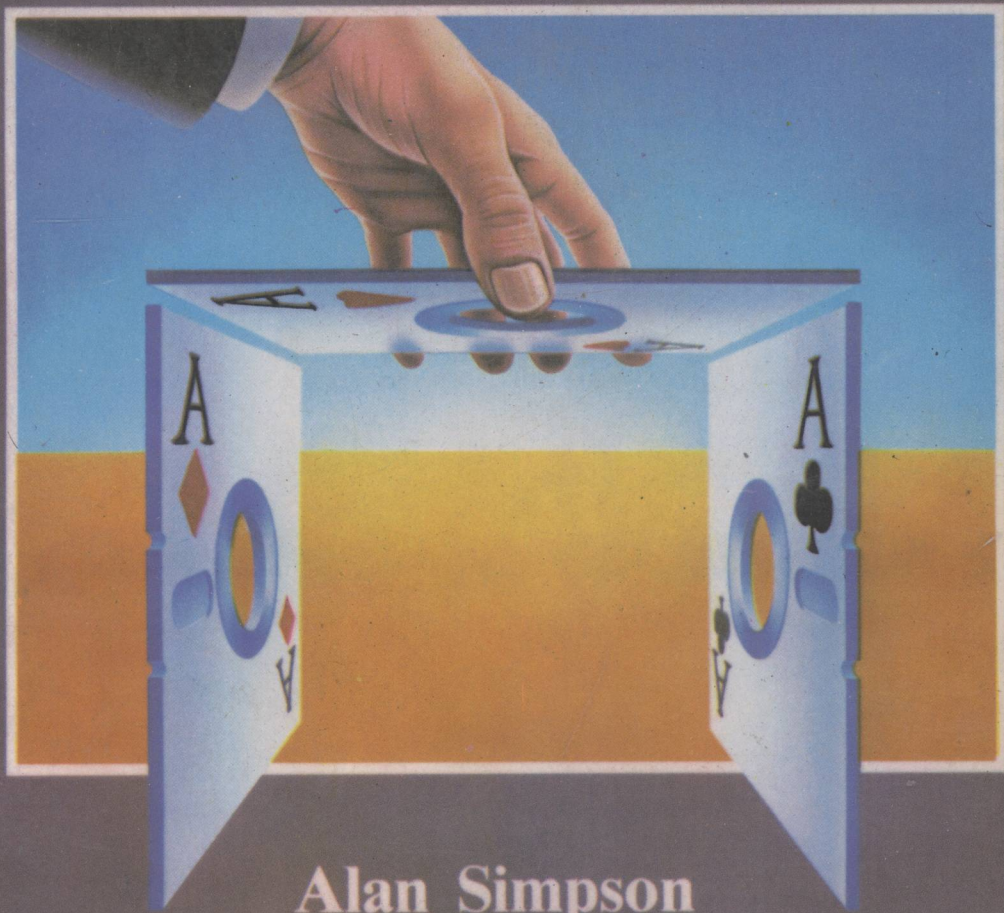




Advanced Techniques in dBASE III



Alan Simpson

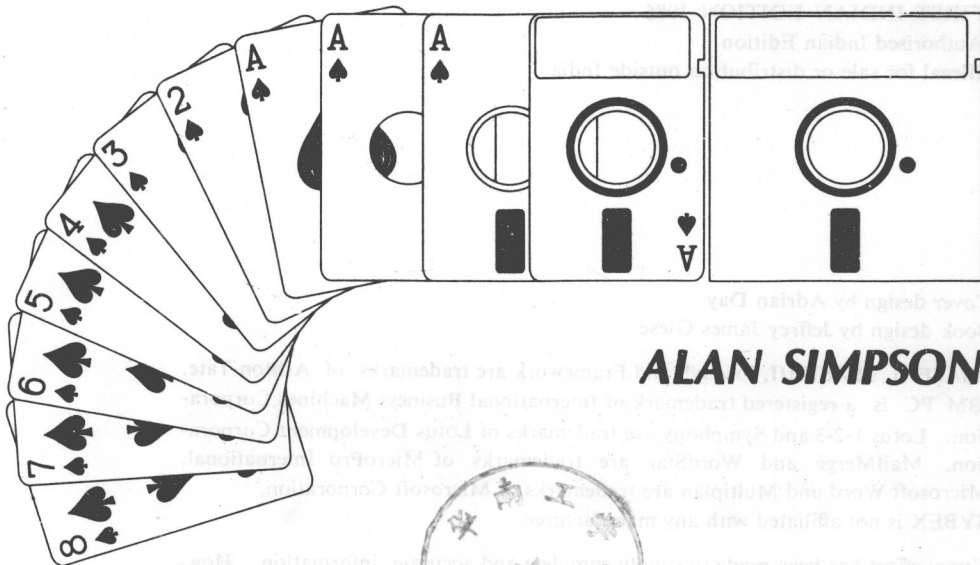
TP312
S613-2

8763911



E8763911

ADVANCED TECHNIQUES IN dBASE IIITM



ALAN SIMPSON



B. P. B. PUBLICATIONS

376, OLD LAJPAT RAI MARKET, DELHI-110006

B.P.B. PUBLICATIONS

376, Old Lajpat Rai Market, Delhi-110006

4794/23, Bharat Ram Road, Darya Ganj, New Delhi-110002

8/1, Ritchie Street, Mount Road, Madras

4-3-269 Giriraj Lane, Bank Street, Hyderabad-500001

Published in India by arrangement with
SYBEX INC., U.S.A.

FIRST INDIAN EDITION 1986

Authorised Indian Edition

Illegal for sale or distribution outside India

Cover design by Adrian Day

Book design by Jeffrey James Giese

dBASE II, dBASE III, Friday!, and Framework are trademarks of Ashton-Tate. IBM PC is a registered trademark of International Business Machines Corporation. Lotus 1-2-3 and Symphony are trademarks of Lotus Development Corporation. MailMerge and WordStar are trademarks of MicroPro International. Microsoft Word and Multiplan are trademarks of Microsoft Corporation. SYBEX is not affiliated with any manufacturer.

Every effort has been made to supply complete and accurate information. However, SYBEX assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which would result.

Copyright © 1986 SYBEX Inc., (U.S.A.) World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

Published by Manish Jain for B.P.B. Publications,
376, Old Lajpat Rai Market, Delhi-110006, and

Printed at Kay Kay Printers, Delhi-110007

ACKNOWLEDGMENTS

Many heartfelt thanks to Dori Shattuck and Chris Matthews for their many technical and editorial contributions to this book.

Thanks to everyone at SYBEX who contributed to this book and produced it, including Bonnie Gruen for her editorial work; Carole Alden for her encouragement; Jeremy Elliott, technical support; Valerie Robbins and David Clark, word processing; Jeffrey James Giese, book design; Donna Scanlon, typesetting; and Dawn Amsberry, proofreading.

Thanks to Bill and Cynthia Gladstone (and Baby Glad) for being great literary agents and friends.

Thanks to Julie for sharing the good times without complaining about the bad.

INTRODUCTION

