FirstBase

Distributed by JP Software (tm)

Table Of Contents

Chapter 1: Introduction.....2

What to do first.....5
Equipment you need.....5
How to start.....6
How to end.....6

Customizing FirstBase.....7
Important differences.....8
Navigating the system.....9
Working with multiple disks.....11
Summary.....13

Chapter 2: The Rules of FirstBase.....15

Installation.....16
Specifications.....19
Database Conventions.....20
Filenames, Field attributes,
Text/numerics, Keypresses, Wildcards

Search formats.....25
Case, Delmimiters, Partial strings
Wildcards, BROWSE formats, QUERY formats

Summary.....30

Chapter 3: The CREATE Program.....32

Create new database.....34
CREATE's command prompt.....37
List, Change, Add, Save, Delete

Change database in memory.....39
Load definition.....39
Change drives.....39
Summary.....40

Chapter 4: The BROWSE program.....42

Browse records.....44

Displaying records.....44
Displaying deleted records.....45
Displaying indexed status.....46

The BROWSE prompt.....46
Cur,Fd,Bk,Top,Last,Ind,Add,Edit,
Save,Del,Undel,Key,Print

Move to new database.....50
Change system configuration.....50
Summary.....51

Chapter 5: The QUERY program.....53

Query command format.....55
Query destinations.....59
Number formats.....60
Query examples.....61
Summary.....63

Chapter 6: The UPDATE program.....65

Update command format.....66
Arithmetic, String constants
Operator precedence

Using DELETE and UNDELETE.....68
Examples of update commands.....69
Examples of DELETE and UNDELETE.....69
Summary.....70

Chapter 7: The SORT program.....71

Sorting the database.....72
Fields to sort, Ascending/descending
New file/existing, sort index/db

Sorting example.....73
Summary.....74

Chapter 8: Macros.....75

Where to use macros.....76
How to save a macro.....76
How to load a macro.....76
Example of query macro.....77
Example of update macro.....77
Example of sort macro.....77
Summary.....79

Chapter 9: The DEFINE program.....80

Use of OUTPUT DRIVE and temporary files.....82
Define report.....83
Define prompt.....83
Define page format.....83
Record format.....87
Rows & columns explained.....87
Define record format.....88

Change report in memory.....89
Display, Change, Save, Load

Load report.....90
Summary.....91

Chapter 10: Report Generator.....93

Selecting records.....94
Using the list file
Printing ALL records

The report name.....95
Printing the report.....95
Summary.....96

Chapter 11: Tutorial.....97

Creating a database.....98
Using BROWSE.....102
Using QUERY.....103
Defining a report.....105
Printing a report.....108

Appendix A: Error messages.....109

Appendix B: FirstBase file formats.....117

CHAPTER 1 - INTRODUCTION

This chapter serves as a brief introduction to FirstBase; here you will learn what the program can do, and how to move around and use the wealth of functions available. By learning the differences between menus and commands prompts, you'll discover how to navigate FirstBase with little fuss. By reading this chapter, you're given all of the information you need to get up and running quickly.

If you are an adventurous soul and feel that manuals are for the timid, please take a few minutes to at least read this chapter before plunging ahead...you won't be disappointed!

What follows is a brief outline of how the Introduction chapter is organized.

Program Introduction.....3

What to do first?.....5
- Equipment you need.....5
- How to start.....6
  - Extended BASIC.....6
  - Editor/Assembler....6
  - Ti-Writer.....6
- How to end.....6

Customizing FirstBase.....7

Important differences.....8
- Indexing.....8
- Text fields.....8
- Numeric fields.....8
- Data entry.....8

Navigating the system.....9
- Menus.....9
- Command prompts.....10

Working with multiple disks..11
- EXTRAS program.....11

Summary.....13

Section 1 - Program Introduction

Welcome to FirstBase, a flexible database manager that allows you to perform functions never before possible on a TI99/4A. As with any good database manager, FirstBase allows you to store, change, organize, and retrieve all kinds of information. But the program goes a step further, giving you the ability to perform sophisticated searches with less effort. You can manipulate data faster, too. For example, you can subtract 10% from all the PRICE fields with a single command. A powerful report generator allows you to capture on paper exactly the information you want, formatted exactly to your needs.

FirstBase is a collection of individual programs (stored as program image files), each one performing a specific database task.

Here is a list of each program's function:

CREATE: This program allows you to create a new database. By specifying the names, sizes, and characteristics of the database fields, you create a database definition, after which you can add records to your new database.

BROWSE: You can add a new record to the database, save the record to disk, edit an existing record, and perform common but essential tasks such as printing and deleting individual records. You can also do single-field searching.

QUERY: In addition to the common functions in the BROWSE program, FirstBase allows you to perform complex searches (called queries), similar to what you can do on IBM-compatible systems. A query allows you to find a group of records according to field contents, such as finding all California residents with an income over $40,000, or all the Smiths in the state of Colorado. Once the program finds the records you're looking for, it can display them to the screen, send them to a new or existing database file, or print them in a report.

UPDATE: With FirstBase, you have at your disposal the very powerful capability to do batch processing, which allows you to store a string constant or a calculated result in a single field within a group of records. For example, you can give everyone in your New York office a 2% raise.

SORT: You can sort a database in ascending or descending order, and you can specify as many keys on which to sort as you have indexed fields. In other words, if a field is indexed, it can be used for sorting. Therefore, you can theoretically sort on up to 75 keys!

REPORT: This versatile program generates custom printed reports of your data. You can print all of the records in a database, or only those you've previously selected with a QUERY command.

DEFINE: The DEFINE program lets you create custom report formats that you use in the REPORT program.

A feature that distinguishes FirstBase from the rest of the pack is its ability to use *MACROs*. A macro is like a tape recording of the commands you type: they can be played back at any time. You can store the keystrokes of the QUERY, UPDATE, and SORT commands on disk by a simple menu selection; later you can perform the same commands just by recalling the macro you've stored to disk. This makes complicated and repetitive tasks a breeze.

Section 2 - What to do First?

BACK UP YOUR DISKS!

FirstBase is not copy-protected, so you can make archival copies of all your disks for safe-keeping. Also, always work from copies of your original disks! This prevents you from inadvertently damaging your originals. Once you've copied your disks and put them away in a safe place, you're ready to get down to the business of running the program.

Equipment You Need

In order to run FirstBase, you need:

- TI-99/4A computer (or a Myarc 9640 running in GPL mode)
- 32K memory expansion
- One disk drive (two drives recommended)
- One of the following cartridges:
    - Extended BASIC
    - Editor/Assembler
    - Ti-Writer

Other equipment that FirstBase can use, but is not required:

- RAM disk
- Printer
- Hard Disk

NOTE: FirstBase cannot store data on the old hard drive systems which use the Myarc Personality Card. The new Myarc HFDC, however, is compatible.

How to Start

Here are loading instructions organized according to the cartridge you have plugged in the computer.

For Extended BASIC:

1. Turn on your monitor, expansion box (if you have one), and computer.
2. Insert the Extended BASIC module into the cartridge slot.
3. Insert the FirstBase program diskette in drive one.
4. Bypass the main title screen by pressing any key, then select (2) for "Extended BASIC." The program will automatically load and run.

For Editor/Assembler:

1. Follow step (1) under Extended BASIC.
2. Insert the Editor/Assembler module into the cartridge slot.
3. Insert the FirstBase program diskette in drive one.
4. Bypass the main title screen by pressing any key, then select (2) for "Editor/Assembler."
5. At the Editor/Assembler menu, select (5) for "Run Program File."
6. At the "Filename?" prompt, simply press <enter>. The program will automatically load and run.

For Ti-Writer:

1. Follow step (1) for Extended BASIC above.
2. Insert the Ti-Writer module into the cartridge slot.
3. Insert the FirstBase program diskette in drive one.
4. Bypass the main title screen by pressing any key, then select (2) for "Ti-Writer."
5. At the Ti-Writer menu, select (3) for "Utility."
6. Accept the filename of "DSK1.UTIL1" by pressing <enter>. The program will automatically load and run.

How to End

Each program in the FirstBase system has a main menu. The last item on this menu, no matter which program you are running, is "Quit FirstBase." Always end a session by using this menu option! In some cases, you may lose data if you shut the machine off without quitting properly.

Section 3 - Customizing FirstBase

FirstBase is a large software package, consisting of many separate programs. All of these programs communicate with each other through a *system configuration file* on the program disk. This file contains such information as:

- printer devicename
- input data drive
- output data drive
- program drive
- wildcard characters

The first thing you may want to do after running FirstBase for the first time is change the configuration file to fit your system. Please turn to the *Installation* section in chapter 2, *The Rules of FirstBase,* for a detailed discussion on what the file contains and how to change it.

Section 4 - Important Differences

FirstBase differs in some fundamental ways from most databases.

Indexing: Indexing does two things. It determines the order in which records appear in the database, and it enables searching and sorting.

Only indexed fields may be searched and sorted. You have no limit on the number of fields you may index, only that the maximum number of characters in the index not exceed 246.

Text Field: There are two field types in FirstBase; one of them is *text*. A text field can contain any kind of information.

You cannot perform numerical searches or calculations on text fields.

Numeric Field: A *numeric* field can only contain decimal numbers. You cannot type text into a numeric field, and you cannot store text in a numeric field with the UPDATE program.

You can perform numerical searches and calculations on numeric fields.

Data Entry: FirstBase uses a mini-wordprocessor to enter your data. You can use the arrow keys and functions such as delete line and word tab.

Please refer to *CHAPTER 2 - The Rules of FirstBase* for a detailed look at these and other important conventions used in FirstBase. You'll find some very important information in those pages...read it before working with FirstBase!

Section 5 - Navigating the System

There are two major ways to communicate with FirstBase: using *menus* and using *command prompts*. Before we dive into a thorough discussion of all the neat things FirstBase can do, let's take a few minutes to show you how menus and command prompts work in FirstBase.

Menus - moving between programs

FirstBase is a system of interrelated programs; each program performs a specific task, such as selecting records with a QUERY command or generating a report. Each program contains a *main menu*, which you use to move between programs. You can find this type of menu in the BROWSE program, which you see below:

```
================== FirstBase ================

          1 = Browse Records
          2 = Move to New Database
          3 = Change System Configuration
          4 = Extras
          5 = Create
          6 = Query
          7 = Update
          8 = Quit FirstBase

DSK2.MAGS
=============================================
Please make your selection:_
```

A blinking cursor (which looks like an underscore) awaits your response at the bottom of the screen.

> **NOTE: Whenever you see a blinking cursor, the program is waiting for you to do something.**

To make a selection from the list above, press the number corresponding to the action you want to take.

> **NOTE: You need not press <enter> when making a selection from a menu.**

If a menu selection needs to load a program, FirstBase looks for it on the disk named "FIRSTBASE". We call this the *program drive*,

it on the disk named "FIRSTBASE". We call this the *program drive*, and it's stored in the ***system configuration file***.

If your system has only one disk drive, you should see the following message when trying to load a new program:

Please insert program disk..._

Insert the necessary program disk in the drive (if it isn't already there), and hit a key. If FirstBase cannot find the proper programs you see this message:

Program DSK.FIRSTBASE.XXXX not found.

Again the program waits for a keypress, which brings you back to the main menu, where you can insert the proper disk or make another selection.

NOTE: We ship FirstBase with the program disk set for DSK.FIRSTBASE., so that it doesn't matter which physical drive contains the programs. See the *Installation* section in Chapter 3 for details on how to change the program drive.

Command Prompts - moving within a single program

Another way that the user communicates with FirstBase is through a *command prompt*, which allows you to select several functions from within a program. The command prompt you'll see most often is the *Browse Prompt* in the BROWSE program:

Cur,Fd,Bk,T,L,I,Add,Ed,Sv,D,U,Key,P,Q?

Command prompts contain short phrases and single letters, which are abbreviations for functions. In this case, *Cur* stands for Current Record, and *Ed* stands for Edit. When you see a command prompt, type the first letter (upper or lower case) of the command to invoke it. As with menu selections, pressing <enter> is not necessary.

Section 6 - Working with Multiple Disks

FirstBase comes shipped on three diskettes, and their contents are listed below:

```
Disk 1:    CREATE1, CREATE2, CREATE3 (Create program)
           MAIN1, MAIN2, MAIN3             (Browse program)
           QUERY1, QUERY2, QUERY3          (Query program)

Disk 2:    UPDATE1, UPDATE2, UPDATE3 (Update program)
           SORT1, SORT2, SORT3             (Sort program)
           XTRA1                         (Extras program)

Disk 3:    DEFINE1, DEFINE2, DEFINE3 (Report definition)
           REPORT1, REPORT2, REPORT3 (Report generator)
           XTRA1                         (Extras program)
```

We had to spread the programs among three disks simply because they cannot all fit on a single 90K (ss/sd) disk. Feel free, however, to copy all of the programs on a single, larger capacity disk. In fact, you can reorganize the programs anyway you see fit. The only requirement is that you name the program disk(s) *FIRSTBASE*. Each of the three disks already come shipped with this disk name.

