



**USER
GUIDE**

5

6

7

1. New Features

Your copy of TRIMbase includes three additional features:

a) In the MAINTAIN module, the F6 key changes the typeface used to display variable names. You may choose from standard, bold, italic, underlined and faint. If your record card is complicated, this option gives you a convenient way telling apart values from names.

b) In the REPORT module, a report form which does not match the data file is not rejected outright as stated in section 5.11. You may still use it, but all the expressions will be replaced by # signs and you will need to re-instate them.

c) In the DEFINE module there is a security check. As stated under the 'conditions of purchase', you are encouraged to make a security copy of the TRIMbase system and use it for everyday work in preference to the original; but when you come to define a new card layout or modify an existing one you will be asked to load the original TALENT disk into Drive A to unlock this part of the system. A copy will not work.

The security system has been designed so that it will continue to work correctly even if the information on the TALENT disk is damaged. If it ever fails, TALENT will replace your disk immediately and without question.

2. File Recovery Program

In spite of every precaution, files are sometimes damaged so that they become unreadable. Typical causes are power cuts, lightning strikes and serious operator errors such as changing disks in the middle of a file update.

If your work is perfectly organised, you will always have back-up copies of all files, so that the inevitable failure will not cause you too many problems. For users who are not so far-sighted, TALENT provides a 'file doctor' - a program which will examine a corrupted file and recover as much information from it as possible: maybe the bulk of the file, maybe none at all.

The File Doctor costs £30, plus £1 postage and VAT. To buy a copy, please send us a credit card number (Access or Visa) or a cheque for £24.15. Allow 14 days for delivery except in urgent cases. Note that recovery is not guaranteed: sometimes a file can be so badly damaged that no information can be recovered at all.

3. Errata

a) In Section 5.7 (Selecting Records) the second sentence should read

"You can pick the records you want to be included by choosing the Record Choice menu and clicking on the Set Selection Rule option."

b) We regret that for reasons of Copyright we cannot supply the 'GEN' disk mentioned in Appendix 3.





TALENT Relational Information Management for the ATARI ST,
AMSTRAD PC and PC compatibles with GEM

TRIMbase was designed and programmed by Andrew Colin and
Ronnie Brown



TALENT COMPUTER SYSTEMS

Curran Building
101 St. James Road,
Glasgow G4 0NS,
Scotland.

Telephone: 041-552 2128
Telex: 77472 Answerback UNSLIB G

CONDITIONS OF PURCHASE:

The purchaser of any subsequent owner of this program is bound by the laws of Copyright, and shall make no copy, by any means whatever, of the printed manual.

The owner may make one copy of the software for security purposes only. A further copy may be made only if the existing copy becomes unusable.

The owner shall take reasonable precautions against theft or unauthorised copying of the manual and the software.

If the owner disposes of the program to a subsequent owner, he or she shall ensure that any copy made for security purposes is either passed to the new owner or destroyed.

TALENT believe TRIMbase to be free of major errors, but no guarantee is offered that TRIMbase is suitable for any particular application. In the event of unsatisfactory operation, TALENT's liability is limited to the return of the purchase price.

The purchase of this program implies acceptance of these conditions.

COPYRIGHT © TALENT COMPUTER SYSTEMS 1987.

AMSTRAD is a registered trademark of AMSTRAD Consumer Electronics plc. AMSTRAD PC is a trademark of AMSTRAD Consumer Electronics plc. ATARI ST is a registered trademark of Atari Corp. IBM and IBM PC are registered trademarks of International Business Machines Corp. GEM, GEM Desktop and the GEM logo are trademarks of Digital Research, Inc.

PREFACE

TRIMbase, your Data Management system, belongs to a new generation of software. It is fully GEM-based, and many hundreds of users have found it reliable, effective, and simple to use.

To control TRIMbase you use the standard tools of the modern computer interface: drop-down menus, icons and the mouse. There is no need to learn a formal programming language. At the same time, TRIMbase has the power needed to handle large amounts of complex data. Users have found that as their applications grow in size and complexity, TRIMbase continues to supply the efficient, unobtrusive support they need.

There are several ways you can learn about TRIMbase.

- ★ If this is your first data base system, read the USER GUIDE.
- ★ If you have previous data base experience, read the REFERENCE SECTION at the end of the manual. This gives a complete condensed description of the system.
- ★ To get immediate instructions and advice, use the 'help' options in the program itself. The system displays useful panels of information which tell you what step to take next. However, the User Guide will give you a more general view of the system, and we suggest you keep it handy.

"As a last resort, if all else fails, try reading the instructions!"

THE USER GUIDE

The USER GUIDE is divided into several parts.

Part 1 gives you a gentle introduction to the system, using information in a file which has already been prepared.

Part 2 tells you how to set up new files.

Part 3 is about file maintenance and information retrieval: entering new data, updating and deleting records, searching and sorting.

Part 4 shows you how to alter the descriptions of files which have already been defined and loaded with data.

Part 5 is concerned with generating reports. It tells you how to make the system print out information in a form directly suitable for sending out to other people. TRIMbase can produce reports both as tables and as 'free text', as in mail merge systems.

Part 6 is about the relational features of TRIMbase.

Part 7 covers some miscellaneous features, such as the exchange of data between TRIMbase and other software systems. It also includes some general advice about file security and the handling of large files.

To begin, we suggest you read Parts 1, 2 and 3.

SECURITY

Security of files is discussed in section 7.3.1. However, this topic is so important that we have reprinted this section. Please read it **NOW**.

Data files are precious, and one should take every possible precaution against losing them. Modern computers are reliable, but errors and breakdowns still happen because of external causes such as power cuts, voltage surges due to lightning, excessive dust or smoke in the environment, and so on. **If you use a data base regularly, it is not a question of whether this will happen: merely of when.** If you are well-prepared, you can survive a computer breakdown or the loss of a file with hardly any difficulty; but if you have not taken the right precautions the effects can be catastrophic.

The essence of file security lies in making frequent backups, or copies of your data base at particular dates. If the computer fails, you will only have to go back a short way to find a valid data base.

In some cases, a computer can damage a file it is reading as well as the one being written. It is therefore sensible to keep at least two separate backup disks, so that there is always one not mounted on the machine.

Similarly, if you have the slightest suspicion that a disk drive is not working correctly and damaging disks, don't load your precious backup into it until you have copied it on a different machine.

REGISTRATION

- ★ We strongly suggest that you use the card at the back of the manual to register with TALENT. This should be done within 10 days of buying TRIMbase.
- ★ If you are a registered user, we will send you regular newsletters with answers to queries and advice on using TRIMbase in various situations. You may also phone or write to TALENT to get advice from our Technical Department. Please be prepared to quote your user number, which is stamped on both halves of your registration card.

Using GEM

Before you start the next section of the book, make sure that you understand how to use the GEM system. In particular, you should know how to:

- ★ Switch on and reset the computer
- ★ Load and start the GEM system

- ★ Load and run programs
- ★ Find character keys on the keyboard
- ★ Use the mouse to select and drag items about the screen
- ★ Format a blank disk
- ★ Make a backup copy of a disk

All these matters are fully explained in the User Manual which comes with your machine.

Finally, a word of caution. If you plan to use TRIMbase to store information about living people, then in Great Britain you may need to register under the Data Protection Act (1984). You can get the necessary details from any post office.

Most other countries have similar laws.

PART 1: INTRODUCTION TO TRIMbase

TRIMbase, the Talent Relational Information Manager, is a system of supreme flexibility and ease of use. It runs in the vivid GEM environment — mouse, windows and icons — and lets you manipulate information in any way you want. You don't need to know about programming to get results — just point the mouse at an option and click. It's as easy as that.

Your information is kept on **files** or sets of record cards, on which you can write such things as names, addresses, personal or business details, class and exam results, birthdays or any other information you may want to record. You can examine the records, change them, add new records or discard the ones you don't need any more. If you have a large collection of data, TRIMbase takes the hard work out of searching: for example, it will take only seconds to find all the records dealing with — say — unmarried female bus drivers or people who live in Kinlochbervie.

TRIMbase is a **relational** data base system. This means that you can extract items of information from more than one file and combine them in any way you choose. For example, one of your files might contain the names and addresses of all your customers; other files might list business transactions for each calendar month. TRIMbase allows you to join up your files to produce a monthly — or yearly — total of each customer's business with you. Relational data bases are considerably more advanced than standard 'flat file' systems and give you the power and flexibility needed to handle any number of complex operations. On the other hand, if your application is straightforward, you can simply forget about the relational aspects altogether.

One of TRIMbase's most attractive and useful features is its Report Generator — a built-in word processor. You can produce any kind of document, form, letter, invoice or label using your data. You are not limited to certain preset forms and you don't need to load your data into another word processing program to get the results you need.

The whole package has been designed to be as straightforward and simple to use as possible.

1.1 Making a backup

Before doing anything else, you should make a safety copy of the TRIMbase system to work with. The way to do this is fully described in your computer owner's manual. Put the original disk away and keep it safely. You will need it for certain key operations such as defining a new file.

1.2 Loading the system

The exact procedure for loading TRIMbase depends on the machine you are using.

If you have an ATARI ST, start up your machine, load the program disk (or a copy) and click the mouse on MAINTAIN.PRG.

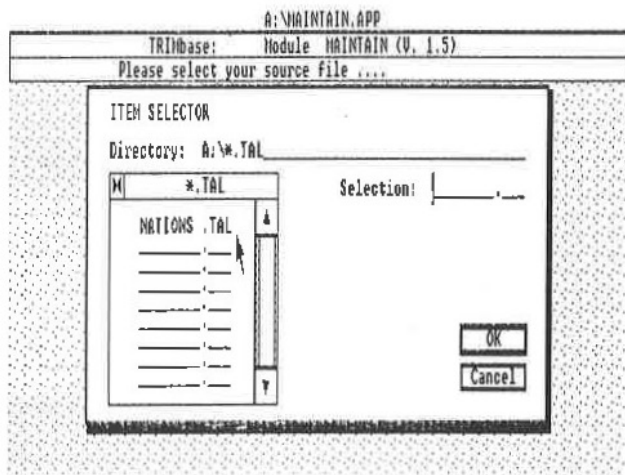
If you are using a PC (or a PC clone) you must begin by loading GEM. Then load the program disk and click on MAINTAIN.APP.

For large data files, we recommend that you make as much RAM available as you can. Various ways of doing this are described in Appendix 3.

1.3 Loading a Data File

When the system is loaded, the first item to appear is a title screen.

Next, TRIMbase shows you a 'item selector' panel. Use the mouse to select the file called NATIONS.TAL, and click on 'OK'.



The system now reaches the **Main Control Point**. The **menu bar** displays the titles of a number of menus, as follows:



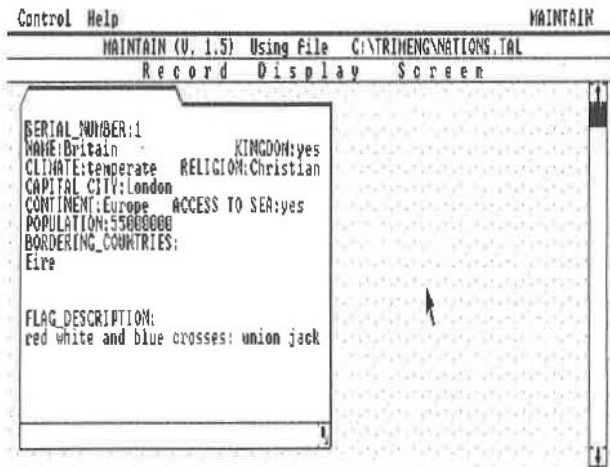
When you point the mouse at one of these titles, the corresponding menu drops down and displays a number of choices. Select the one you need by pointing with the mouse and clicking. If you don't want any of the items on offer, just click the mouse on a part of the screen not covered by the menu. If you want to find out more about the items, click on **Help** and you'll get some advice.

At almost any time, you can get back to the main menu by hitting the **Escape** key. This is a useful way of regaining control if you get stuck.

1.4 Displaying Records

Begin by choosing the **Display** menu and clicking on **All records**. The menu bar changes, and you'll see a record card about Great Britain. It includes the following details:

Serial number, Name, Kingdom, Climate, Capital city, Religion, Access to sea, Continent, Population, Bordering Countries, Flag Descriptions.



This record is the first of a file of records which includes many of the countries in the world. You can move to the next record by pressing the down-arrow key, or clicking on the down-arrow in the right-hand margin, or choosing **Next record** in the **Control** menu. Once you are past the first record, you can move backwards by hitting the up-arrow or clicking on the up-arrow in the margin, or by choosing **Previous record** in the **Control** menu. You can also jump anywhere in the file by clicking on the slider bar between the arrows. The black blob shows how far you are down the file.

If you run off the end of the file, TRIMbase simply returns you to the main control point. To continue examining the file, reselect **All records**.

Spend a few minutes browsing through the NATIONS file. You'll see that there are four different kinds of entry.

- ★ Some entries, such as the name of the country, are short sequences of characters less than one line long.
- ★ Some entries, such as the flag description or the list of neighbouring countries, are often more than one line long.
- ★ Some entries, like the population, are numbers.
- ★ Some entries, like the religion, have only a small number of possible values such as Christian, Buddhist, Muslim and so on. For this example we have classified Communism as a religion.

A fifth kind of entry, not used in the file of nations, is the date — such as a birthday or historical event.

In TRIMbase, each entry in the card is called a **variable**. Any one variable has a value for each card. The variable also has a type, which is:

TEXT for a short sequence of characters

BLOCK for a sequence of characters which can run to several lines

NUMBER for a numerical value

CLASS for a value chosen from a short list of possibles

DATE for a date

Before reading further, examine the record card and decide on the type of each variable.

You should have:

TEXT: Name, Capital City

BLOCK: Bordering Countries, Flag description

NUMBER: Serial number, Population

CLASS: Climate (tropical, desert, temperate, arctic)
Continent (Europe, Asia, Africa, America, Other)
Kingdom (yes or no)
Religion (Christian, Muslim, Hindu, Buddhist, Jewish,
Communist, Animist, Other)
Access to sea (yes or no)

DATE: (None)

You'll notice that sometimes a record doesn't have a value for a variable. Instead, it has an exclamation mark, and a footnote at the bottom of the card. For example, Andorra, the tiny Pyrenean state, is ruled jointly by the President of France and the Spanish Bishop of Seo de Urgel. Does this make it a kingdom? We don't know; so we put a 'non-standard' value which is the exclamation mark and a footnote. This feature is one of the ways in which TRIMbase is more flexible than many other systems.

1.5 Record Selection

In many cases, you will want to find particular records in the file, without searching the whole file card by card. Suppose you are interested in Nepal, and need to find out the name of its capital city. You need to set up an 'expression' which is true for Nepal and false for all other countries.

Here's how to do it.

First, return to the main control point (if necessary) by clicking on **End display** in the **Control** menu, or hitting the **Esc** key.

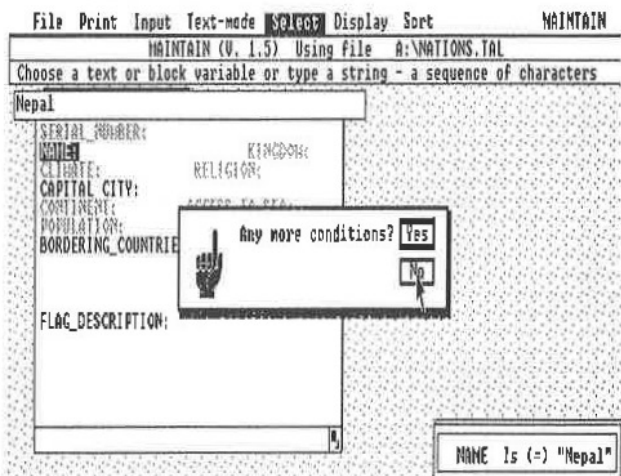
Next, find the **Select** menu and click on **Select records**. TRIMbase shows

you a copy of the record card, with variable names but no values. Point at NAME and click.

Now TRIMbase displays a list of possible relationships between words or bits of text. Choose the first one, 'Is'. You can see your expression building up in the lower right-hand corner:

NAME Is (=)

Next, TRIMbase again puts up the record card, but this time there is a blank box above it. Type the characters N e p a l and then hit the RETURN key. The expression is now complete, so answer 'No' to the question about more conditions.



At this point, TRIMbase picks out any records for which the expression is true. There is, of course, only one.

You can set up any expressions you like. For example,

POPULATION More than (>) 100000000

will pick out all the countries with populations of more than one hundred million people. Similarly,

RELIGION Is (=) Muslim

will select all the Islamic countries.

One useful relation with text and block variables is 'includes'. This is true if the second sequence occurs **anywhere** inside the first. For example,

FLAG_DESCRIPTION Includes ([]) red

will select all the countries with any red in their flags.

Try selecting a few groups of countries, such as:

All countries with fewer than 50,000 inhabitants

All countries with arctic climates

All countries without red in their flags (use "does not include")

All countries where the name of the capital city includes the name of the country itself, such as Singapore or Mexico. (Hint: the condition will contain two variable names on either side of the relation "includes".)

Sometimes it is useful to combine two or more conditions into a single expression, or rule for selecting records. This facility is essential for information retrieval, but it is also an area where careful thought is vital. The problem is that natural languages (such as English) are ambiguous. For example, suppose you are asked to:

"Make a list of all the countries with yellow and blue flags."

This could mean two different things:

- (a) Include any country which has both yellow and blue in its flag.
- (b) Include any country which has yellow or blue (or both) in its flag.

Of course list (b) will generally be longer than list (a).

In a TRIMbase expression you can have as many conditions as you like, but if there is more than one you must indicate the way in which the conditions are combined.

The first method implies that **all** the conditions must be true for a record to be selected. This is represented by the sign & ("and") in the conditions box.

The other method selects a record if at least **one** of the conditions is true. The sign for this method is | ("or").

Let's suppose that you have clarified the request, and decided that you are really looking for a list of countries, each of which has both yellow and blue in its flag. The two conditions are

FLAG_DESCRIPTION Includes ([]) "yellow".

and

FLAG_DESCRIPTION Includes ([]) "blue".

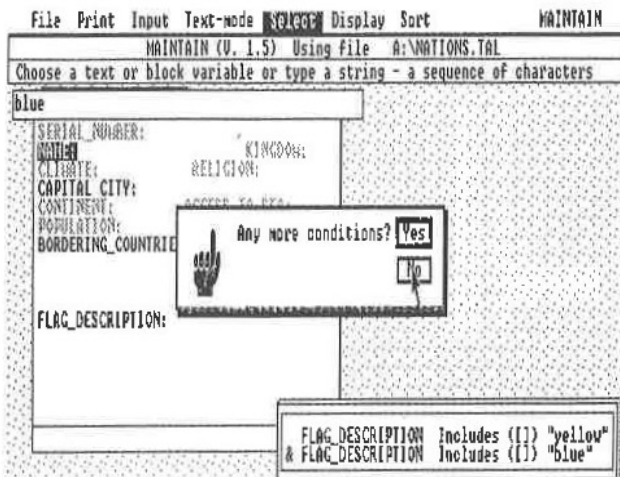
Both the conditions must be true for a record to be chosen.

First select the condition

FLAG_DESCRIPTION Includes ([]) "yellow".

TRIMbase then asks you if there are any more conditions. When you click on 'yes', you are asked whether you want all the conditions to

apply or whether any one condition is sufficient. Select the first option and continue with your selection. You will see the condition building up in the expression box.



The "all of" rule is useful in setting up ranges of numerical values. For example, this expression will pick out countries with populations between 40 and 50 millions:

```
POPULATION More than (>) 40000000
& POPULATION Less than (<) 50000000
```

The "any of" rule is useful if you want to select all the records which belong to any of two or more groups. For example, suppose you want a list of all countries in Europe or America. The right expression is:

```
CONTINENT Is (-) Europe
|
CONTINENT Is (-) America
```

If you had chosen the "all of" rule, you would have netted nothing; there is no country which is both in Europe and America!

1.6 Sorting

Next, we'll look at sorting. It's often convenient to have your records in some particular order, such as increasing order of population or alphabetical order of name. To try this facility, select the **Sort** menu and follow the instructions. When the sort is complete, you are asked if you want to remove exact duplicates. Answer **no** at this stage. When you display the file, you'll find it in the order you want.

When TRIMbase is sorting, it will often come across pairs of records with the same 'key field' — such as two people called Smith. When it finds such a pair, TRIMbase will not change their original order.

This fact can be made the basis of a two-level sort. Suppose you have a file of people, where each record has a Christian name and a surname as separate fields. To arrange the file in alphabetical order of names:

- a) First sort on Christian names
- b) Then sort on surnames.

This sequence is the opposite of most people's intuitive feeling — but it is correct.

To achieve a multi-level sort, you sort several times, starting with the 'innermost' field.

1.7 Statistics

TRIMbase can display simple statistics about any variable. The statistics can either refer to the whole file, or to any section of it which is selected at the moment. Choose the **Display** menu, click on **Statistics** and select the **For all records** option. When you select a variable, a panel with useful information is displayed. Try it and see.

The screenshot shows the TRIMbase software interface. At the top, a menu bar includes 'File', 'Print', 'Input', 'Text-mode', 'Select', 'Display', 'Sort', and 'MAINTAIN'. Below the menu bar, a status line reads 'MAINTAIN (V. 1.5) Using file C:\TRIMENG\NATIONS.TAL'. A prompt says 'Hit any key or click the mouse to go on'. The main window displays a list of variables: SERIAL_NUMBER:, NAME:, CLIMATE:, CAPITAL CITY:, CONTINENT:, POPULATION:, BORDERING_COUNTRIES:, KINGDOM:, ACCESS TO SEA:, and FLAG_DESCRIPTION:. A mouse cursor is pointing at the 'KINGDOM:' variable. A statistics panel titled 'Global statistics for RELIGION' is open, showing the following data:

	Unknown	Special	Other
Unknown	0	2	141
Christian	:	69	
Muslim	:	28	
Hindu	:	1	
Buddhist	:	10	
Jewish	:	1	
Communist	:	14	
Animist	:	17	
Other	:	1	

Here is an example of how to use 'selected' statistics. First, make up and enter an expression which selects all the countries which are **kingdoms**. Then display statistics for the 'selected' records. The figures about the variable RELIGION will now tell you how many **kingdoms** there are of each religious denomination. Similarly, statistics on POPULATION show the smallest, largest and average population of the **kingdoms** of the world.

To get statistics about the other countries, you'll have to select:

KINGDOM Is (=) no

1.8 Printing Records and Statistics

Finally, if you have a printer, you can print out both records and statistics. The system will want to know the height of the page (in lines) so that it can start new pages in the right place. If you are using A4 stationery, remember that is 71.5 lines long!

If your printer is not of type EPSON MX80 Type III (or a compatible model), try running the configuration program, ENGLISH.APP (ENGLISH.PRG on ATARI ST). This is described in Appendix 1.

If you do not have a printer, don't try to print.

1.9 Text-mode

When TRIMbase compares two segments of text (in selecting or sorting), it normally takes upper and lower case letters to be equivalent. You can make the cases different by choosing the Text-mode menu and selecting the appropriate line.

The rule you choose will stay in force until you change it back (or until you reload the system).

1.10 Changing the Colour of the Display

You may change the appearance or colour of the screen display at any time while TRIMbase is running. Hit the appropriate function keys until you arrive at a combination which you like. Some of the possible combinations may be quite dramatic! The colour and appearance you select are retained when you transfer to a different module.

- F1 — changes the background colour
- F2 — changes the background pattern
- F3 — changes the record card colour
- F4 — changes the record card texture
- F5 — changes the colour of the text

If you hold down the Shift key when pressing any of these keys, the previous change made with that key is reversed.

If you have a mono display, the colour changes will be restricted to a range of grey tones or even to black and white.

1.11 Useful Keys

You will need to know about the following keys:

ATARI

PC

Esc	Esc	— pressing this key will generally get you back to the main control point
Backspace	← Del	— these keys delete characters to the left of the cursor
RETURN	↵	— this is used to terminate items
Tab	↔	— the Tab key is used to shorten data input (see 3.1.7)
Delete	Del →	— these keys delete characters to the right of the cursor
n/a	Alt/Ctrl/Del	— pressing all these keys at the same time (a deliberately awkward operation) resets the machine

To make full use of the **Del** → key on the AMSTRAD PC, you must redefine it using the "NVR" program described in Appendix 11.2 of the AMSTRAD PC 1512 user instruction book. The correct setting for TRIMbase is "537F". If you can't be bothered to do this, TRIMbase will still work: you just won't be able to delete characters going forward.

Again referring to the AMSTRAD PC 1512, it is worth knowing that if you hit 'control' by itself you will change the function of the arrow keys, making them move the menu pointer rather than the cursor. If you hit 'control' again, you will return to the normal function of the arrow keys.

1.12 Quitting

When you finish, choose the **File** menu and click on **Quit**. The colours and patterns you have chosen for the screen will be recorded, and will come up automatically next time you run the system.

PART 2: SETTING UP NEW FILES

At this point, before reading further, you may need to run the system configuration program, ENGLISH. This program, which only needs to be used once (ever), serves to describe your printer and set up conventions for TRIMbase. You won't need to run the program at all if:

- a) Your printer is EPSON-compatible
and
- b) you are content to use dates in the 'short European' format
— i.e. "31/12/86".

The system configuration program is fully described in Appendix 1.

When you set up a new file, the first stage is to design a record card.

To make the section easier to read, let's set up a running example. Suppose you're organising a literary competition. The entrants send you copies of their work together with a form like this:

Name
Personal Title (Dr/Mr/Mrs/Ms/Miss)
Address
Age
If at school, name of school
Title of entry
Length of words
Type (Ring one)
Subject (Ring one)

When the closing date for the competition is reached, you plan to distribute the entries amongst a panel of judges, each of whom is a specialist in one (or more) of the types of entry. The judges will 'mark' each submission on a scale of 0 to 1000. Then you will arrange the entries in order of merit, and get the judges together to decide on the prizewinners.

