from the opening menu and select the SYS-???? form. This will enable you to make a note of any Constant values.

You should now close down 'DBUSE', reset the computer and load the new version of AtLast. You should then follow the procedures described in the chapter on 'Database Definition', to create your new database, making use of the details from the old version print out. Once you have defined the database, you should create your new Forms, using the Define Forms option. You should also create a form with fields in exactly the same order as the form that you created to export your data files to disc. Return to the Database Manager Menu.

Select the 'Access Database' option, then the 'Scan/Update' option, in order to enter up the values of any Constant fields. You should also set the starting value for any Serial fields (normally 1). Having done that, you should save the System record and return to the Database Access Menu using EXIT (ESC).

You can now select the 'Add Records in batch' option and select the form that you have created to import your data. You will then be asked whether you wish for 'Keyboard or File input?'. Select 'File' and give the name of the first of the files you want to import. AtLast will then read in all the data. Once complete, you can repeat the process for any other files you want to import.

If any of the imported data is found to be unsuitable by AtLast, it will pause inputting data and ask whether you wish to continue or abandon the import. Depending on the situation, you may decide to continue and subsequently amend the faulty data manually at a later date, or if things are obviously very wrong, you may decide to abandon the import and find out the cause of the problem. The most likely reasons are:-

* You have defined a new field of an unsuitable type for the data being imported
* You have incorrectly defined the Constant values. You may have unwanted spaces in the imported data.
* The number or order of fields differs in the import file from the number or order of fields in the form you are using to import the data

2. Re-organising a Database

There are a number of reasons why you might want to re-organise a database. It sometimes happens that, despite the best of forward planning, one outgrows the design of a database. You may decide that:-

* You need some new Forms
* You need another Index option
* You need extra fields
* You wish to delete a field
* The 'Type' you specified for a field is not the most suitable one for your purposes after all
* The length of existing fields is incorrect

In general, there are few restrictions on what can be changed in a database definition, but some alterations may require making further changes. Re-organising an AtLast database may be broken down into four different groups of procedures:-

a). Changes that require no other alterations.
b). Changes that may require alterations to Forms or Index files.
c). Changes that will require the use of the Tidy/Repair database option.
d). Changes that require existing data to be transferred from the existing database into the amended database.

Note: It is important that before you attempt to make any changes to the fields, you make a complete back up of all data and definition files (See Technical Summary for details of files that must be saved). If it does not work as you expect, you can then go back to your back up and carry out the changes by another method.

a). Changes that require no other alterations.

Edit/Add a New Form

New forms may be created at any time by going to the Form Definition option in the Database Definition Menu. The number of forms is limited only by the amount of disc space available. Unwanted Forms may also be deleted without restrictions.

Changing Field Types

Field types that may be freely changed without corrupting data:-

Alpha	to	Upper	- The case of existing data will not be changed.
Upper	to	Alpha	
Serial	to	Integer	- As long as the related field in the System record is of type Integer.
Serial	to	Fixed	- As long as the related field in the System record is of type Fixed.
Integer	to	Serial	- As long as the related field in the System record is of type Integer.
Fixed	to	Serial	- As long as the related field in the System record is of type Fixed.

b). Changes that may require alterations to Forms or Index files.

Changing Field lengths

The length of any type of field may be changed without any loss of, or corruption to existing records. It may be necessary to amend existing forms, to allow for any increase in field lengths, that might upset the existing layout and also any Index definitions that use the changed fields. If an index is altered, it will be necessary to re-index, using Tidy/Repair.

Note: In the event that the contents of any existing Alpha or Upper fields are longer than the new length of the field, the excess data will not be displayed in a Form and the surplus text will be discarded if the record is re-saved.

Adding New Fields at End

With this version of AtLast it is possible to add new fields (or extra elements) to the END of an existing database definition. This is done by selecting '(Re-)Define Database', then 'Data Definitions'. You should select the file and then, once the field definitions are displayed, you may move down to the bottom of the list and add the new field(s). The altered definition should then be saved. In order to make use of any new fields, you will have to modify your existing forms to include field markers for them.