EXTRAS program

Notice that disks 2 and 3 both contain the same program, XTRA1. This program merely displays a menu with the following choices:

```
============ FirstBase:Extras =============

              1 = Sort Database
              2 = Print Report
              3 = Define Report
              4 = Change Drives
              5 = Browse Program
              6 = Quit FirstBase

    DSK2.MAGS
    ========================================
    Please Make Your Selection:_
```

Like any other menu, it allows you to move to other programs. In other words, the EXTRAS program lets you load other programs, and

other words, the EXTRAS program lets you load other programs, and nothing more.

Why write a program that is nothing more than a menu loader? Simple: the SORT, PRINT, and DEFINE options would not all fit on the other menus, so we devised a secondary menu, accessible only through the EXTRAS program.

All of this boils down to one important point: the only way to sort a database, define a report, or print a report, is through the EXTRAS program.

Section 7 - Summary

Chapter 1 gave you a brief overview of FirstBase's capabilities, and how to access them. You were shown how to get the program running, how to end, how to handle mistakes, and how to use both menus and command prompts.

- FirstBase consists of a number of programs (stored as program image files), each of which performs a specific task.
- The CREATE program allows you to create a database. This must be done before you can do anything else.
- The BROWSE program lets you enter new records, edit existing ones, delete records, print them, and perform simple searches.
- The QUERY program allows you to perform complex searches, and send the results to a database file, the screen, or print them in a report.
- The UPDATE program lets you perform batch-processing on a group of records; this is where you can perform calculations on fields.
- The SORT program sorts a database in ascending or descending order, and on as many keys as you have indexed.
- The REPORT program prints custom-formatted reports of your data.
- The DEFINE program allows you to create custom report formats for use in the REPORT program.
- You can load FirstBase through Editor/Assembler, Extended BASIC, or Ti-Writer.
- Always exit FirstBase by using the menus; never shut off the machine in the middle of a task!
- You can customize FirstBase by modifying the *system configuration file*. See the *Installation* section in chapter 2 for more information.
- The *system configuration file* defines such things as the printer, data drives, program drive, and wildcard characters.

- Indexing determines the order in which records appear, and it allows searching and sorting.

- A text field can contain any kind of information. You cannot perform calculations or numeric comparisons on text fields.

- A numeric field can contain only decimal numbers. Calculations and numeric comparisons are perfectly legal.

- Menus are used to move between programs; each program has a main menu. Press the corresponding number to select a menu item.

- Command prompts activate functions within the same program. Press the corresponding letter to select a function.

- FirstBase looks for all programs on the disk called "FIRSTBASE". This could be a physical disk, a hard drive emulation, or a RAM disk emulation.

CHAPTER 2 - THE RULES OF FIRSTBASE

FirstBase is a complete database environment, replete with its own set of rules. These rules are not necessarily complex, but a complete understanding of them is vital in harnessing the full power of FirstBase. This chapter is divided into four sections, which deal with everything from installing FirstBase, database capacities, and indexing to search string formats.

Installation.....16
- Default settings.....16
- Definitions.....16
- Location of config file.....18
- Changing config file.....18

Specifications.....19
- Database capacities.....19
- Indexing.....19
- Index size.....19
- Sorting.....19

Database Conventions.....20
- Filenames.....20
- Field attributes.....20
- Text vs. numerical data.....22
- Mini-wordprocessor.....22
  - Keypresses.....22
- Wildcard symbols.....23

Search Formats.....25
- Case.....25
- Delimiters.....25
- Partial strings.....25
- Multiple wildcard character..25
- Search formats in BROWSE.....26
- Search formats in QUERY.....28
- Fieldnames in QUERY, UPDATE..28
- Fieldnames in BROWSE.....29
- Partial fieldnames.....29

Summary.....30

Section 1 - Installation

This section will show you how to customize FirstBase according to your system. It involves changing the *System Configuration File*, which you do through the BROWSE program.

On the program disk is a file named PTR_ , which is the system configuration file. It tells FirstBase how your system is set up: where to look for the program and data disks, and what kind of printer you have.

The first thing you may want to do after running FirstBase for the first time is change this file to match your system.

Default settings

The program comes configured in the following manner:

```
Printer.................. PIO
Input data drive.........  DSK2.
Output data drive........  DSK3.
Program drive............  DSK.FIRSTBASE.
Multiple wildcard........  *
Single wildcard..........  ?
```

Definitions

Here is a complete description for each item in the system configuration file.

Printer:

This is the devicename for your printer. Please consult your printer and RS232 interface manuals for the proper devicename.

The maximum length of the devicename is 26 characters.

Examples are PIO, RS232.BA=9600

Input Data Drive:

This is the devicename (not filename!) where FirstBase looks for data that you are entering or already have entered and saved; it is used in the BROWSE, QUERY, UPDATE, SORT, DEFINE, and REPORT programs.

If you are using a floppy disk controller or a RAM disk, the data drive must be in the form DSKn. where n is the drive number.

If you are using a hard drive with the Myarc HFDC, the data drive must be in the form WDSn. where n is the drive number. You may also have a pathname following the period (such as WDS1.FB.)

The main point to remember is that the device must have either DSKn. or WDSn. -- the drive number must be present!

Do NOT use any other devicenames (like RD. or a volume name like DSK.DATA.).

The maximum length of the devicename is 16 characters.

Output Data Drive:

This device can be any storage device described above. It's used only in QUERY, when you choose to send the results to a new database, and in the SORT and DEFINE programs, where the program stores temporary files. Since a database can take up an entire disk, it's sometimes desirable to specify one drive for reading data and another drive for writing data. Of course, both the input and output drives can be the same.

Again, maximum length of the devicename is 16 characters.

Program Drive:

This is the drive where FirstBase looks for programs such as CREATE, QUERY, and UPDATE. Unlike the others, it can be a physical drive or a disk name.

The maximum length of the devicename is 16 characters.

Multiple Wildcard:

This is a single character denoting the multiple wildcard symbol; the default is an asterisk. See the *Database Conventions* section.

Single Wildcard:

This is a single character denoting the single wildcard symbol; the default is a question mark. See the *Database Conventions* section.

Location of configuration file

No matter what you specify for the *program drive*, FirstBase will always look on the disk labeled FIRSTBASE (eg. DSK.FIRSTBASE.) for the configuration file; this cannot be changed. Therefore, we strongly recommend that the program disk be always labeled "FIRSTBASE" (in case you make a copy) or, in the case of hard drive usage, the subdirectory containing the program be called "FIRSTBASE".

You can change the program drive to be something other than FIRSTBASE; this would result in the programs residing one disk, and the configuration file still on the disk called FIRSTBASE. That seems illogical; the configuration file and program files should be in the same spot.

Changing the configuration file

Now that you're armed with the knowledge you need, go ahead and change the configuration file to suit your system. Here's how to do it:

1. Load the BROWSE program. See the section in Chapter 1 titled *How to Start* if you don't know how.

2. Press 3 on the main menu for *Change System Configuration*. The program displays the first devicename, "Printer", along with the default setting of PIO. To leave it alone, press <enter>. To change it, type in your printer devicename, and press <enter> when you're done.

3. The program will continue to display the settings with their default values. Change whatever is necessary.

4. After the last setting, *Single Wildcard*, you're asked if you'd like to save your settings to disk. Press Y for Yes or N for No. If Yes, make sure your program diskette is in the proper drive and hit any key.

Section 2 - Specifications

In many ways, FirstBase surpasses all other 99/4A databases in regards to field and record size. Here is a complete list of FirstBase specifications:

Maximum field size.....	720 bytes
Maximum record size.....	3000 bytes
Maximum number fields/record.....	75
Maximum number records/file.....	32,767
Maximum file size.....	100 megabytes
Maximum indexed fields.....	75
Maximum keys for sorting.....	75
Maximum index size.....	246 bytes

Indexing:

Specifying a field as indexed (When you CREATE a database) allows you to include it in searches and sorts. If a field is not indexed, it cannot be searched or sorted.

When you index a field in the CREATE program, you need to specify how many bytes (or characters) to index. If, for example, you specify 20 bytes for a certain field, each time you enter and save a record, the program takes the first 20 bytes of that field and saves it in the index. Then, whenever you perform a QUERY or a single-field search, the program looks at only those first 20 bytes.

Index size:

The total size of an index entry is 246 bytes. This means that if you add up all the bytes you index in a single record, the total cannot exceed 246. For example, you cannot index two fields, each having an index size of 200 bytes; the total would be 400 bytes (200 x 2), which is greater than 246. On the other hand, you could have 75 fields in a record, and index 3 bytes of each field. That comes out to 225 (75 x 3), which is less than 246. So, in this way, you can index all of the fields in a record, as long as the <u>total number of bytes</u> indexed doesn't exceed 246.

Sorting:

The same limitation for indexing applies to sorting. You can only sort records using indexed fields as keys. Therefore, if you have 75 fields indexed, you can sort on 75 keys!

Section 3 - Database Conventions

This section will detail the major differences between FirstBase and traditional database systems. This knowledge is vital to using the program on even the most basic level. Everyone should read this section, no matter if you're a beginner or a seasoned pro.

Filenames

FirstBase requires you to type in filenames from time to time. The program requests filenames when creating and opening databases, and creating and printing reports. The restrictions on filenames are:

- 1-8 alphanumeric characters.
- No periods, spaces, or quotation marks.
- Filenames may be in UPPER CASE ONLY.
- In other words, follow the standard restrictions for filenames, except that they may be no longer than eight characters.

Field Attributes

A field can take on several attributes which describes its behavior while entering data and performing queries, updates, and sorts: field type (text or numeric), field length, index status, index length, and fixed status.

Field types - TEXT

A *text field* is one in which you enter any kind of information you like. There are absolutely no restrictions on a text field.

Field types - NUMERIC

A *numeric field* is one which can only contain numbers (with or without a decimal point). You cannot type dollar signs or commas...only numbers and the decimal point.

Field length:

The field length specifies the size, or how much information a field can hold; it can hold less (the remaining space is filled with blanks), but it can never hold more than its maximum length.

The size of a field can be between 1 and 720 bytes.

Indexed fields:

Indexing a field makes it available for searching. The index status of a field can be *YES* or *NO*. *YES* means a field is indexed, and *NO* means it is not. When FirstBase displays a record on-screen, an arrow symbol (>) before a fieldname indicates the particular field is indexed.

Index length:

If you define a field as indexed, you must also stipulate the number of bytes to index. Only those bytes indexed will be visible in any searching you do. By indexing the entire field, you can search for data anywhere. By indexing 10 bytes, you can search for data only in those first 10 bytes. Why would you want to index only a portion of a field? Remember that you can only index a maximum of 246 bytes per record. If you have many large fields, you may not be able to index each one in their entirety. By decreasing the index size of each field, you can index more fields at once. Another benefit is that a smaller index takes up less disk space.

The index length must be a number between 1 and 246.

Fixed fields:

A fixed field is simply a one that holds default data when entering new records. For example, in a database which keeps track of computer magazine articles, all of the entries for one issue of a magazine would have the same magazine name and issue number. If you define those two fields as fixed, then you only have to enter the information once; every new record after the first will contain the same data instead of being erased. Of course, you can change the contents of the fields at any time; the program just remembers the previous field's contents when you advance to a new record.

The fixed status of a field can be either *YES* or *NO*.

Textual versus Numerical Data

Most full-featured databases allow you the option of having true numerical fields, on which you can perform a variety of math functions. You can usually specify a special format, such as "money", where the program automatically inserts a dollar sign and decimal point.

Because of memory limitations, this is not possible for FirstBase, but we believe our solution is satisfactory. FirstBase employs a simple form of the numeric field; it can contain only numbers and the decimal point. It does no validation (the program will consider 33.45.343 a valid number). In effect, the field is a special type of text field - the only text allowable is any combination of the numbers 0-9 and the decimal point.
The drawback is that the program does no special formatting (like automatically inserting decimal points, commas, or dollar signs). Even so, FirstBase does allow you to perform calculations on numeric fields from within the UPDATE program, and the QUERY program allows you to use numerical comparisons like > (greater than).

Mini-Wordprocessor

Another way that FirstBase differs from conventional databases is that since it's mainly text-oriented, we've incorporated many wordprocessing features for entering data. A field can span many screen lines, depending on its size. Most other database do not have this capability, and are limited to a single screen line of information. As a result, FirstBase contains a full-screen editor for entering your data, so you can use the arrow keys to move the cursor, as well as insert/delete character and line, erase line, clear field, home cursor, and word tab.