If there is more than a handful of entries, TRIMbase will be a considerable help. It will do most of the boring clerical jobs such as:

Printing address labels for the competitors

Matching up entries with judges

Sending reminders to judges who are slow to make their returns

Sorting the entries into order of merit

To begin, you'll set up a 'competitor' file with all the information on the form, and some additional facts for each entrant:

Date the entry was received

Name of the judge assigned

The mark awarded

You've now done enough planning to design a record card and set up a TRIMbase file.

2.1 Defining a new TRIMbase file

First, format a blank disk and have it ready. Then insert the program disk and load program DEFINE. This is the module you use to design new record cards. It also has several other uses which are covered later; so if you make a mistake and find yourself in unknown ground just press **Esc** to get back into control.

A title screen is then displayed, followed by a screen with the main menu bar, which looks like this:

File Define card Relational Text-mode Help DEFINE

(On some versions of GEM, the DEFINE menu on the right may be absent, but instead there will be a "Desk" menu on the left. Both menus serve the same purpose).

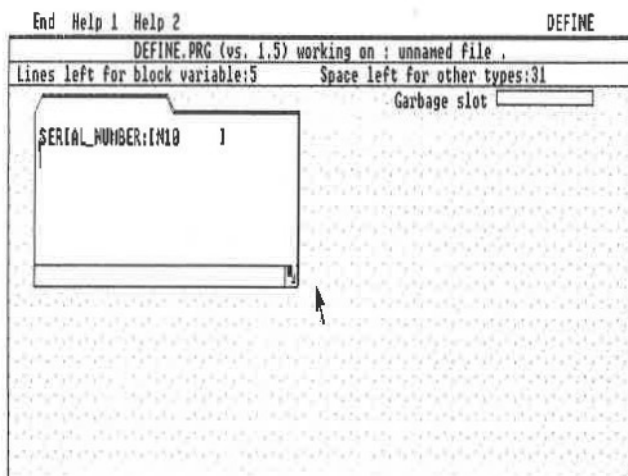
At present, don't use the **Relational** menu. Its purpose is described later.

The **DEFINE** menu lets you call up various utilities such as the calculator or clock.

The **Help** menu contains a number of entries, each of which gives you a 'help' screen on some particular subject. You can bring these panels up whenever you like.

2.2 Defining a new Record Card

To define a new record card, point at the **Define card** menu and select the option **Make new card**. Next the system displays a new menu bar, a small record card and a 'garbage slot' at the top right of the screen. The record card is blank except for an entry called SERIAL_NUMBER which is compulsory for all records.



To design your record card you need two tools: the keyboard and the mouse.

Begin by stretching the blank record card to roughly the size you want. Point the mouse at the lower right-hand corner, press the left button and drag the corner away.

2.2.1 Defining variable fields

Next, start entering the variable fields. There is one variable for each item of information, such as an address, age or title.

In general, a variable has three attributes:

- * a name
- * a type
- * a layout

The name of a variable must start with a letter, which can be followed by any sequence of letters, digits or the symbols (underline) , . - or / or a space.

Examples are

PERSONAL_TITLE POST CODE SCORE2

All the variables in a file must have different names. They can be of any reasonable length.

The type of a variable is always one of:

- NUMBER (a numerical value)
- DATE (a date consisting of day, month and year)
- TEXT (a short sequence of characters)
- BLOCK (like a text variable, but extending over several lines. This could be an address or a continuous piece of writing such as a recipe or an abstract of a learned paper)
- CLASS (a variable with a limited number of possible values which you can list in advance. Examples of class value sets are male/female and rain/hail/snow/sleet/dry)

The layout of a variable determines how much space it takes on the card. For Date and Class variables, this decision is made automatically. Number and Text variables can take up to one line each, and you must decide how many characters you are likely to need. Block variables can take several lines (up to 15) and you must say how many lines are actually needed.

TRIMbase is so flexible that there is no need to decide all these matters before you start using the system. You can design your card as you go along, making as many corrections and changes as you wish.

When you start entering a field, the cursor will show where the field is to go. You can move the cursor about by using the cursor keys

↑ ↓ ← and → or by clicking the mouse at the place at which you would like the cursor.

Begin by typing the name of the field. Typing mistakes can be rubbed out with the Delete keys.

End the name with RETURN. A small menu appears for you to select the type of the variable. You can choose an entry:

- a) By stepping down with the space bar and hitting RETURN
- b) By typing the initial letter of the type and hitting RETURN (for example 'B' will move you to Block). If you use the keyboard to select an item, keep the mouse out of the way.
- c) By pointing with the mouse and clicking.

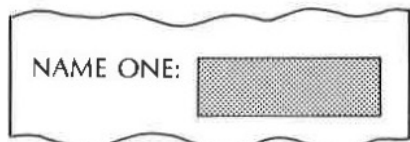
The next action depends on the type:

Date variables: no further action is needed.

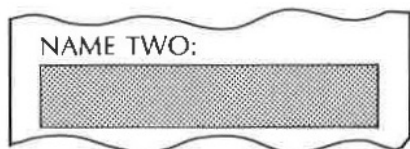
Number and Text variables: you must give the maximum length

in characters. Type the size, followed by RETURN. Note that the system expects to be told the number of characters, not the largest expected value. For example, if you are sure that a particular number variable will always be less than one million, type '6' — not '999999'. Remember to allow space for a - (minus) sign if your values can be negative.

Block variable: you must state the number of lines, and give some further information about layout. Every block variable extends to the right of the record, but at the left two arrangements are allowed:



or



Choose the one you prefer.

Class variables: you must give the list of possible values. You can have up to 100 values which should all be different. They can be built up of any characters except '/' and 'space', and should be separated by slashes, typed as /. End the list with RETURN. You'll notice that the list always starts with special values ? and !

When you've entered a variable, the field appears like this:

SURNAME: [T25]

This indicates that SURNAME is a Text variable with a maximum length of 25 characters. Another example is:

ADDRESS: [B3*37]
[]
[]

Here ADDRESS is a block with 3 lines of 37 characters each.

Very short fields use a different notation:

N	(1 character)
NN	(2 characters)
[N]	(3 characters)
[N4]	(4 characters)

At this point, we suggest you key in a few fields using the running example.

2.2.2 Editing the Record Card

As you enter the various fields, you can edit the card with the mouse in several different ways:

- * Change the size of the record card by dragging its lower right-hand corner. The sizes of block variables will change correspondingly, but you are prevented from moving the edge of the card over variables of other types.
- * Change the size of an individual variable by dragging its closing bracket — up or down for a block and left or right for the other types. Note that the sizes of Date and Class variables are fixed and you cannot change them this way.
- * Change the position of a variable on the card by dragging the name. Of course you can't drag a variable outside the card or obscure another variable. You are welcome to have several short fields on the same line if they all fit.
- * Delete a variable by dragging it to the garbage slot. This is useful if you discover a spelling error or a wrong type when the field is already entered.
- * Find out more about a field by double-clicking its name.

When you've keyed in all the variable fields, adjust their positions and the overall card size in any way you see fit.

If you've worked through the example, the screen should show something like this:

```
End Help 1 Help 2 DEFINE
DEFINE.PRG (vs. 1.5) working on : unnamed file .
Lines left for block variable:1 Space left for other types:
Garbage slot:
SERIAL NUMBER:CN10 ]
NAME:LT30 ]
PERSONAL TITLE:LC5 ]
ADDRESS:CB3 x35 ]
[ ]
AGE:CN1 ]
NAME-OF-SCHOOL:LT20 ]
TITLE-OF-ENTRY:LT20 ]
LENGTH-IN-WORDS:CN6 ]
TYPE:CC11 ] SUBJECT:LC0 ]
```

To complete the process of definition, select the **Disposal** menu and click on **Accept**. Alternatively, hit the **Esc** key.

The system shows you the GEM item selector panel and asks for a file name. At this point you should always choose a new name unless you want to destroy an existing file. Enter the name, which should consist of up to eight letters, followed by a dot and the three letters TAL. You can leave out the .TAL suffix if you want — the system will automatically add this to the file name. Examples are:

```
ENTRANTS.TAL
JUDGES.TAL
```

When you've typed the name, make sure you've loaded your formatted disk and then hit RETURN. Your record card definition will be transferred to the disk as a file.

The process of definition is now complete. Choose the **File** menu and click on **Quit**. Alternatively, you may jump straight to one of the other TRIMbase modules, such as MAINTAIN, described in PART 3. The name of the most recent file to have been created is passed on automatically. Don't forget to label the disk with the name of your file.

PART 3: INPUT AND MAINTENANCE OF DATA

Two key activities in using a data base are:

- a) The input of new data
- b) The maintenance — that is, the correction and updating — of existing data

Both activities are supported by the second module of TRIMbase, MAINTAIN. The module also offers several other functions which operate on a single file such as selection or display of records, sorting, display of simple statistics and listing on a line printer.

There are two ways of loading new information into TRIMbase.

- a) You can type it on the keyboard
- b) You can 'import' it from a disk file made by a different application program

In this section, we shall be discussing only the first of these two alternatives. The 'import' facility is described in PART 7.

Before you can input any data at all, you must first define a suitable record card using module DEFINE. We shall assume that you have set up a card on the lines of the running example and have got the definition available on a disk.

Load program MAINTAIN, as described in PART 1. When the item selector panel comes up, load the disk with your file definition and click on its name. If you've just come from the DEFINE module and don't want to start using a different file, you need do nothing — your file is loaded automatically.

3.1 Input of New Data

When the main menu bar appears, point at **Input** and select **New data**. The system displays a blank record card, and you can start filling in the values of the variables for the first record in the file.

The rules which govern data input are carefully designed to make this process as fast and as error-free as possible. It is well worth reading through them carefully.

The main control keys are those you learned to use in PART 1: RETURN is used to end items (or lines in Block variables).

The two **Delete** keys are used to get rid of characters which may be wrong.

Esc is used to return directly to the control point. However, it only works when you have completed a record card; you are not allowed to abandon a partially filled record unless you strike the **Tab** key.

The cursor control keys → and ← serve to move the cursor about within a field. They only work in text and block fields, since the other types are generally so short that correction can be made by deletion. The cursor control keys ↓ and ↑ generally serve to end fields. ↓ works just like RETURN, but ↑ takes you back to the **previous** field.

In a Block variable, ↓ and ↑ are used to move between the lines of data. If the cursor is already at the first line, ↑ will take it back to the previous field.

If you forget what a variable represents, point to it with the mouse and double-click. You are presented with a full description of the variable. If it is a class variable, you are given a list of its possible values.

If the cursor is not at the end of the card, **Ins** will make it move to the next field. When the cursor is at the end **Ins** completes the input or editing of the card and moves on to the next record.

The **Tab** key takes you directly to the end of a card, skipping any unfilled areas.

3.1.1 Serial Numbers

The first field in every card is the serial number. This number is restricted to the range 1 to 999999 and must be different for each record in the file. If you want to supply your own record number, you can do so, ending it by RETURN. Any non-decimal characters you hit are simply ignored. If the number you give it already in use, TRIMbase displays a warning and lets you try again.

Alternatively, you can simply strike RETURN and TRIMbase automatically supplies a unique record number for you. This is usually the simpler way of setting up the record number.

The other fields in the card will be of the five basic types: Number, Date, Text, Block and Class. The system expects you to give their values in the order in which they appear on the record card.

3.1.2 Number Variables

A number consists of a sequence of decimal digits. It may contain one decimal point and it may be preceded by a sign.

Examples of number values are:

89 -78.56 0 0.0009

TRIMbase won't respond to any character which doesn't conform to these rules, or which would make the number longer than the size of the field you have allowed. The number should be ended with RETURN.

If the value of a number is unknown, just type RETURN and the field is left blank.

Sometimes numbers have 'special' values which don't fit into the framework you have set up. For example, some people may refuse to give their ages, but simply say 'Over 21'. To insert such a value, type "!" (the exclamation mark) followed by RETURN. The cursor moves to the footnote area at the bottom of the card, and lets you insert a comment, which should also be ended by RETURN.

3.1.3 Date Variables

A date consists of exactly six (or eight) decimal digits. The format depends on the choice you made when you ran the configuration program. For example, if you chose to put days before months and to use four-figure years, you would type a date as:

25/12/1986

Notice that the /s appear automatically, so you actually type the sequence "25121986". However, you still need RETURN to end the field. The entry of dates is simplified because the system ignores any character which is not part of a date. If the date you type is invalid (such as February 29th 1900) TRIMbase displays a warning message and lets you enter the date again.

A date can take a 'special' value, signalled by the "!" sign. It can also take the value 'unknown', shown by a blank field.

There is a quick way of entering today's date: just hit the "#" key. This is useful when you make daily entries to your file.

3.1.4 Text Variables

A Text field consists of any sequence of characters, ended by RETURN. Since anything is allowed, the system makes no checks on what you type. An 'unknown' value is shown by a blank field, and no provision is needed for 'special' values.

If you discover a mistake when you're entering a Text field, you don't need to backspace all the way to the error and retype everything after it; you can use the left and right arrow keys to reposition the cursor. If the cursor is in the middle of a text field, a Delete key will remove one character and 'drag' the rest of the field one place to the left. Any other character is simply inserted, pushing the rest of the field towards the right. This is all much easier to do than to describe!

3.1.5 Block Variables

A Block field consists of a sequence of characters, which may, but need not, include RETURNS. In general, there are two types of Block field.

- a) Those where the contents are naturally divided into lines, such as addresses or lists of names
- b) Those where the contents are a continuous piece of text such as the description of a work of art or a recipe.

To enter a field with its own line divisions, type each line and end it with RETURN. When you have typed enough RETURNS, the system automatically moves on to the next field.

To enter a field of continuous text, just type it in without any RETURNS until you come to the end. TRIMbase automatically arranges the material so that words are not split across lines. This is known as 'word-wrap'.

You can end a Block field at any point by hitting the **Ins** key.

A Block field can be corrected by moving the cursor about, in the same way as a Text field (except that you can use the ↑ and ↓ keys to move between lines).

3.1.6 Class Variables

The entry of a Class field is very straightforward. You just need to type enough characters to specify the value unambiguously. Use **Text-mode** to determine whether upper and lower case letters are taken as being the same. If you choose the value names carefully, one letter per value should be enough. Like other fields, the Class value is terminated with RETURN.

Class variables can be 'unknown'. They can also have special values, like number and date variables.

3.1.7 Completing a card

At this point, try working through your record card and putting in some values. When you enter the last value, a message appears saying 'Card Complete'. You now have a choice:

If you are satisfied that a card is correct, press the **Ins** key, and the new card will be entered into the file.

If you wish to make a correction, press the **Del** key (or **Backspace**) and the cursor moves to the end of the first field. Use **Ins** to step through the fields until you reach the one to be corrected, then fix the error by deleting and entering the correct value. **Tab** then takes you to the end of the record.

As soon as you press **Ins**, the card is cleared for you to type the next record. You can keep on entering records until all your data is loaded.

If you hold down **SHIFT** when you press **Ins**, the values in the card you have just completed are not deleted but are transferred on to the next card in the sequence. You can alter the card as appropriate. This is particularly useful if you are entering several records which have common fields, such as members of a family who all live at the same address.

When you have entered your last record, hit **Esc** or select the **End input** option in the **Disposal** menu. Don't supply a new serial number or you will be forced to complete another card.

3.2 Saving Files

As soon as you have keyed in a batch of new records, you **must** save or 'write' your file back to the disk. Don't wait until you have finished entering in all your data — it is sensible to stop at regular intervals and save what you have done so far. This action protects you from losing a large amount of work through power failures or other accidents such as children tripping over the power cable.

Saving or 'writing' files is an area where you should exercise deliberate care, since a blunder on your part could well lose you the work you have just completed. A vital rule is

DO NOT REMOVE THE DISK WITH THE ORIGINAL FILE

Return to the Main Control point, select the **File** menu and choose either of the two **Save whole file** options. They have quite different implications.

Save whole file as means that you create a new file with a different name from the old one. You specify the name on the GEM item selector panel. The contents of the old file are not altered from their state at the beginning of the session, so this choice is inherently safe — if you have made mistakes in changing or deleting items it is easy to go back. On the other hand, there must be enough room on the disk to hold the entire new file. This means that if you have only one disk drive, the largest file you can safely handle this way is one half of the disk's capacity.

Save whole file means that you choose to 'write back' or overwrite your old file. If you select this option, you don't need to specify anything as the name of your file is not changed. However, if you decide to follow this course, you take somewhat more of a risk. First, it will no longer be so easy to go back to the original version. Second, if there should happen to be a power cut or machine failure while you are writing back, the entire file, with all its data, will be lost. On the other hand, the scheme does have compensating advantages:

- a) Writing back is faster because only the alterations are recorded.

- b) You can handle bigger files since there is no need to set aside space for a new file.

There is one more aspect of 'writing back' which should be mentioned. A file is like a bound notebook. New records are added at the end. Deleted records are crossed out but not actually removed: they remain as 'junk' cluttering up the structure of the file. It follows that if you keep a file with frequent additions, deletions and amendments, it will gradually get longer even though the total number of records stays the same. To cure the problem, occasionally choose a new name to write the file to: this will dispose of the accumulated trash.

The management of disk space is never easy. The wise user will know roughly how large his files are, and how much free space there is on his disks before he starts a TRIMbase session. In practice, the 'writeback' method can be made perfectly safe by making a backup copy of your file, on to another disk, before starting any new work.

3.3 Adding Records to Existing Files

To add new records to an existing file, the process of loading a file and inserting data can be followed exactly. New serial numbers are supplied automatically.

3.4 File Amendment

Once a file is already in the system, it may need to be amended from time to time.

To change a file, first load it. If you know which group of records you want to change, set up a selector expression (see PART 1.5). Then choose one of the options **All records** or **Selected records**.

Shuffle through the records until you find one you want to change. Then point at the **Control** menu and choose one of the options:

Delete record will remove the record from the file. This is such a drastic action that you are asked to confirm that you really want to do it.

Amend record lets you change any (or all) of the fields in the record. Use **Ins** to move down the fields until you come to one you would like to change. Edit it, and then either:

Move on to the next field by hitting **Ins**.

or

End the amendment by striking **Tab** and then **Ins**.

In the latter case the system responds by showing you the next record.

Search and Replace helps to edit a record by going straight to the sequence of characters you want to change. When you choose the

Search and Replace option, the system asks you to type a sequence of characters or string into a GEM panel. Then it looks for the first occurrence of that string in any text or block field of the current record. If it finds an occurrence, it stops with the cursor pointing to the start of the string so that you can edit it as you wish. You complete the process just as if you were doing an ordinary 'Amend'.

If the string you type cannot be found in the record, the Search command has no effect at all. Remember that Search does not look in Number, Date or Class fields!

The Search command is only moderately useful if you are editing records one by one. It is just as easy to use Amend and move the cursor manually. However, the command is designed chiefly for use with global updates, which are described below.

Global Update is one of the simple, powerful features which make TRIMbase so useful. It gives you a straightforward way of making systematic changes all the way through a complete file.

To start a global update, locate and edit the first record you want to change. Use either Amend or Search and Replace — whichever is most convenient. In either case, the system takes the change as an example and remembers the exact sequence of keystrokes you use.

Then give the Global Update command. TRIMbase works its way through all the records (or all the selected records if a selection rule is in force) and applies exactly the same sequence of keystrokes to each one. It follows the example you set and changes every record in exactly the same way as you altered the first one.

As the Global Update runs, you can watch the alterations on the screen. If you have made a mistake and don't approve of what is happening, you can stop the update by hitting any key.

We shall give two examples:

First, suppose that in your country the voting age is suddenly lowered from 25 to 18. You have a file of people with fields called AGE (a number) and VOTER (a class with answers YES and NO). You want to pick out all the records of people between the ages of 18 and 25 and change the value of VOTER from NO to YES.

To do the global update, begin by selecting the records you want to change.

```
AGE More than or equal to (>=) 18  
& AGE Less than (<) 25
```

Then display the selected records and amend the first one, changing the value of VOTER from NO to YES. Now call Global Update and all the other records are altered automatically.

Second, imagine that you have a file with names and addresses, and that your typist is not a good speller — many instances of 'London' have

been spelled as 'Lungdung'. In this case, straightforward amendment won't work in a global update, because the mis-spelled word will be in different places in different addresses, and the number of cursor movements you need will change from one record to the next. Many records will not need to be changed at all.

This is where the Search and Replace command is useful. Find the first occurrence of 'Lungdung' by automatic search, and correct it to read 'London'. Then call for the Global Update, and every occurrence of 'Lungdung' (more correctly, the first occurrence in any selected record) is found and altered.

All the entries in the file amendment menu have keys which give the same effect. They are:

Previous record	↑
Next record	↓
Delete record	Delete (Del ←)
Amend record	A or a
Search and replace	S or s
Global update	G or g

3.5 Removing Duplicates

TRIMbase offers a useful facility for removing adjacent duplicates of records. First sort the file (this was explained in PART 1.6) and then answer 'Yes' to the question "Do you want to remove exact duplicates?". Only records which are identical in all respects except for their record numbers are taken out of the file. For technical reasons you are not offered this option if you have just sorted on a **class** variable.

3.6 Saving Selected Records

You can create a new file consisting of selected records if you wish. This is also possible from DEFINE (See PART 6.2). First select the records you want (See PART 1.5) using the appropriate selection procedures. Then go to **Save selected records** in the **File** menu and type in the new file name.

3.7 Statistics

As mentioned in PART 1, TRIMbase will display and print statistics about each of the variables on your card. These statistics are normally kept up to date as records are added and deleted. However, if you delete a record with an extreme value — say, the highest known age — the system cannot determine the new extreme value without searching through all the other records. When this happens, the extreme value is shown as a row of stars, meaning "I don't know just at the moment".

The special option **Revise statistics** will bring all the statistics up to date.

3.8 Moving on to other modules

When you have saved your current file, you can jump directly to either of the two other modules, DEFINE or REPORT. If you choose REPORT, the file you have been handling is loaded automatically for you to produce a report.

PART 4: CHANGING A FILE DEFINITION

According to the standard textbooks on the management of data bases, you are supposed to get your file definitions absolutely right before entering any data. In practice, however, data base managers are human and make mistakes. Quite often these errors do not become evident until a considerable amount of information has already been keyed into the system.

There are several types of blunder:

- a) You might forget to include some important variable.
- b) You might include a variable which is unknown in the large majority of cases, or is of no real value to the user.
- c) You might allow insufficient space for a number, text or block variable.
- d) You might forget some of the possible answers to a class variable.
- e) You might choose inappropriate names for some of the variables, or some of the class values.
- f) You might not like the appearance of your record card.

In TRIMbase, all these errors can easily be corrected at any stage. Load module **DEFINE**, run through the initial dialogue, and choose the **Define card** menu. Then pick **Change existing card**.

Next, use the item selector menu to indicate the file you want to change. The file is loaded, and its record card appears on the screen.

You can now continue much as if you were defining a new card. Thus:

- a) You can change the overall card size by dragging the lower right corner
- b) You can move an individual field about by dragging it with the mouse.
- c) You can change the size of an individual field (except a date or a class) by moving its closing bracket
- d) You can discard a variable by dragging it to the garbage slot
- e) You can add a new variable by defining it in the usual way
- f) You can redefine an existing variable, by double-clicking it with the mouse. TRIMbase replies by showing you an information panel and the question: **Change or No change?**

If you select **Change**, TRIMbase takes you through a normal definition sequence, with one key difference: the type of the new variable must be the same as the type of the one you are changing. This is essential to allow the values of the old variable to be transferred.

When you complete the new card definition, TRIMbase will ask for a new file name, which must be different from the original. The system then constructs the new file. Each record in the old file is transferred, and the values of variables are inherited according to the following rules:

- a) Rearrangement of variables on the card has no effect whatever on values unless the size of field is changed.
- b) Increasing the size of a field also has no effect on any existing value.
- c) Decreasing the size of a number field does not change any values, but it may affect the way the values are displayed. If the field is too short to contain an existing value, it will show a row of stars : ****. Such values can always be displayed again by redefining the field to be sufficiently long.
- d) Decreasing the size of a text or block field will cause any values which no longer fit to be truncated or cut off at the end. This may lose you some information, so be careful!
- e) If a variable is discarded, all its values are lost, except for any related footnotes, which remain on the new card.
- f) If a variable is newly defined, its value will be set as 'unknown' in all existing records. The appropriate values can be filled in by amendment.
- g) If a variable is redefined, its values will be inherited, following the general rules set out above. However, class values are transmitted following their positions in the answer list, not according to similarity of answer names (if any). If the new answer list is shorter than the old one, then 'overhanging' values will be recoded as "?" (unknown). If the new list is longer, there will be no instances of the values at the end of the new list. These ideas are illustrated in the diagrams below:

- 1) Old list: ? / ! / single / married / widowed / divorced
 New list: ? / ! / seule / mariée / veuve / divorcée
- 2) Old list: ? / ! / Conservative / Labour / Alliance / Other
 New list: ? / ! / Other / Alliance / Labour / Conservative
- 3) Old list: ? / ! / yes / no / sometimes
 New list: ? / ! / yes / no
- 4) Old list: ? / ! / piston / turbo / jet
 New list: ? / ! / piston / turbo / jet / rocket

The second of these examples would probably be a blunder (or

an attempt at sabotage), since a variable with the original value Alliance would be recoded Labour.

h) If you start defining a variable and then abandon the attempt, you will lose that variable and all its values in the field.

If you change your card layout as described in this section, you may not be able to use any previously defined reports — see PART 5.

PART 5: THE REPORT GENERATOR MODULE

Note: To take full advantage of the Report generator, you need a printer. However, you will still be able to display reports on the screen, even if you don't possess one.

5.1 Introduction

The Report Generator in TRIMbase gives the package a decisive advantage over other comparable data bases. You can construct any kind of report you like using a system like a word processor. The system is so flexible that it doesn't need a list of 'features' such as 'labels', 'tables', 'summaries', 'form letters' and so on. All these types of report, and infinitely more, can be designed by TRIMbase's Report Generator.

