



D64 IMAGES  
ATTACHED  
TO THIS PDF



Precision  
Software

# *Superbase*

**The Programmable Database  
for your Commodore  
128, 64 and Plus/4**



### Proprietary Rights

This manual shall not be copied, reproduced or translated in part or in whole without the prior written permission of Precision Software Limited.

The magnetic media accompanying this manual contain proprietary information of Precision Software Limited, embodying confidential ideas and expressions, no part of which may be reproduced or transmitted in any form or by any means, without the prior written permission of Precision Software Limited.

### Limited Warranty

The contents of this manual are subject to change without notice and do not represent any commitment on the part of Precision Software Limited.

No warranty is made with respect to the software described herein. The sole obligation of Precision Software Limited shall be to make available modifications and updates in accordance with a policy of continuous improvement, and subject to its normal trading terms and conditions.

Precision Software shall have no liability or responsibility to purchaser or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by this product, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this product.

The above is a limited warranty and the only warranty made by Precision Software Limited. Any and all warranties for merchantability and/or fitness for a particular purpose are expressly excluded.

### Acknowledgements

CBM is a trademark of Commodore Business Machines Inc.  
Diablo is a trademark of the Xerox Corp.  
Qume is a trademark of Qume.  
Epson is a trademark of Epson.

This manual is based on the Superbase 64 edition, published August 1983. Editorial revisions and additional material by Dr. Bruce Hunt and Valerie Challis.

Published September 1985

Copyright © Precision Software Limited, 6 Park Terrace, Worcester Park, Surrey KT4 7JZ, England.

# S U P E R B A S E   f o r   C O M M O D O R E

## YOUR SUPERBASE SYSTEM

Superbase is the creation of Simon Tranmer and Tom Cranstoun, two outstanding designers with wide experience of database and word processing software.

Superbase won the Consumer Electronics Showcase 1984 award for Innovative Personal Productivity Software. The system has more than 50,000 users world-wide and runs on the Commodore 128, 64, Plus/4, 8096, 8296 and 700/B series. It is also available on the Apple // range of computers.

Versions of Precision Software's Word Processor and Spelling Checker, also designed by Simon Tranmer, have sold more than 150,000 copies and are available on the Commodore 128 as Superscript, the Commodore 64 as Easyscript and the Plus/4 as Script/Plus from Commodore. They are available from Precision Software as Superscript or Superscript II on the Commodore 2000, 3000, 4000, 8000 and 700/B series.

### Your Superbase Package

Your Superbase package should contain these items:

- \* The Manual
- \* 1 program disk (backup on other side)
- \* The audio learning cassette
- \* A registration certificate

A separate backup disk can be obtained at the time of registration only.

## **Quick Start**

Experienced users may wish to skip the beginner's instructions.

- (1) With computer off, insert Superbase program disk in drive 0.
- (2) Switch computer on. C64 and Plus/4 users type LOAD"SB",8,1 and press RETURN. Wait for the program to load.
- (3) At Start Up Menu, leave program disk in drive 0 and press RETURN. When you see the prompt 'Enter Database Name', insert any formatted disk and type a name of your choice, maximum 16 characters.
- (4) Now follow the prompts for creating a database and a file within it. File definition commands are summarized on page R-24. Printer defaults are set for Commodore serial type.

## **Software Piracy**

Superbase disks are copy protected. Copying them is illegal. If we find out that you have copied our program, we will take you to court. Each violation of the Copyright Act can bring a fine of up to £50,000. Keep software legal!

## **Attention -- Users Upgrading to Superbase 128**

You can upgrade to Superbase 128 from most of the existing Precision Software database products, i.e.

Superbase 64 (Version 1 or 2) on the Commodore 64 or Plus/4  
Superbase on the Commodore 8096, 8296, or 700/B128

If you do this, we strongly recommend that you read the manual as there is much new material.

Databases and files used in the other products can be used in Superbase 128, but to change from 40 to 80 column format, you should follow the conversion procedure in Appendix G.

## **Attention -- Superscript Users**

Superbase 128 links to Superscript 128. You can load both programs into memory together, and exchange information between them. You can define database fields in a Superscript 128 document, and you can also program Superbase to control Superscript 128, specifying parameters.

## **INTRODUCTION**

This manual presents the Superbase database systems in different ways to suit individual learning preferences.

We recommend that everyone works through the first two TUTORIALS. These take you through the main Superbase activities:

- \* Setting up files
- \* Entering data
- \* Viewing data records
- \* Searching the database
- \* Updating records
- \* Obtaining screen and printed output

A more advanced tutorial explains about features such as Sorting Records, and how to use the Report Generator.

The REFERENCE SECTION is a kind of Superbase Encyclopedia, containing full details of all the Superbase Menu Options.

Most people find the Menu Options cover everything they need to do with Superbase. But for those who want to go a step further, we provide more advanced facilities.

The PROGRAMMING SECTION describes every Superbase command from the programmer's point of view. We tell you how to use the Program Editor, and how to write Superbase programs for things like multi-file access and word processor integration.

At the back, we've included some essential extra information:

- \* Printer control
- \* Error messages
- \* Glossary
- \* Maximum values
- \* The "satellite" programs: labels, delete, and utility
- \* The automatic "start-up" program
- \* Compatibility with earlier versions of Superbase

You'll find a REGISTRATION and BACKUP DISK ORDER FORM enclosed with the package. Please fill it in, and mail it - TODAY! You must be registered to qualify for free technical support and/or backup disks. Among the benefits of registration are free technical support, advance notice of new products, a chance to be considered for our new product evaluation panel, and special limited offers.

<b>TUTORIAL INTRODUCTION</b>		<b>T-1</b>
Instructions for the Tutorial		T-1
You'll Need		
How To Do the Tutorial		
Setting up Instructions		T-2
Loading Superbase and Creating a Data Disk		
<b>1 TUTORIAL ONE</b>		
1.1 Introduction		T-5
1.1.1 Getting Your Computer System Up and Running		
1.1.2 Some of the Keys You'll be Using		
The Shift Lock Key		T-6
The Function Keys		
1.1.3 Loading Superbase and Creating a Data Disk		
1.1.4 What is a Database?		T-7
1.2 Creating a Database and a File		
1.2.1 Giving the Database a Name		
1.2.2 Giving the File a Name		
1.2.3 The Format Mode		T-8
1.2.4 Designing the Layout of the File		
Setting up the First Field		
Setting Field Names		
Setting Field Types		T-9
Setting Field Lengths		
1.2.5 A Summary of the Field Setting Procedure		
Correcting Errors when Setting Fields		T-10
Setting up the Second Field		
Setting up the Address Fields		
1.2.6 A Summary So Far		T-11
1.3 Friend or Business Contact?		
1.3.1 Setting Fields for Birthday and Professions		
1.4 Inverting One Line of Lettering		T-12
1.4.1 Inverting Everything on the Screen		
1.4.2 A Summary So Far		T-13
1.5 Saving the Form Layout on Disk		
1.5.1 Duplicate Keys		
1.6 Using the Menus		T-14
1.7 Entering Names and Addresses		
Entering the Lastname and Firstname		
Entering the Address		T-15
Making an Entry in the B/P Field		
Entering the Birthday		
1.7.1 Storing the Record		
1.7.2 Entering Six more Example Records		T-16
1.8 Superbase Commands		
1.8.1 The Quit Command		
1.8.2 A Summary So Far		
1.9 Viewing the Records You've Stored		
1.9.1 The Select Menu		T-17
One Letter Abbreviations of the Select Commands		

## TUTORIAL CONTENTS

1.9.2	Picking out Records	
1.9.3	Partial Matching	T-18
1.9.4	The Match Option	
	Selecting all the Friends in the File	
	Viewing the Next Record	T-19
1.10	Exiting from Superbase	T-20
1.11	A Summary of Tutorial One	T-21
	Example Records for Your Name and Address File	
<b>2</b>	<b>TUTORIAL TWO</b>	<b>T-22</b>
2.1	Introduction	T-22
2.1.1	Loading Superbase	
2.2	The Help Facility	
2.3	The Database Catalog	T-23
2.4	Creating a New File	
2.5	Setting the Fields	
2.5.1	Copying the Form Design from the Tutorial	T-24
2.6	Some New Field Types	
	Setting a Numeric Field	
	Setting Decimal Places	T-25
	Setting a Result Field	
	Setting a Field as a Constant	
2.7	Saving your Work	T-26
2.8	Entering the Calculations into the Fields	
	The Total-Price Field	
	The Tax Field	
	The Amount Field	T-27
2.9	Entering Records into the Invoice File	
	Saving the Record	T-28
2.9.1	Entering More Records from the Tutorial	
2.10	The Calc Option	
	Using Calc to Update an Invoice	
2.10.1	The Store Command	T-29
2.11	The Batch Option	
	Using Batch to Reduce Prices by 10%	T-30
	Checking that the Batch Calculation has Worked	
2.12	Some Ways of Selecting Records from the File	T-31
2.12.1	The Find Option	
	Telling Superbase What to Search For	
2.12.2	Naming the List	
2.12.3	Viewing the Records in the List -	
	the Output Option	
2.12.4	Changing the Display - Down and Across	T-32
2.12.5	Displaying Records on Screen and Printer	
2.12.6	Some Other Ways of Controlling the Output	T-33
2.13	A Final Summary	T-34
	Invoices - Form Design	T-35
	Example Records for Your Invoice File	T-36

<b>3</b>	<b>TUTORIAL THREE</b>	<b>T-37</b>
3	Introduction	T-37
3.1	Escaping from Commands	
3.2	Disks, Databases, Files, and Lists	
3.2.1	The Database Command	
	Database Names	T-38
	Database Copying	
3.2.2	Disks and Databases	
3.2.3	File Names / Recovering a Lost File	T-39
	Recovering a Lost File	
	Printing Your File Format	
3.2.4	Key Lists / Disk Files	
3.2.5	Backup	T-40
3.3	Creating a Memo Screen	
	Select Memo	
	Write Memo	T-41
	Recall Memo	
3.3.1	Help Memo Screens	
3.4	Changing the File Format	T-42
3.4.1	Changing Fields	
	Fieldnames	
3.4.2	Unique Keys and Duplicate Keys	
3.4.3	Using Numbers as Keys	T-43
3.4.4	Multiple Screens	
	Redefining the File Format	T-44
	Obtaining Extra Screens	
	Text Field	
3.5	Output / Word Processing Link	T-45
3.5.1	Output Command	
3.5.2	Output Format Commands	
3.5.3	'Print' and 'Display' without 'Output'	T-46
3.5.4	Output to Disk for Word Processing	
3.6	Commands and Programs	T-47
	Output Using the Command Line	
3.6.1	Abbreviating the Command Line	
3.6.2	Command Lines with More Than One Command	T-48
3.6.3	Repeating Commands with the Recall Key and Storing Them	
3.6.4	Line Numbers and Programs	T-49
3.6.5	Running Totals / Use of Variables	
3.6.6	Application Programs	T-51
3.7	Sorting Records	T-52
3.8	Producing Reports	T-52
	Report File	
	Report Title	
	Totals and Subtotals	
	Subtotal Break Point	T-53
	Key Order or Use a List	
	Report Detail	T-54
	End of Report	
	Store the Report	

<b>1</b>	<b>The Main Menu - Menu 1</b>	<b>R-1</b>
1.1	Using Menu 1	
1.2	The Options	
<b>2</b>	<b>The Secondary Menu - Menu 2</b>	<b>R-3</b>
2.1	Using Menu 2	
2.2	The Options	
<b>3</b>	<b>Command Lines</b>	<b>R-5</b>
3.1	Using Direct Commands	
3.1.1	Entering Command Lines	
3.2	Repeating Commands	
3.3	The Syntax of Command Lines	R-6
3.4	Further Secondary Commands	
3.5	Filenames in Command Lines	R-7
3.6	Interrupting Commands	
<b>4</b>	<b>Format</b>	<b>R-8</b>
4.1	The Format Option	
4.1.1	Obtaining the Format Option	
4.2	Designing the Record Layout	
	Fields	
	Descriptive Text	R-9
	Fieldnames	
4.3	Setting the Fields	R-13
4.3.1	Setting a Forced Field	
4.3.2	Setting a Key Field	
4.3.3	Setting a Text Field	R-15
4.3.4	Setting a Date Field	
4.3.5	Setting a Numeric Field	R-16
4.3.6	Setting a Result Field	
4.3.7	Setting a Constant Field	R-17
4.3.8	Setting a Calendar Field	
4.4	Further Format Commands	
	Formatting Additional Screens	
	Moving to the Previous Screen	
	Inverting a Line	
	Erasing a Line of Descriptive Text	R-18
	Erasing a Field	
	Deleting a Line	
	Inserting a Line	
	Inverting a Line	R-19
	Inverting the Screen	
	Changing the Color of the Screen	

## REFERENCE SECTION CONTENTS

	Drawing a Border	R-20
	Screen Dump	
	Quit Format	R-21
	Clear Format	
	End Format and Store	
	Duplicate Keys	
4.5	Editing an Existing Record Format	R-22
4.6	Duplicating a File Format	R-23
4.7	Summary	R-24
<b>5</b>	<b>Enter</b>	R-27
5.1	Entering Information	
5.1.1	Obtaining the Enter Option	
5.1.2	Filling in the Blank Record	
5.1.3	Forced Fields	R-29
5.1.4	Date Fields	
5.1.5	Adding, Editing and Replacing, or Deleting Records	R-30
5.1.6	Modifying Numeric Fields	
5.2	Summary	
<b>6</b>	<b>Select</b>	R-32
6.1	Selecting a Record	
6.1.1	Obtaining the Select Option	
6.2	The Select Option Control Keys	R-33
	The '+' and '-' Keys	
6.2.1	Key	
6.2.2	Current	
6.2.3	Next	
6.2.4	Last	R-34
6.2.5	Previous	
6.2.6	First	
6.2.7	Match	
	Summary of Selection Criteria	R-35
	Record Template	
	Entering Criteria for Matching	
	Starting the Search	R-36
	Interrupting and Resuming the Search	
	Ending the Search	
	Sliding Match	R-37
	Field-Independent Matches	
	The 'and' Operator and the 'or' Operator	
	Pattern Matching	R-38
	Using the Delayed Request Character	
	Match Summary	R-39
6.2.8	Output	
6.2.9	Add	
6.2.10	Replace	R-40
6.2.11	Delete	
6.3	Alternative Ways to Execute the Select Commands	
	Selecting from a List	R-41

<b>7</b>	<b>Find</b>	<b>R-42</b>
7.1	Using the Find Option	
7.1.1	Obtaining the Find Option	
7.2	The Default List	
7.3	Naming and Storing a Key List / Appending to a List	R-43
	Appending to a List	
7.3.1	Bypassing the Record Template	
7.4	Using the Key List	R-44
	Renaming the List	
<b>8</b>	<b>Output to Screen, Printer, or Word Processing Disk File</b>	<b>R-45</b>
8.1	Outputting Information from Files	
8.1.1	Obtaining the Output Option	
8.1.2	Changing the Direction of Output	
8.1.3	'Print' and 'Display' Direct Formatted Output	
8.1.4	Output Control Commands Summary	R-46
8.2	Displaying All Records	R-47
8.3	Displaying from a Key List	R-48
8.4	Printing Output	
8.5	Outputting Selected Fields	
8.6	Output Across or Down	R-49
8.7	Outputting Extra Text with Information from Records	R-50
8.8	Outputting Calculations on Information in Fields	
8.9	Formatting and Positioning Fields and/or Variables	R-51
8.9.1	Truncating Output	
	Summary of Output Truncation Variables	R-52
8.9.2	Positioning Output	
	Summary of Output Positioning Commands	R-54
8.10	Abbreviating Output Commands	
8.11	Output to a File	R-55
	Method 1: As for Normal Output	
	Method 2: Using Secondary Command 'Fill'	
8.12	Entering Output Commands in a Command Line	R-56
8.13	Additional Output Control Features	R-57
8.14	Using 'Output To' to Reorganize a Database File	
<b>9</b>	<b>File / Database</b>	<b>R-58</b>
9.1	Files and Databases / Changing Disks	
9.1.1	Obtaining the File Option	
9.2	Changing File by Direct or Program Command	R-59
9.3	Reorganizing a Database	
<b>10</b>	<b>Sort</b>	<b>R-60</b>
10.1	Sorting the Records	
10.1.1	Obtaining the Sort Option	
10.2	Entering the Sort Parameters	

<b>11</b>	<b>Calc - Calculation</b>	R-62
11.1	The Calc Option	
11.1.1	Obtaining the Calc Option	
11.2	Entering Calculations	
	Storing Calculations	
	Semicolons	R-63
11.2.1	Storing Results in Basic Variables	
	Numeric Variables	
	Display Result	
	Text Variables	
11.3	Entering Calculations as Command Lines	R-64
11.4	Entering a New Record with the Calc Command	
<b>12</b>	<b>Batch</b>	R-65
12.1	The Batch Option	
12.1.1	Obtaining the Batch Option	
12.2	Performing Calculations on all Records or Selected Records	
12.3	Calculating Totals	R-66
12.4	Using the Batch Command in a Command Line	
<b>13</b>	<b>Report</b>	R-67
13.1	The Report Generator	
13.1.1	Obtaining the Report Option	
13.2	Using the Report Generator	
	Selecting the File	
	Entering the Title	R-68
	Totals and Subtotals	
	Subtotal Break	R-69
	Using a List of Sorted Records	
	The Subtotal Text	
	Using a Key List	R-70
	Selecting the Main Line of the Report	
	End of Report	
	Saving the Report Program	
13.3	Executing a Report Program	R-71
13.4	Loading a Report Program	
13.5	Other Ways of Creating a Report Program	
13.6	Additional Report Program Features	R-72
<b>14</b>	<b>Execute</b>	R-74
14.1	The Execute Option	
14.1.1	Obtaining the Execute Option	
14.2	Executing Programs from Disk	
14.2.1	Removing the Current Program from Memory	
14.3	Executing the Current Program	
14.4	Saving Programs on Disk	R-75
14.5	The Labels Program	

## REFERENCE SECTION CONTENTS

<b>15</b>	<b>Maintain</b>	R-76
15.1	The Maintain Option	
15.1.1	Obtaining the Maintain Option	
	The Maintain Menu	R-77
15.2	Import and Export	R-78
15.2.1	Import	
15.2.2	Export	R-79
15.2.3	Quick Guide to Importing and Exporting	R-80
	How to Export	
	How to Import	R-81
15.3	Backup	
15.3.2	Single Drive Backup	R-82
	Prepare Disk	
	Insert Source Disk	
	Insert Destination Disk	R-83
	Repeating the Cycle	
15.3.1	Dual Drive Backup	
15.4	New Disk	
15.5	Using the Disk Commands	R-84
	Rename	R-85
	Copy	
	Scratch	
	Scratching a Database File	
	Scratching a Database	R-86
<b>16</b>	<b>Memo</b>	R-87
16.1	Using the Memo Option	
16.1.1	Obtaining the Memo Option	
16.2	Multi-Screen Memos	R-89
16.3	40/80 Column Format	
<b>17</b>	<b>Help</b>	R-90
17.1	Help Screens	
17.1.1	Obtaining the Help Option	

<b>1</b>	<b>Prog - Programming</b>	<b>P-1</b>
1.1	The Prog Option	
1.1.1	Obtaining the Prog Option	
1.2	Using the Program Editor	
1.2.1	Editing Facilities	P-2
1.3	The Superbase Programming Environment	P-4
1.3.1	Superbase Programs: Memory Size and Cnaining	
1.3.2	Important Basic Commands	
1.4	Creating Programs	
1.4.1	Programs and Command Lines	P-5
1.4.2	Program Loops	
1.4.3	Conditionals	P-6
1.4.4	Other Conditionals	P-7
	EOF - End of File	
	IF...THEN	P-8
	ELSE	
	NMAT and PMAT	
1.5	Linking Between Files	P-9
1.6	Additional Program Features	P-10
1.6.1	Direct Selection by Key	
1.6.2	Deleting a Group of Records	
1.7	Integration with Superscript	
1.7.1	Loading Superscript with Superbase	
1.7.2	The 'Superscript' Command	P-11
	The Command Word	
	All or 'From "Listname"'	
	The Command String	
	Returning to Superbase	
	Restoring the Full Text Area	
1.7.3	Examples	P-12
	Command Line Example	
	Program Example 1	
	Program Example 2	P-13
1.7.4	The Superscript Document	
1.7.5	Changing the Text and Program Variable Space	P-14
1.8	Error Handling	
1.8.1	Error Trapping	
1.8.2	Program Errors	P-15
1.8.3	Errors in Superscript	
1.9	Variable and Field Name Rules	
1.10	Permitted Basic Commands	P-16
1.10.1	Disallowed Basic Commands and Expressions	P-17
1.11	Superbase Graphics on the Plus/4	
1.12	Compatibility	
<b>2</b>	<b>Additional Commands</b>	<b>P-18</b>
<b>2.1</b>	<b>The Commands</b>	
2.1.1	Across	P-18
2.1.2	All	
2.1.3	Ask	P-19
2.1.4	Brkoff - STOP Key off	P-20
2.1.5	Brkon - STOP Key on	
2.1.6	Check	
2.1.7	Clear	P-21

## PROGRAMMING SECTION CONTENTS

2.1.8	Close	
2.1.9	Cols	
2.1.10	Cont - Continuous Print	P-22
2.1.11	Convert	
2.1.12	Count	P-23
2.1.13	Database	
2.1.14	Date	P-24
2.1.15	Detail	
2.1.16	Display	
	Do	P-25
2.1.17	Down	
2.1.18	Dump	
2.1.19	Elink - End or Reverse Link	
2.1.20	Endreport	P-26
2.1.21	Eof - End of File	
2.1.22	Eol - End of List	P-27
2.1.23	From	
2.1.24	Is	P-28
2.1.25	Lfeed - Line Feed	
2.1.26	Link	P-29
2.1.27	Lmarg - Left Margin	
2.1.28	Load	P-30
2.1.29	Menu	
2.1.30	Mode	P-31
2.1.31	Nmat - No Match Conditional	
	Nographic	
2.1.32	Open	P-32
2.1.33	Pdef - Printer Definition	
2.1.34	Pdev - Printer Device Number	P-33
2.1.35	Perform	
2.1.36	Plen - Paper Length	P-34
2.1.37	Plus	
2.1.38	Pmat - Partial Match	P-35
2.1.39	Print	
2.1.40	Protect	P-36
2.1.41	Quit	
2.1.42	Restart	
2.1.43	Rlink - Return from Link	P-37
2.1.44	Rmarg - Right Margin	
2.1.45	Save	
2.1.46	Screen	P-38
2.1.47	Set	
2.1.48	Setlink	P-39
2.1.49	Space	P-40
2.1.50	Store	
2.1.51	Subtotal	P-41
2.1.52	Superscript	
2.1.53	Title	P-42
2.1.54	Tlen - Text Length	
2.1.55	Total	P-43
2.1.56	Varspace	
2.1.57	Wait	P-44
2.1.58	Where	

<b>3</b>	<b>Menu Commands</b>	<b>P-45</b>
<b>3.1</b>	<b>The Commands</b>	
3.1.1	Backup	P-45
3.1.2	Batch	
3.1.3	Calc	P-46
3.1.4	Catalog	
3.1.5	Copy	P-47
3.1.6	Directory	
3.1.7	Enter	
3.1.8	Execute	P-48
3.1.9	Export	
3.1.10	File	P-49
3.1.11	Find	
3.1.12	Format	P-50
3.1.13	Help	
3.1.14	Import	
3.1.15	Maintain	P-51
3.1.16	Memo	
3.1.17	New Disk	
3.1.18	Other	P-52
3.1.19	Output	
3.1.20	Prog	
3.1.21	Rename	P-53
3.1.22	Report	
3.1.23	Scratch	
3.1.24	Select	P-54
	Select Key	
	Select From	P-55
	Select Current	
	Select First, Last, Next, Previous	
	Select Match	P-56
	Select Output	
	Select Add	
	Select Replace	P-57
	Select Delete	
3.1.25	Sort	P-58
3.1.26	Status	
3.1.27	Validate	
3.1.28	View Directory	

## **APPENDICES**

### **APPENDICES**

A	Technical Appendix	A-1
B	Error Messages	A-5
C	Selected Glossary	A-12
D	Printers	A-18
E	Superbase Satellite Programs: "Labels", "Delete" and "Utility"	A-24
F	Superbase Versions and Machine Compatibility	A-28
G	Upgrading and Converting from 40 to 80 Column Format	A-29
H	Escaping From Commands	A-31

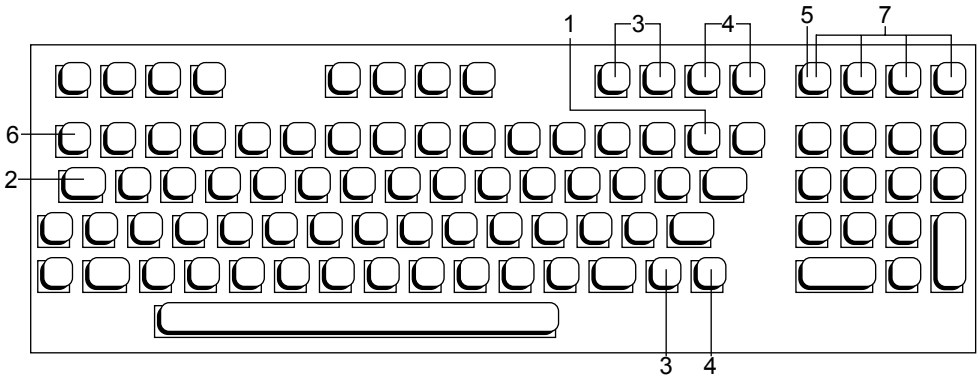
### **INDEX**

I-1

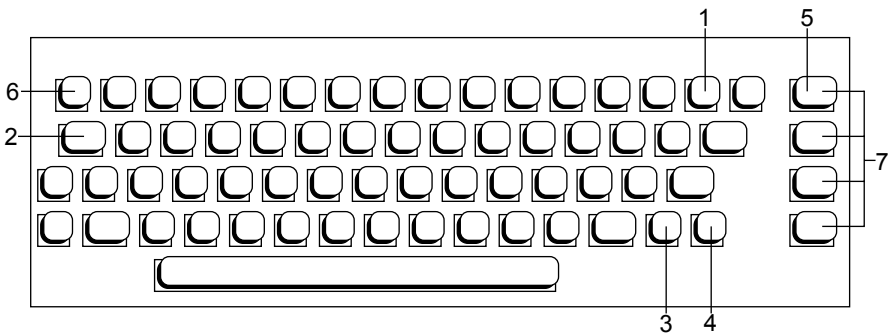
**SUPERBASE FOR COMMODORE 128, 64 AND PLUS/4**

Superbase operates in the same way on the Commodore 128, 64 and Plus/4. The Superbase Audio Learning Cassette refers to the Commodore 64. Familiarize yourself with your keyboard before starting the Audio Tutorials. In particular ensure that you understand how to operate all eight function keys. The keyboard differences are highlighted in the following diagrams.

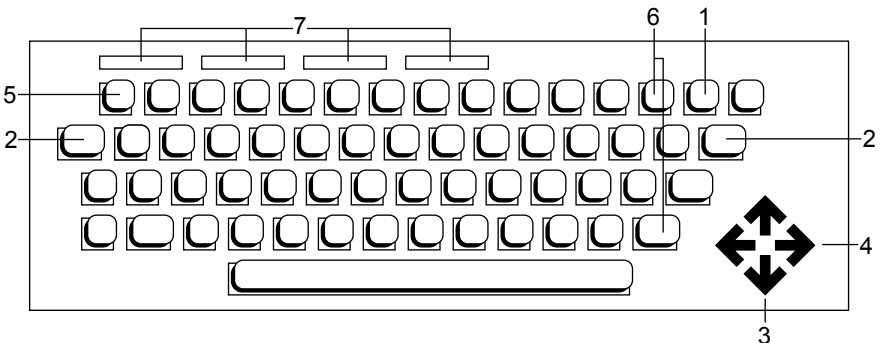
**The Commodore 128**



**The Commodore 64**



**The Commodore Plus/4**



**KEYBOARD DIFFERENCES**

**DIFFERENCES**

**COMMODORE 64/128**

**COMMODORE PLUS/4**

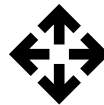
1 Legend



2 Legend



3 Legend



4 Legend



5 Superbase Command Key



6 Retrieve Command Line



7 Function Keys :  
Establish exactly how to  
access each function key  
on your machine.  
See Tutorial T-6.



### INSTRUCTIONS FOR THE TUTORIAL

#### YOU'LL NEED

- \* A COMMODORE 128, 64 or Plus/4 computer, comprising keyboard, disk drive unit and a monitor (either color or black and white). It will also be useful if you have a printer, but this is not essential, as you can follow the tutorial with or without a printer.
- \* A SUPERBASE system disk.
- \* A blank, unformatted disk which we will show you now to format as a data disk.
- \* The SUPERBASE written Tutorial which contains information you'll need to complete the exercises in the Tutorial.
- \* Paper and pen, in case you want to make notes.
- \* If you want to follow the audio tutorial, an ordinary cassette recorder, preferably with headphones to cut out distracting noises. Make sure you know how to PAUSE, STOP and REWIND the tape.

#### HOW TO DO THE TUTORIAL

- \* The procedure is quite straightforward. You start with your computer turned OFF and without a disk in the disk drive. You sit at the computer and follow the instructions you're given.
- \* Set aside enough time for the training session so that you don't have to rush. Remember, you're in control of the training session, so take as much time as YOU need, you won't get a medal for finishing in record time.
- \* Don't worry about making mistakes - most people do at first.
- \* You'll see that the SUPERBASE written Tutorials are split into three levels. Levels 1 and 2 are an edited transcript of the tape. Don't start level 3 of the Tutorials until you've completed the first two levels.
- \* If you're following the written Tutorials, start on page T-5. Pages T-2 to T-4 contain some setting up instructions that the tutorial requires you to read at a certain point.

**SUPERBASE TUTORIAL - SETTING UP INSTRUCTIONS**

**LOADING SUPERBASE AND CREATING A DATA DISK**

Follow these instructions for loading the Superbase program and creating a data disk. Then continue with the tutorial.

You'll need your Superbase PROGRAM disk and an unused BLANK disk.

If you have any problems during these instructions, take the disk out of the disk drive and switch your computer system off. Then switch the system back on and start the instructions from the beginning.

1. First make sure your computer and printer are properly connected and switched OFF. Your disk drive should be connected and switched ON. (It's also useful to know that you should take care to place your disk drive at least 12-18 inches from your screen, or you may get error messages.)
2. Insert your Superbase program disk into your disk drive, or drive 0 of a dual-drive system, and close the door. If you're not sure how to do this, refer to your Commodore User's Manual.
3. Now switch on your printer, monitor, and computer. If you are using a Commodore 128, Superbase will start to load automatically. Commodore 64 and Plus/4 users, load Superbase by keying in:

**LOAD"SB",8,1**

and pressing RETURN.

4. The Superbase logo is displayed while the program is loading into the computer's memory.
5. When it's finished, you'll see this start-up screen:

**Remove Program Disk  
Insert Data Disk and Press Return  
or  
Press F1 to Create Data Disk**

6. The program disk is the Superbase disk. Press the catch on the disk drive door inwards, then take the disk out of the drive and put it back in its envelope.
7. Normally you'd already have a data disk ready to put into the disk drive on which to store your information. But I'm assuming you haven't yet prepared one, so the next steps show you how to create a data disk ready to receive the information you're going to be entering.
8. Press f1 (as the message says) and you'll get another message asking you to insert a blank disk. So put your blank disk in the drive and close the door.

9. Press the RETURN key. You get another message asking for confirmation that you DO want to format this disk and delete any information held on it. The full message reads:

**All Data on Drive 0 Will be destroyed, Are you sure?**

10. Superbase is telling you to stop and ask yourself if there's any information on this disk which you may want to keep. There isn't, so press the letter "Y" for Yes.

A prompt now appears:

**Enter Disk name, id**

Superbase is asking you to give the disk a name and an identification code so that the computer can tell which disk it's working on at any time. This name can be up to 16 characters long, and the identification code can be any two characters.

11. We'll call this disk:

**work disk,a1**

Key that in, including the comma, and press RETURN.

12. The new prompt reads:

**Insert Source Disk**

Your computer needs various pieces of information loaded into its memory before you can start using Superbase. These pieces of information are stored on the Superbase program disk, so the Superbase program disk is the SOURCE disk you must now insert into your drive. Remove the data disk from the drive and replace it with the Superbase disk.

13. Press RETURN. A message asking you to 'Please Wait' now appears in the top left hand corner of the screen, and a series of dots appears across the screen. Superbase is in the process of loading the necessary information from the disk into the computer's memory. So wait until you get the next message.

14. The next message reads:

**Insert Destination Disk**

The destination disk the message is referring to is your data disk. What Superbase has done is to copy the information from the SOURCE disk (the Superbase disk) into the computer's memory. Now it wants to copy that information from the computer's memory onto the destination disk (your data disk).

15. Take the Superbase disk out now and insert your data disk.
16. Press RETURN, and as before, you'll get a series of dots appearing across the screen while the computer copies the information onto the disk. Again, wait until you get the next prompt.

17. Depending on which version of Superbase you're using, you may need to repeat this copying procedure. If this is so in your case, you'll get the message asking you to insert your SOURCE disk again. So you'll have to go through the procedure again (starting at instruction 15). When Superbase has finished creating this disk, it returns to the start up screen. To start work, all you need to do is press RETURN, as your data disk should already be in the drive.
18. You've now completed this disk formatting procedure. You've created a training version of a regular data disk, with extra screens on it that will come in useful for learning about Superbase. You only have to format a disk before you use it for the first time -- so you won't have to go through this procedure again. In fact, other data disks can be prepared with the NEW DISK Option on one of Superbase's menus.

Commodore 128 users can now continue with the tutorial where you left off.

**LOADING SUPERBASE USING A PREPARED DATA DISK**

19. Now you need to load Superbase into your computer once again. So remove your newly formatted data disk from the drive.
20. Insert the Superbase program disk into the drive.
21. Key in:  
  
    **LOAD"SB",8,1**
22. Press RETURN.
23. You'll get a brief message telling you that the computer is searching, then you'll see the PRECISION SOFTWARE logo displayed. The screen will flash as Superbase is being loaded into your computer. The loading process will take about two minutes.
24. When it's finished, you'll get a message  
  
    **Remove Program Disk  
    Insert Data Disk and Press Return  
    or  
    Press F1 to create Data Disk**
25. This time remove the program disk from the drive and insert your newly formatted data disk again.
26. Press RETURN.
27. Now continue with the tutorial where you left off.

**1     TUTORIAL ONE**

**1.1   INTRODUCTION**

Welcome to Superbase, the complete database for Commodore computers from Precision Software. What follows is an edited transcript of the Superbase Audio Training Cassette that comes with the Superbase manual. The cassette covers the first two of the three tutorials.

The first two tutorial sessions will take you through the basics of Superbase. Each tutorial takes you through the construction and development of an actual data file. You'll learn to use Superbase to store information, retrieve information, update it, print it and much more besides. Tutorial Three goes a stage further, and covers some of the more advanced facilities of Superbase. By the time you finish all three tutorials, you should have a good understanding of how a database system actually works and be ready to set up your own system.

The best way to learn anything is to try it out for yourself. That's the idea behind these tutorials. Basically we'll guide you through Superbase while you drive the computer. Don't worry, you won't break anything and you won't cause a disaster if you press the wrong button.

You'll be told exactly WHAT to do and WHEN to do it, so just follow the instructions. You'll be prompted to do various things, for example, press a key, or enter some information. So, try things out for yourself as we go along. The Superbase functions have been designed with ease of use in mind. Starting up the system, selecting the options and making a permanent record of your information are all simple routines that require only a few keys to activate. Remember you're in control, so work at your own pace and take as long as you need.

In the tutorials, we're going to concentrate on the things you'll need to use Superbase on a day to day basis. At the end of the tutorials, you'll be able to explore all of Superbase's facilities for yourself.

**1.1.1   GETTING YOUR COMPUTER SYSTEM UP AND RUNNING**

The first thing to do is to check your computer's ready. Make sure all the components of your system are connected up properly and that they are switched on. If you're not sure how to do that, check with your Commodore User Manual.

If you have a printer, this should also be linked up and switched on. Again, consult your User Manual.

**1.1.2   SOME OF THE KEYS YOU'LL BE USING**

Before we start Superbase, let's have a quick look at your keyboard. I'm assuming you're reasonably familiar with the layout of the keys.

During this session, you'll be using all the ordinary keys together with:

**The CURSOR MOVEMENT keys**  
**The RETURN key**  
**The SHIFT keys**  
**And the INST/DEL key**

If you can't locate any of these, or if you're not sure what they do, check with your Commodore User Manual.

### **THE SHIFT LOCK KEY**

Find the key marked 'SHIFT LOCK' (it's on the left hand side of the keyboard). It's best to leave this key alone when you're using Superbase, as it can affect the way Superbase works if you leave it locked on by mistake. In general, try to use the ordinary SHIFT keys rather than SHIFT LOCK when you want to type a capital letter.

### **THE FUNCTION KEYS**

Take a look at the pad of four keys known as the 'Function Keys'. They're marked with an 'f'.

Each one is assigned a job or in Superbase terms an 'OPTION', so that you can tell your computer to do a specific task by simply pressing the relevant function key, instead of having to press several keys to reach the same end. You'll be using these function keys a lot with Superbase.

Although there are only four separate function keys, each one doubles up giving a total of EIGHT. As you can see, each one has two options on it. You use the Shift key to get the other function on each key. For instance, if you were to press 'f1' on its own you'd operate function one, but if you 'shifted' the same key, you'd get the other function. Again you'll see how to use these in a minute.

If you're using the Commodore Plus/4, the fourth of these function keys is marked 'HELP/f7'. The first of these, when selected from the Superbase Menus 1 and 2, will give you the Help Screens that Superbase includes, and which you can create yourself. However, the 'Select' and 'Maintain' Menus have options marked 'f8' that are not Help Screens. To obtain these, you press 'HELP', as if it were 'f8', the eighth function key.

#### **1.1.3 LOADING SUPERBASE AND CREATING A DATA DISK**

Before we go any further, we need to set up a disk for storing information or 'data'. Superbase is on one disk (the 'program disk'), but we want to store information on a separate disk (a data disk).

The instructions for loading Superbase and creating a data disk are listed separately on page T-2. Remember, if you make any

typing errors while you're following these instructions, use the INST/DEL key to correct your errors. Turn to these pages now, follow the instructions, and then resume the tutorial here.

#### **1.1.4 WHAT IS A DATABASE?**

We're ready to start using Superbase. First let me explain some of the terms we'll be using. To begin with, what is a 'Database'?

Well if you like, it's an electronic filing cabinet! And just as in an ordinary filing cabinet, the information is stored in batches called 'files', and you can think of Superbase as being an office containing a number of electronic filing cabinets.

So you have a number of FILES in a number of DATABASES within Superbase. And you can have as many DATABASES as you want.

Superbase will allow you to hold up to fifteen files in each database you create. And each of the files is made up of separate bits of information known as RECORDS. For example, you may have a file containing names of companies. In that case each company name would be one RECORD. (But don't worry about remembering all these terms right now, they'll make more sense when we come to use them.)

### **1.2 CREATING A DATABASE AND A FILE**

Let's have a go at using Superbase to create a file for holding the names and addresses of your friends and business contacts.

You should see a message at the top of the screen which reads:

**mode : processing**

#### **1.2.1 GIVING THE DATABASE A NAME**

You get a message asking you to ENTER A DATABASE NAME. Superbase is already set up with a database called 'TRAINING', so key in the word 'TRAINING' (in upper or lower case) now. Remember, if you make a typing error, use the INST/DEL key to put it right. And then press RETURN.

You're now in your first database. The next thing you must do (as the prompt says), is to enter a name for the file you're about to create.

Note: Double quotes should not be used in database names and filenames.

#### **1.2.2 GIVING THE FILE A NAME**

This one's for names and addresses, so we'll call it:

**addresses**

Key that in and press RETURN.

The next prompt is telling you this file does NOT already exist and it's asking if you want to use this name, 'Addresses' to create a new file. Well you do, so press the letter 'Y' to answer 'Yes'.

After a few moments, you get a message which reads:

**mode : Format**

The rest of the screen goes blank.

### **1.2.3 THE FORMAT MODE**

Remember, we want to create a file to hold names and addresses, and we want them in an organized format. That's exactly what you do in the FORMAT MODE. You start with a blank screen and draw an empty form with spaces for the information you want to store. If you like, it's a template for the records you'll be storing later.

Let's start designing the layout of our first file.

### **1.2.4 DESIGNING THE LAYOUT OF THE FILE**

This file's going to hold names and addresses of friends and business contacts. For friends we'll keep a record of their birthdays and for business contacts, a record of their profession.

We're going to split each part of the name and address into separate entries or 'FIELDS', as they're known. We'll have a separate field for 'Lastname', a separate field for 'Firstname' and so on. You'll see what I mean if we have a go setting up some fields.

#### **SETTING UP THE FIRST FIELD**

##### **SETTING FIELD NAMES**

Look at the screen and you'll see the cursor; that's the small flashing square which marks where you are on the screen. Use the cursor control keys to move the cursor down one line and then to the right one space,

And now key in the first field name, which is:

**Lastname**

To finish setting up this field, you have to tell Superbase how much space you're going to allocate for 'Lastname', in other words, you have to set the 'field length'.

To set the field lengths you use function key number 1, that's f1. So first move the cursor one space to the right (that's where we want the field to begin), and then press the function key marked f1. The word 'mode' at the top of the screen now flashes to tell you that Superbase is ready to set the field.

### **SETTING FIELD TYPES**

We can set up various types of fields; text fields, date fields, numeric fields and so on, all of which I'll explain as we go along.

Right now we're going to make this 'Lastname' field the 'Key' field. To do this, press the letter K once.

And you get a prompt asking you to 'Set Key'. Superbase is asking you to set the field length. You'll also see a small rectangular marker which tells you where the 'Lastname' field will start.

### **SETTING FIELD LENGTHS**

On the right hand side of the message area you'll see the number 1. This indicates the length of the field at the present moment. As you increase the length of the field so this number increases.

Not many people have lastnames more than 15 letters long, so move the cursor across to the right until the indicator reads 15. If you go too far across, simply move the cursor back until it's in the right place.

If your field length indicator now reads 15, press RETURN.

You'll now see a square marker appear. This marks the end of the 'Lastname' field.

#### **1.2.5 A SUMMARY OF THE FIELD SETTING PROCEDURE**

You'll be repeating this procedure for setting up fields time and time again, so let me summarize it for you:

- STEP 1. Key in the name of the field.
- STEP 2. Move the cursor to where you want it to start.
- STEP 3. Mark the start of the field by pressing f1.
- STEP 4. Set the type of field.
- STEP 5. Move your cursor to where you want the field to end, i.e. set its length.
- STEP 6. Press RETURN.

**CORRECTING ERRORS WHEN SETTING FIELDS**

If you make a mistake setting up a field, you can easily erase it after you've pressed RETURN. You place the cursor on either of the field markers, then press 'f1' and the letter 'E' for 'Erase'. The field will then disappear and you can start again.

**SETTING UP THE SECOND FIELD**

Next we'll create a field for firstnames. So move the cursor down two lines. And now back to the left until it's in line with the 'L' of Lastname. And key in the word:

**Firstname**

This time we're going to make it a 'Text' field, which as you'll see later has a different use to a 'Key' field.

Move the cursor to the right one space. Now as you did for the 'Lastname' field, press the 'f1' key. But this time, press the letter 'T' to tell Superbase you want this to be a 'Text' field.

You'll see that this field is marked with a different marker, so you can tell the difference between field types.

We want this field to be 18 characters long, so move the cursor across until the indicator reads 18 and now press RETURN.

Ok you've set your first two fields, for 'Lastname' and 'Firstname'. Next we'll set the address fields. Move the cursor down two lines and across until it's situated below the letter 'F' of 'Firstname'.

**SETTING UP THE ADDRESS FIELDS**

We'll split each address into four separate fields, that's address line 1, address line 2, address line 3, and address line 4. You could call them anything you like, for instance, Street, City, State, Code, whatever you prefer.

Let's key in our first address field. Key in:

**address1**

Again we want the address lines to be text fields, so move the cursor along one space press the 'f1' key and then letter 'T'. Not many address lines are more than 24 letters long, so we'll set all four address fields at 24 letters long, starting with this one. Have a go at this one on your own.

Now move the cursor down one line and across until it's under the 'a' of 'address1'. Then key in:

**address2**

Then move one space to the right and press 'f1', followed by the letter 'T'. Then set the field length on your own, to 24 again.

Now I want you to set the next two address fields on your own and make them each 24 letters long.

### **1.2.6 A SUMMARY SO FAR**

You've finished setting up the fields for the names and addresses of your friends and business contacts and you made all of them 'Text' fields, except for the 'Lastname' field which you made a 'Key' field.

## **1.3 FRIEND OR BUSINESS CONTACT?**

Next, you need some way of noting whether each person is a business contact or a personal friend, so we'll set up a field one character long into which you can enter a 'B' for Business or a 'P' for Personal.

Move your cursor down two lines and position it below the first 's' of address line 4.

Now type in:

**B/P**

That's B for Business and P for Personal. Now move the cursor to the right one space and on your own, mark the field, set it as another 'Text' field, one character long.

Just to remind ourselves what B/P stands for in future, let's just make a note to the right of the field so we don't forget. Move the cursor to the right three spaces and type in:

**Business/Personal**

### **1.3.1 SETTING FIELDS FOR BIRTHDAY AND PROFESSIONS**

You're going to be recording the names of your friends, so it would be useful to also keep a record of their birthdays. So let's now set up a field for birthdays.