Note: Any new records you enter will make use of the new fields, but existing records obviously cannot (as they don't yet contain any information). You may, of course, go through the existing records, adding the details, using Scan/Update. If you set a Condition of the new fields having a 'null' value, then only those records that require updating will be selected. The selected index must NOT be one that will be altered when adding new information, nor must it be the Physical Index, otherwise some records may be missed.

c). Changes requiring use of the Tidy/Repair option.

Deleting an Existing Field from End

In the same way that new fields/elements may be added to the **END** of a file definition, it is possible to delete a field/element, although there is not a lot of point unless you are short of fields, as the overhead imposed by having empty fields in a record is usually very small. Only the last field(s)/element(s) in a file definition may be deleted.

If a field/element is deleted, it is important that the Tidy/Repair option is used, with the 'Collect free space at end of file' option selected. This ensures that all existing records are updated.

Note: If the Tidy/Repair option is not used and subsequently a new field is added to the database definition, it may result in the new fields appearing to be corrupted, as they may contain all or part of the contents of the field that was deleted previously, but had not been removed from the data files by using the Tidy/Repair option.

Adding a New, or altering an existing Index

New Indices can be added to an existing file at any time, although it is preferable that you decide on the required Index files at the time you define the database, if only because it saves time if you get it right first time.

Note: An existing Index may also be altered in a similar fashion and will require re-indexing before use.

The procedure to create a new Index is straightforward. You should select the '(Re-)Define Database' option from the opening menu and then select 'Data Definition'. Select the file that you want to add the index to and then press EXIT (ESC) once the field definitions are displayed. this will take you to the 'Index Definition' stage and you should then move the cursor down to the first vacant IndexName, using the Down Cursor key. Enter the name and relevant information for the new index, as described in the chapter on 'Database Definition'. Finally, save the amended definitions.

You should then leave Database Definition and select the 'Access Database' option from the Database Manager Menu. Select 'Tidy/Repair Database' and move the cursor down to the file that you wish to re-index. Select 'Y' and then move across to the new Index name, which will be displayed and select 'Y' for this also. You may then press EXIT (ESC) and will be asked whether to 'Do Repair'. When you are sure you have made the correct selection, enter 'Y' and AtLast will then work its way through the data file, creating the new index.

d). Any Other Changes

Any other changes may still be carried out to an existing database, but the procedure to use is different and revolves around the principle of exporting the existing data to a temporary file, making the necessary alterations to the database definitions and then importing the data into the modified database.

There are very few restrictions regarding what you can change when using this method. All field Types can be changed for any other.

Note: Before making any changes, do not forget to make a back up of both the data files AND any Definition, Form and Index files that relate to the file to be changed. This is important as it will be necessary to remove the existing data files from the disc before the exported data is imported back into the database, in order to avoid a possible clash of filenames with existing data when the new files are created.

Having safely stored away a complete copy of the database, you should use 'Form Definition' to create a form that contains only the field markers for those fields that you wish to transfer to the new database. Each field should be on a separate line and positioned on the left of the form. The form should consist only of a Body, there should be no blank lines and there should be no Head or Tail.

Note: Serial fields should NOT be included in this form as they must not be exported, as AtLast will add them when you import the data back. The only exception to this is where you want the contents of a Serial field to be inserted into a non-Serial field (say, an Integer field) in the new database, in which case you should include it in the form.

You should then use the 'List Data' option to create a disc file holding the data. This is done by selecting the file you wish to alter, then the form that you have just created. Select a suitable Index and 'Ascending' order and then select the 'List' option. Change the 'Print parameters so that the listing will be to 'File' and give it a suitable filename. You should select 'N' in answer to 'Use Form Feeds' and select a page length of 1 and the number of lines per page to 1 also, so that there can be no gaps between records and select 'N' to the 'Page Each Record' question.

