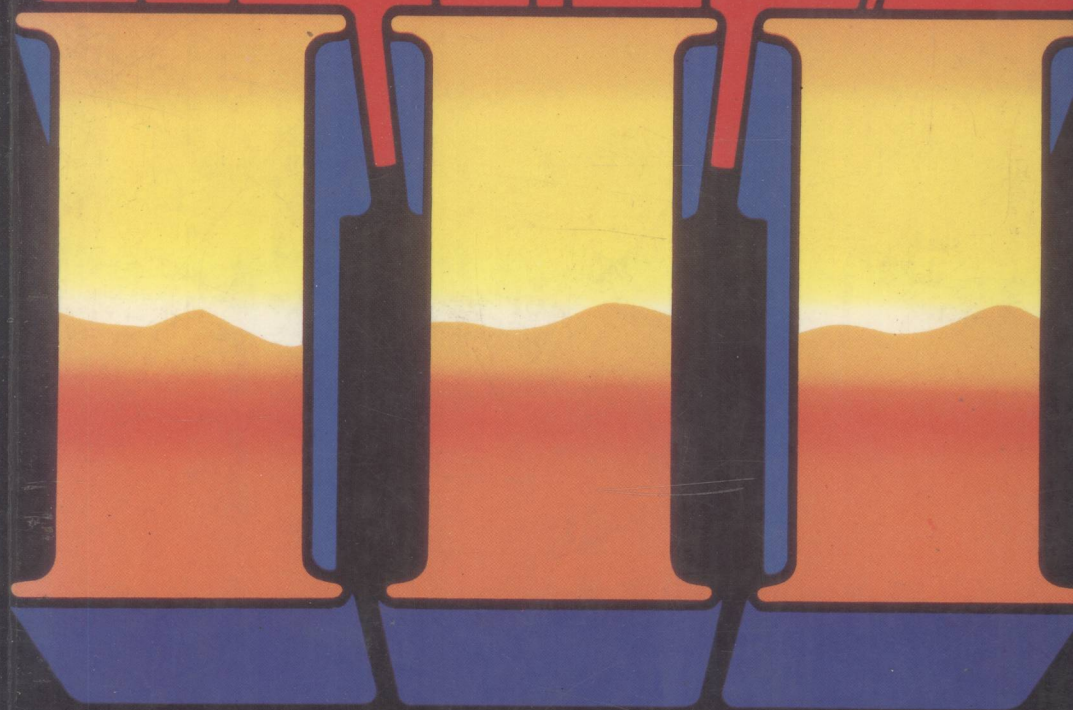


M.DE PACE

dBASE



A PRACTICAL
GUIDE

TP31
D278

8662472

dBase III

A Practical Guide

M. de Pace



E8662472



COLLINS

8 Grafton Street, London W1

Collins Professional and Technical Books
William Collins Sons & Co. Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Collins Professional and Technical Books 1985

Distributed in the United States of America
by Sheridan House, Inc.

Copyright © M. de Pace 1985

British Library Cataloguing in Publication Data

De Pace, M.

dBASE III: a practical guide.

1. dBASE III (Computer program)

I. Title

001.64'42 QA76.9.D3

ISBN 0-00-383076-4

Typeset by V & M Graphics Ltd, Aylesbury, Bucks
Printed and bound in Great Britain by
Mackays of Chatham, Kent

All rights reserved. No part of this publication may
be reproduced, stored in a retrieval system or transmitted,
in any form, or by any means, electronic, mechanical, photocopying,
recording or otherwise, without the prior permission of the
publishers.

Trademark acknowledgements, other than those given in the
text, are as follows: dBASE III is a registered trademark of
Ashton-Tate; WordStar is a registered trademark of Micropro
International; SuperCalc is a registered trademark of Sorcim
Corporation; 1-2-3 is a registered trademark of Lotus
Development Corporation.