Move the cursor down three lines and position it below the letter 'a' of address4. Now type in the word 'Birthday', then one space, then press 'f1' again.

This time we're not setting a text field, we're setting a DATE field, so watch the cursor as you press 'D' for 'Date', press D now. You'll see the cursor has jumped across the screen seven characters, that's because Superbase automatically gives you seven characters for your date field. Now press RETURN.

There's only one more field to set up now in your name and address file. This field will keep a note of what profession each of your business contacts is in. So let's call this field 'Profession' and make it another 'Text' field. Start by moving the cursor so it's positioned two lines below the letter 'B' of 'Birthday'.

Now key in the word 'Profession' and then set this as a 'Text' field, 15 characters long.

That's it, you've finished setting up all the fields, so the next thing to do is to save this format on disk.

#### **1.4 INVERTING ONE LINE OF LETTERING**

But just before we do that, I want to show you a few of Superbase's features which will help to improve the look of our form layout.

First I'm going to show you a way of 'Inverting' the lettering.

Start by moving the cursor onto the letter 'L' of the word 'Lastname'. Then press 'f1' and then the letter 'I'.

That 'I' stands for 'Invert', and that's just what it's done to the 'Lastname' line. Instead of having dark lettering on a light background, you now have light lettering on a dark background. To get the screen back to normal, simply repeat the process. Try it now, press 'f1' and the letter 'I'.

##### **1.4.1 INVERTING EVERYTHING ON THE SCREEN**

Now we'll invert the WHOLE screen. Simply press 'f1', then the letter 'S', and everything's now inverted.

If you're using a color screen, we'll try changing the colors. It's easy, just hold the CONTROL key, and at the same time, press the number 1 key a few times.

As you see, the color of the background changes each time you press number 1. The number 2 and 3 keys change the color of the other parts of the screen in the same way. So it's the control key and either 1, 2, or 3. Have a go at changing colors now on your own until you get a combination you like.

### **1.4.2 A SUMMARY SO FAR**

You've entered all the field names and field lengths needed for your name and address file. You've inverted the lettering on the whole screen with 'f1 S'. Finally you had a go at changing the colors on the screen.

## **1.5 SAVING THE FORM LAYOUT ON DISK**

So far the form you've designed is stored in the computer, but you'd lose it if you switched the computer off. So you need to store it permanently on disk. Your data disk should still be in the disk drive, so press 'f1', and then the key marked RUN/STOP.

Superbase now works its way down your form and changes the field markers to 'greater than and 'less than signs (> and <). It will then display a question on the message line:

**Allow Duplicate Keys?**

### **1.5.1 DUPLICATE KEYS**

If you remember, we set the 'Lastname' field as a 'Key Field'. That means that when you enter names in future into this field, Superbase automatically sorts and stores them in alphabetical order.

We chose 'Lastname' as the key field, but we could have made any of the other fields the Key Field, for example 'Address1' or maybe 'Firstname', as long as we only have one 'KEY FIELD' for each record. Superbase would then have stored these in alphabetical order and used them for reference purposes.

The question on the screen is asking you if you're going to allow Superbase to accept more than one record with the same key field. In other words, is it okay to accept more than one Mr Brown into our name and address file? If you answered 'Yes', you could have more than one Mr Brown in the file.

In this case, we do want to allow duplicate keys, but there are times when duplicate keys are NOT a good idea. When you've finished the tutorial, have a look in your Reference Manual to find out more about this. Now press the letter 'Y' to answer 'Yes'.

Superbase now displays the word 'Processing', while it's saving your form. And that's it, you've finished designing the file format for our name and address file.

## **1.6 USING THE MENUS**

You should have a message on your screen which reads:

**mode : Menu**

Every time you create a new file, Superbase automatically brings up a FORMAT option so you can design the format for the new file. But every time you call up a file that's already been used, you'll be automatically put into the MENU mode.

There are two menus available, MENU 1 and MENU 2. They both list the various things you can do with the file you're working on. Right now, you should be looking at MENU 1. To see MENU 2, simply press the RETURN key once and then have a look at it to see the difference.

You'll see a list of 'f' keys down the screen, these refer to the function keys on your keyboard. Each one gives you a different option, for example option 2 says 'FORMAT'. You'd use this one if you wanted to design another format.

## **1.7 ENTERING NAMES AND ADDRESSES**

We want to enter some names and addresses into our name and address file so we need MENU 1. Press RETURN again to get back to MENU 1 and we need to use the Enter option, so press 'f1'.

The screen displays the heading:

**mode : Entry**

You'll see the form you've just designed. Your cursor is flashing at the start of the 'Lastname' field. And if you look in the top right corner of the message area, you'll see:

**# 1 k**

This tells you you're now in field number one and the letter 'K' tells you it's a KEY field.

### **ENTERING THE LASTNAME AND FIRSTNAME**

We'll imagine you have a friend called John Baker, whose name and address you want to put in the file. Key in his last name, 'Baker' and press RETURN.

The cursor jumps down to the beginning of the next field, the 'Firstname' field. Here key in 'John' and press RETURN. The cursor moves to the 'address1' field.

You'll find that the cursor will move about quite freely within the fields, but will not move anywhere outside.

Try it. Press the upward cursor control key just once and you'll see the cursor moves back up to the previous field, NOT into the spaces between the two fields.

**ENTERING THE ADDRESS**

Press RETURN just once to move the cursor back to the address1 field. Then key in:

**20 Rose Terrace**

And press RETURN.

Now type the town as 'Woking'. And press RETURN. Now type the final part of John Baker's address, the code, 'WR2 6TJ', and press RETURN.

There's no address line 4 for Mr Baker's address, but this one will come in handy in future for people with longer addresses. We can skip it now by pressing RETURN.

**MAKING AN ENTRY IN THE B/P FIELD**

Next the Business or Personal field. John Baker is a personal friend of yours, so key in the letter 'P' and press RETURN.

**ENTERING THE BIRTHDAY**

Now you need to fill in the birthday field. Remember, we set this one up as a DATE field. And with Superbase, you always have to enter dates in a particular form, either the DAY of the month, a three letter abbreviation of the MONTH and the last two digits of the YEAR, or if you prefer, you can put MONTH, DAY, YEAR. Try entering a date. Key in:

**12jul54 or jul1254**

And press RETURN.

The message area now displays 'MON' (short for MONDAY). Superbase is telling you which day of the week John Baker was born.

We don't want to fill in the profession field for John Baker, so that completes your first record. All you need to do is to save your work.

**1.7.1 STORING THE RECORD**

Press RETURN to skip the Profession field and Superbase asks you to press RETURN to store. Press RETURN.

Superbase now invites you to enter another record. Well, we DO want to enter another, so do as the message tells you, and press the space bar just once.

### **1.7.2 ENTERING SIX MORE EXAMPLE RECORDS**

At the end of Part 1 of the written tutorial, you'll find six more example records which I want you to enter into your name and address file on your own. Enter the six records now.

After you've pressed RETURN to store the last of the six records, your screen will be prompting you to press the space bar to enter another. We don't want to enter any more, so press RETURN to get back to MENU 1.

## **1.8 SUPERBASE COMMANDS**

Before we take a look at some of the other options on the menus, I want to tell you about Superbase's COMMANDS.

Superbase can be controlled in two ways, either by using the menus, or by using commands. In other words you can either use the menus and the function keys to get to certain options OR you can key in the commands on the command line.

Try it. Instead of pressing the 'f1' key to use the enter option, just key in the word ENTER. You'll see it appear on the command line at the top of the screen. Now press RETURN, and you'll see you're in ENTRY mode. You can get any of the menu options in this way, by simply keying them in on the command line and pressing RETURN.

### **1.8.1 THE QUIT COMMAND**

But we don't want to enter any more records right now so to get rid of this option, you have to tell Superbase to 'quit'. First press 'f1', then the letter 'Q' for 'quit'. And you're back at MENU 1. There are other ways of quitting the options which you'll find out about in Tutorial 3. A full list is provided in Appendix H.

### **1.8.2 A SUMMARY SO FAR**

Up to this point, you've designed the format of your name and address file, you've entered a number of records and saved them on disk, you've seen how to use the menus and options, and you've had a go at entering commands on the command line.

## **1.9 VIEWING THE RECORDS YOU'VE STORED**

Now you'll want to view the names and addresses you've stored. Look at the menu and you'll see the SELECT option and it's function key number 2, 'f2'.

### **1.9.1 THE SELECT MENU**

You'll now see one of Superbase's 'SUB-MENUS'. This one's called the SELECT menu, and it lists eleven options. The SELECT menu offers you various ways of displaying the records in a file.

To start with, let's just 'browse' through the records in our file. It's like flicking through the pages of a name and address book. We'll turn to the first page.

If you look at the SELECT Menu on the screen now, you'll see an option called 'First', so press 'f6'.

You'll see the record for 'Julie Adams'. Although this was not the first record you entered, it's now the first in the file, because Superbase sorts key fields into alphabetical order, and your key field was for 'Lastname' so the name 'Adams' is now first in the file.

### **ONE LETTER ABBREVIATIONS OF THE SELECT COMMANDS**

Across the top of the screen you'll see a row of letters 'n,l,p,f,k,c' etc; These are abbreviations for the commands you've just been looking at on the SELECT menu. These abbreviations are very logical. For instance, to view the next record, simply press the letter 'n' for 'Next', and you get the second record on the file for John Baker.

Press 'n' again and you get George Collins' name and address. To go backwards through the file is just as easy. Press 'p' for 'Previous', and you get the previous records.

To get to the last record in the file, press the letter 'l' once. And here you are at the end of the file with Richard Thomas record. You can prove it's the last record in the file by pressing the letter 'n' once more, and you'll see a blank record and a message on the screen telling you it's the end of the file.

Now press 'f' for 'First' to get to the first record.

Now I want you to practice browsing through the file using the one letter commands I've just shown you, (f,l,n,p) First, Last, Next, Previous.

When you've finished practicing, press RETURN just once to get back to the SELECT menu.

### **1.9.2 PICKING OUT RECORDS**

Next we'll have a look at ways of picking out particular records, because if you're dealing with a much larger file, you won't want to browse through them all to find the one you want!

First, we'll try picking out records by particular key fields. We want the option which says 'Key' and it's function key number 1, so press 'f1'. Superbase asks you for the key field you're looking for. Let's imagine you want to find the address of Mr Smythe, so key in:

**Smythe**

Press RETURN and the record for Jeremy Smythe appears on the screen.

### **1.9.3 PARTIAL MATCHING**

Now press RETURN to get back to the menu. You can often save time by typing in an abbreviated version of the key field entry you're after, you don't always have to type in the full key option and so this time key in:

**Smy**

Those are the first three letters of SMYTHE. And press RETURN.

You'll still get Jeremy Smythe's record, even though you abbreviated his name.

You'll see a message on the screen telling you it's only a partial match. It's not the exact one you asked for, but it is the nearest one Superbase could find. Now press RETURN to get back to the menu.

### **1.9.4 THE MATCH OPTION**

Next, we'll have a look at one of Superbase's most powerful options, the MATCH option.

This allows you to call up a group of records which all have something in common, for instance you could call up (or 'Match') all those people whose first name was David, or maybe all those people who live in the same town. Try it.

### **SELECTING ALL THE FRIENDS IN THE FILE**

We'll match all those people who are personal friends of yours and view their records. Press the 'f7' key now to get the 'Match' option and you'll get a blank record and a message prompting you to 'Select Match Data'.

So you must tell Superbase what criterion it must base its selection of records on. There are many ways of doing this and you'll find a list of them in the reference section of your manual.

For now, we want to match all those people who are personal friends. Move the cursor down to the B/P field and key in a Capital 'P' for Personal.

This is the only criterion we're going to give Superbase to make its selection on, so press SHIFT and the RETURN key together. This tells Superbase to start searching through the records. Superbase now searches for all those people with a 'P' in the B/P field.

After a few seconds, Superbase displays the first of the records it's found that matches the criterion. If you look in the B/P field, you'll see a letter 'P' to signify this person is a Personal friend.

### **VIEWING THE NEXT RECORD**

To view the next record that matches this criterion, simply press the letter 'm' for 'Match' and you get the next personal friend, John Baker. I want you to carry on pressing the letter 'm' to check that Superbase has found all the relevant records.

You ALWAYS have to go through all the relevant records in the file to finish the matching process, but it's a bit slow having to look at each one in turn to get to the end. But you can reach the end quickly by simply pressing 'l' for 'Last'.

Now press RETURN to get back to the SELECT menu. You'll use the 'Match' option a lot in future, so we'll try a few more ways of using it.

We'll find out which of the people in our file is a doctor by profession. If you think you know how to do it, have a go now on your own. If you're not sure, then read on.

To find the doctor in our file, press the 'f7' key to get the match option. Then move the cursor to the profession field and key in the word 'Doctor'. Then press SHIFT and RETURN together to start the search. Superbase finds the first doctor in the file and displays the record on the screen.

To check that there are no more doctors in the file, simply press the letter 'm' to continue the search for doctors. There are no more doctors, and Superbase displays a End of File message. Now press RETURN to get back to the Select menu.

Next we'll try using the match option in a slightly different way. We'll find those records which DO NOT contain a particular field. For instance, imagine you wanted to pick out all your friends who live OUTSIDE Birmingham. Try it.

Press 'f7' to get the match option. In all the addresses in our file, the town or city, in this case Birmingham, will appear in the third or fourth address line. So move the cursor down to address line 3 and key in :

**#Birmingham**

Now do the same in address line 4. The hash symbol tells Superbase you want to find all those addresses which DO NOT contain the word Birmingham in address line 3 or 4.

Remember we're looking for all your 'Friends' who live outside Birmingham, so next we must tell Superbase to only find personal friends, in other words, those records with a letter 'P' in the B/P field.

Move down to the B/P field and as we did before, key in the letter 'P', to tell Superbase to find personal friends only. Then press SHIFT and RETURN together as usual and Superbase begins its search.

Superbase finds the first record that matches our criteria, it's John Baker's record. Now press 'm' once again to find the next matching record in the file. Superbase can't find anymore, and gives you the 'End of File' message as before. Your name and address file is only a small one but if you needed to search through a larger file the MATCH Option would come in very useful.

We've finished with this now, so press RETURN to get back to the SELECT MENU and RETURN again to MENU 1.

**1.10 EXITING FROM SUPERBASE**

We've finished with Superbase for the moment, so try exiting from the program. Key in:

**quit**

And press RETURN.

Your screen now shows the Commodore display. If you want to take a break, take the disk out of the disk drive and switch off your computer. But just before you do that, let's just check what you've done so far.

**1.11 A SUMMARY OF TUTORIAL ONE**

You've created a file and designed its format. You've entered records into the file, and you've seen how to browse through those records and pick out particular ones with the Key and Match Options. You've seen how you can control Superbase with commands or by using the options from the menus and sub-menus.

In Tutorial Two we're going to leave our name and address file and create a brand new file called 'Invoices', so that you can try out some of the other things you can do with Superbase.

**EXAMPLE RECORDS FOR YOUR NAME AND ADDRESS FILE - TUTORIAL ONE**

Here are six more example records for you to enter into your name and address file. Enter them in any order and Superbase will sort and store them in alphabetical order of the last names.

**Note.** It is important that you enter all six records as we shall be using them later in the tutorial.

Adams, Julie  
783 Hilton Road  
Selly Oak  
Birmingham  
B32 3TC  
Personal - 12may60

Roddis, Kevin  
67 Apple Tree Road  
Kempsey  
Worcester  
WR3 6FG  
Business - builder

Collins, George  
Taps Plumbers  
34 High Street  
Worcester  
W2 8JS  
Business - plumber

Smythe, Jeremy  
Mansion House  
Hilltops  
Worcester  
WH1 7TY  
Business - MP

Evans, Paul  
22 St John's Road  
Malvern  
Worcester  
WR12 2HA  
Business - doctor

Thomas, Richard  
23 Malvern Road  
Selly Park  
Birmingham  
B27 5TQ  
Personal - 14may56

**IMPORTANT** This tutorial example allows duplicate keys, but we advise you to keep to unique keys wherever possible, if necessary by adopting a coding system for your records. See Tutorial 3, Section 3.4.2.

**2      TUTORIAL TWO****2.1    INTRODUCTION**

At the end of the first session, we closed down Superbase with the quit command and we switched the computer off. So before we start the second session, make sure your computer's switched on.

**2.1.1   LOADING SUPERBASE**

Now you need to load Superbase again. I'd like you to do this on your own using the instructions from Tutorial Introduction. C128 users, follow the instructions on page T-2, steps 1 to 6. When you've removed the program disk, insert the data disk you used in the last session, and press RETURN. C64 and Plus/4 users, find the section headed 'Loading Superbase Using a Prepared Data Disk' and follow the instructions from step 19 to step 27. After a few seconds, you'll see a screen display of the following prompt:

**ENTER DATABASE NAME**

Now key in:

**TRAINING**

And press RETURN.

You'll now see a list of all files in this database. It's called a 'DATABASE CATALOG' and we'll come back to it in a minute. For now, press RETURN to get to MENU 1.

During this session, we're going to use Superbase to create a file called 'Invoices'.

**2.2    THE HELP FACILITY**

First, let's look at Superbase's HELP facility. There's a HELP option on both Menu 1 and Menu 2 and as the name suggests, this facility is designed to help you out should you forget what the various menu options do. Have a look at one of them.

Press key 'f8' or 'HELP'. You use this key for Help Screens on both menus.

At the top of the screen, Superbase is asking which Help screen you require. See what it says about ENTER option, that's the one we've been using. Key in the word 'ENTER' and press RETURN.

You'll now see a 'Help Screen'. This one describes how to use the ENTER option. The instructions at the top of the screen tell you how to view the next screenful of information about the ENTER option, so follow the instructions while you read the screen.

When you've seen all the available screens about the ENTER option, you should now be back at the menu. That's a help screen, and of course, there's a help screen for each of the options on both menus. So in future if you're not sure about one of the options, call up the help screen.

### **2.3 THE DATABASE CATALOG**

Make sure you have Menu 2 on your screen, the one that begins with 'f1 File'. If you haven't got this on the screen, then press RETURN.

We're going to create a new file, so we'll use the file option, Press 'f1' and your screen displays a list of all the files held in the database you set up right at the beginning of the first session. Remember, you called your first database 'TRAINING'. The list you now see is called a 'DATABASE CATALOG'. As you see this Database catalog includes your file, your name and address file called 'Addresses'. You may also see some others, but you can ignore them for now.

You'll notice the catalog also gives you the number of records held in each file. Your name and address file should contain seven.

### **2.4 CREATING A NEW FILE**

Now key in the name of your new file. Key in:

#### **Invoices**

Now press RETURN. Superbase asks if you want to create a new file. You do, so press the letter 'Y' for Yes.

You're in the FORMAT option, as we were at the beginning of the last session. Again, we must design the format for our new file.

We'll imagine you're running a small company selling stationery, and you want to set up a file to hold details of all invoices your company has sent out.

### **2.5 SETTING THE FIELDS**

Move your cursor down two lines and then right two spaces. Now in capital letters, key in:

#### **INVOICE NUMBER**

Make sure you turn shift off for the space in between 'INVOICE' and 'NUMBER'. If you don't, you'll get an unwanted character between the two words.

Make sure shift is off again, then move one space and make this a Key field using 'f1' the letter 'K' for 'Key'. Watch the indicator in the top right corner of the screen and move the cursor along until it reads 6. Then press RETURN to set the field length.

### **2.5.1 COPYING THE FORM DESIGN FROM THE TUTORIAL**

Look in the back of Tutorial 2, and find the printed copy of the Invoice file you're now designing. It's headed:

#### **INVOICES - FORM DESIGN**

You'll see the printed copy tells you what type of field each field should be. Remember it's 'f1' and the letter 'T' for a 'Text' field, and 'f1' and the letter 'D' for a DATE field. The length of each field is also given. For the fields not marked, just key in the field name and we'll put in the field type and length later on.

When you've finished, you should have nine field names altogether, and four of them should have fields set up: 'INVOICE NUMBER', 'Customer', 'Date', and 'Goods'.

### **2.6 SOME NEW FIELD TYPES**

Up to now, you've only tried setting up 'Key', 'Text' and 'Date' fields. In this file you're going to set up three different types of field.

The first field, 'INVOICE NUMBER' is a 'Key' field so that in future you can easily pick out any record you want simply by quoting the Invoice Number.

Below that are fields for the 'Customer', the Date the invoice was sent, and, further down, the 'Goods' field is to hold a description of what was sold.

#### **SETTING A NUMERIC FIELD**

The next field is 'Quantity' and it'll hold the number of items purchased. As this is for holding a number, we'll set it as a 'Numeric' field. First move the cursor one space after the word 'Quantity'.

Press 'f1' to set the field, then the letter 'N' for 'Numeric'. You'll notice Superbase has marked this field with a plus sign and a hash sign. The plus reminds you it's a NUMERIC field and the hash tells you that at the moment there's only room for one digit.

**SETTING DECIMAL PLACES**

In the top right hand corner of the screen, the field length indicator reads 1,0. This means that the moment there's room for only one digit before the decimal point and none after it. But we want there to be enough room for three digits.

You need to make extra room. So move the cursor two spaces to the right and the indicator should read 3,0. Now press RETURN. You've increased the length to hold three digits.

The next field down is the 'Unit-Price' field. This one will hold the price per unit of the goods bought.

This one will also need to be a 'Numeric' field, so move the cursor until it's positioned one space after 'Unit-Price', and as before, press the 'f1' key, then the letter 'N'.

As this field will be holding amounts of money, it should be formatted for Dollars and Cents or Pounds and Pence. We'll format it so it will hold three digits before the decimal point, and two digits after. Move the cursor two spaces and you'll see the field length indicator reads 3,0. For the decimal point, press '.' then move the cursor another two places to the right and press RETURN. The field length indicator should now read 3,2. That's the format you want, three digits before the point and two after, so press RETURN.

**SETTING A RESULT FIELD**

The 'Total-Price' will be the result of multiplying Quantity by the 'Unit-Price'.

That makes the 'Total-Price' field a 'Result' field, so move the cursor one space after the words 'Total-Price'. Then press the 'f1' key and then the letter 'R' to make it a 'Result' field.

This field is also going to hold an amount of money so again we'll make room for four digits before the point and two after it. So move the cursor along three spaces, key in a full-stop, then move the cursor along another two spaces and press RETURN.

**SETTING A FIELD AS A CONSTANT**

The next field we must set up is the 'Tax' field. So move the cursor one space after the word 'Tax'. The rate of Tax will be the same on all our invoices so we can set this field as a CONSTANT. Press 'f1', then the letter 'C' for CONSTANT.

Now move the cursor along until the indicator reads 5. We need to allow 5 spaces including decimals. Now press RETURN. We'll imagine the rate of TAX is 15% and as a reminder, we can type this onto the screen next to the field.

Move the cursor along two spaces and key in 15%. That completes the TAX field so move on to the last field, the 'Amount Due' field. This will be a RESULT field to hold the result of the Total-Price plus 15% TAX.

So set it up as a 'Result' field. Move the cursor down so that it's positioned one space after the word 'Due' and then on your own, set this field as a 'Result' field with room for five digits before the decimal point and two after.

## **2.7 SAVING YOUR WORK**

You've finished setting the field TYPES and LENGTHS so you should save what you've done so far. Press 'f1', then the key marked RUN STOP and your work is saved.

## **2.8 ENTERING CALCULATIONS INTO THE FIELDS**

Now you must finish designing this form by entering calculations into some of the fields. Superbase will later use these calculations to automatically work out the totals on your invoices for you.

### **THE TOTAL PRICE FIELD**

Superbase is prompting you to 'Enter Calculation' and the 'Total-Price' field is highlighted. So this is the first field that needs to contain a calculation. This calculation will be 'Quantity' multiplied by the 'Unit-Price'.

Whenever you refer to a field name in a calculation or in a command, you must put square brackets around the field name in question. Right now we're going to multiply the contents of the 'Quantity' field, so key in:

**[quantity]\*[Unit-Price]**

Press RETURN and that's finished this first calculation.

### **THE TAX FIELD**

You'll now see that the 'Tax' field is highlighted. Superbase is prompting you to enter a calculation into this field. The 'Tax' in your case is fifteen percent, so key in the number 15 (you don't have to put the percentage sign, Superbase will treat it as a percentage). And press RETURN.

**THE AMOUNT FIELD**

Superbase now asks for a calculation to go in the 'Amount Due' field. This will be the Total Price plus 15% Tax, so key in:

**[TAX]/100\*[Total-Price]+[Total-Price]**

That completes the calculation in the 'Amount Due' field. Now press RETURN.

There are no more fields which need calculations, so you've finished designing the form. Superbase now asks if it should allow duplicate keys. This time we want each record to be unique, in other words to be able to call up one record at a time by the key field. So in answer to this question press 'N' for no. You then have to wait while Superbase processes the information.

**2.9 ENTERING RECORDS INTO THE INVOICE FILE**

You can now start entering some records into your newly created Invoice file, and as before, you start by pressing 'f1' just once to get the ENTER option from the menu.

Notice that the 'Tax' field is already filled in. That's because you set it as a CONSTANT at 15% for every record you enter. Now enter the first record.

Make this invoice number 100. Key in:

**100**

And press RETURN. Then key in the customer's name. Key in:

**Senders**

And press RETURN. Next the date. Key in:

**9sep83**

And press RETURN. Next the 'Goods' field. Key in the goods as:

**packets of envelopes**

And press RETURN. In the 'Quantity' field enter the figure 20 and press RETURN.

For 'Unit-Price', imagine that one packet of envelopes costs 99 cents, so key in the figure 0.99, using a period as the decimal point. Then press RETURN. You'll see that Superbase has automatically calculated the 'Total-Price' and the 'Amount Due' so this invoice is finished.

**SAVING THE RECORD**

You now need to save this record. You could just press RETURN to take the cursor down to the last field and then press RETURN again. But there's a quick way of saving a record wherever you are within the form, (as long as you've made an entry in the key field). You simply hold down the SHIFT KEY and press RETURN. Try it. You get the message prompting you to press RETURN to store, so do as the prompt says.

**2.9.1 ENTERING MORE RECORDS FROM THE TUTORIAL**

As you did with the name and address file, have a go on your own at entering some more example records. You'll find these records at the back of Tutorial 2.

When you've entered the last record, the one numbered 106, press RETURN instead of the space bar, and Superbase will return you to MENU 1.

**2.10 THE CALC OPTION**

Look down this menu and you'll see the option called CALC. CALC is very useful for performing all kinds of calculations. Here's an example. Select the CALC option by pressing 'f5' from MENU 1. And key in the following calculation:

**676\*2.3**

That means 676 times 2.3. Now press RETURN to get your answer. So you can use CALC as a calculator any time you like from the menu. Now press RETURN again to get back to MENU 1.

You can use CALC to change the contents of any field, but before you can do that, you have to get the relevant record into the computer's memory. We want to use CALC to update Invoice number 100, so get invoice 100 into memory.

**USING CALC TO UPDATE AN INVOICE**

First press the 'f2' key for the SELECT option, then 'f6' to get the 'First' record into the computer's memory. This first record is the one you want to perform the calculation on, so let's look at it to see what needs changing.

You'll see this record details a quantity of 20 packets of envelopes. We'll imagine you made a mistake and the customer really purchased 30 packets of envelopes. This is where the CALC option comes in very handy, try it.

Press RETURN to get back to the SELECT menu, and then RETURN again to get back to MENU 1. (Don't worry about the record you selected, it's still in the computer's memory.) Press 'f5' for the CALC option and you'll get a message prompting you to enter the calculation required.

We want the field called 'Quantity' to equal 30 instead of 20. So key in:

**[Quantity]=30**

Remember, the square brackets tell Superbase 'Quantity' is the field we want the calculation to work on. Now press RETURN.

Now press RETURN again. You won't see much happen, so use the SELECT option to check that Superbase has carried out the calculation. Press 'f2' for the SELECT option, then the letter 'c' to get the current record on the screen, that's the one you were last working on. You'll see that invoice number 100 now has a quantity of 30.

### **2.10.1 THE STORE COMMAND**

CALC does not save the record permanently unless you use a command called 'Store', try it now. Press RETURN twice to get back to the MENU 1 and type in the word STORE. Finally, to activate the STORE command, press RETURN.

Superbase proceeds to overwrite the original record of Invoice 100 and replaces it with your amended version. Just to prove that Superbase has saved a copy of the amended invoice, have a look at it. Press the 'f2' key for SELECT and then the letter 'f' for 'First', and the amended invoice 100 appears on the screen.

You've seen only two uses of the CALC option, but you'll discover further uses if you look in your Manual later on.

There's a simpler way to update records using the letter 'r' for REPLACE on the select Menu. We won't try the REPLACE option now, but you can try it yourself, when you've finished this Tutorial.

## **2.11 THE BATCH OPTION**

Now press RETURN twice to get back to MENU 1. Then press RETURN once again to get to MENU 2, and have a look for the BATCH option.

CALC is fine if you only want to perform calculations on one record at a time, but what happens if you want to change a value on more than one record in a file. Well that's where the BATCH option can help. Here's how it works.

Press the 'f3' key, that's the BATCH option. You get a prompt on your screen which reads:

**all/from 'list' (item list)**

The first part of this prompt is asking which records you want the calculations you're about to enter to be performed on.

You can tell Superbase to operate on 'all' the records in the file or just on those 'from' a separate list of records you've previously created and stored. You'll see how to create one of these lists later on. Right now we'll get the BATCH option to operate on 'All' the records in this file.

### **USING BATCH TO REDUCE PRICES BY 10%**

Imagine you decide to reduce all your prices by 10%. So in each record in the file, you'll have to find 10% of the 'Unit-Price' and then deduct that amount.

You key in the calculations in exactly the same way as with the CALC option. Key in:

**all [Unit-Price]=[Unit-price]-0.1\*[Unit-Price]**

Your calculation will overflow the first line and continues on the next line down, this is perfectly normal.

The calculation is saying, take the 'Unit-Price' and find 10% of it, then subtract that value from the present 'Unit-Price' and then put the answer back into the 'Unit-Price' field. Now press RETURN.

Superbase takes a while to work its way through all the records and when its finished, MENU 1 appears back on the screen.

### **CHECKING THAT THE BATCH CALCULATION HAS WORKED**

Superbase has made the BATCH calculation, so check that it's done as you asked. First call up the SELECT option, then press the letter 'f' for 'First'. If you now look at the 'Unit-Price' field on this first invoice, you'll see it's been reduced by 10% and now reads 36, that's 10% less than the original 'Unit-Price' of 40. Use the letter 'n' for 'Next', to look through the records and you'll see that each 'Unit-Price' has been reduced in the same way. And of course, Superbase has recalculated all the 'total-prices' and 'amounts due'.

The BATCH option differs from CALC in that it automatically saves each record that's amended, so you don't have to tell Superbase to store each one separately. Now get back to MENU 1 by pressing RETURN twice.

**2.12 SOME WAYS OF SELECTING RECORDS FROM THE FILE**

You've already seen how to SELECT records from a file and display them in the screen by using the MATCH option. But Superbase has some other ways of selecting records. First, there's the FIND option.

**2.12.1 THE FIND OPTION**

Look at the menu and you'll see it's function key number 3. The FIND option selects records in the same way as the MATCH option, but when you use FIND, Superbase keeps a list of the records you've selected and stores the list on your data disk for future use, for example, when you're printing or doing batch calculations. Try it now, press 'f3'.

You're going to sort out all those records where the Amount Due is more than 25 dollars.

**TELLING SUPERBASE WHAT TO SEARCH FOR**

Your screen should now be displaying a blank record. Superbase is waiting for you to key in the criteria on which to make its selection of records. Move the cursor down to the 'Amount Due' field and key in:

>25

And press SHIFT-RETURN. As Superbase works its way through the records it'll display each key field on the screen. It then compiles all those invoices owing more than 25 dollars into a separate list. Superbase stores the list on disk and then returns you to MENU 1.

**2.12.2 NAMING THE LIST**

Superbase automatically gives all lists of records the name "h8list" (or "hlist" if you're working in 40 column mode), unless you tell it otherwise. So it's a good idea to give each list you create a different name, then Superbase won't keep overwriting the previous list. But DON'T give a list the same name as a file or you'll cause an error.

You'll find the procedure for changing names of lists in your Reference Manual. That's something else you can try out on your own later.

**2.12.3 VIEWING THE RECORDS IN THE LIST - THE OUTPUT OPTION**

Now you need to be able to have a look at the records on the list. You can either display the records on the screen or you can get a print out on your printer. Either way you use the same option, the OUTPUT option.

Press the 'f4' key to get to the OUTPUT option. You'll get the prompt which reads:

**all/from "list" (item list...)**

Superbase is asking you whether you want to display All the records in the file or just those 'From' a selected list. We want to have a look at the records on our "h8list" (or "hlist" if you're using a 40 column screen), so key in:

**from "h8list"**

And press RETURN.

You'll see the records displayed on the screen, but this display doesn't include field names. Superbase allows you to change the way the records appear on the screen. Try it.

#### **2.12.4 CHANGING THE DISPLAY - DOWN AND ACROSS**

First press RETURN to get back to MENU 1. Then key in the word 'down' and press RETURN. The next time you use the OUTPUT option, this 'down' command will display the field names and their contents down the screen rather than across. As before, press the 'f4' key to get the OUTPUT option. Then key in:

**from "h8list"**

Press RETURN and this time the records are displayed one at a time and you'll see they're a lot easier to read.

To view the next record on the list simply press RETURN. Have a go on your own at looking through the rest of the records and you'll end up back at the menu.

Just now, you typed in the word 'down' to change the way Superbase displays the information (in Superbase terms, its 'Output Format'). That new setting, 'down', will stay in force until you tell it otherwise. When you DO want to change it back, you simply key in the word 'across'. So you can set the format to 'down' or 'across'.

#### **2.12.5 DISPLAY RECORDS ON SCREEN AND PRINTER**

You can control the OUTPUT option in much the same way. So far, you've used the screen to display information, but if you have a printer, you can get a printer output. To get a printed output you simply enter the command 'print' and to switch back to screen display, you enter the command 'display'.

You've just been looking at a list of all those people who owe more than 25 dollars and you've been viewing whole invoices.

But it would be useful to be able to view or get a print out on paper. Superbase has a way of doing this.

If you've got a printer, make sure it's connected up properly and that it has paper in. If you haven't got a printer, don't worry.

Key in the word 'across', which will save paper, and then press RETURN.

Next, if you have a printer, you can tell Superbase you want a PRINTED copy of the records. You'll need to key in the word 'Print' and press RETURN. But only do that if you want a PRINT-OUT!

Make sure you have MENU 1 on the screen and then press 'f4' for the Output option. Now key in the following instruction:

**From"h8list"@10&[customer]@25&5,2[due]**

(Type "hlist" if appropriate.) Don't press RETURN yet.

#### **2.12.6 SOME OTHER WAYS OF CONTROLLING THE OUTPUT**

You're probably thinking that looks a little complicated. That's because, in this string of instructions, we've included a few new examples of other ways in which you can control the output with Superbase, as follows:

**@10**

gives the position at which the printing will start, in this case it's column 10.

**&[customer]**

tells Superbase to remove any trailing spaces, but it can have another use, as you'll see in a moment.

**@25**

sets column 25 as the print position for the next bit of information.

The next ampersand means something different. This one is not in front of a 'Text' field, it's in front of a 'Result' field. The bit that reads 5,2 sets the number of digits in the result, in this case, it's 5 places in front of the decimal point and 2 places after.

Those are just a few of the ways you can control the output with Superbase. You can find out more from the Reference Section 8 after you've finished this tutorial. Now to see what effect this command will have, press RETURN and you'll get a screen (or print out) showing customers and amounts due.

**2.13 A FINAL SUMMARY**

During this Tutorial, you've seen how to create databases and files, you've used the FORMAT option to design layouts and you've seen how to store records. You've recalled records and printed them out, and you've had a go at performing calculations.

We didn't have time to cover every single Superbase facility, there are a few Menu options and some commands we haven't covered. But you should now have enough knowledge to use Superbase on a day to day basis AND use your Manual to explore the full power of Superbase. Once you're confident with the material you've covered so far, move on to Tutorial 3.

You can now go back over any of the areas you're not sure of and have another go. But if you've finished, then close down the system by pressing RETURN to get back to MENU 1. Then key in the word 'quit' and press RETURN. Your screen now returns to the Commodore display. So you can take the disk out of the disk drive and switch off your computer.

Don't forget to fill in and send your Precision Software Registration Card which comes with your Superbase Package. You'll then be registered as a Superbase user and be able to get backup disks, as well as details of Precision Software's Superbase applications and other productivity software packages.

**INVOICES - FORM DESIGN - TUTORIAL TWO**

Key in the invoice form as shown below. Remember, press 'f1' and then the appropriate letter for each field, i.e. K for a key field, D for a date field, T for a text field. The length of each field is also given below.

Some of the fields below have not been given a field type and length, so for these fields, just key in the field title and we'll put the field types and length in later on.

You will notice that two of the field names in our example have hyphens. This is because each field name must be unique. If you want to know more about 'field names' look in Reference Section 4.

Resume the tutorial when you've keyed in all fields.

\*\*\*\*\*

INVOICE NUMBER                      (Make this a KEY field, 6 characters long.)

Customer                              (Make this a TEXT field 20 characters long.)

Date                                    (Make this a DATE field.)

Goods                                  (Make this a TEXT field 20 characters long.)

Quantity

Unit-price	)	Remember the hyphens! Without them, Superbase will think both fields have the same name, 'Price'.
	)	
Total-price	)	

Tax

Amount Due

**EXAMPLE RECORDS FOR YOUR INVOICES FILE - TUTORIAL TWO**

Here are six more example records for you to enter into your invoices file. Enter them in any order and Superbase will sort and store them in numeric order of the invoice numbers.

Start the tape again when you've entered all six records. Enter them all as we shall be using them later.

Note: dates can equally be in the format 'sep1283'

Inv No: 101  
Customer: D. Signer  
Inv date: 12sep83  
Goods: graph paper packs  
Quantity: 23  
Unit-price: 2.75

Inv No: 104  
Customer: R. Jones  
Inv date: 1nov83  
Goods: A4 paper pads  
Quantity: 100  
Unit-price: 3.99

Inv No: 102  
Customer: H. Old  
Inv date: 20sep83  
Goods: box paper clips  
Quantity: 34  
unit-price: 1.60

Inv No: 105  
Customer: T. Smith  
Inv date: 21nov83  
Goods: spiral binders  
Quantity: 6  
Unit-price: 1.99

Inv No: 103  
Customer: I. Pearce  
Inv Date: 2nov83  
Goods: stapler  
Quantity: 1  
Unit-price: 3.99

Inv No: 106  
Customer: T. Page  
Inv Date: 12dec83  
Goods: A1 pads  
Quantity: 25  
Unit-price: 1.99

**3     TUTORIAL THREE**

**3     INTRODUCTION**

This is a more advanced tutorial than the previous two, so you should be prepared to take more than one session to complete it. The tutorial begins with two sections on topics that everyone needs to know about: how to escape from functions, and the rules about changing disks, naming files, and backing up disks. Then we look at the MEMO Option, and look again at the FORMAT and OUTPUT Options. A section on COMMANDS and PROGRAMS is followed by one on SORT. The tutorial ends with a look at the Superbase REPORT generator.

Begin the tutorial by loading Superbase in the usual way. Insert your training disk, and select the "TRAINING" database and then the "invoices" file. Now work through the sections that follow. Some expect you to do some keying in, others are just for reading.

**3.1   ESCAPING FROM COMMANDS**

From time to time, everyone selects a command by mistake. You need to know how to get out of every command quickly.

Simply hold down the CONTROL key and press Q. If Superbase is processing and CONTROL-Q doesn't work, press the STOP key.

CONTROL-Q is not recognised by many commands in the Plus/4 version of Superbase. For a list of alternative keys (which will also work with Superbase 64 and 128) see Appendix H.

In BACKUP and NEW DISK you can quit by entering 'n' to the Are You Sure?' prompt.

To gain confidence, try out some of these escapes for yourself before doing any more of this tutorial.

**3.2   DISKS, DATABASES, FILES, AND LISTS**

As you get more used to Superbase, you'll find that you want to set up several different databases. Probably you won't be able to keep all your data on one disk. This section explains the relationships between disks and databases and gives more information about files and lists.

**3.2.1   THE DATABASE COMMAND**

Whenever you change from one database to another, you MUST give the database command before selecting a file.

If you enter the name of a database that doesn't exist and if you keep responding 'n' to the 'Create it?' prompt, you may get stuck in a 'loop'. Give the name of a known database, such as 'TRAINING', to escape, or simply press CONTROL-Q.

As with all the menu options, you can give the '**database**' command on the Command Line, as in:

**database"training"**

Finally, look at your disk directory from the MAINTAIN sub-menu. You'll see the database "TRAINING" in upper case letters so you can tell it from other files.

### **DATABASE NAMES**

A database name can be up to 16 characters long. Don't put any double quotes ("), colons (:), or numbers in the name.

### **DATABASE COPYING**

You can make a backup of a disk with a database on it. But you can't COPY a database with the file 'copy' command. Data transfer between databases is done with the 'export' and 'import' commands. See Reference Section 15.

### **3.2.2 DISKS AND DATABASES**

You **must** give the '**database**' command whenever you change disks.

To change from a database on one drive to a database on the other, for example from 0 to 1, give the database command in the form

**database "test",8,1**

The '8' is the 'device number' of a floppy disk unit, and the '1' signifies the drive number. See Programming Section 2.1.13 for further details.

You can have as many databases on a disk as will fit. You cannot have a database continuing from one disk to another. If your data needs to be divided between disks, you'll have to set up more than one database.

Running out of disk space can cause serious problems, so keep a watchful eye on the amount of disk space you have left. Use 'f5' Directory on the Maintain menu. The number of blocks free is shown at the end of the directory. Make sure that the number of blocks free is at least 80-100 blocks. There is a utility for dividing databases, see Appendix E.

**3.2.3 FILE NAMES / RECOVERING A LOST FILE**

The actual data in a database is stored in the database file. This is quite separate from a file format, which is an ordinary disk file. Among the advantages of this system is the ability to COPY a file format ('file definition' is another common term) and use it in more than one database.

You need to know that the file format can be lost by accident. This happens if you use the name of an existing file format when specifying a key list or a data file created with 'export' or 'output to'.

**RECOVERING A LOST FILE**

Recovering is quite easy. First give the key list or data file another name - use the file 'rename' command from MAINTAIN OTHER (see Reference Section 15). Then use FILE and FORMAT to set up the lost file format exactly as it was, under its original name. You'll find that you can get at your data.

**PRINTING YOUR FILE FORMAT**

This should help you see why it's a good idea to keep printed copies of all file formats. Print them with one of these command lines:

<b>print:status:display</b>	(C128)
<b>print:maintain status:display</b>	(C64/+4)

Finally, remember that database filenames can be up to 10 characters long. If you make one longer it will be cut short.

**3.2.4 KEY LISTS / DISK FILES**

These are external to the database, like file formats and help screens.

A KEY LIST is a list of some or all of the key fields in a database file. Key lists are a very important part of the system, helping you to work with selected groups of records in many different ways. They are created by FIND or SORT. The functions that can use key lists are:

SELECT FROM	REPORT	SORT
OUTPUT	BATCH	EXPORT

Notice that SORT both uses and creates key lists.

You can ADD to a key list. This is done by placing a comma and the letter 'a' after the name in FIND or SORT:

**find "newlist,a" where [LASTNAME] is "=Jones"**

DISK FILES are quite different from 'database files'. They consist of data copied from the database, and are created either by 'output to' or 'export'. They are usually intermediate files that will be used by another function such as 'import', or another program.

Word processing often requires name and address data from a database. 'Output to' is the function to use. See Reference Section 8.11.

Disk files are plain sequential files, each line consisting of a number of characters (usually a field from a database file) followed by RETURN, usually with a RETURN on a line by itself separating the records.

If you give such a file a name beginning with 'h', such as "h8datafile," you can view its contents with the HELP Option.

Disk files and key lists may have up to 16 characters in their names.

Remember that if you don't explicitly name a key list, it will be named "h8list" or "hlist" by Superbase. But each new "h8list" or "hlist" overwrites the previous one.

### **3.2.5 BACKUP**

Do disk backups regularly to protect your data. See Reference Section 15 for details of how to make a backup copy of a data disk.

C64 and Plus/4 versions only: It's best to do backup at the end of each session, as the single disk backup function clears Superbase itself out of the computer.

If you have a dual disk drive, use the dual drive option given at the beginning of the backup operation.

## **3.3 CREATING A MEMO SCREEN**

Superbase has a function for writing documents for display or printing. This is the MEMO function, available from Menu 2.

Create a Memo Screen now.

### **SELECT MEMO**

Use the RETURN key to obtain Menu 2 and from it select the MEMO Option by pressing the 'f7' key. You will see a blank screen with the words 'Enter Memo Name' at the top of the screen.

Type 'diary' and press RETURN (a Memo file can have any name of up to sixteen characters). You will see the words 'Mode: Memo Writer' above the screen with the cursor in the HOME position.

**WRITE MEMO**

Type a few appointments onto the screen:

Monday 4th 11.00 dentist

Wednesday 6th 8.30 pm party at Sam's

Weekend - up at the lake

The full set of Memo editing commands is given in Reference Section 16. If your printer is connected hold down the 'Control' key and press 'P' to print out the current screen. When you are finished press the 'f1' key followed by the 'RUN/STOP' key to store the Memo. You can recall the Memo screen at any time from Menu 2 by selecting the MEMO Option and typing 'Diary' when prompted to 'Enter Name'.

**RECALL MEMO**

Try recalling the Memo screen in this way to prove to yourself that it is there, and when you are finished with it press 'f1' and then RUN/STOP.