Having set the Print Parameters, you should answer 'N' to the question 'Print selection conditions' and the data will then be automatically written to the disc file. Once complete, you should return to the Database Manager Menu.

If the amount of information to be exported is larger than the capacity of the disc drives to cope, then it is possible to export the data in a number of files by exporting the data in parts, using the 'Range' option in the Listing screen to select only a part of the Index for a file, then repeating the process with another part of the Index. If, for example, you had a large name-and-address file to transfer, then you could select just those with surnames beginning with the letters A-D, followed by E-H etc.

You can now go into the Define database option and make whatever alterations are required to the file definitions.

Note: It is usually better to add new fields to the end of a record definition rather than insert them in the middle of existing records. (This is because AtLast refers to fields, internally, by their number rather than their name.) Otherwise you may have to re-edit the index definitions and all the existing field markers in your forms and not just the ones you have changed.

If necessary, you should also modify any forms that may have been affected by the changes to the fields. This may be necessitated by any changes in the length of existing fields and by the addition of new fields that will also need to be displayed in a form.

At this point, you should delete any data and index files (from the existing database) from the disc. See the Technical summary for details of which files. If you do not do this, then, if your modified database uses the same file names as the original, which it most probably will, AtLast will attempt to open them instead of creating new files.

You can then use the 'Add Records in batch' option on the new database, select the form that you used to create the export file and read the data back in by selecting the 'File' option and specifying the name.

Although this method may seem clumsy and tedious, it is very powerful, giving you the option to change most field types if you wish (as long as the old and new types are compatible). For example, you could transfer numeric or Constant data to Alpha/Upper, or vice-versa.

3. Importing Data from Other Programs

Since most other programs, such as databases, word processors and spreadsheets have a facility to export data as an ASCII file, the method used to import data that was used in the previous section, may be used to import data from another program.

All that is required is to create a suitable form in the database, containing only those fields that are to contain the data to be imported, in the correct order. If the data to be imported contains a blank line between each record, you should ensure that a spare field in the database is included on the form at the end, or else one that is not being used for the import of data. In this way, the blank line (which will be treated as a field containing nothing by AtLast) will be read from the import file and entered into this field. This is important, as AtLast keeps a count of how many fields it has read in and would otherwise get out of step.

4. How to Relate Records Between Files

Most database programs are single file systems. That is only one record type is allowed and all the data conforms to that type. Other databases , including AtLast, are multi-file, in that several different record types are permitted in the same database. With AtLast, you may have up to ten such record types in one database, each type with its own data file and index files.

If you put common fields in different files then they provide a logical link between records in different files. For instance, consider a college in which each lecturer teaches a range of courses; suppose a maximum of five. We could set up a two-file system, one file for lecturers and one file for courses containing, at its simplest the following information.

lecturer	Course
Name	Title
lecturer number	Course code
Personal information	lecturer reference
Course References (5)	Description of course
	Other details

Indices would be created on:-

Name	Title
lecturer number	Course code

'Personal information' could be anything that was administratively useful, such as address, room number, phone number, pay details, and so on. 'Other details' could be hall number, time, day, duration and so on. The 'Course References' could be the titles of the courses or their reference numbers. The 'lecturer Reference', likewise, could be either the lecturer's name or their number. If there were few lecturers and all their names were unique, we could even set up their names as Constants. Likewise, with the course titles or codes.

(We could, of course, have set up a single file in which space was allocated in each lecturer record for all the details of all five courses, but this could get cumbersome if a lecturer only taught one course and will create difficulty in generating a list of classes. Alternatively we could have had a single file of courses, with all the personal details of the lecturer in each Course record, but this would have meant duplication of all the personal information in every course taught by the same lecturer. Suppose also we wished later to add a file of students with their personal details and references to the courses they took and possibly an extra reference to a lecturer who was their personal tutor. This would be next to impossible if we had not divided up the data to separate the lecturer data from the course data.)