What follows is a chart listing all of the special features of the FirstBase editor and their corresponding keypresses.

```
right arrow..... <fctn-d>        delete character... <fctn-1>
left arrow...... <fctn-s>        delete line........ <fctn-3>
up arrow........ <fctn-e>        clear from cursor
down arrow...... <fctn-x>          to end of line... <fctn-4>
word tab........ <fctn-7>        home cursor........ <fctn-5>
insert character... <fctn-2>     home cursor and
insert blank line.. <fctn-8>       clear field...... <fctn-6>

                  end data entry..... <fctn-9>
```

Word wrap is NOT a feature of the FirstBase editor due to a lack of memory. The additional features above, however, should make typing much more enjoyable.

The program uses this mini-wordprocessor is two distinct areas:

- Record entry
- All other prompts

The first instance is self-explanatory; that's when you're entering new data into a record or editing an existing record.

The second instance needs a bit of explaining. Whenever you're asked to type something in, like a database name, or a printer device, or a QUERY command, the same editor is in use. Sometimes, as with the QUERY command, the input field is 254 long. All 254 characters cannot fit on one screen line, so the input field wraps around the screen several times. Therefore, if you enter a short command that doesn't extend to the end of the input field, hitting <enter> will only advance you to the next screen line, and the program will wait for you. If this happens, you can continue by pressing <fctn-9>. This keypress ends data entry.

Wildcard Symbols

Wildcards make your life easier by saving keystrokes when typing fieldnames or search strings. There are two kinds of wildcard symbols: single and multiple. Briefly stated, the single wildcard replaces a single character, while the multiple wildcard replaces many characters.

Multiple Wildcard:

> The multiple wildcard can replace any number of characters in a string. For example, assume you are searching the database for the string *FirstBase*, but you're worried that you might have spelled it wrong. Instead of *FirstBase* use *F*Base* and the program will find all occurrences of *FirstBase*, along with any other strings that have first letter an *F* and the last four letters being *Base* (like *fistbase* or *FooBase*). The multiple wildcard symbol can be any single character, but we recommend using an unused symbol such as '*'.

Single Wildcard:

> This kind of wildcard character replaces only a single letter in a search string (see *Multiple Wildcard* for more information). For example, if you were looking for the string *receive* but couldn't remember how to spell it ("i" before "e" except after "c"), type in *rec??ve*. The program will find all the strings that fit the pattern of *rec*, any two letters, and *ve* (like *receive* or *receeve*). If you had typed in *rec*ve*, it will find all strings that fit the pattern of *rec*, any number

of letters, and ve. Both methods will find the word *receive*, but using the single wildcard is more specific. Again, the single wildcard symbol can be any single character, but it must be different from the multiple wildcard symbol! An often-used symbol is '?'.

Section 4 - Search Formats

One of the most common things to do with a database is search for a piece of information. You can do that several ways with FirstBase: a single field search or a multiple field search using a QUERY command. Depending on which method you use, the format of the *search string* (the information you are looking for) differs slightly.

This section first deals with the issues of *case* and *delimiters*, and then discusses the search string formats in both the BROWSE and QUERY programs.

Case

FirstBase ignores uppercase and lowercase when performing any kind of searching. This means that the strings *elephant*, *ELEPHANT*, and *ElEpHanT* are identical.

Delimiters

FirstBase normally thinks a string (whether it be a search string or a command) is a word with no spaces. Once it sees a space, it thinks it has reached the end of the string. In computerese, the space would be called a *delimiter*.

To force FirstBase to ignore spaces as a delimiter, enclose the string with quotations marks. When would you want to ignore spaces? Maybe you want to find the address *12345 North Pole*. That string has two spaces in it. Therefore, typing in *"12345 North Pole"* instead, with the quotes, forces the program to consider the address as one string stead of three.

Partial strings

Normally, any string you type in for a search command is considered an exact string. That is, it will look for an exact match; the string cannot be part of something else. To look for partial strings (ones that are part of other strings), you need to use the multiple wildcard character.

Multiple wildcard character for searching

Using the multiple wildcard character is necessary when you are looking for just part of a word, or a word that's embedded in another. If you typed in *SORTING* as a search string, the program

would look for a field that contains only that string and nothing else.

If you want to find an instance where the field begins with the word, use *SORTING** (assuming that you've defined the multiple wildcard character to be an asterisk). The asterisk means "any number of characters." So in effect, you are asking FirstBase to look for a field that starts with the word *SORTING* and followed by anything.

On the other hand, if you wanted to find an instance where the field ends with the word, use **SORTING*. As you might expect, this tells the program to look for a field that begins with anything, but ends with the string SORTING.

The last case is where you want to find an instance where the word appears anywhere in the field, in which case you would type **SORTING**. This means find the word *SORTING* with any number of characters preceding and following it.

Search formats in the BROWSE program

You can execute single-field searches in the BROWSE program to locate a record easily.

You can only search one field at a time; once you indicate the field you want searched and the string(s) for which to search, the program displays all records meeting your requirements.

The search string format is greatly simplified as compared to the QUERY program. If you want to search for several strings at once, simply type each string, separated by spaces. To include spaces within a string, enclose the string with quotation marks.

Example 1:

If you were to use **LANGUAGES** **SORTING** for a search string, the program would think that you're looking for two separate strings, *LANGUAGES* and *SORTING*. Remember, the space is a delimiter. It would try to find all records containing the words LANGUAGES and SORTING, but not necessarily in that order. Because of the wildcards, the two strings can appear anywhere (even as part of other words).

What happens if you try to search for *LANGUAGES SORTING* ? Without the wildcards, FirstBase would never find a matching record. As detailed in the preceding paragraphs, if a search string has no wildcards, the program looks for an exact match. Since there is a space between the two words,

FirstBase will look for two exact matches. It will try to find a field that matches perfectly first with the word *LANGUAGES*, and then with the word *SORTING*. Now how can a field match two different words perfectly?? It can't!

Example 2:

If, on the other hand, you need to look for a string that contains spaces, surround the string with quotation marks. To look for the single string AN AUTOBIOGRAPHY, you would type:

"AN AUTOBIOGRAPHY"

This is considered one long string.

Please note that in the BROWSE program, if you type in several strings, all of the strings must be found before FirstBase will retrieve a record. This differs significantly with the QUERY program, which is covered in the next section.

Here are a few more examples of searching with the BROWSE program.

HARDWARE* *MUSIC

FirstBase will find the records that contain the two strings MUSIC and HARDWARE. Each string can appear anywhere in the field, and they may be embedded within other words (due to the wildcard characters). *MUSICALLY TALENTED HARDWARE* contains the words *HARDWARE* and *MUSIC*, so it would be considered a match.

"HARDWARE MUSIC"

FirstBase treats this as one string, rather than two separate words, because of the quotes. It will not find *MUSIC HARDWARE* or *MUSIC* or *MUSICALLY*, only the string *HARDWARE MUSIC*. No wildcards are used, so it must match perfectly; the field may not contain anything else.

FORTRAN "*RISC PROCESSORS*"*

You can combine the two formats; the program will first look for the occurrence of the string FORTRAN (which must appear at the BEGINNING of the field), then the string RISC PROCESSORS. The latter contains a space, and can appear anywhere in the field.

Search formats in the QUERY program

You can perform much more sophisticated searches in the QUERY program as compared to BROWSE, since you can search the contents of more than one field at a time.

The rules for search strings are almost the same as when using the BROWSE program, except for the use of multiple strings. You cannot search for multiple strings by just listing them out, separating them with spaces. As explained in *Chapter 5 - The QUERY program*, spaces not only delimit search strings, but delimit commands as well. You can only list one search string per comparison. The program interprets anything following the search string as a new comparison. To look for more than one string, you must use the OR operator. The following examples of queries should make things clear.

Example 1: (valid)

*NAME = *SMITH* AND ADDRESS = "12345 SUNSET STRIP"*

This is a valid query using two kinds of search strings. The first is a single word, no spaces, with wildcards. FirstBase will first find any record where the word *SMITH* occurs anywhere in the NAME field, and then find those records where the ADDRESS field contains the string *12345 SUNSET STRIP*, spaces included. The address must match perfectly, since there are no wildcards.

Example 2: (invalid)

CITY = "NEW YORK" PITTSBURGH AND AGE < 45

This is a totally invalid query. *CITY = "NEW YORK"* is just fine, but *PITTSBURGH* is meant to be yet another search string. The program however, after seeing the first search string, interprets *PITTSBURGH* to be a command (which it's not), and aborts with an error.

Example 3: (valid)

CITY = "NEW YORK" OR CITY = PITTSBURGH AND AGE < 45

This is the correct way to do example 2.

Fieldnames in QUERY and UPDATE

As you will see in *CHAPTER 3 - The CREATE Program*, fieldnames can be up to 20 characters long, and they may contain spaces. This can cause problems when you have to specify a fieldname in a query or

be up to 20 characters long, and they may contain spaces. This can cause problems when you have to specify a fieldname in a query or update command, since FirstBase recognizes spaces as a delimiter.

For example, let's say you want to search for some specific data in the field called *FIRST NAME*. If you type in *FIRST NAME*, the program will think that *FIRST* is the fieldname, and *NAME* is the next command, which would generate an error. To avoid this, enclose the fieldname within quotes. The program would consider *"FIRST NAME"* a single fieldname.

Fieldnames in BROWSE

Because BROWSE prompts you separately for the fieldname when doing searches, embedded spaces do not cause a problem. In fact, the program gets confused if you put quotes around a fieldname! This may seem inconsistent at first, but sacrifices had to be made due to memory constraints.

Partial fieldnames

FirstBase allows the use of partial strings for fieldnames. If you want to search on the *ADDRESS* field, typing in *AD** will identify the same field. A problem arises, however, when two or more fields start with the same characters *AD*. As a rule, FirstBase will use the very first field in the record that begins with AD. As an example, say you have a record that looks like this:

```
NAME: Santa Claus
ADVERT. COSTS: 15000
ADDRESS:  12345 North Pole
```

If you ask for the field AD*, the program would think you are looking for the *ADVERT. COSTS* field, not the *ADDRESS* field. *ADVERT. COSTS* occurs first in the record.

You gain an advantage to using partial fieldnames when the fieldname has more than one word. Instead of enclosing it in quotes (in the QUERY command), just remove the quotes and use the first word of the field. For example, *"FIRST NAME"* and *FIRST** can be used interchangeably. Just make sure that there are no other fieldnames that start with the word *FIRST*. This strategy also works in the BROWSE program.

To avoid these problems, be careful to give your fields unique names in the CREATE program. Avoid things like SALES TAX, SALES PERSON, SALES TOTAL, etc. Instead use unique names such as TAX, SLS-PERSON, SALESTOTAL, etc. Be creative!

Section 5 - Summary

- The file called "PTR_", which is on the program disk, is called the *system configuration file*, and contains information about how your system is set up.

- The configuration file contains information about the printer, input data drive, output data drive, program drive, multiple wildcard symbol, and single wildcard symbol.

- FirstBase will <u>always</u> look for the configuration file on the disk labled FIRSTBASE (eg. DSK.FIRSTBASE.PTR_). This can never change.

- The program disk should always be labeled FIRSTBASE, so the programs and the configuration file are in the same location.

- Indexing a field allows its inclusion in searching and sorting.

- The index size specifies how many bytes (or characters) to index; only those bytes are visible in searching and sorting.

- The total index size for one record cannot exceed 246 bytes. Therefore, if space is tight, you can search and sort on every field by indexing only part of each.

- Filenames can be up to 8 alphanumeric characters, must contain no spaces, periods, or quotes, and must be in upper case.

- A *text* field can contain any kind of data.

- A *numeric* field can only contain decimal numbers (numbers and the decimal point). No special formatting is done.

- A *fixed* field remembers what the previous record contained when you add new records. In this way you can avoid retyping the same data that appears in every record.

- Sometimes, when typing in a filename or a long command, the program doesn't do anything after you press <enter>. Press <fctn-9> to end data entry. This means the input field wraps around the screen more than once, and the cursor hasn't reached the end of it.

- The *multiple wildcard* is used in search strings and fieldnames; it takes the place of any number of characters. Using it, you may type in only part of a fieldname, or look

fieldnames; it takes the place of any number of characters. Using it, you may type in only part of a fieldname, or look for embedded strings with a search command.

- The *single wildcard* takes the place of a single character in search strings or fieldnames. It is much more restrictive than the multiple wildcard.

- FirstBase ignores case (upper/lower) throughout the program.

- The blank space is considered a delimiter, which separates search strings and commands. Enclose a string with quotes if you want the space included in the search.

- Searching fields in the BROWSE program is simple: simply type a fieldname when asked (no quotes allowed here), and then list out all of the search strings to be found, separated by spaces. Use quotes to include spaces in the strings.

- Searching fields in the QUERY program is comprehensive; you can't simply list out the strings as with the BROWSE program. Refer to the section on the QUERY program for more information.

CHAPTER 3 - The CREATE program

This chapter documents each and every feature the CREATE program has to offer. This program lets you create a new database by defining its structure. FirstBase stores all of the information about how the database is structured in a *database definition* file. This file is stored on the data disk, and is easily recognizable in a disk catalog -- the filename is always in the form of *XXXXXXXX_D*. The XXXXXXXX is the database name, and the "D" stands for "definition."