A Memo Screen can be edited by recalling it by using the 'MEMO' command from the Main Menu, typing your amendments, and then storing it by using 'f1' followed by 'RUN/STOP'.

**3.3.1 HELP MEMO SCREENS**

You have already looked at examples of the special kind of Memo Screen called HELP SCREENS which can be obtained from either of the Main Menus by pressing either the 'f8' or HELP key.

You may want to redesign one of the Help Screens or to create a Help Screen for one of the sub-options for which there is no built-in Help Screen available, or for one of your programs.

This can be easily done by means of the MEMO Option. All you have to do is create a Memo Screen as normal, but prefix its name with an 'h8' (for an 80 column screen) or 'h' (for a 40 column screen.) For example, if the screen is to contain advice about the Invert Screen and Invert Line commands you encountered above, then call the Memo Screen you create "h8invert".

Having done this you can then obtain this advice by selecting the HELP Option and typing 'Invert' when asked 'Enter which help required'.

When you have gained some experience in programming Superbase, you can use the 'help' command in programs to display your own menu screens.

**3.4 CHANGING THE FILE FORMAT**

While you're getting to know Superbase, you'll probably want to make changes to the file formats you set up. This is quite easy, but to avoid unnecessary mistakes we include here explanations of a number of aspects of formatting.

Before you change a format, print out the file status as described earlier in this tutorial.

To change an existing format, first select the file with the FILE command, then press the 'f2' key on Menu 2 for the FORMAT Option. After a few seconds Superbase will display the field names and markers as they were when the file was being set up.

**3.4.1 CHANGING FIELDS**

Full details of the rules for changing field types are given in Reference Section 4. The main point is that you can only add or remove fields from the END of a format.

To change a field type, or set it as a FORCED FIELD, first erase it by positioning the cursor over either field marker and pressing 'f1' followed by 'E'. Now simply reposition the cursor where you want the new field to start, and set the field in the normal way.

**FIELD NAMES**

Field names can also be changed. First position the cursor on the field name and press 'f1' followed by 'E' to erase it. Then simply type in the new name using SHIFT-INST/DEL to insert extra space if necessary. (Note. If you have referred to a field by name in a program, you must change that too.)

But remember that field names should be unique, unless you want a piece of data to appear more than once in a record: such fields are called REPLICA fields.

Field names can be up to 12 characters long, counting rightwards from the first space to the left of the field name. For example, 'Amount Due' will actually have the field name of 'Due'.

**3.4.2 UNIQUE KEYS AND DUPLICATE KEYS**

A Superbase file may have either DUPLICATE or UNIQUE keys. The KEY to a record is, as you will remember, the field that is used for fast access to a record in a file. Usually it is best to keep every key in a given file UNIQUE: that is, not matching any other key in the file. Partially matching keys are fine, and may come in very useful in some situations. And it is also all right to give the same key to records in different files.

DUPLICATE keys are keys within the same file that are exactly the same. Suppose you want to use peoples' names as keys - as in the Addresses example in Tutorial 1. If you know two 'Smiths', you can only give the records for the two individuals the same key if you have previously replied 'y' to the 'Allow Duplicate Keys?' question at the end of the FORMAT Option.

The major drawback of duplicate keys is that when retrieving records for display or printing it can be difficult to access the second record or subsequent records with the same key as another.

You can overcome this difficulty by adding a new field to the end of the record. You change the old key field to an ordinary text field, and then make the new field the key. You put into the key field a CODE based on the name of the person, combined with a unique number if necessary, for example, 'SMI/01' and 'SMI/02' for two different people named Smith.

Remember that you can use the FORMAT Option and leave a file unchanged except for changing the reply to the 'Allow Duplicate Keys?' question at the end of the process.

#### **3.4.3 USING NUMBERS AS KEYS**

Superbase key fields are text fields. You can put numbers into a key field, but each digit will be treated as a separate character when it comes to storing the record in order. This means that records with the keys '1', '2', '10', '110', and '200' would be stored in the order

1	10	110	2	200
---	----	-----	---	-----

The solution is to type in enough leading zeros to make all keys the same length:

Maximum 9999	Maximum 999
0001	001
0002	002
0010	010
0110	110
0200	200

#### **3.4.4 MULTIPLE SCREENS**

Superbase records can include up to four separate screens. So far in the tutorials we have been working with single screen records, but we shall now add an extra screen to the invoices file.

**REDEFINING THE FILE FORMAT**

To do this we will have to edit the screen format again. Make sure that you are in the "TRAINING" database and have selected the "invoices" file. Press RETURN to get Menu 2 and then 'f2' to select the FORMAT option.

**OBTAINING EXTRA SCREENS**

Once you have the Invoice Format on the screen, press 'f1' and then the '+' key. The message 'Forward Screen' will indicate that you have moved forward to a new screen. This is how you obtain an extra page for your records.

The '+' command can also be used in the SELECT Option. The key '-' is used for returning to the previous screen.

**TEXT FIELD**

On line 2 of the new screen, type the field name 'Paid', move the cursor along a couple of spaces and set a TEXT FIELD by pressing 'f1' followed by 'T'. Make the field three characters long to allow you to enter 'yes' and 'no', and then set the end of the field by pressing RETURN. Then press 'f1' followed by STOP to finish the format, and end the function as you have done before. Check the results with the SELECT Option, using '+' and '-' to move from screen to screen.

**3.5 OUTPUT / WORD PROCESSING LINK**

You've already seen how to use the OUTPUT command to produce printed or displayed lists of selected items from your data. In this section we go over the different forms of output and the output formatting commands. Then we look at how to produce a file from the database for use in word processing.

**3.5.1 OUTPUT COMMAND**

OUTPUT can be combined with the following options:

<b>all</b>	<b>or</b>	<b>from "list"</b>
<b>across</b>	<b>or</b>	<b>down</b>
<b>display</b>	<b>or</b>	<b>print</b>

Here are some examples using the single field [customer]:

```
output display all across [customer]
output print all down [customer]
output display from "h8list" down [customer]
output print from "newlist" across [customer]
```

These are full command lines. In practice you won't need all this detail every time, since some items remain set after being included. These are the 'print' or 'display' selection, and the 'down' or 'across' selection. And you can give the initial output command with the 'f4' function key (on Menu 1), following it with the rest of the command on the command line.

**3.5.2 OUTPUT FORMAT COMMANDS**

These are the '@' and '&' signs, used before field names to position and control the form of the output. The numbers can of course be varied.

**@10 [customer]** displays the contents of the field at column 10.

**@10,4 [customer]** displays the contents of the field at column 10, row 4, each item on a new page or screen.

**@0** positions the cursor at line 1, column 1, without clearing the screen.

**@5,0** starts printing at column 5 on the next line. Use @1,0 to print a blank line.

**& [customer]** removes all trailing spaces from a text item.

**&5 [customer]** cuts the text item short to 5 characters.

**&4,2 [unit-price]** puts a number into the form of 4 digits before and 2 after the decimal point. Rounding is automatic. If insufficient places are allowed, '#' signs are displayed to indicate overflow.

If your printer allows underlining, these commands can be used with 'Output':

**@-[customer]** @- switches underlining on the 'customer' field and then off again.

**@+ [customer]** switches underlining on for the next text item only.

Here is an example of a fully formatted output command on the Command line:

**output print all across @10,5@-&12[customer]@25&5,2[unit-price]@-**

Try this for yourself, and then vary the row, column, and formatting values to see the effects. See Reference Section 8 for more details.

### **3.5.3 'PRINT' AND 'DISPLAY' WITHOUT 'OUTPUT'**

You can only use the output command options 'all', 'from "list"', 'across', and 'down' in the output command itself. This means that you must either press the 'f4' key or type 'output' before you use any of them.

However, 'print' and 'display' will work as first word commands with the output formatting commands described in the previous section.

See Reference Section 8.1.3.

### **3.5.4 OUTPUT TO DISK FOR WORD PROCESSING**

If you want to obtain a list of, for example, name and address data to be used for printing standard letters, you use the OUTPUT TO command. This is described in full in Reference Section 8.11.

The sequence of operations is:

1. Unless you want data from all records, use 'find' to produce the list of selected records you want to output.
2. Use **'output to'** to create a disk file containing the actual data.
3. Quit from Superbase and load your word processor, such as Superscript, Easy Script or Script/Plus.

4. Load or type the standard letter.
5. Output using the mail merge feature.

**Note.** If you use Superscript 128, you can use Superbase to carry out any word processing operation, including mail merge, with direct access to database files -- no intermediate disk file is needed. See Programming Section 1 for full details.

### **3.6 COMMANDS AND PROGRAMS**

We have explained that Superbase has a dual control structure, that it is both Menu driven and Command driven. You will now have had experience of both kinds of control.

Anything that you type on the lines at the top of the screen is a command. In fact, you can get at all of Superbase's options and facilities without the use of the MENUS at all.

#### **OUTPUT USING THE COMMAND LINE**

You shouldn't use capitals in the command words themselves, except when abbreviating as explained below.

From either Menu, type the Command Line 'output all the records' and press RETURN. This has exactly the same effect as pressing 'f4' to obtain the OUTPUT Option and then responding 'all the records' to the prompt.

A Command Line always begins with a Primary Command such as those which appear on Menu 1 and Menu 2, and is followed by one or more of the associated secondary commands and expressions as defined in Programming Sections 2 and 3.

#### **3.6.1 ABBREVIATING THE COMMAND LINE**

The words 'the' and 'records' and the spaces are not actually needed for Superbase to understand your Command Line but serve simply to make the syntax more natural. You could have entered the last Command Line simply as 'Outputall' if you had wanted to.

You could even abbreviate further to the first letter of the command with the second letter SHIFTED as in 'oUaL' or 'oUfR"hlist"[goods][unit-price]'.

The rule is that any command word can be reduced to the shortest possible unique form of the command, with a minimum length of 2 characters, with the last character SHIFTED.

You may want to try to obtain the results you achieved earlier via the Menus by using command lines instead. Try 'file "invoices": select first'.

### **3.6.2 COMMAND LINES WITH MORE THAN ONE COMMAND**

A Command Line may include more than one Primary Command if they are all in different CLAUSES separated by colons. For example, this was how we suggested you print out a copy of your file status: 'print : status : display' or 'print : maintain status: display'.

Two screen lines less one character position are available for multiple commands.

CALC in particular often requires you to use multiple commands. You can use:

**calc [unit-price]=500;[quantity]=25**

(note the compulsory semicolon) to change more than one field at once (use 'select c' if you want to see the results). But if you refer to a field more than once, including references in a result field formula, you must use colons to separate the commands:

**calc [unit-price]=500;[quantity]=25:calc[total-price]\*1.1:  
calc [total-price]**

The reason we cannot display these fields in a single CALC Command line itself is that field references within a single command are always to the original value of the field. Only by typing a full command line with colons to separate the individual commands to modify and then to display a field (or its derived result fields) can you achieve both modification and display in one operation.

### **3.6.3 REPEATING COMMANDS WITH THE 'RECALL' KEY AND STORING THEM**

To recall the last Command Line entered, we use the 'recall' key. This is the back-arrow key, located on the top row of the keyboard. The Command Line will be displayed at the top of the screen, either to be executed again (press RETURN) or to be modified (type modifications).

If you don't want to execute it but want to go back to the menu, use CONTROL-Q.

For a simple example, type the command:

**display "Customer is: "[customer]**

and press RETURN. After the display, press the back-arrow key. The command re-appears. Press RETURN again to execute it. You can repeat this a few times.

An extra reason for the 'recall' key displaying the Command Line instead of immediately executing it is not only so that you can modify it but also so that you can store it permanently. How this is done will be explained in the next section.

**3.6.1 LINE NUMBERS AND PROGRAMS**

You have seen how the current Command Line can be recalled and used again. However, once a new Command Line is typed in, the old one is lost.

To store a Command Line permanently for future use you must first give it a LINE NUMBER. And before that you should make sure that Superbase's program area is clear. Do this now by typing the command 'new'. Press RETURN. Now type in the command:

**output display all across @20[customer]**

Before you press RETURN, move the cursor to the beginning of the line, hold down SHIFT, and press the INST/DEL key 4 times. Then type the number 100, and press RETURN. No processing takes place, but that line is now treated as a program, and is stored in the program area. To view it, type 'list' and press RETURN. Any menu command can be placed in a program line in this way. When you're back at the menu, you can execute the program by pressing the 'f7' key. The EXECUTE option works on the current program if there is one, or asks you for a program name if there isn't.

Let's add one line to the program. Type the command

**200 display "program finished" : wait**

and press RETURN. Again type 'list' and press RETURN to view the program. Now execute it once more with the 'f7' key.

Lastly, SAVE the program for re-use later. Type

**save "myprog"**

and press RETURN. Superbase will put the program on the disk under the name "myprog.p". To get it back, you can type

**load "myprog"**

and press RETURN. Remember that programs are always saved from and loaded into the program area. Once a program is loaded you can add to it as we did above, or do more extensive editing using the PROG option from Menu 2.

**3.6.5 RUNNING TOTALS / USE OF VARIABLES**

This section is mainly for people who already know something about programming. To understand it fully you should study your Commodore manual, or other books on BASIC programming.

Superbase uses variables in the same way as BASIC. You can set a string variable by placing a '\$' after it, as in

**a\$ b\$ day\$ name\$**

String variables hold alphanumeric text characters. Numeric variables have no dollar sign, and hold actual numbers:

**t x product amt**

The value of a variable changes as a program runs. Among other things, variables make it possible to execute program lines in a loop, or repetition, thus saving much memory space in the computer.

In Superbase, a common use of a variable is in a RUNNING TOTAL batch operation. First, set the variable that will accumulate the total to zero:

**calc t=0**

Press RETURN. Nothing is displayed, but 't' is now labeled (brought into existence) and set to zero. Then give the batch command:

**batch all t = t + [due]; "total due is: ";t**

Press RETURN. This command goes through all the records in the "invoices" file, adding the contents of the [due] field to 't' and displaying 't' as it goes ('t' must appear at the end of the command if you want to display it).

When you're back at the menu, type the command:

**display t**

and press RETURN. The final value of 't' will be shown.

For more explanation of how to program Superbase, see the Programming Section of the Manual.

### **3.6.6 APPLICATION PROGRAMS**

Superbase's programming feature lets you design quite sophisticated applications when you're ready for it. The menus and the command line are stepping stones to more complex programs. An example of a Superbase program is the "labels" program that comes free on your disk. This lets you print labels in up to 4 columns. You can try it on ordinary paper with your "addresses" file. See Appendix E.

(If you want to interrupt the program near the beginning while it's still asking for input, use CONTROL-Q.)

**3.7 SORTING RECORDS**

The SORT Option gives you greater control over the order in which you output your records. You may want to FIND a group of records, SORT that group into a different order, and then output some details using the OUTPUT or DETAIL (see REPORT) commands.

From Menu 2 press the 'f4' key to obtain the SORT Option.

You will be prompted with:

**all/from "list" (item list....)**

Type in the response:

**all on [customer][due] to "sorted list"**

and press RETURN.

The message 'Processing' will be displayed. The record counter will change as you watch and after an interval you will be returned to the Main Menu.

Your response to the prompt signified that you wanted to produce a list of all the invoices sorted in order of the Customer name, and for those records with the same name, in order of the 'Amount Due'.

The list you have named "sorted list" will now contain the Keys of the records in this sorted order.

If you had included the characters 'D-' before the word 'on', the customer names and amounts would have been sorted in DESCENDING order, i.e. beginning with the alphabetically last one for text items, and with the largest one for numeric items.

Verify that the SORT has taken place. Use the OUTPUT Option to Display from "sorted list":

**output from "sorted list" across [customer] [due] [number]**

SORT is very important if you're producing reports with several levels of subtotal. You must do a sort on all the fields for which subtotals are required. Imagine a database containing a set of accounts for customers with data for type of business, county, region and city. To get subtotals for these, you would have to sort on all the relevant fields, in order:

**sort all on [type][county][region][city] to "sorted list"**

Then a Superbase REPORT can use "sorted list" to obtain subtotals for each group of customers in each city, in each region, in each county, for each type of business.

### **3.8 PRODUCING REPORTS**

We shall now conclude this tutorial by using the REPORT Option to create a report on how much each customer owes and has paid on the goods bought.

For a more comprehensive account of the powerful REPORT facility see Reference Section 13.

From Menu 1 use the 'f6' key to obtain the REPORT Option.

The REPORT GENERATOR is entirely prompt driven and constructs for you a Report Program which when executed will produce the report.

#### **REPORT FILE**

The first prompt is:

**Enter File to Report on**

Type "Invoices" (you must use double quotation marks) and press RETURN.

#### **REPORT TITLE**

You will then be asked:

**Enter Report Title**

Enter "REPORT ON GOODS SOLD" and press RETURN. Note that the title must be within double quotation marks. This is the heading of your report and will appear at the top of each page of the report.

In case you wanted a title larger than would fit in the Command Area you are asked:

**Any More?**

Type 'N' for no.

#### **TOTALS AND SUBTOTALS**

The next prompt is:

**Enter Total Calculation(s)**

These are the calculations required for Totals at the end of the report and subtotals during the report itself. They are entered in the same way as the total we specified in the BATCH option at 3.6.5 above.

You have ten special Total Variables, t0, t1.....,t9 and ten Subtotal Variables s0, s1.....,s9 to use in this option. Type:

**s0=s0+[total-price];t0=t0+[total-price]**  
(Use 'zero' not capital '0'!)

and press RETURN. You will be asked 'Any More?' in case the line of totals and subtotals is longer than can be fitted in the two line command area. Reply 'Y' as this is the case now. The next prompt is:

**Enter Total Calculation**

Continue by typing the rest of the calculation and press RETURN.

**s1=s1+[due];t1=t1+[due]**

You will be asked 'Any More?' again. This time reply 'N'. We have specified that we want to total and subtotal the total-price and the amounts due on all of the invoices.

**SUBTOTAL BREAK POINT**

After specifying totals and subtotals you will be prompted with:

**Enter Field for Subtotal Break**

Type '[customer]' and press RETURN. The field for subtotal break determines when the subtotals are to be printed. Each time the field specified changes its value we will get a subtotal. By specifying the 'Customer' field we will have a different subtotal for each customer. This is why SORT must be used before printing the report, to group customers' invoices together so as to produce meaningful sub-totals.

Next there is a prompt:

**Enter Subtotal Text**

This is the text you want printed with the subtotals to indicate what it is. Type the following (avoid spaces so as to keep the command on one line):

**@1,0"Total for customer:"@19s0@1"Total due"s1**

and press RETURN.

You will be asked 'Any More?' in case you want more subtotals, but since we do not, type 'N'.

**KEY ORDER OR USE A LIST**

A prompt will next be given to find out whether you want the records from the file in the order in which they are stored (i.e. their key order - this would be ALL the records), or those from a Key list. Although we do want to use all of the records we do not want the records

to be printed in key order. Instead we want to use the Key List "sorted list". So that the information is grouped according to 'Customer' and 'Amount Due' (see the section on SORTING records above), so respond with:

**from "sorted list"**

and press RETURN.

### **REPORT DETAIL**

Superbase next prompts you for the detail that will form the main substance of the report. The contents of these fields will appear for each record used in the report:

**Enter Report Detail**

Type

**@1,0 [customer] [goods] [due]**

and press RETURN. Again you will be prompted 'Any more?' Type 'N' as before.

### **END OF REPORT**

The final prompt for the specifications of the report itself is:

**Enter End of Report Text**

This is the information you want displayed at the end of the report including the final totals. Type

**@1,0 "Report Total-price"t0 @1 "Report Total due" t1**

and press RETURN. The 'Any more?' prompt is presented for the last time. Type 'N'.

### **STORE THE REPORT**

The program created by the Report Generator to your specifications will now be displayed and you will be asked:

**Save Report Definition**

Type 'Y' for 'Yes' and the following prompt will ask you what you want the Report Program to be called on disk:

**Enter Report Name**

The screen shows

**save "**

Type

**Sales"**

(ending with double quotation marks), and press RETURN. The Report Program will be stored for you to use at any future date.

If you wish to print the report type 'print' first to switch output to the printer. To print the Report itself press the 'f7' key from Menu 1 or type in the command 'Execute' directly from either Menu.

There may be errors in your Report Program. This is reasonable, since you have just used the Superbase Report Generator for the first time. However, you will wish to correct them so that the Report will run smoothly. The Program should look like this:

```
100 report "invoices"
200 title "REPORT ON GOODS SOLD"
300 total s0=s0+[total-price];t0=t0+[total-price]plus
400 s1=s1+[due];t1=t1+[due]
500 subtotal [customer]@1,0"Total for customer"@19s0@1"Total due:"s1
600 detail from "sorted list"@1,0[customer] [goods] [due]
700 endreport @1,0"Report Total-price:"t0@1"Report Total due"t1
```

Correct your Report Program so that it looks like this, using the PROG editing facilities. PROG is option 5 on Menu 2. You must then re-save the Program from the Main Menu by typing 'save "sales"' and pressing RETURN.

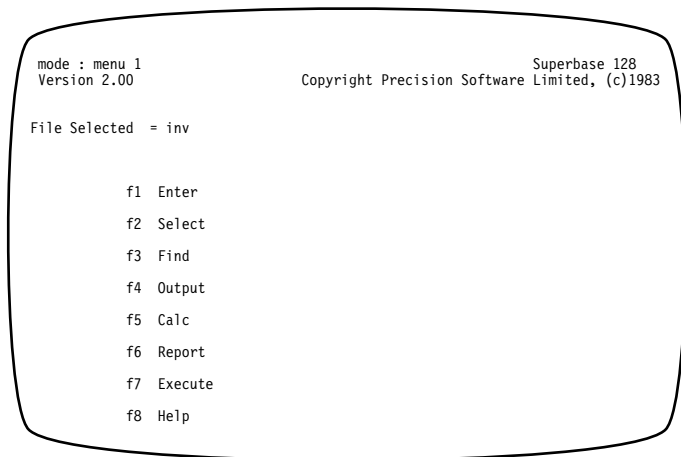


## **1     THE MAIN MENU - MENU 1**

### **1.1   USING MENU 1**

Superbase combines great sophistication with simplicity of operation. By means of the various Menus you can access all of the options provided in an easy and natural manner.

MENU 1 is your entry point into Superbase's most often used facilities, each of which is described briefly below beside the Function key used to obtain it. The Function keys are positioned on the right of the keyboard on a C64, above the numeric pad on a C128, or at the top of the keyboard on a Plus/4.



### **1.2   THE OPTIONS**



#### **ENTER**

This is the option used to **ENTER** information into your files. Deselect with CONTROL-Q. See Section 5.



#### **SELECT**

The **SELECT** Option is used to find and display any record stored in the database. It has a Menu of its own from which you can choose a wide range of facilities. Deselect with CONTROL-Q. See Section 6.



**FIND**

The **FIND** Option is used to find records matching a particular set of criteria. It stores a list of the keys of these records that can be used by the **OUTPUT**, **REPORT**, **SELECT**, **BATCH**, **SORT**, and **EXPORT** options. Deselect with **CONTROL-Q**. See Section 7.



**OUTPUT**

The **OUTPUT** Option is used to display or print information from all records or a previously selected list of records. Text (including headings) as well as the contents of fields can be output and BASIC variables and calculations included. **OUTPUT** can also create a disk file. Deselect with **CONTROL-Q**. See Section 8.



**CALC**

The **CALC** or **CALCULATE** Option is used to evaluate and/or display any expression. The full range of BASIC functions including trigonometrical functions can be performed. Results can be stored in fields or BASIC variables or simply displayed. Deselect with **CONTROL-Q**. See Section 11.



**REPORT**

This **REPORT** Option provides you with a full set of commands for producing printed **REPORTS** from the information in your files. Deselect with **CONTROL-Q**. See Section 13.



**EXECUTE**

The **EXECUTE** Option allows you to run previously designed **PROGRAMS** which can automatically perform whole sequences of operations on your files. Whole jobs can be carried out at the touch of a button. Deselect with **CONTROL-Q**. See Section 14.



**HELP**

The **HELP** Option is used to display a **HELP SCREEN** designed to give reference information and memory joggers for all of the major facilities offered by Superbase. You can also call up screens you typed in yourself using the **MEMO** Option. Deselect with **CONTROL-Q**. See Section 17.

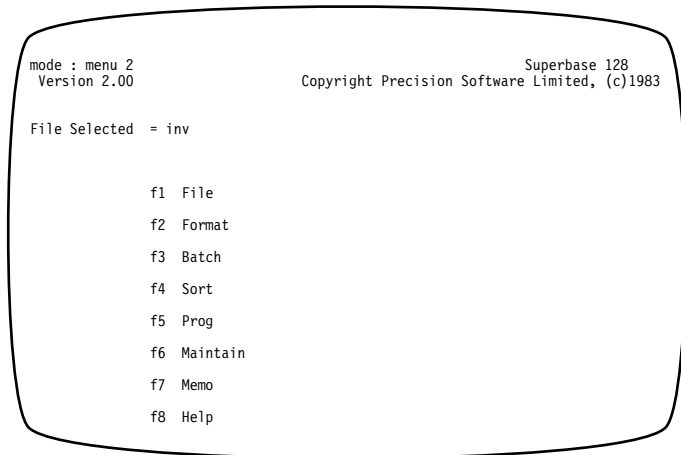


## 2 THE SECONDARY MENU - MENU 2

### 2.1 USING MENU 2

Menu 2 provides a range of further options, mainly those used less frequently. These options are also obtained with the Function Keys on the right of the keyboard, above the numeric pad, or the top of the keyboard depending on your computer.

Menu 2 is obtained simply by pressing RETURN from Menu 1, and vice versa.



### 2.2 THE OPTIONS



#### **FILE**

The **FILE** Option is used to change the current file you are working with or to create an entirely new file in the database. Up to fifteen files can be created for each database. Deselect with CONTROL-Q. See Section 9.



#### **FORMAT**

The **FORMAT** Option is used (1) automatically from FILE to define the screen layout of a new file, and (2) from the menu to revise the screen layout of the current file. Each file may have up to four screens formatted for each of its records. Deselect with CONTROL-Q. See Section 4.



**BATCH**

The **BATCH** Option is used to perform calculations using information from all or selected records in the current file. This is the option you use to carry out UPDATING operations on your files. Deselect with CONTROL-Q. See Section 12.



**SORT**

The **SORT** Option is used to sort all or a selected list of records by fields other than the KEY field (records are already stored in key order). The result of a Sort operation will be a file on disk of the record keys only (not the whole records) in the order of the Sort parameters. The original file remains unchanged. This list can then be used in the OUTPUT, BATCH, REPORT, EXPORT or SELECT options. Deselect with CONTROL-Q. See Section 10.



**PROG**

This is the Option with which PROGRAMS can be created and stored on disk. The **PROG** Option gives Superbase the power of an applications generator, providing a BASIC extended by all of the Superbase commands. Deselect with CONTROL-Q. See Programming Section.



**MAINTAIN**

This **MAINTAIN** Option provides you with a further menu of options to enable you to carry out various utility functions on your files including the EXPORT and IMPORT of data to and from other programs. Deselect with CONTROL-Q. See Section 15.



**MEMO**

The **MEMO** Option enables you to create screens of information which can be accessed later by yourself or by other users of Superbase. In particular you can modify or design HELP SCREENS by means of this option. Deselect with CONTROL-Q. See Section 16.



**HELP**



The **HELP** Option is used to display a HELP SCREEN designed to give reference information and memory joggers for all of the major facilities offered by Superbase. You can also call up screens you typed in yourself using the MEMO Option. Deselect with CONTROL-Q. See Section 17.

### **3      COMMAND LINES**

#### **3.1    USING DIRECT COMMANDS**

Although Superbase allows the user to control its many facilities by means of the two Main Menus, it is also possible to bypass the menu system of control and enter all Superbase commands directly.

Each of the commands included on the Main Menus can be entered together with a number of supplementary commands provided for the advanced user. See the Programming Section.

Commands are typed in the two lines at the top of the screen. They can be either entered individually or joined together with separating colons into a more complex COMMAND LINE. The maximum length of a command line is two lines less one character position.

##### **3.1.1   ENTERING COMMAND LINES**

**Commands must be typed in lower case (small) letters only, except for the last letter when abbreviating (see Section 8.10).**

From either of the Main Menus, although you cannot see a prompt or a cursor, you can just type a Command Line such as:

**select next:display [Lastname]**

which would obtain the next record in the file and display one field from it. Or you could type:

**find "chase-list" where [due] is ">100":output print the records from "chase-list"**

which would print out the records of customers who owe you more than \$100.

For the more experienced user of Superbase this method of control is a faster way of using Superbase's facilities since several functions can be carried out consecutively with no further intervention.

#### **3.2    REPEATING COMMANDS**

Whenever you execute a Command Line in this way, you can recall it without typing it in again with the back-arrow key at the top left of the keyboard (or SHIFT and '=' on the Plus/4).

This will result in the Command Line last entered being displayed at the top of the screen, ready for you to execute it again by pressing RETURN or for you to modify it.

The Programming Section describes how to insert line numbers and link Command Lines together to form a PROGRAM.

### 3.3 THE SYNTAX OF COMMAND LINES

A Command Line consists of a number of CLAUSES each of which is separated by a colon.

Each clause consists of a PRIMARY COMMAND followed by one or more SECONDARY COMMANDS.

The Primary Commands are commands such as those whose names appear on the Main Menus.

Other Primary Commands are provided for more advanced users and are detailed in the Programming Section.

The Secondary Commands are commands that must follow other commands and cannot be executed on their own, such as those which appear on sub-menus. They differ according to which Primary Command you are using but examples are as follows:

WHERE: Used with FIND as in:

**find "listname" where [city] is "=New York"**

FROM: Used with OUTPUT as in:

**output the records from "listname"**

or used with SORT as in:

**sort from "listname" on [due]**

ALL: Also used with OUTPUT as in:

**output all the records**

Each of the above Command Lines consist of only a single clause but several clauses can be strung together separated by colons as follows:

**find "listname" where [city] is "=New York":output the records  
from "listname"**

### 3.4 FURTHER SECONDARY COMMANDS

Further Secondary Commands and other uses of those above are detailed in the section relating to the Primary Command with which each is associated.

If you are new to Superbase we recommend that you use the menu system of control until you are familiar with the individual commands.

### **3.5 FILENAMES IN COMMAND LINES**

Many commands, when typed in full onto the command line or operated via system prompts, require filenames. All filenames must be enclosed in double quotation marks, although the final quotation marks may be omitted if it is the last character on the line.

### **3.6 INTERRUPTING COMMANDS**

The rules for interrupting commands are:-

Control-Q	quit or abort any function where the system is awaiting user response.
Run/Stop	stop any process or program where the system is active.
Control-Restore	unconditionally terminates any process, program or other function and returns to the Menu. This feature should NOT be used whilst the disk drive is active as files may not be closed correctly if unconditionally interrupted.

Plus/4 users see Appendix H for a list of alternative ways to interrupt commands on your machine.

**4      FORMAT****4.1    THE FORMAT OPTION**

Superbase stores information in files of records. A file may be a collection of invoices or business cards, or perhaps a collection of recipes for meals, depending on how the file has been set up by means of the FORMAT option.

Each of the individual invoices or business cards or recipes is stored on a separate record, which is put into the file by the use of the ENTER option (see Section 5).

Before the records can be entered, however, the FORMAT option must be used to design the layout of the records and to determine the type of information they are to contain. The FORMAT option can also be used to change the layout of an existing file. If you have created a record format and wish to change it, skip to '**Editing an Existing Record Format**' later in this chapter.

Type the Command Line 'print:status:display' (C128) or 'print:maintain status:display' (C64/+4) to obtain a printout of your file layouts. This can be useful if you need to reformat a file.

**4.1.1    OBTAINING THE FORMAT OPTION**

To obtain the FORMAT option, press the 'f2' key from Menu 2. You should see a blank screen with the words 'MODE: FORMAT' in the message area at the top left of the screen.

The one exception to this is that Superbase automatically puts you in FORMAT mode whenever you set up a new file.

**4.2    DESIGNING THE RECORD LAYOUT**

The screen is the 'blank page' on which your screen layout will be designed. Each record may have up to four such screens, each of which can have a different layout and contain different information.

A record layout consists of two basic types of item: FIELDS, the blank slots where information will be entered, and DESCRIPTIVE TEXT.

**FIELDS**

The maximum number of characters per record is 1107. This includes every character visible between field start and end markers for text, constant, key, numeric, and result fields; maximum of 5 characters for date and calendar fields; and a field separator between every 2 fields (i.e. the total number of fields minus 1). Note that a decimal point counts as 1 character, and every numeric and result field has an additional character reserved for the sign. Sign and decimal place

character positions are visible between field markers and are counted as part of the field length.

You should avoid quitting from a completely blank format without any field names. A blank format cannot be accessed if no fields are defined. Overcome the problem by reselecting the file and entering at least one field.

### **DESCRIPTIVE TEXT**

You may type any descriptive text onto the screen, the most important being FIELD NAMES. These are the names you decide to give to the fields, such as 'address1' or 'LASTNAME', which help make clear what the information in the fields is supposed to be. Other types of descriptive text include lines, borders, or text such as 'Business/Personal' which you typed to the right of the 'B/P' field in your 'address' file. Apart from FIELD NAMES themselves, up to 1000 characters of descriptive text is allowed.

### **FIELD NAMES**

These FIELD NAMES are of vital importance since they are the means by which Superbase keeps track of where the data belonging to the file is to be displayed.

More importantly, the field names are your means of referring to the information held in the fields throughout the file.

The field name can be of any length up to twelve characters and is always the nearest piece of descriptive text to the left of the field, regardless of how many spaces there are between the field and its name. Field names must be on the same line as the field start marker.

The name must all be on one line and must not contain any double quotes. If you were to type a field name containing spaces, such as 'TOTAL PRICE', Superbase would take 'PRICE' as the field name since it would be the last word to the left of the field. 'TOTAL' would be just a piece of descriptive text. To make 'TOTAL PRICE' different from 'PRICE' insert a hyphen between TOTAL and PRICE.

Superbase will also ignore the case that the fieldname is typed in. That is to say that 'ADDRESS' will be treated as the same as 'address' or 'Address'.

#### **4.2.1 FIELD TYPES**

There are nine types of fields, each corresponding to a different type of information you may wish to store in your records.

**FORCED FIELDS** A **FORCED** field differs from a normal field in that, during use of the ENTER option, you must enter some data into the field.

Any field can be a **FORCED** field, but the **KEY** field is ALWAYS a **FORCED** field.

**KEY FIELDS** These are the same as **TEXT** Fields except that they can be used to locate the record quickly in a key search (see section 6.2.1), and to keep the records sorted in alphabetical order. Every record must contain a **KEY** Field, which can be anywhere on any screen of the record. A **KEY** can be up to 30 characters but we recommend short keys for maximum efficiency. The **KEY** field is always a forced field.

**TEXT FIELDS** **TEXT** Fields are used for storing items of information like names, addresses or phone numbers, which consist of strings of letters or numbers mixed with letters or other characters such as hyphens. Examples would be '21 Highview Avenue' or '654 8989'.

**NUMERIC FIELDS** **NUMERIC** Fields are for storing numeric information such as prices, or other quantities of any sort and will not accept non-numeric characters. Examples are '12.75' or '2'.

**DATE FIELDS** These are for storing dates. They have a fixed length of seven characters; two each for the year and the day and three for the month. Dates must be entered in the form '10Jul84' or 'Jul1084' (either capitals or lower case). Superbase will calculate the day of the week from the date and display it in the message area at the top of the screen. Dates are valid within the range 1 January 1900 and 31 December 1999.

A **DATE** Field can, however, be set to be eleven characters long so that the day of the week can be displayed in the field itself. In such a case you would enter the date as above and

leave the day of the week for Superbase to calculate and display automatically. To achieve calculations, date fields have numeric characteristics -- see the Programming Section, DATE command.

**CONSTANT FIELDS** **CONSTANT** Fields are used for holding information which is the same throughout the file of records. An example would be a field for storing the TAX rate in a file such as the Customer Invoice file in Tutorial Two, or a field for storing PI or some other constant in a scientifically oriented application. The initial value to be held in a **CONSTANT** Field is set during the formatting process and this will appear as a default value in each record during data entry.

If the value in the **CONSTANT** Field is altered by reformatting the file, then all subsequently created records will display the new value as a default, leaving the old value unchanged in previously existing records. Thus if there were a change of tax rate then new records would contain the new tax rate but records entered while the old rate was in effect would continue to display the original value. To change the constant field in old records use the BATCH option.

When entering data the **CONSTANT** Field can be overwritten for the current record.

**RESULT FIELDS** These are **NUMERIC** Fields which are to contain a value dependent on the value of the contents of other **NUMERIC** Fields within the record. A formula such as '[PRICE]\*[MARGIN]' is specified, where the square brackets with names enclosed denote the fields referred to by those names and '\*' signifies that we want the contents of these two fields to be multiplied together. In such a case the result field would contain the mark-up value for the product.

Superbase will automatically calculate the value to be stored in a **RESULT** Field without any further intervention by the user. If the values in the fields specified in the formula are altered, Superbase will readjust the value it holds in the **RESULT** Field.

You may use any constant or BASIC function in your formula, and by using parentheses the formula can be made more complex (e.g. '([PRICE]\*0.15) + [PRICE]' which would increase the price by 15%). Square brackets around field names as in the above example signify the contents of that field.

Note that field names must be full and complete. Up to two full screen lines (less one character position) can be used to describe the calculation as it is entered at the top of the screen (the last example above uses 22 characters). Field names used in a calculation can be anywhere in the record and on any screen. **RESULT** Fields can make use of other **RESULT** Fields, so if 79/159 characters are insufficient, intermediate **RESULT** Fields can be created on screens that are not normally viewed.

It is important to bear in mind that you may even use the names of non-numeric fields in a calculation, in which case Superbase will pick up the first item of numeric information in those fields. In a **TEXT** Field containing the information '2 8amp Fuses' for example, the value '2' would be picked up in a calculation which refers to the contents of that field, as it would if 'Fuses, 2' were contained in the field. If the field contained '8amp fuses, 2' however, the value of the field would be taken as '8'.

It is not until after you have finished designing your record layout and have selected the 'End Format and Store' option that you will be asked to enter the formulae for your **RESULT** Fields.

**CALENDAR FIELDS** **CALENDAR** fields store dates like **DATE** fields, but are also like **RESULT** fields in that what they contain depends on calculations carried out on other dates within the record.

They could be used for the automatic calculation of regular appointments or of a deadline which is a fixed number of days from another date in the record.

**REPLICA FIELDS**

You may use the same field name more than once in the same record layout. If you do so, the contents of the first occurrence of the field will be duplicated in all subsequent occurrences of the field. Occurrences after the first cannot be edited. Under program control you may therefore ensure that data is protected by creating a field on Screen 0 and making Screen 1 (which would contain an uneditable copy of the original field) the default screen. Moreover, the length of the field and consequently the number of characters displayed may be varied for each copy. This allows you to display only a limited portion of the field contents; but you must always display from the first character rightwards.

Each type of field is marked by a start and end marker. When you are formatting a file, you will be giving each field a name and perhaps giving your layout a title and explanatory comments. Do not use the field start and end symbols in field names or descriptive text.

**4.3 SETTING THE FIELDS**

The previous sections have described each of the field types and what they are for. In the sections to follow you will be shown how to set fields of each of these types in your record layouts.

You will see that to set the start of any of the fields, you can use the 'f1' key followed by the first letter of the field type to be set (e.g 'f1' + 'T' for a Text field).

You will notice that each field type has a different field-start symbol, but there are only two field-end symbols.

If the stripes run diagonally from bottom left to top right, then the field-end marker signifies the end of a normal field. But if the stripes run diagonally from bottom right to top left, then the field-end marker signifies the end of a FORCED FIELD.

When the cursor is placed over either the field-start or field-end marker, the field type is indicated at the top of the screen.

#### 4.3.1 SETTING A FORCED FIELD



Any field may be set as a **FORCED** Field simply by pressing SHIFT/RETURN instead of RETURN when setting the end of the field. This will force the user to enter data into the field before continuing to enter data into subsequent fields. Very useful where a field contains vital information which might otherwise be left out.

Note that KEY Fields are always FORCED fields. In this case you may press either RETURN or SHIFT/RETURN to set the end of the field. CONSTANT Fields are not forced, but are automatically filled with the constant value when records are being added to the file.

#### 4.3.2 SETTING A KEY FIELD



Having typed a field name such as 'NAME' you may set the start of a **KEY** Field by pressing the 'f1' key followed by the 'K' key.

The message 'Set Key' will appear in the message area at the top left of the screen and a small rectangle will appear just before the flashing cursor. You will also see the number '1' in the right hand message area at the top of the screen.

The small rectangle signifies the position where the KEY Field starts and the number signifies the current length of the field.

After choosing the length of the field (perhaps 15 to allow for names up to fifteen characters long) press the cursor right key at the bottom right of the keyboard to move the cursor along to the end of the key field. The maximum length of a key field is 30 characters.

**Note:** The key should be as short as possible while allowing each key to be unique. The shorter your keys the faster your record access times will be, and the less disk space will be occupied by the file index.

You will notice that as the cursor moves along the line the number at the top right of the screen will increase to show you the current length of the field.

You may also use the cursor left key to reduce the length of the field.

Once the field is the size you require (check the length count) press the RETURN key to set the end of the field.

Note that field lengths can be easily changed at any time without loss of data, so choosing a field length at this point does not limit you in any way. However, once information has been entered into the Key field, it should not be shortened to less than the maximum used length in any record. REPLACING a record that has had its Key field shortened so that it becomes the same as another record might result in damage to your records.

#### **4.3.3 SETTING A TEXT FIELD**



To set a **TEXT** Field press the 'f1' key followed by the 'T' key.

You will see another field-start marker appear on the screen, this time a small square. The message 'Set Text' will be displayed in the left-hand message area, and the number '1' on the right to indicate the field length.

Use the cursor right key again to set the length that you require. Check the number of characters with the counter in the right of the message area and press RETURN to set the end of the field. The maximum length of a Text field is 255 characters, so the field start and end markers may be on different lines. You cannot start a field on one screen and end it on the next.

#### **4.3.4 SETTING A DATE FIELD**



A **DATE** field can be set by positioning the cursor where you want the date to be shown on the record and pressing the 'f1' key followed by the 'D' key.

The word 'Date' will appear in the left-hand message area and the cursor will jump 7 characters to the end of the field. Press RETURN to mark the end of the field.

Unlike other fields mentioned above, the DATE Field will be set to a fixed length of seven characters, the minimum length for a date field, unless you use the INSERT key to extend it to eleven characters. The automatic calculation of the day of the week can then be displayed in the field. DATE fields are stored in numeric form using up to a maximum of 5 characters.

#### **4.3.5 SETTING A NUMERIC FIELD**



A **NUMERIC** field, typically one that is to contain a money amount, can be set by pressing the 'f1' key followed by the 'N' key.

The message 'Set Numeric' will appear in the message area. The field-start marker is immediately followed by the characters '+#'. The '+' marks the position of the plus or minus sign.

This time there will be two numbers separated by a comma in the right-hand message area. The first represents the number of digits before the decimal point, the second represents the number of digits after the decimal point.

A NUMERIC Field may contain up to a maximum of nine digits. There may be up to nine before and up to four after the decimal point. A numeric field that is to contain financial information such as a price will normally have anything up to seven digits before and two digits after the decimal point. Numbers are rounded automatically when necessary, but it should be noted that the results of rounding can be unpredictable when very large numbers are involved, due to limitations in the computer's way of doing arithmetic. However, only very small fractions are involved.

Note that although a NUMERIC field contains a sign for plus or minus, it will not contain a currency sign such as '\$' or '£'. The currency sign can be included with the descriptive text immediately before the start of the field.

#### **4.3.6 SETTING A RESULT FIELD**



To set a **RESULT** field press the 'f1' key followed by the 'R' key. You will see the message 'Set Result' at the top of the screen. The format of a RESULT field is like that of a NUMERIC field.

A RESULT field is set in the same way as a NUMERIC field. The formula for the Result field is specified at the end of the FORMAT operation.

A combined maximum of 32 RESULT, CONSTANT, and CALENDAR fields may be used in a record format. When setting a calculation, you cannot refer to the field itself.

#### **4.3.7 SETTING A CONSTANT FIELD**



You can set a **CONSTANT** Field by pressing the 'f1' key followed by the 'C' key. The message 'Set Constant' will be displayed at the top of the screen and you may use the cursor control keys to set the size of the field in the same way as for a TEXT field. The maximum size of a Constant field is 30 characters. The contents of the Constant field are specified at the end of the FORMAT operation.

#### **4.3.8 SETTING A CALENDAR FIELD**



Set a **CALENDAR** Field by pressing 'f1' followed by SHIFT 'C' key. Apart from the message 'Set Calendar' the process is the same as that for setting a DATE field above. The formula for the Calendar field is specified at the end of the FORMAT operation.

### **4.4 FURTHER FORMAT COMMANDS**

There are a number of further commands which you can use while creating a record format. They can be used to increase the number of screens in the record, enhance the appearance of the records, and make them easier to read.

#### **FORMATTING ADDITIONAL SCREENS**



When you have finished formatting your current screen you can obtain a new blank screen to format by pressing the 'f1' key followed by the '+' key. Up to four screens are available for each record layout. The same command is used to move forward to an existing screen.

#### **MOVING TO THE PREVIOUS SCREEN**



While formatting a multiple screen record layout you can move to the previous screen by pressing the 'f1' key followed by the '-' key.

**ERASING A LINE OF DESCRIPTIVE TEXT**



A whole line can be erased by positioning the cursor anywhere on the line to be blanked other than on a bold marker, and pressing the 'f1' key followed by the 'E' key.

Used in this way, the ERASE command will only erase lines of descriptive text such as field names and other text entered onto the screen during formatting.

**ERASING A FIELD**



If you want to remove fields from the record you must position the cursor over the field-start marker or the field-end marker before pressing the 'f1' key followed by the 'E' key.