Some databases offer a means to automatically manipulate the references between files in various ways, such as providing automatic links between related records and merging records from different files in reports. AtLast is not quite as powerful as this, but, with a little thought it is possible to effectively simulate the best part of this within AtLast.

You will have noticed, when using the Scan/Update option from the 'Access Database' menu, the facilities to (a) Search for a given record and (b) switch Forms and thereby switch files. This provides a means to achieve relational referencing. Suppose we have a Course record on the screen and we wish to contact the lecturer involved. We can see the lecturer reference and we know that it is also an indexed key to the lecturer file. Remembering the reference, we press 'F' to switch forms, select the lecturer form we need, select the index containing the reference field and type in the remembered reference. Up comes the lecturer record and we can read off his/her room or phone number to contact them.

Suppose now we wish to have a list of all the courses taught by a lecturer positioned below the personal details of the lecturer. If we were content to list the course references, this could be done simply by printing the lecturer record. If, however, we need more details of the courses to appear on the list, then we must find some way of extracting and merging the information from the Course records. This is where the powerful selection system comes into its own.

Firstly, we use the scan edit option to find the lecturer record. We print this out at the top of the paper, using the appropriate form, and specifying 'N' for 'Page each record'. The printer will stop on the next line of the paper. We now exit to the Access Database menu and select the List Database option. We select the desired listing form, choose whichever index order we wish to list in and the range and then 'S' to select. We enter only one selection condition, that the lecturer Reference should EQual the one for the lecturer we have printed out. Then when the listing is performed the list of courses with the relevant details we need will appear below the lecturer details.

Of course, relating records in this way is not limited to references between different files. It can also be used to good effect in relating records of the same type. In genealogy, for instance you may have only one file called People. The main index, of course, will be on their name. If you include a field each for mother and father, then these can provide relational references to other records in the People file which can be searched for by the index without even changing forms. To list all the children of a particular couple you select on a combination of Father and Mother fields.

5. File Protection

AtLast provides facilities to impose restrictions on the use of any file in a database. Any restrictions apply to a file and not the database as a whole, so that it is possible to have different restrictions for different files.

Each file can have three levels of protection and any, or all three, may be applied. The levels are:-

Can Edit - Alterations may be made to the contents of existing records.
Can Delete - Existing records may be deleted from a file.
Can Add - New records may be added to an existing file.

By selecting different combinations of these, it is possible to restrict the degree by which the file may be changed. For example, you might wish that the file may be viewed, but not altered in any way, in which case you would set the value for each of these to 'N'. On the other hand, you might want someone to enter in new records, but without the facility to delete or alter any existing records, in which case you would set 'Can Add' to 'Y' and the others to 'N'.

These levels of protection may be set when you have completed the definition of a file and AtLast returns to the 'File Menu', or they may be set/changed at any time by selecting '(Re-)Define a Database', 'Database Definitions' and specifying the database, after which the File Menu will be displayed.

In order to make sure that the settings cannot be changed by other people, you should make sure that once the database has been defined and is working as intended, the DBDEF.CHN file is removed from the program disc(s) that will be used by other people. Without this program file and its associated overlay files, it will be impossible to use the Definition option of the program to change the protection settings. If any changes are later required to the definitions, the DBDEF.CHN file and associated overlay files must be temporarily copied back onto the disc.

6. Printing

When printing or listing data to a printer, the default settings of the printer will be used, as far as typestyle, number of characters per inch etc. are concerned. Some reasons why you might want to change the settings are:-

* To select a Near Letter Quality style.
* To select Elite or condensed mode, so that more characters may be printed on a line if a wide listing is to be printed.

If you wish to change any of these, this may be done in a number of ways:-

a) Some printers allow you to set typestyles and sizes from their 'front panel' switches.
b) A number of Public Domain programs are available that provide the facility to change printer settings.
c) The CP/M SETLST.COM program may be used before loading AtLast, to send the required printer control codes to the printer.