DBASE III
A Practical Guide

Other books of interest

Working with dBase II

M. de Pace

O 246 12376 1

SuperCalc Prompt

Randall McMullan

O 00 383004 7

Introducing Macintosh

Francis Samish

O 246 12544 6

The Apricot Personal Computer

M. de Pace

O 00 383002 0

The Automated Office

Joseph St John Bate and Ross Burgess

O 00 383008 X

The IBM Personal Computer

M. de Pace

O 00 383003 9

WordStar Prompt

Randall McMullan

O 246 12446 6

Office Automation Using the IBM Personal Computer Systems

Ross Burgess and Joseph St John Bate

O 00 383104 3

Preface

The dBASE III database package was written to run on personal computers and it does so very successfully. However, it is powerful enough to provide serious competition to many mainframe computer systems, i.e. those running on computers that cost hundreds of thousands of pounds and require special environments with false floors and air conditioning. In my own company, dBASE III is used in conjunction with comparatively inexpensive hard disk personal computers to provide systems that otherwise would not have been possible not only for reasons of cost but also because there would not have been the time or resources to develop those systems on mainframe computers. With dBASE III, we are in the fortunate position of being able to tackle the processing and information needs of our users quickly and with the knowledge that the end product will be just as sophisticated as if we had taken the slower course of developing the system on one of the mainframe computers. It also means that my mainframe computer resources can be concentrated more successfully on achieving our corporate targets. It should be clear from the above that if dBASE III can make such a significant impression in a company already using mainframe computers, it must also be capable of tackling successfully the computer needs of those companies which are still considering the use of computers.

Like many earlier forms of technology such as the spinning jenny or the steamship, the computer has brought prosperity to those who have harnessed its powers for their own use. Not surprisingly, therefore, the demand for computer systems has so far outstripped the resources of those able to supply them. There are many software packages on the market but few provide the flexibility that allows taking on seriously the all round processing needs of even the smallest business. One of the most successful software packages to do so was dBASE II which sold several hundred thousand copies to eager computer users. Indeed, while it could be used immediately by those

who bought it, it was so flexible and so quick to use in developing applications that many software developers used it to produce specialised software for vertical market sectors such as farmers or estate agents. These features are also to be found in dBASE III, with the difference that any doubts that there may have been regarding the ability of dBASE II to support systems that grew in size, and the amount of use it had, were banished by the arrival of dBASE III.

The whole of my adult working life has been involved in designing or providing a computer service to people who need a computer to solve their business problems. The businesses have varied from airlines to wholesalers, from factories to insurance companies, and from a baker who simply wanted to recalculate his prices when the cost of ingredients changed to a firm that has to hold the vessel details of over 30% of the world's mercantile tonnage. In nearly twenty years of providing computer services I have never before been in such a happy and powerful position of knowing that I can provide immediate, affordable, and satisfying computer assistance to those who need computer support, from the sole trader or small business user to a department in a large company. It is this knowledge, and the fact that one is recommending a good product, that so endears dBASE III to me.

M. De Pace

Contents

1	Introduction	1
2	Creating and Using a Database	7
3	Listing Your Data	19
4	Working at the Screen	32
5	Changing Your Mind	49
6	More Than One Database	63
7	Writing dBASE III Programs	77
8	Programming Your Work	96
9	Advanced Programming	120
10	Functions and Techniques	142
11	dBASE II and Other External Programs	161
	<i>Appendix A: Creating a CONFIG.SYS File</i>	170
	<i>Appendix B: MODIFY COMMAND Function Keys</i>	172
	<i>Appendix C: Documentation</i>	174
	<i>Index</i>	177



Chapter One

Introduction

The importance of a database

A database package is essentially a piece of computer software that helps you to organise and use your information with as little effort on your part as possible. It is general purpose in the sense that it does not care whether your information consists of a list of customers, an inventory of stock items, or a collection of gramophone records. It will let you hold a large number of different bits of information about each item, and it will let you group, summarise, and sort your information in any way that you want. When you want to locate a specific item, it will find it for you almost immediately.

The term 'database package' is a quick way of saying 'database management system', a system that may be likened to a swan that appears to glide effortlessly over the water while beneath the surface it is paddling like fury. Although it gives the impression of simplicity itself, it is really much more complicated than it sounds. It is a complex program that performs the task of managing information contained in a database file. It provides the user with facilities for adding, amending and deleting data, producing information reports from more than one file and making enquiries that access more than one file. A database package also provides special file management facilities, such as merging database files and even amending the structure of a database file without any loss of data. Such a database manager contrasts with programming languages in that most of the basic functions such as creating a file, loading data, editing the data, and so forth are ready-made. There is thus an immediate and tremendous saving of development effort for the user who does not have to spend several weeks writing programs to provide these basic functions.