In principle, TRIMbase reports can be divided into three categories:

- a) Each report can be based on the data in a **single** record.
- b) Each report can be based on the data in a **batch** of records with common data values, such as people who share the same address.
- c) A report can be based on the data in **all** the records. There will be only one report, irrespective of the number of records in the file. Such a report is generally called a summary, and contains totals, averages and subtotals rather than values from individual records.


As examples, here are some typical TRIMbase records and the reports you can produce from them.

Example 1: Consider a file of college records like this:

```
SERIAL NUMBER: 327
SURNAME: Bloggs
CHRISTIAN NAME: Jayne
SEX: FEMALE
STATUS:SINGLE
ADDRESS: 43 Warwick Way,
         Westward Hol
         Devon
TUTOR: Dr. Crippen

ENGLISH: 56
MATHS: 19
COMPUTING: 87
```


Each record in this file could be used to produce an exam results letter such as:

Miss Jayne Bloggs	March 10th 1987
43 Warwick Way Westward Ho! Devon	
Dear Jayne,	
Here are the results of your examinations:	
English	PASS
Maths	FAIL
Computing	DISTINCTION
Your tutor, Dr. Crippen, will no doubt wish to discuss these results with you next term.	
Yours sincerely,	
	

With the appropriate instructions, TRIMbase will print a similar letter for each student in the file.

Example 2: This shows how the results can be batched, to list the results obtained by each tutor. A typical report would read.

TUTOR: Dr Crippen				
Name	Mark in English	Maths	Computing	Average
Jayne Bloggs	56	19	87	54
Agnes Cook	73	87	93	84
Clive Jones	51	23	32	35
Bill Green	21	12	0	11
Mary Black	86	100	95	94
Susan Smith	67	56	34	52
Mohd. Ali	59	72	61	64
Averages	59	52	57	56

Similar results will be produced for other tutors. In each case, the length depends on the number of students involved.

The third example shows a summary, which is a complete report over the whole file.

Class Summary		
Number of Students	=	54
Tutors: with Dr. Crippen	=	7
with Dr. Foster	=	29
with Dr. Strangelove	=	18
Average mark in English		52.6
Maths		61.1
Computing		73.9

To make a report, you start by setting up a **report form** — that is, a 'skeleton' report which will be filled up with live data later. For example, the report form for the letter about exams would be:

A	B	C	D ✓
	E ✓		
	F ✓		
	G ✓		
Dear H ,	✓		
Here are the results of your examinations:	✓		
English		I ✓	
Maths		J ✓	
Computing		K ✓	
Your tutor, L , will no doubt wish to discuss these results with you next term.	✓		
Yours sincerely,	✓		

In the report form, the bold capital letters stand for the actual words, phrases or numbers to be filled in from each student's record. In each case the details will be different but the overall report structure will be the same.

In the example:

A is Mr, Mrs, or Miss depending on the student's status and sex

B is the student's Christian name

C is the student's Surname

D is today's date

E is the first line of the address

and so on.

When you define a report form, you have complete freedom to make it any size and to lay it out in any way you like. The only significant restriction is that it cannot be wider than one page of line printer paper.

5.2 Starting Up

Every report form in TRIMbase is related to a particular data file. This means that you must have defined a file layout (with the file definition module DEFINE) before you can start setting up a report form. It is not essential, but extremely helpful, if your file contains some real data; this will allow you to examine sample reports as you design the report form.

Have the disk with your file definition ready. Then load program REPORT.

The program takes you through the normal dialogue, asking you for the name of the file you want to report on. When your data file has loaded, the screen displays a blank report, and shows you the main menu.

5.3 Choosing a Report Size

The blank report displayed at the start is 70 columns wide and 15 rows deep. It is unlikely that this is exactly the size you want, so you normally begin by setting the dimensions you actually need. Do this by choosing the **File** menu and selecting **Change report size**. Now enter the number of columns and rows into the dialogue box. A rough estimate of the final size will be acceptable unless you are designing the report to fit stationery of a particular size such as address labels. Remember that to be consistent with the GEM conventions, you should use the down-arrow (not RETURN) to move from one line to the next when choosing the report size.

In specifying the size of the report, it is worth remembering that when reports come to be printed, the system will fit as many reports as it can on to one page, without overlapping. If the reports are narrow enough, they will be printed side by side: this is convenient for such things as sticky labels. At its widest, the report is limited to 132 characters. This allows for most line printers in common use. The report can be up to some 200 lines long: the exact figure depends on the machine you are using and the amount of RAM you have made available.

When more than one report appears on a page, the standard separation is two columns (horizontally) and one row (vertically). If you are printing labels on standard label stationery, you will find that the gap between labels is taken into account.

When displaying (not printing) reports, keep the size down to 70 columns and 15 rows if you want to see the entire report on the screen: but larger reports can be scrolled about using the GEM slider bars.

Sometimes you won't know how large your report will be until you have actually typed it. TRIMbase lets you re-adjust the size at any time without losing any text. A good plan is to start with a large blank report and then to shrink it down to the size you really need.

5.4 Typing Text

When you have got the report form the size you want it, you can start typing text. If the report is bigger than the window on the screen you can, of course, move it about with the scroll bars. The report form will also move automatically so as to keep the cursor in view.

Type the text of your report on the keyboard, just as if you were using an ordinary word processor. The cursor keys work in the usual way, except that you are not allowed to move past the end of the text you have already typed. The Delete keys also work the way you would expect.

The **Style** menu allows you to choose bold or underlined characters as you need them.

The system does automatic word-wrap, so don't use the RETURN key unless you are at the end of a paragraph or other break in the text. Any line which ends with RETURN is identified by a ✓ after the last character.

At this point, we suggest that you experiment by typing up a poem or copying a section from this manual. It will take you only a few minutes to get the hang of the controls. If you get really stuck you can always bring down the **File** menu and select **Clear report**.

Remember that you can alter the size and shape of the report form (within limits) even after you have filled it with text.

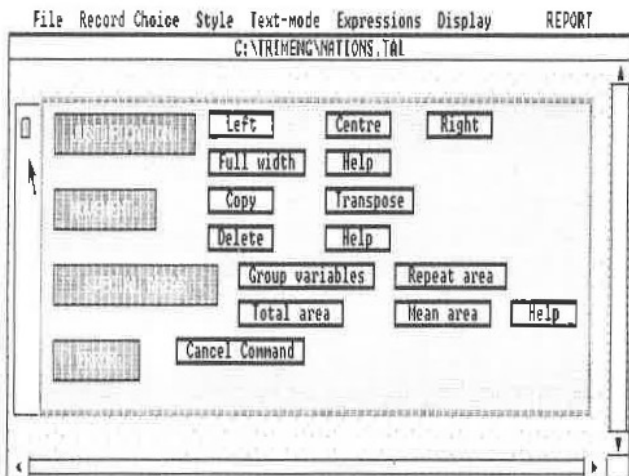
5.5 The Mouse

The mouse has two main uses.

- * You can use it to move the cursor quickly to any position in the report form, just by pointing and clicking.

- * You can use it to define an **area**. An area is a sequence of characters, which may extend over one or several lines. Areas have various uses, which are described below. To select an area, point the mouse at the first (or the last) character, press the left-hand button and keep it down. Now move the mouse to the other end of the area, and let go the button. If the other end of the area is off-screen, TRIMbase scrolls the report form automatically.

When the button is released, the system gives you a choice of twelve options like this:



The first four options have to do with justification, or the way the text is arranged on the page.

Justify to the left

Justify to the right

This line is centered

This paragraph is justified on both sides, and shows you how every line, except the last, is lined up on both sides of the paper. This paragraph includes a considerable admixture of long words — 'extended units of utterance' — because it has to be long enough to show you that the effect is not happening by accident.

Once you choose a type of justification, you will find that the text is rearranged automatically. It remains justified even if you make changes to it such as adding or deleting words. Of course you can always change the justification style at any time.

The next three options are concerned with deleting areas or moving them about:

Copy lets you make a new copy of the area anywhere in the report (except inside the original area). Once you've chosen the copy option, just point to the place you want the copy to go and click.

Transpose is the same as **Copy** except that it deletes the original area. It is a good way of rearranging your text — swapping paragraphs and so on.

Delete gives you a quick way of throwing away everything in the area.

The next group of options lets you **label** parts of your report as **Group**, **Repeat**, **Total** or **Mean** areas. They will be discussed later.

Finally **Cancel Command** returns you to the main control point without making any use of the area you have defined.

5.6 Expressions

Expressions are used to specify those parts of the report which depend on the data in the records in your file.

To put an expression into your report form, choose the **Expressions** menu and select one of the entries. When you have defined the expression (in a way to be described below) it will initially appear in the report form in symbolic form, as a single reversed letter (i.e. white on black). If you want to find out what the letter actually stands for, point at it with the mouse and double-click.

Later, when the reports are being generated from the data file, each symbolic expression is replaced by an actual value.

There are five types of expression:

Simple Compound Count Date Page

Each of these types is described in the following sections.

5.6.1 Simple expressions

A simple expression generally stands for a variable in a record. For example, AGE, NAME and DATE OF BIRTH could all refer to variables in a file about people.

If you specify a simple expression in a report, then each copy of the report will include a different value of that variable. For example, if your report form has the phrase

Dear **A**

and **A** is defined as:

CHRISTIAN_NAME

the various copies will include phrases like

Dear Samantha, Dear Jonathan, Dear Elizabeth,

with the names coming from the records in your data file.

When the value of an expression is copied into the report, the entire text is re-justified.

Simple expressions for text variables are entirely straightforward: you just click on the variable name.

Simple expressions which involve number variables are more flexible, since they allow you to use **formulas**, like this:

PRICE (2)
PRICE * VAT_RATE (5)
12 * FEET + INCHES (0)

In all these examples, the final bracketed number gives the number of decimal places to be used. Thus a value for the first formula might appear as "87.36".

The first part of the formula consists of a sequence of numerical variables or numbers, linked by the signs +, -, * (times) and / (divide). TRIMbase follows the ordinary rules of algebra in working out the formula. Brackets inside the formula are not allowed! You should also ensure that your formula does not involve division by zero.

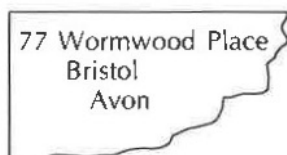
If you choose a class variable, its values are selected from amongst the possible answers.

If you specify a date variable, you are given a choice of seven different ways of writing it, which include figures and words. Perhaps the most interesting format is the **age**, which is defined as the number of days between a date value and today's date. This lets you prepare reports which say:

"I have not heard from you for 60 days . . ."

When you select a block variable, one option is to ask for the entire value straight off. The report generator will behave as if the value had been typed in, and apply the normal justification rules. Newline symbols (if any) will be handled correctly.

On the other hand, you may want to have rather more control over layout. For example, you might decide to stagger the line of an address, like this:



77 Wormwood Place
Bristol
Avon

To make this possible, you can also specify a simple expression which is a particular line (1st, 2nd, 3rd . . .) of a block. In the report, you could define the letters **P**, **Q**, **R** and **S** to be the four lines of address, and you could arrange them like this:



P ✓
Q ✓
R ✓
S ✓

If a block value has no newlines in it, it all counts as the 'first' line. If there are not enough newlines to reach a given line number, nothing is delivered.

As a further illustration, look at the 'exam letter' report at the beginning

of this section. Several of the reversed letters can stand for simple expressions, as follows:

B	CHRISTIAN__NAME
C	SURNAME
E	ADDRESS (Line 1)
F	ADDRESS (Line 2)
G	ADDRESS (Line 3)
H	CHRISTIAN__NAME
L	TUTOR

To define a simple expression, choose the **Expressions** menu and click on **Simple expression**. Then use the mouse to indicate a variable from the record card, or type a number followed by RETURN. If you choose a date or a block, the system will ask you for further details. If you choose a number variable or type a number, TRIMbase assumes that you are entering a formula and asks for an arithmetic symbol. However, you can end the formula at any stage by choosing the last menu item

End formula.

When you enter a formula, remember not to **type** the arithmetic signs. If you do, TRIMbase takes the whole sequence to be a text constant, not a formula at all!

5.6.2 Compound expressions

Compound expressions give you a much more flexible way of deciding what to put in your report. A compound expression consists of a list of conditions (like selection conditions). Each condition is followed by a simple expression, or a constant sequence of characters, with an understanding that it is only to be used if the condition turns out to be **true** (and all the ones before to be **false**). The final condition is always 'Otherwise', which acts as a catch-all.

Here are some examples, which we suggest you study carefully:

HEIGHT More than (>) 1.90	→ "Too tall"
HEIGHT Less than (<) 1.50	→ "Too small"
Otherwise	→ "Acceptable"

This expression would appear as "Too tall" whenever the value of HEIGHT was over 1.90, and as "Too small" if the value of the variable was below 1.50; otherwise it would be displayed as "Acceptable".

EDUCATION Is (=) PHD	→ "Dr"
PROFESSION Is (=) Physician	→ "Dr"
SEX Is (=) MALE	→ "Mr"
PROFESSION Is (=) Cook	→ "Mrs"
STATUS Is (=) SINGLE	→ "Miss"
STATUS Is (=) ?	→ "Ms"
Otherwise	→ "Mrs"

Notice that the conditions are searched from the top down, and the search is abandoned as soon as a condition is found to be true. This expression is designed to address anyone with a Doctorate of Philosophy, or any physician, as "Dr", irrespective of sex or status. Likewise, any male who is not a doctor (real or titular) is addressed as "Mr." and any lady who is a cook receives the courtesy title of "Mrs." whether she be married or no.

The value which follows each condition can also be a variable or a formula. A possible expression might be:

AGE More than (>) 21	→	"Over 21"
Otherwise	→	AGE (0)

Compound expressions can handle most of the remaining fields in the exam letter. In particular we would define the following:

A :	SEX Is (=) MALE	→	"Mr"
	STATUS Is (=) SINGLE	→	"Miss"
	Otherwise	→	"Mrs"
I :	ENGLISH More than (>) 69	→	"Distinction"
	ENGLISH More than (>) 49	→	"Pass"
	Otherwise	→	"Fail"

and similarly for **J** (Maths) and **K** (Computing).

5.6.3 Count

The **Count** option is a shorthand way of writing compound expressions like:

AGE More than (>) 50	→	1 (0)
Otherwise	→	0 (0)

It is used in defining summary reports, as described in 5.14.

5.6.4 Today's Date

One option for an expression is today's date — that is, the date on which the report is generated. The date can be presented in the same choice of formats as dates taken from individual records. Expression **D** in the exam letter is an expression of this kind.

5.6.5 Page Number

The last kind of expression is the page number. This is not derived from a record, but simply takes the values 1, 2, 3, . . . for successive reports. A well designed report would include the date and the page number as a heading, like this;

Current Stockholding on A	Page B
----------------------------------	---------------

where **A** represents today's date and **B** the page number.

If this line is placed at the head of the report form, then an individual report might be headed:

Current Stockholding on 31st May 1987	Page 7
---------------------------------------	--------

with the details dependent on the choice of date format. Each report will have a different page number.

5.7 Selecting Records

You will often need to make a report which uses only part of the data file. You can pick the records you want to be included by choosing the **Select** menu and clicking on the **Record Choice** option. You are then allowed to enter a selection rule in the same way as in the MAINTAIN module. Once such a rule is set up, the only records to be included in the report are those which conform. For example, if your rule is

SEX Is (=) FEMALE

your report will only include those records where the class variable SEX has the value FEMALE.

The other options in the menu are **Display rule**, which displays the current selection rule, and **Delete rule**, which cancels it altogether and allows **all** the records in the file to be included in reports. If you set up a new selection rule the previous one is cancelled automatically.

5.8 Trial runs

While designing a report form, you may wish to see how it looks with genuine data, as opposed to symbolic expressions. Choose the **Display** menu and click on **Sample**. TRIMbase then shows you a report based on the first record in the file. To get further samples, click the button on the mouse or press RETURN. To go back to the main menu, press **Esc** or double-click.

TRIMbase goes back to the main menu automatically as soon as it runs out of records. If there are no records in the file or if all the existing records are eliminated by the selection rule, this happens immediately.

You can ask for a trial run at any time, even if the report form is only part-complete.

5.9 Printing reports

When you are satisfied with your report form, you can print the reports. Select the **Display** menu and click on **Print**.

First, TRIMbase will ask you for details of your printer, if it has not already done so in the current session. Then it pauses while you load and line up the stationery. It is useful to know that printing starts without any preliminary paper movement.

When you are ready, click the Ready panel and TRIMbase will ask you for a starting page. The first page is always '1', but you may give another page if — for example — you previously ran out of stationery half-way through a run. If you give a high page number, expect a short delay before printing starts.

When the printing has started, you can stop it at any time by pressing the **Esc** key or switching off the printer. When you restart, you must re-align the paper.

When you are preparing a report, TRIMbase allows you to embed printer control codes in the text, or in string expressions. This is particularly useful if your printer is a sophisticated model. You will have full control over colour printers and most laser printers as well.

A printer control code is typed as a reverse slash followed by a decimal number in the range 0 to 255. For example, suppose that your printer gives you a choice of typestyles and that 'italic' is selected by the sequence 'ESC I'. The table in Appendix 1 shows you the corresponding decimal values are 27 and 73, so to switch your printer into italic mode, type '\27\73'.

Two points should be noted:

- 1) The screen display does not take account of control codes but displays them in clear, like '\27' or '\73'. This could cause problems with justification and layout. Isolate your control codes with newlines.
- 2) If you wish a reverse slash in a report to appear as such, type two reverse slashes '\\'.

If printing gives you problems, make sure that, if necessary, you have run the configuration program described in Appendix 1.

5.10 Printing individual reports

When you are examining samples (as described in 5.8) you may wish to print the one currently on the screen. To do this press the 'P' key. The report is printed without preliminary page movement, so it is up to you to have the paper lined up in the printer before you start.

5.11 Saving and restoring report forms

TRIMbase supports a facility for preserving report forms on the disk, and bringing them back for use later. Files which contain report forms are subscripted ".REP".

To save a report form, select the **File** option and click on **Save report form**. Then use the item selector panel to choose a name.

To retrieve a report form, choose the **File** option and pick **Load report form**. Then select the name of the report form you need. The report form is only loaded if it matches the data file you specified at the beginning of the run; if it doesn't, the system displays a warning message.

If you alter a card layout after you have set up a report for that card, be careful:

You may: Move variables about
 Define new variables

You may not: Delete variables
 Alter the definition of existing variables

Either of these changes will invalidate the report form, and the system will not allow you to use it.

5.12 Layout Control

When TRIMbase constructs a report, it treats the various expressions and phrases from the records exactly as if you were typing them on the keyboard. Since the expressions are typically of different lengths, you need some way of controlling the layout of the report.

To arrange your report in regular columns, you should make use of **Anchor points**. An anchor point is set by typing the character ~ . This is over the # sign at the right of the keyboard. When the report form is being designed, an anchor point looks like this: ◆. It is handled the same way as any other character. However, when the report form is actually being filled in with real data, the anchor point turns into a space with its horizontal position fixed. If you put an expression between two anchor points, it will always occupy the same horizontal position regardless of its length, or of the lengths of any other expressions. It follows that in a tabular report, the entries across the line should be preceded by anchor points, like this:

Name	◆	Crime	◆	Sentence	◆	Date of Release	◆	Category
A	◆	B	◆	C	◆	D	◆	E

An expression between anchor points can be justified in all the usual ways, so that it can appear on the left, on the right, or in the middle of its field.

If an expression is too long to fit between two anchor points on the same line, TRIMbase gives you a display with a warning. When you press RETURN, the expression is cut down to size (losing one or more characters) and the system continues. Truncated numbers, however, are replaced by asterisks to avoid the display of incorrect values.

5.13 Reports made from groups of records

This section describes how to make reports which cover groups of related records. It is one of the most powerful features of the TRIMbase report generator.

We start with a simple example. Consider a file with records about people, like this:

SURNAME: [T20]
FIRST NAME: [T17]
AGE: [N]	
NUMBER OF PETS: NN	

We wish to make a series of reports about families. Each report is to list the members of a single family, thus:

Family name:	Cadwallader
Members:	
Jemima	(age 35)
William	(age 37)
Annabel	(age 17)
George	(age 10)

This particular report uses information from four separate records — one for each member of the family. Other reports in the same series will have different numbers of entries, depending on the sizes of the families concerned.

The report form which generates these reports has one key feature — the 'repeat area'. First, look at a simple report form like this:

Family name:	A ✓
Members: ✓	
B	◆ (age C) ✓

where **A** is SURNAME
B is FIRST NAME
and **C** is AGE (0)

This report form has no repeat area, and will therefore produce one report per record, like this:

Family name:	Cadwallader
Members:	
Jemima	(age 37)

Family name:	Cadwallader
Members:	
William	(age 39)

and so on.

To make the essential modification to the report form, we identify the part of the report which is to be repeated, and label it as a 'repeat area'. This is done with the mouse, in the same way as you would select an area to be copied, or to be justified in some particular way. The repeat area is marked by a distinctive background.

In this case, the information to be repeated for each member of the family is the first name, the age, and the other (fixed) characters in the same line such as the spaces, the anchor point, and the word 'age'. We put them into a repeat area, like this:

Family name:	A ✓
Members: ✓	
B	◆ (age C) ✓

← Repeat area

Now, when TRIMbase constructs the report, it fills in the repeat area for **each** of a number of records. Which records? Precisely those which have identical values for the expression (or expressions) which are **not** in the repeat area. In this case, the only such expression is the surname Cadwallader, so TRIMbase includes **all** the records which have this value of surname. The result is what we need, namely:

Family name:	Cadwallader
Members:	
Jemima	(age 35)
William	(age 39)
Annabel	(age 17)
George	(age 10)

Variables like SURNAME, which are used to identify the members of a group, are called **group** variables. By contrast, variables which may be different over the records in the group, such as FIRST NAME and AGE, are called **individual** variables.

At this point, we introduce an essential rule. For the 'repeat' feature to work correctly, the file must first be **sorted** so that all the records in each report are next to one another. In our example, the file would have to be sorted by SURNAME.

When you set up a report form to handle more than one record, the simplest way is to construct the form as if you were planning to display one record only. Get the layout right, and then, as the last change, mark in the repeat areas. Anchor points inside repeat areas are specifically useful in arranging for neat lined-up columns of items.

Remember to include at least one 'newline' symbol \checkmark in the repeat area if you want the information from the various records to fall on different lines.

If you mark a repeat area incorrectly, you can always cancel it by marking a 'group area' in the same place.

There are two further kinds of area you can use. If you mark a total area, everything inside will only appear once but any numerical expression will be totalled over all the records in the report. Similarly, a mean area will display the average of all the numerical expressions.

Here is a report form which uses these features:

The A Family \checkmark		
\checkmark	Number of members :	B \checkmark ← Total area
\checkmark	Average age :	C \checkmark ← Mean area
\checkmark	Number of pets :	D \checkmark ← Total area

where **A** is SURNAME
B is 1(0)
C is AGE (2)
and **D** is NUMBER OF PETS (0)

The corresponding report would be:

The Cadwallader Family	
Number of members :	4
Average age :	25.75
Number of pets :	9

The maximum number of records which can be included in a summary report is 64.

To complete this section, here is the report form which could be used to produce the Tutor Summary report we described earlier:

File Record Choice Style Text-mode Expressions Display RT				
C:\GEN3\NEWDT\STUD2.TAL				
Tutor: GW				
Name	Mark in English	Maths	Computing	Average/
1	0	0	0	0
2	0	0	0	0
Average:	0	0	0	0

Expressions are defined as follows:

- A Tutor name
- B, C Student name
- D, H English mark (0)
- E, I Maths mark (0)
- F, L Computing mark (0)
- G, M English mark/3 + Maths mark/3 + Computing Mark/3 (0)

The line with expressions **B** to **G** is put into a repeat area. Expressions **H** to **M** are in the Mean area.

5.14 Summaries made over a complete file

A common need is for reports with totals and averages over a complete file (or over all the selected records if a selection rule is in force). Such a summary is produced automatically with any report for where **all** the expressions are in **Total** or **Mean** areas. The limit of 64 records does not apply in this case.

- * To get a total or average, specify an expression and put it into a Total or Mean area.
- * To get the total number of records included in the summary, use the simple expression

1(0) (**one** with no decimal places)

in a Total area.

(This works because the system adds 1 for each record, thus producing the total number of records.)

- * To count the number of records which have a specific value, such as a particular value for a class variable or one which conforms to a specific condition, use a compound condition like:

CLIMATE is (=) TROPICAL → 1(0)
Otherwise → 0(0)

or

Age greater than (>) 65 → 1(0)
Otherwise → 0(0)

(This works correctly because the system counts 1 for each record where the condition is true and 0 for all the others).

TRIMbase offers a shorthand way of entering such an expression. The **Count** option in the **Expression** menu lets you supply the condition (such as CLIMATE is (=) TROPICAL) and fills in the rest of the expression automatically.

5.15 Filing Reports

The **File report** option lets you send a complete report (not a report form!) to a file instead of having it printed. This is useful if you wish to use the report as input to another program.

Please note that filed reports do not distinguish bold or underlined characters.

5.16 Moving to other modules

Use the commands **Go to DEFINE** or **Go to MAINTAIN** to transfer control to either of the other two modules. If you go to **MAINTAIN**, the name of the current data file is transferred, and this file is opened and loaded automatically. This is particularly useful in sorting a data file to produce reports in different orders.

5.17 A note on layout

This section describes the way TRIMbase deals with reports that have grown too big to fit on the page.

In a report form, expressions are represented by single inverted letters; but when the actual report is constructed, each expression is replaced by text which can be of any length. Actual reports are therefore generally longer than the corresponding report forms, and it is possible for a report to outgrow the size allocated to the report form.

When this happens, a message is displayed and the printing or display process is aborted. The cure is to redesign your report form.

When a report form contains repeat fields, the situation is more complex.