If you want to write a custom software system in dBASE III, then this book is for you. The emphasis of the book is on practical business programming; writing programs that get the job done, quickly and efficiently. The book presents working business systems that not only perform useful business tasks, but also demonstrate programming techniques that can be used in many business applications.

Unlike most programming books, this book provides step-by-step descriptions of virtually every technique used in every program. Therefore, you don't need to try to figure out "what's going on" from a mass of dBASE III commands—it's already been done for you. As the rationale for each routine in a large software system is revealed, the mystery of programming dwindles. And as the mystery dwindles, your own ability to create custom software systems grows.

WHO THIS BOOK IS FOR

This book is not intended for the computer novice. However, familiarity with the basic commands used in either dBASE II or III will be sufficient background. No prior programming experience is necessary.

STRUCTURE OF THE BOOK

The book is divided into six major sections. The first four chapters discuss general programming considerations, and emphasize techniques for maximizing the speed and performance of a software system. These chapters also provide firm advice for planning ahead and getting the most out of dBASE III's many capabilities.

The second part (chapters 5-10) presents a custom software system for managing a single database. The system is designed to manage data on a membership database, but the techniques presented can be used to manage any single database. The system is specifically designed for the novice programmer, and teaches the basics of creating user-friendly, "menu driven" systems, using index files for maximum speed, creating and using custom screens and reports, and other basic programming techniques universal to all business applications.

Chapters 11–15 present an inventory management system, and demonstrate more advanced techniques for managing multiple databases.

Chapters 16–20 discuss an Accounts Receivable software system. This system demonstrates additional techniques for managing multiple databases, and advanced programming techniques that are unique to dBASE III.