The definition for a database contains the names and sizes of all the fields, plus their various attributes: index status and length, field type, and fixed status. You will learn all about these attributes in the following pages.

There are basically two functions the CREATE program serves: creating a new database and changing an existing definition for use in a new database. You currently cannot modify the structure of a database once it has been created; however, you can load in the definition of an existing database and use it to create another database.

The CREATE program can be accessed from any main menu in FirstBase, and the files it loads are CREATE1, CREATE2, and CREATE3.

Introduction.....33

Create new database.....34
- Fieldname.....35
- Field length.....35
- Index this field.....36
- No. of bytes to index...36
- Fixed field.....36
- Text or numeric.....37

CREATE's command prompt.....37
- List.....37
- Change.....38
- Add.....38
- Save.....38
- Delete.....38
- Quit.....38

Change database in memory.....39
Load definition.....39
Change drives.....39
Summary.....40

Section 1 - Introduction to the CREATE program

Purpose: If you want to create and maintain your own databases (as opposed to strictly using the enclosed sample files), you must use the CREATE program first. After asking you questions about what kinds of information the database is to contain, the program saves the data in a *dictionary file*. It also creates an empty index and database file.

When creating a database, you provide the following information:

- database name
- fieldnames
- index status
- fixed status
- field length
- index length

The menu for the CREATE program looks like this:

```
==================FirstBase:Create=============

                1 = Create New Database
                2 = Change Database In Memory
                3 = Load Definition
                4 = Change drives
                5 = Main Program
                6 = Extras
                7 = Query
                8 = Update
                9 = Quit FirstBase

      DSK2.
      ============================================
      Please Make Your Selection:_
```

Briefly, the menu choices are:

Create New Database:

Allows you to enter the definition of a new database. This clears any previous definitions from memory so you may start with a clean slate.

Change Database in Memory:

After creating a database and returning to the main menu, you can go back and make additional changes or fix mistakes with this option; the current definition is NOT cleared from memory. You can also make carbon copies of database definitions or make minor changes and save them as

copies of database definitions or make minor changes and save them as separate definitions.

Load Definition:

With this option, you can load an existing definition off of disk, examine it, modify it, and save it back out as a new definition.

For example, suppose you have been using a database to keep track of user-group members, and you run out of space. You decide to create a second, identically structured database to take care of additional members. Instead of re-creating the definition, you can merely load in the original definition and save it back out to a different disk. With just a few keystrokes, you can start entering new members in no time.

Change Drives:

The CREATE program always saves definitions to the disk in the input drive (see the **Installation** section in Chapter 2 for more information). You can temporarily change the input drive with this menu choice. Changes here are not permanently saved.

Extras, Query, Update:

These menu options load in the corresponding programs from disk.

Quit FirstBase:

This option exits FirstBase and brings you to the main TI title screen.

What follows is a detailed explanation of how to use each menu option.

Section 2 - Create New Database
(menu selection 1)

This is where you go to create a brand-new database. It clears any definition already in memory so you may start with a clean slate. With this option, you are asked to answer six questions for each field you want in your database.

The screen for entering information is reproduced below:

```
Create New Database==================================

Defining field 1

Field name: [20 ASCII characters maximum]

Field length: [1-720,0,<enter>]

Index this field? [YNyn,<enter>]

Number of bytes to index: [1-246]

Fixed field? [YNyn,<enter>]

Text or Numeric? [TtNn,<enter>]

=====================================================
```

Items in brackets do not appear on the screen; they indicate all the possible values you may enter. A description of each question follows.

Fieldname:

Field names can contain any of the following characters. maximum length is 20 characters.

- Alphabetic characters. Upper/lower case are treated the same.
- Numeric characters, 0-9.
- All printable symbols, eg. !@#$% etc. Blank spaces.

You cannot enter a blank line for a fieldname!

Field length:

A number from 1 to 720. This specifies the length of the field in bytes.

Typing a 0 or just <enter> at this prompt ends the data entry for this definition.

Index this field?

Valid entries are Y and N.

Y will cause the contents of the field to be saved in the index file. This field can then be searched and sorted. The number of characters saved is determined by the index length, which you define next.

Pressing <enter> without answering the question defaults to 'N'o.

Number of bytes to index:

Valid entry is a number between 1 and the length of the field.

Also referred to as the "index length." When a record is saved to disk, a fixed number of bytes from each indexed field are saved in the index file. Here is where you determine how many bytes to save for each field. If a field is 20 bytes long, you can index up to 20 bytes, which is the entire field. If you just told the computer to index a field, you MUST give it an index length!

If the field is not indexed, the program skips this question.

> **NOTE**: The total number of bytes you can index in a record is 246. That means if you add up all the index lengths of each field, the total cannot exceed 246. Please keep this in mind when designing your database. The program warns you if you reach the maximum.

Fixed field?

Valid entries are Y and N.

A fixed field is one in which old data is not cleared when you add a new record. This avoids repetitive data entry.

Typing <enter> without answering the prompt automatically enters a 'N'o response.

Text or Numeric?

Valid entries are T (for text) and N (for numeric).

A text field will allow you to enter any kind of information.

A numeric field only allows you to enter numbers (with or without a decimal point). See **Chapter 2 - The Rules of FirstBase** for more

decimal point). See **Chapter 2 - The Rules of FirstBase** for more information about field types.

Hitting <enter> without typing anything automatically defines the field as 'T'ext.

After entering the information for the first field, all of the prompts clear and the program waits for another field name. Continue entering information for all of your fields; when finished, press <fctn-9> at the *Fieldname:* prompt. The next thing you see is a command prompt.

Section 3 - The CREATE program's COMMAND PROMPT

You will see the following prompt often in the CREATE program, whenever you have finished a task, like creating a database:

List, Change, Add, Save, Delete, Quit?

Type the first letter of the command you want to use. What follows is a rundown of each command's function.

List:

Displays the database definition in a tabular form so you may review your entries. A sample display is given below.

Number of fields: 4

FIELDNAME	LEN	I	F	T	INDEX
Disk Number	3	Y	N	N	3
Disk Name	42	Y	N	T	42
Author	30	N	N	T	n/a
Version	13	N	N	T	n/a

The top of the screen indicates how many fields the database contains.

Under the FIELDNAME heading, each field's name is listed. The LEN column indicates the field length. The I column contains the index status (Y for YES and N for NO). The F column indicates the fixed status. The T column is the field type column; a T stands for TEXT, and an N stands for NUMERIC. Finally, the INDEX column shows the index length for this field. Only the number of bytes shown (starting from the beginning of the field) can ever be searched or used in sorting. The "n/a" appears if the field has not been indexed.

If the definition is longer than one screen, you are prompted to hit any key before viewing the next screen of information.

After the display, you are brought back to the command prompt.

Change:

Here you can change the information for any field you've defined. Type in the name of the field you wish to change when asked, and the rest of the info is displayed. You can go through and re-type anything you like.

When finished, press <fctn-9> at the *Fieldname:* prompt.

Add:

This command allows you to add a field onto the end of the database definition currently in memory. This action is identical to defining a new database; you type in a fieldname, along with all its attributes.

Press <fctn-9> to exit at the fieldname prompt to finish.

Save:

This saves the definition out to disk and initializes the database files. Type in a valid database filename and press <enter>. See **Chapter 2 - The Rules of FirstBase** for information on filenames. Make sure the data disk is in the input data drive!
If the database already exists, FirstBase asks you if you're sure you want to overwrite it. All of your existing data will be lost!

Delete:

You can delete any field in the definition with this command. After typing in a fieldname, press 'Y' to delete, or any other key to abort.

Quit:

This returns you to the main CREATE menu. If you hit this by accident, don't worry...you can return to this prompt by selecting 2, *Change Database in Memory*. Doing so will not clear your work from memory.

Section 4 - Change Database in Memory
(menu selection 2)

Unlike menu option 1, *Create New Database*, this option does NOT clear any previous definition from memory. It allows you to manipulate the one already in memory.

Once you choose this option, you are presented with the familiar CREATE command prompt explained in the previous section. To change a field, press C, and you see the field entry screen where you can enter a fieldname to change. If you press <fctn-9> for the fieldname, you get back the command prompt. You can make changes, list the definition, and/or save it under a new definition.

> **NOTE**: You can save a new definition over an old one by using the same filename, but the program gives you one last chance to back out. If you write over the definition of an existing database, all of the old information will be LOST, including any records the database may contain!

Section 5 - Load Definition
(menu selection 3)

If you'd like to look at an existing definition or make a copy of one, you can use this option to load in an existing definition from disk. You are asked for a database name, and it looks for the definition in the input data drive. After loading, you are brought back to the main menu, where you can choose menu option 2, *Change Database*, to view the definition, make changes, and save it back out as a new database.

Section 6 - Change Drives
(menu selection 4)

If you want to temporarily change the input data drive where the definitions are saved, you can use this menu option. Please note that the drive must be in the form *DSKn.* or *WDSn.* where 'n' is the drive number. See the **Installation** section of Chapter 2 for more information about the data drive name.

> **NOTE: Changing drives here is a temporary action; the new drive is not saved in the configuration file.**

Menu options 5, 6, 7, and 8 load the indicated program.

Menu option 9, "Quit FirstBase", quits the program.

Section 7 - Summary

- Use the CREATE program to create a new database. This must be done before you can start adding information.

- When creating a database, you provide the database name, fieldnames, field length, index status, index length, and fixed status.

- Choosing *Create New Database* from the main menu clears out any previous definitions so you can start with a clean slate.

- A fieldname can contain any alphanumeric characters, spaces, and any printable symbols, such as !@#$%. No blank lines allowed.

- The field length can be a number between 1 and 720.

- If a field is indexed, the contents of the field will be saved to a separate index file; only this index file is used when you search and sort.

- The index length specifies how much of a field to index.

- The maximum size of an index for one record is 246 bytes.

- Selecting *Change* from the command prompt lets you change all the information about a field you've already defined.

- Selecting *Add* from the command prompt lets you continue defining additional fields; the fields you type in will be added onto the end of the definition in memory.

- Selecting *Delete* from the command prompt lets you remove a field from the definition in memory.

- You can load in a definition that already exists by selecting *4 - Load Definition* from the main menu. Once loaded, you can make changes (if any), and save it back out as a new definition.

- You cannot modify the definition of a database once it's created in this version of FirstBase. Modifying a defintion and saving it back out under the same name will erase any records in the database!

- Selecting *5 - Change Drives* from the main menu lets you change the data drive and database name, but it is temporary. Once you move to another program within FirstBase, the changes you make will be forgotten.

- To change the data drives and/or database name permanently, use the BROWSE program.

CHAPTER 4 - The BROWSE program

Once you create a database with the CREATE program, you can start adding records. Perhaps after you've entered all your data, you need to make some changes, or you decide to delete some records. Maybe you'd just like to browse through your information, one record at a time. This is where you do all that - in the BROWSE program.

In may ways, BROWSE acts as the heart of FirstBase, since most of your time will be spent here. Nothing is more time-consuming than typing information into a database, so we went to great pains to make this part of FirstBase as easy to use as possible.

Most of the work you will do in BROWSE happens at the *browse prompt:* a long command prompt that sits at the bottom of the screen. With a single kepress you can move forwards and backwards through your database, making changes as you go, printing records, deleting/undeleting, etc. If you need to locate a specific record quickly, you can use the single-field search option to find exactly what you need.

You can also customize FirstBase by changing the *system configuration file*. The changes you make in the BROWSE program can be temporary or permanent.

The BROWSE program can be accessed through any menu in FirstBase, and loads the files MAIN1, MAIN2, and MAIN3.

Introduction.....43

Browse records.....44

Displaying records.....44
The browse prompt.....46
Current.....46
Forward.....46
Back.....46
Top.....47
Last.....47
Individual.....47
Add.....47
Edit.....48
Save.....48
Delete.....48
Undelete.....48
Key/field search...48
Print.....48
Quit.....49

Move to new database...50

Change configuration...50

Summary.....51

Section 1 - Introduction to the BROWSE program

Purpose: Most of your mundane tasks, like adding new records, deleting records, or editing existing records, take place in the BROWSE program, which acts as the heart of FirstBase. By and large, most of your time will be spent in this area of the program.

Basically, there are five major tasks that you can accomplish in BROWSE:

- Add new records
- Save a record
- Edit an existing record
- Delete a record
- Execute single-field searches

There are also various "housekeeping" chores that come in handy...printing an individual record, sequentially browsing through your database, and changing the system configuration. We will discuss all of these functions in the following pages, so snuggle up in front of a crackling fire and read on.

Starting Out

The FirstBase MAIN menu looks like this:

```
========== FirstBase 8/88 ==========

              1 = Browse Records
              2 = Move to New Database
              3 = Change System Configuration
              4 = Extras
              5 = Create
              6 = Query
              7 = Update
              8 = Quit FirstBase
DSK2.
=======================================
Please Make Your Selection:_
```

Pay special attention to the line that reads "DSK2." above the double lines. This is the *file indicator;* it displays the current input drive and database file. If you're starting a new session, there is NO database currently in use, so all you see is the input drive.