Consider a file where a large number of records share the same group variables — too many to fit on to a single page. TRIMbase will automatically split the records into several reports, ensuring that no information is lost. The print or display process is not aborted unless the fixed material plus **one** record is enough to exceed the report size.

When TRIMbase prints reports, it will arrange them side by side if there is enough space on the page. This provides a simple way of printing labels on 'two-across' or 'three-across' stationery. Likewise, several short reports can appear below one another on one page of line-printer paper.

PART 6: RELATIONAL OPERATIONS

6.1 Introduction and Overview

A relational data base is far more flexible and powerful than a standard filing system. It can extract data from more than one file and combine this data in various useful ways. This means that record cards can be kept small and manageable and data should only ever need to be typed in once!

This chapter is about the relational operations in TRIMbase. We begin with a brief overview of the main ideas involved. Readers already familiar with the relational data model are welcome to skip this section and go straight on to the more practical discussions.

Relational operations are always carried out on complete files. Using the TRIMbase notation, a file consists of a number of records, each of which holds some variables. We can represent a file as a table, as shown below.

Note that SN stands for 'serial number'. The serial numbers play almost no part in the relational operations, but we include them because every record in TRIMbase carries its own number.

SN	Name	Department	Car owned
1	Gordon Grubb	Toys	Fiesta
2	Maria Gobstopper	Ladies Lingerie	Robin Reliant
3	Geoffrey Stallibrass	Toys	
4	Percival Foulkes	Hardware	Porsche, Mini
5	Agatha Bassey	Ladies Lingerie	Saab
6	Rajnikant Patel	Toys	Landrover, Sprite
7	George Green	Nightwatchman	Cortina

The two simplest relational operations, **Projection** and **Selection**, are used to discard some of the data in a file. **Merge** and **Join** link the data in two files in different ways and **Subtraction** is yet another way of getting rid of unwanted data items.

6.1.1 Projection

When you project a file, you make a new file which has some (but not usually all) of the columns of the old one. For example, if you project the sample file, omitting the Department variable, you'll get:

SN	Name	Car Owned
1	Gordon Grubb	Fiesta
2	Maria Cobstopper	Robin Reliant
3	Geoffrey Stallibrass	
4	Percival Ffoulkes	Porsche, Mini
5	Agatha Bassey	Saab
6	Rajnikant Patel	Landrover, Sprite
7	George Green	Cortina

6.1.2 Selection

Selection works by omitting certain rows of the original file. For example you might make a selection which includes only the employees in the Toy Department. This gives

SN	Name	Department	Car owned
1	Gordon Grubb	Toys	Fiesta
3	Geoffrey Stallibrass	Toys	
6	Rajnikant Patel	Toys	Landrover, Sprite

6.1.3 Merging

Merging is a way of linking two files which have exactly the same layout and arrangement for their records. Suppose, for example, that there are two new departments, with the following personnel file:

SN	Name	Department	Car owned
1	James Allen	Dress hire	Mini
2	Agnes Arbuthnot	Dress hire	Cortina
3	Anne Forsythe	Perfumes	
4	Jemima Cadwallader	Perfumes	Jaguar XK140

Since both personnel files have the same arrangement, they can be merged to produce:

SN	Name	Department	Car owned
1	Gordon Grubb	Toys	Fiesta
2	Maria Gobstopper	Ladies Lingerie	Robin Reliant
3	Geoffrey Stallibrass	Toys	
4	Percival Ffoulkes	Hardware	Porsche, Mini
5	Agatha Basse	Ladies Lingerie	Saab
6	Rajnikant Patel	Toys	Landrover, Sprite
7	George Green	Nightwatchman	Cortina
8	James Allen	Dress hire	Mini
9	Agnes Arbuthnot	Dress hire	Cortina
10	Anne Forsythe	Perfumes	
11	Jemima Cadwallader	Perfumes	Jaguar XK140

Note that the records from the second file have been renumbered.

6.1.4 Joining

Joining allows for the extraction of information from two files with **different** record cards. When two files are joined, **every** record from the first file is compared with **every** record from the second. Whenever a pair is found to match (according to some rule defined by the user), a new record is made which holds **all** the data from **both** the original records. To give an example, suppose that there is a file about departments, in which each record has information about the manager and the location. The file might read:

SN	Department	Manager	Location
1	Hardware	William Green	3rd floor
2	Toys	Rajnikant Patel	Basement
3	Ladies Lingerie	Olga Swerchkovska	4th floor
4	Dress Hire	Angus Ogilvie	Mezzanine
5	Perfumes	Shane Brain	2nd floor

Let us now join this file with the expanded personnel file. We say (in this case) that two records match if they have the same **department**. When files are cross-matched, Gordon Grubb's record will match the Toy Department (since that is where he works), Maria Gobstopper's

FIGURE 6.1

SN	Department	Manager	Location	Name	Department	Car owned
1	Toys	Rajnikant Patel	Basement	Gordon Grubb	Toys	Fiesta
2	Ladies Lingerie	Olga Swerchkovska	4th floor	Maria Gobstopper	Ladies Lingerie	Robin Reliant
3	Toys	Rajnikant Patel	Basement	Geoffrey Stallibrass	Toys	
4	Hardware	William Green	3rd Floor	Percival Ffoulkes	Hardware	Porsche, Mini
5	Ladies Lingerie	Olga Swerchkovska	4th Floor	Agatha Bassey	Ladies Lingerie	Saab
6	Toys	Rajnikant Patel	Basement	Rajnikant Patel	Toys	Landrover, Sprite
8	Dress Hire	Angus Ogilvie	Mezzanine	James Allen	Dress Hire	Mini
9	Dress Hire	Angus Ogilvie	Mezzanine	Agnes Arbuthnot	Dress Hire	Cortina
10	Perfumes	Shane Brain	2nd Floor	Anne Forsythe	Perfumes	
11	Perfumes	Shane Brain	2nd Floor	Jemima Cadwallader	Perfumes	Jaguar XK140

Data from 1st File

Data from 2nd File

will match with that for Ladies' Lingerie, and so on. The only personnel record which does not match at all is that of George Green, who is the night watchman. The result is a new file, with one record for every person who works in a department. Apart from the serial number, each new record includes **all** the information from **both** the records which were combined to construct it. This makes it rather wide, as shown in Figure 6.1.

Such a file could be used, for example, to discover where a given employee works, or who is his or her departmental head.

In practice, a joined file only needs a few of the variables from each record. As we shall see, TRIMbase allows you to select the variables you need.

Join is one of the most useful operations in a relational data base.

Consider a personnel file, like the one we have just described. In a simple 'flat' filing system, everything you might ever need to know about an employee (such as his Head of Department, exact place of work, list of dangerous materials handled, and so on) must be stored in that employee's own record. This means a tremendous amount of duplication since all the employees in one department will have a great many facts in common. It also means that if anything is changed about the department (such as the name of its head), all the employee records have to be updated.

A relational data base, with its **join** operation, allows you to escape from this trap. You use the employee records only to store facts about them as individuals — such as names and addresses — and you put the information about the departments into a separate file. This saves space and maintenance time whilst the more general information you need about individual employees can easily be retrieved by joining the two files. In a relational system, records are generally short. They do not need to be as big as those in a simple non-relational data base.

6.1.5 Subtraction

Subtraction needs two files with identical layouts, which we shall call A and B. When B is subtracted from A, the result is a file which contains every record in A unless it is also represented in B. For a record to be deleted from A, the copy in B must be identical except for the serial number.

Here is an example:

File A:

SN	Name	Car
1	James Bond	Aston Martin
2	Jemima Cadwallader	Jaguar XK140
3	George Mulder	Mini
4	Craig Renfrew	Astra
5	Maria Gobstopper	Robin Reliant
6	Jayne Bloggs	Fiesta
7	Bill Stewart	Maxi
8	Shayne Brain	Porsche

FILE B:

SN	Name	Car
1	Yvonne Mayhew	Citroen 2CV
2	Bill Stewart	Maxi
3	Shayne Brain	Lamborghini
4	Maria Gobstopper	Robin Reliant
5	James K. Bond	Aston Martin
6	Jayne Bloggs	Fiesta
7	George Mulder	Mini

The result of subtracting B from A is

SN	Name	Car
1	James Bond	Aston Martin
2	Jemima Cadwallader	Jaguar XK140
3	Craig Renfrew	Astra
4	Shayne Brayne	Porsche

You are invited to work out for yourself the result of subtracting A from B!

Subtraction provides a useful way of listing exceptions. For example, suppose we have a file which records who sells what:

SN	Name	Item sold
1	Bill	Apples
2	John	Bananas
3	Nell	Oranges
4	Geoff	Strawberries
5	Susan	Bananas
6	Andrew	Grapes
7	John	Oranges
8	Mark	Lemons
9	Veronica	Lychees
10	Craig	Bananas
11	Andrew	Bananas
12	Nell	Grapes
13	Veronica	Apples

To make this example sensible, please imagine that this file is thousands of records long!

As a banana importer, you need to make a list of all the people who don't (already) sell bananas. This can be done by a sequence of relational operations:

- a) Project the main file to give a list of names.

SN	Name
1	Bill
2	John
3	Nell
4	Geoff
5	Susan
6	Andrew
7	John
8	Mark
9	Veronica
10	Craig
11	Andrew
12	Nell
13	Veronica

b) Sort this list and remove duplicates (in TRIMbase this is done by the MAINTAIN module). The result is called the 'list of people'.

SN	Name
6	Andrew
1	Bill
10	Craig
4	Geoff
2	John
8	Mark
3	Nell
5	Susan
9	Veronica

c) Now go back to the original file and select all the people who sell bananas. This gives

SN	Name	Item sold
2	John	Bananas
5	Susan	Bananas
10	Craig	Bananas
11	Andrew	Bananas

d) Next, project this file to keep only the names. The result is called 'The banana sellers':

SN	Name
2	John
5	Susan
10	Craig
11	Andrew

e) Finally, subtract the banana seller file from the 'people' file. The result is what we need: a list of people who don't sell bananas.

SN	Name
1	Bill
4	Geoff
3	Nell
9	Veronica

6.2 The Relational Operations in TRIMbase

This part of the chapter explains how the relational operations are supported in TRIMbase, and discusses the details of their use.

In TRIMbase, all the relational operations are implemented in module DEFINE. This is consistent with the role of the module as the maker of new files. The operations are commanded by menu selection, and it is never necessary to use the keyboard except when you are typing a constant value or specifying the name of a new file.

6.2.1 Projection

Projection consists of discarding some of the fields in an existing file. To project a file, pick the Projection item in the **Relational** menu. TRIMbase then takes you through the following steps:

- It shows the item selector panel. Choose your source file.
- The system shows you a dummy card with field names but no values. Click on the fields you want to **keep** in the projected file. To correct a mistake, click again. When you are ready, hit RETURN.

Notice that the serial number is included automatically.

- TRIMbase now displays a new record card with the variables you have chosen. You can adjust the size and layout of this card in the usual way, by moving fields about and changing the shape of the card itself. Again, when you are ready, hit RETURN.

- Finally, the system displays the selector panel again. Choose a new name for the target file. Don't remove the disk with the original data until the operation is complete. You should make sure there is enough room on the disk to hold the new file; it helps to know that it must be shorter than the original.

6.2.2 Selection

You will already be familiar with Selection, since it is used in both the other modules of TRIMbase.

In the context of relational operations, selection is useful if you want to partition a very large file into a number of smaller ones for easy manipulation. It is also handy for removing records which you prefer to keep secret before giving other users access to the file.

Selection can also be used (in a negative sense) to delete batches of unwanted records or even to create a new 'empty' file with the same record card description as an existing one.

To use the Select operation, choose **Selection** on the **Relational** menu. Then:

- a) Pick a source file, just as you would do for Projection.
- b) Define a selection expression. This is done in exactly the same way as in MAINTAIN and REPORT.
- c) Name your target file.

The rest is automatic.

6.2.3 Merge

The Merge operation forges a single target file from two source files with **identical** record cards. Merging is the opposite of selection; for example, if you use selection to split a given file into two separate parts, you can reconstruct the original file (except for the order of entries) by merging the two parts.

When you merge, you name two source files. The records from the first are appended to the end of the second and are renumbered to avoid clashes of serial numbers. There is no need to name a target file since this is the same as the second source.

An important point to note is that two files cannot be merged unless their record card formats are **exactly** the same. It is not enough for the formats to **look** alike, since there will often be internal differences which prevent the files from being merged. The only way to ensure compatibility is to 'grow' your files from a single definition.

Suppose that you have defined a file called — say — MASTER.TAL. When you come to enter data for the first time, you use MASTER.TAL, but record the result as FILE1.TAL, using the **Save whole file as** option. This leaves MASTER.TAL empty.

The second time you again use MASTER.TAL, but store the results as FILE2.TAL. FILE1 and FILE2 have now grown from the same definition, so they are guaranteed to merge correctly.

To implement a merge, choose the **Merge** option in the **Relational** menu. Then name the two files concerned. Remember that the second one will be altered by the operation!

6.2.4 Join

The theory of the Join operation was discussed at length at the beginning of this chapter. The TRIMbase version of Join lets you discard all the variables you don't need, so in theoretical terms it is a combined Join and Project operation. It is perhaps worth repeating the example, using TRIMbase conventions.

a) The file about individual employees might have record cards like this:

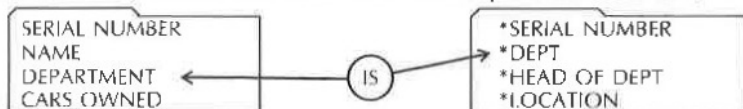
SERIAL NUMBER: 1 NAME: Geoffrey Stallibrass DEPARTMENT: Toys CARS OWNED: Fiesta	SERIAL NUMBER: 2 NAME: Maria Gobstopper DEPARTMENT: Ladies Lingerie CARS OWNED: Robin Reliant
SERIAL NUMBER: 3 NAME: Geoffrey Stallibrass DEPARTMENT: Ladies Lingerie CARS OWNED: Saab	SERIAL NUMBER: 4 NAME: Percival Ffoulkes DEPARTMENT: Toys CARS OWNED: Landrover, Sprite
SERIAL NUMBER: 5 NAME: Agatha Bassey DEPARTMENT: Ladies Lingerie CARS OWNED: Saab	SERIAL NUMBER: 6 NAME: Rajnikant Patel DEPARTMENT: Toys CARS OWNED: Landrover, Sprite
SERIAL NUMBER: 7 NAME: George Green DEPARTMENT: Nightwatchman CARS OWNED: Cortina	

b) The file about departments will have records like this:

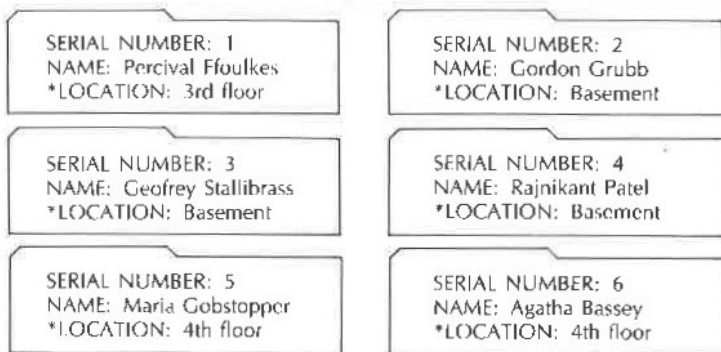
SERIAL NUMBER: 1 DEPT: Hardware HEAD OF DEPT: William Green LOCATION: 3rd Floor	SERIAL NUMBER: 2 DEPT: Toys HEAD OF DEPT: Rajnikant Patel LOCATION: Basement
SERIAL NUMBER: 3 DEPT: Ladies Lingerie HEAD OF DEPT: Olga Swerchkovska LOCATION: 4th Floor	

This sets the scene. Now suppose that the store manager asks for a list of all employees with their locations of work. The information is clearly present in the data base, but it is scattered across two different files. However, it can be retrieved automatically by joining these two files in the right way. We shall produce a new record whenever the Department in an 'employee' record is the same as the Department in a 'department' record.

This is shown in the diagram below. We use the convention that variables from the second file have their names prefixed with a *.

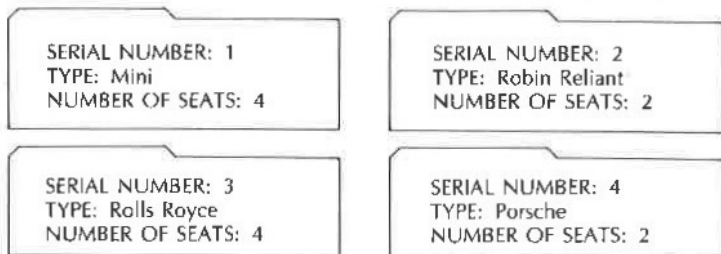


In our example, the records for Gordon Grubb, Geoffrey Stallibrass and Rajnikant Patel all match with the Toy Department record. Likewise Maria Gobstopper's and Agatha Bassey's records match the one for Ladies' Lingerie, and Mr. Ffoulkes' the one for Hardware. In this case, the join produces one new record for each employee except the nightwatchman. All the redundant variables are thrown away, and the new set of records looks like this:



When two files are joined, the number of records produced can be considerably more than the number in either of the two source files.

For example, suppose that the store manager is organising the annual summer trip to the sea-side, and wants to know how many car seats are available in each department. The individual records show that some employees have no car, some have one, and some have even more than one. The manager gets hold of a car catalogue and sets up a 'car file' which gives the number of seats in each type of car. Some of the entries might be:



The information needed to organise the trip can be produced by joining the employee file with the car file, using the relation

CARS OWNED Includes ({})*TYPE

This generates a record for every car owned by an employee. With a proper choice of variables, some of the records could look like this;

SERIAL NUMBER: 1
NAME: Gordon Grubb
DEPARTMENT: Toys
*TYPE: Fiesta
*NUMBER OF SEATS: 4

SERIAL NUMBER: 2
NAME: Percival Ffoulkes
DEPARTMENT: Hardware
*TYPE: Porsche
*NUMBER OF SEATS: 2

SERIAL NUMBER:3
NAME: Percival Ffoulkes
DEPARTMENT: Hardware
*TYPE: Mini
*NUMBER OF SEATS: 4

SERIAL NUMBER: 2
NAME: Maria Cobstopper
DEPARTMENT: Ladies Lingerie
*TYPE: Robin Reliant
*NUMBER OF SEATS: 2

Once such a file has been produced, it can be sorted (possibly into departmental order) and passed over to the report generator for a suitable summary to be printed.

In TRIMbase the practical details of specifying a Join operation are straightforward. When you select the **Join** option, the system takes the following steps:

- a) It shows you the item selector panel and lets you choose the first source file. This file is then opened.
- b) This is repeated for the second source file.
- c) TRIMbase now asks you to supply a 'joining' expression. This is slightly different from a selection expression because each condition includes one variable from the first file, and one variable from the second. Constant values (such as numbers or specific dates) are not needed.
- d) The system then presents you with a blank record card for the first file, and lets you choose the variables you would like to include. Don't choose too many!
- e) This is repeated for the second file.
- f) The record card for the joined file is displayed, and you can adjust its appearance in the usual way.
- g) Finally, TRIMbase asks you for the name of the new file. Don't choose an existing name, don't remove the disk (or disks) with the source files, and make sure that there is enough room on the disk for the new file.

When all this information has been supplied, TRIMbase constructs the new file and tells you how many records it has made.

The Join operation has many different uses. It is possible to join a file to itself; for example, here is an expression which would find all the pairs of twins from among a large file of personal records:

```
& BIRTHDAY Equals (=) *BIRTHDAY
& MOTHERS NAME Equals (=) *MOTHERS NAME
& SERIAL NUMBER Less than (<) *SERIAL NUMBER
```

The first two conditions select any pair of records for people who were born on the same day of the same mother. The last condition is necessary to prevent records from being joined to themselves.

It is always best to use great care when you define a join expression. It is easy to set up expressions which have no sensible meaning but are true in most cases, like

```
SURNAME Is not (!=) *TYPE OF CAR
```

The result will be a very large file where none of the records has any real meaning!

There is one obscure error which the system will catch. Suppose you are setting up a join expression, and you choose a variable of some given type (say — a date) from the first card. If the second card has no variables at all of the same type, the join expression can't be completed. TRIMbase issues a warning and takes you back to the main control point.

An extended example of the use of a relational data base is given in the next chapter.

6.2.5 Subtraction

You will remember that when one file is subtracted from another, every record represented in the second file is removed from the first, if present. For example, if you start with:

```
                (a b e f i k l)
and subtract    (b c e i j k)
you will be left with    (a f l)
```

The two source files must have identical record cards. To qualify for removal, two records must be exactly the same except for serial numbers.

To specify the subtraction of two files, bring down the **Relational** menu and choose **Subtraction**. Then name your two source files and the result file.

6.2.6 Deletion

This command can be used to delete any file in the system (not only TRIMbase data files). The command must be used with care, since files, once deleted, cannot be brought back. Delete is not strictly a relational operation, so the command is not in the relational menu but the file menu.

6.3 Conclusion

Relational operations form a powerful if somewhat opaque way of handling files. In many cases, applications can only be carried out by the right sequences of operations. A method of setting up such a sequence and obeying it automatically is described in the next chapter.

PART 7: MISCELLANEOUS MATTERS

This final part of the TRIMbase User Guide deals with two features which may interest the more advanced user.

7.1 Import

You may wish to set up a TRIMbase file using data currently stored in a different data base. There is no need to retype all the data, but you must convert it from its current form into **Standard TRIMbase Import Format**. This is a once-only operation. If you have any problems, you should consult someone who is expert in your present system.

The Standard TRIM Import Format is a file of ASCII characters. It has a file name ending in ".IMP".

All characters at the start of the file are ignored, up to and including the first "*" (asterisk). This allows for the invisible file headers produced by some programming languages and word processors.

Next, the file contains characters which correspond exactly to the key depressions you would make if you were entering the file manually. The only difference is that the 'insert' character is expressed by the reverse slash (' \ ') which is reserved for this purpose.

To give a simple example, consider a personal record card like this:

SERIAL NUMBER: [N10]
CHRISTIAN NAME: [T20]
SURNAME: [T25]
AGE: [N	
SEX: [C5	

A two-record import file could consist of the characters

```
xxxxxxxxxxxxx *  
Julius  
Caesar  
56  
M  
\  
Lucretia  
Borgia  
25  
F  
\
```

A file like this one can be produced by many programming systems including BASIC. Here is a short AMSTRAD BASIC program to set up such a file from manual input:

```

INPUT "Generic file name"; f$
file = 6
OPEN # file OUTPUT f$ + ".imp"
PRINT # file, "*"
REPEAT
  INPUT "Christian name"; c$
  PRINT # file, c$
  INPUT "Surname"; s$
  PRINT # file, s$
  INPUT "Age"; a
  PRINT # file, a
  REPEAT
    INPUT "Sex (M or F)"; g$
    UNTIL (g$="M" OR g$="m" OR g$ = "F" OR g$ = "f")
    PRINT # file,g$
    PRINT # file, "V"
  INPUT "Any more records? (Y or N)"; m$
  UNTIL (m$="N" OR m$ = "n")
CLOSE # file

```

This example leaves out the serial numbers because it makes use of the automatic serial numbering feature.

The last 'V' inserts the final record and should not be followed by a newline, but in practice it makes little difference if this newline is not removed.

Once an import file has been set up on a disk, it can be loaded into a TRIMbase file by the following process:

Load and run the MAINTAIN module in the usual way. Then open the **Input** menu and click on **Import**.

Next, ensure that the disk with the import file is loaded and select the file, using the item selector panel.

From now everything is automatic. You will see your records being read in as if they were being keyed by a demon typist. When the import file ends, control is transferred to the mouse and keyboard in the usual way. The import file should be correct. If it has systematic errors (such as too few fields per record) there is nothing you can do except make another import file and try the whole process again.

To avoid time-wasting catastrophes, TRIMbase imposes a deliberate limit of about 500 new records in any one session of input. This limit also applies to data read from import files. The rule is: no single import file should exceed 500 records in length.

In practice this restriction is easily handled by partitioning a large file into several segments.

7.2 Sequences

The DEFINE module has a special facility for remembering sequences of commands and playing them back later. This is useful if you do any complex operations on a regular basis — weekly or monthly.

To remember a sequence of commands, open the **File** menu and click on **Start sequence**. From now on, any command you give will be stored in a temporary memory buffer as well as being obeyed. When the sequence ends, open the **File** menu again and click on **Record sequence**. Then choose a file name, and the complete sequence will be recorded permanently on your disk. The file name has the subscript ".SEQ".

To play back a given sequence, open the **File** menu and click on **Playback**. Then choose the sequence file from the Item Selector panel. Note that all the disks you need for the sequence should already be loaded since there is no opportunity to change them as the sequence is obeyed. Furthermore, the sequence is replayed exactly as it was recorded, and there is no scope for variation such as alternative file names. If you need flexibility, or if you need to change disks, then split the sequence into two parts separated by manual commands.

If a command in a sequence cannot be obeyed (for example, because the file it refers to has been deleted) the system simply aborts the sequence and returns to manual control.

7.2.1 A Case Study

To illustrate the use of sequences and a number of other commands, there now follows a detailed example of a TRIMbase application.

To set the scene, imagine that you run a wholesale distribution business for musical instruments. You carry a large stock of clarinets, french horns, guitars etc. Your customers place orders (typically by phone) for so many instruments of one kind or another. You have installed TRIMbase to help with the paperwork.

The data base consist of three permanent files:

1) A file of customers. Each customer is described by a record card as follows:

SERIAL NUMBER: [N10]	
NAME: [T30]
PHONE: [T13]	CREDIT RATING: [C5]
ADDRESS: [B30x3]
[]
[]
DISCOUNT RATE: [N5]		

discount, bought a stated item (or items) at a given price. The transaction file is used to produce monthly bills or invoices which are sent out to the customers. It also serves as a historical record of your company's business. The record card will be arranged like this:

SERIAL NUMBER:	[N10]	
NAME:	[T30]	
ADDRESS:	[B30x3]	
	[]	
	[]	
DISCOUNT RATE:	[N5]	
DATE:	[D11]	QUANTITY: [N5]
ITEM:	[T30]	
PRICE:	[N6]	

To start the system, begin by setting up the customer and item files. To cope with the regular flow of orders, you also define a "daybook" file, in which each order is recorded at the end of the day. Only the minimum information is stored.