Chapter 21 discusses some handy programs that are useful when working with dBASE III, and also presents some advanced “tips and techniques” for solving tricky programming problems and modifying existing software to better suit your needs.

The last part consists of appendices. Appendix A describes differences between dBASE II and dBASE III, and is intended for readers who are upgrading from dBASE II to III. Appendix B describes techniques you can use to interface dBASE III data with other popular software systems, including Microsoft Word and WordStar (both very effective in creating form letters), dBASE II, Multiplan, Framework, Lotus 1-2-3, and Symphony. Appendix C presents a summary of dBASE commands and cursor control keys for quick reference.

For consistency, the book assumes that you are using an IBM PC or similar computer with two floppy disk drives, A and B. In drive A you should store the dBASE III system disk, and in drive B store the databases and programs you develop. To ensure that the files you create are always stored on drive B, you might want to set up a CONFIG.DB file, as discussed in Appendix A.

If you are using a hard disk system, you’ll want to store both dBASE III and the files you create on the same directory. In this case, you don’t need to change the default drive as long as you access dBASE from the DOS C> prompt and the appropriate directory.

HOW TO BUY THE PROGRAMS IN THIS BOOK

If you wish to purchase the programs in this book, send a check or money order for \$25.00 to:

IBM Data Files
P.O. Box 2802
La Jolla, CA 92038-2802

(California residents please add 6% sales tax). Make the check payable to *IBM Data Files*, and be sure to specify that you wish to buy the programs from Alan Simpson's *Advanced Techniques in dBASE III* book. These programs are currently available in 5-1/4 inch double-sided, double-density, PC-DOS Version 2.1 disk format only (IBM PC and XT format).

HOW TO BUY THE PROGRAMS IN THIS BOOK

If you wish to purchase the programs in this book, send a check or money order for \$15.00 to:

IBM Data Files
PO Box 2802
La Jolla, CA 92038-2802



TYPOGRAPHICAL CONVENTIONS

The following typographical conventions are used throughout the book:

- dBASE III commands are entirely capitalized (CREATE).
- Variables and field names are initially capitalized (Memo field).
- Keyboard keys are upper- and lowercase (PgUp).
- Control-key commands are indicated with the caret (^) symbol.
^C means: hold down Control and C at the same time.