Section 2 - Browse Records
(menu selection 1)

By browsing you can perform operations on individual records: add, edit, delete, undelete, print, etc. It also facilitates single field searches.

Selecting "Browse Records" from the main menu opens a database file. Exactly which file it opens depends on the following:

Cold Boot:

> If no other database has been used since initially loading FirstBase, the *file indicator* does not display a database name, only the input drive. Therefore, the program asks for the *database name*, the same name you used in the CREATE program. The program then attempts to open the database files. A message indicates if it was successful. If so, you are shown the first record in the database.

Warm Boot:

> FirstBase remembers the last database in use. The database name appears in the *file indicator* at the bottom of the screen. If such a database was in use, the program does not ask for a database name, but attempts to open the current file.

Error:

> If opening the database fails, you are notified on-screen. Hit any key to return to the main menu; you may change the database name (See **Move to New Database**) or insert the correct disk and try again.

Displaying records

Once the database is open, you will see the first record in the file. If the database is empty, the program immediately goes into *Add* mode, where you can start entering records.

A typical record might appear like this:

```
================================================================
<1> Display                                      Page 1 of 1
>Article Title:
SQL Database Management Systems

>Magazine:
BYTE

>Issue:
Jan 88

>Page:
111

>Abstract:
A look at Informix-SQL, Ingres, Oracle, SQLBase, XBDII, and XQL.


Next page,Previous page,Quit:_
================================================================
```

The first line indicates the record number, the "mode" of operation (Display), and the page (or screen) number. This example indicates that the record is one page in size.

One of two things may happen after displaying the first page of a record:

- If the record fits on one screen, you get the browse prompt.
- If the record is larger than one screen, you see the *screen prompt*.

The *screen prompt* looks like this:

Next page, Previous page, Quit?

This prompt lets you flip to different screens in the record. Hitting 'n' displays the next screen of information; pressing 'p' displays the previous screen. Pressing 'q' brings you to the browse prompt, which is dealt with below.

Displaying deleted records

If the current record happens to be deleted (explained later), a 'd' follows the record number (e.g. "<1d> Display")

Displaying index status

An arrow (>) precedes each field that is indexed.

The BROWSE prompt

The Browse prompt looks like this:

Cur,Fd,Bk,T,L,I,Add,Ed,Sv,D,U,Key,P,Q?

Each word or letter is an abbreviation for a command. In order, the abbreviations stand for: **Current** record, **Forward**, **Backward**, **Top**, **Last**, **Individual**, **Add** record, **Edit**, **Save** record, **Delete** record, **Undelete** record, **Key**/field search, **Print**, **Quit**.

Current record:

Displays the current record in memory. The top line of the screen displays the record number in brackets, the current "mode" of operation (in this case 'Display'), and the page screen number. *"Page 2 of 3"* means that you are viewing page 2, and the record has a total of 3 pages. FirstBase will display as much of a record as it can on one screen; if there isn't enough room, it continues on the next screen. If so, you will see the *page prompt* explained earlier.

A 'd' follows the record number if the record is deleted.

Forward:

Loads in and displays the next record in the database. If there are no more records, you get this message:

End of data: hit any key...

Please note that records normally appear sequentially, in the order that they were stored. If, however, the index were to be sorted (using the SORT program), records would appear in sorted order. The index determines the sequence of records.

If the next record is deleted (see Delete below), it will be skipped.

Back:

Loads in and displays the previous record stored on disk. See "Forward" for more information. This command skips deleted records.

Top:

Loads in and displays the first record in the database. It skips deleted records.

Last:

Loads and displays the last record in the database. It skips deleted records.

Individual:

Allows you to load a record using the **record number**. Each record has a number, which you see in the upper left hand corner of the screen. Upon selecting 'I', you are asked for a record number between 1 and 32767.

Unlike all of the other commands, you CAN view deleted records as well.

Error messages:

"Out of range: re-enter" means you entered a number not between 1 and 32767.

"Record x not on file: hit any key..." means the record does not exist. For instance, if the last record in the database is #27, you cannot load record #30 because it doesn't exist.

Add record:

Lets you type in a new record. The program displays the new record number at the top of the screen, along with the mode ("Add") and the page number. Each field appears, along with any default items (only with fixed fields). The cursor appears at the first field, where you may enter your data. See **Chapter 2 - The Rules of FirstBase** for a table of special keystrokes you may use while in the record editor.

Remember, you can enter anything in a text field, but only numbers in a numeric field.

In addition, you can use the arrow keys to move between fields, and pressing <fctn-9> anywhere will end data entry for the current screen.

After you've finished entry of the current screen, you see this:

Next page, Previous page,Save,Quit?

Press N or P to move to a different page (if there is one), or hit S to save your record to disk. Pressing Q ends data entry mode and brings you back to the browse prompt.

Edit:

Allows you to edit or change the record in memory. The mechanics of working this function are the same as with the **Add** command.

Save:

Saves the current record in memory to disk. If the record already exists in the database, FirstBase asks you to confirm the command.

Delete:

Marks the current record as DELETED. This does NOT remove a record from the disk! The program marks the index entry for this record as DELETED, so the program will ignore this record except for the **I** and **Key** commands. In effect, the record is rendered "invisible." Even the QUERY, UPDATE, and SORT programs normally ignore records marked as deleted. If you want to "unmark" such a record, use the UNdelete command, explained next. Records are permanently removed when you perform a query to a new file or when you sort a database (all explained in due time).

Undelete:

Undoes the DELETE command. First load in the deleted record by using the **I** or **Key** command (explained next), then press **U** to undelete it.

***Key**/field search:*

K for search? Sorry, ran out of letters! You can search on any single indexed field with this command. When asked for a fieldname, you may do three things:

SEARCH: Type an indexed fieldname (wildcards are allowed), then <enter>. When the program asks for a search string, enter the string(s) for which you want to search. This option skips deleted records.

Do NOT enclose fieldnames with quotes! This is the exception to the rule. Since this is not a query or update command, the space is automatically ignored as a delimiter.

DELETE: Typing the word *DELETED* in place of a fieldname will find only deleted records.

FirstBase does not ask you for a search string; you cannot combine search strings with the *DELETED* option.

ABORT: Entering nothing for the fieldname aborts the search.

Wildcards can be used in the search strings as well. Read the **Search Formats** section in Chapter 2 for more information.

Each record meeting your search criteria is displayed. At the end of the record you will see the prompt *"<enter> for more;"* Press <enter> to continue the search or any other key to abort.

Please note that unlike the QUERY program, you can only perform simple field searches. If you type in more than one string, the program looks for records containing ALL of the information. You can't search on more than one field; for more versatile searching, use the QUERY program.

NOTE: Pressing <fctn-4> during a search will abort the process.

Print:

Prints the record in memory to the printer defined in the system configuration file. The record comes out in 40-column format, very similar to the screen display.

NOTE: Press and hold <fctn-4> to abort the printing.

Quit:

Closes all database and index files and returns you to the main menu. **ALWAYS USE THIS COMMAND TO EXIT THE PROGRAM! VALUABLE DATA MAY BE LOST IF THIS PROCEDURE IS NOT FOLLOWED.**

Section 3 - Move to New Database
(menu selection 2)

This menu selection allows you to change the input drive and database name.

Upon selecting this option from the main menu, you are asked for the *current drive/path;* type in the input data drive which contains a database or hit <enter> to leave it alone. See the **Database Conventions** section of chapter 2 for more info on data drives.

Next you need to type in a database name. Remember, the database name can be up to 8 characters long, and must not contain spaces, quotation marks, or periods. Just press <enter> if you want to leave it alone.

Once you enter a new database, selecting menu option 1, *Browse Database,* will open the new file, and all processing will occur on this database. FirstBase will forget all about the database file you were working on beforehand.

Section 4 - Change System Configuration
(menu selection 3)

This menu selection lets you change the default data drives, program drive, printer device, and wildcard characters. You can optionally save the configuration to disk under the filename PTR_.

The function is self-explanatory; each default setting is displayed, with the cursor resting on the first letter. Type in your changes, or just press <enter> to accept the current settings. After the last setting, you have the opportunity to save the configuration file to disk. **Make sure the PROGRAM disk in its proper drive if you choose to save the information for later sessions.**

All of the system configuration settings are defined in detail in the **Installation** section of Chapter 2.

EXTRAS, CREATE, QUERY, UPDATE

Menu selections 4 through 7 will load in the appropriate programs from the program drive. Make sure the program disk is in the proper drive before making one of these selections.

Section 5 - Summary

- If loading FirstBase at the beginning of a session (cold boot), the program prompts you for a database name to open.

- If you load the BROWSE program in the middle of a session, FirstBase remembers the database you were last working on and opens it automatically.

- In FirstBase terminology, a screen of information is often called a 'page'.

- When displaying records, FirstBase fits as much of a record on the screen as it can; if it continues to the next screen, the program prompts you to turn the page.

- When displaying a deleted record, a 'd' appears after the record number at the top of the screen.

- When displaying an indexed field, an arrow precedes the fieldname.

- If a record is deleted, it remains invisible from normal database operations like searching and browsing.

Tips on using the BROWSE prompt:

- To display the record in memory, use the **C** command.

- To browse through the database record by record, use the **F** and **B** commands.

- To view the first and last records, use **T** (for top) and **L** (for last).

- To load a specific record by number, use the **I** command. You must know the record number to use this command. Unlike the other commands, this will also load in deleted records.

- To add a new record to the database, use the **A** command.

- To edit the record in memory, use the **E** command. A record must be loaded before it can be edited.

- To save the current record, use the **S** command.

- To delete a record, use the **D** command. The record remains invisible until you undelete it, or it gets removed by the QUERY or SORT programs.

- To undelete a record, use the **U** command. The record must be in memory in order to undelete it. Use the **I** command or the **K** (key/search) command first.

- To get a quick print-out of the record in memory, use the **P** command.

Tips on the changing the system configuration:

- To make permanent changes to the system configuration (data drives, printer device, wildcards, etc) you must use the *Change System Configuration* option in the BROWSE program.

- If you don't save the changes to the config file to disk, FirstBase will revert to the original settings whenever it loads a different program.

CHAPTER 5 - The QUERY program

In chapters 3 and 4, you discovered all of the ins and outs to creating and maintaining a database. You were shown how to add records, edit them, delete them, and perform simple searches.

If that was all FirstBase could do, you would be no better off than using someone else's program. Only when you start to use the QUERY program do you realize the power of FirstBase.
Basically, a query allows you to find things. In this case, "things" are records. You tell FirstBase to find things by describing what the things look like: you describe what the records are to contain. More specifically, you need to specify which *fields* to look at, and exactly what each of those fields is to contain. You can look for more than one thing at once; you can list many fields and many pieces of information in a query command.

Once FirstBase has found exactly the records you desire, you have a choice of what to do with them: list them to the screen, send them to a new database file, glue them onto the end of an existing database file, or ship them over to the REPORT program for printing.

Introduction.....54

Query command format.....55
- Field.....55
- Equals symbol.....56
- Data.....56
- Logical connectors.....57
- Parentheses.....58

Query destinations.....59
- Screen.....59
- New database.....59
- Old database.....59
- Report program.....59

Number formats.....60

Examples.....61

Summary.....63

Section 1 - Introduction to QUERY

PURPOSE: The QUERY program is where you can start to witness the power of FirstBase. Quite simply, a query is a command that finds a group of records according to your specifications. You could think of it as similar to the 'find string' function in a word processor. The command specifies contents of fields (like find all records where the STATE field contains "MI"). Once it finds a group of records, they can be displayed, sent to a new database file, appended onto the end of an existing file, or sent to the REPORT program. This allows you, with one stroke, to split files apart or glue them back together.

IMPORTANT! You can only look at INDEXED fields in the query command.

NOTE: Records marked as *deleted* with the BROWSE program will be ignored in all searches within the QUERY program.

There are two basic steps in performing a query:

- Type in a query command
- Specify the destination

The first thing you must do to perform a query is type in a query command. Basically a query command is a simple expression describing the contents of a field. *SALES = 50* is a valid expression,which indicates that you're looking only for the records in which the *SALES* field contains the number 50. You can glue together several expressions by using the connectors **AND** and **OR**.

The query command can be a maximum of 254 characters long, or contain a maximum of 30 expressions strung together, whichever comes first. Since the screen is only 40 characters wide, the input area for the query command wraps around the screen several times. Therefore, if your command is short, pressing <enter> may only advance you to the next line (since full-screen editing is in effect). If that's the case, just press <fctn-9>, which always completes the entry, no matter where you are.

Section 2 - QUERY Command Format

Here is a summary of the basic format for a query:

<field> <eq> <data> [<logic> <field2> <eq> <data2>..]

example: MAGAZINE = BYTE AND YEAR >= 1985

(**Note**: Everything within square brackets is optional.)