SERIAL NUMBER:	[N10]	
DATE:	[D11]	
CUSTOMER:	[T30]	
ITEM:	[T5]	QUANTITY	[N4]

To save time, the item is entered as a code, such as FHBH for "French Horn Boosey and Hawkes". Similarly the customer's name could be a contraction, such as "Blastitt" for "Blastitt and Sons Ltd." The date and serial number can both be entered with single key strokes, and the total number of key depressions for each order will be about 20 to 30.

The 'daybook' file is updated every day. At the end of the week, it contains the bare details of all the week's business.

Now we come to the regular weekly updating of the full transaction file, and the production of invoices.

First, we join the customer file to the daybook file, using equivalence of customer names. To allow for contractions of names we specify that the name of the customer file must "include" the name in the daybook file.

We choose variables so that the intermediate file looks like:

```
SERIAL NUMBER: [ N10          ]
NAME: [ T30                      ]
ADDRESS: [ B30x3                ]
        [                        ]
        [                        ]
PRODUCT CODE: [ T5   ]
QUANTITY: [ N5 ]  DATE: [ D11      ]
DISCOUNT: [ N5 ]
```

Note this record has everything needed for the transaction file except the full name of the product and its price.

The next stage is to join the intermediate file to the product file, using the equivalence of product. The product code is discarded, and the full product description and its price are put in instead.

We now have a file in which contains complete transactions, but only for the previous week. We can call it "RECENT".

Next we merge this file with the main transaction file, to maintain correct records. The first week is exceptional in that there is no main file to merge with, so we can simply rename the week's file instead.

These manipulations involve a considerable number of commands, and of course it is vital that they should follow exactly the same pattern each week, otherwise the RECENT file will not merge with the existing transaction file.

This is where the sequence facility becomes really useful. The full set of commands is issued just once, and recorded; and after that the sequence is simply played back each week.

7.3 File management

This section considers the management of large, long-lived files and makes a number of recommendations.

7.3.1 Security

Data files are precious, and one should take every possible precaution against losing them. Modern computers are reliable, but errors and breakdowns still happen because of external causes such as power cuts, voltage surges due to lightning, excessive dust or smoke in the environment, and so on. **If you use a data base regularly, it is not a question of whether this will happen: merely of when.** If you are well-prepared, you can survive a computer breakdown or the loss of a file

with hardly any difficulty; but if you have not taken the right precautions the effects can be catastrophic.

The essence of file security lies in making frequent backups, or copies of your data base at particular dates. If the computer fails, you will only have to go back a short way to find a valid data base.

In some cases, a computer can damage a file it is reading as well as the one being written. It is therefore sensible to keep at least two separate backup disks, so that there is always one not mounted on the machine.

Similarly, if you have the slightest suspicion that a disk drive is not working correctly and damaging disks, don't load your precious backup into it until you have copied it on a different machine.

7.3.2 Large Files

In any data base put to practical use, the files grow at a steady rate. TRIMbase is designed to handle files of any size up to 32000 records, but it runs faster if it can hold the entire file in the Random Access Memory at any one time. To get the best results from the system, configure your machine so as to have as much space as possible — don't use a RAM disk, and don't fill the memory up with unneeded utilities.

When a file grows much beyond the size which can be held in the RAM (the system will tell you when this happens) it is worth while splitting it into two or more segments — such as Males and Females. The split can be made with the **Selection** command, and if need be the two files can always be merged again later.

APPENDIX 1: CONFIGURING TRIMbase

TRIMbase is designed for use in many different contexts. Thus you may choose to have dates expressed in either American or British style, you can specify a 'default' disk drive for your data files, and you can interface a printer of almost any type.

To let TRIMbase work in full harmony with its environment, you must run the configuration program, ENGLISH. In principle the program need only be executed once, but in practice most people will need to run it several times to get everything right.

The configuration program packs all the information about the environment into a short file called ENGLISH.LP. This file is constantly used by the main TRIMbase modules and must be present on the program disk, in the same folder or sub-directory as the program modules.

On the TRIMbase disk supplied by TALENT, you will find a ready-made version of ENGLISH.LP. which assumes that you have an EPSON-compatible printer, and that you prefer dates in the European style. If this is so, you don't need to run the configuration program at all. Otherwise, load the TRIMbase disk (preferably your "working copy"), making sure that writing is permitted. Display the directory and run program ENGLISH.

After the title screen, you will be asked to fill in a number of 'forms'. All of them use the standard GEM conventions:

1. You will be asked which style of date you prefer. There are four options:

Day before month, two-figure years : eg, 31/08/86

Month before day, two-figure years : eg, 08/31/86

Day before month, four-figure years : eg, 31/08/1986

Month before day, four-figure years : eg, 08/31/1986.

Once you choose a style of date, it will remain in force for the whole of your data base application.

2. Next, the program asks for details about the printer. TRIMbase has been carefully designed to support a wide range of different printers, and to take full advantage of such facilities as compressed characters, underlining and bold printing. To find the answers to the various questions you will have to look carefully at the printer manual, unless your dealer supplies you with a set of "magic numbers", or one of the TRIMbase newsletters contains the details for a printer of the right type.

If you successfully interface a new model of printer, please write to TALENT with the details. We will publish them so that others can benefit from your experience.

On the other hand, if you have persistent difficulty, and have returned your registration card, try contacting TALENT by letter or phone. Please note that we do not guarantee to be able to interface every printer that was ever made!

The questions on the next panel are as follows:

a) Do you have a printer? **Yes** **No**

If you don't have a printer at all, select **No** and click on **Ready**. There will be no more questions. Otherwise select **Yes** and continue.

b) Is its port **Parallel** or **Serial**?

Your printer has a **Serial** port if it plugs in to the modem socket on the back of the computer; otherwise it has a **Parallel** port.

c) Does it support modes? **Yes** **No**

Some printers accept commands such as "from now on print all characters in bold type" or "now go back to ordinary type". The current style of printing is called a 'mode', and we say that a printer supports modes if it can accept and remember commands about how to print characters still to come. Modes generally refer to bold printing (on or off), underlining (on or off), and the size of characters (compressed or normal).

Printers which don't support modes have no memory: they can only accept commands to type individual characters, or to do special functions such as backspacing or starting a new line. In many cases these printers can still be used to type underlined characters, but each one has to be constructed by the sequence "print character", "backspace", "print underline". Similarly, bold typing is done by printing a line once, then backspacing to the beginning of the bold section, shifting the carriage a fraction to one side (about 1/5 of a millimetre) and typing the bold characters again. This makes the characters stand out because they are blacker and thicker than usual.

If your printer doesn't support modes for underlining and bold type, TRIMbase can produce the necessary sequences of characters automatically.

To decide whether your printer supports modes, look at your printer manual. Find the section with a title like "Control Codes", and look for an entry such as

"Turns on the underlined character printing mode".

If you find one, your printer supports modes.

If your printer allows for condensed characters, but not for bold typing or underlining, then for the purposes of this program it does **not** support modes.

In general, matrix printers support modes, but daisywheel printers do not. This is not a universal rule!

d) What does your printer expect at the end of a line?

CR alone LF alone CR and LF

CR stands for "Carriage Return", that is, moving the print and head back to the left.

LF stands for "Line Feed", that is, advancing the paper one line.

Most printers assume that LF implies CR, and return the carriage automatically. Some require the CR as well. We don't know of any printers that take CR alone.

If you don't know which of the three options to choose, start with CR and LF. Then if all your printing appears double-spaced, change to LF alone.

3. The next form asks you to supply certain numbers. These numbers are always entered in decimal — not Hex or Octal.

In printer manuals some of the numbers, like the width of the printer carriage, are normally specified in decimal. Character codes, which form the bulk of the data you have to supply, can be given in a variety of notations — decimal, hexadecimal, or as mnemonics such as SI or ESC. You will have to convert some of these numbers into decimal, and the tables at the end of this appendix will help you do so.

The form you actually get depends on whether your printer supports modes.

3a. Printer details (with modes)

This form is displayed with default values which are correct for EPSON-compatible printers. You may only have to change a few of them:

Normal printer width: Give the width of the line in characters.

Condensed printer width: If your printer allows for compressed characters, give the number of compressed characters which can fit on to one line. 132 is the maximum that TRIMbase can handle. If your printer has no compressed characters, put 0 (zero).

Sequence for condensed type: Fill in the decimal value of the character (or sequence of characters) which switches the printer into the 'condensed' mode. If one character is enough, put 0 for the second. If the printer does not support the condensed mode, put zeros in both positions. (Similar conventions apply for the other entries).

Sequence for normal type: Specify the character(s) which make the printer return to normal-sized type.

Sequence for starting bold type: Specify the sequence which starts bold type (sometimes called 'emphasised printing').

Sequence for ending bold type: Give the sequence which ends bold type.

Sequence for starting underlining: Give this sequence.

Sequence for ending underlining: Likewise.

Print code for the British currency symbol.

3b. Printer details (without modes)

Normal printer width: Give the width of the line in characters.

Extended printer width: If your printer allows for compressed characters, give the number of compressed characters which can fit on to one line. 132 is the most that TRIMbase can usefully handle. If your printer has no compressed characters, put 0 (zero).

Printer code for extended width: Fill in the decimal value of the character (or sequence of characters) which switches the printer into the 'condensed' mode. If one character is enough, put 0 for the second. If the printer does not support the condensed mode, put zeros in both positions. (Similar conventions apply for the other entries).

Some daisywheel printers have an adjustable pitch which allows for 10, 12 or 15 characters per inch. From the point of view of TRIMbase, these printers do not have compressed characters unless the change in pitch can be made electronically, by receipt of a code sequence from the computer.

Code for normal width: Specify the character(s) which make the printer return to normal-sized type, or zeros if this does not apply to your printer.

Code for backspace: If your printer accepts backspace, the code is usually 8. The second figure should be 0.

Code for Underline: This code is often 95. The second figure should be 0.

Code for "Move right one pixel": This is the code which makes the print head move sideways a fraction to make bold type. In practice 'move left' command will serve nearly as well, unless the first character on a line is to be emboldened; if the printer head is already at the left it can't move any further.

Code for the British currency symbol.

3c. Initialisation sequence

Next, the program asks for the printer initialisation sequence. This sequence is sent to the printer at the start of any new document, and should include at least a reset and the selection of the appropriate national character set. For more complex printers it can also incorporate

margin settings, colour selection and line spacing commands.

The initialisation sequence is given as a set of decimal numbers, terminated by one or more zeros.

When you have entered all this information, click on Ready. The program will construct a configuration file and record it on the program disk.

As you enter the figures, it is best to record them in writing.

Next, test your configuration file by doing some printing with TRIMbase.

If you get no printing at all, ensure that your printer is properly connected, and if it is a serial model, that the baud rate is set correctly.

Try it out by printing a file from the desk-top. If this doesn't work, find and correct the problem before trying TRIMbase.

If the printing looks wrong, then you will easily be able to identify the trouble: for example, characters may be crossed out instead of underlined, or lines may be double-spaced. Whatever the error, it will point clearly to one of the answers you gave to the configuration program. Run it again with a different set of answers!

To translate character codes into decimal:

*Some codes have ASCII names. Use the following translation table:

NUL	BEL	BS	HT	LF	VT	FF	CR	SO	SI	DC2	DC4	ESC
0	7	8	9	10	11	12	13	14	15	18	20	27

*Some codes will be given as single characters. They are:

Space	!	"	£	\$	%	&	'	()	*	+	,	.	/	
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{		}	~	
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

*Some codes are given in hexadecimal — two characters including digits and letters A to F.

In a hexadecimal number the first digit is worth:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240

and the second is worth:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

To get the decimal value of the code, add the values of the first and second digits. For example, the code for 'underline' may be given as '5F'. This is equivalent to '80+15' or 95.

APPENDIX 2

This Appendix consists of queries about TRIMbase and TALENT's replies.

QUESTIONS AND ANSWERS

Q. I used a number variable to record telephone numbers. To my dismay, numbers starting with 0 don't seem to be recorded correctly, and international numbers with long prefixes are also garbled. What should I do?

A. A 'telephone number' isn't really a number at all — just a series of digits. Use a text variable.

Q. I am setting up a file about my patients. I had devised a 47-digit serial number which includes coded information about each patient's date of birth, marital status, number of children, social group and so on. I have just realised that I may have considerable difficulty in picking out patients in any one group, since my code '03' which means 'Widowed' in one part of the serial number also means 'March' in another. Can TRIMbase cope with this difficulty?

A. Yes — provided you don't try to compress all your information into one variable but use a separate variable for each property. The serial number should be a number — nothing else — and ideally should not carry any information about the record.

Q. I would like to use two block variables arranged side-by-side — but the system won't let me. It insists that a block variable has to go right across the card.

A. This is true. If you are very short of space, try using Text variables which can be placed several to a line. You can also pack variables of other types to the left of a block variable.

Q. The card isn't big enough for my needs.

A. Does your record card include lots of repeated information? For example, if your file is a list of club members, does each record include full details of the club (as opposed to the member)? If so, consider moving the common information to a separate file, and using the relational operations whenever you need to relate a member with his or her club.

Q. I want to use TRIMbase, but I already have a data base on a different machine. Can I transfer the data without retyping it?

A. Probably — the **Import** facility is designed for just this purpose, and is described in PART 7 of the User Guide.

When you have transferred your data, please write to us with full details. We shall file your letter and send a copy to any other user who wants to carry out a similar operation.

Q. When I use **Display records** option in the MAINTAIN module, I keep on falling off the end of the file back to the main control point. This infuriates me.

A. This is a design feature of TRIMbase. It is so easy to get back in to your file that the automatic return is simpler than a panel which says "You have just looked at the last record. Do you want to return to the main control point?"

Q. I have a big data file which takes several minutes to load. How can I make this faster?

A. Try filing the records in the same order as they were entered. You can do this by sorting on the serial number before saving.

Q. I had just spent about five hours typing in new data when there was a flash of lightning, the lights flickered and my computer froze. When I reloaded TRIMbase, all my data was gone. Can I get my data back?

A. No. When you enter new data it goes into the RAM, and will be lost if the computer is switched off or loses power for any reason. We recommend that you save your file every hour or so to minimise the damage by accidents like the one you describe.

Q. In the Report Generator, can I have more than 26 expressions in a report?

A. No.

Q. How do I print address labels?

A. With TRIMbase, this is quite straightforward.

To begin, measure up your label stationery. Let's suppose that each label (including the gap on the right and below) is 36 character columns wide and 9 rows deep. Incidentally, this is the size of the labels we use at TALENT; there are 16 of them to a 72-line page, arranged in two columns of eight.

Next, design a report. Set up the report size to be two columns less than the label width, and one row less than label depth. This allows for the gaps which TRIMbase inserts when arranging reports on the printed page. In our example, the report size will be 34 columns by 8 rows.

When you've set the report size, fill it in with expressions which specify title, name, address and any other information you want on the label. Try a few samples to check that everything is working correctly.

Now load your label stationery into the printer, and line it up so that the paper is correctly placed for the first line of typing. Choose the **Print** option, and when the system asks you for the paper size, be sure that the number of lines you give is correct — or at least an exact multiple of the label size such as 63.

The system should now print the complete file of labels. If anything goes wrong (for example, if the paper gets snarled up) stop the run by pressing the Escape key and restart, giving the number of the report you want to start at. This, of course is somewhat easier if you include the 'Report number' as one of the items you print on the label.

Q. I sometimes get disks where the data files don't reload into TRIMbase. I get messages like "This file was not made with TRIMbase", or sometimes the system crashes completely. What should I do?

A. It is **very important** not to remove the disk with your data file when the file is 'open' i.e. being used. The GEM system cannot tell if a disk has been removed and another one put in. It is easy for information to be written to the wrong disk, causing corruption and other problems.

Q. Should I use a RAM disk?

A. If you have RAM disk software on your computer and **at least** one megabyte of RAM, it is sensible to use it for storing the various modules, DEFINE, MAINTAIN, REPORT and their resource files. It is not recommended to use the RAM disk for data files, since TRIMbase includes its own memory manager and makes full use of all the RAM available to it.

Q. How can I interface to other packages (such as 1st Word)?

A. The Report Generator will send information to a disk file instead of the printer. It is generally quite simple to format the information so that it is acceptable to other programs. It would be difficult for us to give you detailed instructions for all individual packages!

Q. Since I began entering records, I have decided on a new numbering system. How can I change the serial numbers of my records?

A. The short answer is, you can't. The serial number is attached to each record once and for all and serves as a permanent unique identifier. It can never be changed.

The slightly longer answer is that although you can't change the serial numbers within a file, you can effectively renumber the records by making a new file. Use the 'Select' operation and an expression which is bound to be true in all cases, like

DATE After (>) 01/01/01

Q. I am having great difficulty in producing 'Total' fields for some numerical data.

A. Are you sure you haven't entered the data as a 'text' field? You can't total text fields, even though the texts consists entirely of digits.

Q. Can I send a file to disk and then re-import it as a reconfigured set of records?

A. Yes — but you may need to do some editing in between. In particular, you should make sure that the records are in the right format, and that they are separated with the \ character. The first record should be preceded by a *.

APPENDIX 3: CONFIGURING YOUR MACHINE

This section discusses ways of setting up your computer so as to get the best performance from TRIMbase.

1. Program Modules

To work correctly, all the program modules, together with their resource files and the configuration file ENGLISH.LP should be in the same folder or sub-directory. The contents should be:

ATARI ST	PC
english.prg	english.app
english.rsc	english.rsc
english.lp	english.lp
define.prg	define.app
define.rsc	define.rsc
maintain.prg	maintain.app
maintain.rsc	maintain.rsc
report.prg	report.app
report.rsc	report.rsc

If you have an ATARI ST with a large amount of RAM (say 1Mb or more) you may decide to put the program folder into a RAM disk. This will speed up the time to transfer between modules. On a PC this is not really feasible.

2. Tuning TRIMbase

To get the best speed out of TRIMbase when it is handling large files, we recommend that you allow it as much random access memory as possible. Don't use a RAM disk for anything unless your files are small and you have a great deal of memory to spare.

If you own an AMSTRAD PC 1512, there are several steps you can take to increase the amount of available RAM.

1) If you run TRIMbase under GEM with DOS+ (which is the normal way of running GEM) you will find that the random access memory fills up with files of only a few hundred records. When a file grows beyond this size the system will begin to swap data between the RAM and the disk, with a loss of speed.

2) To gain an additional 35k of space, run TRIMbase under MSDOS rather than DOS+. To do this, load MSDOS from disk 1 and give the command 'gem'. Then follow the instructions. The various comments which appear on the screen at this stage are not relevant to TRIMbase.

3) To gain 70k of space over DOS+, and to simplify the loading of TRIMbase, you can make yourself a special 'gem' disk, as described at the end of the section. In broad terms, this process copies all the parts of the operating system actually needed by TRIMbase on to a single disk, and discards all the others.

The process of making the disk is quite complex, but you need only do it once. Alternatively, TALENT will supply any registered user with a ready-made 'gem' disk for a handling charge of £5.

4) To gain 128k of space, you can expand your machine to its full limit of 640k RAM. We recommend that this be done by your dealer.

Preparing a Disk to allow larger database files to be RAM-based

1. Boot the system with MS-DOS (Disk 1)
2. FORMAT a new disk (hereafter called "Disk 5") using /S switch.
3. Use the command:
XCOPY a:*.* B:/W/S
with Disk 2 (Gem Startup) as disk A: and Disk 5 as disk B:
changing disks as requested by the program.
4. Use the COPY command to copy from Disk 1 (MS-DOS) in drive A: to Disk 5 in (virtual) drive B: the following:
COPY COMMAND.COM B:
COPY MOUSE.COM B:
COPY KEYBUK.EXE B:
5. Use the COPY command to copy from the TRIMbase disk in drive A: to Disk 5 in (virtual) drive B: the following:
COPY CONFIG.SYS B:
COPY AUTOEXEC.BAT B:
6. Delete the following files from Disk 5:
GEM.BAT
GEM2.BAT
GEMSTART.BAT
GEMSYS\OUTPUT.*

7. Create a new directory on Disk 5 by obeying:
MD GEMDESK
8. Use the command:
XCOPY A:GEMDESK B:GEMDESK/W/S
with Disk 3 (Gem Startup) as disk A: and Disk 5 as disk B:
changing disks as requested by the program.
9. The new disk (Disk 5) you have just created should then be labelled as "MS-DOS Gem".
10. Re-boot from this disk (Ctrl+Alt+Del with Disk 5 loaded). When the Desktop appears, select the RAM disk icon (single click), Select from the "Options" menu the "Install" option. Then select "remove". Again select the Options menu and "Save Desktop". These actions remove the RAM disk which has not been installed on Disk 5.

The disk you have thus created should always be used to run TRIMbase. The loading process is simplified to:

1. Boot from Disk 5.
2. Load the TRIMbase disk and press **Esc**.
3. Select the program you wish to run and double-click it.

For The Technically Minded

This section explains some of the mechanisms used in TRIMbase, and shows how the system takes full advantage of modern methods of programming and data organisation. A key feature of the design philosophy behind TRIMbase is that sophisticated features and machine dependencies are kept hidden from the user, so that the interface presented remains the same irrespective of the volume of data actually stored.

1. Strings and the Heap

Many database systems use fixed-length storage areas for text strings. This tends to make inefficient use of memory, since short strings are padded out with unnecessary spaces.

To deal with this problem efficiently, TRIMbase uses a heap. All strings are kept in a common area, and use just as much store as they need. If a string is changed in any way, a new copy is made and the old one discarded. From time to time a 'garbage collection' process recovers all the space used by the old strings and returns it to the pool of available memory.

2. Use of the disk

Some data bases are purely RAM-based. This means that all the records in any file are held in the Random Access Memory. Selection, sorting and so on are very fast, but a RAM-based system is strictly limited in the number of records it can hold. As your data base grows it will suddenly hit this limit and you will have to reorganise everything.

Other data bases are disk-based. The data is held on the disk and brought down to the RAM for processing as necessary. If you have a high-capacity disk there will be plenty of room for records, but processing tends to be very slow. This can often be improved with various types of indexing schemes, but they place a demand on the unsophisticated user, who has to decide — often in advance — which fields are to be indexed.

TRIMbase uses the notion of a cache store. Most of the program is constructed on the assumption that all the records are present in the RAM. If the file is a small one (up to several thousand records), this is true, and TRIMbase behaves with the speed of a purely RAM-based system. If the file is too big to fit into the store in its entirety, then a selected group of records is kept immediately available, and the rest are referred to by an index which is produced automatically. The system tries to make sure that the records actually in the RAM are the ones the user is currently interested in: for example, they may be a selected group, or the ones which have just been input and are liable to correction. The choice of records to keep in the cache is varied according to circumstances. This means that in most cases the system

continues to perform nearly as well as it did when the file was wholly in the Random Access Memory. However, if the selection of records is wrong (as it is bound to be from time to time) or if the file is so large that only an insignificant fraction of the data can be kept in RAM, the performance of TRIMbase gradually deteriorates to that of a purely disk-based system. The transition between the RAM-based and disk-based regimes in TRIMbase is invisible to the user, apart from a gentle ebbing of speed.

3. Indexing

Many disk-based systems let the user specify 'index variables'. If a variable (such as "Town of residence") is specified as an index, this causes such a system to set up a RAM-based directory, so that the record of any one who lives in — say — Glasgow can be retrieved without the need to search through the entire file. With a large file this an undoubted advantage when single records or small groups have to be retrieved, but it is only available at a cost — the index has to be stored as a separate item, and above all the user has to understand the indexing scheme and decide which variables to use as indices. When the whole of a file has to be used (for example, when it is being sorted or printed) indices are of no particular use.

TRIMbase maintains its own index system, but this is hidden from the user. The index is closely connected with the cache store described in Section 2. Since the index is not kept permanently, there is a brief pause if a large file is being used and the method of selection is suddenly changed. This is the technical price which must be paid for hiding the indexing system from the user.

4. Sorting

Since it has to cope with files of all different sizes, TRIMbase uses no fewer than three different sorting algorithms.

a) If the data is entirely in the Random Access memory, the system uses a classical Shell Sort. This is fast. A typical speed on the ATARI ST, with its 68000 microprocessor, is 43 seconds for 4000 records. A PC based on an 8086 is about four times lower.

b) If there is too much data to fit into the RAM, the system first tries to construct an index which contains the sort keys from all the records. It will always succeed for every type of variable except text or block, and usually succeed in these cases also. When the index has been made it can be sorted. Total time is 2-4 times as long as with a pure RAM-based sort.

c) If there is not enough space to construct the index, the system falls back, as a last resort, to a pure disk-based sort. This will be slow, but it will not be a common occurrence.

5. Relational Operations

TRIMbase provides a full set of relational operations. These are essential for many data-base applications such as matching files of customers and goods available; but they also provide good opportunities for saving space and simplifying file maintenance. In many cases, data base packages only allow for 'flat files', where every fact which may be relevant to an individual must be included in that individual's record. TRIMbase allows much of this data to be factored out, and stored in separate files. The factored data can easily be reconnected to the individual record by joining. With proper use of the relational features, records do not need to be as large as they sometimes are in other systems.

TRIMBASE

A REFERENCE MANUAL

This reference manual takes a direct approach and is intended mainly for the reader with some prior knowledge of data bases. Note that apart from some explicit references, the manual does not use the technical terms associated with relational data bases. We speak of records, not tuples; we do have relations, but not with the same meaning.

Following the normal conventions for reference manuals, we do not give much space to illustrations and examples. Where an example would help, we give a reference to the main manual, like this: (See 5.16).