An important feature of a database manager is its flexibility, both in allowing the user to create a database file tailored to his own

needs, and in providing him with a trouble-free means of changing his mind about what he wants his database to contain. The same database package can be used by a marketing department to maintain records of prospective customers, and by a sales manager to keep track of the performance of his sales team. The domestic user will use it to maintain an index of his books or do his accounts while the shopkeeper will use it for stock control. In fact, anybody who has to maintain information which is voluminous, or changes quickly, or is complex, will benefit from a database package. It enables the user to control information and, just as important, to obtain quick and correct answers to questions.

Personal computer databases

There are a large number of software packages on the market which are listed under the heading *Data Management*. Some of these are powerful database managers, while others are simpler file maintenance and report-producing packages. Yet others are based on the card indexing method of storing and retrieving information, which does not include functions such as totalling a numeric item – for example, the cost of a stock item.

The most successful database package, in any category, has been Ashton-Tate's dBASE II. It is being used by hundreds of thousands of computer users. It is a complete database manager which has, in addition, its own programming language. Indeed it has proved to be such a successful basis for developing applications that a large number of software packages have been developed using the dBASE II programming language.

The success of dBASE II led Ashton-Tate to develop a new product using the same approach but taking advantage of the power and capacity of the new generation of personal computers, like the IBM PC, with 16-bit microprocessors able to run far larger programs than their 8-bit predecessors. The new product, dBASE III, needs 256K characters of memory but its specification is quite impressive. It allows up to a billion records in each database file, with a single record size of up to 4000 characters. It will let you use ten database files at once, using its relational features to obtain information from all ten.

It provides efficient multi-indexing on each database file so that an item of information can be looked up from more than one starting point – for example, Surname as well as Customer Code.

Apart from being more powerful than dBASE II, it also has several new features such as the ability to produce name and address labels, or the means to store word processing text in conjunction with database information.

To new users, who are not concerned with comparisons, it simply represents an extremely powerful computer tool that is easy to use and that offers the means of deriving substantial benefit from a personal computer without having to wait for expensive and time-consuming programs to be written.

Users of dBASE II

dBASE III treads ground similar to that of dBASE II and many existing users of the latter will be interested in learning about dBASE III. The packages are similar, yet there are enough differences to require a relearning exercise on the part of those who have experience of dBASE II. To assist such users the format of this book will follow along similar lines to the author's book on dBASE II. This will, hopefully, make it easy for the reader who has the earlier book to identify the differences. Unfortunately, it has not been possible to identify these differences in more than overall terms because there is much that is new in dBASE III. Nor will you find continual distinctions made between the two packages, because to do so would be distracting to those who are not familiar with dBASE II.

Users of the earlier package will, however, find something to interest them in Chapter 11 which discusses some of the important points to be observed in changing over from dBASE II to dBASE III. Included in that chapter is a description of a conversion program, dCONVERT, which is contained in the dBASE III package and which may be used to do much of the work of converting dBASE II programs, database files, and so forth, to run under dBASE III.

Preliminaries

Before you load dBASE III there is one bit of preparation that will come in very useful as you use more and more of the features of dBASE III. It is mainly a technical item that will not affect you during your initial use of the package (and it is the only technical

4 dBASE III

item to intrude at this early stage of explaining the use of the software). Unless you are aware of this point, however, you may be confused by a certain reluctance on the part of dBASE III to let you use as many database files as you would expect, or even to let you do certain things such as creating an extra printed report. These are unlikely events while you are using the package as described in Chapters 2 and 3 and you may, therefore, safely skip to the next section and return to this section at a later stage, but by explaining it here you will at least be aware of such a possible complication.

The operating system reads a file called CONFIG.SYS when it starts up and this file may be used to tell the operating system the maximum number of files that it should allow to be in use at any one time. Since one of the strengths of dBASE III is its ability to handle a number of different files all at once, we do not want the operating system to restrict this in any way so we will use the CONFIG.SYS file to tell it to allow up to 20 files in use at the same time. You can also use this method to set up a large number of buffers. These are the memory areas which are used to hold the data being read from or written to a disk file. The more buffers there are the less time will be wasted in reading or writing. It is recommended that you set the CONFIG.SYS file to contain the following:

```
files=20  
buffers=24
```

The method of creating the CONFIG.SYS file is described in Appendix A.