The query command is one or more expressions glued together with the words AND or OR. *MAGAZINE = BYTE* and *YEAR >= 1985* are two valid expressions connected with the word *AND*.

Each expression specifies:

- a field.
- a piece data.
- a relationship between them.

The program looks at each record in the database and checks to see if fits the query command. In the above example, the program will find all the records in which the *MAGAZINE* field contains the string *BYTE* and the *YEAR* field contains a number greater than or equal to *1985*. Note that because we use *AND*, both conditions must be met before a record can meet the requirements. If *OR* was used instead, only one condition needs to be fulfilled.

An explanation of each component in the query command follows.

<field>

This can be one of two things, each specifying a different kind of action.

ALL : By typing the word ALL and nothing else for the QUERY command, FirstBase will find all of the records in the database. Deleted records are ignored. This comes in handy if you want to view every record before performing a complex query, or for making a copy of a database. Anything typed after the word ALL will be ignored.

NOTE: A blank line for a query automatically defaults to ALL.

FIELD: If the first word is a valid, indexed fieldname, the program will search that field for the information you give it.

NOTE: you can use partial fieldnames, and if you need to specify a fieldname that contains spaces, you must enclose it with in quotes. All the rules for partial strings apply here. For example, TOTAL* and "TOTAL SALES" are valid fieldnames, and they can be considered the same field since TOTAL* is partial string of "TOTAL SALES."

<eq>

This stands for an equals symbol, which can be any of the following.

=	equal
<>	not equal
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

It specifies the relationship between <field> and <data>. For info on <data>, see below. Example:

YEAR >= 1987 (looks for *YEAR* greater than or equal to 1987)

<data>

This is a piece of data -- a string constant, for example -- which is compared to the contents of the field indicated in **<field>** above. It can be one of three things: a string constant, a number, or another valid, indexed fieldname.

You don't have complete freedom in the type of data you can use in a query command. Basically, the rule is: you can only compare information of the same type. That means you must compare numeric fields to numeric constants (and fields), but you can compare text fields to any type of constants (and fields).

This should make sense. A numeric field will never contain text, so you shouldn't be able to search for text in such a field. A text field, however, can contain anything, including numbers. Therefore, you should be able to search for any kind of data in a text field.

Here are some brief notes about each data type:

string: If the data is a string constant, the program tries to find the indicated relationship between this string and the field. Spaces are considered delimiters, so surround the data in quotes if you need to include spaces.

Examples:

MAGAZINE = "Personal Computing"
DATE <> JUNE88

You can use all the equality symbols, such as > (greater than). In such cases, FirstBase will determine if the string constant is alphabetically less than or greater than the indicated field.

number: If **<field>** is numeric, than the data can be a numeric constant.

Example:

SALES >= 2000.50

@field: You can select another field as your data if you want to compare the contents of two fields within the same record. The same rule about data types applies here: if **<field>** is numeric, then **<data>** can be a numeric field. If **<field>** is text, **<data>** can be a text or numeric field.

Fieldnames MUST be preceded by the @ symbol (shift-2) if they are used as data!

Example: SUBTOTAL > @GRANDTOTAL

That's it. All you need for a query command is a fieldname, an equals symbol, and a piece of data. If you need something more complex, you can string together multiple expressions with logical connectors.

<logic>

If you want specify two or more relationships between fields and data, connect them with either AND or OR.

AND: The expression on <u>both</u> sides of the AND must be true before the record can be considered a match.

OR: Only one expression, on either side of the OR, needs to be true for the record to match. Additionally, both can be true.

You can string as many expressions together (up to 30) with logic operators as will fit within the 254 character maximum.

Parentheses

The expressions in the QUERY command are evaluated left to right; you can use parentheses to change the order in which they are evaluated.

Remember to precede and follow each parenthesis with a space.

Example:

```
MAG = "COMP* LANG" OR ( MAG = BYTE AND ISSUE = "MAY 1987" )
```

Section 3 - Query Destinations

The *destination command prompt* comes after the query command, and looks like this:

Screen, New, Append, Report?

Sending records to screen:

If you've selected this option (by pressing S), FirstBase will display each record found in a format similar to that used in BROWSE. Once the entire record is shown, you will see the prompt:

Continue search, Quit?

As with other command prompts, pressing the first letter activates the command. C continues searching and Q ends the query immediately. In addition, pressing and holding fctn-4 while the program is searching the file will abort the query.

Sending records to a New database:

If you select this option, you are asked for a new database name. FirstBase creates a new database on the disk in the OUTPUT DATA DRIVE. This database will be identical in structure to the current database; the dictionary is simply copied to create the new files.

Sending records to an old database:

This feature can be very handy; it appends the records found with the query command to an existing database. You need to specify the database to append to, and it must be in the OUTPUT DATA DRIVE.

> **NOTE**: Both the current database and the one to be appended MUST have identical definitions in every way, including fieldnames, lengths, indexes, etc.

Sending records to the REPORT program:

If you want to print a custom-formatted report of a group of records, you first run the QUERY program and press 'R' for the destination. You then must type in something called a *Report Record List* filename. This file contains a list of records that QUERY finds. Then, furnishing the REPORT program this same filename allows will print the records out nice and pretty. It's a two-step process: do a QUERY, then run the REPORT program.

Section 4 - Number Formats

There are two restrictions on the kinds of numbers the QUERY and UPDATE programs accepts.

1 - Decimal numbers. Any integer value from 0 to 32767, and any decimal number with a total of seven digits.

    Examples: 12 1.2 33456.99

2 - Positive numbers ONLY. Currently, negative numbers (like -34.50) are not supported.

You cannot use dollar signs or commas (like $34,598.05); instead use 34598.05 . If you need dollar signs, commas, etc., define the field as TEXT instead of NUMERIC when you create it. However, doing so will preclude you from doing any kind of numeric searching and calculating.

Section 5 - Examples of Query Commands

```
MAG = BYTE AND ( YEAR = 1985 OR YEAR = 1986 )
```

Finds records with MAG containing 'BYTE' and YEAR containing either 1985 or 1986.

```
( DEPT = 5 OR DEPT = 6 )  AND MAILSALES > @STORESALES
```

Finds records with DEPT containing a 6 or 7 and the number in the MAILSALES field greater than the number in the STORESALES field. First it checks the department field (going left to right), then it compares the MAILSALES and STORESALES field. Also note the @ sign preceding the STORESALES field; this is because the field is being used as a data item.

```
SALES >, 5000 AND ( DEPT = 6 OR DEPT = 7 )
```

First the program examines the SALES field for a number greater than 5000. If found, it checks to see if the DEPT field contains a 6 or 7. If either number is found, the record is sent to the specified destination. Compare the above to this variation, which does not use parentheses:

```
SALES > 5000 AND DEPT = 6 OR DEPT = 7
```

Everything is evaluated left to right; so first the program checks to see if the SALES field contains 5000 and the DEPT field contains a 6. If so, it then checks for the DEPT field to contain a 7. This is meaningless, because if the first two commands are true, then DEPT = 6; DEPT will not equal 7. The program groups commands into pairs, from left to right. The SALES and DEPT fields are checked first, and evaluated. Then the DEPT field is checked again. Obviously, this is not the order in which the command should be evaluated.

To illustrate, here is the second command again:

```
SALES > 5000 AND DEPT = 6 OR DEPT = 7
```

Now assume that a record contains the following values:

```
SALES = 6000    DEPT = 7
```

The steps the program would take to evaluate the second command would be:

1) Is SALES > 5000? YES.

```
2) Is DEPT = 6? NO.
3) YES <AND> NO evaluates to NO.
4) Is DEPT = 7? YES.
5) NO <OR> YES evaluates to YES.
6) We have a match!

That worked out fine. But now look:

SALES = 4000 DEPT = 7

1) Is sales > 5000? NO.
2) Is Dept = 6? NO.
3) NO <AND> NO evaluates to NO.
4) Is Dept = 7? YES.
5) NO <OR> YES evaluates to YES.
6) We have a match, which is INCORRECT!

Now switch back to the first command, with the parentheses:

SALES > 5000 AND ( DEPT = 6 OR DEPT = 7 )

SALES = 4000  DEPT = 7

1) Is SALES > 5000? NO.
2) Aha, a parenthesis. Don't evaluate immediately.
3) Is DEPT = 6? NO.
4) Is DEPT = 7? YES.
5) Aha, close parenthesis. Evaluate what's between them.
6) NO <OR> YES evaluates to YES.
7) Now go back and pick up the SALES = 5000.
8) NO <AND> YES evaluates to NO.
9) We don't have a match, which is correct!
```

Section 6 - Summary

- A query is a command that describes the contents of a record. FirstBase uses this command to select a group of records.

- The QUERY program can send a group of selected records to the screen, a database file, or the report program.

- A query command contains at least one *expression*, which describes the contents of one field.

- A query command can be up to 254 characters long, or contain up to 30 expressions, whichever comes first.

- To list more than one expression, connect them together with AND and OR.

- While you enter a query command, and the command happens to be short, the program may not do anything after you press <enter>. This means the cursor is not on the last line of the input field. Simply press <fctn-9> to continue.

- A query expression contains three things: a field, a piece of data, and a relationship between them.

- Alternately, a query command can be the single word *ALL*; this will select every record in the database that is not deleted.

- The *field* portion of the query command must be a valid, indexed fieldname.

- You indicate the relationship between the field and data by using an equals symbol: =, <>, >=, <=, >, <.

- The *data* portion of the query command can be one of three things; this is determined by the field type of the *field* in the command.

- If the *field* is numeric, the *data* can be either a numeric constant (like *25.95*) or the name of a numeric field.

- If the *field* is text, the *data* can be anything: a numeric constant, a numeric field, a text constant, or a text field.

- If the *data* is a fieldname, it must be preceded by the @ sign. Example: SALES88 >= @SALES87

- Query commands are evaluated left to right; to change the order, you can use parentheses. Remember, a space must precede and follow each parenthesis.

- To obtain a formatted report from the query program, you must select *Report* as the query destination. FirstBase will ask for a *Report Record List Filename*. This will create a file in the output data drive containing a list of the records to print. You must then run the REPORT program and furnish this same filename.

- You cannot perform calculations on fields that contain dollar signs, commas, or any other kind of text. The only kind of valid numeric items are pure numbers, with or without a single decimal point.

CHAPTER 6 - The UPDATE program

Chapter 5 showed you all you needed to know about querying a database to select a group of records. You also found out that you can send those records to a variety of destinations: to the screen, a database file, or the report generator. There is yet another action you can take on those records: batch processing.

What is batch processing? Simply put, it's performing a single action on a batch of records using only one command. This action can take on two forms. The first is merely storing a string constant in a field. As an example, perhaps you need to change the membership status of user group members who haven't attended a meeting in over a six months from ACTIVE to INACTIVE. The hard way would be to load in each record and manually type in INACTIVE for the *Status* field. With FirstBase, however, you can do that with a single, two-part command. The first command is a query, which selects the records to update. Then you can type in an update command, which would be *STATUS = INACTIVE*. This would set the *STATUS* field of each selected record to *INACTIVE*.

The second form of update command stores a calculated result in a field. For a simple example, suppose you want to give everyone in your New York office a 2% raise. Typing in a query of *OFFICE = "NEW YORK"* will select the records you need to change. Then, for the update command, use *SALARY = @SALARY + @SALARY * .02* . This would multiply whatever number is in the *SALARY* field by .02, yielding 2% . The number would then be added to the number in the *SALARY* field, and would finally get stored back into the *SALARY* field.

Introduction.....66

Update command format.....66
Arithmetic.....67
String constants.....67
Operator precedence.....67

Using DELETE and UNDELETE.....68
DELETED in a query.....68
DELETE & UNDELETE in update...68

Examples of update commands.....69

Examples of DELETE and UNDELETE....69

Summary.....70

Section 1 - Introduction to UPDATE

PURPOSE: The UPDATE program lets you perform mass-updating (sometimes known as batch processing) of a select group of records. You can think of it as a 'search-and-replace' function in a word processor. In order to update records, you must find them first with a query command. The query command used in this program works exactly as it does in the QUERY program with one exception; you can also search explicitly for deleted records.

After typing in your query command, you need to provide an update command, which specifies the field to be updated and the data to be stored there.

NOTE: You can only updated one field at a time.

Section 2 - UPDATE command format

The basic structure of the update command is as follows:

<field> = <data> [<op> <data2> <op> <data3> ....]

(**Note:** everything in square brackets is optional.)

This command translates into "replace the data in **<field>** of each record found in the query command with **<data>**." **<data>** may be a string constant, a fieldname, or a calculation consisting of fieldnames and/or numbers. The nature of **<data>** is determined by the field type of **<field>**.

The maximum length of the update command is 254 characters.

A null (blank line) for the query command defaults to ALL.