Used in this way, the ERASE command will only erase the field, NOT the descriptive text.

**DELETING A LINE**

This command differs from the ERASE command. The latter will replace an original line of descriptive text with a blank line, but the DELETE command will remove it from the record altogether, shifting all subsequent lines up to fill the gap.

Note that you cannot use this command to delete lines which have fields on them. This is to ensure that you do not remove fields accidentally.



Just press the 'f1' key followed by the 'INST/DEL' key at the top right of the keyboard.

**INSERTING A LINE**

Similarly, a new line can be inserted into the record, shifting all subsequent lines down to make room.



Position the cursor at the start of the line before which you want the new line to appear, and press the 'f1' key followed by the 'INST/DEL' key with the 'SHIFT' key held down.

**INVERTING A LINE**

You can invert a line of the record so that it appears light on dark rather than dark on light. This has the effect of highlighting the line you have inverted.



Position the cursor on the line to be inverted and press the 'f1' key followed by the 'I' key.

To start entering text in reverse video at any point, hold down the CONTROL key and press 9. Then type. CONTROL-0 switches the feature off.

**INVERTING THE SCREEN**

If you prefer, you may invert the whole screen so that the entire record appears light on dark.



Press the 'f1' key followed by the 'S' key.

You may reverse the effect of either Invert Screen or Invert Line by repeating the command.

**CHANGING THE COLOR OF THE SCREEN**

If you are using a color screen, these commands will allow you to change the colors displayed. Instead of having to do this each time you load a file, you can change the colors in format mode so that they will be stored as part of your layout. Then, each time you select a file it will be displayed in color.



Changes the color of all the text on the screen, rather like changing the color of the 'ink'. Sixteen colors are available by repeating the command.



Changes the color of the background, rather like changing the color of the 'paper'. Sixteen colors are available by repeating the command.



Changes the color of the screen border. Sixteen colours are available by repeating the command.



Changes the color of the top two lines of the screen.



Removes color. That is, it reverts the whole screen to its normal (monochrome) colors.



'Reverts' screen to original color. That is, to the screen colors with which the record was formatted.

When text is displayed in reverse on a colored background, the letters appear in the color set with CONTROL-2.

Note that if you wish to use an interface requiring memory, you may do so, but the user-defined colors may not be displayed.

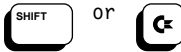
**DRAWING A BORDER**



You may wish to draw a box around the record or to draw a divider between two parts of the record. Press the 'f1' key followed by the 'B' key. You will then be asked 'Border Character?'.

Pressing a key at this point will enable you to draw repeated characters of a particular kind simply by using the cursor control keys.

The cursor will leave in its wake a stream of characters of the type you have chosen, until a key other than a cursor control key is pressed.



Note that many of the graphics symbols on the front of the keys are available if you hold down the 'SHIFT' or 'CBM' key while selecting the character to be repeated, but you cannot use the symbols used by Superbase as field-start or field-end markers.

Experiment with different keys until you find the symbol you want.

To erase a border that has been created in this way, simply select the BORDER option again but choose the space character as the border character and draw a border of spaces.

**SCREEN DUMP**

There are two ways of printing out the contents of a screen.



You can obtain a paper printout of the user section of the screen at any time by holding down the CONTROL key while pressing the 'P' key.



This command prints out the whole screen, including the top two Superbase lines. (C128 version only).

**QUIT FORMAT**



If you wish to escape from the Format Option at any point and return to Menu 1, you can do so with 'f1' then 'Q', or CONTROL-Q.

You may wish to do this if you have entered the Format Option by mistake. Any changes made prior to this command will not be saved.

**CLEAR FORMAT**



If you are totally dissatisfied with the format you have designed, you can clear the screen to start again by pressing the 'f1' key and then the 'Clear' key at the top right of the main keypad.

Note that only the current screen is cleared if you are working with a multi-screen file.

This is another key which requires the SHIFT key to be held down.

**END FORMAT AND STORE**



When you have finished formatting your record you can store the screen layout by pressing the 'f1' key followed by the 'STOP' key.

You will notice each of the field-start and field-end markers being replaced by angle brackets. If you have included RESULT, CALENDAR, or CONSTANT fields in your record format then you will be asked:

**'Enter Calculation?'**

Type in the formula you require and press RETURN (see earlier in this section for details of formulas in RESULT fields).

**DUPLICATE KEYS**

After specifying the calculations and constants you will be asked:

**'Allow Duplicate Keys?'**

We strongly recommend that you do not use duplicate keys so, if your application can be set up using a UNIQUE KEY type 'N' for 'no' in response to this question, otherwise type 'Y' for 'yes'.

A duplicate key file is a file in which more than one record can have the same KEY. In a file such as the Address Book file in Tutorial One, where the Key Field is NAME, there could be two separate 'John Smiths'.

The disadvantage of duplicate keys lies in accessing records with identical keys from a key list. Although a duplicate key such as 'John Smith' will appear on the key list as many times as there are records with that key, when the list is used only the first actual record will be accessed, over and over again, for each identical key encountered. This obliges you to devise a program using the SELECT NEXT command to obtain the second and subsequent 'John Smith' records.

We recommended that you always use UNIQUE KEYS. Only use DUPLICATE KEYS where there is an overriding reason for doing so. One reason might be to have the facility to obtain the most recently created record of a number of records sharing the same key, where records are records of orders, for example. If Superbase has to add a duplicate key, it inserts the key in the key index before any existing identical keys. This means that every search for such a key will retrieve according to the principle of 'last in, first out'.

You can access duplicate keys using the SELECT NEXT command from the menu or using a program. But, if you have duplicate keys and find it difficult to access records, and want to change them to unique keys follow these steps: Access the first duplicate key record. Next use 'a' for ADD to edit the key. Move your cursor to the field (eg Smith) and change its contents into a unique key, for example SMI/01. Save this record with its new key (using SHIFT RETURN) and with 'd', delete the old record.

You have to repeat this procedure for each duplicate key record - but remember to give them different keys (eg SMI/02, SMI/03). When you've processed the whole file, use FORMAT to select the unique key option for the file.

#### **4.5 EDITING AN EXISTING RECORD FORMAT**

If at any time you wish to change the format of one of your files you may do so without losing any data. You should always print a STATUS of the file before embarking on a file reformat operation. Type in: 'print: status: display' (C128) or 'print: maintain status: display' (C64/+4) and press RETURN.

FIELDS MAY ONLY BE REMOVED FROM OR ADDED TO THE END OF THE RECORD.

You can change the length of the fields or change the descriptive text. If you change a field name you must make sure that any references to the field in programs are also changed. Although field types may be changed, this should be done with caution as you may inadvertently cause damage to your data.



From Menu 2, obtain the FORMAT option by using the 'f2' key.

Move through the existing record format using the INST/DEL key to insert or remove spaces inside field markers. Add lines, borders, etc. as required.

Change field types with the following rules and guidelines in mind.

1. If a date or calendar field is converted to a non-date or calendar field, the date that was in it will be displayed as a number.
2. If you change a field to become a result or calendar field, the formula for the next following result or calendar field will be displayed for editing at the end of the operation. You must overtype the display with the new formula, and re-enter the old formula into the proper result field.
3. If you change a text field to a numeric field any text data in that field will be lost when the field is subsequently edited.
4. If you specify another field as the key field, the current key will be re-assigned to that field and whatever was in that field will be lost. This procedure is NOT recommended!
5. If you change a replica field to a unique field, you will create a new field. This will not affect the order of fields within the record. The new field will have no data in it. Likewise, converting unique fields to replica fields is permissible, but will result in the data from the original of the replica being displayed in that field.

Full examples of setting up record layouts are given in Tutorials One and Two.

#### **4.6 DUPLICATING A FILE FORMAT**

If you wish to create a file format in either the current or another Superbase database, there is an easy way of doing so.

Use the MAINTAIN OTHER Option to copy the file format information to a file with a new name (see Reference Section 15). The records themselves cannot be copied as they are stored in the Database file itself, which is listed in the disk directory in upper case letters.

Once you have created the new file format, enter its name after selecting the FILE Option, either in the current or the new database. Superbase will respond:

##### **File Does Not Exist: Create It?**

to which you respond 'Y', and the new format will be entered in the list of files. You can then modify it as you wish. If you need to transfer data to the new format, use either EXPORT or OUTPUT TO to create a sequential file, then IMPORT to load it into the new format.

#### 4.7 **SUMMARY**

The Format Option is used to set up screen layouts for the records in a file, or to modify already existing screen layouts.



You can obtain the **FORMAT** option to revise a screen layout by pressing the 'f2' key from Menu 2.

You are automatically put into the **FORMAT** option whenever you select an as yet non-existent file from the Database Catalog, either at start-up or while using the **FILE** option.



Sets a **Key** Field.



Sets a **Text** field.



Sets a **Date** field.



Sets a **Numeric** field.



Sets a **Result** field.



Sets a **Constant** field.



Sets a **Calendar** field.



Gives you the **next** screen to format for this file (up to four, numbered 0 to 3).



Takes you back to the **previous** screen.



**Erases** a line of descriptive text where the cursor is currently positioned.



**Erases** the field which has the cursor positioned over its field start or field end marker.



**Deletes** the line the cursor is currently on and moves up the subsequent text to fill the gap left behind.



**Inserts** a line just before the current cursor position and pushes the subsequent text down to make



**Inverts** the line of the screen where the cursor is currently positioned.



Switches **invert** mode on.



Switches **invert** mode off.



Inverts the whole **screen**.



Changes color of text on whole screen.



Changes color of background on whole screen.



Changes color of screen border.



Changes color of top two lines of screen.



Reverts color to monochrome.



Reverts screen to original color.



Enables the cursor control keys to be used to draw a line or **border** of characters of your choice.



Causes whatever is currently on the user section only of the screen to be **printed** out.



Causes whatever is currently on the screen to be **printed** out. (Superbase 128 only)



Enables you to **quit** the Format option and return to Menu 1. The Format is abandoned.



Clears the whole format on the current screen to enable you to start again.



Ends the formatting process, asks for calculations and constants, and stores the record format.

## **5 ENTER**

### **5.1 ENTERING INFORMATION**

Once you have formatted a file (see section 4) you can enter information into it.

With the ENTER option, you fill in a blank record with the information you want to keep, then add that filled-in record to the file.

#### **5.1.1 OBTAINING THE ENTER OPTION**



From Menu 1 press the f1 key.

You should see the blank record from the file selected appear on the screen.

Superbase uses the message area at the top of the screen to tell you:

**MODE: ENTER** that you are in the ENTER option.

**#1 k** that the field of the record you are about to enter information into is the first field, and is a **KEY FIELD**.

Note that the first field of the record need not have been a KEY FIELD. Every record must contain a key field but it can be in any position in the record.

You are now ready to fill in each item with the desired information. All characters are valid except double quotation marks ("), but we advise you to avoid the following, which are used by Superbase's record matching function: = # \* > < / & ?

#### **5.1.2 FILLING IN THE BLANK RECORD**

The following control keys allow you to move the cursor so that you can enter information where you want:



or






Moves the cursor to the right unless the cursor is at the end of the field, in which case it moves the cursor to the next field.








or








Moves the cursor to the left unless the cursor is at the beginning of the field, in which case it moves the cursor to the previous field.



 or  or  Moves the cursor directly to the next field, changing screen if necessary.



  or  Moves the cursor directly to the previous field, changing screen if necessary.


  Sends the cursor to the first field in the default screen. Any of the four screens can be set as the default screen by typing 'Screen n' from the Main Menu, where n is the number of the screen wanted as the default screen. This screen will be the first to be displayed when ENTER is selected.

   Clears all of the fields in the record and sends the cursor to the first field of the default screen.

  Enables you to exit from the ENTRY option without any information being stored. This is useful if you have selected the ENTRY option by mistake.

  Dumps the whole screen to the printer, including the top two lines used by Superbase for messages. (Superbase 128 only)

  Dumps the current screen to the printer to provide you with a hard-copy print out of the screen layout or the record you are currently entering.

 Moves the cursor directly to the next field unless the cursor is in the last field of the last screen, in which case it displays 'Press Return to Store' in the message area. Pressing RETURN again at this point stores the record, along with the information you have entered, into the file.



Displays 'Press Return' in the message area. Pressing RETURN will then store the record in the file. This command can be entered wherever the cursor is at the time. Cancel it by pressing any key other than RETURN.

If the KEY FIELD contains information identical to that in an already existing record, the message 'Key Already Exists' will be displayed in the message area together with 'Press Return to Continue'. The record you have just defined will NOT be stored. You will remain in the ENTER option so that you can type a new key.

To change an existing record you should use the REPLACE function of the SELECT option (see Section 6.2.10).

### **5.1.3 FORCED FIELDS**

If the cursor is in a field which has been set up as a FORCED FIELD (see Section 4), you will be prevented from storing the record until some information has been typed in all FORCED FIELDS.

The message 'Forced Field: Please Enter Data' is displayed in the message area at the top of the screen and the cursor is placed in the first empty FORCED FIELD.

### **5.1.4 DATE FIELDS**

In a field which was defined as a DATE Field during record formatting (see Section 4), Superbase will only accept entries of the form '18oct83' or 'oct1883' (upper or lower case). Superbase accepts single character dates so it is unnecessary to enter 04oct83 as 4oct83 will be accepted.

Attempting to enter a sequence of characters of any other kind will result in the message 'Invalid Date' being displayed in the message area, and the cursor will be prevented from moving to another field until a valid date has been entered.

Once a valid date has been entered, Superbase will calculate which day of the week corresponds to that date and will display that day in the message area at the top left center of the screen.

If the DATE Field is long enough (11 or more characters) Superbase will also display the day of the week in the field itself.

To enter a date a designated number of days forward, merely add days to the date and Superbase will calculate the valid date. For example, if the date is 12Jan and you require 20 days on, type in 32Jan.

See the DATE and CONVERT commands in Programming Section 2. Single keystroke input of a pre-assigned date may be achieved under Program control.

Full examples of the use of the ENTER Option are given in Tutorials One and Two.

### **5.1.5 ADDING, EDITING AND REPLACING, OR DELETING RECORDS**

ADD, REPLACE, and DELETE commands are provided on the SELECT sub-menu to facilitate data entry and record modification following record selection. See Section 6.

### **5.1.6 MODIFYING NUMERIC FIELDS**

There is an easy way of entering multiples of numbers into NUMERIC Fields. Suppose you know that your customer has bought 17 items at \$37.95. First enter 37.95 into the NUMERIC Field. Then place the cursor at the beginning of the field and type in '17\*' and press RETURN. The value stored in the field will be the result of multiplying 37.95 by 17.

You may modify NUMERIC Fields in this way using any of the arithmetic operators; '+', '-', '\*', or '/' provided space is available in the field for the additional character.

You cannot insert operators if the field is full.

If you want the number and operator to be after the field contents, as in '/6', you can use the DELETE key to pull the field contents over to the left so that you can fit in your number and operator.

This facility for modifying the contents of a NUMERIC Field is also available with the ADD and REPLACE Options in the SELECT Option.

## **5.2 SUMMARY**

The ENTER option is used to store information into new records in your file.



Selects the ENTER option from the Main Menu.



or



Moves the cursor to the right within a field, or if at the end of a field, moves the cursor to the next field.

## REFERENCE - ENTER



Moves the cursor to the left within a field, or if at the beginning of a field, moves the cursor to the previous field.



Moves the cursor directly to the next field.



Moves the cursor directly to the previous field.



Moves the cursor directly to the next field, or if in the last field, when pressed again will store the record in the file.



With the cursor in any position, displays 'PRESS RETURN' message. Pressing RETURN again stores the record in the file.



Moves the cursor to the first field of the default screen.



Clears all of the fields in the record and sends the cursor to the first field of the default screen.



Exits from the ENTER option without any data being entered.



Prints the whole of the current screen, including the top two Superbase message lines. (Superbase 128 only)



Prints the user section of the current screen.

## **6      SELECT**

### **6.1    SELECTING A RECORD**

The SELECT Option provides you with a number of ways of obtaining particular records from your file.

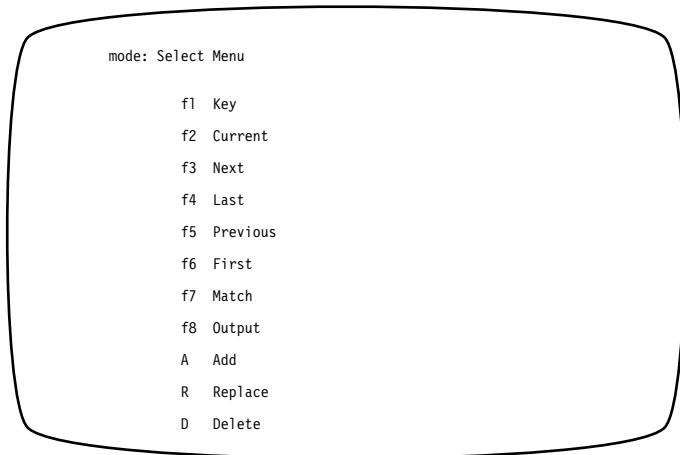
Note that records are stored in the file alphabetically according to the contents of the KEY Field of each record (see section 4). Numbers are treated as coming 'alphabetically' before letters.

#### **6.1.1   OBTAINING THE SELECT OPTION**



To obtain the SELECT Option from Menu 1, use the 'f2' key.

You should now see a further Menu, the SELECT Menu, with a number of sub-options listed on the screen. To return to this Menu after a SELECT operation, press RETURN. Press RETURN again for Menu 1.



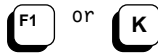
## 6.2 THE SELECT OPTION CONTROL KEYS

The various choices available to you at this point, and the control keys used to obtain them, are as follows. From the SELECT Menu or from any SELECT option use the function key selection or the initial character of the SELECT option.

### THE '+' AND '-' KEYS

Use '+' and '-' or cursor up/down keys to display adjacent screens (the number of the last screen used becomes the default until changed again). These keys are available whenever a record is displayed.

#### 6.2.1 KEY



This command enables you to search your file for a record with particular information in the **KEY** Field. On selection you will be asked: 'Key?'

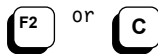
Type in the information you think is held in the KEY Field of the record you want and press RETURN. You may enter the full or partial contents of a Key Field for a full or partial match.

Superbase will then display either the record whose key matches your entry exactly, or the following record that contains the nearest key to the characters entered.

If a record is found with a key that starts with the same characters as those you entered, but has more characters, then the message 'Partial Match' will be displayed.

If no record has a KEY starting with your specified characters, then the message 'Key Not Found' will be displayed together with the record which has the closest following key.

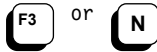
#### 6.2.2 CURRENT



This command obtains the record you are currently working on in this file.

Superbase will remember the **CURRENT** record for up to the last three files used. This can be used in multifile programming situations. If there is no current record, you will see the 'End of File' message and a blank screen.

**6.2.3 NEXT**

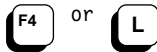


If you have just been using a record or have just viewed a record using one of the other commands, this command obtains the **NEXT** record in the file.

This will be the record which follows alphabetically according to the key field. Use 'M' to obtain next 'Matching' record (see MATCH below).

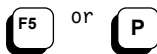
If you have not yet accessed any records in this file, then attempting to use 'Next' will give you the first record in the file.

**6.2.4 LAST**



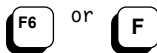
This command obtains the **LAST** record in the file. In a file of invoices with the invoice number as the key, the LAST command would display the invoice with the highest number. This would be the most recent invoice stored. This command is used to end a MATCH, so you can enter new criteria.

**6.2.5 PREVIOUS**



After viewing a record you may wish to see the **PREVIOUS** record in the file, in which case use this command.

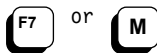
**6.2.6 FIRST**



Use this command to obtain the **FIRST** record in your file.

**6.2.7 MATCH**

The **MATCH** Command provides you with a slower but much more flexible method of searching your file for particular records.



You may use this option to view selected records based on a wide range of criteria in any field or combination of fields.

**SUMMARY OF SELECTION CRITERIA**

Note that "text" can be letters or dates, as appropriate.

Exact match	"=text"	"=number"
Sliding match within field	"text"	
Exact exclusive match (not equal to)	"#text"	"#number"
Sliding match from field onwards	"text-"	
Terminate the sliding match	"*"	
Match greater than field content	">text"	">number"
Match less than field content	"<text"	"<number"
Match alternatives	"t1/t2/etc"	"n1/n2/etc"
Match range of values	">t1&<t2"	">n1&<n2"
Match partial text	"=text*"	
Match with 'wild' character	"=text??"	
Request match in command area	"<-"	(back-arrow)

You can specify that the records selected have particular values in specified fields, or that they have that value in a field within a particular range of fields. You should not use the characters = # - \* > < ? / or & as part of field contents if you wish to use them for matching, because they will be interpreted by the match function as operators from the list above.

Text for matching may be in upper or lower case, as the system does not discriminate. You can also specify that a particular field or one of a range of fields holds a value within certain limits, such as between 100 and 500, or one from a set of alternatives.

Facilities such as these, alone or in combination, ensure that even in a file of thousands of records you can select any particular record or group of records you want with ease.

**RECORD TEMPLATE**

On entering the MATCH option you will be asked: 'Select Match Data'. Superbase will display a blank record on the screen with the cursor in the first field. This is the 'Record Template' into which you enter the criteria to be matched during the search.

Using the cursor movement keys and RETURN to move from field to field, you may now enter the specification that will determine which record or records are to be selected and displayed.

**ENTERING CRITERIA FOR MATCHING**

Each field may either be left blank or may have some characters (numbers, letters, or other characters) typed into it. These characters are called the CRITERION for that field. The criterion typed into a field will consist of a string of characters ( e.g. 'London' ) optionally preceded by an operator ( e.g. '=London' ). The operators available are '=' (equal to), '<' (less than), '>' (greater than) and '#' (not equal to).

If no operator is provided then SUPERBASE will simply look for the string of characters ANYWHERE in the specified field. This is called a SLIDING MATCH and is detailed below.

If a field is not left blank then the specification entered into it will be compared with the contents of each record and only those records which match that specification will be selected.

For example, If '=Jones' is typed into the 'Customer' field of an Invoice record, then only those Invoice records for Jones will be selected.

Similarly, if the specification consists of '=Jones' in the 'Customer' field and '>500' in the 'Amount Due' field then only those of Jones's invoices for more than \$500 will be selected.

### **STARTING THE SEARCH**

Once you have finished entering your specifications into the fields you can store the match criteria and start the search by holding down SHIFT and pressing RETURN. If the cursor is in the last field of the record then just RETURN will do.

The first such record will be displayed and any others can be obtained by pressing 'M' for 'Match' repeatedly. Each time 'M' is pressed, the next record in the file which matches the specification will be displayed.

### **INTERRUPTING AND RESUMING THE SEARCH**

At any time you may instead interrupt the display of selected records by pressing 'N' for 'Next' or 'P' for 'Previous', in which case the next or previous record alphabetically according to the key field will be displayed, whether or not it matches your specification. You can use '+' or '-' to view adjacent screens.

Pressing the 'M' key again will resume the display of records matching the specified criteria.

In this way you can view the records selected by your 'Match' criteria and pause occasionally to browse through neighboring records in the file.

You may even use the ADD or REPLACE commands to modify the file and then continue viewing the selected records where you left off.

**ENDING THE SEARCH**

To cancel the current MATCH operation in order to set up new match criteria, use the LAST command to jump to the end of the file. You may then use MATCH to enter new specifications.

**SLIDING MATCH**

You may search for a string of characters anywhere in a field by omitting the '=' symbol or other operator in the match specification.

This is known as a 'Sliding Match'.

Whereas '=Desk' would only select records with precisely the word 'Desk' in the appropriate field, 'Desk' would also select records with 'Large Desk', 'Pine-desk', 'desktop' and so on in the specified field.

You can also combine a sliding match criterion with other criteria using operators such as '#', '=' or '>'.

**FIELD-INDEPENDENT SLIDING MATCHES**

Another facility offered by the MATCH Option is a field-independent sliding match.

Place the cursor in any field and type the specification followed by a hyphen (e.g. 'London-' or '50-'). The records selected will be those with the specified criterion occurring in any subsequent field.

As described above, the field-independent match will look for your specified criterion in all of the fields including and following the field where the specification was entered. It is also possible to restrict the range of fields to be included by typing the FIELD-INDEPENDENT MATCH DELIMITER symbol '\*'. This will exclude the field it occurs in and all subsequent fields from the fields to be searched, thus defining the search more closely. Also, if you place a field-independent match in field 1 and an exact match in field 4, the search will be cancelled for fields after the exact match.

This facility may also be combined with other types of match criteria. Other criteria must be before the field-independent match and after the delimiter '\*'.

**THE 'AND' OPERATOR AND THE 'OR' OPERATOR**

You may also use the operator '&' (AND) to specify a range of values, and the operator '/' (OR) to specify more than one possible criterion within a given field.

For instance, the specification '=London/=New York' would select those records with EITHER 'London' OR 'New York' appearing in the chosen field.

Similarly, the specification '>100<150' would select only those records with a value in the given field above 100 AND below 150.

**PATTERN MATCHING**

Pattern matching may also be used within a 'Match' specification. The '\*' symbol indicates that any string of characters beginning with the characters preceding it will be accepted. Pattern matching characters should be used in conjunction with the '=' operator, as sliding matches are available for wider ranging matches.

Thus '=Sm\*' would select any record with 'Smith', 'Smythe', 'Smullyan' etc. appearing in the chosen field.

Note that the '\*' symbol can only be put after a string of characters and must have no other characters following it.

Similarly the '?' symbol can be used as a 'wildcard' character so that 'Hutch?ns' would select both 'Hutchens' and 'Hutchins'.

These pattern matching symbols are very useful if you cannot remember how a name is spelled, or if you suspect that it was spelled incorrectly on entry, but they are also useful if you want to ignore characters in a string for the purposes of the match.

**USING THE DELAYED REQUEST CHARACTER**

To enter a criterion that is longer than the space allowed for it on the screen, such as the specification of combinations and alternatives, the back-arrow '<-' provides a way of doing so. (Note: this is not the cursor movement left arrow key.)

Instead of entering the specification, place a back-arrow in the first character position of the field. Press RETURN and continue with the other criteria if any.

When you have finished specifying criteria in the record template, Superbase will prompt you in the command area to enter the criterion for the field with the back-arrow in it. Only when all such fields have had their specifications entered will the search begin.

**MATCH SUMMARY**

These various types of Match Criteria, used individually or in combination, add up to a powerful facility for selecting records from your files. You will see in Section 7 that groups of records selected in this way can be indexed by a list called a KEY LIST which can be stored for future use.

**6.2.8 OUTPUT**

or



The **OUTPUT** command available from the SELECT Option provides a display of the field names and field contents of the CURRENT record, either on the screen or to the printer. To output to the screen use 'display' and to output to the printer use 'print'.

The contents of the record are displayed across the page. The full field sizes are used and information wraps over onto following lines.

You can also change the format of the output from ACROSS to DOWN by typing 'down' from either Main Menu. The two commands can be typed in together as in 'display down' to display each field on a new line.

The field names are displayed down the left hand side of the screen if the display is DOWN and only if all fields are output, with the contents of each field to the right of the field name.

If you want the output to be directed to the printer instead of the screen you must first change the direction of output by typing 'print' from either Main Menu.

Whichever of the commands 'display', 'print', 'down' or 'across' were last used will remain in effect until the converse commands are used.

**6.2.9 ADD**

The **ADD** command is for replicating a record -- adding a new record based on an existing record. The new record must have a different key field and may also be changed in any other fields.

This is useful if you want to enter records which have much information in common with an existing record. It allows you to avoid typing all the information in again.

Using the KEY, NEXT, PREVIOUS or other command, select the record you wish to base the new record on and then use the ADD command to create it.

Note that unless you are working with a Duplicate Key File, the key field of the new record created by means of this command must differ from that of the record on which it was based. If you have not changed the key field, Superbase will display the message 'Key Exists' and will return the cursor to the key field.

**6.2.10 REPLACE**

**REPLACE** is used to edit the records in your files. It is like the ADD command except that instead of producing a new record based on an existing record, it will replace the selected record with the modified version you have created.

Whereas the ADD command requires that the Key field is changed so that the new record can be distinguished from the record it was based on, the REPLACE command does not allow the key to be changed, thus preserving the existing key order of the file.

**6.2.11 DELETE**

This command is used to delete the record that is currently selected from the file, or an empty file format from the database.

You will be asked: 'Confirm Deleted Record' so that you can change your mind before any harm is done. Type 'N' for 'no' if you have made a mistake, otherwise type 'Y' for 'yes' and the record will be permanently removed from the file. If you select this option when there are no records in the file, the file format will be deleted from the database catalog.

Groups of records, or entire files, can be deleted automatically with a simple Superbase program. See Programming Section 1.

**6.3 ALTERNATIVE WAYS TO EXECUTE THE SELECT COMMANDS**

Note that each of these commands can also be chosen by typing the first letter of the command (e.g. for 'First' type 'f'). A list of these letters is provided in the message area above the main screen throughout the SELECT Option.

It is also possible to bypass the Select Menu by typing one or more of the Select commands directly from either of the Main Menus. The command line you type (e.g. 'Select f') will appear in the message area above the screen. Just press RETURN and the command line will be executed.

If you are selecting and matching on more than one criterion, you must separate them with semicolons:

```
select match where [customer] is "=Jones";[goods] is "radio"
```

You can even join commands together, separating them with colons (':') to create longer commands such as:

```
select match where [Customer]is "=Jones":select next:display  
[Customer]
```

Superbase will execute these commands one after the other without any pause between them. (See the Additional Commands Section for an explanation of the 'where' command.) If there is an error in your command line, Superbase will display an error message and stop. You can recall the line and make changes to it by pressing the back arrow key. More details about these COMMAND LINES, as they are called, are given in Section 3 and the Programming Section.

### **SELECTING FROM A LIST**

In Superbase programs, you can use the form 'select from "listname"' to retrieve records one by one and process them individually. See Programming Section 1.

**7 FIND****7.1 USING THE FIND OPTION**

The FIND command enables you to set up a list of the KEYS of a selected group of records which can then be used with the SORT, OUTPUT, BATCH, REPORT, EXPORT and SELECT commands to access and process just those records on the list.

The process of specifying which records are to be included in the KEY LIST is exactly the same as in the MATCH command of the SELECT option (see Section 6.2.7).

**7.1.1 OBTAINING THE FIND OPTION**

To obtain the FIND option press the 'f3' key from Menu 1.

You will see a blank record screen as in the MATCH option, ready for you to enter the match criteria governing the selection of the records to be included in the list.

**7.2 THE DEFAULT LIST**

Once you have entered your match criteria as detailed in section 6.2.7, the message 'Processing' will be displayed at the top of the screen while Superbase searches through the file for the records specified, adding the keys of all matching records to its KEY LIST. You will then be returned to Menu 1.

Unless you specify a name for the KEY LIST as will be detailed below, Superbase will give the list the default name "h8list" or "hlist" depending on the column mode at creation. Like all lists, "hlist" is automatically stored on disk. Every time a new "hlist" is created, the old one is lost.

You can refer to "hlist" or "h8list" with double quotation marks only: "".

The reason for the leading 'h' or 'h8' is to allow you to display the list by using the HELP command. As explained in Section 17, all file names preceded by 'h' or 'h8' are treated as HELP SCREENS and so can be viewed by means of the HELP option.

To view the current "hlist" or "h8list," obtain the HELP option and type 'list' when asked to specify which help you require.

Whatever Key List you last created by the FIND command will remain the current "h8list" until you create another.

**7.3 NAMING AND STORING A KEY LIST / APPENDING TO A LIST**

Using the default Key List is convenient for constructing temporary lists of records for various purposes, but there will be many Key Lists that you will want to keep to use again and again.

You may, for example, want a permanent list of customers who have bought a particular product from you, or a list of products with a particular discount rate.

For purposes such as these you need to be able to store a Key List on disk with the appropriate file. In order to do this, all that is required is to give your Key List a name other than "h8list". The maximum length of a list name is 16 characters.

From either Main Menu, type the COMMAND LINE 'find "listname"', where the listname can be any name not already used as the name of a Superbase file, and must be enclosed in double quotation marks.

**Do not use the name of a database. If you use the name of an existing list or file, it will be overwritten by the new list.**

To avoid confusion it is a good idea to call all of your Key Lists by a name ending in 'list'. Examples would be "update-list," "10%d-list" and so on.

**APPENDING TO A LIST**

To append to an existing list, use the form: 'find "update-list,a"'. This is useful when you already have a list (e.g. last month's orders) and want to add to it without repeating the original processing.

After typing 'find "listname"' from either Menu, you will be presented with the blank record template for entering your match criteria just as you are when you press 'f3' to obtain the FIND command from the list of options on Menu 1.

**7.3.1 BYPASSING THE RECORD TEMPLATE**

You can even bypass the record template altogether by entering your match criteria in a COMMAND LINE from either Menu. See Section 3 and the Programming Section for further details of command lines.

In a FIND command line you can specify several match criteria one after the other, using the WHERE secondary command to indicate the start of the list of criteria sought for the record. Instead of placing the operator (if any) and text or number value inside the field angle brackets, you must place it inside double quotation marks. You must separate multiple search items with semicolons. The rules for matching are the same as when using the record template.

You can if you wish place the back-arrow "<" inside quotation marks instead of a value. This will produce prompts for any such fields before the search begins, which is useful when long strings of characters must be entered. When done under program control this technique will produce requests for run-time parameters.

An example would be:

**find "newlist" where [date] is ">01JAN85";[goods] is "<"**

This is an optional method of specifying the match criteria -- you can always use the record template if you wish, but the Key List must be given its name by means of the direct command 'find "listname"' if it is to be stored on disk.

Note that you must separate multiple search items with semicolons.

#### **7.4 USING THE KEY LIST**

Once a Key List has been created it can be put to use in a number of ways. You might want to obtain printed copy from the records in the list. Either obtain the OUTPUT option (see section 8) and in response to its prompt type 'the records from "listname"' (don't forget that "listname" here can be "h8list" or "hlist" if you want), or you can type the following Command Line directly from one of the Main Menus:

**output the records from "listname"**

This is only one limited example of the use of a command using a Key List. Details of its use with the SORT, BATCH, REPORT, EXPORT and SELECT commands are given in the sections relating to those options.

#### **RENAMING THE LIST**

Because the old "h8list" is overwritten every time a new default list is written, you will want to rename some "h8lists" immediately after creation. Select the OTHER Option from the Maintain Menu, and type:

**r0:newlistname=0:h8list**

See Section 15.

## 8 OUTPUT TO SCREEN, PRINTER, OR WORD PROCESSING DISK FILE

### 8.1 OUTPUTTING INFORMATION FROM FILES

The OUTPUT option is used to display or print information from all or selected records in the current file. See Section 8.11 for information on files for use in word processing. Selected or all fields, descriptive text, calculations, and BASIC variables can be output. Special commands can be used to format the output fields. Output commands, like other Superbase commands, may be abbreviated.

#### 8.1.1 OBTAINING THE OUTPUT OPTION



From Menu 1, press the 'f4' key. You will be prompted with:

**'Enter: all/from "list" (item list....)'**

You are being asked whether you want the output from ALL of the records or just those from a predefined KEY LIST, and which fields you want to output. The '(item list....)' signifies that field names and/or variables may be included in the command.

#### 8.1.2 CHANGING THE DIRECTION OF OUTPUT

You can direct the results of your 'output' command to the screen, printer, or to a disk file. The disk file option 'OUTPUT TO' is discussed later in this chapter.

The commands for changing the direction of your output are 'display' and 'print'. These commands can be included as part of an OUTPUT command as in:

**output print all**

#### 8.1.3 'PRINT' AND 'DISPLAY' DIRECT FORMATTED OUTPUT

The 'print' and 'display' commands can be used without the 'output' command:

**print @10[lastname]**

**display @10,5[lastname]**

The formatting and positioning commands described in the following section are permitted.

However, you may NOT use the secondary address commands 'all', 'from "list", or 'the records', unless the word 'output' begins the command line.

#### **8.1.4 OUTPUT CONTROL COMMANDS SUMMARY**

This section provides an at-a-glance checklist of the various output secondary commands.

<b>display</b>	Directs output to screen. Can be Primary command. Remains in force until 'print' is used.
<b>print</b>	Directs output to printer. Can be Primary command. Remains in force until 'display' is used.
<b>down</b>	Fields shown each on a separate new line. Remains in force until 'across' is used.
<b>across</b>	Fields shown one after the other, overlapping onto new lines if necessary. Remains in force until 'down' is used.
<b>all</b>	Output includes all records in the currently selected file.
<b>from "listname"</b>	Output includes only those records whose keys appear in "listname". The list can be on a different drive from the database.
<b>fill</b>	Specifies standard format for disk file output.
<b>to "filename"</b>	Specifies name of a sequential disk file to contain output field contents. The file can be on a different drive from the database. Add ",a" to the filename to append to an existing file.
<b>[fieldname]</b>	If one or more [fieldname] is included in the command, only the fieldnames defined are output.
<b>a\$</b>	Any previously defined string variable may be included in the command.
<b>a</b>	Any previously defined numeric variable may be included in the command.
<b>"text"</b>	Text may be output by inserting it within quotes in the output command.
<b>&amp;</b>	Is the formatting and truncation command. See later in this chapter.
<b>@</b>	Is the position command. See later in this chapter.

## 8.2 DISPLAYING ALL RECORDS

The simplest response to the output prompt is 'all'. The contents of fields are displayed one after another across the screen with data wrapping over the ends of lines. Pressing RETURN displays the next screenful of records from the file. Display of records can continue until the end of the file is reached.

Responding to the output prompt with:

**all the records down**

will result in information from the first record in the file being displayed down the screen with the field names on the left of the screen and the field contents to their right. The message 'Press Return to Continue' will be displayed on the command line.

Pressing RETURN results in a similar display of the next record in the file, record by record until the end of the file is reached. If you want to go back to the menu at any time, press STOP.

```

Press Return to Continue                                7

No.              00907
Ref              SMI/10090
Date            23MAR83
Goods           12 Tapes
Price           2.00
Amount         24.00
Disc.           0
Total          24.00
Tax            15%
Due            27.60
Paid           no
Total-paid     10.00
Outstanding    17.60
Last-Paid      10.00
Date-Paid      01MAY83
Update         no
  
```

### **8.3 DISPLAYING FROM A KEY LIST**

Alternatively you could respond to the prompt with:

**the records from "listname"**

where "listname" is the name of any KEY LIST you have created by means of the FIND command or with SORT, including "h8list" (see section 7). Note that 'the records' is an optional element in the command, included to make it more natural to use.

The display would be the same as with the all command, but would be of only those records whose keys appear on the Key List you have specified.

### **8.4 PRINTING OUTPUT**

To obtain your output on the printer rather than the screen, the command following 'f4' is:

**print all the records**

or

**print the records from "listname"**

The converse of the 'print' command is the 'display' command which redirects the output to the screen, as in:

**display all the records**

The command last used, 'print' or 'display' will remain in effect until the direction of output is changed by the converse command.

### **8.5 OUTPUTTING SELECTED FIELDS**

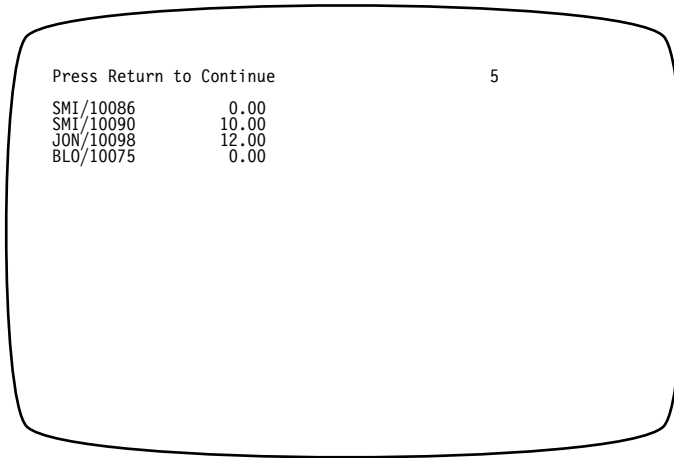
The commands above output information from all of the fields in each record. You can restrict the amount of information to be displayed or printed to one or more fields by responding to the output prompt:

**all the records [number]**

**all the records [number] [due]**

or any combination of fields in the records, where each field name is enclosed in square brackets. Superbase automatically inserts a space between fields unless you use '+' to concatenate them, as in [number]+[customer].

The result is as in the following screen:



Press Return to Continue		5
SMI/10086	0.00	
SMI/10090	10.00	
JON/10098	12.00	
BLO/10075	0.00	

Output record from demonstration file "cust.inv"

## 8.6 OUTPUT ACROSS OR DOWN

If your display is 'across' the screen, a space will be inserted after each selected field unless the fields (and/or variables) are joined with '+'. If output is 'down' each selected field will be on a new line.

To display across respond to the prompt that follows f4 with:

**across all the records [customer] [due]**

After filling the screen with the information from each record displayed horizontally, the message 'End of Page' will appear at the top of the screen.

If more fields were displayed than could fit across the screen, the information would wrap around the edge of the screen. This can be avoided by use of the formatting commands detailed below.

To reverse the 'across' command you can use the 'down' command as in:

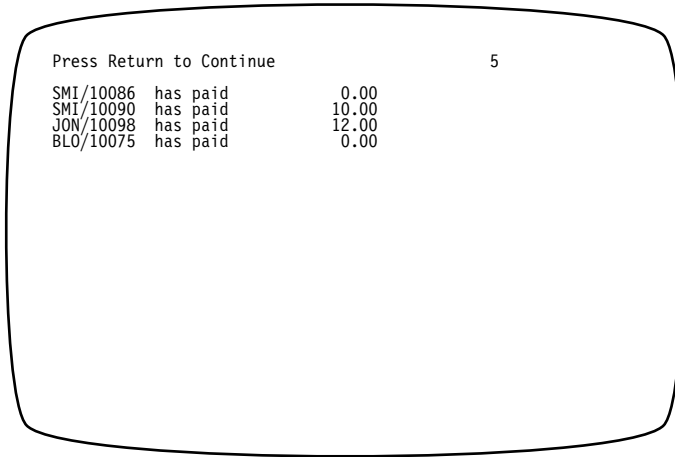
**down all [customer] [goods] [due]**

Just like the 'print' and 'display' commands, whichever of the 'across' or 'down' commands was used last will remain in force until the converse command is used. The default on start up is 'across'.

### 8.7 OUTPUTTING EXTRA TEXT WITH INFORMATION FROM RECORDS

You may also include strings of text with your output, for headings or as in:

**across all [number] "has paid" [due]**



Press Return to Continue		5
SMI/10086	has paid	0.00
SMI/10090	has paid	10.00
JON/10098	has paid	12.00
BLO/10075	has paid	0.00

Note that if the results of formulas and BASIC variables are to be output they should not be enclosed in quotation marks like text strings.

### 8.8 OUTPUTTING CALCULATIONS ON INFORMATION IN FIELDS

The information displayed or printed need not be restricted to how it appears in the records since you can specify calculations to be performed on the fields before they are displayed.

Suppose that you wanted to output a list of items showing a 10% price increase. You could respond to the output prompt with:

**across all [goods] "New price is" 1.1\*[unit-price]**

Any calculation on the field contents can be used including calculations using BASIC functions. More details on performing calculations on the fields is provided in the section on the CALC command in section 11.

BASIC variables can also be output in addition to, the fields, calculations and text strings discussed in this section. This is detailed in the Programming Section.

## 8.9 FORMATTING AND POSITIONING FIELDS AND/OR VARIABLES

These commands are essential if you want to produce output in regular columns, and/or in groups of lines (e.g. three lines per record). They also give you control over decimal place format and text field length.

### 8.9.1 TRUNCATING OUTPUT

Text fields will be printed or displayed with the length specified when the fields were set by means of the FORMAT Option (see section 4) unless the TRUNCATION COMMAND is used.

A text field such as 'goods' may have been set to eighteen characters long to allow for big product names, but most of the product names you are using may be less than 18 characters long. This would mean that when these product names are output, the extra spaces after the product name would be included unnecessarily. To avoid this you can include the truncation symbol with your response to the output prompt as in:

**display across all &[goods] [unit-price]**

This would chop off all of the trailing spaces from [goods].

It is also possible to truncate the output to a particular length. Typing a number after the truncation symbol will truncate the contents of the field concerned to the number of characters specified. For example

**print all &6[goods] [unit-price]**

would print just the first six characters from the 'goods' field.

The contents of numeric fields can also be truncated, but two numbers separated by commas must be included after the truncation symbol, one for the number of digits before the decimal point and one for the number of digits after the decimal point, for example '&5,2[amount-due]'. The default numeric format is 10 positions before and 2 after the decimal point.

You can mix truncation commands for text and numeric items, as each truncation command applies to the NEXT appropriate item.

Truncation forces rounding. If the field contains a number with more digits after the decimal point than specified by the truncation command the number will be rounded up. For instance, if the price of an item were '75.87' and the output specification were:

**all [goods] &2,1[unit-price]**

Then the price would be output as '75.9'.

You must take care, however, not to specify a truncation of a numeric field which has fewer digits BEFORE the decimal point than a number to be output. If you do, the number will be printed or displayed as a string of '#' symbols to indicate that the number overflowed the format you specified for it.

The command &0,x will suppress leading spaces in the next numeric item, causing left justification with one space for the sign (only the '-' sign is printed).

**SUMMARY OF OUTPUT TRUNCATION COMMANDS**

<b>[field]</b>	Remove trailing spaces from a text field.
<b>&amp;n[field]</b>	Output 'n' characters of the text field defined.
<b>&amp;n,n[field]</b>	Output numeric fields in decimal point format.
<b>&amp;&amp;n,n[field]</b>	Sets the default decimal place format for subsequent items. (C128 only)
<b>&amp;0,n[field]</b>	Output numeric fields left-justified with leading spaces suppressed.