1. OVERALL STRUCTURE

The overall structure of a data base is shown in Figure 1. It is a tree with several levels:

1.1. The Data Base

The uppermost node represents the whole data base. It is a collection of files. In TRIMbase the data base as a whole is an abstract entity. There is no object (such as a dedicated directory) which represents the data base as a whole.

1.2 Files

The next level down is occupied by individual files. Each file has a generic name chosen by the user, and the suffix ".TAL". Files are permanently stored on magnetic disks. They can be copied, deleted or renamed by the standard Operating System utilities, and for this reason their names must be compatible with Operating System conventions.

A file consists of two main parts: a schema, or description of the information held, and a set of records, each of which holds data items about a particular entity (person, transaction, etc). Each record has a unique and unalterable serial number within its file. A file may consist of a schema with no records at all. It will be in this state when it has just been defined but no data has yet been supplied.

The standard literature on relational data bases refers to a TRIMbase file as a "relation".

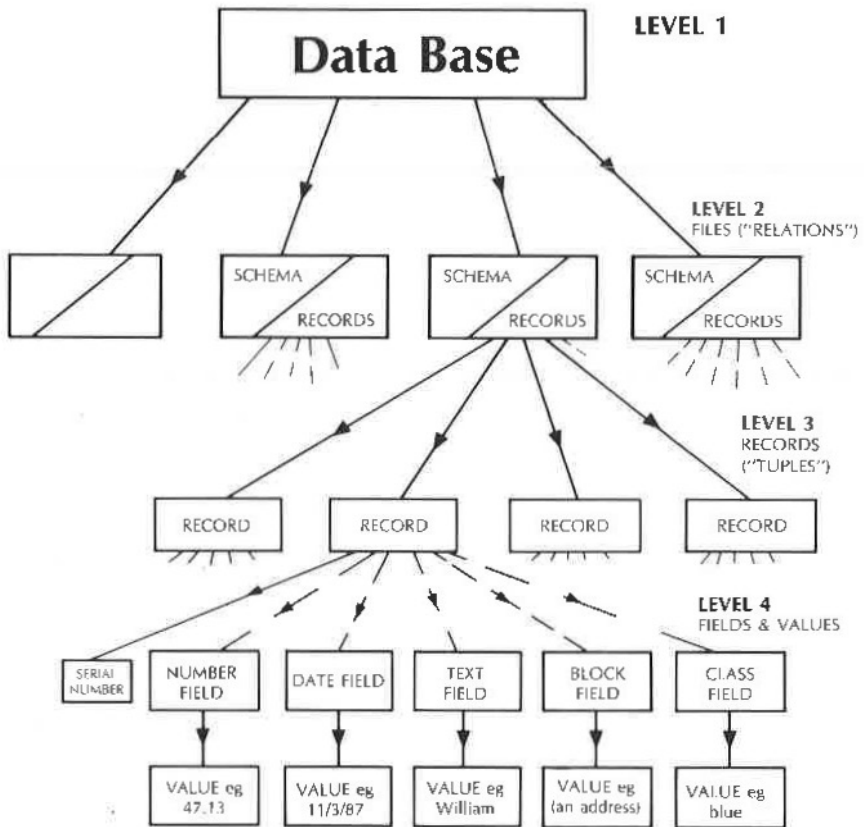
1.3. Records

The next level down consists of individual records. A record is represented by the metaphor of a physical card. The record card can be any size up to a full screen. The bottom line is always reserved for 'footnotes'.

All the records in any one file have precisely the same format, which is determined by the schema. The information is held in a number of fields, each with a specific name, type and position in the record.

Note that different files within the data base can use widely varying formats.

FIGURE 1



The name of a field must be unique within a file. It is constructed from any number of upper case letters, decimal digits and the symbols - _ / . and space. The first character must be a letter.

The serial number of each record is held in a field called SERIAL__NUMBER. This field is fixed at the top of the record.

The relational name for a record is a "tuple".

1.4. Types of field

The final level consists of individual fields with values. Where appropriate, a field is sometimes called a "variable". There are five types of field:

1.4.1. Numbers.

A number appears in the standard decimal notation, with a decimal point if it is not an integer. The internal representation of a number is as a real variable, which gives a precision of about 8 decimal places and a range between 10^{36} and 10^{-36} (approximately).

1.4.2. Dates.

A date can appear in one of four formats, as follows:

dd/mm/yy	(Day first, two-figure year)
mm/dd/yy	(Month first, two-figure year)
dd/mm/yyyy	(Day first, four-figure year)
mm/dd/yyyy	(Month first, four-figure year)

It is normal to keep the same format through the entire data base. The format is selected by the configuration program (q.v.).

When a date is entered on the keyboard, a useful shorthand for today's date is '#'.

1.4.3. Text fields

A text field is a short segment of arbitrary text. It must not be more than one line long, and may contain any sequence of visible characters which can be typed on the keyboard. Text fields are typically used for names, codes and other short items of information.

1.4.4. Blocks

A block is a segment of text occupying several lines. Unlike other variable fields, blocks are constrained to abut the right-hand edge of the card, but they need not occupy the full width.

Text entered into a block is automatically word-wrapped and justified on the left.

1.4.5. Class fields

A class field is one which is constrained to a small number of values which are known in advance, such as

Yes/No

or

Conservative/Liberal/SDP/Labour

The values are strings which may consist of any visible character except space and '/'. Upper and lower case may be treated as distinct or not, depending on the user's choice.

The limit to the number of possible values is set by the 'defining string', which consists of all the possible values separated by '/'. The maximum length of this string is 255 characters, or about 100 distinct values. The maximum length of an individual value is set by the position of the field in the record: no value may cross the right-hand edge of the card.

Every class value has an internal representation which is an integer in the sequence 0, 1, 2, These integers are allocated to the possible values in left-to-right order, so that in our examples both 'No' and 'Liberal' would have the internal representation '1'.

1.4.6. Unknown values

Every field in a record except the serial number may have the value 'unknown'. This corresponds to the field being left blank.

1.4.7. Special values

Numbers, dates and class fields can have the special value designated by '!'. This implies that the value is so unusual that it cannot be fitted into the existing framework. A card with a special value may have a footnote by way of explanation.

To give an example, consider a personal file where the title is a class field with possible values:

Mr/Mrs/Miss/Ms/Dr/Rev/Prof/Capt/Major/Col

This can handle the vast majority of people, but peers, popes and presidents would need special values.

2. FILE HANDLING

In general, data base systems are of two varieties: RAM-based and disk-based.

A RAM-based system starts by copying its data into the RAM of the computer. Sorting and searching are very fast, but there is an insuperable limit on the amount of data which can be held.

A disk-based system, on the other hand, keeps its data on the disk and only fetches those records in which there is an immediate interest. This allows for a much higher capacity, but inevitably brings slowness in operation.

TRIMbase is an adaptive system. At the beginning of each session some particular files are named and the system starts by fetching into RAM as many records as it can. If the whole file can be fitted, TRIMbase behaves just like a RAM-based system. If the file is too large, TRIMbase pages records in and out as necessary. This slows matters down but does not prevent correct operation.

To sum up, TRIMbase is fast on files which fit into the RAM, and correct on those which don't.

When planning a new file, it is worth remembering the amount of storage used by each type of variable. It is:

Serial number	:	6 bytes
Numbers	:	6 bytes
Dates	:	4 bytes
Text	:	6 + n bytes, where n is the number of characters
Blocks	:	6 + n bytes, where n is the number of characters
Class	:	1 byte.

Overhead (per record): 6 bytes.

Thus TRIMbase uses only as much store as is needed for a text or block value, and class values are highly economical of space.

3. EXPRESSIONS (See 1.5)

TRIMbase uses expressions in many places:

- ★ To select records of interest
- ★ To define conditions for 'joining' two files
- ★ To define values to be included in reports.

In this section we give a general view of expressions, and show how they can be built up from their primitive components.

3.1 Elements

The elements of an expression can include the following:

- ★ A field name (of any type).
This is always selected by pointing at the record card.
- ★ A constant, which can be a number, a date, a sequence of characters called a 'string' or a possible value for a class. Every constant is typed except for a possible value for a class, which is selected from a menu.

- ★ A relationship (such as 'equals' or 'before') which connects two values of the same type, and which can give any of the results 'TRUE', 'FALSE' or 'UNKNOWN'. For example, if a person weighs 85 Kg, the relationship

WEIGHT Less than (<) 60

is FALSE.

Again, if someone is named Brown, the relation

NAME Includes ([]) row

is TRUE, because the name includes the sequence "row".

Relationships are selected from menus.

- ★ An arithmetical operator such as + - * (times) and / (divide.) Operators are selected from menus.
- ★ Various connectives, such as & (and) | (or), → (implies) and 'otherwise'.

3.2 The condition

The most elementary grouping is the condition. This is a relationship between a named field and either another named field or a constant, both of suitable type. For any given record the relation may be TRUE, or FALSE, or UNKNOWN if either of the values is unknown.

The relationship must be chosen from one of the following groups:

To compare numbers:

Less than (<)
Greater than (>)
Less than or equal to (<=)
Greater than or equal to (>=)
Equals (=)
Not equal to (!=)

To compare dates:

Before (<)
After (>)
Not after (<=)
Not before (>=)
On (=)
Not on (!=)

To compare strings and blocks:

- Less than (<)
- Greater than (>)
- Less than or equal to (<=)
- Greater than or equal to (>=)
- Equals (=)
- Not equal to (!=)
- Includes ([])
- Does not include (![])

here 'less than' and 'greater than' refer to alphabetical order.

The condition

xxx Includes ([]) yyy

is true if the sequence yyy occurs **anywhere** within xxx.

When two sequences of characters are compared, upper and lower case letters may (or may not) be treated as equivalent. This is decided by the user, who can set the option either way.

To specify the null string (a constant with no characters) one may type the character 'home'.

To compare class values:

- Is (=)
- Is not (!=)

In most cases, a class field will be compared with one of its possible values, as in

SEX Is (=) female

When two class fields are being compared with each other, they are deemed equal if they both have the same internal representation (see Ref. 1.4.5). For example, consider the following class fields:

Name

Possible values

SPORT

Billiards/Cricket/Skiing/Boxing

RELIGION

Christian/Muslim/Jewish/Buddhist

The relationship

SPORT Is (=) RELIGION

would be true for the pairs of values (Billiards : Christian), (Cricket : Muslim) and so on; and false for all other pairs. Whether this result is sensible depends on the meanings of the values, which are (of course) unknown to TRIMbase.

3.3 The Relation

Note: This is not the relation of a relational data base!

A relation is made up of one or more conditions. If there is more than one, the group is connected either by the 'and' operator &, or the 'or' operator | (but only one type is allowed in any one condition). Examples of relations include

HEIGHT Less than (<) 1.65

COLOUR Is (=) green
| COLOUR Is (=) blue

NAME Includes ([]) bert
& NAME Does not include ([]) Albert

(satisfied by Bert, Egbert, Ethelbert, Bertha, but not Albert).

In general, relations can be evaluated over individual records and yield one of the three values TRUE, FALSE or UNKNOWN. The & and | operators combine values according to the following rules:

- "TRUE | x" (or "x | TRUE") yields TRUE for all values of x.
- "FALSE & x" (or "x & FALSE") yields FALSE for all values of x.
- "UNKNOWN &| UNKNOWN" gives UNKNOWN.

3.4 The Simple Expression

The simple expression is a variable, a constant, or an arithmetical expression made up of variables, number constants and the four operators + - * and /. The operators follow the normal rules of precedence (with left-to-right association) but no brackets are allowed.

The values in simple expressions may be qualified as follows:

Numbers and **numerical** expressions are followed by a number in brackets which indicates the number of decimal places to be used in displaying the value. Thus an expression such as PRICE (2) will yield a value such as 14.95.

Dates can be presented in any of eight formats, which include

Date in figures
Day number
Month number
Year number

Name of weekday
Name of month
Full date in words (e.g. 26th. February 1987)
Age in days (that is, the difference in days between the given date and today)

Blocks may be followed by an indicator which shows which line is required. For example,

ADDRESS (Line 4)

will, under appropriate conditions, produce a postcode.

Blocks may also be kept intact.

3.5 The Compound expression

A compound expression is a list of simple expressions, each one with a 'guard' condition. The system evaluates the conditions until it finds one which is TRUE; it then takes the corresponding simple expression. The last expression always has the guard 'otherwise' which is invariably TRUE.

An example of a compound expression is:

MARK Greater than (>) 80 → "Good"
MARK Greater than (>) 60 → "Fair"
Otherwise → "Bad"

This expression evaluates to one of the three strings "Good", "Fair" or "Bad", depending on the value of MARK.

4. THE MODULE STRUCTURE

In order to leave as much memory as possible for data, the functions in TRIMbase are allocated to three modules:

- ★ DEFINE, which is used to define and manipulate complete files.
- ★ MAINTAIN, used to insert, edit, delete, sort and select records within a given file.
- ★ REPORT, used to generate reports from a given file.

Movement between the modules is semi-automatic, and does not have to be effected through the GEM desk-top.

A diagram of the functions is shown in Figure 2.

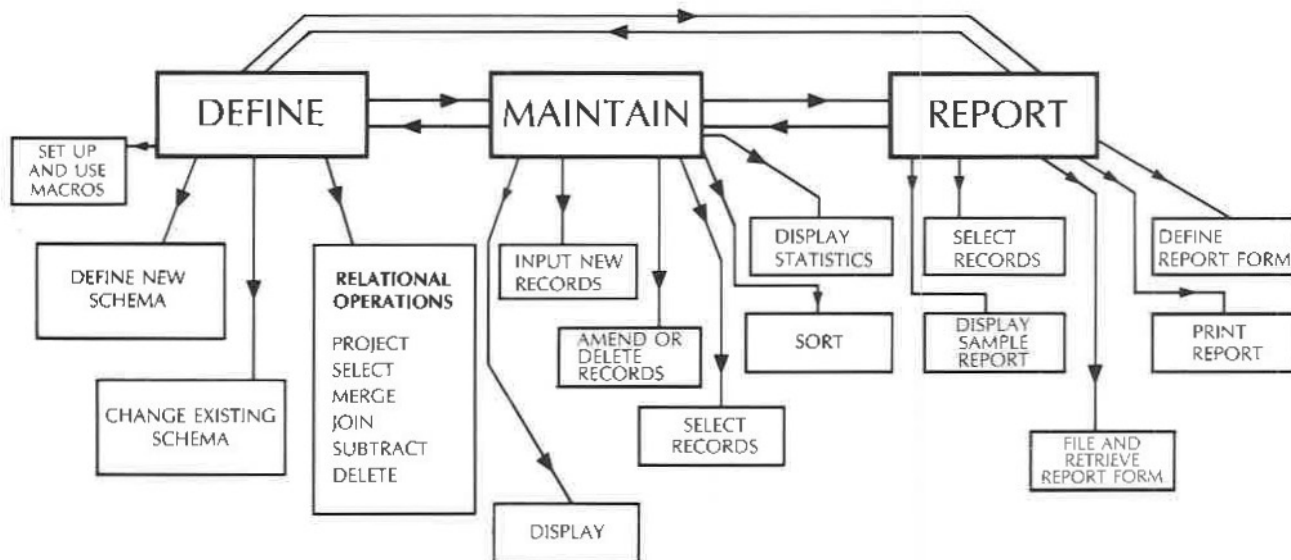
TRIMbase also includes a configuration program, which lets the user predetermine local characteristics of the system:

- ★ The Date format
- ★ Details of the printer to be used with the system. In foreign language versions of TRIMbase this includes codes for special national characters.

FIGURE 2

TRIMBASE

100



4.1 Loading TRIMbase

TRIMbase requires all its modules (DEFINE, MAINTAIN and REPORT) to be in the same folder or sub-directory, together with their respective resource files and the configuration file ENGLISH.LP. Any of the modules can be loaded from the GEM desk-top as a normal application program.

TRIMbase runs best if it is given as much random access memory as possible. Ways to provide more RAM are discussed in Appendix 3 of the main manual.

4.2 Control of Colour and Background

The general appearance of TRIMbase can be adjusted by the user. The function keys F1 to F5 are dedicated to this purpose.

On a machine with a colour monitor, the range of choice is as follows:

Background:	16 colours	selected by F1
	32 patterns	selected by F2
Record card:	16 colours	selected by F3
	8 'textures'	selected by F4
Characters:	16 colours	selected by F5

Each depression of one of the function keys causes the corresponding feature to 'cycle' to its next value. If the SHIFT key is held down, the features cycle in the reverse direction. There is an interlock which prevents characters from becoming invisible by being assigned the same colour as the record card.

On a system without a colour monitor, various shades of grey may be available.

When a background pattern and set of colours has been set up, it is automatically stored in the configuration file and used in every module. The choice can be changed at any time.

5. THE DEFINE MODULE

The DEFINE module has a number of functions, all related to the setting up and manipulation of complete files.

When the DEFINE module is loaded on an ATARI ST, it asks the user to supply the date or confirm that the date in the machine is correct. This step is omitted on PC-based versions, since they are generally fitted with real-time calendar clocks; but users are cautioned that the date must be correct or TRIMbase may produce incorrect or even absurd results.

The various options described below are accessible from the 'main control point' through a number of drop-down menus.

5.1 The Define card menu

This has two options:

- Make new card
- Change existing card

5.1.1. Make new card (See 2.1)

This option is used to define a new file, or record card. The process of definition takes a number of steps, which may be carried out in any order. The process starts with a small record card, blank except for a serial number field.

5.1.1.1. To adjust the size of the record card

Capture the bottom left-hand corner of the record card and drag it to any position on the screen. The card may not be larger than the screen, and it cannot be shrunk so as to exclude any existing variable fields.

(Here and elsewhere, the term 'capture' means "Point with the mouse and press the left-hand button". 'Drag' means "Move the mouse, keeping the left-hand button pressed").

5.1.1.2. To move the cursor

The cursor (a thin vertical bar) determines the initial position of the next field to be defined. The cursor can be moved

- by the RETURN and ENTER keys
- by any of the cursor control keys
- by pointing with the mouse and clicking.

It is impossible to place the cursor over an existing variable, or outside the record card.

5.1.1.3. To define a new field

- First, type the name of the field.
- Next, hit the RETURN key.
- Next, choose a type from the menu:

Number
Date
Text
Block
Class

There are three ways of making the choice:

- ★ Select a type with the mouse and click
- ★ Type the initial letter of the type, followed by RETURN
- ★ Step down the menu with the space bar, and hit RETURN

Note that the last two methods will not work if the mouse is inside the menu!

From this point, the defining sequence depends on the type chosen. For a **Number** variable, give the maximum length of the number, in decimal digits.

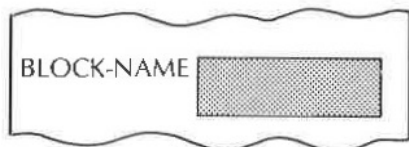
For a **Date** variable, no further action is needed. The format assigned is determined by the configuration program (q.v.).

For a **Text** variable, give the maximum length in characters.

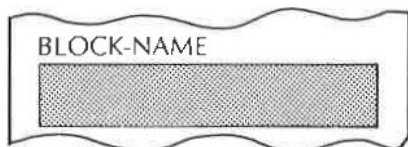
For a **Block** variable, give two items of data:

- a) The number of lines required
- b) Whether the data field is to start at the extreme left of the card.

The answer to b) gives rise to one of two different layouts:



or



Block fields always extend to the extreme right of the card, and are moved if the card size is adjusted. If the 'narrower' width is specified, other variables can be placed to the left of a block.

For a **Class** field, give the list of possible answers. The answers should be separated by / signs.

When a field is defined, the system displays square brackets [and] to delimit the data area.

The system will reject a field definition if the name is not unique, or there is not enough room on the card.

5.1.1.4. To move a field

Capture the field and drag it to its new position. Fields may not overlap, but may appear on the same line if there is room.

5.1.1.5. To adjust the size of a field

The length of a Number or Text field can be adjusted by capturing the closing square bracket and dragging it left or right. The number of lines in a block can be altered by capturing the lower left square bracket and dragging it up or down.

The size of a date field cannot be adjusted, and the size of a class field can only be altered by redefinition.

5.1.1.6. To discard a field

Capture the field and drag it to the garbage slot in the top right corner of the screen.

5.1.1.7. To get more information about a field:

Put the mouse over the field and double-click.

5.1.1.8. To finish defining a card:

Either: Press the **Esc** key to keep the definition

Or : Select the Disposal menu and choose "Accept" to retain the definition, or "Reject" to discard it.

If you keep the definition, the system displays the file selector panel and asks for a file name. The name should end with the suffix .TAL, but if you leave out these characters they will be inserted automatically.

If you choose the name of a file which already exists, the old file will be destroyed.

When a file is defined, a file is given an internal signature, or quasi-unique identifier. This is used later to ensure that report forms (see Ref. 7.3) are only used with matching files.

5.1.2. Change existing card (See Part 4)

This option allows you to change the definition of a card which already exists. This can be done even when the existing file contains records with data.

The process starts by asking you to name a file, and then displaying the record card. You may now change the card in any of the ways permitted when a new card is being defined:

- ★ Fields may be moved about
- ★ Fields can be changed in size (except data and class fields)
- ★ Fields may be deleted
- ★ New fields can be defined
- ★ Existing fields can have their definitions replaced, as follows:
 - (i) Double-click on the field to be changed, and answer 'Yes' to the question "Changes?"

- (ii) Define a new field. It need not have the same name, position, size or answer list as the field being replaced. However, the type menu is not presented, so that the new variable is forced to be of the same type as the one being replaced.

In general, a new field defined in this way inherits the data values associated with the old field. More details are given below.

When the changed definition is accepted, by pressing the **Esc** key or selecting "Accept", the system asks for a new file name. This must be different from the file being changed. The system then reads the data in the old file and writes a new file with the new specification. The values in each record are copied according to the following rules:

- ★ If a variable is left alone or merely moved, its values are copied without alteration.
- ★ If a variable is increased in size, data values are unchanged.
- ★ If a variable is decreased in size, text and block values are truncated if necessary. Numeric values, which are stored internally as real numbers, are unchanged and this may cause difficulty with display later.
- ★ If a variable is deleted, its values are not copied to the new file.
- ★ If a new variable is defined, its values are set to 'unknown'.
- ★ If a variable is redefined, the values of the old variable are inherited as follows:

Number and date values are copied without alteration.

Text and block values are copied with truncation if necessary.

Class values have their internal representations copied. (See Ref. 1.4.5). If the new variable has fewer values than the old, then old values which cannot be mapped on to the new range become 'unknown'.

When a card is redefined, its signature is left unaltered if the only changes are these:

The record card size is adjusted
Variables are moved
Variables have their sizes changed
New variables are defined

Any other changes cause the file to be assigned a new signature, which implies that report forms defined for the old record card cannot be used. (See Ref. 7.3).

5.2 The Relational menu (See 6)

The Relational menu includes the following options:

- Projection
- Selection
- Merge
- Join
- Subtraction

The operations in this group carry out the functions needed to maintain a relational data base. Readers not closely familiar with relational data bases are referred to Part 6 of the main tutorial, which contains detailed explanations and numerous examples.

In most cases (but with certain exceptions) the relational operations use one or two files as input, and generate a new file as output. These files are selected through the normal 'item selector' screen. In general, output files must have different names from the input files.

Other parameters of the operations, such as lists of variables, selector and join expressions, and reformatting information are determined through graphic interaction with the user.

Every relational operation is done on a disk-to-disk basis, so that the disks with the input files must not be removed from the drives during the operation.

5.2.1. Projection (See 6.1.1, 6.2.1)

This function uses a single input file and a list of variables selected from its record card. The system generates a new file with the copy of each record from the input file, but includes only the variables in the list, discarding the others.

The serial number is automatically included on the list.

The output file has a new schema, and a new record card layout which the system generates automatically. The user is given the opportunity to reformat this record card by altering the card size and moving the variables about.

Note that the results of a projection operation can also be achieved by redefining a record card (see Ref. 5.1.2).

5.2.2. Selection (See 6.1.2, 6.2.2)

This function uses a single input file and a selector relation. The system examines each record in the input file and copies it to the output if the relation is satisfied. The output file is automatically assigned the same schema as the input file (and the same signature).

5.2.3 Merge (See 6.1.3, 6.2.3)

This function uses two files which must have identical schemas. The function enlarges the first file by appending the records in the second

file. The serial numbers of the appended records are changed so as to follow on serially from the highest-numbered record in the first file.

Note that Merge is the only relational function which changes one of its input files.

5.2.4 Join (See 6.1.4, 6.2.4)

In strict relational terms, this is a combined 'join and project' operation, although it becomes a simple inner join if all the variables in both files are retained.

The function uses two input files and a 'join expression' involving variables from both files.

A 'join expression', like other relations, consists of one or more conditions linked by the 'and' or 'or' operators. However, each condition consists of three elements;

- a variable from the first file
- a relationship of suitable type
- a variable from the second file

Variables from the second file are identified by having a '**' prefixed to their names.

To set up a join operation, the system takes you through the following steps:

- 1) It asks you to select the two input files.
- 2) It asks for the join expression
- 3) It asks which variables from each of the two records are to be included in the output file. If all variables are chosen from both files, the operation becomes a simple inner join; otherwise it is a 'join-and-project'.
- 4) It generates a schema and card layout for the new file, and invites you to adjust it by moving fields and changing the size of the card.
- 5) It asks you to name an output file.
- 6) It calculates the join by evaluating the join expression over every possible pair of records taken from the two files. Where the expression is true, the system takes appropriate values from the two records and builds a new record for the output file.

The records in the output file have the serial numbers 1, 2, 3 ... irrespective of the serial numbers in the input files.

The order of the records in the output file is determined by the internal details of the join algorithm. It is worth knowing that when the second file is small enough to fit into the RAM, the process scans the whole of the first file for each record of the second file. This implies that the ordering of the second file is preserved.

If the second file does not fit into the RAM, the order of matching is more complex and no simple rule can be given for the ordering of the output. It should not be assumed that any ordering apparent on one occasion will always be present!