Alternatively, you might prefer to send the listing to a disc file and then load it into a word processing program so that you can further process it and introduce different typestyles wherever you want and then print it from the word processor.

Note: The PCW computer printer defaults to an A4 'single sheet' setting. If any other stationery is to be used, the CP/M PAPER command should be used, otherwise it will override any settings in AtLast. The SUBMIT files that are created during Installation include the command:-

PAPER 11

which will set the printer to a page length of 11" and a continuous stationery setting. AtLast will then handle correct pagination itself whatever the length of paper or labels being used, single sheet or continuous. In general Form Feeds should only be used with single sheet stationery.

TECHNICAL SUMMARY

DATABASE DIMENSIONS

The following are the maximum permissible database dimensions.

Number of files per database	: 10
File size	: Limited only by disc size, upto 4 megabytes
Number of records per file	: 32767
Number of field arrays	: 20 (See Elements below)
Number of elements	: 99 per field array
Element size	: 79 characters
Number of Indices	: 5 (per file)
Number of Index elements	: 3 (per index)
Maximum Index entry size	: 30 characters (total of 3 elements)

DATABASE FILE LAYOUT

A database consists of a number of files which must all be present on the same disc. It is not possible to split a database over several discs. The database may, however, be on a different disc from the program files.

When AtLast is loaded, the 'specified' drive should always contain the program files.

AtLast creates a number of files to store database definition details and also Index details as well as the data. Listed below is a summary of the file types that are created and their purpose.

Every Database consists of three files which contain the data file definitions, the form definitions and an index to the form definitions. In addition, each data file that forms a part of the database creates an index file for each index that it contains. Each of these types of file has a filename extension that reflects its type or purpose.

```
dataname.DEF    Contains the file definitions for the complete database
dataname.FRM    Contains the form definitions for the complete database
dataname.FIX    Contains an index to the FRM file, to locate forms

filename.DAT    Contains all the records for a single file

indxname.IDX    Contains a single index for a file
```

'dataname' is the name that you specify for the database and there will only ever be one '.DEF', '.FRM' and '.FIX' file for a database.

'filename' is the name you specify for a data file within the database and there will be a '.DAT' file for each data file specified in the File definitions. There may be several of these in a database, including one called 'SYS-????', where '????' is the first four letters of the database name and constitutes the System file.

Note: If there are no fields defined in the System file, then it will not appear on disc as a file.

'indxname' is the name of an index as specified during Index definitions. Each index in a database will have its own '.IDX' file. there may, therefore, be a number of '.IDX' files.

If you wish to back up the data files in a database, then you should copy the 'filename.DAT' files that relate to that database and also all the 'indxname.IDX' files that relate to it. Frequently you would only have the data for one database on a disc, in which case you would copy all data using:-

```
PIP B:=A:*.DAT
PIP B:=A:*.IDX
```

to copy files from drive A: to drive B:. On a single drive computer, this will result in requests for the disc to be changed when required.

If you wish to back up all database definition files, then typing:-

```
PIP B:=A:dataname.*
```

will copy all files for that database from drive A: to drive B:.

FIELD TYPES SUPPORTED

Nine types of data fields are supported and the type selected determines which of the subsequent parameters are relevant. The following list describes each type of data field and the relevant following entries, together with the permissible range of values:-