A null (blank line) for the update command aborts the operation and returns to the main menu.

<field>

This is the destination of the operation, and can be any valid indexed fieldname. This is where the new data will be stored.

You must follow the destination field with an equals sign; doing otherwise results in an error message and a return to the main menu.

<data>

This can be another indexed fieldname, a string constant, or a number.

If **<field>** is a numeric field, **<data>** must be numeric in nature. That means you can only use a numeric constant, another numeric field, or a calculation.

If **<field>** is a text field, **<data>** can be anything: a string constant or field, a numeric constant or field, or a calculation. By using a fieldname for **<data>**, FirstBase retrieves the contents of that field and uses it in evaluating the update command. As a result, you can copy one field to another in the same record, or do a calculation using the values in other fields from the same record.

You can optionally extend the command line to perform calculations, as described next.

Performing arithmetic

<op> stands for an arithmetic operator. Four are currently supported: +,-,*,/.

<data2> is just like <data>, with one exception. It can be another fieldname or a numeric constant, but NOT a string. Why? When was the last time *you* tried to add or multiply two words together??

Remember that if you specify a fieldname after the equals sign, you need to include the @ symbol, as shown in the examples at the end of this chapter.

String constants

As discussed in the **Introduction** section of chapter 2, a string constant may or may not be surrounded by quotes. If the string is only one word, no quotes are necessary. If there are several words to the string, separated by spaces, they MUST be surrounded with quotes.

Precedence of operators

The word *precedence* simply means "what gets done first." In the UPDATE program, a calculation is evaluated just like in most programming languages and like a scientific calculator would: all

multiplications and divisions are evaluated first, and all additions and subtracts are evaluated last. If there is more than one multiply and/or divide, they are done left-to-right. The same goes for more than one add/subtract. To alter the order of precendence, use parentheses. Precede and follow each parenthesis with at least one space.

Section 3 - Using DELETE and UNDELETE

A powerful feature of FirstBase is the ability to delete (or undelete) a group of records and, conversely, to search for deleted or undeleted records. THIS CAN ONLY BE DONE IN THE UPDATE PROGRAM.

Using DELETED in a query

Normally, a query ignores records marked as deleted. If you need to force the opposite to happen, the single word DELETED for a query command searches for records marked as deleted. This cannot be combined with other expressions -- it must appear alone in the query command.

Using DELETE and UNDELETE in an update

The words DELETE and UNDELETE for an update command either deletes or undeletes all of the records found with the query command. Again, the command cannot be combined with any other expressions -- it must appear alone.

Examples of using DELETE and UNDELETE appear at the end of this chapter.

Section 4 - Examples of UPDATE commands

```
TAX = @SALES * .04
```

Takes the number in the *SALES* field, multiplies it by .04, and stores the result in the *TAX* field. *SALES* must be numeric.

```
COMPLETED = NO
```

Stores the string constant NO in the *COMPLETED* field. No quotes are necessary. *COMPLETED* must text; you cannot store text into a numeric field.

```
NUM = @N1 * ( ( 4 + @DAYS ) / 365 )
```

Adds 4 to the number in the *DAYS* field, divides it by 365, multiplies it by the number in the *N1* field, and stores the result in the *NUM* field. *DAYS* and *N1* must be numeric.

Section 5 - Examples using DELETE and UNDELETE

If, for some reason, you want to perform a calculation on deleted fields, you can use the word DELETED for the query command:

```
Query: DELETED

Update: PRICE = @PRICE + @PRICE * .5
```

If you want to undelete all deleted records, you can use the word UNDELETE for the update command as well:

```
Query: DELETED

Update: UNDELETE
```

If you want to delete a group of records, you can use this:

```
Query: MEMBERSHIP-EXP-DATE < 1986

Update: DELETE
```

If you want to delete ALL records in the database:

```
Query: ALL

Update: DELETE
```

Section 6 - Summary

- The UPDATE program lets you either store information into a single field of many records. The records are chose with a query command.

- You can store a single string constant, a numeric constant, or a complex calculation with the update command.

- You can update only one field at a time.

- You can use only indexed fields in an update command. You cannot use non-indexed fields as the destination field, or as part of a calculation.

- The maximum length of the update comand is 254 characters.

- A null (blank line) for the query command defaults to ALL.

- A null (blank line) for the update command aborts the entire operation and returns to the main menu.

- The first thing you list in an update command is the destination field.

- You must follow the destination field with an equals sign.

- The data following the equals sign can be another indexed field, a string constant, a numeric constant, or a calculation.

- If the destination field is numeric, the final result of the update must be numeric in nature.

- If the destination field is text, the final result of the update can be either numeric or text.

- Four arithmetic operators are currently supported: + - * / .

- You can use parentheses to alter the normal order of evaluation; FirstBase evaulates a calculation left-to-right.

- You can use the DELETED keyword (only by itself) in the query command to search for deleted records.

- You can use the two keywords DELETE and UNDELETE in the update command to alter the "deleted" status of records. Only one keyword (and nothing else) may be used at a time.

CHAPTER 7 - The Sort Program

The world at large seldom comes in a neat, sorted package. You are assaulted by information and random times, and most likely in random order. That is why people invented the sort -- to put things in their proper order. That is exactly what the SORT program in FirstBase does: arranges records in sorted order.

When designing FirstBase, we wanted to incorporate a flexible sorting procedure that would satisfy most, if not all, requirements people might face.

FirstBase provides many options in sorting a database. You can specify what fields (called keys) to use for sorting (up to a maximum of 75). You can sort a database in ascending or descending order. You can choose to generate a new file, leaving the original untouched, or to sort the existing file. If you choose to sort the existing file, you have the extra option of sorting just the index (which lessens the sorting time), or the entire database, records and all.

Also, please note that this is one of two program which permanently removes delete records; the other program is QUERY. If you choose to sort an existing database, any deleted records will be gone forever. If, however, you choose to generate a new database, the original is untouched; therefore, the delete records will remain in the original file, but will be gone in the new file.

SORT can only be accessed through the EXTRAS program, which is a selection on all main menus.

Introduction.....72

Sorting the database.....72
- Fields to sort.....72
- Ascending/descending.....72
- New file or existing.....72
- Sort index or database.....73

Sorting example.....73

Summary.....74

Section 1 - Introduction to the SORT program

Purpose: The SORT program allows you to re-arrange the records in the database, in either ascending or descending alphabetical order. Numerical sorting is not supported. Either the index or the database records can be sorted; sorting only the index saves time.

NOTE: Sorting automatically removes all deleted records from the database! They can never be recovered.

Section 2 - Sorting the database

Sorting a database is simple. Each of the following subsections explain the information you must provide.

Fields to sort:

The first thing you must furnish the program is a list of the fields to be compared. YOU MUST USE INDEXED FIELDS! You can list as many fields as you like, so long as you don't exceed 254 characters.

For example, to sort by LAST NAME, then FIRST NAME, you would enter:

"LAST NAME" "FIRST NAME"

or

LAST* FIRST*

Yes, you can use wildcards in the fieldnames.

Ascending or Descending?

Type 'A' if you want the records sorted in ascending order, or type 'D' for descending order. This applies to alphabetical information only. Numeric comparisons are not made when sorting; FirstBase always uses the ASCII values, even of numeric data. Sorting on numeric fields can yield unexpected results.

New file or existing?

FirstBase needs to know if you want to *Generate a (n)ew file or sort an (e)xisting one*. If you hit **N**, the program will make a

sorted copy of the database, leaving the original file untouched. The program will ask you for a new database name as well.

If you choose to sort an existing file by hitting **E**, FirstBase will first generate a temporary file that contains the sorted records, then delete the original database, and then rename the temporary file to the original name. This presents an interesting problem. The temporary file is always opened on the output data drive. If the output drive is different from the input drive, the database will seem to jump to the output drive! This can be confusing, so we suggest that if the output and input drives are defined differently in your system configuration, temporarily change them to be THE SAME using menu option 2 - *Move to new database*.

Sort index or database?

If you choose to sort an existing database, you need to answer one additional question: whether you want to sort just the index or the entire database.

Sorting the index is much faster than sorting the entire database, which entails rearranging all of the records. Sorting the index doesn't touch the records, only the index. Do whichever you prefer.

Section 3 - Sorting example

What follows is a screen representation of a typical session with the SORT program. This will show how to sort a database on the following fields:

STATE
LAST NAME
FIRST NAME

Items in **bold** indicate entries the user makes.

```
==============================================================
Enter fields to sort: STATE LAST* "FIRST NAME"

Ascending or descending order? A

Generate a (n)ew file or sort an (e)xisting one? E

Sort (i)ndex or (d)atabase? I

Sorting, please wait.
==============================================================
```

Section 4 - Summary

- Sorting an existing database automatically removes records marked as deleted. They can never be recovered once the file is sorted. If you generate a new file, however, the delete records will remain in the old file and will be gone in the new file.

- You can sort on as many keys as you need; the only limitation is that the fields used as keys must be indexed.

- The standard rules for naming fields apply here: use quotes if fieldnames contain spaces, and you may also use wildcard characters.

- FirstBase sorts alphabetically - you cannot sort numbers so they come out in numerical order. Numbers are treated like text, and their ASCII values are used in the sorting process.

- If you choose to generate a new database, you need to provide a database name different from the one being sorted. This new database will be identical in structure to the original, only the records will be in sorted order.

- If you choose to sort an existing database, the program will store the sorted database in a temporary file in the output data drive. Once it finishes sorting, it deletes the original file and renames the temporary file back to the original.

- Temporarily change the output drive to be the same as the input drive if you want to sort an existing file; that way the database will not seem to jump from one disk to another.

- Changing the output drive can be done through the SORT program's main menu. Hit 2 for *Move to new database*.

- If you choose to sort an existing database, you also have the choice of sorting the index or the database; the only difference is that sorting the index takes less time.

- Sorting is a one-shot deal. Once an index (or database) is sorted, any new records added to the file will not be put in sorted order..they will be tacked on to the end of the database.

CHAPTER 8 - Macros

Macros are a short-cut when you need to perform complicated queries, updates, and sorts. Macros record your keystrokes and save them to disk; you can recall them later by typing in a macroname and watch your program obey your commands!

When you save a macro, you need to give it a name; use the same rules as when naming a database. FirstBase will append the characters "_M" to the macro name when saving your query, update, or sort commands to disk.

Where to use macros.....76

How to save a macro.....76

How to load a macro.....76

Example of query macro.....77

Example of update macro.....77

Example of sort macro.....77

Summary.....78

Section 1 - Where to use them

Macros can be used in the following programs:

QUERY: Records the query command and the destination.

UPDATE: Records the query and update commands.

SORT: Records fields to sort, ascending/descending, sort index/database, and new/existing database information.

There are two steps in using macros: first you must *save* the macro, and then you must *load* the macro in order to use it. Once a macro is saved to disk, it can be recalled at any time.

Section 2 - How save a macro

Saving a macro is easy. Three simple steps are outlined below:

1. Actually perform the actions you want to record. If you want to save a macro for the SORT program, you need to do the actual sorting first.

2. When finished, choose **Save Macro** from the main menu.

3. Type in a macro name when asked. The same rules apply as when naming a database...maximum of 8 characters, no spaces, periods, or quotation marks.

The disk spins, and the commands you just performed are saved to disk. Now they can be recalled any time you like.

NOTE: Macros are always saved to the input data drive.

Section 3 - How to load a macro

Loading and running a macro is even easier than saving one. Whichever program you are in, proceed as you would normally until you reach the first prompt. In the QUERY and UPDATE programs, that prompt would be the **"Query:"** prompt. In the SORT program it would be the **"Fieldnames to sort"** prompt. Instead of typing in the usual commands, type in the macroname preceded by a percent sign (%).

For example, to execute the macro called SORTNAME, type in *%SORTNAME*. The macro will load in and run automatically.

Since filenames can change often, macros do not record any filenames. You need to type in a filename, for example, when sending a query to a new database, or generating a new database with the SORT program. In that case, the program still asks you for a filename when you execute a macro.

Section 4 - Example of QUERY macro

What follows is a screen representation of executing a QUERY macro. Letters in bold type are what you, the user, should enter.

```
================================================

Enter query command: %MYQUERY

Enter output db name:
TESTDB2

================================================
```

This macro was saved under the name of **MYQUERY**, and the query had a destination of New Database, so it asked for a new database name. If no database name is required, the program would immediately execute the macro.

Section 5 - Example of an UPDATE macro

Here is the screen representation of calling a macro in the UPDATE program.

```
================================================

Enter query command: %MYUPDATE

================================================
```

Section 6 - Example of a SORT macro

Here is the screen representation of calling a macro in the SORT program.

```
================================================

Enter fields to sort: %SORTMAC

Enter new database name: TESTDB3

================================================
```

The sort macro was saved under the name of **SORTMAC**; the original command asked to generate a new database, so FirstBase asks for a new database name. Had the original command been to sort an existing database, the program would not have asked for a database name.