If a join expression is wrongly formulated, the resulting file can be very large. If it exceeds the capacity of the disk, the operation is terminated and a message is displayed.

5.2.5 Subtraction (See 6.1.5, 6.2.5)

This function takes two input files, which must have identical schemas. Every record in the second input file is matched against every record in the first input file, and duplicates are tagged. The output file consists only of the untagged records in the first file, that is, those records which are different from every record in the second input file. In this context two records are 'different' if any of their variables have different values, except for the serial numbers which are ignored.

The records in the output file are numbered from 1 up.

5.3. Text-mode

This gives the user two options in evaluating expressions such as those used in selection and joining.

5.3.1. A-Z, a-z same

In this mode, upper and lower case letters are treated as identical.

5.3.2. A-Z, a-z different

In this mode, upper and lower case letters are taken as different characters, with upper case coming before lower case in lexical order.

5.4. The Help Menu

This menu lets you display helpful information panels on many different topics.

5.5. The File menu

The options here include

- Start sequence
- Record sequence
- Playback sequence
- Go to MAINTAIN
- Go to REPORT
- Quit
- Delete file

5.5.1 Start sequence (See 7.2)

5.5.2. Record sequence

5.5.3. Playback sequence

In many applications of TRIMbase, each session of data input (such as a batch of experimental data) is followed by a standard sequence of relational operations. For instance, the file with the new batch can have abbreviations expanded by joining it with a file with full names, or the file can be merged with a cumulative 'master file'.

TRIMbase supports the technique sometimes called 'programming by example'. This allows for a sequence of operations to be recorded and 'played back' on future occasions, to perform exactly the same set of operations on a new set of data.

There are three sequence commands, all in the File menu.

The command "Start sequence" starts remembering a sequence of operations, so that they can be recorded in due course. The maximum length of a sequence is about 200 operations.

When the Record sequence command is given, the sequence being remembered is terminated, and stored under a file name supplied by the user. The file name normally ends with the suffix ".SEQ" but these characters can be omitted when choosing the file name.

The Playback sequence command invites the user to select a sequence, which is then played back. If any of the operations should fail, the sequence is aborted and control is returned to the user.

It should be noted that the sequence is stored not as high-level commands, but as a succession of key depressions, mouse movements, and so on. It follows that if a file involved in such a sequence has its schema modified, even in the trivial sense of having a field moved from one place to another, the sequence will no longer be valid and may fail or produce incorrect results. You have been warned!

5.5.4. Go to MAINTAIN

5.5.5. Go to REPORT

The user can transfer directly to the other modules in TRIMbase. This is done by selecting one of the File menu options

Go to MAINTAIN

or

Go to REPORT

When control is transferred, the system remembers the name of the file most recently defined, redefined or written, and this file is opened automatically by the new module.

5.5.6. Quit

This option terminates the DEFINE module and returns control to the GEM desk-top.

5.5.7. Deletion

This operation allows the user to delete a file of a given name. To save disk space it is often useful to delete intermediate results in a sequence of relational operations.

6. THE MAINTAIN MODULE (See 3)

The MAINTAIN module looks after operations on a single file — input, amendment, deletion, sorting, display and printing. The functions are divided into groups represented by drop-down menus.

When the module is entered from the GEM desk-top, the ATARI version takes steps to establish the correct date. This step is omitted from the PC version.

Next, the module uses the item selector screen to ask you for the name of the file to be processed. This file is then loaded into the RAM. If the file is too large to be fitted entirely into RAM, a message is displayed. In most respects, the behaviour of the system is exactly the same whether or not the file fits into RAM; but there are two significant differences:

a) If the file is not wholly held in RAM, many operations are significantly slower.

b) If the file is not completely in the Random Access Memory, the disk from which it is read must be left in position until every operation on the file is complete. Changing disks in mid-stream is a serious operator error which cannot be prevented by the operating system, and can cause widespread corruption of data.

If the file fits into RAM, the disk can generally be removed without ill effect. However this practice is not recommended, since it may lead to a habit which could prove disastrous on the first occasion that a growing file reaches the RAM size limit!

Once the file is loaded, the system offers a number of options which are accessed through several drop-down menus.

6.1. The Input menu (See 3.1)

The options are

- New data
- Import
- Help

6.1.1. New data

This option provides the normal route for typing new data on the keyboard.

To input a series of new records, the system displays a record card with variable names but no values. Field values are generally typed one by one, working across and down the card. A cursor is used to show the current input position. Input is governed by the following rules:

6.1.1.1. Deletion

If a character is typed incorrectly it can be erased by using either of the two delete keys. In normal circumstances, 'forward delete' removes the character to the right of the cursor, whilst 'backward delete' discards the character to the left of the cursor. Further characters to the right are all moved left by one place. If the cursor is at the extreme right-hand end of a field, then both delete keys remove the last character. If the cursor is at the start of the field, 'backward delete' has no effect.

The delete keys do not have uniform markings. They are inscribed as follows:

	ATARI ST	AMSTRAD PC1512
Forward delete	Delete	Del →
Backward delete	Backspace	← Del

On the AMSTRAD PC 1512 the 'forward delete' key will not work correctly until it has been assigned a value of 537F (hex) by the NVR program.

6.1.1.2. Termination of entries

Every entry is terminated by typing **Insert** or **Tab**. RETURN, ↑ and ↓ also terminate all entries except blocks. Within a block RETURN inserts a new line, and the up- and down-arrow keys move the cursor from one line to the next. Even in a block, RETURN and ↓ terminate the field if the cursor is on the last line, and ↑ ends it if the cursor is on the first line.

6.1.1.3. Entering serial numbers (See 3.1.1)

The serial number may either be typed explicitly, or it may be filled in automatically by hitting the RETURN or **Ins** keys. If an explicit serial number clashes with that of a record already in the system, it will be rejected.

6.1.1.4. Entering numeric values (See 3.1.2)

A numeric value is typed as a sequence of decimal digits, possibly preceded by a '+' or a '-' sign. There can be at most one decimal point.

If a value is unknown it is simply left blank.

If a value is 'special', this is indicated by typing a '!'. The system then accepts a footnote at the bottom of the card.

6.1.1.5. Entering dates (See 3.1.3)

A date is entered by typing the numbers of its day, month and year. The

separating '/'s are supplied automatically. Each component of the date must have the right number of digits. This implies that the day and the month always need two digits, one of which will often be a leading zero.

The order of the components, and the style of the year (2 or 4 digits) is determined by the configuration program (see Appendix 1).

An unknown date is simply left blank.

A 'special' date is marked by the symbol '!' and a footnote.

The character '#' can be used as shorthand for 'today's date'.

6.1.1.6. Entering text fields (See 3.1.4)

A text field is entered by typing the characters. Any visible characters may be used, including space. Trailing spaces are not recorded.

The left and right cursor keys can be used to edit text fields as appropriate.

Text fields do not have 'unknown' or 'special' values, although they may be completely blank.

6.1.1.7. Entering block fields (See 3.1.5)

A block field is entered by typing the characters in it. Any visible characters, plus space and RETURN, can be used.

If a block field is typed as a continuous string without RETURN's, it will be broken into lines so that, if possible, words are not split in two. This is the most appropriate mode for entries which represent running text, like descriptions or abstracts. On the other hand, entries can be broken into explicit lines by using RETURN. This is suitable for items like addresses.

All four cursor keys can be used to edit a block.

RETURN terminates a block if the cursor is currently on the last line.

A block can be terminated anywhere by typing **Ins** or **Tab**.

Block fields have no 'unknown' or 'special' values, though they may be left blank.

6.1.1.8. Entering class values (See 3.1.6)

A class value is entered by typing enough characters to identify it uniquely. Characters which do not help with identification, or which are not part of a possible answer, are simply ignored.

The details of recognition depend on the current text-mode (see Ref. 6.3). Thus upper and lower case letters may or may not be treated as equivalents.

An 'unknown' value is simply left blank.

A 'special' value is indicated by typing '!'. A footnote can then be added at the bottom of the card.

6.1.1.9. Moving from one field to the next

When a field is completed, subsequent events depend on the terminator used:

Ins, ↓ (where appropriate) and RETURN move the cursor to the next field. If the current field is the last, the system moves to the 'card completed' state.

↑ moves the cursor to the previous field, if there is one.

Tab moves directly to the 'card completed' state, ignoring any fields not yet visited.

6.1.1.10 Completing a card (See 3.1.7)

At the 'card completed' state, there are several options:

Backward delete (or 'backspace') moves the cursor back to the start of the card so that values can be changed. **Ins** will then skip through the variables without changing them until the one to be altered is reached.

Ins (when typed in the 'card completed' state) causes the record to be entered into the current data file. A new blank card is displayed for the next record.

Ins and SHIFT, if pressed together, cause the current record to be filed. The next record form, however, is not blank but starts as an exact copy of the previous one (except that its serial number is undefined). This feature is useful in entering several records with many identical fields, such as the members of the same family or a batch of orders from the same customer.

6.1.1.11. Ending the input phase

When a record has been accepted and a new record form has been displayed, the input phase can be ended either by hitting the **Esc** key or by making the appropriate menu selection. It is deliberately made impossible to quit the data entry phase in the middle of a card.

6.1.1.12. Help screens during data entry

At any point one of the 'help' screens can be displayed. The 'what next' screen is context sensitive, and will always tell you exactly what characters you can sensibly type at that instant.

6.1.1.13. Displaying details of a variable

If you double-click the mouse on a variable, the full details of that variable are displayed.

6.1.2. Import (See 7.1)

This option lets TRIMbase read data from a file generated by another program. The file must have a name with the suffix ".IMP".

In principle, the file must contain exactly the same characters as would be used to enter the records on the keyboard. Thus most fields consist of strings of characters ended by RETURN.

There are two exceptions to this rule:

a) The **Ins** character which terminates a record must be represented by the reverse slash (\).

b) The file must begin with a single '*'. All characters before the * are ignored. This allows the system to read files produced by word-processors which preface documents with invisible layout information.

Normally the file should end with an 'Escape' character (Hex 1B). In practice, files can be loaded without this character, as the system returns to the main control point when the end of the file is reached.

To read an external file, select the 'Import' option on the Input menu, and use the item selector panel to choose a file name.

Note that if the external file contains dates, then they must conform to the standard TRIMbase representation: two digits for the day or month, and either two or four for the year.

6.1.3. Help

This option displays an information panel about input.

6.2. The Display menu

The items in this menu include:

- All records
- Selected records
- Statistics
- Revise statistics
- Help
- Help (colours)

6.2.1. All records

6.2.1.1. Displaying records

This option is used to display all the records in the file. When the menu item is picked, the first record in the file appears on the screen, together with a 'slider bar' on the right. There is also a different menu bar, with the following items:

Control

- Next record
- Previous record
- Delete record
- Amend record
- Search and replace
- Global update

Help

- Next record
- Previous record
- Delete record
- Amend record
- Search and replace
- Global update

The other records can also be examined, and there are several methods of moving around the file:

To go one record down:

- Press the ↓ key
- or click on ↓
- or choose "Next record" in the Control menu.

To go one record up:

- Press the ↑ key
- or click on ↑
- or choose "Previous record" in the Control menu.

To go to any record in the file:

Capture the block in the slider bar and drag it up or down.

Note that this method of record selection does not conform to GEM conventions!

6.2.1.2. Deleting a record

A record being displayed can also be deleted, either by typing the 'Forward Delete' key or by choosing the Delete item from the menu. The system asks for confirmation before the record is removed.

6.2.1.3. Amending a record (See 3.4)

When a record is displayed, it can be 'opened for amendment' by selecting the Amend menu item in the Control menu, or by typing "A" (or "a"). The record can now be altered, using the same set of conventions as when a new record is being typed. The alteration is completed by typing **Ins** when the end of the record has been reached.

When a record is being amended, every keystroke is stored in a buffer for possible use by the Global update operation (see Ref. 6.2.1.5).

Note that the SHIFT and **Ins** option is not available when amending an existing card.

6.2.1.4. Search and replace (See 3.4)

When a record is displayed, this option can be invoked by selecting the Search and replace item in the control menu, or by typing "S" (or "s"). TRIMbase then displays a GEM dialogue box, and requests a string of characters. When this has been entered and the 'accept' button selected, the system scans the current record, searching its texts and blocks for an occurrence of the string. Searching is done in the same order as values are entered into the card; namely, left to right and downwards. As soon as an occurrence is found, the system stops with the cursor pointing at the first character in the string. The record may then be amended in the normal way.

The Search and replace operation is void if

- a) The 'cancel' button is selected in the GEM dialogue.
- b) No match is found in the current record.

Every keystroke in a Search and replace operation is stored in a buffer for possible use by a Global update (Ref. 6.2.1.5.).

6.2.1.5. Global update (See 3.4)

Global update is used to modify all the records in a file in some regular way. The principle of the operation is that the user finds and amends the first record to be altered. The keystrokes needed to carry out the amendment are stored automatically. Then the system takes over and applies exactly the same sequence of keystrokes to all the subsequent (selected) records in the file.

A Global update is started by selecting the Global update item in the Control menu (or by typing "G" or "g") immediately after a successful Amendment or Search and replace operation. The system then issues a suitable caution, and goes on to apply the change to all the records on display. The alterations can be observed as they take place, and the whole process can be aborted at any stage by hitting any key.

6.2.1.6. The Help menu

The items in this menu provide simple explanatory messages about the various options in the Control menu.

6.2.2. Selected records

The selection menu (See Ref. 6.4) is a way of identifying and marking those records which are currently of interest. For example, one might select all records which conform to a relation like

Hair is (=) red

The "Selected records" option in the display menu gives exactly the same facilities as "All records" (Ref. 6.2.1) except that only the selected records are displayed and made available for amendment and deletion.

6.2.3. Statistics (See 1.7)

This option can be used to display simple statistics on each variable in the record. Statistics can cover either the currently selected records or all the records in the file.

The statistics offered include the following:

- | | |
|------------------------|---|
| For numeric variables: | Number of known, unknown and special values, total of known values, highest and lowest value. |
|------------------------|---|

For dates:	Earliest and latest dates
For Text and Blocks	Alphabetically first and last values Length of shortest and longest values
For Class variables	Number of records with each possible value. If the list of possible class values is very long, only the first 15 values are delivered.

In every case, the number of unknown and special values is also given.

6.2.4. Revise statistics

As new entries are made in the file and old entries deleted or amended, the statistical information is normally kept up to date automatically so that the Statistics command can be executed without any delay. A difficulty arises when a record which contains an extreme value (such as the earliest date) is deleted. The new extreme value can only be found by scanning all the other records in the file, and this could be a time-consuming process. In practice, therefore, the extreme value is temporarily marked as 'unknown' and appears as a row of stars in the statistics panel. It can be recalculated at any time by calling the option "Revise statistics". This option does not, of itself, produce any output even though it is included in the Display menu.

6.2.5. Help

This option displays an information panel about the Display menu.

6.2.6. Help colours

This option displays a panel about colour changing.

6.3. Text-mode

This gives the user two options in evaluating expressions such as those used in selection and sorting.

6.3.1. A-Z, a-z same

In this mode, upper and lower case letters are treated as identical.

6.3.2. A-Z, a-z different

In this mode, upper and lower case letters are taken as different characters, with upper case coming before lower case in lexical order.

6.3.3. Help

This displays a panel about text-modes.

6.4. The Select menu

The options here include

- Select records
- Help

6.4.1. Select records

This option lets the user define a relation which picks out certain records of interest. It then searches the file and takes the following action:

If no interesting records are found, this is stated.

If exactly one interesting record is found, it is displayed.

If more than one interesting record is found, the system displays the number of records found. It then gives you the option of jumping directly into the "Display selected record" option (Ref. 6.2.2.) to examine, amend or delete the selected records.

When the relation is being evaluated, the system takes note of the current text-mode in handling upper and lower case letters (Ref. 6.3).

Once a relation has been set up for selection, it remains in force until a new one is set up.

6.4.2. Help

This supplies an information panel about selection.

6.5. The Sort menu (See 1.6)

The options here include

- Sort records
- Help

6.5.1. Sort

This option lets you sort the records into any order. Any field can be named as a key field, and sorting can be done into increasing or decreasing order. The comparison process takes account of the current text-mode setting.

The sorting algorithm ensures that when two records have identical sort key fields, their order is left unchanged. This implies that multiple key sorting can be done by a sequence of simple sorts, using the innermost key field first.

When the sort is complete, this system normally offers you the chance to purge the file of records which are exact duplicates (other than the serial number). The option is not offered if the most recent key field was a class variable, since the algorithm used could be very slow under these conditions.

The time taken for the sort is highly dependent on the size of the file, and on whether it is entirely held in the RAM. For the ATARI ST, a

typical sorting time for 4000 RAM-based records is 43 seconds. If the file as a whole cannot fit into the RAM, TRIMbase attempts to reorganise its internal storage, to fill the RAM only with the key values and pointers, and to sort on that basis. This approach will nearly always succeed, and will give typical sorting times of 3-5 minutes for 4000 records. If the attempt fails (as it might do if the key field is a block or long text) the system has to fall back on a disk-based sort which could take a long time. At this point the user is warned and given the option to escape. The moral is:

"If you need to sort a large file on a long text variable, think of it as a batch job."

6.5.2. Help

This option displays an information panel about sorting.

6.6. The Print menu

The options include

- Print all records
- Print selected records
- Print statistics
- Help

To use the printer successfully, it will normally be necessary to run the configuration program (ENGLISH) before the MAINTAIN module is used.

This is not necessary if the printer is EPSON-compatible.

The stationery page size is not included in the configuration program, since it may vary between runs. Instead, the user is required to state the page size when the printer is first used in any one run of the MAINTAIN module.

The page size is measured in lines. This need not be a whole number; for example, the correct height of A4 stationery is 71.5 lines.

Once the printer has been set up and the page size declared, TRIMbase keeps the paper in alignment automatically. However, if a print operation is aborted for any reason, the paper should be re-aligned by hand before the next operation is called.

When a printing operation is in progress, it can be stopped by hitting the **Esc** key. It will also stop automatically if the printer fails to respond. You then have the option of restarting if the problem has been fixed.

6.6.1. Print all records

This option prints out all the records in the current file. Each record is enclosed in a 'box' of characters. Several records may be printed on one page.

6.6.2. Print selected records

This option prints out the currently selected records, together with a statement of the selection formula.

6.6.3. Print statistics

This option works like "Display statistics", except that the tables are printed.

6.6.4. Help

This option displays a panel of information about printing.

6.7. The File Menu

The items in this menu include

- Save whole file
- Save whole file as
- Save selected records
- Quit this file
- Go to DEFINE
- Go to REPORT
- Delete a file
- Quit

6.7.1. Save whole file (See 3.2)

This option updates the current file by copying any changes from the Random Access Memory to the disk. If there have been few changes, the operation is fast; but on the other hand the process carries a certain risk, since a machine failure at the moment of writing may corrupt the whole file.

6.7.2. Save whole file as (See 3.2)

This option writes a copy of the file on to the disk, generally using a new name. The original file remains intact. The operation takes longer than "Save whole file" (Ref. 6.7.1.) because every record has to be written whether it has been altered or not; but it is inherently safe since the original file cannot be corrupted. Before writing, the system checks the amount of free space on the disk and aborts the operation if there is not enough.

6.7.3. Save selected records (See 3.2)

This allows the selected records to be stored as a new file. The original file is not disturbed unless the same name is chosen for the selected file; in this case the system gives a warning before proceeding. The free space on the disk is checked before the operation is allowed to proceed.

6.7.4. Quit this file

This allows the user to move to a different data file. If the previous file was altered and has not been saved, a warning message is issued.

6.7.5. Go to DEFINE

This calls up the DEFINE module. Transfer is direct, without going through the GEM Desktop.

6.7.5. Go to REPORT

This calls up the REPORT module. Transfer is direct, and on entry the REPORT module will automatically select the data file which was the last to be used in MAINTAIN.

6.7.7. Delete a file

This allows for any file to be deleted. The option should be used with care!

6.7.8. Quit

The system returns to the GEM desk-top.

7. The REPORT MODULE (See 5)

The REPORT module is used to construct and print reports based on individual files. It gives a wide range of freedom in formatting, and in combining information from different records.

When the REPORT module is entered from the desk-top, it goes through the usual dialogue and then invites the user to select a data file on which to base the reports. If the module is called from one of the other two modules, the name of the last file to be used is carried over and this file is opened automatically.

In essence, the module consists of a word processor with the extra facilities needed to substitute values and expressions from the data file. Text can appear in two guises:

a) Text can be displayed as a report form, that is, a general template for reports. A report form will generally contain symbolic expressions, which stand for items of information to be filled from the data file.

b) Text may be displayed as a genuine instance of a report. The various symbolic expressions are instantiated with items taken from the data file.

7.1. The Word Processor Mechanism

The word processor is a mechanism for building a report form. This form is constructed within a frame of selected size, which is limited to 132 columns and at most 230 rows. The maximum length of the frame depends on the amount of free RAM remaining when the program has been loaded, and will be less than 230 if memory is in short supply.

Once the frame has been set up, a cursor appears and the frame can be filled with text. The text is keyed in, with automatic word-wrap. The following features can be used:

7.1.1. Styles

The characters can be given one of three different styles: normal, bold and underlined. Styles cannot be assigned retrospectively. Once a character has been typed, nothing will alter its style except erasing it and typing a new one. The 'current style', which is assigned to all new characters, is set and changed by the Style menu.

7.1.2. Cursor Control

The four cursor control keys can move the cursor to any part of the frame.

The cursor can also be repositioned by pointing with the mouse and clicking.

7.1.2. Delete and Backspace

The delete keys can be used to delete forwards, and backwards, respectively.

7.1.3. Return

The RETURN key, if used, starts a new line. When the report form is being typed, newline characters are shown thus: ✓

7.1.4. Tilde (~)

The 'tilde' symbol is used as an 'anchor point'. In a report form the tilde appears as a diamond (◆), but is otherwise handled exactly the same way as any other character. When the report form is instantiated, the anchor points are used as 'tabs' to control the layout. This is explained in Ref. 7.2.

7.1.5. Control codes (See 5.9)

The text in a report form may include control codes for a printer. A control code is typed as a reverse slash followed by a decimal number in the range 0 to 255. When an instantiation of the report form is printed, the sequence is interpreted as a single character.

To put a reverse slash into a printed report, type two reversed slashes.

7.1.6. Sections of text (See 5.5)

A section of text can be identified with the mouse. Then:

- ★ An identified section may be
 - moved to a different part of the report
 - copied to a different part of the report
 - deleted
- ★ An identified section can be labelled according to the desired method of justification:
 - left
 - right
 - centre
 - both sides

The justification rule is implemented immediately. Characters inserted into lines which are already justified are interpreted correctly, even though this may cause the whole line to be rearranged. The previous line and following lines may also be altered.

★ An identified section of text may be labelled as belonging to:

- a Group area
- a Repeat area
- a Total area
- a Mean area

Text in Repeat, Total or Mean areas is marked with a distinctive background.

The meaning of these areas is discussed in Ref. 7.2.

7.1.7. Expressions (See 5.6)

Expressions may be inserted into the text. When the text is displayed as a report form, each expression is shown as a single inverted letter. The letters A,B,C ... are allocated automatically to successive expressions, imposing a limit of 26 different expressions in all.

Like other characters, expressions carry information such as the style, the justification rule and the membership of an area.

Expressions are specified with the expression menu. There are five types:

★ **Simple expressions** (Ref. 3.4)

★ **Compound expressions** (Ref. 3.5)

★ **Count expressions**, which provide a shorthand way of specifying compound expressions of the form

- condition → 1 (0)
- otherwise → 0 (0)

★ **Today's date.** The current date can be formatted in eight different ways, analogous to the ways in which a Date variable can be displayed.

★ **Page number.** This is an integer which is incremented by one for each report produced.

An expression, which appears as a single letter in a report form, can be 'opened' by double-clicking on it with the mouse. This causes it to be displayed in full.

7.1.8. Large reports

If the report form is larger than the screen, it can be scrolled about using the normal GEM conventions. There is also automatic scrolling to ensure that the cursor is kept in sight.

7.2 The Instantiation Mechanism

At any moment a symbolic report form can be instantiated. This means that the symbolic expressions are filled in with 'live' data, and the result is displayed on the screen or printed. Instantiation can lead to any number of distinct reports between one and the number of records in the data file.

When a symbolic report is submitted for instantiation, it undergoes the following (temporary) transformations:

1) The anchor points become real 'tabs'. Each one is marked with its current position (just before instantiation) and used to define the (fixed) boundary between two justification fields.

2) Spaces next to anchor points are discarded.

The instantiated report is now constructed, character by character. Each expression is replaced by a group of characters which represent its value. These groups inherit the style and justification rule specified for the expression.

From this point, the report is taken as a series of segments. Each segment consists of the group of characters which falls between two fixed boundaries such as anchor points or the sides of the page. Each segment is justified individually.

Since a segment may contain evaluated expressions, it could be too long to fit between its boundaries. In this case a warning is displayed. Then, if the segment contains any non-numeric characters, it is truncated from the right so as to fit between the boundaries; but if it appears to be a number it is converted into a row of stars. This helps to prevent the display of incorrect numerical values.

The matching of records with the report form depends on the way that expressions have been used and allocated to areas.

It is possible to set a 'record selection rule', or boolean expression which can be evaluated for every record in the file. Records for which the expression is true are called 'interesting records'. If no rule has been selected all records are interesting.