Alpha	- Any letter, number or punctuation mark.	
	Elements	1-99
	Length	1-79
Upper	- Automatically converted to upper Case on entry.	
	Elements	1-99
	Length	1-79
Integer	- A whole number between -32768 and +32767.	
	Elements	1-99
	Length	1-6
	Sign	Y/N
Fixed	- A number with a fixed decimal point.	
	Elements	1-99
	Length	3-13
	Sub-length	0 to length-2
	Sign	Y/N
Real	- Any real number of up to 11 digits and in the range of 10^{-38} up to 10^{+38}. (Note, some rounding errors may be encountered).	
	Elements	1-99
	Length	7-17
	Sign	Y/N
Date	- Dates in the form DD/MM/YY or DD/MM/YYYY.	
	Elements	1-99
	Length	determined by program
	Sub-length	2 for YY or 4 for YYYY format
HMS	- A numeric field to hold Hours, minutes and seconds. Elements 1-99	
	Length	4-11
	Sub-length	3 for Mins or 6 for Min and Secs
Serial	- A numeric field, incremented by one for each new record.	
	Elements	1
	Field[El]	Field element from System file
Constant	- An item from a list in the System record.	
	Elements	1
	Field[El]	Fieldname from System file

INDEX FUNCTIONS

When an Index is created, the data stored in the Index may be modified in a number of ways, to increase the efficiency of the indexing. The available options are:-

0 - Unchanged. Sorting are case sensitive and in the order of the ASCII character set.
1 - Convert to Upper. All alphabetic characters converted to Upper case in index.
2 - Abbreviate. All characters not in the range 'A'-'Z' are ignored.

CONDITIONAL OPERATORS

The following are the conditional operators that may be used when Scanning or Listing a data file.

EQ - EQuals. The same as.
BF - BeFore. Lower numerically or in ASCII value.
AF - AFter. Higher numerically or in ASCII value.
NE - Not Equal. Not the same as.
EB - Equal or Before. The same as, or lower than.
EA - Equal or After. The same as, or higher than.
CN - CoNtains. Includes. Used with Alpha/Upper.
NC - does Not Contain. Used with Alpha/Upper.

PRINTER SETTINGS - DEFAULT AND RANGE

The default Print Parameter settings are:-

Destination	:	Printer
Single Sheet Paper	:	N
Use Form Feeds	:	N
Page Length	:	66
Printed Lines	:	58
Page Each Record	:	N

Allowable values are:-

Destination	:	Printer, Screen or File.
Single Sheet Paper	:	Y/N.
Use Form Feeds	:	Y/N.
Page Length	:	0-255.
Printed Lines	:	0-Page Length.
Page Each Record	:	Y/N.

SUMMARY OF SPECIAL EDITING KEYS

Under the CP/M operating system it is possible to re-define the keyboard to return whatever codes you wish. This is done using the CP/M 'SETKEYS' command and supplied with AtLast are two files, called 'ALKEYS.CPC' and 'ALKEYS.PCW' that may be used to re-define the keyboard to perform the following functions. The 'Code' and 'Key' columns give the values that are returned by default and these may still be used even when the definitions of the special keys have been defined, it preferred.

Command	PCW	CPC	Code(s)	Key(s)
Escape	EXIT (STOP)	ESC	27	^[
Enter	RETURN	RETURN	13	^M
Enter	ENTER	ENTER	13	^M
Left	←	←	19	^S
Right	→	→	4	^D
Up	↑	↑	5	^E
Down	↓	↓	24	^X
Tab	TAB	TAB	9	^I
Backtab	ALT TAB	CTRL TAB	2	^B
Begin	ALT EOL	F7	17 83	^Q S
End	EOL	F8	17 68	^Q D
BackSpace	←DEL	DEL	8	^H
Insert Mode	[+] or [-]	F9	22	^V
Delete Char	DEL→	CLR	7	^G
Delete To End	ALT DEL→	CTRL CLR	17 89	^Q Y
Insert/Split Line	LINE	F4	14	^N
Delete/Join Line	ALT LINE (CUT)	F5	25	^Y
Re-Margin	RELAY	F1	6	^F
Set Field Marker	PASTE	F0	23	^W

Note: **LINE is obtained by pressing SHIFT and the LINE/EOL key.**

Note: **The '^' before a character means that you should hold down the ALT (CTRL) key**

INDEX