**8.9.2 POSITIONING OUTPUT**

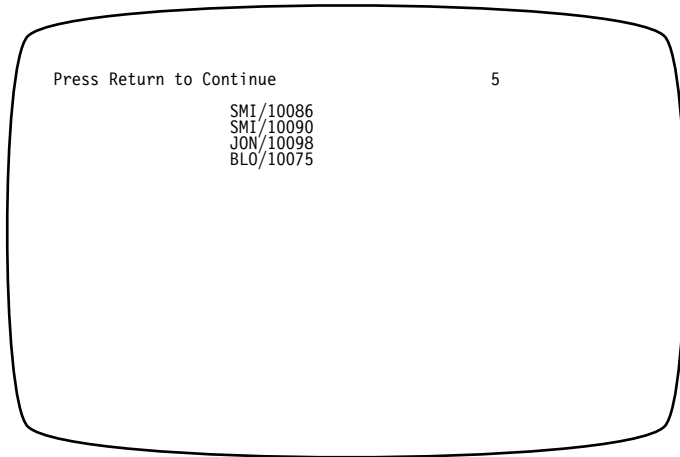
The second formatting command available for use with OUTPUT is the POSITIONING COMMAND.

This command is used to position your output on the screen or printed page and must have either one or two numbers following it, the first for the COLUMN and the second for the LINE.

For instance, the command

**all the records @20[number]**

will position the contents of the 'Number' field at column 20 on the screen, as in the following diagram:



The command applies only to the NEXT item. Following items without their own positioning commands will appear in sequence either across or down as appropriate. To achieve column and line location use a command such as:

**the records from "h8list" @10,15[customer]**

which would position the output at column 10 and 15 lines down the screen.

Note that if you specify just a column number it is possible to display information from many records, one below the other starting at the same column, but if you specify both a column and a line number then only information from one record at a time can be displayed on the screen.

This is for the simple reason that two items of information can be in the same column by being one below the other, but cannot be in the same line AND column without one overwriting the other.

Two further forms of the command exist. The command @0 positions the cursor at line 1, column 1 of the current screen, without clearing the screen.

The command @x,0 will cause a carriage return to column x. Use @1,0 to force the printing of blank lines.

Used in combination with the other output formatting commands, the position command gives you enormous flexibility as to how your output should appear on either screen or printer.

SUMMARY OF OUTPUT POSITIONING COMMANDS

@n	Defines column position.
@n,n	Defines column and row position.
@0	Positions the cursor at column 1, line 1.
@0,147	As above, but clears the screen. (C128 only)
@0,12	Used with 'print' to cause a form feed on the printer (substitute your form feed code if 12 is not correct). (C128 only)
@n,0	Prints a blank line.
@-	Switches on printer underlining or screen reverse video. Repeat to switch off.
@+	Switches on underlining for next item only.

**8.10 ABBREVIATING OUTPUT COMMANDS**

Various examples were given above of output commands using the words 'the' and 'records' as in:

**down all the records [number]**

and

**across the records from "h8list" [customer]**

In fact these two words, and also the spaces between words, are not needed at all by Superbase and are allowed merely to make the syntax more natural. These commands could instead have been entered as:

**downall[number]**

and

**acrossfrom""[customer]**

We can even abbreviate further since all Superbase commands can be reduced to their shortest unique form, with a minimum of 2 letters. Usually you can type just the first letter followed by the second letter with the SHIFT key held down. The previous two commands then become:

**d0aL[number]**

and

**aCfR""[customer]**

Once you are fully familiar with the commands you may wish to use these abbreviations to cut down the amount of typing required.

**8.11 OUTPUT TO A FILE**

Although Superbase is completely and directly integrated with the Superscript 128 word processor (see Programming Section 2 and the SUPERScript command for details), some users will want to create disk files for use with other packages such as Easy Script or Script/Plus. Integration is achieved by outputting data (in CBM ASCII code) from a Superbase file into a file that can be read in by the other package. The command is entered on the command line in the usual way when the OUTPUT Option is selected from Menu 1.

**METHOD 1: AS FOR NORMAL OUTPUT**

There are two forms of the command. The first one outputs to a file exactly as to the screen or printer, following all truncation commands. If the output is DOWN, no field names will be output, as they usually are with OUTPUT DOWN. If the output is ACROSS, field contents will be output one after the other on the same line, up to the maximum specified with the right margin setting command, RMARG. The command is:

**down all to "addresses" [lastname] [address1] [address2]**

This would produce a file called "addresses" containing the name and two lines of address from all the records in a Superbase file. Be sure that the filename you give is NOT that of an existing database file. Each field will appear on a separate line and each line will end with a carriage return character. A blank line with just a carriage return on it will be output between records.

A Key List may also be used:

**across from "" to "addresses" [lastname] [address1] [address2]**

This would use the Key List "h8list" or "hlist" (usually obtained with a FIND operation) to produce a file called "addresses" containing the name and two lines of address from the selected records in a Superbase file. All three fields will appear on the same line with a space between fields, and each line will end with a carriage return character. A blank line with just a carriage return on it will be output between records.

**METHOD 2: USING SECONDARY COMMAND 'FILL'**

The other form of the command is especially designed for Easy Script, Script/Plus or earlier versions of Superscript, and uses the additional command FILL:

**all fill to "addresses" [Lastname] [address1] [address2]**

The use of FILL implies the following:

1. Output is 'down'.
2. Trailing spaces are truncated unless the '&' with a specific value is used.
3. Numbers have one leading space for the sign.
4. Numbers are left-justified.
5. Numbers have two digits after the decimal point, unless the '&x,y' formatting command specifies otherwise.
6. Each line has a carriage return at the end.
7. A blank line with only a carriage return is output between records as a separator.

The last point means that you must insert one "dummy" variable block in your Easy Script document, to "trap" the extra blank line. The dummy block should appear in a comment (\*nb) line, anywhere after the last of the "active" blocks. Because it appears in a comment line, it is trapped and not printed.

A space is normally output after each field when more than one field appears on the same line. You can join fields and text together with the '+' sign.

## **8.12 ENTERING OUTPUT COMMANDS IN A COMMAND LINE**

As with all of the Superbase commands it is possible to bypass the Menu selection and type in your commands directly as a COMMAND LINE from either of the Main Menus. All you have to do is to prefix the commands with the Primary Command 'Output' as in:

**output all the records [date][customer]**

You can also combine output commands with others in the same Command Line such as:

**find "h8list" where [due] is ">500" : output from "h8list"  
[customer];[number]**

Further examples of using the Output command can be found in Tutorial Two and Three and more details about Command Lines can be found in Section 3.

**8.13 ADDITIONAL OUTPUT CONTROL FEATURES**

1. Clear the screen with 'display @0,147' (C128) or 'display chr\$(147)' (C64/+4).
2. Display data without clearing the existing screen by first using '@0' to position at column 1, line 1, for example 'display @0@4,20[Name]'.
3. Prevent carriage return of cursor or print head by placing a semicolon directly after the data item, for example 'display [name];' or 'print&5,2x;'
4. Separate data items with semicolons to avoid confusing the system, for example 'display a;b' -- do not use commas.
5. ASCII codes can be displayed in the form '\$157' (C128) or 'chr\$(157)' (C64/+4). This can be used to control screen features such as graphics, flashing, underlining, etc. Such characters do not occupy a column position.

**8.14 USING 'OUTPUT TO' TO REORGANIZE A DATABASE FILE**

You can change the order in which fields are stored in a database file. The steps are as follows:-

1. Set up a file format with the new order. You can omit fields or add extra fields as you wish. Insure that the last field is a "dummy" text field -- this is to hold the blank line record separator.
2. Use 'output to' to create a disk file, naming the fields in order of the format you have just created. For new blank fields, insert a variable, a\$, that you have previously set to the value of a space with 'calc a\$=chr\$(32)'.
3. Insure that the new format is the selected file. Use the 'import' command (see Section 15) to bring the data from the disk into the new format.
4. If you want, delete the disk file and the records in the original database file.
5. Don't try to rename a database file within a database.

You can use the "utility" program to copy and recover most damaged databases. See Appendix E.

## 9 FILE / DATABASE

### 9.1 FILES AND DATABASES / CHANGING DISKS

Superbase organizes your stored information in files of records, each of which has a different filename and may have a different record format.

These files are held in groups of fifteen and each such group is known as a database. Information may be exchanged between databases by using the OUTPUT TO, IMPORT and EXPORT commands. **Never remove a database disk and replace it with another without using the 'DATABASE' command.** The procedure is to return to either Main Menu, change disks, and type the command: 'database "name"' where "name" is the name of the new database. See also Programming Section 2.1.3.

#### 9.1.1 OBTAINING THE FILE OPTION

The FILE Option is used whenever you wish to change the file you are currently working with to another file in the database or to create a new file.



From Menu 2 press the key marked 'f1'.

The message 'Mode: Enter Filename' will be displayed in the message area at the top of the screen and the main screen area will display a list of files available in the current database. Superbase will wait for you to type in the name of the file you wish to select, which may be either one of the existing files listed on the screen or the name of a new file that you want to set up.

New files may be added to the current database up to a maximum of fifteen files. Use letters, numbers and the full-stop or period only for filenames.

If you have selected an existing file, you will be returned to Menu 1. Notice that the 'File Selected' Indicator shows the name of the file you have chosen.

You may now use any option from the Menu, such as ENTER if you want to enter information into the file, or SELECT if you want to access a particular record.

If, on the other hand, you have selected a file that does not exist, the screen will clear and the following prompt will appear:

**'File Does Not Exist: Create It?'**

If you respond 'y', you will be placed in the FORMAT Option, ready to design the screen layout for the new file.

If you respond 'n', you will be returned to the 'Enter Filename' prompt. The new filename will not show in the 'File Selected' Indicator until the format has been completed.

Directions on how to FORMAT a new file are given in Section 4.

## **9.2 CHANGING FILE BY DIRECT OR PROGRAM COMMAND**

A new file may be selected using the 'file' command followed by the filename in quotes.

Examples

```
file "invoices"  
a$="addresses":file a$
```

The above commands avoid the standard 'Enter Filename' prompt and allow programs to transfer to another file without user intervention.

The command 'file' on its own in a program will cause the 'Enter Filename' prompt to appear on the command line and await operator response. See the "start" program at Appendix A.

## **9.3 REORGANIZING A DATABASE**

This can be done using the OUTPUT TO command. See Reference Section 8.14.

**10     SORT****10.1   SORTING THE RECORDS**

Superbase automatically keeps the records in a file sorted alphabetically according to the contents of the KEY Field of each record.

You may, however, want either all of the records or the records in some Key List sorted according to the contents of a field other than the KEY Field.

For instance, in a file of invoice records you might want to SORT the records according to the amount each customer owes you.

With the SORT command you can create a sorted list which is just like a Key List except for the order in which the records are listed. This list can then be used as a Key List in the OUTPUT, BATCH, REPORT, EXPORT and SELECT options.

**10.1.1 OBTAINING THE SORT OPTION**

From Menu 2 press the 'f4' key.

**10.2   ENTERING THE SORT PARAMETERS**

On entering the SORT option you will be prompted with:

**'all/from "list" (item list....)'**

If you want to produce a sorted list of all the records in the file, type 'all'. Otherwise type 'from' followed by the name of the Key List of records to be sorted. The (item list....) signifies that field names but not variables or functions may be included in the command.

You can omit the name of the Key List and just type 'From"' in which case the current default Key List, "h8list" or "hlist," will be used (see Section 7.2).

You can also choose a DESCENDING SORT or an ASCENDING SORT. A Descending Sort will sort the records in a reverse alphabetic or numeric order and is specified by typing 'D-' after the selection of records to be sorted. The order is from z to 'space', including from 9 to 0. If you leave out the 'D-' then Superbase will assume that you want a normal Ascending Sort. With 'D-', all fields are sorted in descending order.

Next you must specify which field or fields you want the sort to be performed 'on', enclosing the name of each field chosen in square brackets. If you specify more than one field, Superbase will take those records having the same contents in the first specified field and sort those according to the second field specified and so on. The key is automatically included at the end of the list as a field to sort on.

The sort will use ten characters from each chosen field unless otherwise specified by the Field Truncator command '&' (see Section 8). For instance, 'on &6[lastname]' will only use six characters from the 'lastname' field and ignore any others for the purposes of sorting.

To include the whole field contents in the sort, even if the length is greater than ten characters, '&[lastname]' can be used. The maximum number of characters to sort on is 256 minus the length of the key.

Finally you can specify the name of the new list of records which the sort will produce by typing 'TO "new listname"'. This will result in the sorted list being stored on disk under that name. If you leave out the 'TO' command then the sorted list will be stored under the name "h8list" as in the FIND Option.

After typing in these parameters, such as:

**from "h8list" D- on [due] [customer] to "order-list"**

or perhaps

**all on [customer]**

press RETURN and the message 'Processing' will be displayed at the top of the screen. During the sort Superbase may require space for its intermediate sort file called "h8list". This file, if created, will be 8K in size initially, and will increment in size by 8K each time it is recreated during the process. You can delete an "h8list" left on the disk after a sort if you need to create space. A record counter is used during the sort to indicate the number of records sorted and written to disk. After Superbase has finished creating the sorted list you will be returned to the Menu 1 and the sorted list will be ready for use.

**Note that the fewer or shorter the parameters specified, the quicker the sorting process will be.**

**11 CALCULATION****11.1 THE CALC OPTION**

The CALC Option is provided to enable you to carry out calculations. You can operate on the current record or on data input via the keyboard.

As well as normal arithmetic operations, a wide range of BASIC functions, including trigonometric functions and string functions, is available.

The results of your calculations can be stored in the fields of the record, stored in BASIC variables or simply displayed on the screen.

RESULT Fields and CALENDAR Fields cannot be altered by the CALC Command. KEY Fields can be set and the new record can be added to the file. See later in this section.

**11.1.1 OBTAINING THE CALC OPTION**

From Menu 1, press the 'f5' key. The message 'Enter Calculation' will be displayed at the top of the screen.

**11.2 ENTERING CALCULATIONS**

The simplest type of calculation you can enter is a straightforward expression such as `'15*48.6'`. This would display the result of the expression, namely `'7.29'`. Any such expression entered which does not include an `'='` sign will cause the result of the expression to be displayed in this way.

Expressions can of course be a good deal more complex, such as `'cos(sin(9)/log(10))*2'` and may include string functions such as `'left$([Name],3)'` to obtain just 3 characters of a field.

Using the `'='` sign will cause whatever is to the left of the `'='` sign to be assigned the value of whatever expression is to the right. For instance `'[unit-price]=18.50'` will assign the value 18.50 to the PRICE field in the current record. Similarly, `'[Lastname]="Hodgkins"; [street]="35 Sunnyview Crescent"'` will result in the 'Lastname' and 'street' fields being changed accordingly. Note that when assigning strings of text to non-numeric fields, the text assigned must be put in double quotes as above.

**STORING CALCULATIONS**

It is important to remember, however, that when the contents of fields are modified in this way the modifications are only retained temporarily unless you issue the STORE command to tell Superbase that you want the modified version of the record to be stored on disk.

In fact without the STORE command the modifications will only be retained until another record becomes the current record. This is useful since you may be modifying numeric fields just to see what effect the modification would have on the RESULT fields in the record. In this way, the CALC option provides you with a 'What if?' facility for exploring the effects of various changes such as changes in discount or tax rates.

The STORE command can be typed in when you have returned to the Main Menu after using the CALC option. Simply type 'Store' and press RETURN.

**SEMICOLONS**

Also note that CALC lines involving more than one calculation must have the individual calculations separated by semicolons.

**11.2.1 STORING RESULTS IN BASIC VARIABLES****NUMERIC VARIABLES**

Another possibility is to assign values to BASIC variables such as 'x=[unit-price]-[unit-price]\*0.15', which would give 'x' the value of the price minus a fifteen percent discount. 'x' could then be used in further calculations either in the same CALC Command or later on in a new calculation.

**DISPLAY RESULT**

This facility can be exploited in the BATCH option to produce running totals (see Section 12.3).

Since any expression not including the '=' sign displays the result on screen, you can evaluate an expression and then display the result as in:

**x=[unit-price]-([unit-price]\*0.15);x**

Information to be displayed in this way should always be the last item in the CALC line. Only BASIC variables, not fields can be displayed in this way.

**TEXT VARIABLES**

The examples above are both examples of assigning numeric results to variables. It is also possible to assign strings of TEXT to variables, but care must be taken that TEXT VARIABLES are used for this purpose as in:

**x\$="1 Portable TV"**

A text variable differs from a NUMERIC VARIABLE in that it always ends with a '\$' sign as in 'a\$', 'a1\$', and 'x\$'.

### **11.3 ENTERING CALCULATIONS AS COMMAND LINES**

As with other Superbase commands, you can type a CALC command in directly from either Main Menu (see Section 3). Just type in a CALC line as above but with the Primary Command 'CALC' preceding it as in:

**Calc x=[unit-price]/2;y=[unit-price]/3;x;y**

which would display one half and one third of the 'unit-price' field in the current record.

Calculations can be combined with other Superbase commands as in:

**select match where [customer] is "=Robins":calc x=[Due]\*1.1;  
[Due]=[Due]+x;x:store**

Note the use of the STORE command here to make the modification to Robins' record permanent.

Remember to separate the various Command Clauses by colons and the various calculations within the CALC Command Clause by semicolons.

If you modify a field with CALC and wish to display it subsequently, you must use a further CALC clause to do so. This is because all references to a field within the same clause will use the original value ('value' here meaning both text and number value) of the field. However, you can assign the value of a modified field to a BASIC variable and display that all in one command.

### **11.4 ENTERING A NEW RECORD WITH THE CALC COMMAND**

Since CALC can be used to set the contents of fields, it can be used to enter a new record into the file. To do this however, you must first CLEAR the current record by issuing the CLEAR command and you must end by storing the new record with the STORE command. It is also mandatory to enter a KEY in the KEY Field. You cannot CALC a RESULT or a CALENDAR field, only the fields that they are derived from. Nor can you CALC a replica field. You can, however, CALC a CONSTANT field.

The CLEAR Command, which is detailed in the Programming Section with the STORE command and others, does not delete the current record. Instead, CLEAR produces a "blank" record which may be "filled in" with the CALC command (the ASK command can also be used). An example would be:

**clear:calc [number]="01670";[goods]="1 FR80 Camera";  
[unit-price]=79.85:store**

where [number] is the KEY Field.

If there are any errors in the CALC commands, the STORE command will not work, so you cannot create invalid records.

**12     BATCH****12.1   THE BATCH OPTION**

The BATCH option, like the CALC option, is for carrying out calculations on the information in your files. But whereas the CALC option performs calculations on the fields in the CURRENT record, the BATCH option carries out calculations on the fields in ALL the records in a file or on a predefined selection of records using a KEY LIST (for details about setting up a Key List see Section 7).

This ability to perform operations on selected fields of records throughout the file makes UPDATING your files a simple, automatic process.

**12.1.1 OBTAINING THE BATCH OPTION**

To obtain the BATCH option from Menu 2, press the 'f3' key.

**12.2   PERFORMING CALCULATIONS ON ALL RECORDS OR SELECTED RECORDS**

When you enter the BATCH option you will be prompted with:

**'Enter: all/from "list" (item list....)'**

If you type ALL before the calculations you specify, they will be performed throughout the file. If you type 'From "listname"' then they will only be performed on the records specified in the Key List you have named. The '(item list....)' signifies that field-names and/or variables may be included in the command.

The calculations themselves are entered in exactly the same way as in the CALC option detailed in the last section. You do not have to use the STORE command to make modifications to fields in the records permanent. The BATCH command automatically stores the records if they have been modified.

An example of using BATCH to update a file such as the "invoices" file from Tutorial Two is as follows:

**from "price list" [unit-price]=[unit-price]\*1.1**

**12.3 CALCULATING TOTALS**

The BATCH option can also be used to display running totals of fields in the records. Suppose you wanted to total the amount outstanding on all invoices in the Customer Invoice File. The BATCH command for this would be:

```
all x=x+[due];x
```

Here the BASIC variable, 'x', is being used as an ACCUMULATING variable. SUPERBASE will go through all the records in the file, adding the value in the OUTSTANDING field to the previous value of 'x' and displaying the result. In this way 'x' will accumulate the values in the OUTSTANDING field for each record. The direct command 'display x' will then show the final value of 'x'. Note that the variable should be set to a value of zero before you begin, with a command of the form 'calc x=0'.

**12.4 USING THE BATCH COMMAND IN A COMMAND LINE**

Like all of the other Primary Commands, BATCH commands can be entered directly from either of the Main Menus. Simply prefix the command with the word 'Batch' as in:

```
batch all x=x+[due];x
```

The BATCH command can also be combined with other commands to form a more complex Command Line as in:

```
find "h8list" where [goods] is "Radio";[Quantity] is ">10":  
  batch from "h8list" [unit-price]=[unit-price]-0.1*[unit-  
    price]
```

This Command Line would find all the invoices for more than ten radios and reduce the unit price by 10%.

**SEMICOLONS**

Remember to separate multiple items with semicolons, as in CALC.

## 13 REPORT

### 13.1 THE REPORT GENERATOR

The REPORT Option provides you with a REPORT GENERATOR for producing printed reports based on information from your files including TOTALS and SUBTOTALS calculated on any field.

The reports can be formatted in a number of different ways with the commands used in the OUTPUT Option (see Section 8). Text of your choice and calculations of the kind available in the CALC and BATCH Options can be included.

Superbase prompts you to input the different elements that make up the report. The prompting process is like a dialogue, with Superbase asking the questions and you giving the answers.

From your responses to these prompts the Report Generator will produce a PROGRAM which creates the report itself. This program becomes the current program in memory and can be modified like the user-created programs of the PROG option or can be simply executed or stored on disk for future use. You can use the special report commands to write report programs without using the Report Generator.

The report can either be displayed on screen or printed out on paper depending on which of the commands 'display' or 'print' was last used. You can change from one to the other by typing either 'display' or 'print' from either Main Menu before executing the report.

#### 13.1.1 OBTAINING THE REPORT OPTION



From Menu 1 use the 'f6' key to obtain the REPORT Option.

### 13.2 USING THE REPORT GENERATOR

#### SELECTING THE FILE

On entering the REPORT Option you will be asked:

**'Enter File to Report on'**

Type in the name of the File containing the information you wish to include in the report and press RETURN. **Note that the file name must be enclosed in double quotation marks.**

All parameters from here on are optional. If you wish to omit any of them, just press RETURN when you receive the prompt concerned.

**ENTERING THE TITLE**

The next prompt is:

**'Enter Report Title'**

The Report Title, which must also be enclosed in quotation marks, can be as long as you like and can be positioned on the page with the POSITION command '@' (see Section 8).

Your title may be just a line or two of text to head or to explain the nature of the report, or it may include the headings of any columns of information in the report. To force a blank line, use the formatting command '@1,0'.

If your title takes up all of the 79/159 characters available in the command area at the top of the screen you should split the title up and enter it in parts. You will be prompted with:

**'Any More?'**

If you want a longer title than you have so far entered then type 'Y' for 'Yes' otherwise type 'N' for 'No'.

**TOTALS AND SUBTOTALS**

After entering your title and pressing RETURN you will be prompted with:

**'Enter Total Calculation(s)'**

Each report can contain up to ten TOTALS on any of the fields in the file you have specified. These totals, which can be printed at the end of the report are designated t0, t1,..., t9 and can be specified by using any of the parameters available in the CALC option. A simple example which would total the amount due from every record is 't1=t1+[due]'. Multiple totals may be specified, and must be separated with semicolons.

You could also specify calculations on the totals such as:

**t1=t1+[unit-price]\*0.8; t2=t2+[unit-price]\*1.1**

Another option open to you at this point is to specify up to ten SUBTOTALS which are printed at various break points in the report. The subtotals are designated by s0, s1,..., s9 and are specified in the same way as totals:

**s1=s1+[quantity]**

Again, any of the calculations available in the CALC Option can be performed on subtotals.

If you want to specify more totals and subtotals after using up the available space, respond 'Y' to the 'Any More?' prompt.

**SUBTOTAL BREAK**

After specifying the totals and subtotals you will be prompted with:

**'Enter Field for Subtotal Break'**

Subtotals will be printed every time the field you specify as the field for subtotal break changes its value.

For example, you may be producing a report about sales of various goods based on a file containing quantities of goods sold. Suppose you want a subtotal of the number of each product sold. You would specify a subtotal such as 's0=s0+[quantity]' and respond to the Subtotal Break prompt with '[goods]'.

**USING A LIST OF SORTED RECORDS**

SUBTOTAL BREAK FIELDS are only relevant if you produce a key list sorted on all these subtotal fields, before executing the report. If you do not do this first then the records will be printed in the ordinary alphabetical order of their key fields, and the value of the field specified as the Subtotal Break Field will change arbitrarily, giving you subtotals with no significance.

Having sorted the records you should use the name of the SORT LIST when asked 'All or From "list"' (see below).

If you want more than one Subtotal Break Field then specify all of the break fields in the SORT.

**THE SUBTOTAL TEXT**

Next, a prompt of

**'Enter Subtotal Text'**

will allow you to specify any text to be printed when the value of the field for subtotal break changes. The text must be inside double quotation marks. You might wish to print the words "Subtotals for department" each time the 'department' in a Sales Report changes. The quoted text may be preceded or followed by the subtotal variables to be printed, i.e. s0, s1, etc. Only if these are specified in the first line of Subtotal Text will they be automatically set to zero after being printed. Type in the text and commands you require and press RETURN.

You will again be prompted with:

**'Any More?'**

If there is to be more than one Subtotal Break Field then type 'Y' for 'Yes' otherwise type 'N' for 'No'.

**USING A KEY LIST**

The next prompt displayed is:

**'Enter All or From "list"'**

As in other options like OUTPUT and BATCH you can specify whether you want ALL of the records in the file to be used or just those in a particular KEY LIST. Type 'all' or 'from' followed by the name of the key list enclosed in quotation marks and press RETURN.

If you omit both the 'all' and the 'from' from your detail line, the report will be from the current record. You can cause the current record to be selected according to your own requirements by appropriate programming (with select match, for example).

**SELECTING THE MAIN LINE OF THE REPORT**

The next prompt governs the main details in the report. You are requested to:

**'Enter Report Detail'**

The detail may consist of the contents of fields, strings of characters or the values of BASIC variables. You can use computed fields such as '[Sales] - [Costs]'. Most often the Detail section will consist of fields with text to describe what they are such as:

**"Goods: ";@8[goods];"Price: ";[unit-price]**

If you are producing a report consisting of columns of information, however, it would be preferable to use column headings in the TITLE section and just use the word 'TOTAL' in the ENDREPORT section.

Any of the facilities provided with the OUTPUT Option can be used to specify and format the report detail (see section 8) and there are additional commands available such as LMARG, RMARG, PLEN, TLEN, LFEED, SPACE, CONT, PDEV, PDEF and PLUS which are detailed in the Programming Section.

**END OF REPORT**

The next prompt in the report sequence is:

**'Enter End of Report Text'**

You should enter here any line of text and totals you want printed at the end of the report such as '"Total Due is";t1'.

**SAVING THE REPORT PROGRAM**

Having specified the report parameters in this way you will next see the Report Program itself displayed on the screen and will be asked if you want to save the Report Program on disk for future use. The prompt is:

**'Save Report Definition?'**

Type 'Y' for 'Yes' if you do want to save it otherwise type 'N' for 'No'. If you type 'N', the report program will still be available for execution or modification as the current program in memory and you will be returned to Menu 1.

If you answered 'Y' to the last prompt then the final prompt in the report sequence is:

**'Enter Report Name'**

Below the prompt you will see 'Save "' followed by the cursor. Type the name you want to give the Report Program, prefixing it with '1:' if you want to store it on drive 1 of a dual drive unit, and press RETURN. The Report Program will be stored on disk and you will be returned to Menu 1.

**13.3 EXECUTING A REPORT PROGRAM**

Once you have created a Report Program you can produce the report itself by using the EXECUTE Option (see Section 14).

Either select the EXECUTE Option by pressing the 'f7' key from Menu 1 or type the direct command 'Execute' from either Main Menu.

**13.4 LOADING A REPORT PROGRAM**

By storing Report Programs on disk you can build up a library of different types of Reports for future use. If you want to produce a Report from one of these Report Programs and it is not the CURRENT PROGRAM then you must first LOAD that Report Program by issuing the LOAD command from either Main Menu.

Just type 'Load "reportname"' where 'reportname' is the name you gave to the Report Program when it was created. That Report Program will then become the Current Program and can be executed in the normal way. As for all programs, only a copy is loaded into memory, with the original remaining on disk.

### **13.5 OTHER WAYS OF CREATING A REPORT PROGRAM**

Once you have become thoroughly familiar with the procedure of setting up the Report Program there is no reason why you shouldn't produce reports by modifying already existing programs or even writing your own. Details about creating and modifying programs are given in the Programming Section.

### **13.6 ADDITIONAL REPORT PROGRAM FEATURES**

To complete your knowledge, or if you are going to edit the programs produced by the Report Generator, you should be aware of the points discussed in this section.

1. Use the 'plus' command to obtain multiple lines of title, totals, subtotal text, detail, and report ending. See Programming Section 2.1.37.
2. Subtotals are only cleared to zero if they are declared in the first line of a subtotal command.

**subtotal[total]"Total is: "&5,2s1**

However, there may be applications that need a large amount of text to be printed in the subtotal line itself, forcing the use of 'plus' commands to create multiple subtotal text lines. In these cases, if the subtotals to be printed were to appear on lines other than the first, they would not be cleared. If you want them to be cleared, put the subtotals to be cleared after the 'plus' command but on the first line of the 'subtotal' command. See point 3 for an example.

3. To retain the old subtotal break field for printing after a change of subtotal break field has been detected, assign the subtotal break field to a string variable as part of the total command: e.g. 'a\$=[town]'. Here is part of a report that uses this feature:

```
10 total s1=s1+[amount1];s2=s2+[amount2];a$=[town]
20 subtotal [town] plus s1 s2
30 @5,0 "SUBTOTALS FOR " a$ " are" @40 &5,2 s1 @50 &8,2 s2 plus
40 @41"-----"
```

This example also makes use of the feature described in point 2.

4. Superbase 128 only: To cause a controlled form feed, use the positioning command '@0,n' where 'n' is the printer's form feed character, usually 12. This can be included at the end of a subtotal line to change pages after a subtotal break.
5. The 'report' and 'endreport' commands are compulsory in a report program.

6. If you have difficulty printing the £ sign or any other unusual character, you can use a direct command to assign the necessary escape sequence (which should be described in your printer manual) to a BASIC variable before printing the report. For example, 'P\$=chr\$(27)+"Y"' will work for some printers. Then when 'P\$' is placed in the report detail line, a £ sign will appear.
7. You may wish to produce underlined text or fields if your printer supports this feature. Two forms of the '@' command are provided:

**@-[Field]@-**

This switches underlining on for [Field] and then switches it off again.

**@+[Field1] [Field2]**

This switches underlining on for the next text item only. In this case, 'Field1' will be underlined, but not 'Field2'.

If this command is used with screen output, the underlined areas will appear in reverse video. On Commodore matrix printers that don't support underlining the reverse effect will also appear.

8. Single-column labels may be printed without programming. Multi-column labels may be printed with a program that stores details of the name of more than one record in an array in memory, then prints them out line by line. A program for printing labels is provided on the Superbase disk. See Appendix E.
9. Superbase's Programming facility includes a number of commands for error handling. With these, you can equip all your report programs with routines for dealing with unexpected behaviour. See Programming Section 1.
10. Appendix D contains a summary of all printer control features.

## **14     EXECUTE**

### **14.1   THE EXECUTE OPTION**

The purpose of the EXECUTE Option is to run Superbase Application Programs previously created by means of the PROG Option, the REPORT Option, or by numbering individual Command Lines (see Tutorial 3).

Programs are sequences of numbered COMMAND LINES made up from the Superbase commands, both those appearing on the various Superbase MENUS and the extra commands detailed in the Programming Section. Most BASIC commands are also available, so you can program highly sophisticated Superbase operations.

#### **14.1.1 OBTAINING THE EXECUTE OPTION**



To obtain the EXECUTE Option press the 'f7' key from the Main Menu.

### **14.2   EXECUTING A PROGRAM FROM DISK**

There are two ways in which Superbase holds programs: on disk and in memory.

If you have no program in memory, and you wish to execute an applications program from disk, select EXECUTE. The message 'Enter Program Name' will be displayed. Type in the name of the program required and press RETURN. Superbase will now load the program into memory and run it.

#### **14.2.1 REMOVING THE CURRENT PROGRAM FROM MEMORY**

The program in memory is called the current program. If you no longer want to use the current program but want to execute a different program from disk, you use the NEW comand. Type this from either menu and it will remove the current program from memory. You can then select EXECUTE and enter the new program name you require.

### **14.3   EXECUTING THE CURRENT PROGRAM**

If you wish to execute the current program, you simply select 'EXECUTE'. This time no prompt will be displayed. Superbase executes the program already in memory.

#### **14.4 SAVING PROGRAMS ON DISK**

If you want to save the current program on disk for future use you should use the 'SAVE' command.

From either Main Menu type 'save "programe"' and press RETURN. You can give a '1:' prefix to save on drive 1 of a dual drive unit. The Program will be saved permanently on disk under the name you have given it.

**Note:** When saving, you do not need to supply the '.p' suffix that Superbase uses to distinguish programs from other files.

You may wish to build a library of Programs on a separate disk. If you exchange disks (while in the Main Menu) in order to load a Program, you should re-initialize the database from the Main Menu with the 'database' command.

#### **14.5 THE LABELS PROGRAM**

On your Training disk there is an application called 'Labels'. When executed, this program prints multi-column labels. See Appendix E for full details of how to execute "labels".

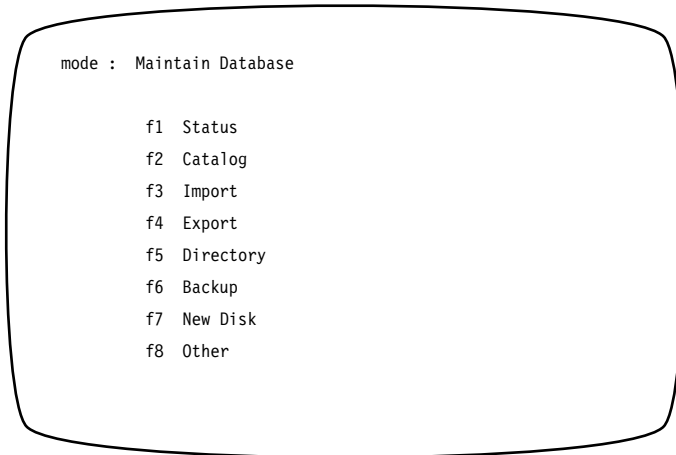
**15     MAINTAIN****15.1   THE MAINTAIN OPTION**

The Maintain Option provides a series of utilities for performing various operations on your databases. It allows you to obtain either a list of the fields in your current file together with their types and lengths, or a list of all the files in the current database. It also enables you to transfer data to and from an external program, and to carry out various disk commands including a single-drive back-up, a disk formatting command and a disk directory command.

**15.1.1 OBTAINING THE MAINTAIN OPTION**

From Menu 2 press the 'f6' key.

You should see a further menu screen. This is the 'Maintain' Menu.



**THE MAINTAIN MENU**

The Maintain Menu offers eight sub-options each of which is detailed below alongside the function keys required to obtain them. Certain items are covered more fully later in this section.



**STATUS**

This option provides a list on the screen of each of the fields of the current file along with their field-types and their lengths. The output may be directed to the printer.



**CATALOG**

A list of all of the files in the current database can be obtained with this option. This database CATALOG is the same as that provided by the FILE Option (see Section 9) when you want to change the file you are working on. The output may be directed to the printer.



**IMPORT**

This option allows data from external programs to be read into Superbase. Deselect with CONTROL-Q.

This may be data from another Database Management System you wish to convert to Superbase or it may be data from a different type of program altogether. You can also re-import data that has been exported by Superbase as a means of transferring files from one of Superbase's databases to another. See below for details.



**EXPORT**

The EXPORT Command is used to produce a SEQUENTIAL File of data from any Superbase file which can be kept as an ARCHIVE copy to be stored on disk or tape, or can be IMPORTED back into Superbase as a means of transferring data from one Superbase database to another. Deselect with CONTROL-Q. See below for details.

These sequential files can also be used by external programs. To output data to a word processor other than Superscript 128, use the command OUTPUT T0, Reference Section 8.11. For details of direct integration of the database with Superscript 128, see Programming Section 1 and the SUPERScript command.

**DIRECTORY**

A list of all programs and files on a disk can be generated by this option. The number of blocks free is also shown.

**BACKUP**

This option allows you to make a back-up copy of your data disks. For full details see Section 15.3. Deselect with CONTROL-Q.

**NEW DISK**

Formatting a blank disk or a disk you wish to re-use can be done by means of this option. In this way you can avoid having to leave Superbase to type in the Commodore disk formatting command. Deselect with CONTROL-Q.

**OTHER**

Other Commodore disk commands can be executed by means of this option. Deselect with CONTROL-Q. Examples are renaming a file, copying a file and deleting files from the disk. For full details see Section 15.5.

**15.2 IMPORT AND EXPORT**

The IMPORT and EXPORT commands transfer data between a Superbase file and a sequential file (Commodore ASCII format) on your disk. A sequential file is a sort of common currency in which information can be accepted by another program. To carry out IMPORT or EXPORT on a single drive both the Superbase file and the sequential file must be on the same disk. On a dual drive system data can be transferred from one drive to another by addressing the appropriate drive e.g. '1:filename'.

**15.2.1 IMPORT**

The IMPORT command is used to transfer data from a sequential file on disk into a Superbase file. It has two primary purposes: to transfer data from an existing database or mail list into Superbase and to transfer files across databases within Superbase itself.

In the first situation the sequential file must be created in a form in which it can be read by Superbase. Superbase normally accepts a carriage return, chr\$(13) at the end of each field and record. In some cases, conversion must be done prior to the IMPORT operation. For example, certain screen code files must be processed to remove double quotes, strip trailing spaces, and change the end of line marker, chr\$(31) to a carriage return, chr\$(13). Public domain software or the Superscript word processor on Commodore 2/3/4/8000 series and Commodore 64 may be used to create correct ASCII file format.

When transferring files between databases in Superbase, the file is first created as a SEQUENTIAL FILE with the EXPORT command.

On selecting IMPORT from the Maintain Menu you will be prompted with 'Enter Import Filename'. If you have a single drive and are importing a sequential file in which the fields of data are separated only by RETURN'S, type just 'filename' and press RETURN. If you have a dual-drive system, you can type '1:filename' to specify that the source file is on a disk in drive one. If the source file had separators between its fields and records other than 'RETURN', you can also specify these on the Command Line from the Main Menu as in 'import "filename", "/".' which specifies slashes between fields and full stops between records.

When transferring files from 4040 to 1541 format, use the BACKUP Option to read a 4040 disk and copy it to a 1541 disk. Use the same technique when transferring from 1541 to 4040.

**Note.** The disk error light (and tone if fitted) will signal extensively during the IMPORT operation. This does not normally signify an error.

### **15.2.2 EXPORT**

The EXPORT command is used to create a sequential file version of the current file. The file created this way may then be read by other programs or used as 'archive' or long-term backup.

A disk file created by Superbase consists of items of information separated by markers. These are called "End of Field" and "End of Record" markers. They are used by the systems to determine where the end of a field or the end of a record occurs. Whenever EXPORT or IMPORT are used without defining end of field or end of record markers the system uses a RETURN (chr\$(13)).

On selecting the EXPORT command you are prompted 'Enter Filename'. Enter the name you want for the EXPORT file. Used this way, EXPORT will export ALL of the fields from ALL of the records.

The data is exported in the order of the record format. If you want to change the order of your fields, use the 'OUTPUT TO' command (see Section 8.11).

The way the EXPORT command works is as follows. First it opens a SEQUENTIAL file on the disk. It then writes into that file data from the first field of the first record of the current database file. It adds a RETURN after the first field as an 'End of Field' marker and does the same for each field in the record.

At the end of the first record it adds an 'End of Record' marker. Then it selects the next record and repeats the procedure until the last record in the database file has been processed.

If you wish to separate fields and records with a character other than RETURN, the command must be on the Command Line in the form: 'export "filename","/.'" The fields in the example are separated by slashes and the records by periods (full stop). An ASCII code such as chr\$(10), not inside quotes, may be used as a separator if you prefer. If you give your EXPORT file a 'h8' or 'h' prefix, then you will be able to call it up to see the contents by selecting the HELP option and entering the filename.

If you wish to EXPORT only some of the records in a file, first create a list with FIND and use the command EXPORT FROM on the Command Line in the form: 'Export from "listname" to "export filename"'.  
.

### **15.2.3 QUICK GUIDE TO IMPORTING AND EXPORTING**

As this topic is rather difficult to understand, we have provided a step by step guide to creating a sequential file and importing it.

#### **HOW TO EXPORT**

1. Select the file you wish to export.
2. Print out the Status of the file (from Maintain sub-menu) so that, if you later wish to import the data, you'll have the information you require at your fingertips.
3. Select 'Export' from the Maintain sub-menu.
4. You are prompted 'Enter Export Filename'. To export all records from the currently selected file, enter a unique name such as "transrec," and press RETURN.
5. Superbase exports the records one by one, and displays the number on the command line.
6. If Superbase finds a corrupt record, it displays the message 'Data Mismatch'. Press RETURN to bypass this record, so that the exported file contains only "clean" records, or press RUN/STOP to abort.
7. When exporting is complete, you are returned to the Maintain sub-menu.

**HOW TO IMPORT**

1. Select or create the file you wish to import data to.
2. Display the file Status and make sure that its format is compatible with that of the sequential file to be imported. It must have the same number of fields, in the same sequence, and of the same type as those of the records to be imported. If it does not, select the format mode and amend it.
3. If you have used 'OUTPUT TO' to create your sequential file, you MUST add a temporary text field (call it "dummy") at the end of your format to receive the extra blank line that 'OUTPUT TO' adds to each record. The "dummy" field can be removed after the import operation is complete.
4. Select 'import' from Maintain sub-menu.
5. You are prompted 'Enter Import Filename'. Enter the name of the sequential file you wish to import, in this case "transrec," and press RETURN.
6. Superbase will then import the data from the file on disk into the currently selected file.

**15.3 BACKUP**

The BACKUP Option allows you to make duplicate copies of a disk. All files on a disk are copied. BACKUP handles both single and dual disk drives. Please note that two single disk drives do not constitute a dual disk drive.

Single drive backup can be done with a 1541, 1570, or 1571 disk drive, and provided a suitable interface is installed, with a 2031, SFD1001, or drive 0 of a 4040, MSD, 8050, or 8250 dual drive unit.

Owners of a hard disk unit should use the EXPORT Option to create a backup data file on the hard disk, which may then be copied to floppy disk or tape using DOS utilities. Superbase file formats, text files, programs, and lists may be copied freely.

The disk to be copied is referred to as the SOURCE DISK, and the disk onto which the data is to be copied is referred to as the DESTINATION DISK.

When BACKUP is selected the first question asked is:

**Single Drive? (y or n)**

Respond 'y' or 'n' as appropriate.

### **15.3.1 SINGLE DRIVE BACKUP**

The first screen prompt is:

**Insert Blank Disk in Drive 0**

**Press Return to Continue**

Insert the destination disk in drive 0 and press RETURN. The next prompt is:

**All Data on Drive 0 Will Be Destroyed**

**Are You Sure?**

This allows you to double-check that the disk you are using as the destination disk is the correct disk. It can be either a new disk from the box or an old disk that you want to re-use. When you are satisfied that the destination disk is correctly inserted, press 'Y'.

### **PREPARE DISK**

The next prompt is:

**Enter Disk Name, id**

You must give your disk a name and identification code. The name can be any combination of up to 16 characters, including spaces but excluding : , \$ " and @. We recommend that you use a meaningful name such as 'work disk'. The identification code must be 2 characters, such as 'aa'. Give every disk a different code, 'ab', 'ac', etc.

Type the disk name and code, including a comma between them, and press RETURN. The message 'Processing' will be displayed while the disk is prepared for use.

### **INSERT SOURCE DISK**

When the disk is ready, the flashing message 'Insert Source Disk' will appear. Remove the newly prepared disk, insert the source disk into the drive, and press RETURN.

The message 'Please Wait' appears with a line of dots following it. The line extends dot by dot as Superbase reads in the data to be copied.

**INSERT DESTINATION DISK**

When Superbase is ready, the flashing message 'Insert Destination Disk' will appear on the screen. Remove the source disk and insert the destination disk. Press RETURN. Again the message 'Please Wait' appears with a line of dots showing that Superbase is writing data onto the destination disk.

**REPEATING THE CYCLE**

If there is a lot of data to be copied, Superbase may require the process to be repeated, and several cycles may occur. The message 'Insert Source Disk' will appear again if the cycle must be repeated. In this case, go through the actions as before, starting at the INSERT SOURCE DISK paragraph.

**CONCLUDING BACKUP**

Superbase 64 and Plus/4 only: When all data has been copied, Superbase will close down, resetting the computer to the state it was in just after it was switched on. To continue using Superbase, reload it in the normal way.

**15.3.2 DUAL DRIVE BACKUP**

The first screen prompt is:

**Insert Blank Disk in Drive 1**

**Press Return to Continue**

Insert the destination disk in drive 1 and press RETURN. The next prompt is:

**All Data on Drive 1 Will Be Destroyed**

**Are You Sure?**

This allows you to double-check that the disk you are using as the destination disk is the correct disk. It can be either a new disk from the box or an old disk that you want to re-use.