The arrangement of expressions inside the report form leads to five distinct cases:

a) The report form contains no expressions at all. In this case the system generates only one report, which is an exact copy of the report form except that anchor points and new lines do not appear as visible characters. Such a report arises if the REPORT module is being used simply as a word processor, with no display of data from the file.

b) The report contains only expressions which are in 'group' areas — that is, there are no expressions in repeat, total or mean areas. This leads to a separate report for each interesting record. In each report, the expressions are replaced by the corresponding values from the data record. The order of reports is the same as the order of the records in the data file. (See 5.1).

c) The report form contains only expressions which have numerical values, and which are in total or mean areas. This generates a single summary report which covers all the interesting records in the data file. Every expression in a total area results in the total of all the individual values in the records, whilst an expression in a mean area produces the average value of the expression. Non-numeric expressions in total and mean areas are treated as if they were in a group area. (See 5.14).

d) All the expressions in the report form are in repeat, total or mean areas. If the data file is sufficiently small, this leads to one report covering the whole file. All the material in the repeat area, including expressions and text, is repeated for each interesting record in the file, whilst variables in total and mean areas are displayed as file totals and averages. Where values are repeated, the order is the same as the order of records in the data file.

If the report produced by this process cannot fit into the predefined report form, then it is split into a number of sub-reports, and the totals and averages become page totals and averages, respectively. Another reason for splitting is that the maximum number of records which can be used in a single report is 64; if there are more records, they will be included in separate sub-reports. The limit of 64 does not apply to summary reports under section c).

e) The report form contains at least one expression in a group area and at least one expression in any of a repeat, total or mean area. (This is, of course, the general case).

This case is treated in much the same way as case d); values from consecutive interesting records are collected and put into the report. The additional condition is that all the records to be included in any one report (whether as individual values or as contributions towards total or averages) must be adjacent to one another in the data file, and they must have precisely the same values for all the group variables mentioned in the report form. (See 5.13).

7.3. Saving and restoring report forms (See 5.11)

Once a report form has been designed, it may be saved in a file with the suffix ".REP", and reloaded on a future occasion. Each report form is closely related to a particular schema, and if the schema is changed (other than in certain ways) the report form becomes invalid. This is managed by assigning to each new schema a 'signature', which is a check-sum evaluated over all the names, types and other details of the variables. When a report form is saved, the signature of the current data file is included together with the other details. When the report form is reloaded, the system checks that the signature for the data file is the same.

It is worth ensuring that existing report forms are not invalidated unnecessarily. When the DEFINE module is used to redefine an existing card, it does not change the signature if the only amendments consist of

- ★ Changes in card size
- ★ Changes in layout
- ★ Changes in variable names
- ★ Changes in lists of class answers names, provided that the total number of answers to any variable remains unchanged
- ★ The addition of new variables.

7.4. Menus in the REPORT module

7.4.1 The File menu

The options are:

- Clear report form
- Load report form
- Save report form
- Change report size
- File report
- Quit this file
- Go to DEFINE
- Go to MAINTAIN
- Help

7.4.1.1. Clear report form

This option clears all text from the report form and returns the cursor to the top left-hand corner.

7.4.1.2. Load report form (See 5.11)

This option lets you load a report from a disk, replacing the current report form (if any). The selection of the report form is done with the GEM Item Selector panel. The suffix ".REP" may be omitted, for it is supplied automatically.

The load operation is abandoned (with a suitable warning message) if the signature of the selected report fails to match that of the current data file.

7.4.1.3. Save report form (See 5.11)

This option allows the current report form to be saved to a disk. The name for the report is selected using the item selector panel. The suffix ".REP" is supplied automatically.

The save operation is aborted (with a warning message) if the selected disk does not have enough free space for the report.

7.4.1.4. Change report size (See 5.3)

This option allows the dimensions of the report form to be changed. The new size is keyed into a standard GEM dialogue box. This method is preferred to the more informal 'box-stretching' because

- a) The report size may exceed the size of the screen and
- b) It is often necessary to specify precise dimensions, such as '80 * 64'.

The report size can be changed even after text has been inserted. The operation is aborted if

- a) The requested width is more than 132 characters
- b) The requested length is more than the limit set by the available RAM
- c) The new size is so small that the existing text will not fit.

7.4.1.5. File report (See 5.15)

This option provides a method of sending a series of instantiated reports (NB not report forms) to a disk rather than to the printer. This is useful when TRIMbase data is exported to another program.

The overall operation of the option is similar to the Print command described in Ref. 7.4.6.2. The main difference is that when a file of reports is written to the disk, style information is discarded, leaving a simple file of ASCII characters. Control codes are converted to single ASCII characters.

A file of reports has a name with the suffix ".FIL". The name itself is chosen with the item selector panel, and the suffix can be omitted since it is supplied automatically.

7.4.1.6. Quit this file

This option lets you stop using the current data file and select another one instead. The current report form is cleared.

7.4.1.7. Go to DEFINE

7.4.1.8. Go to MAINTAIN

These two commands switch TRIMbase to either of the other two modules. The code for the new module should be available, in the same directory as the current module.

If control is switched to MAINTAIN, the name of the current data file is passed across, and MAINTAIN opens it automatically on starting.

7.4.1.8. Help

This displays an informative panel about the File menu.

7.4.2. The Record Choice file

The options are:

- Set selection rule
- Display rule
- Delete rule
- Help

7.4.2.1. Set Selection rule

This option lets you set up a selection expression to determine which records are to be included in the report. Any new selection will replace the existing one.

7.4.2.2. Display rule

This command displays the current selection rule (if any). **Esc** or a double click on the mouse returns control to the main control point.

7.4.2.3. Delete rule

This deletes the current selection rule, causing all records to be included in subsequent reports.

When a selection rule is being changed, there is no need to delete the old one before the new one is supplied.

7.4.2.4. Help

This option displays a panel about the Record Choice menu.

7.4.3. The Style menu

The selection is:

- Normal
- Bold
- Underline
- Help

This menu determines the style or typeface of characters used in the report. The style can be changed at will, and applies to expressions as well as normal text.

Note that spaces cannot be underlined.

The Help entry displays a small panel about styles.

7.4.4. The Text-mode menu

The options are:

- A-Z, a-z same
- A-Z, a-z different
- Help

As in other modules, the text-mode determines the way that expressions are evaluated.

For A-Z, a-z same, upper and lower case forms of letters are treated as identical.

For A-Z, a-z different, they are treated as distinct, and all capitals come before all lower case letters in alphabetical order.

The Help option displays an explanatory panel about text modes.

7.4.5. Expressions

This panel is used to put expressions into the current report form. The options are:

- Simple expression
- Compound expression
- Count
- Today's date
- Report page number
- Help

Whenever one of the first five options is chosen, TRIMbase enters a dialogue to let you specify the details of the expression. The code letter (in the range A-Z) is allocated automatically.

The Help option displays a lengthy panel about expressions.

7.4.6. Display

The display menu is used to instantiate report forms. The items are:

- Sample
- Print
- Help
- Help (colours)
- Help (control codes)

7.4.6.1. Sample (See 5.8)

The Sample option uses the current report form and data file to generate the first instantiated example of the reports which the report form can be expected to produce. If the report form is larger than the computer screen, it can be scrolled about using the GEM slider bars.

When the first report has been examined, the next one in sequence can be called up by giving the mouse a single click. This will be repeated until either the mouse is double-clicked, or all the records in the data file have been used up.

The sample option can be used repeatedly as a report form is constructed, and is extremely useful in getting the layout right. It is interesting to contrast this scheme with the more common system of exporting a data file to a word processor with 'mail merge' facilities. TRIMbase allows for immediate feedback, and permits rapid construction of well-designed reports.

An individual sample can be printed by typing 'P' (or 'p') when the sample is displayed. The first time the printer is called, the system will enter a dialogue to ask for the page size, and whether condensed printing is to be used.

When individual reports are printed in this manner, only one report appears on each page.

7.4.6.2. Print (See 5.9)

The Print option allows for all the reports generated from any one data file to be printed in one operation. Before calling the command, the user

is responsible for lining up the line printer paper so that the print head is at the top of a page.

When the print command is called, the system begins by asking for the printer page height, and whether condensed printing is to be used. It then requests a 'starting report' number. If this number is higher than 1, the appropriate number of reports is generated and discarded before printing starts. This feature allows for printing to be restarted in the middle of a run if — say — it was necessary to change the paper in the printer.

When printing starts, the reports are arranged so as to use paper as economically as possible. If the report form is narrow enough, then two or more reports are printed side by side, with a standard spacing of 2 characters. For example, if the page width is 80 characters and the report width is 18 characters, the reports are arranged in rows of four.

If the report form is short enough, two or more reports appear below one another on the same page, with a standard spacing of one line. If the report form is longer than a page, the instantiated report always occupies two or more complete pages.

When a page is full, the system automatically advances to the beginning of the next page. However there is no movement if the height of the report, plus 1, is a sub-multiple of the page size. This fact is useful when using pre-printed stationery or labels.

When printing has started, it can be stopped by pressing the Esc key. If the printer is switched off, TRIMbase will eventually time-out.

7.4.7.3. Help

This option presents a brief panel about the Display menu.

7.4.7.4. Help (colours)

This option displays a useful panel about changing the colour and the texture of the display.

7.4.7.5. Help (control codes)

This option displays a panel which states the facts of life about control codes.

8. THE CONFIGURATION PROGRAM (See App. 1)

The Configuration Program is used to generate a short file which contains details about the user's printer, his or her selection of colours and background patterns, and the date convention. On English-language versions of TRIMbase, the configuration program is called ENGLISH.APP or ENGLISH.PRG, and the configuration file is called ENGLISH.LP. This file must be present in the same directory as the other TRIMbase programs when any module of TRIMbase is run.

The configuration program consists of a series of dialogue screens fully described in Appendix 1.

Addendum: TRIMbase file types

The various types of file used by TRIMbase are distinguished by their suffices. They are:

Suffix	Purpose
.APP or .PRG	Program file
.RSC	Resource file
.LP	Configuration file
.TAL	TRIMbase data file
.IMP	File of data for import
.SEQ	Sequence file
.REP	Report form
.FIL	Instantiated report file

INDEX

- ! symbol, 1.4, 3.1.2, 3.1.3, 3.1.6
- # key, 3.1.3
- & 1.5
- * character, 7.1, Ref 6.1.2
- * prefix, Ref 5.2.4
- ***** 3.7
 - ← key, 3.14, Ref 6.1.1.6
- .TAL suffix, 2.2.2
 - key, 3.1.4, Ref 6.1.1.6
- A key, 3.4, Ref 6.2.1.3
- Abort print command, 5.9, Ref 6.6, Ref 7.4.6.2
- Accidents, 3.2
- Add a new variable, 4 Ref 5.1.4
- Add records to an existing file, 3.3
- All of Rule, 1.5
- Amend a record, 3.4, Ref 6.2.1.3
- AMSTRAD PC 1515, 1.2, 1.10.1, Ref 6.1.1.2
- Anchor point, 5.12, Ref 7.1.4, Ref 7.2
- AND, 1.5, Ref 3.1
- Any of Rule, 1.5
- Areas, 5.5, Ref 7.1.6
- Arithmetical operator, Ref 3.1, Ref 3.4
- ATARI ST, 1.2, 1.10.1, Ref 5, Ref 6, Ref 6.1.1.2
- Automatic serial number, 3.1.1., Ref 6.1.1.3
- Background, 1.10.1, Ref 4.2
- Backspace, 1.10.1, 3.1.7, Ref 6.1.1.10
- Backup, 1.1, 3.2, 7.3.1
- Backward delete, 1.10.1, Ref 6.1.1.1, Ref 6.1.1.10
- Block variable, 1.4, 3.1.5, Ref 1.4.4, Ref 5.1.1.3, Ref 6.1.1.7, Ref 6.2.3
- Blunders, 4
- Bold type, 5.2, 5.15
- Cancel, 5.5
- Card complete status, Ref 6.1.1.10
- Catastrophe, 7.3.1
- Changing disks, Ref 6
- Changing existing cards, 4, Ref 5.1.2
- Changing position of a variable, 2.2.2, 4, Ref 5.1.1.4
- Changing report size, 5.3, Ref 7.4.1.4
- Changing size of a record card, 2.2.2, 4
- Changing size of a variable, 2.2.2, 4, Ref 5.1.1.5
- Changing size of a record card, Ref 5.1.1.1
- Class value, Ref 1.4.5, Ref 6.1.1.8
- Class variable, 1.4, 3.1.6, 5.6.1, Ref 1.4.5, Ref 5.1.1.3, Ref 6.1.1.8, Ref 6.2.3
- Clearing report form, 5.4, Ref 7.4.1.1
- Colour, 1.10, Ref 4.2
- Completing a card, 3.1.7
- Completing a card definition, Ref 5.1.1.8
- Compound expression, 5.6.2, Ref 3.5
- Compound relations, 1.5
- Condition, 5.6.2, Ref 3.2
- Conditions of purchase, Preface
- Configuration file, Ref 4.2
- Configuration program, 2, 5.9, Appendix 1, Ref 6.6, Ref 8,
- Constants, Ref 3.1, Ref 3.4
- Control codes, Ref 7.1.5
- Control menu, 1.4
- Copy, 5.5
- Count expression, 5.6.3, 5.14, Ref 7.1.7
- Cursor, 2.2.1, 3.1.4, 5.4, 5.5., Ref 5.1.1.2, Ref 6.1.1, Ref 6.1.1.2, Ref 6.1.1.9, Ref 6.2.1.4
- Cursor control, Ref 7.1.2
- Cursor control keys, Ref 5.1.1.2
- Cursor keys, 2.2.1, 3.1, 3.1.5, 5.4, Ref 6.1.1.7
- Data Protection Act, Preface
- Date format, 2, Ref 1.4.2, Appendix 1
- Date variable, 1.4, 3.1.3, 5.6.1, Ref 1.4.2, Ref 5.1.1.3, Ref 6.1.1.5, Ref 6.2.3
- Decimal places, 5.6.1, Ref 3.4
- DEFINE, 2.1, 4, Ref 4, Ref 5
- Define card menu, Ref 5.1
- Defining a new file, 2.1
- Defining a new record card, 2.2
- Defining a new variable, Ref 5.1.1.3
- Defining variable fields, 2.2.1
- Delete keys, 1.10.1, 2.2.1, 3.1, 5.4
- Delete rule, 5.7
- Delete selection rule, Ref 7.4.2.3
- Deleting a file, 6.2.6, Ref 5.5.7, Ref 6.7.7
- Deleting a record, Ref 6.2.1.2
- Deleting a variable, 2.2.2, 4, Ref 5.1.1.6
- Deletion (of characters), Ref 6.1.1.1
- Display format of dates, 5.6.1, Ref 3.4
- Display menu, Ref 7.4.6
- Display rule, 5.7
- Display selection rule, Ref 7.4.2.2
- Displaying records, 1.3, Ref 6.2.1.1
- Displaying selected records, Ref 6.2.2
- Disposal menu, 2.2.2, 3.1.7, Ref 5.1.1.8
- Divided by, 5.6.1, Ref 3.1, Ref 3.4
- Down-arrow key, Ref 6.1.1.2

Editing a record card, 2.2.2
 Elements (of expression), Ref 3.1
 End formula, 5.6.1
 Ending the input phase, Ref 6.1.1.11
 FNTR key, Ref 5.1.1.2
 Escape key, 1.3, 1.10.1, 2.2.2, 3.1, 3.1.7, 5.8,
 5.9, Ref 5.1.1.8, Ref 5.1.2, Ref 6.1.1.11,
 Ref 6.1.2, Ref 6.6, Ref 7.4.6.2
 Exclamation mark, 1.4
 Expression, 5, 5.6, Ref 3, Ref 7.1.7, Ref 7.4.5.
 Expression code letter, 5.6, Ref 7.4.5
 Extreme values, 3.7, Ref 6.2.4
 F keys, 1.10, Ref 4.2
 FALSE, Ref 3.2
 Field terminator, Ref 6.1.1.9
 File amendment, 3.4
 File handling, Ref 2
 File management, 7.3
 File menu, 1.11
 File name, 2.2.2
 File report, Ref 7.4.1.5
 File signature, Ref 5.1.1.8, Ref 5.1.2, Ref 7.3
 Files, 1, Ref 1.2, Ref 2
 Filing reports, 5.15
 Footnote, 1.4, 3.1.2, 4, Ref 1.3, Ref 1.4.6,
 Ref 6.1.1.4, Ref 6.1.1.5, Ref 6.1.1.8
 Formulas, 5.6.1
 Forward delete, 1.10.1, Ref 6.1.1.1
 Garbage slot, Ref 5.1.1.6
 GEM, Preface, 2.1, 5.3
 Global update, 3.4, Ref 6.2.1.5
 Go to DEFINE, 3.8, 5.16, Ref 6.7.5, Ref 7.4.1.7
 Go to MAINTAIN, 2.2.2, 5.16, Ref 5.5.4,
 Ref 7.4.1.8
 Go to REPORT, 3.8, Ref 5.5.5, Ref 6.7.6
 Group area, Ref 7.2
 Group variable, 5.13
 Guard conditions, Ref 3.5
 Help, 1.3
 Help menu, 2.1
 Home key, Ref 3.2
 IBM PC, Ref 5, Ref 6
 IBM PC clone, Ref 5, Ref 6
 Import, Ref 6.1.2
 Import file, Ref 6.1.2
 Import of data, 7.1
 Information about a field, Ref 6.1.1.13
 Information about a variable, 2.2.2, 3.1,
 Ref 5.1.1.7
 Inheritance of values, 4, Ref 5.1.2
 Input menu, Ref 6.1
 Input of new data, 3
 Input of new data by keyboard, 3.1
 Insert key, 3.1.5, 3.1.7, 3.4, Ref 6.1.1.2,
 Ref 6.1.1.7, Ref 6.1.1.9, Ref 6.1.1.10,
 Ref 6.1.2
 Instantiation, Ref 7.2
 Interesting records, Ref 6.4.1, Ref 7.2
 Internal representation of class value, Ref 1.4.5,
 Ref 5.1.2
 Invalid date, 3.1.3
 Invalidation of report form, 5.11
 Item selector panel, 3.2
 Join, 6.1.4, 6.2.4, Ref 5.2.4
 Join expression, 6.2.4, Ref 5.2.4
 Justification, 5.5, 5.12, Ref 7.1.6, Ref 7.2
 Key field, Ref 6.5.1
 Labels, 5.17
 Large files, 7.3.2
 Large reports, 5.17, Ref 7.1.8
 Layout control, 5.12
 Layout of block, 5.6.1
 Layout of block variable, 2.2.1, Ref 5.1.1.3
 Layout of report 5.17 Ref 7
 Layout of reports on page, Ref 7.4.6.2
 Layout of reports on the page, 5.3
 Layout of variable, 2.2.1
 Left-arrow key, 3.1.4, Ref 6.1.1.6
 Line number in block, 5.6.1, Ref 3.4
 Line printer, Ref 7.4.6.2
 List of answers, Ref 5.1.1.3
 Load report form, Ref 7.4.1.2
 Loading a data file, 1.3
 Loading TRIMbase, 1.2, Ref 4.1
 Logical connectives, Ref 3.1
 Lower case, 1.9, Ref 3.2, Ref 5.3, Ref 6.3,
 Ref 7.4.4
 Main menu (DEFINE), 2.1
 Main menu (MAINTAIN), 1.3, 3.1
 MAINTAIN, 3, Ref 4, Ref 6
 Maintenance of existing data, 3
 Make new card, Ref 5.1.1
 Management of disk space, 3.2
 Mean area, 5.13, Ref 7.2
 Memory occupancy, Ref 2
 Merge, 6.1.3, 6.2.3, Ref 5.2.3
 Minus, 5.6.1, Ref 3.1, Ref 3.4
 Module structure, Ref 4
 Mouse, 1.3, 2.2, 5.5, Ref 5.1.1.1
 Move cursor, Ref 5.1.1.2
 Moving around a file, 1.4, Ref 6.2.1.1
 Multi-level sort, 1.6

NATIONS. TAL, 1.3
 Null string, Ref 3.2
 Number variable, 1.4, 3.1.2, Ref 1.4.1,
 Ref 5.1.1.3, Ref 6.1.1.4, Ref 6.2.3
 NVR Program, 1.10.1
 Opening an expression, Ref 7.1.7
 Operator errors, Ref 6
 OR, 1.5, Ref 3.1
 Order of sorting, Ref 6.5.1
 Otherwise, 5.6.2
 P key, 5.10, Ref 7.4.6.1
 Page number, 5.6.5, 5.9, Ref 7.1.7
 Page size, Ref 7.4.6.2
 Parallel printer, Appendix 1
 Passing values to next card, 3.1.7
 Play back sequence, 7.2, Ref 5.5.3
 Plus, 5.6.1, Ref 3.1, Ref 3.4
 Precision (internal) Ref 1.4.1
 Print all records, Ref 6.6.1
 Print selected records, Ref 6.6.2
 Print statistics, Ref 6.6.3
 Printer, 2
 Printer control codes, 5.9, Ref 7.1.5
 Printer details, 5.9
 Printer initialisation, Appendix 1
 Printing, 5.9, Ref 6.6, Ref 7.4.6.2
 Printing a sample, Ref 7.4.6.1
 Printing individual reports, 5.10
 Printing records, 1.8
 Printing statistics, 1.8
 Problems in merging, 6.2.3
 Programming by example, Ref 5.5.3
 Projection, 6.1.1, 6.2.1, Ref 5.2.1
 Quit, 1.11, 2.2.2, Ref 5.5.6, Ref 7.4.1.6
 RAM, Ref 4.1, Ref 6
 Reading new data, Ref 6.1.1
 Record choice, 5.7, Ref 7.4.2
 Record selection, 1.5, 5.7, Ref 7.4.2
 Record sequence, 7.2, Ref 5.5.2
 Records, Ref 1.2, Ref 1.3
 Redefining an existing variable, 4
 Registration, Preface
 Relation, Ref 6.2.2, Ref 6.4.1
 Relational operations, 6, Ref 5.2
 Relations, 1.5, Ref 3.2, Ref 5.2.2, Ref 5.2.4
 Relationships, Ref 3.1
 Removing duplicates, 3.5, Ref 6.5.1
 Repeat area, 5.13, Ref 7.2
 Replacing an existing variable, Ref 5.1.2
 REPORT, 5, Ref 4, Ref 7
 Report form, 5
 Report form, 5
 Report size, 5.3
 Reports on groups of records, 5.12
 Restore report form, 5.11, Ref 7.3
 RETURN key, 1.10.1, 2.2.1, 3.1, 3.1.1, 3.1.2,
 3.1.3, 3.1.5, 3.1.6, 3.1.7, 5.4, Ref 5.1.1.2,
 Ref 6.1.1.2, Ref 6.1.1.7, Ref 6.1.1.9,
 Ref 7.1.3
 Right-arrow key, 3.1.4, Ref 6.1.1.6
 Sample, 5.8, Ref 7.4.6.1
 Save report form, 5.11, Ref 7.3, Ref 7.4.1.3
 Save selected records, 3.6, Ref 6.7.3
 Save whole file, 3.2, Ref 6.7.1
 Save whole file as, 3.2, Ref 6.7.2
 Saving files, 3.2, Ref 6.7
 Schema, Ref 1.2
 Scroll bars, 5.3, 5.4
 Search and replace, 3.4, Ref 6.2.1.4
 Section of text, Ref 7.1.6
 Security, 7.3.1
 Select records, Ref 6.4
 Selection, 6.1.2, 6.2.2, Ref 5.2.2
 Selection expression 6.2.2
 Sequence, 7.2, Ref 5.5.3
 Serial number, 3.1, Ref 1.2, Ref 6.1.1.3
 Serial printer, Appendix 1
 Set selection rule, Ref 7.4.2.1
 Setting up new files, 2
 SHIFT and insert, 3.1.7, Ref 6.1.1.10,
 Ref 6.2.1.3
 Simple expression, 5.6.1, Ref 3.4
 Slider bars, Ref 6.2.1.1, Ref 7.4.6.1
 Sorting 1.6, 3.5, 5.13, Ref 6.5
 Sorting time, Ref 6.5.1
 Space, Ref 6.1.1.6, Ref 6.1.1.7
 Special value, 1.4, 3.1.2, 3.1.6, Ref 1.4.6
 Standard separation of reports, 5.3
 Start sequence, 7.2, Ref 5.5.1
 Starting report number, Ref 7.4.6.2
 Stationery page size, Ref 6.6
 Statistics, 1.7, 3.7, Ref 6.2.3
 Stretching the record card, 2.2
 Style menu, 5.4
 Sub-report, Ref 7.2
 Subtraction, 6.1.5, 6.2.5, Ref 5.2.5
 Summary over a complete file, 5.14
 Summary reports, 5, Ref 7.2
 Tab key, 1.10.1, 3.1, 3.1.7, 3.4, Ref 6.1.1.7,
 Ref 6.1.1.9
 Termination of entry, Ref 6.1.1.2

Text mode, 1.9, 3.1.6, Ref 5.3, Ref 6.1.1.8,
Ref 6.3, Ref 7.4.4
Text variable, 1.4, 3.1.4, Ref 1.4.3, Ref 5.1.1.3,
Ref 6.1.1.6, Ref 6.2.3
Tilde, 5.12, Ref 7.1.4
Times, 5.6.1, Ref 3.1, Ref 3.4
Today, 3.1.3, 5.6.4, Ref 1.4.2, Ref 5,
Ref 6.1.1.5, Ref 7.1.7
Total area, 5.13, Ref 7.2
Transpose, 5.5
Trial run, 5.8
TRIMbase file types, Ref 8
TRUE, Ref 3.2
Truncated segments, 5.12
Two-level sort, 1.6
Type of variable, Ref 1.4
Type selection, 2.2.1
Type styles, Ref 7.1, Ref 7.4.3
Typing text, 5.4
Underlined type, 5.2, 5.15
Unknown, Ref 3.2
Unknown value, 3.1.2, 3.1.3, 3.1.6, Ref 1.4.6
Up-arrow key, Ref 6.1.1.2
Upper case, 1.9, Ref 3.2, Ref 5.3, Ref 6.3,
Ref 7.4.4
Utilities, 2.1
Variable, 1.4, Ref 3.4
Variable name, 2.2.1, Ref 1.3, Ref 3.1
Variable types, 2.2.1
Word processor, Ref 7.1
Wordwrap, 5.4, Ref 1.4.4
\
\ character, 7.1, Ref 6.1.2

5

6

7