Getting started

dBASE III is loaded by entering DBASE against the DOS prompt. Since the software is protected against illegal copying, it is necessary to have the System Disk or the Backup Copy in the drive from which the program is being loaded. On a hard disk computer the software may be copied to the hard disk and loaded from there, but one of the two diskettes already mentioned must be inserted in the A: drive until dBASE III has loaded. Note that the diskette to be inserted must be the same one that was used to copy the software to the hard disk. If you copy the software from the Backup Copy and try to use the System Diskette when you load the program (or vice versa) a message claiming that you are using an 'unauthorised duplicate' will appear and the program will return you to the operating system prompt. If you have loaded the program correctly,

it will display a copyright screen which also contains the dBASE III version number and the date this version of the software was released. At the foot of the screen the dBASE prompt, a dot, will be waiting for your command.

dBASE III is command driven in the same way that DOS is. It has a large number of commands giving the user a very powerful and flexible tool with which to organise and reproduce information. A command processor needs a prompt that invites you to enter a command, and you are no doubt familiar with the DOS prompt that allows you to enter a command such as DIR to list the files on a diskette. dBASE III has a similar prompt, the dot, which allows you to enter one or more of its large number of keywords as a communication medium. As you will see shortly, you can go a long way by using no more than a dozen commands. dBASE III also has a special command for new users. This is the ASSIST command which calls up the dBASE III Assistant. It provides the means of using *menus* to carry out dBASE III operations, but it is also a learning tool since it explains the purpose and syntax of each command that is being used. First time users will find it possible to do much of their initial processing with the help of the ASSIST command, but will soon find that they are able to use dBASE III in its command mode quite naturally and without any difficulty as you will discover when you read on.

Returning to the dBASE III prompt, when you enter a command incorrectly, dBASE III will inform you by means of its error messages where you have not followed the rules. It will also offer you a HELP facility which explains the syntax of the command, provides a description of the command, and allows you to browse through like descriptions of associated commands. Examples of error messages include the following:

- File does not exist
- Unrecognised command
- Unrecognised phrase
- Syntax error

Certain error messages try to show you where you have made a mistake. For example, you will find out in Chapter 3 how to list your information by specifying the names of your various information items or fields. If you specify an item that does not exist in the database, dBASE III will not only tell you that it could not find the item, it will also place a question mark over the item in question. This is especially useful when you have specified a whole list of items.

Another feature of dBASE III is its Help menu which may be called up by pressing the Help function key. The Help menu contains options such as 'What is a ...' or 'How do I ...' to take you to Help screens that provide the appropriate explanations. It is possible to browse about from screen to screen so that the Help feature can also be used to provide an introduction to the software.

dBASE III commands come in many sizes. Some are short, such as `USE`, while others are long, such as `DISPLAY STRUCTURE`. Once you have become familiar with the commands, you may find it tedious to enter the longer commands in full. You may, therefore, prefer to use the abbreviated form which consists of the first four letters of each word – for example, `DISP STRU`.

As mentioned above, certain commands allow you to specify a list of items. If the list is a long one, it could run over a line. When this happens, you may signal to the program that there is more to come by ending the first line with a semicolon. You then simply continue the command on the line below. The second line may similarly run on if necessary. There is no fixed position for the semicolon; it is placed wherever the line finishes. It is wise, however, to allow a single space to the left of the semicolon because the next line will start exactly where the semicolon is positioned which could cause two separate words to be joined together.

With this minor preparation out of the way, you can proceed to use the program which will be far more interesting than reading about rules as you have just been doing. In the next chapter you will be shown how to create a database file, add information to it, change the information afterwards, and then list the details you have entered.

Chapter Two

Creating and Using a Database

Designing a database

Although dBASE III is capable of working with ten different database files at once, we are going to start with just one. To illustrate the use of dBASE III in its basic mode we are going to create a database file that will allow us to maintain a record of a fleet of vehicles belonging to a company called Swiss Windows. For the purposes of the illustration, it does not really matter what the business of Swiss Windows is but let us assume that they manufacture and install windows, and that apart from the directors' cars they also have a large fleet used by the sales team. The database will be a simple one purely to illustrate the first steps of using dBASE III, so do not be surprised if obvious details such as the vehicle registration details are not recorded.