When you are satisfied that the destination disk is correctly inserted, press 'Y'. Superbase will duplicate the disk on drive 0 onto the disk in drive 1.

For next (and final) prompt, see 'Prepare disk' above.

**15.4 NEW DISK**

This option allows you to format a disk. A disk formatted with this process will be suitable for storing a database and the files, lists,

and programs that go with it. However, the supplied HELP screens and the 'start' program will not be present. If you require a disk to have these on it you should create a new disk with the facility available when Superbase is started up.

The process is like the first stage of the BACKUP process.

The first screen prompt is:

**Insert Blank Disk in Drive 0**

**Press Return to Continue**

Insert the destination disk in drive 0 and press RETURN. The next prompt is:

**All Data on Drive 0 Will Be Destroyed**

**Are You Sure?**

This allows you to double-check that the disk you are using as the destination disk is the correct disk. It can be either a new disk from the box or an old disk that you want to re-use. When you are satisfied that the destination disk is correctly inserted, press 'Y'.

The next prompt is:

**Enter Disk Name, id**

You must give your disk a name and identification code. The name can be any combination of up to 16 characters, including spaces but excluding :, \$ " and @. A meaningful name such as 'work disk' is advised. The identification code must be 2 characters, such as 'aa'. Give every disk a different code, 'ab', 'ac', etc.

Type the disk name and code, including a comma between them, and press RETURN. The message 'Processing' will be displayed while the disk is prepared for use. During the process the current database is deselected. To restart, type 'database' and press RETURN. Then enter the database name, without quotation marks.

## **15.5 USING THE DISK COMMANDS**

The 'OTHER' option allows you to enter most Commodore disk commands without leaving Superbase. The following commands are permitted:

<b>rename</b>	<b>initialize</b>	<b>new</b>	<b>scratch</b>
<b>copy</b>	<b>validate</b>	<b>duplicate</b>	<b>directory(\$)</b>

Do NOT use the 'Reset' disk command (u0).

See your Commodore disk drive manual for further details.

**RENAME**

The RENAME Command renames an existing file. The new name specified in the command must not already exist. If it does, 'FILE EXISTS' error message will be displayed. The format of RENAME is:

**r0:newname=0:oldname**

If you rename a database, it must be in UPPER CASE letters. Do not rename a file format definition. The database would not be able to identify the records belonging to that file. Be sure to rename Programs with a '.p' suffix, and HELP screens with 'h8' or 'h' as the first letter. These rules also apply to any renaming done with the COPY command.

**COPY**

The COPY Command allows you to create multiple copies of files (under different names if they are to be on the same disk). Do not attempt to make a copy of a database: the only way to do this is with the EXPORT and IMPORT commands. The format of COPY is:

**c0:newname=0:oldname**

or, to join files together:

**c0:newname=0:file1,file2,file3**

**SCRATCH**

The SCRATCH command is for removing unwanted files from the disk. The format of SCRATCH is:

**s0:filename**

You will be asked Are You Sure?'

**SCRATCHING A DATABASE FILE**

To completely remove a data file from the database you must first SELECT DELETE all the records in the file and then do an extra SELECT DELETE to delete the file from the database catalog. Then, and only then, scratch the file definition using the SCRATCH command. A utility has been provided to help with this process -- see Appendix E.

**SCRATCHING A DATABASE**

A database may be scratched from within Superbase. You must first remove all the file formats from the disk using SCRATCH, then use the command again as in:

**s0:DATABASENAME**

You may wish to remove the TRAINING database from your disk and create an uncluttered source disk.

**Note:** Always insure that you select a new database BEFORE you scratch one.

16 **MEMO**16.1 **USING THE MEMO OPTION**

The MEMO Option is a text editor. You can use it to store and edit documents, and print them with the CONTROL-P screen dump facility. Memo is also used to create text screens for use in programs or as Help.

16.1.1 **OBTAINING THE MEMO OPTION**

From Menu 2, press the 'f7' key.

You should see a blank screen with the words: 'Enter Name' in the message area at the top of the screen.

You may call your MEMO by any name you choose up to 16 characters long, but if you want it to be a HELP screen, recallable by means of the HELP option, then you must prefix the name with an 'h8' if in 80 column mode or 'h' if in 40 column mode.

For example, if you want to create a HELP screen which offers advice on the IMPORT facility of the MAINTAIN option (see Section 15.2), you should call your MEMO 'h8import'.

Type in the name you have chosen and press RETURN.

If a MEMO already exists with the name you have typed, that MEMO will now appear on the screen for you to read or to edit with the flashing cursor at the top left.

The words 'Mode: Memo Writer' will be displayed in the message area above the screen.

If there is no MEMO with that name, the screen will remain blank and the cursor will appear ready for you to create it.

The following control keys are available for use while creating your MEMO screen:



These keys are used to move the CURSOR to the LEFT or RIGHT across the screen.



These keys are used to move the CURSOR UP or DOWN the screen.



This key takes the cursor directly to the 'HOME' position at the top left of the screen.



Characters can be INSERTED or DELETED with this key.



If you wish to DELETE a line from the memo screen use this key. All subsequent text will be moved up to fill the gap left by the deleted line.



You can also ERASE a line without moving up the text which follows it. Use this key to erase the line and whatever text was on that line will be replaced by blank spaces.



Use this command to print the current MEMO screen.



Lines can be INSERTED into the Memo screen with this key. Subsequent lines will be moved down to make room for the new line that has been inserted.



If you want to CLEAR the whole screen to start again, use this key.



You can QUIT the Memo option and return to the Main Menu by using this key.

Any changes you have made to an existing memo or any new memo you have created will be disregarded.

This option is mainly for returning to the Main Menu if you have entered the Memo option to read an existing Memo.



When you have completed the Memo screen, use this key to STORE the memo.

The word 'Finished' will appear in the message area and after a short delay you will be returned to the Main Menu.

## **16.2 MULTI-SCREEN MEMOS**

Memos may be multi-screen. Create each page separately with a different name. Use the disk copy command (see Section 15.5) to join the files together:

```
c0:newfile=0:memo1,memo2,memo3
```

Up to four files can be linked at a time, but you can repeat the process as many times as you wish to build up long text files. The resulting file can be edited by moving forward from page to page with the normal finishing command.

### **CAUTION**

Use CONTROL-Q and f1 Q with care, as quitting from MEMO erases the current and all subsequent screens.

## **16.3 40/80 COLUMN FORMAT**

When a Memo is created, it is affected by the screen configuration used at the time. A MEMO created on a 40 column screen will be 40 column and one created in 80 columns will be 80 column.

## 17 HELP

### 17.1 HELP SCREENS

The HELP Option enables you to display special screens for advice on the use of Superbase facilities.

Superbase has a Help screen already provided for each of its major options, but you may design your own by using the MEMO Option (see section 16).

The first line of the built in HELP SCREENS follows a standard pattern. On the left a reference is given to the User Manual for further details. On the right the name of the Help Screen is given in capitals followed by the page number you are viewing.

You may obtain a printed copy of any Help Screen by holding down the CONTROL key and pressing 'P'. Before you do this, enter the direct command 'tlen 50' -- and enter 'tlen 60' after completion.

#### 17.1.1 OBTAINING THE HELP OPTION



OR



From either Menu 1 or 2, press the 'f8' key, or if you have a Commodore 128 or Plus/4, the key marked 'HELP'.

You should see the message: 'Enter which help required' in the message area at the top of the screen.

Type in the name of the option which you want advice about, e.g. 'labels', and press RETURN. Do NOT type the 'h' or 'h8' that prefixes every Help text file.

You will see the Help Screen displayed line by line. If the Help is more than one screen long, you will see the message 'Press Return for More', and then finally the message 'End of Help, Press Any Key'.

When you are ready, press any key to return to Menu 1.



## **1 PROG**

### **1.1 THE PROG OPTION**

The PROG Option is a program editor with which users can write their own Superbase programs. PROG is the means by which Superbase is revealed to be not only a Database Management System but also a powerful Applications Generator and Database Programming Language.

Many users will confine themselves to using just the Menu-driven system of control which is perfectly adequate for routine Database Management. The PROG option however, enables more advanced users to write their own PROGRAMS utilizing more than eighty additional commands that Superbase provides in addition to BASIC.

These user-written programs can be designed to automate complex sequences of Superbase operations which can then be executed with just one key, the EXECUTE key, 'f7'.

It is even possible to use the PROG Option to set up user-defined Menus to give access to these programs in the same way that the Main Menus give access to the built-in options of Superbase.

#### **1.1.1 OBTAINING THE PROG OPTION**



To obtain the PROG Option, press the 'f5' key from Menu 2. The message 'Program Writer' will appear above an editing screen.

### **1.2 USING THE PROGRAM EDITOR**

When you use the program editor, you will be working either on an existing program or on a new program.

Programs are held in a special area of the computer's memory called the program area. When you select the PROG Option, any program lines in this area are displayed for editing. If there are no program lines in the program area, you'll see a blank screen.

If you want to edit a program that's on your disk, you need to load it into the program area first. This is done with the 'load' command typed in on the Command Line or within the PROG editing area itself.

Enter:

**load "0:programname"**

and press RETURN (the 0 represents an optional drive specifier).

When you've finished editing a program, you can store it on the disk with the 'save' command, which may also be entered either on the Command Line or within the editor.

Other useful commands are 'new' and 'list'. The former clears out any existing program lines. Both can be entered either on the Command Line or in the editor.

The program editor presents an editing window of 10 double lines (maximum length 79 or 159 characters) of program text. This may be scrolled up and down with cursor control keys. Programs are written by entering line numbers and Superbase commands in the desired sequence. A knowledge of the BASIC programming language will help the user to construct effective programs. The editing facilities of PROG are described in the next section. When a program has been written, exit from the editor with CONTROL-Q. The program will be held in memory, and may be executed by pressing the f7 'Execute' key.

To save a program after exiting from PROG (i.e. while at the main menu), use the command 'save "programname"' (omit the .p extension -- this is supplied automatically). Specify a "1:" prefix if you wish to store the program on drive 1 of a dual drive unit.

### **1.2.1 EDITING FACILITIES**

<b>Vertical scrolling</b>	Use the cursor up and down arrow controls.
<b>Horizontal movement</b>	Use the cursor left and right arrow controls.
<b>Insert and delete</b>	Use the INST/DEL key.
<b>Abbreviate commands</b>	Optionally use short form commands which will be expanded when the program is listed.
<b>Redisplay screen</b>	Use SHIFT/RETURN to redisplay after changing order of line numbers, etc.
<b>End of line above</b>	Use the back-arrow key '<-' to move the cursor to the right-hand end of the previous line.
<b>Create work space</b>	Use SHIFT/CLEAR to obtain a clear screen for writing program lines. This command does not delete existing lines, which may be scrolled into view. Line numbers are automatically placed correctly when program is listed.
<b>Insert blank line</b>	Press 'f1' followed by INST to insert a blank line at the cursor position.
<b>Copy line</b>	Change the line number to the new line number and press RETURN. The old line will still exist.
<b>Delete line</b>	To delete one program line, type the line number and press RETURN.

Superbase 128 only: To delete several lines, type 'delete <line number> - <line number>'.

**List program**

Type the command 'list' or 'prog', without a line number, and press RETURN. You can also type 'prog <line number>' to see a listing from the entered line number. (The '<' and '>' are not actually typed in.)

**Run program**

Type the command 'run' or 'execute', without a line number, to run the program. You can also type 'run <line number>' to start execution at the specified line number. At the end of the program you will be returned to the Main Menu unless you specify the instruction 'PROG' as the last line of the program, which will return you to the Program Writer.

**Invalid line**

All lines entered must have a line number except when using the direct commands 'list', 'prog', 'run', 'execute', 'load', 'save', 'delete', 'protect', 'new', and 'renumber'. The error message 'Invalid line: Re-enter' usually indicates a missing line number or a command syntax error. The highest line number allowed is 63999.

**Check available memory**

On the Commodore 64, up to 4K is available for programs and variables. The Plus/4 has up to 8K unless graphics are being used, in which case, the program area is reduced to 2K. On the Commodore 128, up to 62K can be available. The command 'display fre(0)' tells you how much space is left for program code, and, on the Commodore 128 only, 'display fre(1)' shows you how much variable space is available (type 'clr' first to clear out existing variables).

**Print program listing**

The best way to obtain a hard copy program listing is to return to the Main Menu and type 'print: list: display'.

Superbase 128 only: To list a Superbase program to disk, for example so you can edit it with a word processor, enter the following command line:

**open "listfile":list:close**

Note that you can't reverse the process and use a word processed listing as an actual program.

### **1.3 THE SUPERBASE PROGRAMMING ENVIRONMENT**

Superbase is a complete operating environment and does not allow other programs to co-exist, except for I/O data from certain approved interface products. It is therefore not relevant to the user to know how much space Superbase uses in the machine -- but for your information, Superbase is about 40K of machine code.

#### **1.3.1 SUPERBASE PROGRAMS: MEMORY SIZE AND CHAINING**

(Superbase 128 only)

Space for program code is over 9K, but this may be varied with the 'load ... at' command, up to about 11.5K.

When Superscript is not present, variable space is 55K. When it is present, variable space ranges from 12K down to 1K, adjusted with the 'varspace' command.

There are techniques for producing suites of program modules. The 'set' command reads variables from a text file. 'Load' has a flexible syntax and can cause one module to execute another or append one module to another in memory. See individual sections on 'load', 'set', and 'varspace'.

#### **1.3.2 IMPORTANT BASIC COMMANDS**

Some of these commands have been mentioned elsewhere in the manual. This summary brings them together in one place for easy reference, but you should study your BASIC manual to learn the full syntax.

<b>clr</b>	Clears out program variables.
<b>delete</b>	Removes program line numbers. (C128 only)
<b>list</b>	Displays or prints (whichever is in effect) the current program.
<b>new</b>	Clears out any program in memory.
<b>renumber</b>	Renumbers a Superbase program. (C128 only)
<b>run</b>	Executes the current program.

### **1.4 CREATING PROGRAMS**

In the sections which follow the creation of programs will be explained and you will be introduced to some elementary programming techniques. All the examples refer to the "cust.inv" and "cust.rec" file formats supplied on your disk.

It will be explained how a program is developed out of a sequence of COMMAND LINES and how such devices as LOOPS and CONDITIONALS can be utilized to make your programs more versatile.

If you are unfamiliar with the commands used in these sections which do not appear on the Main Menu you should refer to the section on ADDITIONAL COMMANDS, Programming Section 2.

#### **1.4.1 PROGRAMS AND COMMAND LINES**

In Reference Section 3 the notion of a COMMAND LINE is explained. A Command Line is one or more Superbase commands separated by colons (:).

A PROGRAM is essentially nothing more than a sequence of such Command Lines where each has been given a LINE NUMBER determining the order in which the instructions are to be carried out.

In fact Programs can be created without the use of the PROG Option at all, by typing in Command Lines from either Main Menu as normal, but with a line number in front of them as in:

**100 find "update-list" where [update] is "=yes"**

(Type a space to get to the Command Line.) This could then be followed by another such as:

**200 output the records from "update-list"**

All we need to make these two lines into a simple program is another line to signify that the program has ended:

**300 end**

The reason for numbering lines in hundreds is to allow plenty of room for further lines should you want to edit the program later with the PROG Option.

Having typed in a sequence of numbered Command Lines in this way you then have a program resident in memory which can be executed with the 'f7' key from Menu 1, or from within PROG with the command 'RUN'.

Programs are only held in the computer's memory until the power is switched off unless they are stored on disk with the SAVE command.

To store a program you have created in the PROG Option, use the SAVE command from the Main Menu, as described above.

#### **1.4.2 PROGRAM LOOPS**

The program created in the last section simply constructs a Key List of records that need updating and displays them. Suppose however that we wanted to actually carry out operations on these records such as adding the contents of one field to the contents of another.

We might have a field called 'Last-paid' which we want to add to the contents of a field called 'Total-paid', as in the CUSTOMER INVOICE file "cust.inv" supplied on the disk.

The Command line for this calculation is:

**calc [Total-paid]=[Total-paid]+[Last-paid]:store**

Replacing line 200 in the three line program above with this Command Line however, would only result in the calculation being performed on the current record. To carry out the operation on all of the records in the Key List requires what is known as a PROGRAM LOOP.

A Program Loop consists of two lines, the second of which returns to the first, in between these two lines will be a number of other Program Lines which carry out the operations required. These lines constitute the MAIN BODY of the loop.

For example, in the following program:

```
100 find "h8list" where [update] is "=yes"
200 select from "h8list"
300 calc [Total-paid]=[Total-paid]+[Last-paid]:store the
      record
400 goto 200
```

lines 200 and 400 constitute the loop itself and line 300 is the body of the loop. (Note that this program as it stands will loop indefinitely.)

Line 200 tells the computer to select a record from the Key List, line 300 specifies an operation to be carried out on that record and stores the modified version and line 400 tells the computer to go back to line 200 to select another record for processing.

In this way all of the records in the list would be processed.

A word of warning. The 'goto' command is very useful but can if overused become a source of great difficulty when debugging long programs.

### **1.4.3 CONDITIONALS**

The only thing wrong with the above program is that once the computer enters the loop it will never get out unless the STOP key has been enabled and is used, or the power is switched off.

What is needed is a Program Line in the body of the loop which will return to the Main Menu once all of the records in the Key list have been processed.

For this purpose we can include another line:

**350 eol Menu**

The command 'Eol' which stands for 'End of list' is what is known as a CONDITIONAL since the command following it (in this case the MENU command which returns us to Menu 1) is only executed on the condition that EOL is satisfied. EOL is satisfied whenever the end of the list has been reached.

Any Command Line can follow EOL, as in:

**eol display "All records processed":wait**

or

**eol output the records from "h8list"**

As in our original example, these Command lines will not be executed unless the end of the list has been reached.

#### **1.4.4 OTHER CONDITIONALS**

There are a number of other useful conditionals available. See the entries in Programming Section 2 for full details.

##### **EOF - END OF FILE**

The first of these, EOF is similar to EOL except that it is satisfied (i.e. the rest of the Command Line following it is activated) when the end of the file has been reached.

EOF would be used when ALL of the records in the file were to be processed as in:

```
100  select the first record
200  [Total-paid]=[Total-paid]+[Last-paid]:store the record
300  select the next record
400  eof menu
500  goto 200
```

**IF ... THEN**

Another useful conditional is the BASIC command 'IF...THEN...', as in:

```
300 if [Outstanding] > [Credit-limit] then display  
    [Name] [Address] [Outstanding]
```

This Program Line would display the requested information only IF the particular customer owes more than their credit limit allows.

Any condition can follow the 'IF' and any command may follow the 'THEN', so the general format of the IF...THEN... command is:

```
if <condition> then <command>
```

**ELSE**

In Superbase 128 and Plus/4, ELSE can be used to extend the power of the conditional, as in:

```
300 if [Outstanding] > [Credit-limit] then display  
    [Name] [Address] [Outstanding] : else goto 1000
```

Any condition may be set. If it is evaluated as being 'true' the THEN option is executed. If the condition is 'false' the ELSE option is executed instead. The format is:

```
if <condition> then <command> : else <command>
```

Note that ELSE must always be preceded by a colon.

The ELSE command may be used in other contexts, for example with EOF and EOL (see above) or with NMAT and PMAT (see below).

**NMAT AND PMAT**

The form for these commands, which test to see whether a match has been achieved after an index search, is NMAT <command> : ELSE <command> and PMAT <command> : ELSE <command>. The second option is executed only when the NMAT and PMAT conditionals have not been satisfied.

The commands following NMAT will be executed only if no match occurs. Similarly, the commands following PMAT will be executed only if a partial match occurs.

The NMAT and PMAT conditionals are also used with the 'link' command.

## 1.5 LINKING BETWEEN FILES

Another useful facility is the set of commands SETLINK, LINK, RLINK and ELINK which enable you to update a record with information from a record in another file. They work like this:

```
SETLINK  links to the file
LINK     links to a record in the file
RLINK    comes back to the original file and record
ELINK    ends the link leaving linked file as current file
```

1. Select file A.
2. Specify a link to file B (setlink).
3. Process file A.
4. Use any key from file A as the index key to a record in file B (link). This record is now the current record.
5. Process file B. You can update it with data from file A that has been temporarily stored in variables.
6. Terminate the link (rlink) and return to file A.
7. Carry on processing file A.
8. At the end of your processing, you can make file B the selected file (elink).

Suppose that we wanted to add the contents of the 'Last-paid' field in the Customer Invoice file to the contents of the 'Balance' field in the Customer Records file.

The following would suffice:

```
100  file "cust.inv"
200  setlink "Cust.Rec"
300  find "h8list" where [update] is "=yes"
400  select from "h8list"
500  eof menu
600  [Total-paid]=[Total-paid]+[Last-paid]
700  store the record
800  a=[Last-paid]
900  link [ref]
1000 [Balance]=[Balance]+a:store
1100 rlink
1200 goto 400
```

Note that line 200 sets the link to the Customer Records file, the link remaining in force throughout the program. Line 800 stores the amount last paid in a BASIC variable. Line 900 links to the record in the Customer Records file whose key is the same as the Customer Reference field in the record we are linking from. Line 1000 adds the value stored in the BASIC variable 'a' to the Balance field in the record we have linked to. Line 1100 returns from the file that was linked to, and line 1200 returns us to select the next record key from "h8list".

We could improve this updating program with the following line:

```
650 [update]="no"
```

This modifies the Update field to show that the record has been updated.

(Editing hint: while testing a program, insert a temporary line containing the commands 'wait:prog'. This returns you to the editor automatically.)

## **1.6 ADDITIONAL PROGRAM FEATURES**

### **1.6.1 DIRECT SELECTION BY KEY**

Note that in programs you can 'select' a record directly using any of the forms: 'select a\$', 'select "jones"', or 'select [name]'.

### **1.6.2 DELETING A GROUP OF RECORDS (C128 only)**

First, use 'find' to create a key list of the records you want to remove from the file, which should be the selected file. Then use the following program to delete the records on the list:

```
100 select from "" : eof menu : else select d : goto 100
```

If you want to delete all the records in a file, use this program:

```
100 select f : eof menu : else select d : goto 100
```

## **1.7 INTEGRATION WITH SUPERSCRIP 128**

You can use the 'superscript' command to access the Superscript 128 word processor from within Superbase. This command allows you to have direct access to database files, to specify which documents to use, and to determine how they are processed. You can also specify the use of a Key List to process a sub-group of records from the current database file, and whether to return to the database program after the word processing operation or to remain in Superscript.

### **1.7.1 LOADING SUPERSCRIP WITH SUPERBASE**

If you want to use the two programs together, you must load Superbase from within Superscript. If necessary, quit from Superbase and load Superscript, then use the command 'f1 Document Utility Superbase'. You will be presented with the usual Superbase start up menu. Proceed as normal -- you now have the 'superscript' command available to achieve integration. If you try to use the 'superscript' command when Superscript 128

is not present, you will see the error message 'Superscript not available'.

### **1.7.2 THE 'SUPERSCRIP'T' COMMAND**

The 'superscript' command has three parts: the command word 'superscript', an optional 'all' or 'from "listname"', and an optional command string which must be preceded by a comma.

#### **THE COMMAND WORD**

This is entered like any other Superbase command, abbreviated to 'suP' if you prefer. In many cases, 'superscript' will be part of a Superbase program, but you can also type it directly on the Command Line. Enter 'superscript' followed directly by RETURN if you want to switch straight into Edit mode.

#### **'ALL' OR 'FROM "LISTNAME"'**

If you omit this part of the command, Superscript assumes that the only available part of Superbase is the current record, and executes the command string, if you've included one (see below). Any Superbase fieldnames you may have in a document that you output with "merge" will accept data only from the current record.

If you include 'all', a Superscript mail merge will use all the records in the currently selected database file. If you include 'from "listname"', a Superscript mail merge will use only the records in the current file whose index keys are in the list. Lists are created with the 'find' command.

#### **THE COMMAND STRING**

If there's no command string, you go into Edit mode. The command string can be a maximum of 127 characters long. It works exactly like a Superscript phrase glossary command string, so you should study the appropriate sections of the Superscript manual to see how a command string executes Superscript menu options and/or places text into a document.

#### **RETURNING TO SUPERBASE**

When you want to return to Superbase from Superscript, either put "/qy" at the end of the command string (see examples below), or from Edit mode give the Superscript sequence for the normal 'quit' command: 'f1 q y'.

#### **RESTORING THE FULL TEXT AREA**

You can restore the full Superscript text area at any time with the Superbase command 'quit'. See Section 2.1.41.

### **1.7.3 EXAMPLES**

The main use of integration is to allow direct access to the database for mail merge operations. The logical steps in such an operation are:

1. Select a database and a file.
2. Decide on whether you want to use the whole file, a Key List, or just the current record.
3. Decide which already existing merge document you want to use.
4. Insure that both the document and the Key List, if used, are available on the disk that's in the drive.
5. Give the 'superscript' command, specifying the document and the Key List if you're using one.
6. Your command string can include a return to the database (see below).

#### **COMMAND LINE EXAMPLE**

A typical Command Line use of 'superscript' is:

**superscript from "","/dlmergeletter.1^m/pmp"**

This uses "h8list" (or "hlist") to select records from the database using the previously created Key List.

The first part of the command string, up to "^m", instructs Superscript to load a document called "mergeletter.1" with the Superscript command sequence 'f1 Document Load'. The "^" is obtained by pressing the 'up arrow' key. The "^m" represents pressing RETURN to enter the document name.

The second part of the command string, "/pmp," instructs Superscript to carry out a mail merge with the Superscript command sequence 'f1 Print Merge Print'. Following the merge print run you would be left in Superscript's Edit mode.

#### **PROGRAM EXAMPLE 1**

A more sophisticated example uses database commands to 'find' a Key List, requests input of a merge document, does the print run, and returns to the database, where it displays the number of records printed:

```
10  database "accounts" : file "invoices"
20  ask "name of list" ; list$
30  ask "region to search for" ; region$
40  ask "amount-due to search for" ; amount
50  amount = amount - 1 : rem allows 'greater than' search
```

```

60 find list$ where [region] is region$ ; [amount-due] is
   ">" + str$(amount)
70 count x : if x=0 then display "No records found for
   these criteria" : wait : menu
80 ask "name of merge letter" ; letter$
90 superscript from list$, "/"d1" + letter$ + "^m/pmp/qy"
100 display "Number of letters printed: " ; &3,0x
110 wait : menu

```

## PROGRAM EXAMPLE 2

This program uses database commands to create an ASCII disk file that is then loaded into a preformatted financial statement (the numeric tabs must correspond to the decimal points in the output disk file) and printed after Superscript has added up the columns; the total for the bottom line is stored in a block, read back by the database, and displayed on screen.

```

100 file "sales
199 rem CREATE DISK FILE FROM DATABASE
200 output all across to "stmt"[product][amount1][amount2]
   [amount3][amount1]+[amount2]+[amount3]
299 rem GO TO SUPERSCRIPT, LOAD "form", APPEND "stmt", ADD ALL
   COLUMNS, PRINT
300 superscript ,"/dlform^m^g^m^m/dlstmt^m^g^m^m/ct^m/ppc/qy"
399 rem GO TO SUPERSCRIPT, ADD TOTALS WITH CALCULATOR, PLACE
   IN TEXT, DEFINE AS BLOCK, AND STORE ON DISK
400 superscript ,"^ki/cz/ca^i^r^i^r^i^r/eapam=cr^m^b/ab^u^u
   ^u^u^u^u^u^u^u^u^u^u^u^u^u^u^u^u/dbtemp^m/qy"
500 clr:set "temp
600 display "Total value is:"&7,2am
700 wait:menu

```

#### 1.7.4 THE SUPERScript DOCUMENT

To draw information directly from the database, a Superscript document must have the relevant fieldnames present in the text as markers. Do not include the square brackets. The three types of marker, variable, fixed, and conditional, can all be used.

The selection feature, in which a marker can have 'equal to' / 'not equal to' logic and pattern matching incorporated, is also available, but the Superbase 'find' command gives much greater control over record selection and we advise you to use it and then put the resulting Key List into the 'superscript' command.

If an \*nm command is present in a Superscript document to specify the structure of an ASCII merge file (i.e. a disk file, NOT a database file), it is ignored during a merge controlled by the 'superscript' command.

### **1.7.5 CHANGING THE TEXT AND PROGRAM VARIABLE SPACE**

The 'varspace' command exists so you can change the balance of text space and program variable space when Superscript is present. The maximum text space is obtained with 'varspace 1'. The minimum text space is obtained with 'varspace 12'. Each increment from 1 to 12 represents a decrease of about half an Editing screen of text space.

Programs that have many string variables or use arrays can require extra space to store all this data in memory, though the program may be only a few lines long. 'Varspace 12' gives the maximum variable space when Superscript is present.

Note that the higher the 'varspace', the greater the restriction on the length of the document Superscript can load. Also, remember that 'varspace' clears out any already set variables, so if you put it in a program, it should only be used at the beginning. It also clears any Superscript text.

## **1.8 ERROR HANDLING**

Superbase has two ways of dealing with errors: it uses BASIC error trapping messages and routines in Superbase programs, and it has a number of error messages of its own that appear when a database function malfunctions.

### **1.8.1 ERROR TRAPPING**

The full set of C128 and Plus/4 BASIC commands is available:

<b>trap&lt;line number&gt;</b>	Makes the program go to the specified line when any error occurs.
<b>er</b>	A reserved variable containing the number of the error. See Appendix B for a full list.
<b>el</b>	A reserved variable containing the line number where the error occurred.
<b>err\$(er)</b>	BASIC error messages, subscripted by the error number.
<b>resume</b>	Resume execution at last error line.
<b>resume next</b>	Resume execution at line after error line.
<b>resume&lt;line number&gt;</b>	Resume execution at specified line number.

The importance of these commands is that they allow you to create Superbase programs in which the user never gains access to the Superbase menus. This means data can be completely protected from unauthorized access -- an essential condition for accounting applications, for example.

### **1.8.2 PROGRAM ERRORS**

Errors are usually of two kinds: syntax and logic. Syntax errors occur when you don't type a command exactly right, or you forget a semicolon or quotation mark, for example. Logic errors occur when your program fails to work because you've made a mistake. Such an error may not be detected by the system directly, but it may cause another error that is detected. This can be misleading! Another type of logic error is a mathematical error such as "division by zero;" again, this is the fault of the programmer.

### **1.8.3 ERRORS IN SUPERSCRIPT**

If an error occurs in a 'superscript' command string, it will cause a normal Superscript error condition. If you called Superscript from a Superbase program, the error will be detectable when you return to Superbase.

If you have to interrupt the execution of a 'superscript' command string, for example, if the printer jams, you should use CONTROL-Q. You can resume merged output manually from the next record with the 'f1 Print Merge Print' command. When you return to Superbase with 'f1 Quit Yes', the error caused by your earlier CONTROL-Q will be detected.

**Note.** To prevent format disruption, the Superscript 'Set Mode' command is unavailable when Superscript is entered from Superbase.

## **1.9 VARIABLE AND FIELD NAME RULES**

Field Names may be up to 12 characters long, with no spaces. The characters (!), (#) and (") are not allowed.

Any word may be used in a field name.

BASIC variable names may be any length, but only the first 2 characters are significant, i.e. 'elephant\$' is the same as 'elephant\$'.

The permitted character range is a to z together with 0 to 9. The first character may not be a number.

No part of a variable name may contain a BASIC or a Superbase command. Examples of illegal names are:

do\$	newvalue\$
do%	link\$
screen\$	date\$
total	

A common and perplexing syntax error can be caused by inadvertent combinations of letters in program lines, for example:

**if t and (b=3) then ...**

produces an error because the function TAN will be detected. The solution is to use parentheses as follows:

**if (t) and (b=3) then ...**

### **1.10 PERMITTED BASIC COMMANDS**

Because Superbase incorporates many high-level commands, not all the standard BASIC commands are available. A list of valid commands follows.

#### **PRIMARY**

clr	else*	goto	on	resume*	troff*
data	end*	if...then	perform <sup>1</sup>	return	tron*
def	exit*	let	read	run	until*
delete*	for...next	list	rem	step	
dim	get	loop*	renumber*	stop	
do...while*	gosub	new	restore	trap*	

#### **OPERATORS AND EXPRESSIONS**

abs	dec*	instr*	mid\$	rgr <sup>†</sup>	sqr
and	err*	int	or	rlim <sup>†</sup>	str\$
asc	exp	joy*	'pi'	rnd	tan
atn	fn	left\$	right\$	sgn	val
chr\$	fre	len	rc <sup>†</sup>	sin	vol*
cos	hex*	log	rdot <sup>†</sup>	sound*	xor*

#### **PLUS/4 GRAPHICS**

box <sup>†</sup>	circle <sup>†</sup>	draw <sup>†</sup>	gshape <sup>†</sup>	paint <sup>†</sup>	scnclr <sup>†</sup>
char <sup>†</sup>	color <sup>†</sup>	graphic <sup>†</sup>	locate <sup>†</sup>	scale <sup>†</sup>	sshape <sup>†</sup>

\* Available in Superbase 128 and Plus/4 only

<sup>†</sup> Available in Superbase Plus/4 only

<sup>1</sup> Superbase 64 uses the 'do' command instead of 'perform'

### **1.10.1 DISALLOWED BASIC COMMANDS AND EXPRESSIONS**

append	copy	graphic	poke	spc
auto	dclear	header	pos	sys
bload	dclose	input	print#	using
boot	dopen	input#	pudef	usr
bsave	dsave	key	record	verify
cmd	dverify	monitor	rename	
collect	fast	open	scratch	
concat	go	peek	slow	

Other graphics and sound commands are also not allowed.

### **1.11 SUPERBASE GRAPHICS ON THE PLUS/4**

You may use the graphics features of your Commodore Plus/4 to produce graphics displays based on the data in your database files.

To do this you need to extract data in a suitable form for display. Usually, you will want to measure some sort of quantity against a set of categories. For example, cash book figures (quantity) by code (category); or items purchased (quantity) by product type (category).

To obtain this sort of data, use a short Superbase program. The program must perform the following operations:

1. Read through the file, perhaps using a key list.
2. For each record, add the quantity to a total for the appropriate category. You can use the REPORT option to generate upto ten totals simultaneously.
3. Display the totals in graph form, using axes and columns.

### **1.12 COMPATIBILITY**

A program written on the Commodore 64 will run on the Commodore Plus/4 and 128. However, a program written on the Plus/4 or C128 using any of the extra commands (with the exception of 'PERFORM') will not run on the C64 because the new commands will be flagged as syntax errors. This is because they are unknown and cannot be recognized as being commands.

'PERFORM' will be accepted because it is a direct translation of the C64 'DO' command. If you use a Commodore 64 to 'LIST' a program that uses the Plus/4 or C128 'PERFORM' command, you will find that this command has been translated into 'DO'. Similarly, a Plus/4 or C128 listing of a Commodore 64 program will translate 'DO' into 'PERFORM'. This is because they represent exactly the same operations and so do not pose any problems of compatibility.

## 2 ADDITIONAL COMMANDS

A number of additional commands are provided which do not appear on either of the Main Menus. These commands can either be used in COMMAND LINES typed in from one of the Main Menus or can feature in PROGRAMS written by the user (see Programming Section 1).

While many users will opt to use only the Menu driven commands of Superbase, the advanced user will find that the addition of the commands detailed in this section extend the power of the system considerably.

In the following section, every command is identified as either PRIMARY or SECONDARY (or in some cases both). Primary commands can be used by themselves or with parameters following them. Secondary commands always follow primary commands.

In the examples given, references to file formats are to the "cust.inv" and "cust.rec" formats supplied for training purposes on your program disk.

Many of these commands may be abbreviated during program development. Where appropriate, these abbreviations have been listed.

### 2.1 THE COMMANDS

#### 2.1.1 ACROSS Primary or Secondary Command

FORMAT:           <command> **across** <"string">;<[field]...>

ABBREVIATION: **aC**

PURPOSE:        To set the output to ACROSS the page or screen.

COMMENTS:       See Reference Section 8.6.

#### 2.1.2 ALL Secondary Command

FORMAT:        output **all** <<format command> <[field]...>>

ABBREVIATION: **aL**

PURPOSE:        Used with the OUTPUT command to signify that ALL records in the file are to be used for the output operation. This command is the opposite of FROM.

EXAMPLES:       output **all** the records @20 [Name][Outstanding]

                  Outputs the contents of the NAME and OUTSTANDING fields from every record in the file.

**2.1.3 ASK**

Primary Command

FORMAT:       **ask** x\$  
                 **ask** x  
                 **ask** "text";x\$  
                 **ask** "text";x  
                 **ask** [field]  
                 **ask** "text";[field]  
                 **ask** @x,y <all options>  
                 **ask** &n <all options> (where n is a number, 1-254)

ABBREVIATION: **aS**

DEFAULT:       Enter? (in command area at the top of the screen).

PURPOSE:       To allow input from the keyboard during program execution.

COMMENTS:     The text following the 'ASK' command will be displayed on the screen and whatever you type will become the value of the TEXT VARIABLE (x\$) or FIELD ([field]) specified. Do not enter double quotation marks. RETURN on its own is not accepted. Non-numeric input to numeric variables or fields will produce the 'Invalid Numeric Result' error message. The same message appears if the input number is too large for a field format, or if a result field calculation overflows its format as a result of a field parameter input with 'ASK'.

The "text" need not include the word 'Enter' as this is supplied for all prompts that do not have positioning commands @x,y.

The & option must precede all others if used; the default is 40.

This command can be used to set up user-defined Menus.

EXAMPLES:     100 display @5,5"1> Update Accounts";@5,8"2>  
                  Show Clients over Credit Limit";@5,11"3>  
                  Print Report on Sales"  
                 200 **ask** a  
                 300 if a<1 or a>3 then goto 200  
                 400 on a goto 500,600,700  
                 500 load "Program 1"  
                 600 load "Program 2"  
                 700 load "Sales Report"

Displays a Menu from which user-written programs can be accessed.

**2.1.4 BRKOFF - STOP KEY OFF**

Primary Command

FORMAT:           **brkoff**

PURPOSE:          To disable the STOP key after it has been enabled with the BRKON command.

**2.1.5 BRKON - STOP KEY ON**

Primary Command

FORMAT:           **brkon**

ABBREVIATION: **br**

PURPOSE:          To enable the STOP key to function as a means of breaking out of a program that is running.

COMMENTS:        The STOP key can be disabled again with the BRKOFF command.

**2.1.6 CHECK**

Primary Command

FORMAT:           **check "[fieldname]"**  
                     **check a\$**

ABBREVIATION: **ch**

PURPOSE:          Checks a fieldname against the file definition of the currently selected file.

COMMENTS:        Acts as a conditional. Commands on the same line as 'check' will only be executed if there is an error, i.e. if the fieldname does not exist in the current file definition.

EXAMPLE:          100 ask a\$  
                      200 b\$ = "[+ a\$ + "]"  
                      300 **check** b\$ : goto 1000  
                      400 display b\$  
                      500 stop  
                      1000 display "field does not exist"

**2.1.7 CLEAR**

Primary Command

FORMAT: **clear**

ABBREVIATION: **cL**

PURPOSE: Clears all fields allowing the current record to be used as a new record. Information can be entered into the fields with the CALC command. See Reference Section 11.

COMMENTS: After entering information into a record in this way, the STORE command must be used.

EXAMPLES: **clear:calc [Name]="Davis, A";[goods]="1 CBM 64 Computer":store**

Enters a new record with the details as shown.

**2.1.8 CLOSE**

Primary Command (C128 only)

FORMAT: **close**

PURPOSE: Closes a key list being processed by SELECT commands under program control. The next SELECT FROM command automatically relates to the first entry in the list.

**2.1.9 COLS**

Primary Command (C128 only)

FORMAT: **cols variable**

PURPOSE: Returns the current number of columns available on the screen to a variable, e.g. x.

COMMENTS: To find whether the current screen configuration is 40 or 80 columns. Routines such as the calling up of Help screens or display procedures can be conditional upon the value of this variable.

**2.1.10 CONT - CONTINUOUS PRINT** Primary Command

FORMAT: **cont** <1 or 0>

ABBREVIATION: **c0**

DEFAULT: 0

PURPOSE: To specify whether or not you want Superbase to pause after each page has been printed.

COMMENTS: 'CONT' is normally switched on (1). To force the end of page break, set 'CONT' to 0. Printing may also be paused with the 'WAIT' command.

EXAMPLES: **cont 1**

Suppresses the end of page break.

**cont 0**

Switches end of page break back on.

**2.1.11 CONVERT** Primary Command

FORMAT: **convert** n, x\$  
**convert** [datefield], x\$

ABBREVIATION: **conV**

PURPOSE: To convert a number or numeric expression, or the contents of a date or calendar field to a named BASIC string variable.

COMMENTS: 'CONVERT' allows dates to be changed from numeric form to an easily manipulable string form. Dates are stored internally as numbers.

EXAMPLES: 100 **convert** [Date], d\$  
200 dd\$=left\$(d\$,2)  
300 mm\$=mid\$(d\$,3,3)  
400 yy\$=right\$(d\$,2)  
500 print dd\$;" / "; mm\$; " / "; yy\$

This example assumes European style dates and presents the date in a different format.

**2.1.12 COUNT**

Primary Command (C128 only)

FORMAT:           **count** x

ABBREVIATION: **coU**

PURPOSE:       Returns the number of records processed in the last operation in a variable x. Valid with FIND, BATCH, SORT, OUTPUT, DETAIL, IMPORT, EXPORT, SELECT FROM.

EXAMPLE:       100 find "" where [amount] is ">100"  
                 200 **count** x  
                 300 display "Found ";&3,0x;"records":wait  
                 300 if x=0 then menu

Finds a list, gets the number of records found, displays it and waits for input, returns to menu if no records found.

**2.1.13 DATABASE**

Primary Command

FORMAT:           **database** "name" <,unit number, drive number>

ABBREVIATION: **dA**

PURPOSE:       To select a database to work in or to create a new database.

COMMENTS:      If no database "name" is given, the prompt 'Enter Database Name' will be displayed. Enter a database name without quotation marks. If database "name" is given, it must be enclosed within quotation marks. Used within a Program, the command may transfer processing to another database without user intervention.

Defaults for unit number and drive number are 8 and 0 respectively.

Enter \$ to the prompt 'Enter Database Name' to view the disk directory and ascertain database names. Database names must **not** begin with '\$', and must not contain numbers.

EXAMPLE:       **database** "invoices",9,0

Switches to the database in drive 0 of unit 9.

**2.1.14 DATE**

Primary Command

FORMAT:       **date** x\$, n

PURPOSE:       To validate the string in x\$ as a date, returning the number of the month in the variable n if the data is valid, or 0 in n if the date is invalid.

Also used to set the date style to European (day month year) or American (month day year). The date style is set by each use of the command and date format may therefore be reversed.

COMMENTS:     Use 'Date' in conjunction with 'ask' and test for date validity. Dates must be input using alphabetic month abbreviations.

EXAMPLES:      100 ask &7a\$  
                  200 **date** a\$,n  
                  300 if n=0 then 100

**2.1.15 DETAIL**

Primary Command

FORMAT:       **detail** all/from <"text";<[field]>...>

ABBREVIATION: **deT**

PURPOSE:       To specify the main body of detail in a User-written report. Functions like OUTPUT.

COMMENTS:     Instead of writing your own Report Programs, you could choose to have them written for you by the REPORT GENERATOR (see Reference Section 13).

EXAMPLES:      **detail** from "" [ref] [quantity] [price]

**2.1.16 DISPLAY**

Primary or Secondary Command

FORMAT:       **display** <command line>

ABBREVIATION: **dI**

PURPOSE:       To set the direction of output to the screen as opposed to the printer, and optionally display any parameters following it. The command 'display chr\$(147) chr\$(7)' will clear the screen and sound the tone. Numeric variables should be separated by semicolons.

COMMENTS:     See Reference Sections 8.2 and 8.3.

- 2.1.xx DO** Primary Command (C64 only)
- FORMAT:       **do** <"string">  
                 **do** x\$
- PURPOSE:       To execute a Commodore 64 Command Line stored in a string variable or enclosed in quotation marks.
- COMMENTS:      See 'PERFORM', Section 2.1.35.
- 
- 2.1.17 DOWN** Primary or Secondary Command
- FORMAT:       <command> **down** <"string">;<[field]...>
- ABBREVIATION: **d0**
- PURPOSE:       To set the output to DOWN the page, one record at a time.
- COMMENTS:      See Reference Section 8.
- 
- 2.1.18 DUMP** Primary Command
- FORMAT:       **dump** <"<n:>filename">
- ABBREVIATION: **dU**
- PURPOSE:       To print, display, or store on disk a file of the variables and their current values from the program currently in memory.
- COMMENTS:      A file of variables can be read back into the program at a later date by means of the SET command. If no filename is given the dump will be to the screen or printer, whichever is the current output device.
- 
- 2.1.19 ELINK - END OR REVERSE LINK** Primary Command
- FORMAT:       **elink**
- ABBREVIATION: **eL**
- PURPOSE:       To reverse the link established by the SETLINK command so that the linked file becomes the current file or the opposite. No actual link is active after an 'ELINK'. The 'File Selected' indicator will show which file is the current file after the 'ELINK' by displaying '='.
- COMMENTS:      Used with the SETLINK, LINK, and RLINK commands.

**2.1.20 ENDREPORT**

Primary Command

FORMAT:           **endreport** <"text";<total variables>>

ABBREVIATION: **enD**

PURPOSE:       To specify the text and totals you want to display at the end of a report generated by a User-Written Report Program.

COMMENTS:      Instead of writing your own Report Programs, you could choose to have them written for you by the REPORT GENERATOR (see Reference Section 13).

EXAMPLES:       **endreport** "Total Paid ";t0;"Total Outstanding  
                                  ";t1

Will display the totals accumulated in the Total Variables t0 and t1 together with explanatory text.