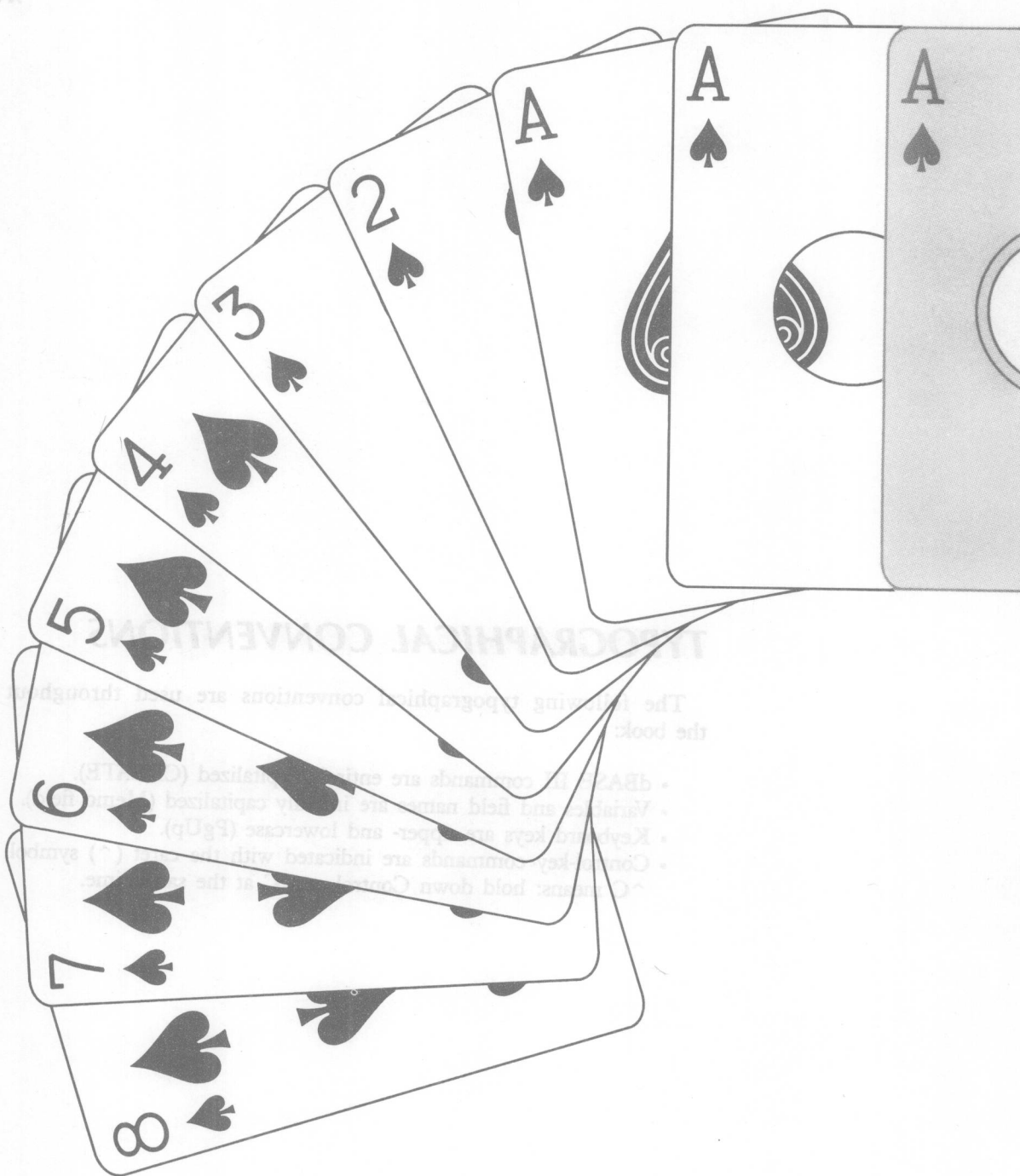


TABLE OF CONTENTS

Introduction	xii
1 dBASE III DATABASE DESIGNS	2
Single Database	2
Single Database with Memo Field	4
Relational Databases	9
Master File/Transaction File Database System	14
2 INDEX POWER	24
Database Indexes	24
Trimming Minutes Down to Seconds	25
Managing Multiple Index Files	41
Points to Remember	44
3 PROGRAMMING IN dBASE III	48
Creating Command Files	48
Interacting with the User	49
Looping with DO WHILE . . . ENDDO	52
Making Decisions with IF . . . ELSE . . . ENDIF	55
Making Decisions with DO CASE . . . ENDCASE	58
Structured Programming	59
Debugging Techniques	64
4 TURNING IDEAS INTO WORKING SOFTWARE: SYSTEM DESIGN	76
Making it Manageable	76

5	MEMBERSHIP SYSTEM MAIN MENU	88
	Getting Started: Writing the Pseudocode	88
	Writing the Command File	89
	Will it Work? Testing the Program	92
6	MEMBERSHIP SYSTEM APPEND PROGRAM	98
	Custom Screens with SED	98
	The AddNew Command File	105
7	MEMBERSHIP MAILING LABELS AND DIRECTORY	112
	Mailing Labels	112
	Custom dBASE III Reports	114
8	MEMBERSHIP SYSTEM SORTING AND SEARCHING	128
	Writing the Pseudocode	130
	Writing the Command File	132
	Testing the Program	140
9	MEMBERSHIP SYSTEM EDITING AND DELETING	148
	Custom Screen for Editing and Deleting	148
	The EditDel Command File	152
10	MEMBERSHIP SYSTEM ENHANCEMENTS	160
	Checking for Duplicate Members	162
	Mailing Labels Alignment Check	166
	The Directory Program	170
	Fancier Menu-Driven Searching	175

11	INVENTORY SYSTEM DESIGN	188
	General Design of Inventory Systems	188
	Inventory System Problem Definition	190
	Inventory System I/O Specifications	191
	Inventory System Database Design	192
	Inventory System Software Design	195
	Inventory System Main Menu	196
12	MASTER INVENTORY FILE	204
	Master-File Software Design	204
	Master-File Menu	205
	Adding Unique Part Numbers	208
	Inventory Master-File Reports	215
	Writing the Command Files	222
	Editing the Master File	233
13	INVENTORY SALES SYSTEM	240
	Sales System Software Structure	240
	Sales System Menu	240
	Point-of-Sale Data Entry	243
	Sales System Reports	253
14	INVENTORY NEW STOCK SYSTEM	268
	Creating the NewStock System	268
	New Stock System Software Structure	269
	New Stock System Menu	269
	New Stock Data-Entry Program	269
	New Stock System Reports	274
15	INVENTORY SYSTEM UPDATING	284
	Updating the Master File	284