The database will help us to keep a record of each vehicle, showing the make and model, its type (for example, saloon or estate), its value, and which member of staff is using the vehicle. The records will be used to list all the vehicles of a certain make – say, all the BMWs; to find out which vehicle is being used by a certain person; to find out whether there is a certain model in the fleet; to provide a full list of the entire fleet; to determine how much money has been invested in estate cars; and so forth.

The first task is to decide what information we want to store and the maximum size of each item of information. It looks like this:

Make	: 10	characters
Model	: 8	characters
Type	: 8	characters
User	: 17	characters
Value	: 8	characters
Total size	52	characters

Each item of information is known as a *field*, and the collective name for the fields representing a single entry, i.e. one vehicle in this illustration, is a *record*. You will see that the total size of the information for each vehicle is one character larger than the sum of the individual field sizes. This is because dBASE III needs the extra character to store information about the record. Except for the Value field, the information to be loaded into each field will be descriptive text such as the name of a make of car. This type of information is described as character data or text data. The Value field will be a numeric field which may be used in totalling or arithmetic operations.

The next thing to do is to consider whether your database file will fit on the diskette that will hold the file. We can estimate how many records we will need to store the details of all the vehicles and then multiply that number by the size of the individual record. In this case we have a fleet of 245 vehicles, so the total file size will be only around the 12K mark, i.e. 52 times 245 giving 12740. On top of that there will be a further 195 characters which are used to store the details of the database structure. The size of the structure is not fixed at 195: it is dependent on the number of fields in the database, but it is in any case a very small part of the database file and may be ignored for present purposes.

Our database file is so small that there is clearly no problem, but in another situation you may find that a database requires a much larger record and you will have to store thousands of records. For example, if you need to store 10000 records each 200 characters in length, you will have to consider using a hard disk giving 5 or 10 million characters of available space. It is important to have spare space available on a diskette to allow for additions to the file, or you might decide to add another item of information to the record. As you will discover later, there will also be a need for indexes and other files in addition to the main database file. Finally, the two main dBASE III software files (DBASE.EXE and DBASE.OVL) take up over 254K of space on a diskette. If you want to have the facilities provided by the Help files (HELP.DBS and ASSIST.HLP), a further 67K will be required.

Creating the database

As explained in the previous chapter, the dBASE III prompt consists

of a dot. Against this prompt we will now enter the command **CREATE**. We could have added the name of the file to be created but, since we did not, we will next be asked for the name of the file. Remember that DOS restricts filenames to 8 characters. We have decided to call our database file **MOTOR** so enter that. **dBASE III** will next clear the screen and display in tabular form an area for you to enter the name, type, and size of each field to be contained in a database record. We have already determined most of that information, so we can start entering immediately underneath the captions which are shown in lower-case in the illustration below. The screen section immediately underneath the captions will be highlighted to show you where to start. When you have entered a fieldname, you press the Return key and the cursor will jump to the next point on the line and wait for you to enter the fieldtype. The latter will already be set to **CHAR/TEXT** and unless you want to change this description, as you will on the Value field, you simply press the Return key again to move on to next point on the line which has the caption **Width**. This is where you enter the size of the field. There is another caption on the line, the **DEC** or number of decimals, but having described the field as **CHAR/TEXT**, the cursor will bypass this point and come to rest on the line below, ready for you to enter the details of the next field.

When you come to enter the fieldtype of the Value field, you will only need to enter the **N** of numeric and the **CHAR/TEXT** description will instantly change to **NUMERIC**. An alternative method of choosing a field type other than **CHAR/TEXT** consists of pressing the space bar to revolve the preset field type description until you see the type you want. After one depression of the space bar you will see the word **NUMERIC** appear in place of **CHAR/TEXT**. You then select it by pressing the Return key. Next you enter the **Width** of the numeric field, after which the cursor will wait underneath the **DEC** caption for you to enter the number of decimals required. As you enter the field details, you will see at the top of the screen a running account of the number of fields defined so far and the number of characters remaining out of the 4000 character maximum that may be used in a single record. When the five fields of our **MOTOR** database have all been entered, the screen will contain the following:

	field name	type	width	dec
1	MAKE	Char/text	10	
2	MODEL	Char/text	8	