**2.1.21 EOF - END OF FILE**

Primary Command

FORMAT:           **eof** <Command(s)>

ABBREVIATION: **e0**

PURPOSE:       Used in user-written programs containing the 'SELECT' command to carry out the stated action when the last record in the file has been reached.

COMMENTS:      EOF is a CONDITIONAL, i.e. the execution of those commands which follow it occurs whenever the end of the file has been reached.

EXAMPLES:       **eof** Menu

Will return to Menu 1 when the last record in the file has been processed.

**2.1.22 EOL - END OF LIST**

Primary Command

FORMAT: **eol** <command(s)>

PURPOSE: Used in user-written programs containing the FROM "list" command to carry out the stated action when the last record in the Key List has been reached.

COMMENTS: EOL is a CONDITIONAL. This means that the execution of those commands which follow it in the same line is conditional on it being true. EOL is true whenever the end of the Key List has been reached. EOL can be used after FIND to test whether a Key List has any keys in it.

EXAMPLES: 100 select the records from "hlist"  
200 **eol** display @1,0 "Press any key":wait  
300 **eol** menu  
400 calc a\$=[goods];a\$  
500 goto 100

After all of the records in the Key List have been processed, the program will pause until a key is pressed and will then return to Menu 1.

**2.1.23 FROM**

Secondary Command

FORMAT: <command> **from** <"<n:>list">

ABBREVIATION: **fr**

DEFAULT: "h8list" or "hlist" depending on 80 or 40 column screen mode.

PURPOSE: To specify which Key List is to be used in a Superbase operation. This command is the opposite of ALL.

EXAMPLES: select **from** "debt list"

Selects just those records whose keys appear on the Key List named "debt list".

**2.1.24 IS**

Secondary Command

FORMAT: <command> where [field] **is** <"condition">

PURPOSE: Used with commands containing WHERE which operate on those records WHERE some field IS of some value or range of values.

EXAMPLES: find "Debt-list" where [Outstanding] **is** ">0"

Creates a Key List of records of people who owe you money.

select match where [Town] **is** "=Paris"

Allows you to view just those records of people who live in Paris.

**2.1.25 LFEED - LINE FEED**

Primary Command

FORMAT: **lfeed** <1 or 0>

ABBREVIATION: **LF**

DEFAULT: 0 (off)

PURPOSE: To send a linefeed to the printer after each carriage return.

COMMENTS: 'LFEED' is either on or off (1 or 0). Printing all on one line means you need to switch linefeed ON. Unwanted double spaced printing means you should switch linefeed OFF.

EXAMPLES: **lfeed 1**

Switches the linefeed on.

**lfeed 0**

Switches the linefeed off again.

**2.1.26 LINK**

Primary Command

FORMAT:        **link** <[field]>  
                 **link** "text"  
                 **link** x\$

ABBREVIATION: **lin**

DEFAULT:       Key field of current file.

PURPOSE:       To look up a record in another file, using data from the current record in the current file as the index key for the other file.

COMMENTS:      Prior to using LINK, set which file you want to link to with the SETLINK command. If no parameter is specified, the key of the current record is used. The 'File Selected' indicator will show a '+' instead of its usual '=' when a link is active.

EXAMPLES:       See the 'SETLINK' example.

**2.1.27 LMARG - LEFT MARGIN**

Primary Command

FORMAT:        **lmarg** <number>

ABBREVIATION: **lm**

DEFAULT:       1

PURPOSE:       Sets the left margin for both screen display and printed output.

COMMENTS:      'LMARG' can be used before an OUTPUT command to set the margin of the output. Maximum value is 250. Use with no value to reset the default value.

EXAMPLES:       **lmarg 20**

Sets the left hand margin to column twenty.

**2.1.28 LOAD**

Primary Command

FORMAT:       **load** <"<n:>filename">                   (C64/+4)  
              **load** <"<n:>filename"> <<at><n>>       (C128)

ABBREVIATION: **10**

PURPOSE:       To load a Superbase program from disk. Superbase 128 also allows optionally specifying a change in the memory space allocated to program code, or specifying that the program should be appended to the code in memory.

COMMENTS:      The ".p" suffix may be omitted. A drive specifier may be used. If a program is loaded by another program (i.e. chained), it will run automatically. Any existing program will be cleared.

C128 only:

Use 'load "filename" at n' without a number or variable following to cause the loaded file to be appended to the existing program. The whole program will start executing again from line 1 following a variable 'clr', so use 'dump' and 'set' to preserve variables if you need to. Modules loaded with this syntax must be carefully numbered so they do not corrupt existing program lines. Appending a module more than once also causes program corruption.

However, the command 'chain' has exactly the same syntax as 'load', except that variables are not cleared when the new module is loaded.

Use 'load "filename" at n' to increase the space available for program code from the default of about 8K. 'load "programname" at 0' give the maximum code space, 2K more than the minimum. Each increment from 0 to 9 decreases the space available by 256 bytes. This feature uses memory space reserved by Commodore for cartridge and some interface devices.

**2.1.29 MENU**

FORMAT:       **menu**

ABBREVIATION: **meN**

PURPOSE:       To return to Menu 1. Often used with a conditional.

EXAMPLES:      eof **menu**

Returns to Menu 1 when the last record in the file has been processed.

**2.1.30 MODE** Primary Command (C128 only)

FORMAT: **mode**

PURPOSE: Switches screen configuration from 40 to 80 or from 80 to 40 columns.

COMMENTS: 'Mode' can only be used if no file is selected. It also causes any Superscript text to be cleared.

If you have selected a file format of the alternate mode to the one you're in, you may be looking at:

\* 'Screen deselected'

\* A blank screen.

See Appendix G.

**2.1.31 NMAT - NO MATCH CONDITIONAL** Primary Command

FORMAT: **nmat** <command(s)>

ABBREVIATION: **nM**

PURPOSE: 'NMAT' is a conditional used with SELECT KEY. This means that the execution of those commands which follow it in the same line is conditional on it being true. 'NMAT' is true whenever a select key MATCH operation is unsuccessful (key not found).

COMMENTS: See Reference Section 6.2.7 for details of the MATCH command.

Both NMAT and PMAT can be used after LINK.

EXAMPLES: **nmat** display "Sorry...I couldn't find that key field":wait

Displays the above message only if no record is found exactly matching the criteria you have specified.

**2.1.xx NOGRAPHIC** Primary Command (Plus/4 only)

FORMAT: **nographic**

ABBREVIATION: **nO**

PURPOSE: A Plus/4 command which sets off the graphics mode.

**2.1.32 OPEN**

Primary Command (C128 only)

FORMAT:       **open** "filename"

ABBREVIATION: **oP**

PURPOSE:       To create a disk file for output.

COMMENTS:     Any output command will use the disk file while an "open" file exists. OUTPUT, DETAIL, PRINT, and DISPLAY, together with the REPORT commands, are affected. To close the file and resume screen or printer output, use 'close'.

You cannot use 'find' while an "open" file is in use, but 'select match' can perform a similar function.

EXAMPLE:       10 file "invoices": **open** "datafile"  
                 20 select m where [balance] is ">100":eof 100  
                 30 detail [name] [telephone] [balance]  
                 40 detail " "  
                 50 goto 20  
                 100 close : end

**2.1.33 PDEF - PRINTER DEFINITION**

Primary Command

FORMAT:       **pdef** <number>

DEFAULT:       0

PURPOSE:       To set the printer type to be used.

COMMENTS:     The start-up program "start.p" may be edited to set the printer definition.

EXAMPLES:     **pdef 0**     CBM Dot Matrix: sends CBM ASCII code with a cursor down character at the start of each line to produce lower case letters. Graphics not available.  
                 **pdef 1**     Epson MX80 type: sends true ASCII.  
                 **pdef 2**     Daisywheel type: e.g. Diablo, Qume, etc., sends true ASCII.  
                 **pdef 5**     Epson MX80 type: sends CBM ASCII.  
                 **pdef 6**     Daisywheel type: e.g. CBM 6400, sends CBM ASCII.

Low cost daisywheel type printers should work with either 2 or 6. RS232 interface and/or printer buffer may be necessary. For non-CBM dot matrix printers, select 1 if you want Superbase to convert to true ASCII, or 5 if you are using an interface device to convert.

**2.1.34 PDEV - SET PRINTER DEVICE**      Primary Command

FORMAT:            **pdev** <parameter1><,parameter2><,parameter3>

ABBREVIATION: **pD**

DEFAULT:          4,7,0 (for 1515/1525/MPS 801/802/803 printers)

PURPOSE:          To set printer device number and parameters.

COMMENTS:        Some IEEE printers require secondary addressing to set print characteristics, such as condensed mode. See Appendix D for further details.

EXAMPLES:        Centronics      Connect directly to user port with no interface device. Enter 'pdev0'.

RS232            Connect to user port via an RS232 interface. See Appendix D to determine values for 'pdev'.

IEEE            Connect via an interface device. Enter 'pdev4,255,0' or '4,0,0'.

CBM Serial      Connect to disk drive and enter 'pdev4,7,0'. See Appendix D for 1526.

**2.1.35 PERFORM**                              Primary Command (C128/+4 only)

FORMAT:            **perform** <"string">  
                      **perform** x\$

PURPOSE:          To execute a Commodore 128 or Plus/4 Command Line stored in a string variable or enclosed in quotation marks.

COMMENTS:        A number of Command Lines can be assigned to string variables and a program can be designed to choose between them depending on predefined criteria. Multiple clauses may not be used. The maximum length of the command string is 80 characters. 'PERFORM' must be the last command if on a multi-clause line. The equivalent Commodore 64 command is 'DO'.

EXAMPLES:        a\$= "display [goods]":**perform** a\$

**2.1.36 PLEN - PAPER LENGTH**

Primary Command

FORMAT:           **plen** <number>

ABBREVIATION: **pL**

DEFAULT:        66

PURPOSE:        To set the length of paper for reports and other printed output. Maximum is 255 and minimum 3.

EXAMPLES:       **plen 78**

Sets the page length for a single sheet printer that does twelve (78 minus 66) linefeeds after each sheet to bring up the top of the next sheet.

**2.1.37 PLUS**

Secondary Command

FORMAT:           <command parameters> **plus** <parameters>

ABBREVIATION: **pIU**

PURPOSE:        For extending the parameter list of a command beyond 1 program line.

COMMENTS:       The REPORT GENERATOR (see Reference Section 13) uses 'PLUS' in the REPORT PROGRAM it creates to join the responses to its prompts after the user has answered 'Y' for 'Yes' when asked Any More?'.  
  
Note that PLUS can only be used in programs and not as a direct command. Parameters may include text within double quotation marks. 'PLUS' is valid in DISPLAY, PRINT, OUTPUT, TITLE, TOTAL, DETAIL, ENDREPORT, and SUBTOTAL.

EXAMPLES:        100 display "this is a string of text too long  
                    to fit in the seventy nine "**plus**  
                    200 "characters allowed for command lines"

Displays the two pieces of text as though they had been entered as one line.

**2.1.38 PMAT - PARTIAL MATCH**

Primary Command

FORMAT: **pmat** <command(s)>

ABBREVIATION: **pM**

PURPOSE: 'PMAT' is a conditional used with SELECT KEY. This means that the execution of those commands which follow it in the same line is conditional on it being true. 'PMAT' is true whenever a select key MATCH operation is PARTIALLY successful, and untrue if a full match is obtained.

COMMENTS: See Reference Section 6.2.7 for details of the MATCH command. You can use PMAT after LINK.

EXAMPLES: **pmat** display [Name]

Displays any name which partially matches the criteria specified.

**2.1.39 PRINT**

Primary or Secondary Command

FORMAT: **print** <parameters>

ABBREVIATION: **prI**

PURPOSE: To set the direction of output to the printer as opposed to the screen, and optionally print any parameters following it.

COMMENTS: The 'print' command is used primarily when switching output or printing variables, field names etc. It must be preceded by 'output' when using the 'output' option syntax. See Reference Section 8.

EXAMPLE: **print** @0,12

Causes a form feed if the code is 12. (C128 only)

**print** \$27,\$14,\$0

Sends the escape sequence 14,0 to the printer (this is how you send nulls). (C128 only)

**2.1.40 PROTECT**

Primary Command

FORMAT:       **protect** <"<n:>program name">

ABBREVIATION: **proT**

PURPOSE:       To save the current program in a form that prevents it from being listed or edited. Always take a reserve copy first.

COMMENTS:     Programs protected by this command can be LOADED, RUN, or DELETED, but not LISTED or edited. Protected programs are encrypted. If an attempt is made to LIST a protected program, the message 'End of Program' will be displayed unless the first statement in the program is a 'REM' statement, in which case text following the 'REM' on the same line will be displayed. This allows Copyright warnings, etc., to be included in users' applications, which may be marketed as Superbase applications modules. Contact Precision Software for further details of marketing agreements.

**2.1.41 QUIT**

Primary Command

FORMAT:       **quit**

ABBREVIATION: **qU**

PURPOSE:       Used to exit from Superbase and return to CBM BASIC, or to restore the full Superscript 128 text area if Superscript and Superbase have been loaded together.

COMMENTS:     'Quit' shuts down Superbase in an orderly manner, resetting the computer for other use, or returns to the Superscript Editing screen if Superscript 128 is in memory. Superbase becomes unavailable until reloaded.

**2.1.42 RESTART**

Primary Command

FORMAT:       **restart**

ABBREVIATION: **reS**

PURPOSE:       To re-initialize the current database and file after severe system errors such as disk errors, I/O error, 'file not open' error, etc.

COMMENTS:     Restart can be used to recover a deleted file if used immediately after a 'File Deleted' message caused by the SELECT DELETE command.

**2.1.43 RLINK - RETURN FROM LINK** Primary Command

FORMAT: **rlink**

ABBREVIATION: **rL**

PURPOSE: To return from the linked file to the original file.

COMMENTS: Used with the SETLINK, LINK and ELINK commands.

EXAMPLES: See the 'SETLINK' example.

**2.1.44 RMARG - RIGHT MARGIN** Primary Command

FORMAT: **rmarg** <number>

ABBREVIATION: **rM**

DEFAULT: 80 for the printer. 40 or 80 for the screen, as appropriate.

PURPOSE: To set the right margin for screen display and printed output.

COMMENTS: Can be used before the OUTPUT command to set the right margin of the output. Maximum value is 255 and minimum 20.

EXAMPLES: **rmarg 76**

Sets the right margin to column 76.

**2.1.45 SAVE** Primary Command

FORMAT: **save** <"<n:>filename">

ABBREVIATION: **sA**

PURPOSE: To save a Superbase program file on disk.

COMMENTS: A ".p" suffix is appended to the filename automatically. A drive specifier may be used. Any existing program of the same name will be overwritten.

**2.1.46 SCREEN**

Primary Command

FORMAT:       **screen** <number>

ABBREVIATION: **sC**

DEFAULT:       Screen 0

PURPOSE:       To set the default screen. This is the screen first displayed in the ENTER, SELECT and FIND options.

COMMENTS:      Any of the four screens in use can be set as the default screen. The range is 0 to 3.

EXAMPLES:       **screen 2 : select c**

Sets the screen to the third screen, and displays the current record.

**2.1.47 SET**

Primary Command

FORMAT:       **set** <"<n:>filename">

PURPOSE:       To execute commands or retrieve variable values stored in a sequential file on disk. Uses include the storage of control values such as incrementing transaction numbers, and the execution of subroutines or command sequences from within a program. The filename may be a BASIC string variable.

COMMENTS:      This allows you to set variables as they were when stored. The file may be stored with 'DUMP' or created with 'MEMO'. If the sequential file is created with 'MEMO', no line numbers are required. Any commands in the file will be executed in sequence without the file being loaded from disk. This means that certain programming operations may be executed with virtually no memory overhead for the program instructions. If this technique is used, do not attempt to produce a Key List, read another file or list, load or save a program, or execute a goto, gosub, or for/next loop. However, you can read the database. 'SET' must be the last command if on a multi-clause line.

**2.1.48 SETLINK**

Primary Command

FORMAT:        **setlink** "filename"

ABBREVIATION: **seT**

DEFAULT:       Current file

PURPOSE:       To establish which of the other files in the database is to be accessed by the LINK command.

COMMENTS:      The SETLINK, LINK, RLINK and ELINK commands enable information to be exchanged between files in the same database. They also allow you to access a record in another file by linking via its key.

They are particularly useful for updating one file with information from another, such as updating a BALANCE field in an accounts file with information from an invoices file.

You can only use SETLINK (and therefore LINK, RLINK, and ELINK) on one file at a time.

EXAMPLES:       100 file "Cust.Inv"  
                  200 **setlink** "Cust.Rec"  
                  300 find "update list" where [update] is "=yes"  
                  400 select from "update list"  
                  500 eol menu  
                  600 [update]="No"  
                  700 [Total-paid]=[Total-paid]+[Last-paid]  
                  800 x=[Last-paid]  
                  900 store the record  
                  1000 link [No.]  
                  1010 [Balance]=[Balance]+x  
                  1020 store the record  
                  1030 rlink  
                  1040 goto 400

For an explanation of this program, see Programming Section 1.

**2.1.49 SPACE**

Primary Command

FORMAT:       **space** <number>

ABBREVIATION: **sP**

DEFAULT:       0 (single spacing)

PURPOSE:       To set the number of spaces between lines in output produced by the REPORT or OUTPUT commands.

COMMENTS:      'Space 1' gives double spacing, 'Space 2' gives triple spacing etc. Range is 0 to 3.

EXAMPLES:       **space** 1

Double spaces your output or reports.

**2.1.50 STORE**

Primary Command

FORMAT:       **store**

ABBREVIATION: **sT**

PURPOSE:       To replace the current record in the database.

COMMENTS:      Unless STORE is used, changes to records made other than by SELECT ADD, SELECT REPLACE, or BATCH will only remain in effect until you select another record.

EXAMPLES:       100 find "h8list" where [Discount] is "=10%"  
                  200 select from "h8list"  
                  250 eof menu  
                  300 display @1,2 [Goods];[Discount]  
                  400 ask @5,5 "Change the discount?";a\$  
                  500 if a\$<>"y" then 200  
                  600 ask @5,7 "Enter new discount";[discount]  
                  700 **store** the record: goto 200

Will store any changes you make to the discount rate on any goods.

**2.1.51 SUBTOTAL**

Primary Command

FORMAT:       **subtotal** [field] "subtotal text"

ABBREVIATION: **suB**

PURPOSE:       To specify a field for which a change in the contents will produce a subtotal break in a report generated by a user-written REPORT PROGRAM.

COMMENTS:      The 'SUBTOTAL' command is only to specify the field for the subtotal break, and any accompanying text. Subtotal break fields are only relevant if the report is using a Key List previously sorted on this field. Subtotal variables are specified with the 'TOTAL' command. To ensure that a subtotal is cleared, place it in the first line of the 'SUBTOTAL' command (see Reference Section 13).

EXAMPLES:       400 **subtotal** [Department]"Subtotal for  
                  department:" plus s0  
                  500 @40 &6,2 s0

Prints the subtotal variable s0 and then clears it.

**2.1.52 SUPERSCRIPT**

Primary Command (C128 only)

FORMAT:       **superscript** <<all><from "<n:>listname">  
                  <,"command string">

ABBREVIATION: **suP**

PURPOSE:       To transfer control to the Superscript word processor when this is present in memory, allowing functions such as mail merge to have direct access to database files.

COMMENTS:      The maximum length of the command string is 127 characters.

See Programming Section 1.7 for full details of how to use this command.

EXAMPLES:       **superscript** from "h8list","/dlletterfile^m/pmp"

Loads the document "letterfile" and executes a mail merge using the records indexed in "h8list".

10 ask "name of list"; a\$  
20 ask "name of document"; b\$  
30 **superscript** from a\$,"/dl"+b\$+"^m/pmp"

As above, but in program mode, requesting user input of a list name and a document.

**superscript, "/pmv/qy"**

Displays the document in memory merged with data from the current record only, and then quits from the word processor, returning control to the database.

**2.1.53 TITLE** Primary Command

FORMAT: **title** <"report title">

ABBREVIATION: **tI**

PURPOSE: To determine the text to appear at the top of a REPORT page; may include column headings.

COMMENTS: Instead of writing your own report Programs, you could choose to have them written for you by the REPORT GENERATOR (see Reference Section 13). Positioning is done with the @x,y command. 'PLUS' is used to extend titles beyond a single program line.

EXAMPLES: **title** @30 "Sales Report 18/10/83"

**2.1.54 TLEN - TEXT LENGTH** Primary Command

FORMAT: **tlen** <number>

ABBREVIATION: **tL**

DEFAULT: 60

PURPOSE: To set the number of lines of text to be printed on the page. Maximum is 255 and minimum 3.

EXAMPLES: **tlen** 50

Sets the number of lines of text to be printed on the page to 50. Set this temporarily if you're printing 25 line screens with CONTROL-0.

**2.1.55 TOTAL**

Primary Command

FORMAT:       **total** <total parameters> <subtotal parameters>

ABBREVIATION: **t0**

PURPOSE:       To specify totals and subtotals to appear in a report generated by a User-written REPORT PROGRAM.

COMMENTS:      The parameters are calculations on Total Variables (t0, t1, ... t9), and Subtotal Variables (s0, s1, ... s9), and have the same format as calculations in the CALC Option.

Instead of writing your own report Programs, you could choose to have them written for you by the REPORT GENERATOR (see Section 13).

EXAMPLES:       400 **total** t0=t0+[total-paid] plus  
                  500 t1=t1+[outstanding] plus  
                  600 s0=s0+[total-paid]; s1=s1+[outstanding]

At the end of the report total paid and amount outstanding will be available for printing. At each change of the field or fields specified in 'SUBTOTAL' any Subtotal Variables specified are available for printing.

**2.1.56 VARSAPCE**

Primary Command (C128 only)

FORMAT:       **varspace** n (range of n: 1-12)

ABBREVIATION: **vA**

PURPOSE:       To vary the amount of memory space available for Superbase program variables.

COMMENTS:      Each change in n alters variable space by 1K. The command is only needed when Superscript is also present in memory. The default setting is varspace 8, which allows approximately three pages of word processor text. 'varspace 1' gives minimum variable space and maximum word processor text of about four pages. 'varspace 12' gives maximum variable space and a text area of about one and a half pages.

If using varspace in a program, make it your first instruction, as it clears out all variables. See Programming Section 1.7 for further details.

**2.1.57 WAIT**

Primary Command

FORMAT:        **wait**  
                 **wait a**  
                 **wait a\$**

ABBREVIATION: **wA**

PURPOSE:        Used within a user-written program to force the program to pause until a key is pressed.

COMMENTS:      'Wait' is useful when a program is displaying information on the screen. It returns a single keystroke from the user - no additional RETURN is needed. 'Wait a' will accept only numeric input.

EXAMPLES:       100 find "hlist" where [town] is "=London"  
                 200 select from "hlist":eol menu  
                 300 select c:**wait**:goto200

The program will pause between each record it retrieves until a key is pressed.

**2.1.58 WHERE**

Secondary Command

FORMAT:        <Command "<n:>list"> **where** [field] is  
                 "condition"; <[field] is "condition";...>

ABBREVIATION: **wH**

PURPOSE:        Used with FIND and SELECT MATCH to specify a condition to be met by those records to be selected.

COMMENTS:      See Reference Sections 7 and 6.2.7 for details of the FIND and SELECT MATCH commands. If multiple items are specified semicolons must be used to separate them.

EXAMPLES:       find "update list" **where** [Pending-Update] is  
                 "=yes"

Creates a Key List of those records which need updating.

select match **where** [Name] is "=Jones"; [City]  
is "=London"

Displays all records belonging to people named Jones who live in London.

### 3 MENU COMMANDS

Superbase contains 34 commands that are available to you from the Menus. Two of these commands ('select' and 'maintain') display further menus. The 'other' command gives access to a series of disk commands. All of these commands are typically those in regular use, and so an understanding of how they operate is needed when you are using Superbase.

The following section is a reference guide to the Superbase menu commands. Many of these commands may be abbreviated to the shortest possible unique form of the command, with a minimum of two characters. The abbreviated form of the command is given where appropriate.

### 3.1 THE COMMANDS

### 3.1.1 BACKUP

### Secondary Command

FORMAT: 'f6' or 'b' from the 'Maintain' Menu or  
maintain b

**PURPOSE:** To create a security or work copy of a disk.

COMMENTS: Single or double disk drives may be used. All files will be copied.

### 3.1.2 BATCH

### Primary Command

FORMAT: 'f2' from Menu 2 or **batch** all <parameters> or **batch** from "<n:>keylist" <parameters>

ABBREVIATION: **ba**

PURPOSE: To carry out operations on all of the records in the current file.

COMMENTS: Updating is automatic once the batch operation is in action. Fieldnames and/or variables are permitted as are all standard arithmetical operators and trigonometric functions.

**EXAMPLES:**

```
batch from "pricelist" [unit-price] =  
      [unit-price]*1.1  
batch all [unit-price]=[unit-price]*1.1
```

**3.1.3 CALC**

Primary Command

FORMAT: 'f5' from Menu 1 or **calc** <expr>; <expr>...

ABBREVIATION: **ca**

PURPOSE: 'Calc evaluates expressions typed onto the command line or supplied in program lines. The expressions can be arithmetical or can consist of an assignment instruction equivalent to the BASIC 'let' command. Use the 'clear' command to obtain a 'blank record' and then 'calc' to assign values to the fields, and 'store' to write the record into the database.

COMMENTS: Numeric or string values may be used. See Reference Section 11.

EXAMPLES: **calc** 23\*57.91  
**calc** b=cos(sin(9)/log(10))t2;b (b will be displayed)  
clear:**calc** [number]="601";[goods]="1 SLR camera"; [price]=79.99 :store

**3.1.4 CATALOG**

Primary Command

FORMAT: 'f2' or 'c' from the 'Maintain' Menu or **catalog**

ABBREVIATION: **caT**

PURPOSE: To display a list of all the files in the current database.

COMMENTS: The database catalog is the same as that shown by the FILE option.

EXAMPLES: **caT**  
**catalog**



**3.1.8 EXECUTE**

Primary Command

FORMAT: 'f7' from Menu 1 or **execute** <"<n:>programname">

ABBREVIATION: **eX**

DEFAULT: Current program in memory.

PURPOSE: Executes a program.

COMMENTS: The current program is the program resident in the computer's memory. To execute a program from the disk, type 'execute "programname"'. The required program is now in memory, and is executed as normal.

EXAMPLE: **execute** "1:start"

**3.1.9 EXPORT**

Primary Command

FORMAT: 'f4' or 'e' from the 'Maintain' Menu or **export** <"<n:>filename"<,"expr">>

ABBREVIATION: **exP**

PURPOSE: Takes a file and converts it to a sequential form for use by an external program or for re-importing by Superbase.

COMMENTS: This option will export all of the data in the current file. Records and fields are separated by RETURN unless alternative characters are specified in 'expr'. Files created with 'Export' are sequential and in CBM ASCII format.

EXAMPLES: **export** "1:newfile","/."

**3.1.10 FILE**

Primary Command

FORMAT: 'f1' from Menu 2 or **file** <"filename">

ABBREVIATION: **fil**

PURPOSE: File selection and creation of new file format.

COMMENTS: This command used without a filename produces a database Catalog and an "Enter File Name" prompt unless the file name is given as a parameter of the command. The file selected must be contained within the current database. If you select a 40 column file on an 80 column monitor, you'll see the message 'Screen deselected'. See Appendix G.

**3.1.11 FIND**

Primary Command

FORMAT: 'f3' from Menu 1 or **find** "<n:>keylist" where <condition> is <"condition">;<"condition"> is <"condition">...

ABBREVIATION: **fi**

PURPOSE: Sets up a list of keys in a separate disk file that may be used with the sort, output, batch, select, export and report commands.

COMMENTS: Produces a file with the default name of "h8list" or "hlist" depending on whether 80 or 40 column screen mode is in use. Names for the key list may be specified by the user under program or command line control. Lists are created from all those records satisfying specified criteria. Criteria may be specific field values, value ranges, or records not containing specific values. Two empty double quotes may be used to abbreviate "h8list" or "hlist".

EXAMPLES: **find** "" where [date] is ">jan184"

```
10 ask "Date";d$: date d$,n
20 if n=0 then 10
30 find "" where [date] is ">" +d$
```

Or:

```
100 find "newlist" where [goods] is "=camera";
    [price] is ">90.00"
200 output from "newlist"
```

**3.1.12 FORMAT**

Primary Command

FORMAT: 'f2' from Menu 2 or **format**.

ABBREVIATION: **f0**

PURPOSE: To amend an existing file definition.

COMMENTS: Fields are set by name, type and length. Each file must have a key field. Nine field types are available (see Reference Section 4). The field name is taken to be the last word of text occurring before the field. Explanatory text may be included on any file format screen. Up to four screens are available for each record (up to a total of 127 fields) and 1,107 characters of information, 23 lines per screen are allowed. See Appendix A. File formats are saved with 'f1' and 'Run/Stop' and aborted with 'f1' and 'Q', or CONTROL-Q.

**3.1.13 HELP**

Primary Command

FORMAT: 'f8' or the 'HELP' key from Menus 1 and 2 or **help** <"<n:>helpname">

ABBREVIATION: **hE**

PURPOSE: To display a help screen.

COMMENTS: Help screen names are prefixed by the characters 'h8' or 'h' depending on whether 80 or 40 column screen mode is in use. This prefix is not typed in when specifying a help name. Help screens provide information on the major Superbase topics. Press RETURN to read the next screen of a multi-screen help file.

**3.1.14 IMPORT**

Primary Command

FORMAT: 'f3' or 'i' from the 'Maintain' Menu or **import** <"<n:> filename">

ABBREVIATION: **iM**

PURPOSE: Allows Superbase access to external files.

COMMENTS: Files must be sequential and in CBM ASCII format. The receiving Superbase file must have at least as many fields as the external file, and the field types must be in the same order. The filename used in the command is that of the file to be imported. The destination file is that currently selected in Superbase.

**3.1.15 MAINTAIN**

Primary Command

FORMAT: 'f6' from Menu 2 or **maintain**

ABBREVIATION: **mA**

PURPOSE: Displays a menu of eight other Superbase options

COMMENTS: The options are: Status; Catalog; Import; Export; Directory; Backup; New Disk; and Other.

**3.1.16 MEMO**

Primary Command

FORMAT: 'f7' from Menu 2 or **memo** "<n:>filename">.

ABBREVIATION: **mE**

PURPOSE: To write text files to be used as memos or help screens.

COMMENTS: If the filename specified does not yet exist, control will be passed to the Superbase Memo Writer. Any text entered will be saved with 'f1 Run/Stop'. 'f1 Q' or CONTROL-Q quits from the Memo Writer without saving your text on the original. File names prefixed by 'h8' or 'h' can be called up as Help Screens.

EXAMPLES: **memo** "h8diary"

**3.1.17 NEW DISK**

Secondary Command

FORMAT: 'f7' or 'n' from the 'Maintain' Menu

PURPOSE: To initialize a new disk.

COMMENTS: Initializing a disk organizes it so that it can be used for data storage. The "start" program and Help Screens are not copied, but "start" may be saved onto the new disk.

**3.1.18 OTHER**

Secondary Command

**FORMAT:** 'f8' or the 'HELP' key or **maintain o** <"command">

**PURPOSE:** Allows access to certain Commodore disk commands without exiting from Superbase.

**COMMENTS:** Permissible commands are Rename, Copy, Scratch, Validate and View Disk Directory. These are entered without double quotes in response to the prompt, and with quotes if following 'maintain o' on the Command Line or in a program.

**3.1.19 OUTPUT**

Primary Command

**FORMAT:** 'f4' from Menu 1 or  
**output** all <"text";[field]...>  
**output** from "<n:>keylist" <"text";[field]...>

**ABBREVIATION:** **oU**

**PURPOSE:** Displays or prints information from the current file or from a named key list according to formatting commands.

**COMMENTS:** "Display" or "print" and "down" or "across" typed on the command line before the output command is selected determines the output device and the layout of the information. Typing "all" produces output from all the records in the current file: "from..." uses a key list. Any field, variable or text string may be used, along with the full range of Superbase formatting and truncating commands.

See Reference Section 8.

**EXAMPLES:** **output** all [name][date][total]  
**output** from "h8test" across [name] "owes"  
[total]  
**output** all @10 [name]

**3.1.20 PROG**

Primary Command

FORMAT: 'f5' from Menu 2 or **prog** <line number>

ABBREVIATION: **pR**

DEFAULT: Calls up current program.

PURPOSE: Enters the Superbase program writer. Specify line number to start display at that line.

COMMENTS: This command will display the current program in memory. To develop a new program, clear the memory by typing 'new'. The program writer gives a screen of ten double lines, with full up and down scrolling. Programs consist of Superbase and BASIC commands sequenced by line numbers. Save a program with 'save "<n:>programname"'.  
  
See Programming Section 1.

**3.1.21 RENAME**

Secondary Command

FORMAT: **maintain o** <"rn:newname=n:filename">

PURPOSE: To give a file a new name.

COMMENTS: If you rename a database, the new name must be in upper case characters. The new names of programs and Help files must be given the '.p' suffix and 'h8' or 'h' prefix respectively.

EXAMPLES: **maintain o** "r0:newinvoices=0:invoices"

**3.1.22 REPORT**

Primary Command

FORMAT: 'f6' from Menu 1 or **report**

ABBREVIATION: **rE**

PURPOSE: To create reports on paper or displays on the screen.

COMMENTS: A Superbase report is a short program generated automatically by your responses to nine questions. The program may be modified using the program editor ('f5' on Menu 2). In addition to titles, column headings, text inside quotes, and formatted and/or positioned fields, you can specify arithmetically derived output, such as multiplied fields, counts, and averages, as well as totals, and subtotals.

**3.1.23 SCRATCH**

Secondary Command

FORMAT:       **maintain o** <"sn:filename">

PURPOSE:       To remove unwanted files from the disk.

COMMENTS:      To scratch a database file, remove all the records first. For further details, see Reference Section 15.

EXAMPLES:      **maintain o "s0:invoices"**

**3.1.24 SELECT**

Primary Command

FORMAT:       'f2' from Menu 1 or **select** <option>.

ABBREVIATION: **sE**

PURPOSE:       Displays a further menu of 11 Superbase options.

COMMENTS:      The options are: Key; Current; Next; Last; Previous; First; Match; Output; Add; Replace; and Delete. To return to Menu 1, press RETURN. The Select options are used to examine and modify specific records within the current file.

**SELECT KEY**

FORMAT:       **select** "Jones"  
              **select** a\$  
              **select** [field]

PURPOSE:       Searches for a record in the file by index key.

COMMENTS:      A record selected by key is displayed if found. If no exactly matching key is found, a 'no match' condition exists (see NMAT). If a key is found whose leading characters match the specified key but which is not an exact match, a 'partial match' condition exists (see PMAT).

**SELECT FROM**

FORMAT:           **select from "<n:>key list"**

PURPOSE:          Obtains a record from the file using the next key from a key list obtained with the FIND Option.

COMMENTS:        The command always starts with the first key in the key list, and then works sequentially through the list. For this reason it is usually used in a program loop. Use the 'eol' test to determine when the end of the list is reached.

**SELECT CURRENT**

FORMAT:           **select c**

PURPOSE:          Displays the current record.

COMMENTS:        Used in programs to keep an image of the current record on the screen, when obtaining input with 'ask' or just to view the record.

EXAMPLE:          ask [name]:**select c**:ask [street]:**select c**:wait

**SELECT FIRST, LAST, NEXT, PREVIOUS**

FORMAT:           **select f**  
                     **select l**  
                     **select n**  
                     **select p**

PURPOSE:          Displays the first, last, next, or previous record as indicated.

COMMENTS:        Used for browsing through the file. Sequential reading through a file is achieved with 'select n' in a program loop. Use 'select l' to terminate a match operation and allow the input of new search criteria. An 'End of File' message is displayed when you attempt to browse beyond either end of the file.

**SELECT MATCH**

FORMAT:       **select m** where <"condition"> is  
                  <"condition">;...

PURPOSE:       Displays the records in the file that match the  
                  criteria indicated. See Reference Section 6 for  
                  detailed explanation of match criteria.

COMMENTS:      When 'select m' is used at the menu, the first  
                  matching record found is displayed. Following  
                  matching records are obtained by typing 'M'. In  
                  a program, you must use a loop to 'select m',  
                  with the 'select c: wait' commands to hold the  
                  image on the screen. To terminate the operation,  
                  type 'L' to reach the last record in the file.

**SELECT OUTPUT**

FORMAT:       **select o**

PURPOSE:       Produces a screen or printer output of the  
                  current record.

COMMENTS:      Useful for quick view or dump of current record  
                  details, where the record extends over more than  
                  one screen. Can be combined with 'select k' for  
                  controlled inquiry.

**SELECT ADD**

FORMAT:       **select a**

PURPOSE:       Adds a new record to the file.

COMMENTS:      First select a current record. Then 'select a',  
                  and you will be able to replicate the current  
                  record, changing its key and any other details.  
                  This can save much data entry time. The message  
                  'Key Already Exists' signifies that you have not  
                  created a new key.

**SELECT REPLACE**

FORMAT:           **select r**

PURPOSE:          Allows amendment of the details of the current record.

COMMENTS:        First select a current record. Then 'select r', and you will be able to change any details except the record key. Use 'Shift' and RETURN to end the editing process. When done under program control, the operator is responsible for ending the process correctly. You may prefer to accept data with 'ask', showing the record image with 'select c'.

**SELECT DELETE**

FORMAT:           **select d**

PURPOSE:          Deletes a record from the file.

COMMENTS:        First select a current record. Then 'select d', and you will be prompted to confirm the deletion of the record. If 'select d' is used in a program, no confirmation is requested. If you 'select d' in a file that has no records, you will delete the file entry itself (this is how you remove a file from the database directory). This means that in a program you should always test for end of file, or you risk inadvertently deleting the file entry. If you want to recover a file definition, use the 'restart' command. See Programming Section 1 for examples of how to delete groups of records from a file.

### 3.1.25 **Sort**

### Primary Command

FORMAT: **sort** <all> <from "<n:>listname" <D->> on <&n>  
[field] <[field]...> <to "<n:>sorted list">

ABBREVIATION: **s0**

PURPOSE: Sorts the keys of the records in the current file or a named key list into an order other than that of the key field.

COMMENTS: The sort command creates a key list ordered by the values held in fields other than the key field. Sorting may be done in ascending or descending order. Unless another name is given, the sorted list will be called "h8list" or "hlist".

EXAMPLES:    **sort** all on [customer]  
               **sort** from "testlist" on [customer] to "newlist"  
               **sort** all D- on [due]

### 3.1.26 STATUS

### Primary Command

```

FORMAT:      'f1' or 's' from the 'Maintain' Menu or
              status                      (C128)
              maintain status             (C64/+4)
              maintain s                   (all)

```

PURPOSE: Displays the field numbers, names, types, lengths and calculation formulas for the current file.

COMMENTS: Output may be directed to the screen or printer.

### 3.1.27 VALIDATE

### Secondary Command

FORMAT: maintain o "vn"

**PURPOSE:** To check for disk errors on the specified drive.

EXAMPLES:     maintain o "v0"

### 3.1.28 VIEW DIRECTORY

### Primary Command

FORMAT: maintain o "\$n"

PURPOSE: To display the files on the current disk.

**TECHNICAL APPENDIX**

The Technical Appendix provides information for both the general user and the applications developer. The four sections are (1) Maximum System Values; (2) Data Storage; (3) File Manager; (4) Start-up Program.

**Note: "Unlimited" in the descriptions below implies limited by disk storage constraints only.**

**MAXIMUM SYSTEM VALUES**

<b><u>Database Level</u></b>	Database Name:	16 characters
	Databases:	unlimited number
	Files in a database:	15
	Programs in a database:	unlimited
	Key Lists in a database:	unlimited
	Memo files in a database:	unlimited
<b><u>File Level</u></b>	File names:	16 characters (key list, memo) 14 characters (program) 10 characters (data)
	Records in a file:	unlimited
	Linked files at one time:	1
	Memo file length:	23 screen lines
	Program size:	C128: 62K including variables C64: 4K including variables Plus/4: 8K including variables
	Key List length:	unlimited (may be appended)
<b><u>Record Level</u></b>	Fields:	127, including key
	Descriptive text in file definition:	1000 characters
	Key - anywhere in record:	1 field
	Combined result, calendar and constant fields:	32 fields
	Record length:	1107 characters
	Screens:	4
	Calculations:	79/159 characters expanded, 30/60 characters compressed (Field names take 2 characters. Functions including '(' and ')' take 1. Spaces do not count).
<b><u>Field Level</u></b>	Field name length:	12 characters
	Key:	30 characters
	Text:	255 characters
	Numeric:	9 digits, up to 4 after decimal pt, plus 1 sign character position
	Date:	7 or 11 chars. - 1Jan1900 - 31Dec1999
	Calendar:	7 or 11 characters
	Constant:	30 characters
	Result:	as numeric
<b><u>Miscellaneous</u></b>	Command line:	79/159 characters
	Program line:	79/159 characters
	Screen width:	40/80 characters
	Screen length:	23 lines
	Printer columns:	255

**DATA STORAGE**

When calculating **maximum** stored record length, follow these rules:

1. Count every character position visible between field angle brackets for text, key, numeric, result, and constant fields. Allow five characters for each date or calendar field. Replica fields do not count.
2. Add one byte for field separators (i.e. number of fields minus one). Replica fields do count. Separator is chr\$(96).
3. Field names and descriptive text do not count.
4. Trailing spaces and numeric zeros are not stored. Nor are leading spaces.
5. Add one record separator chr\$(0) per record.
6. Floating point storage, which requires minimum five bytes per number, is not used, allowing more economical storage of integers < 10000.
7. Database files are linked into the main disk directory allowing the use of the disk VALIDATE command from outside Superbase.

When calculating file length, follow these rules:

1. Minimum data area is 128 bytes per record, five bytes of which is required by the system, leaving 123 bytes for data. If more than 123 bytes are used in a record then a further 128 bytes will be assigned. This is a design feature that optimizes record access times and efficient dynamic disk space allocation.
2. File indexes are stored as one key per record (trailing spaces removed) plus three bytes. Index size varies according to the length of the key (the shorter the better) and the number of blocks in use. Blocking factor (average percentage of index block in use) is approximately 65%. As the index grows it creates a multi-level tree structure with higher levels pointing to index blocks instead of data areas as in the bottom level (B+ tree type).
3. Database files and data will be stored on the same drive. File definitions should be stored on the same drive as the database but could be accessed from another drive. Key lists, help screens, programs, import/export files, and 'output to files may be stored on either drive (use the 'l:' prefix if necessary). Output to a named list, program, memo or export file will overwrite any existing file of the same name.
4. Data is stored from and including track 2 of the disk. 5k is therefore reserved for non-database storage. Directory areas of the disk will be avoided.

**FILE MANAGER**

More than one file definition may reside in memory at any one time, provided that the total number of fields does not exceed 127, the total number of calculations does not exceed 32, and the total amount of descriptive text does not exceed 1k. A maximum of three simultaneously resident definitions is possible if these totals are not exceeded. Multi-file applications will run faster if file definitions do not have to be loaded from disk each time the current file changes with the FILE command.

The process of creating a file clears all definitions from memory.

**START- UP**

A start-up program, "start.p," is included on the Superbase disk. It is optional, and if not present at start-up may be ignored by pressing RETURN when the "File Not Found" message appears. The original start-up program resides on the Superbase disk. It may be edited using the PROG Option to suit each user's requirements. Applications developers may modify the program to present applications of their own design.

"Start" may for example display HELP screens containing program option menus, request and validate input (including passwords), and load programs for execution.

Systems do not have to use the standard Superbase Option menus at all.

Also if "start" and other programs are encrypted with the PROTECT command and full error trapping is implemented, a completely secure application can be created.

An annotated listing of "Start" is included here. Study it to understand how Superbase's system parameters may be set. These parameters are set automatically to the values below at load time but have been included in the program as an explanatory aid:

brkon	STOP key enabled
across	OUTPUT command direction mode
display	OUTPUT command device mode
lmarg 1	Left printer margin 1
rmarg 80	Right printer margin 80
plen 66	Paper length 66 lines
tlen 60	Text length 60 lines
lfeed 0	Linefeed off
cont 1	Continuous print on
space 0	Single line spacing
pdev 4	Output device no. 4
pdev 0	CBM printer type

**THE START PROGRAM (SUPERBASE 128)**

```

10   rem superbase start program
20   cols a:c=0:if a=80 then c=20:rem 40 or 80 column display
100  brkon:gobsub 400:rem allow break key and set system parameters
105  l$="-----":l$=l$+l$:l$=left$(l$,17)
110  display @10+c,9" "+l$+" "
120  display @10+c,10"|"@" Superbase 128 ""|"
130  display @10+c,11" "+l$+" "
140  display @1+c,17"-----"
145  rem
150  display @1+c,19"-----"
160  display @0:rem reset display count
170  ask &16@1+c,18" Enter Database Name :";a$
180  database a$,8,0:rem disk 8,drive 0
190  file:rem superbase asks for file
200  new:rem clear program goto menu
400  rem *** set system parameters ***
410  lmarg 1:rmarg 80:rem margins
420  plen 66:tlen 60:rem page & text length
430  pdev 4:pdef 0:rem printer device 4 cbm code
440  lfeed 0:cont 1:rem no line feeds, continuous print
450  space 0:across
460  screen 0
470  return

```

**THE START PROGRAM (SUPERBASE 64 AND PLUS/4)**

```

100  brkon:gobsub 400:rem allow break key and set system parameters
110  display @12,3" ----- "
120  display @12,4"|"@" Superbase ""|"
130  display @12,5" ----- "
140  display @1,17"-----"
150  display @1,19"-----"
160  display @0:rem reset display count
170  ask &16 @1,18" Enter Database Name :";a$
180  database a$,8,0:rem disk 8,drive 0
190  file:rem superbase asks for file
200  new:rem clear program goto menu
400  rem *** set system parameters ***
410  lmarg 1:rmarg 80:rem margins
420  plen 66:tlen 60:rem page & text length
430  pdev 4:pdef 0:rem printer device 4 cbm code
440  lfeed 0:cont 1:rem no line feeds, continuous print
450  space 0:across
460  screen 0
470  return

```

**ERROR MESSAGES**

**Note: ERROR LIGHT** Superbase uses the error channel extensively when allocating disk space. Error light flashing during disk write operations does not normally indicate disk errors.