Editing the Sales File	290
Editing the NewStock File	296
Summary	307

16 ACCOUNTS RECEIVABLE SYSTEM DESIGN 310

Accounts Receivable Problem Definition	310
Accounts Receivable I/O Specification	310
Accounts Receivable Database Design	312
Accounts Receivable Software Structure	316

17 PROCEDURES AND PARAMETERS 320

The A/R System Procedure File	320
Creating the Procedure File	324
Using the Procedure File	327
A Note on Parameter Passing	327
The GetCust Procedure	329

18 MAIN MENU, DATA ENTRY, AND EDITING 336

A/R System Main Menu	336
Adding New Customers	337
Adding New Charges	341
Adding New Payments	344
A/R System Edit Programs	345
Editing the Customer Database	348
Editing Current Charges	348
Editing Payments	353

19 A/R SYSTEM REPORTS 360

Accounts Receivable Report Menu	360
Printing Invoices	361
Summary and Aging Reports	369

History Reports	373
20 MONTHLY SYSTEM UPDATES	382
21 USEFUL PROGRAMS AND TECHNIQUES	388
Modifying Existing Software	388
An Inventory/Accounts Receivable Linker	393
A Debugging Tool	395
Check Writing Procedure	405
Word-Wrap Procedure	411
Handy Business Formulas	414
Statistical Procedures	416
A FROM dBASE II TO dBASE III	423
Bigger	423
Better	423
Faster	425
What's New	425
What's Different	438
B INTERFACING dBASE III WITH OTHER SOFTWARE SYSTEMS	447
Interfacing with dBASE II	447
Interfacing with Microsoft Word	450
Interfacing with WordStar and MailMerge	458
Interfacing with Multiplan	462
Interfacing with Lotus 1-2-3	463
Interfacing with Framework	465
Interfacing with Symphony	467
C dBASE III VOCABULARY	474
Index	502

ADVANCED TECHNIQUES IN dBASE III

CHAPTER 1

dBASE III DATABASE DESIGNS

Deciding what to store in a database is one of the first steps in designing any software system. Many systems, such as mailing lists, involve only a single database and perhaps an index file or two. More sophisticated systems may use several databases interactively.

In this chapter we'll discuss the four most commonly used database designs: single database, single database with Memo field, relational databases, and master file/transaction file database systems.

SINGLE DATABASE

The simplest database design is the single database. There are just two steps to designing a single database system:

1. Decide what fields to put in the database.
2. Identify key fields for sorting and searching.

A simple mailing list might have the database structure shown in Figure 1.1.

.....
Structure for database : C:mail.dbf

Field	Field name	Type	Width	Dec
1	LNAME	Character	20	
2	FNAME	Character	20	
3	COMPANY	Character	20	
4	ADDRESS	Character	25	
5	CITY	Character	20	
6	STATE	Character	5	
7	ZIP	Character	10	
8	EXP_DATE	Date	8	

FIGURE 1.1: *A sample mailing list database*

Notice that the first and last names are in two separate fields. This is so the database can be sorted by last name only. If there were just a single name field called Name, and the data were stored like this,

Mr. James L. Bower
Andy Zapplbey
Claudia Allen

there would be no way to properly sort the database by last name. By storing the first and last names in separate fields, you can sort by last name (or last name plus first name), and use the sorted file to locate records quickly.

Also notice that the address information is separated into distinct fields: City, State, and Zip. Once again, this is so that the data can be sorted or accessed easily on the basis of any of these independent pieces of information. In general, always break the information in a database into as many distinct fields as possible, since this allows the greatest freedom in sorting and searching.

To create the mailing-list database we've just discussed, type the CREATE command with the name of the file:

CREATE Mail

dBASE III displays a form on which to enter the name, data type, width, and decimal places for each field in the database. When you've completed this screen, enter several sample records, so you can test the index files we'll be creating next.

Now we need to identify key fields for sorting and searching. Mailing lists generally require two sort orders: by last and first names, for printing a directory or looking up individual records, and by zip code, for bulk mailing. To maintain these sort orders permanently, we'll store them in index files. First create an index file of last and first names:

```
USE Mail  
INDEX ON LName + FName to NAMES
```

Then create the index file of zip codes:

```
USE Mail  
INDEX ON Zip TO Zips
```