**ALREADY LINKED**

A LINK command has been issued while a link is already in progress.

**COMMAND SEQUENCE ERROR**

You have entered a command line having a command in the wrong position.

**DATA MISMATCH**

System error or hardware failure. Can be caused by failure to use 'database' command after changing disks. Enter 'database' RETURN and carry on. If this fails, export all files, create a new database using same file definitions and import the data.

**DATABASE NOT FOUND**

The DATABASE you have selected is not on the current disk.

**DATABASE NOT SELECTED**

Commands have been issued that require access to a DATABASE when none is selected.

**DISK ERROR MESSAGES**

Standard COMMODORE disk error messages are displayed on the STATUS LINE, or the top line of the screen. See your disk drive manual for further details.

Certain errors are also detected by Superbase because of device malfunction. These are input/output errors and have error numbers from 1 to 9. A full list of BASIC and Superbase errors, together with their numbers, appears at the end of this section.

**EQUATION ERROR**

You are either attempting to assign a value to a field of the wrong type to accept that value (e.g. attempting to assign a string of TEXT to a NUMERIC field), or else you have missed out an 'IS' or '=' where there should be one.

**FIELD NAME OR BRACKETS ERROR**

Your command line either refers to a non-existent field or you have typed an uneven number of brackets in the command line.

**FIELD TOO LONG**

You have entered a calculation which assigns a string too large for the field you have assigned it to.

**FILE DEFINITION INVALID**

The FILE command has attempted to load a file definition and accessed an invalid file. This error can occur if the file definition is overwritten by a key list of the same name. To recover, reformat the file using 'FORMAT'.

**FILE DELETED**

You have issued a SELECT DELETE command to a file with no records, which has caused the database to delete the file. See the RESTART command.

**FILE NOT FOUND**

The FILE you have selected is not on the current disk.

**FILE NOT SELECTED**

You have attempted to issue a command requiring use of a file before selecting a file.

**FMS COMMAND ERROR**

The file manager has detected an invalid command due to system or hardware error. Reload Superbase if you cannot continue work.

**FORCED FIELD: PLEASE ENTER DATA**

You are attempting to store a record while leaving a KEY FIELD or another field that has been set as a FORCED FIELD empty of information.

**FULL DISK**

There is no more space available on the disk you are using.

**INDEX MISMATCH**

System error or hardware failure. Can be caused by failure to use 'database' command after changing disks. Enter 'database' RETURN and carry on. If this fails, export all files, create a new database using same file definitions and import the data.

**INSIDE FIELD: CAN'T SET**

You are attempting to set a field in a position already occupied by another field.

**INVALID COMMAND PARAMETER**

You have issued a command with a parameter too large, too small, or of the wrong type.

**INVALID DATE**

You have attempted to enter into a DATE FIELD information not of the form '05may83' or 'may0583'.

**INVALID DIRECT COMMAND**

You have issued a command which is either a command only for use within a PROGRAM or COMMAND LINE or is invalid in Superbase.

**INVALID FMS PARAMETER**

You have tried to access a record by using an INVALID KEY. Can be caused by editing a KEY LIST or using an invalid or corrupt KEY LIST.

**INVALID LINE: RE-ENTER**

You have tried to enter a PROGRAM LINE with no LINE NUMBER while in the program writer provided by the PROG Option.

**INVALID LINK FILE**

You have attempted to LINK to a non-existent file or to a file in a different DATABASE.

**INVALID NUMERIC RESULT**

You have either assigned a value too large or of the wrong type to a numeric field by means of an ASK or CALC command, or the result of your using one of these commands has caused a RESULT field to assume a value of the wrong type or too large for its format.

**INVALID SCREEN NUMBER**

You are trying to go to a screen which has not been formatted for the current file, or to format a screen when the current file has already a full complement of four screens.

**KEY ALREADY EXISTS**

You are trying to ENTER a record with a KEY which already exists on another record.

**KEY FIELD NOT DEFINED**

You have tried to format a RECORD LAYOUT without including a KEY FIELD.

**LINE NOT FOUND**

You have issued a GOTO command to a program line that does not exist.

**NO FIELD DEFINED**

You have attempted to format a RECORD LAYOUT without any fields at all.

**NO HELP AVAILABLE**

You have requested a HELP SCREEN that does not exist. Remember, you should not type the 'h8' or 'h' that prefixes the Help text file. Also, omit these prefixes when using 'help' as a program line.

**NO LINK SET**

You or a program have issued a LINK command without having first specified which file is to be linked to with the SETLINK command.

**NO PROGRAM PRESENT**

You have issued an EXECUTE command when there is no program resident in memory.

**NOT A DATABASE FILE**

You have attempted to select a database with a filename which is that of a file other than a database file.

**NOT A PROGRAM FILE**

You have specified a filename in a LOAD or EXECUTE command which is not the name of a program but of some other file.

**OUT OF MEMORY**

Your computer has run out of free memory space.

**RECORD TOO LONG**

You have tried to format a record larger than the permitted size (for limitations on record sizes see APPENDIX A).

**SEMICOLON MISSING ERROR**

You have omitted a semicolon between calculations in a CALC command.

**SYNTAX ERROR**

Incorrect use of Superbase and/or BASIC commands or functions. It may help to look at Programming Section 1.

**TOO MANY COMMENTS**

You have attempted to FORMAT a RECORD LAYOUT containing too much DESCRIPTIVE TEXT (for limitations on amounts of descriptive text see APPENDIX A).

**TOO MANY FIELDS**

You have attempted to FORMAT a RECORD LAYOUT containing too many fields (for limitations on numbers of fields see APPENDIX A).

**TOO MANY FILES**

You have tried to create more than the maximum fifteen files for a given database.

**ERROR MESSAGES ON THE C128 AND PLUS/4**

Error messages up to 41 (C128) or 36 (Plus/4) can be used in programs with the BASIC function 'err\$(er)' where 'er' is the BASIC error number variable. Messages above 41 can be detected with 'er', but the message itself is not available.

A list of all error messages by number follows:

<b>Error Number</b>	<b>Error Message</b>
1	Too many files
2	File open
3	File not open
4	File not found
5	Device not present
6	Not input file
7	Not output file
8	Missing file name
9	Illegal device number
10	Next without for
11	Syntax
12	Return without gosub
13	Out of data
14	Illegal quantity
15	Overflow
16	Out of memory
17	Undefined statement
18	Bad subscript
19	Redimensioned array
20	Division by zero
21	Illegal direct
22	Type mismatch
23	String too long
24	File data
25	Formula too complex
26	Can't continue
27	Undefined function
28	Verify
29	Load
30	Break
31	Can't resume
32	Loop not found
33	Loop without do
34	Direct mode only
35	No graphics area
36	Bad disk
37	Bend not found
38	Line number too large
39	Unresolved reference
40	Unimplemented command
41	File read

The following list of Superbase errors includes all those whose error number will be returned in 'er' when the error occurs.

<b>Error Number</b>	<b>Error Message</b>
65	Invalid date
68	No help available
70	System error
72	No fields defined
73	Command sequence error
74	Field too long
75	Invalid numeric result
76	Semicolon missing
77	Equation error
78	Invalid command parameter
79	Record too long
80	Invalid screen number
81	No program present
82	Out of memory
83	End of program
84	Line not found
85	Invalid direct command
86	Invalid link file
87	No link set
88	Already linked
89	Too many subtotals
90	Field name or brackets error
95	Not a program file
99	Superscript not available
100	Superscript error
101	Invalid FMS parameter
102	Not a database file
103	Full disk
104	DOS or printer error message
105	Create it error
106	Database not found
107	File not found
108	File not selected
109	Index mismatch
110	Too many files
111	Key not found
112	Key already exists
113	Data mismatch
115	Command error
116	Delete error
117	File deleted
118	Database not selected
119	Partial match

**SELECTED GLOSSARY**

**BASIC VARIABLES**

Locations in the computer's memory which can be used to store information temporarily during processing.

**BORDER**

A string of characters, usually graphics characters, used to draw a border around the SCREEN LAYOUT.

**CHARACTER**

A single symbol that appears on the screen. Created by pressing a key or combination of keys.

**COMMAND AREA**

The top two lines of the screen where commands are entered. This is also the MESSAGE AREA where Superbase displays its messages to you.

**COMMAND LINE**

A sequence of commands separated by colons and executed one after the other.

**CURSOR**

The flashing block on the screen that indicates where the next character which is typed will appear.

**CURSOR CONTROL KEYS**

Marked with arrows. Used to move the cursor.

**DATA**

Information stored in FILES.

**DATA DISK**

The disk used for storing your files.

**DATABASE**

A collection of up to 15 files held together on disk.

**DEFAULT LIST**

The KEY LIST created during a FIND or SORT operation if no listname is specified. The default names "h8list" or "hlist" are assigned, depending on whether 40 or 80 column screen mode is operative.

**DEFAULT VALUES**

Those values which Superbase assigns to parameters if no others are specified.

**DELETE**

Remove text without leaving a gap, i.e. the remaining characters move up to fill the space that had been occupied by the deleted text. (Contrast with "erase".)

**DESELECT**

The operation of leaving a selected option without entering any parameters or taking any other action.

**DESCRIPTIVE TEXT**

Text that appears on a SCREEN LAYOUT to make clear what the various items in the record are meant to be. This may include graphics characters to underline or box in sections of the screen or to provide a BORDER around the screen.

**DIRECTORY**

A list of the names of the files on a disk.

**DISK FORMATTING**

The process of setting up a blank or recycled disk for use as a Superbase DATA DISK.

**ERASING**

Blanking a line of DESCRIPTIVE TEXT with spaces without filling the gap left by the line erased (as opposed to DELETING).

**FIELD**

A 'blank slot' where information is stored in the record.

**FIELD END MARKER**

A striped rectangle which appears in the FORMAT Option to signify where a field ends.

**FILE**

A collection of Records stored together on disk with the same screen layout.

**FILE DEFINITION**

The file on disk storing the details of the RECORD LAYOUT.

**FILL FILE**

A file of information used by a word processing program to fill in blank slots (variable blocks) in a standard letter.

**FORCED FIELD**

A FIELD which must have data entered into it and may not be left blank.

**FORMATTING COMMANDS (1)**

Commands which enable you to adjust the way that output is displayed on screen or printed on paper.

**FORMATTING COMMANDS (2)**

Commands which are used to set up the structure of the screens in your RECORD LAYOUT.

**FUNCTION KEYS**

The large keys on the top right of the CBM 128 keyboard, the right of the C64 keyboard or the top of the Plus/4 keyboard, used to control many Superbase operations.

**HOME POSITION**

The top left corner of the screen or text. The cursor can be moved to the top left of the screen by pressing CLR/HOME.

**INSERT**

Add characters, words or lines in between other characters, words or lines in a descriptive or other text.

**INVERTING**

Changing from dark characters on a light background to light characters on a dark background. Single lines or whole screens can be inverted.

**ITEM LIST**

A list of parameters following an OUTPUT, SORT or BATCH command. It may include fieldnames and/or BASIC variables and expressions.

**KEY**

Every record has a key, which is whatever is stored in the key field of the record. Superbase uses the keys to order the records in the file.

**KEY LIST**

A list of KEYS of records which can be used to restrict various Superbase operations to just those records whose keys appear on the list.

**LINE NUMBERS**

Numbers placed at the beginning of COMMAND LINES to turn them into PROGRAM LINES.

**LOAD**

Take a file that is on a disk and copy it into the computer's memory.

**MATCH CRITERIA**

The information you enter into a RECORD TEMPLATE to determine which RECORDS are to be selected for viewing or for incorporating into a KEY LIST.

**MESSAGE AREA**

The top line of the screen where Superbase displays its messages to you. This is also the first of the two lines where you can enter COMMAND LINES.

**PARAMETER**

A named field, named BASIC variable, string or numeric expression that may include field names or BASIC variables, or number, entered as part of a command to tell the program what information to process, and sometimes how to process it.

**PROGRAM**

A sequence of numbered COMMAND LINES which are executed in numerical order.

**PROGRAM LINE**

A COMMAND LINE which has a number in front of it so that it can be grouped with other program lines to form a PROGRAM. These Command Lines are then executed in the order dictated by the LINE NUMBERS.

**RECORD**

A collection of FIELDS which can be spread over up to four screens and can be regarded as a unit. Records are held together in FILES and every record within a file is of the same size and layout and has the same fields as the others in that file.

**RECORD FORMAT**

The SCREEN LAYOUTS of the RECORDS in a FILE.

**RECORD LAYOUT**

See RECORD FORMAT.

**RECORD TEMPLATE**

The blank RECORD FORMAT as provided to enable you to enter MATCH CRITERIA or criteria to determine which RECORDS are to be included in a KEY LIST.

**REPLICA FIELD**

A record field copied from an already defined field in the same record.

**SAVE**

Take a file that is in the computer's memory and copy it onto a disk.

**SCRATCH**

Remove a file from a disk.

**SCREEN LAYOUT**

The skeleton of a record screen consisting of FIELDS and DESCRIPTIVE TEXT. This is a blank record form like an empty card in a card-index.

**SCREEN DUMP**

A printout of whatever is on the screen at any given time.

**SEQUENTIAL FILE**

A file which consists of a stream of data with each field separated by a RETURN. Sequential Files can be used to transfer data between Superbase and other programs or between separate Superbase DATABASES.

**SORT PARAMETERS**

The parameters entered in the SORT Option which determine the order in which the records are to be sorted.

**STRING**

A sequence of characters (letters and/or numbers).

**STORE**

Save a file from the computer's memory onto a disk.

**TRUNCATION**

Trimming or shortening a string of characters to a particular size.

## PRINTERS

This appendix brings together several related aspects of using Superbase with printers, including interfaces and format commands.

### 1. PRINT FORMAT COMMANDS

There are six commands that determine the basic layout of your printed page.

Margins are controlled by **lmarg** and **rmarg** followed by the required column number, e.g. **lmarg 10:rmarg 65**.

Line Spacing is controlled with **space** followed by 0, 1, or 2 for single, double, or triple spacing respectively, e.g. **space 1**.

Line feed can be set on or off with **lfeed 1** or **lfeed 0** respectively. If all printing happens on one line, type in: **lfeed 1**. If you get unwanted double spacing, type in **lfeed 0**.

Paper length is indicated by the command **plen** followed by the maximum number of lines printable on the paper in use, e.g. **plen66** is used for 11 inch paper with six lines per inch.

Text length means the number of lines that can be printed on a page before Superbase moves to a new page, e.g. **tlen 50** will allow 50 lines to be printed and then start a new page. (**tlen 50** allows two 25 line screen dumps per page.)

Use the output menu option to select records for printing. You can position items of data anywhere on the page, and vary the format with special commands. See Reference Section 8.

The print command can be used to send data directly to the printer. This can include record data as well as variables, and the positioning and formatting commands are available.

The form 'print @0,12' is used to send the form feed character, in this case 12. '@0,12' can also be used to position a data item: '@0,12"TITLE"'.

### Obtaining Printer Features

You can send any sequences of control characters to your printer with this technique:

1. Identify the ASCII values to be sent.
2. Put them into a variable with a command such as:  
**a\$=chr\$(27)+chr\$(14)**. Type this on the Command Line from Menu 1 and press RETURN.
3. Type the command: **print a\$**, and press RETURN.

You can include this kind of instruction in the "start" program or in any of your report programs.

### **Secondary Addressing To Obtain Printer Features**

Some IEEE printers require secondary addressing to access their print features. The 'pdev' command can be used to send secondary address values.

The 8023P, for example, requires a secondary address 13 to be sent to switch on condensed print mode. This is achieved with:

```
pdev4,13,0:print"":pdev4,255,0
```

Notice the subsequent printing of a null string and the resetting of the printer with the normal 'pdev' values. This is essential, and the procedure must be repeated in full each time a secondary address value is sent. Follow the same procedure for similar printers.

## **2. PRINTER TYPE DEFINITION**

Define your printer type with the **pdef** command (see Programming Section 2.1.33). You can select from:

**pdef 0** Commodore dot matrix.

**pdef 1** Epson dot matrix. Covers all popular dot matrix printers.

**pdef 2** Daisywheel. Covers Diablo, Qume, etc.

**pdef 5** Also Epson, but sends Commodore ASCII codes in case you have an interface that converts from Commodore to standard ASCII.

**pdef 6** Also daisywheel, sending Commodore ASCII codes.

Low cost daisywheel type printers should work with either 2 or 6; you may need a printer buffer. For non-Commodore dot matrix printers, select 1 if you're not using an interface, or 5 if you are.

## **3. CONNECTING THE PRINTER**

You connect up either an interface cable or an interface device and then use the **pdev** command to tell Superbase what you're using (see Programming Section 2.1.34).

### **Interface cables**

Commodore serial printers such as the 1525 or MPS801 are connected to the 1541 or 1571 disk drive with a Commodore supplied cable. The command is **pdev 4,7,0**.

The 1526 printer requires the following initialization:

**pdev4,7,0:print"" :pdev4,0,0**

Parallel printers such as an Epson type with a Centronics interface are connected directly to the computer's User port. NO OTHER INTERFACE DEVICE IS REQUIRED. The command is **pdev 0**.

There are a variety of Centronics interface cables available, not all will work. The reason for this is that there are a number of ways of connecting the pins to the user port. See the following table for the correct method to use with Superbase.

<b><u>PINS ON 64</u></b>	<b><u>Cable PINS</u></b>
a Ground	33 Ground
b Flag2	10 Acknowledge
c PB0	2 Data1
d PB1	3 Data2
e PB2	4 Data3
f PB3	5 Data4
h PB4	6 Data5
j PB5	7 Data6
k PB6	8 Data7
l PB7	9 Data8
m PA2	1 Data Strobe
n Ground	16 Ground

The above settings will work for the majority of printers.

Your interface cable may have been supplied with a software cassette containing a machine code program or "wedge" designed for addressing the printer and sending the appropriate codes. These programs reside in memory and allow you to operate on your own programs at the same time.

When using Superbase, however, it is not necessary to use any other programs as Superbase has its own printer software.

**Interface devices**

IEEE printers such as the Commodore 6400 or 8023 must be connected via an external device, such as the Brain Boxes interface cartridge, which plugs into the expansion port of the computer. The command is **pdev 4,255,0**. (Some devices may inhibit color definition.)

The 8023P requires the following initialization:

```
pdev4,255,0:print"":pdev4,255,0
```

See Secondary Addressing above.

RS232 printers must be connected via an external device. The **pdev** command must be followed by three numbers whose values you must work out using the tables and rules in the next section.

**Recommended devices**

There are many interfaces and cables on the market. Unfortunately, some are unreliable and do not handle Superbase's output correctly. Superbase's output utilizes standard Commodore kernal subroutines, and if your interface does not work as you think it should, this is not necessarily evidence of a deficiency in Superbase. However, you are welcome to send us details of interface malfunctions and our technical department may be able to help.

<b>IEEE</b>	Brain Boxes IEEE
<b>RS232</b>	Brain Boxes RS232 64/128
<b>Parallel cable</b>	Brain Boxes Centronics to User port

At time of publication, these are the only devices known to work with the commodore 128 in native mode. Other devices may also work with Superbase but the implementation may be less than 100%.

Brain Boxes are at: 25 Lynmouth Rd.,  
Aigburth,  
Liverpool,  
England,  
L17 6AW

Telephone: 051 427 1526

**4. CALCULATING RS232 VALUES**

Users of the RS232 port will need to set the values of 2 CONTROL REGISTERS in order to instruct the computer how to send data to the printer. These values are entered with the command 'pdev 2,x,y' where 'x' is the value described in 3 and 'y' is the value described in 4 below.

1. Study your printer manual and note down the correct settings for these items:

baud rate  
 data word length  
 number of stop bits  
 handshake  
 parity

2. For each of these items, look up the corresponding value for the required setting in the appropriate table below, and note the value down.

BAUD RATE	VALUE	DATA WORD LENGTH	VALUE	NO. OF STOP BITS	VALUE
50	1	8	0	1	0
75	2	7	32	2	128
110	3	6	64		
134.5	4	5	96		
150	5				
300	6				
600	7				
1200	8	PARITY			
1800	9	TYPE	VALUE	HANDSHAKE	VALUE
2400	10				
3600	11	disabled	0	0-3	0
4800	12	odd	32	1-X	1
7200	13	even	96		
9600	14	mark transmit	160		
19200	15	space transmit	224		

3. Add together the values you obtained for baud rate, data word length, and number of stop bits. Enter this as 'x'.
4. Add together the value for parity and handshake. Enter this as 'y'.

Example

baud rate 9600                      - value 14  
 word length 8                        - value 0  
 no. of stop bits 1                   - value 0

total value - 14

Enter 14 as 'x'

parity odd                   - value 32  
handshake 0-3               - value 0

Enter 32 as 'y'

If your printer doesn't seem to work properly, try decreasing the baud rate. If it only works properly below the expected baud rate, this may signify an incorrect value for handshake. Change the 'y' parameter value.

**5. COMMODORE 1520 PRINTER PLOTTER**

The 1520 printer plotter requires instructions to be sent from BASIC before Superbase is loaded.

Type in the following instruction:

**open4,6:open6,6,6:print#4:print#6,1:close4:close6**

Check that you've typed this in accurately, and press RETURN. The printer will then be set-up, and you can now load Superbase in the normal way.

**SUPERBASE SATELLITE PROGRAMS:**  
**"LABELS," "DELETE," AND "UTILITY"**

Three programs are provided on your Superbase program disk to perform useful database support functions. They are:

1. The **"labels"** program. This is an enhanced version of the Superbase Version 4 "labels" program. It's a Superbase program and can only be run from within Superbase.
2. The **"delete"** program. This makes it easier to remove groups of records or entire files from a database. It's also a Superbase program to be run only from within Superbase.
3. The **"utility"** program. This is a machine code program and cannot be run from within Superbase. It has a number of functions for copying and recovering database and other files.

All these programs are self-prompting and easy to run.

**1. LABELS**

Load up Superbase in the normal way. Your first task is to copy the two labels programs from the Superbase program disk onto your data disk. Insert the program disk into the drive, and from Menu 1 type **'load "labels"'** and press RETURN. When the drive stops, re-insert your data disk and from Menu 1 type **'save "labels"'** and press RETURN. Repeat this procedure for the second module, which is called **"makelabels"**. Now you have working copies on your data disk, which you should use whenever you need to print labels from a database file.

**Running "labels"**

Select a database and from Menu 1 type **'execute "labels"'** and press RETURN. The program runs automatically. The main features of the program are summarized below:

- \* Select from one to four columns of labels.
- \* Define label width in columns.
- \* Define label depth in lines.
- \* Select a whole file for processing or use a sub-group list obtained with FIND.
- \* Select up to 48 fields per label -- 12 lines of four items each.
- \* Use any data type -- text, numeric, date, etc.
- \* Use up to four fields per label line, mixed type if desired.

- \* Print labels from data input directly through the keyboard instead of from a database file.
- \* Print serial number on labels by entering "+#" in direct keyboard mode.
- \* Request multiple copies of each label, up to 9999.
- \* Include forced spaces to format label lines (e.g. by centering).
- \* Retain label layouts for future use, so you don't have to re-enter the fieldnames, etc.

Further details are provided on the "labels" help screen. Press the HELP key, enter the name "labels" and press RETURN (you must have the file "h8labels" or "hlabels" on the disk for this to work).

## **2. DELETE**

Use the method described above in the Labels section to put the "delete" program on your data disk.

### **Running "delete"**

Select a database and from Menu 1 type 'execute "delete"' and press RETURN. The program runs automatically. First you choose what type of file to delete: database, help screen, program, or list (this includes other general text files).

### **Record deletion**

You can choose to delete a group of records using a list you obtained with FIND, or all the records in a given file.

### **File deletion**

You can choose to remove the file format from the database catalog (what you see when you select the FILE Option), or leave it there, but without any records in it.

Even if you choose to remove it from the database catalog, it will still be present on the disk, and will appear in a file in the disk directory. This means that if you create another database file with the same name, that format will be used automatically.

If you don't want this to happen, remove the file format from the disk by choosing this option when it's presented, as part of file deletion.

**3. UTILITY**

This general utility program comes on your Superbase program disk. It can only be run on the Commodore 64. To use it, switch on your computer and wait for the READY prompt. On a C64, type in:

**load "utility",8**

and press RETURN.

C128 users, switch to C64 mode by typing:

**go64**

and press RETURN. Answer 'Y' to the 'Are you sure?' prompt. Then type in:

**load "utility64",8**

and press RETURN.

**"Utility" functions**

This program helps you to manage your database better. Everyone encounters floppy disk and other kinds of errors from time to time, and may need to move a damaged database to a new disk. This program may be able to help. A list of its main functions follows.

- \* Split a database.
- \* Copy file definitions onto a formatted disk.
- \* Copy uncorrupted database records and indexes to a formatted disk.
- \* Convert Version 1 databases to Version 2 (see Appendix F).
- \* Recover most corrupted database records and indexes.
- \* Copy sequential files from one formatted disk to another, including database file definitions, lists, memos, help screens, programs, and Easy Script or Superscript files. Maximum file length for this operation is 92 blocks.

The utility program cannot recover deleted records or files.

**Configurations**

File and database copying functions can use any of the following configurations:

Single drive	(drive 0)
Dual drive	(put destination disk in drive 1)
Dual units	(put destination disk in drive 0 of unit 9)

**Important: multiple database situation**

You can only copy a database from a disk that contains one database. If you have more than one database on a disk, follow these steps:

1. Make extra backups of the disk, one per database.
2. Scratch databases until you have one per disk. Press SHIFT and the Commodore key to switch to lower case, then use the following BASIC program:

```
10 open 8,8,15,"s0:filename"  
20 close 8
```

Type it in, substituting your database name for "filename," pressing RETURN at the end of each line. If your database has a number or other non-alphabetic character in it, this must be entered into the name as the corresponding graphics character. Then type 'run' and press RETURN. Repeat for each database.

3. Run "utility" as described above, using your disks as source disks one after the other. You can have one, or more than one, destination disk.

**Database compression**

Databases copied with "utility" may appear smaller than in their original state. This is because "utility" releases unused space and compresses the database.

**SUPERBASE VERSION AND MACHINE COMPATIBILITY**

Superbase 128 will read Superbase 64 or Plus/4 databases from a 1541, 1571 or an appropriately connected 4040 or other IEEE drive. In the process, it converts any Version 1 to Version 2 format, making it unreadable by Superbase 64 Version 1. We strongly advise you to use the copy function of the "utility" program (see Appendix E) to produce a compressed database prior to changing from Superbase 64 to Superbase 128.

The 1571 drive is essentially a double-sided 1541. Superbase 128 takes advantage of the former's extra speed by using its "fast" and "burst" modes to improve database and file access times. Note that 1571 disks are not readable on a 1541 unless only one side has been used.

The Superbase 128 program disk will work with both 1541 and 1571 drives.

**UPGRADING AND CONVERTING FROM 40 TO 80 COLUMN FORMAT**

The Commodore 128 supports both 40 and 80 column monitors, but each screen mode requires a separate kind of output and physical connection. If you are converting formats from Superbase 64, keep your video connector as you will need it for the procedure outlined below.

**Conversion from 40 to 80 Column File Format**

1. Connect your 40 column monitor and load Superbase 128.
2. Insert your Superbase 64 data disk and select a database in the usual way.
3. Print out file formats for all the files to be converted, by selecting the file and typing '**print:status:display**'.
4. Quit from Superbase, change your configuration to 80 column, and reload.
5. Insert your Superbase 64 data disk and select a database in the usual way -- do not select a file.
6. Scratch the file format from the disk with '**maintain o"s0:filename"**'.
7. Reselect the database.
8. Select the FILE Option and give the name of the old file.
9. Redefine the file preserving the same field order and types, using the printout from Step 3 to help you. You can reposition fields as long as you don't change their order.
10. Repeat Steps 6-9 for each file to be converted.

You can now access your data as before.

**Selecting File Format of Alternate Mode**

If you select a file of the alternate mode to the one of the currently active monitor, you will see one of these two displays:

- \* 'Screen deselected' on the 80 column monitor.
- \* A blank screen on the 40 column monitor.

The situation is this:

- \* The keyboard appears to be dead.
  - \* The system is really alive.
  - \* You need to type in some commands to restore the correct display.
  - \* You won't be able to see what you type -- so type carefully.
1. Type '**database**' and press RETURN.
  2. Type '**name**' and press RETURN, where 'name' is the current database.
  3. Hold down CONTROL and press 'Q' once.
  4. Type '**mode**' and press RETURN.

**ESCAPING FROM COMMANDS**

From time to time, everyone selects a command by mistake. You need to know how to get out of every command quickly.

This is a table of Function Keys and how to escape from each selection:

<u>Menu 1</u>		<u>Menu 2</u>	
Enter	f1 Q	File	RETURN
Select	RETURN	Format	f1 Q
Find	f1 Q	Batch	RETURN
Output	RETURN	Sort	RETURN
Calc	RETURN	Prog	f1 Q
Report	RETURN (1)	Maintain	RETURN
Execute	RETURN (2)	Memo	RETURN
Help	RETURN	Help	RETURN

- (1) Once past the first prompt you must go right through all prompts.  
 (2) Except that when a program is in memory it runs.

**Select**

Key	You must enter a character (such as "z") and press RETURN to get to end of file. Then press RETURN.
Current	RETURN
Next	RETURN
Last	RETURN
Previous	RETURN
First	RETURN
Match	f1 Q
Output	RETURN after output and RETURN for the menu
Add	f1 Q
Replace	f1 Q
Delete	Enter 'n' to 'Confirm Deleted Record?' and press RETURN

**Maintain**

Status	STOP
Catalog	RETURN after output
Import	RETURN
Export	RETURN
Directory	STOP
Backup	STOP STOP
New Disk	STOP
Other	RETURN

In BACKUP and NEW DISK you can quit by entering 'n' to the 'Are You Sure?' prompt.

- abbreviations T-17, T-18, T-47,  
R-33, R-40, R-54, P-2
- across T-32, T-45, R-39, R-49, P-18
- add record R-39
- all R-47, P-11, P-18
- all/from "list" T-30, T-32, T-45,  
R-45, R-55, R-60, R-65, R-70,  
P-11
- ask P-19
  
- backup T-40, R-78, R-79, R-81
  - command P-45
  - dual drive R-83
  - hard disk R-81
  - single drive R-82
- BASIC P-2, P-36
  - commands P-4, P-16, P-17
  - errors A-10
- batch T-29, T-30, R-4, R-65, P-45
  - semicolons R-66
- baud rate A-22
- blank disk, formatting R-78
- blank line R-53, R-68
  - insert P-2
- blank record R-27, P-46
- blocks free R-78
- boot program see start program
- border A-12
  - drawing R-20
  - erasing R-20
- brkoff command P-20
- brkon command P-20
  
- calc T-28, R-2, R-62, P-46
  - semicolons, R-63
  - new record R-64
- calculations T-26, T-28, T-29, A-1
- calendar field R-12, R-17
- catalog T-22, T-23, R-77, P-46
- CBM serial interface P-33
- centronics interface P-33, A-20
- changing screen R-17
- check command P-20
- check memory P-3
- clear command P-21
- clear record R-64
- clear screen R-54, R-57
- close command P-21
- color R-19
- cols command P-21
- command line T-16, T-47, T-48, R-5,  
R-40, R-43, R-56, R-64, R-66,  
P-5, P-18, A-12
- maximum length R-5
- recall key T-48, R-5
- command, primary R-6, P-18
- command, secondary R-6, P-18
- commands
  - abbreviating P-2
  - interrupting R-7
- command string P-11
- compatibility A-28
- conditionals P-6
  - else P-8
  - end of file P-7, P-26
  - end of list P-7, P-27
  - if ... then P-8
  - nmat P-8, P-31
  - pmat P-8, P-35
- constant field T-28, R-11, R-17
  - maximum length R-17
- continuous print R-70
  - command P-22
- convert command P-22
- copy R-84, R-85, P-47
- copy program line P-2
- count command P-23
- currency sign R-16, R-73
- cust.inv P-18
- cust.rec P-18
  
- daisywheel printer type P-32
- database T-7, A-13
  - creating T-7
  - catalog T-22, T-23
  - command T-37, T-38, R-58, P-23
  - compression A-27
  - convert A-26
  - copying T-38, A-26
  - deleting R-84, R-85
  - device number T-38
  - maxima A-1
  - naming T-7, T-38
  - recovery A-26
  - reorganizing R-57
- data disk A-12
  - creating T-2
- data storage A-2
- data word length A-22
- date field T-12, T-15, R-10, R-15,  
R-29
  - command P-24
  - range R-10
- day of the week T-15, R-10, R-29
- decimal places T-25
- default values A-3, A-13
- delete utility A-25

- deleting
  - characters R-88
  - field R-18
  - group of records P-10
  - line R-18, R-88, P-3
  - record R-40
- descriptive text R-9, A-13
- deselect A-13
- destination R-81, R-82
- detail command P-24
- detail, in report T-54, R-69
- device T-38
- directory R-78, R-84, P-47, A-13
- disk
  - commands R-78
  - duplicate R-84
  - formatting R-83, A-13
  - other commands P-52
  - space T-38, P-47
- disk drive
  - dual R-71, R-75, R-78, R-83, P-2
  - error light R-79
  - 1541 A-28
  - 1571 A-28
- disk id T-3, R-82, R-84
- disk name T-3, R-82, R-84
- disk training T-4
- display T-32, T-45, T-46, R-39,  
R-45, R-57, R-67
- command P-24
- do command P-25
- dot matrix printer type P-32
- down T-32, T-45, R-39, R-47, R-49,  
P-25
- dummy field for import R-81
- dump command P-25, P-38
- duplicate R-84
- duplicate keys T-13, T-42, R-21,  
R-22
- Easy Script R-55, R-56
- edit record see replace
- elink (end or reverselink) command  
P-25
- else P-8
- encrypting programs P-36
- end of file P-7, P-26
- end of line, move to P-2
- end of list P-7, P-27
- end of page R-49
- end of report T-54
- endreport P-26
- enter command P-47
- enter option T-14, T-27, R-1, R-27  
summary R-30
- eof (end of file) command P-26
- eol (end of list) command P-2, P-27
- epson printer type P-32
- erasing format text R-18  
field R-18  
line R-88
- error messages A-5
- errors
  - BASIC A-10
  - handling R-73, P-14
  - programs P-15
  - superscript P-15
  - trapping P-14
- example records T-21, T-36
- execute R-2, R-71, R-74, P-2, P-48  
from disk R-74
- exit Superbase T-20
- export R-23, R-77, R-79, R-80, P-48
- field length T-9, T-25
- fieldname T-26, R-9  
maximum length R-9  
rules P-15
- fields T-8, R-8, A-1, A-14  
changing R-15, R-23  
money see numeric  
setting, errors T-10  
setting summary T-9, R-13  
start and end markers R-13, A-14
- field types T-9, R-10
  - calendar R-12
  - constant R-11
  - date R-10
  - forced R-10, R-13
  - key R-10
  - numeric R-10
  - replica R-13
  - result R-11
  - text R-10
- file command P-49
- file format T-8, R-8, A-14  
changing T-42, R-22  
clear R-21  
converting 40 to 80 column A-29  
copying A-26  
creating T-23, R-24  
duplicating R-23  
no fields defined R-9  
quit R-21  
saving T-13, R-21
- file manager A-3

- file option R-3, R-58, R-77
- files
  - changing R-59
  - disk T-40
  - format printout T-39, R-8, R-20
  - joining R-85
  - linking P-9
  - maxima A-1
  - multiple programming R-33
  - naming T-7, T-39, R-7, R-58
  - recovering T-39
  - report R-67
  - scratching R-84, R-85
- fill R-55, R-56, A-14
- find command P-49
- find option T-31, R-2, R-42, P-32
  - see also matching
- forced field R-10, R-13, R-14, R-29
- format blank disk R-78
- format command P-50
- format option T-8, T-23, R-3, R-8, R-22, R-58
- form feed R-54, R-72, A-18
- 40/80 column screen P-21, P-31
- from command P-27
- from "list" R-48, P-11
- function keys T-6
- graphics symbols R-20
- h8list/hlist T-31, T-41, R-42, R-60, P-49
- handling errors P-4
- handshake A-22
- hard disk backup R-81
- help T-22, T-41, R-2, R-4, R-42, R-90, P-50
  - printing R-90
- hslist R-61
- IEEE interface P-33, A-21
- if ... then P-7
- import R-23, R-77, R-78, R-80, P-50
- index system A-2
- initialize R-84
- insert blank line P-2
- insert characters R-88
- inserting line R-18, R-88
- integration with superscript P-10
- interface
  - CBM serial P-33
  - centronics P-33, A-20, A-21
  - IEEE P-33, A-21
  - RS232 P-33, A-21, A-22
- interrupting commands R-7
- invalid characters R-27
- invalid line P-3
- inverting T-12, A-15
  - line R-19
  - screen R-19
- invoice design T-35
- is command P-28
- item list A-15
- key A-15
  - code T-43
  - duplicate T-13, T-42, R-22
  - exists R-40
  - not found R-33
  - numbers as T-43
  - partial R-33
  - select T-18, R-33, P-10
  - unique T-42, R-22
- key field T-9, T-14, R-10, R-14, R-27, R-29, R-32, R-60
- maximum length R-14
- key list T-31, T-39, R-42, R-70, P-38, A-13, A-15
  - adding to T-39, R-43
  - default R-42
  - export from R-80
  - naming R-43
  - renaming R-44
  - using R-44
- labels R-73, A-24
- left margin P-29
- lfeed (line feed) command P-28
- line
  - copy P-2
  - delete P-3
  - invalid P-3
- line feed R-70, P-28, A-18
- line numbers T-49
- link command P-9, P-29, P-31
- linking files P-9
- list command P-2, P-3
- lmargin (left margin) command P-29
- load command P-1, P-30
- load "filename" at P-30
- loading Superbase T-2
- loading Superscript P-11
- load report R-71

- mail merge P-41
- maintain R-4, R-76, P-51
- margins R-70, A-18
- matching, T-18, R-34, R-36
  - alternates R-38
  - command P-56
  - delayed request R-38, R-44
  - field-independent R-37
  - finish T-19, R-34, R-37
  - pattern R-38
  - range R-38
  - selection criteria R-35, R-41, R-42, R-43, A-15
  - semicolons R-41
  - sliding R-37
  - start R-36
- maximum system values A-1
- memo T-40, T-41, R-4, R-87, R-90,
  - P-38, P-51
  - 40/90 column R-89
  - clear R-88
  - help memo T-41, R-87
  - multi-screen R-89
  - print R-88
  - quit R-88, R-89
  - store R-88
- memory check P-3
- menu 1 T-14, R-1
- menu 2 T-14, R-3
- menu command P-30
- menu screens T-41
- mode command P-31, A-30
  
- new command R-74, P-2
- new disk command R-84, P-51
- new page R-54
- nmat (no match conditional) command P-8, P-31
- numeric field T-24, R-10, R-16
  - decimals T-25, R-16
  - maximum length R-16
  - modifying R-30
  - rounding R-16
  
- open command P-32
- other disk commands P-52
- output T-31, T-32, T-45, R-2, R-45,
  - P-52, A-18
  - across R-49
  - all R-47
  - blank lines R-53
  - calculations R-51
  - column position T-33, T-45, R-52
  - direction R-45
  - displaying R-48
  - down R-47, R-49
  - fill R-55, R-56
  - format R-51
  - formulas R-50
  - joined data R-48
  - left justified R-52
  - line 1, column 1 R-53
  - line position R-53
  - numeric R-51
  - numeric format T-33, T-46
  - overflow R-52
  - positioning R-52, R-54
  - printing R-48
  - selected fields R-48
  - stopping R-47
  - summary R-46
  - text R-50
  - to disk file R-23, R-55
  - truncation R-51, R-52
  - variables R-50
  
- page length R-70, A-18
- paper length P-34
- parameter A-16
- parity A-22
- partial match P-35
- pdef (printer definition) command P-32
- pdev (set printer device) command P-33
- perform command P-33
- plen (paper length) command P-34
- plotter, 1520 A-23
- plus R-72, P-34
- pmat (partial match) P-8, P-35
- pound sign R-73
- print T-32, T-45, T-46, R-39, R-45,
  - R-67, P-35, A-18
  - program P-3
- print report T-55
- printer definition P-32, A-19
- printer features P-35, A-18
- printers A-18
  - connecting A-19
- printer type R-70, A-19
  - Commodore P-32, A-19
  - daisywheel P-32
  - dot matrix P-32
  - epson P-32
  - 1526 A-20
- prog command P-53

- program T-49, T-55, R-4, P-1, P-5,  
P-18, A-16
  - clear R-74
  - editing facilities P-2
  - encrypting P-36
  - errors P-15
  - list P-3
  - list to disk P-3
  - loops P-5
  - print P-3
  - run P-3
  - save P-2
  - size P-4
  - superscript P-15
  - variable space P-14
- program editor P-1
  - exit P-2
- protect command P-36
  
- quit T-16, T-20, T-34, T-37, P-36
  
- records A-16
  - deleting a group P-10, A-25
  - maxima A-1
  - number of T-23
  - replacing P-40
  - storing T-15, T-28, T-29, R-28,  
R-29
  - template A-16
  - viewing T-16
- registration T-34
- rename R-84, R-85, P-53
- renumber command P-4
- replace T-29, R-40
- replace record P-40
- replica field R-13, A-16
- replicate record R-39
- report T-51, T-52, R-2, R-67, R-72,  
P-53
  - blank line R-68
  - computed field R-70
  - detail T-54, R-69, P-24
  - end of report T-54, R-70, R-72,  
P-26
  - file T-52, R-67
  - load R-71
  - plus R-72, P-34
  - printing T-55, R-67
  - program T-55, R-71, R-72
  - sorted order R-69
  - store T-54, R-71, P-40
  - subtotal break T-53, R-69
  - subtotals T-52, R-68, R-72, P-41
  - title T-52, R-68, P-42
  - totals T-52, R-68, P-43
- restart command P-36
- result field T-25, T-26, R-11, R-16
  - formula R-12, R-21
  - maximum number of R-16
- return from link P-37
- reverse video R-73
- right margin P-37
- rlink (return from link) command  
P-9, P-37
- marg (right margin) command P-37
- rounding R-16, R-51
- RS232 interface P-33, A-21, A-22,  
A-23
- run command P-3
  
- save command P-37
- save program R-75
- save report T-54
- scratch
  - command P-54
  - database R-86
  - file R-84, R-85
- screen
  - blank A-30
  - changing R-17, R-33
  - clear R-54, R-57
  - color R-19
  - command P-38
  - default R-28
  - dump R-20, R-28, R-88, A-17
  - 40/80 columns P-21, P-31
  - multiple T-43, R-17
  - new T-44
  - redisplay P-2
- screen deselected P-31, P-49, A-30
- search see matching
- secondary addressing P-33, A-19
- select T-31, R-1, R-32, P-54
  - add R-39, P-56
  - criteria R-35
  - current R-33, P-55
  - delete R-40, R-85, P-36, P-57
  - first T-17, R-34, P-55
  - from list R-41, P-55
  - key T-18, R-33, P-10, P-31, P-54
  - last T-17, R-34, P-55
  - match see matching
  - next T-17, R-34, P-55
  - output R-39, P-56
  - previous T-17, R-34, P-55
  - record template R-35
  - replace R-40, P-57

select menu T-17, R-32  
semicolons R-57, R-63, R-65, R-68  
separators R-79  
sequential file R-77, R-78, A-17  
    copying A-26  
set command P-38  
setlink command P-9, P-39  
shift key T-6, T-23  
sort T-51, T-53, R-4, R-60, P-58  
    ascending R-60  
    descending R-60  
    fields R-61  
    for report R-69  
    intermediate file R-61  
    list R-61  
    parameters A-17  
source R-79, R-81, R-82  
space command P-40  
space, text and variable P-14  
spacing R-70, P-40, A-18  
start program R-83, A-3, A-4  
start-up screen T-2  
status R-77, P-58  
stop bits A-22  
stop key P-20  
store command T-29, P-40  
string A-17  
subtotal break field T-53, R-69  
subtotal command P-41  
subtotals, in report T-52, R-68  
    cleared R-72  
    retained R-72  
Superbase, loading T-2  
    returning to P-11  
Superscript R-55  
    command P-11, P-41  
    document P-13  
    errors P-15  
    integration P-10  
    loading P-11  
    mail merge P-41  
    restoring full text area P-11  
  
text field T-10, R-10, R-15  
    maximum length R-15  
text length P-42, A-18  
text space P-14  
title command P-42  
title, report T-52, R-68  
tlen (text length) command P-42  
total command P-43  
totals, in report T-52, R-68  
totals, running T-49, T-50, R-65  
training disk T-4  
trapping errors P-14  
truncation A-17  
tutorial introduction T-1  
  
underlining T-46, R-54, R-73  
unit number P-23  
updating R-65  
upgrading A-29  
utilities R-76  
utility program A-26  
  
validate R-34, P-58  
variables T-49, T-50, R-63, P-33,  
    A-12  
    rules P-15  
    values P-38  
variable space P-14  
varspace command P-43  
view directory P-58  
  
wait command P-44  
where command P-44  
word processing T-46  
word processor (Superscript) P-41  
work space, create P-2