Section 7 - Summary

- Macros can be used in the QUERY, UPDATE, and SORT programs.

- In the QUERY program, a macro records the query command and the destination (screen, new database, old database, printer). Even if needed, the macro does not record the output database name.

- In the UPDATE program, a macro records the query and update commands.

- In the SORT program, a macro records the fields to sort, ascending/descending information, sort new file or existing information, and the index/database information. Like the QUERY program, the macro does not record the new database name, even if it's required.

- A macro must be saved before it can be used.

- To save a macro, just select **Save Macro** from the main menu. After asking for a macro name, FirstBase saves the commands last issued to the input data drive.

- Use the same rules for naming a macro as in naming a database: up to 8 characters, and no spaces, periods, or quotes.

- To execute a macro in the QUERY and UPDATE programs, type the macro name preceded by a percent sign (like %MACRO1) when asked for a **query command**.

- To execute a macro in the SORT program, type the macro name preceded by a percent sign (like %SORTMAC) when ased for the **fields to sort**.

CHAPTER 9 - The DEFINE program

In the first eight chapter of this manual, you've been shown everything you need to know in working most of FirstBase features: creating databases, entering data, searching, sorting, and batch-updating. With all of that under your belt, there is one last task to attend to: defining and printing reports.

Reports are the vehicle through which you can communicate information to yourself and others. Accordingly, you must have a great amount of flexibility when formatting reports so they come out just right. FirstBase tries hard to accommodate your needs; the program provides you with many options in both the ways you can print your data and in the final appearance of your reports.

Before you can print a report, however, you need to define its format: how the records are to appear on the printed page, how many records per page, margins, etc. All of this is set up in the DEFINE program. You must define a report before you can print a report!

By now, you can probably tell that FirstBase is a big project, consisting of a great many programs. Unfortunately, all of these programs cannot be loaded off of one menu, simple because the screen isn't big enough! With that in mind, please note that like SORT, the DEFINE and REPORT programs can only be accessed through the EXTRAS program; EXTRAS can be accessed through any main menu.

Introduction.....81
  Use of OUTPUT drive.....82

Define Report.....83
  DEFINE prompt.....83
  Defining Page format....83
  Record format.....87
  Rows & columns.....87
  Defining record format..88

Change Report in Memory.....89
  Display.....89
  Change.....89
    Page format.....89
    Record format.....89
  Save.....89
  Load.....90
  Quit.....90

Load Report.....90
Summary.....91

Section 1 - Introduction to the DEFINE program

NOTE: The DEFINE program can be accessed by first loading the EXTRAS program.

Purpose: Before you start printing reports, you must define (or create) them first. The process is similar to creating a database. With the DEFINE program you provide such information as:

- Margins
- Headers, footers, page numbering
- Fields to print
- Location of fields on the page
- Size of the fields
- Text (labels)

A report definition consists of two separate files: the Page definition and the Record definition. As you might guess, the Page definition contains information about the page layout: margins, headers, footers, page numbers, etc. The Record definition describes how the records are to appear on the page.

Upon loading the DEFINE program, you see this menu:

```
======== FirstBase:Report Definition ========

     1 = Define Report
     2 = Change Report in Memory
     3 = Save Report
     4 = Load Report
     5 = Change Drives
     6 = Report Generator
     7 = Extras
     8 = Main program
     9 = Quit FirstBase

     Current Database:
     DSK2.
     ============================================
     Please make your selection:_
```

Here is a brief capsule of what each menu selection does:

Define Report: Cleans out any report definitions in memory, and lets you create a new report.

Change Report: Allows you to make changes to a report definition already in memory, or one that was loaded from disk.

Save Report: Lets you save the report definition currently in memory. You can save either the page definition, the record definition, or both. The files are saved in the input drive.

Load Report: Lets you load a report definition from the input drive. Report definitions are always stored in the input drive!

Change Drives: Gives you the chance to temporarily change the input drive, output drive, and database name.

Use of the OUTPUT DRIVE

DEFINE is an incredibly large program -- the largest of all the FirstBase programs, in fact. As a result, we had to make some tricky decisions when designing this program. One such trick is using a disk file as a temporary buffer for holding data. This temporary file is created in the output drive when DEFINE first loads, and it's deleted when you exit. The file never gets very big (unless you have a lot of print items), but it can slow down operations like displaying and editing the record format.

If FirstBase cannot open the temporary file in the output drive, you will receive an error message that looks something like this:

Cannot open DSK3.TEMP ...
c99 exit - rerun? (y/n)

Press Y if you want to try and run the DEFINE program again, or press N to quit FirstBase entirely.

A reason for this error message might be that there is no disk in the output drive; if so, just insert a disk and answer Y to error message. If the output drive doesn't exist (eg. the output drive is DSK3. but you don't have a third drive) then press N. The program gets information for the output drive from the configuration file; if the config file is defined incorrectly, the file needs to be changed. Run FirstBase again and change the config file through the BROWSE program. Remember to SAVE YOUR

CHANGES to the config file! For more information on changing the file, see the *Installation* section in chapter 2.

Section 2 - Define Report

One of two things will happen when you select *Define Report* from the main menu:

1. Upon a cold boot, FirstBase asks for a database name and opens it.

2. If a database was already in use prior to loading DEFINE, the same database is automatically opened.

A database MUST BE OPEN before you can define a report!

The DEFINE prompt

After FirstBase successfully opens a database, you see the DEFINE prompt, which looks like this:

Define: Page,Record,Both,Quit?

The first word (*Define*) indicates the task at hand; in this case, defining a report. The three choices are:

Page: Press 'P' if you want to define the page format.

Record: Press 'R' if you want to define the record format.

Both: Press 'B' if you want to define both page and record formats.

NOTE: In order to print a report, you must define BOTH a page and a record format!

Page Format

The page format describes the attributes of all the pages in your report. We cover each prompt below.

Skip perf: Type Y or N:

Y - Make the program skip the perforation at the end of each page.

N - The program will ignore the perforations and print over them.

Page length: Describes the physical length of the page in lines. An 8.5" x 11" sheet has 66 lines. Must be a number between 1 and 32767.

Top margin: The line where printing is to begin on the page. For example, if you want 5 blank lines at the top of each page, set the top margin at 6. Printing will begin on line 6. Must be a number between 1 and the page length.

Bottom margin: The line where printing is to end on the page. See **top margin** for more information. Must be a number between the top margin and the page length.

Left margin: The column where printing is to begin on each line. Must be a number between 1 and 32767.

Right margin: The column where printing is to end on each line. Must be a number between the left margin and 32767.

Fixed # of recs/page: Type Y or N:

Y - Only a fixed number of records will be printed on each page.

N - The program will fit as many records as it can on each page.

Max no. of recs/page: The number of records to print per page. If **fixed # of recs/page** is 'N'o, this prompt will be skipped.

Word wrap:

Type Y or N:

Y - Each field is reformatted to eliminate excess spacing. Words will not be split at the end of a line. If a field runs out of room at the end of a print line, it will wrap around to the next line.

N - Fields are not reformatted, and fields will be truncated if they don't fit on a print line.

Headers? (Y/N)

Type Y or N:

Y - Two headers are printed at the top of each page (one line for each header).

N - No headers are printed in the report.

Header 1:

Type in the first header, which will appear on the first line of the page (as defined by the top margin). Maximum of 132 characters.

Headers are automatically printed flush-left.

Entering a period (.) as the very first character in the header will center the header when printed. The period will not appear.

Entering a blank line will print a blank line.

Header 2:

Type in the second header. See **Header 1** for more information.

Footer? (Y/N)

Type Y or N:

Y - Print a footer. The footer is one line, and appears on the last line of the page (as defined by the bottom margin).

N - No footer is printed in the report.

Footer:

Type in the footer. Same rules apply as for the headers.

Position of page number:

You can specify the existence and position of page numbers. A message at the bottom of the screen displays all the possible entries for this prompt.

NOTE: Page numbers *must* occur in a header or footer. If you have no headers and no footers, you cannot print page numbers.

The following two-letter codes specify the position of the page number:

TR - top right of the page.
TC - top center of the page.
TL - top left of the page.
*** these are only valid for headers ***

BR - bottom right of the page.
BC - bottom center of the page.
BL - bottom left of the page.
*** these are only valid for footers ***

N - No page numbers printed in this report.

Number page 1? (Y/N)

Type Y or N:

Y - Prints the page number for the first page.

N - Suppresses the page number on the first page.

Record Format

The record format is the second component of a report definition, and it describes what information is to be printed for each record.

When you instruct FirstBase to print a report, the first thing it does is print, at the top of the page, any headers that have been defined. Then it starts looking for records to print. How it selects the record to print will be explained in the *Report Generator* section. Once it finds a record, it loads the record into memory, and prints out a bunch of information in accordance to the *record format*.

The record format is a list of *print items*. The two kinds of print items are **fields** and **text**.

A **field** print item causes the contents of the named field to be printed at a certain position on the printed page. A field print item contains the following information:

- Fieldname
- Row
- Column
- Size

The row and column information instruct FirstBase where to print the field in relation to the current record. The size indicates how much of the field should be printed.

A **text** print item merely causes a piece of text (perhaps a label) to be printed at a certain position on the printed page. A text print item contains the following information:

- Text to be printed
- Row
- Column

As with a field print item, the row and column information indicate where to print the text.

How FirstBase prints items: rows and columns

The row positions you provide for all the print items in your report are relative to the current record being printed. They are not absolute positions on the page.

For instance, if you have a print item with a row of 3 and a

column of 10, the program will NOT move to the third row on the page and tab over to column 10. Instead it will move down to

the third row for the current record relative to the current print position. Every time the program starts printing a new record, the current row position is reset to 1.

Defining the Record Format

With all that out of the way, you can now proceed to define the record format.

# of print lines: This number indicates the maximum number of print lines one record will take up on the printed page.

Fieldname or text? (F/T/Q) Press F, T, or Q:

F - Defines this as a field print item. The contents of a field will be printed.

T - Defines this as a text print item. Straight text will be printed.

Q - End the entry of record formats.

Fieldname: *or* ***Text:*** Depending on what you selected for the previous prompt, the cursor moves to either the Fieldname or Text prompt.

In the case of a fieldname, enter the name of the field you want printed. **The field does not have to be indexed.** Maximum of 20 characters.

In the case of text, type in the text to be printed. Maximum of 132 characters.

Row: The row position where this item is to be printed. The position is relative to the current record.

Column: The column position where this item is to be printed.

After entering the first print item, the cursor moves back up to the **Fieldname or text?** prompt. Continue entering all of your data, and when done, type Q (for Quit) at this prompt.

Section 3 - Change Report in Memory

Selecting this option from the main menu lets you modify a definition already in memory. The prompt you see is shown below:

Display,Change,Save,Load,Quit?

Display

This command lets you display the current definition. At the Display prompt, press 'P' displays the page format, 'R' for the record format, or 'B' to display both formats.

Change

This command allows you to change either the page format or record format.

Changing the page format:

FirstBase displays the same entry screens as if you were defining the format for the first time. Type in any changes, or just <enter> to accepts the current information.

Changing the record format:

Again, FirstBase displays the same entry screen as if you were defining the record format. First enter the type of print item you want to change (Field or Text), and then type either the fieldname or text to change. The cursor jumps back up to the top, where you can change anything about that print item, including changing its type from *Field* to *Text* or from *Text* to *Field!*

Save

This command lets you save the definition in memory to disk. The save prompt needs some explaining:

Page,Record,Both,Quit?

Since a report definition needs both a Page and a Record format, you would choose to save **Both** formats by pressing 'B' at this prompt. You can also choose, however, to only save one of the formats. This allows you to mix and match different page and record formats for maximum flexibility.

All you need to do now is provide a Report filename.

The Record format is saved under the name FILENAME_C.

The Page format is saved under the name FILENAME_P.

The files are saved in the input drive.

Load

Here you can load a report definition off of disk to examine, modify, and resave as you wish. For more information on the prompts and filenames, see Save above.

Quit

This returns you back to the main menu.

Section 4 - Load Report

This menu option allows you to load in a report definition from the input drive. It is identical to the **Load** command described in the previous section. Turn to that section for more information.

Menu options 5, 6, and 7 load in the indicated programs from the program drive.

Menu option 8, **Quit Firstbase**, exits FirstBase.

Section 5 - Summary

- The DEFINE program can only be accessed through the EXTRAS program. The EXTRAS program can be loaded through any main menu.

- Before printing any report, you must define it first with the DEFINE program.

- Choose *Define Report* from the main menu if you want to create a new report format.

- Choose *Change Report* from the main menu if you want to modify a report format that's already in memory.

- Choose *Load Report* from the main menu if you want to load a report format off of disk.

- Report formats are always saved to and loaded from the input data drive.

- The *page format* describes the layout of all the pages in your report; it defines things like margins, headers, footers, and page numbering.

- If you say **No** to *skip perf*, the program will print right over the perforation and not do any page breaks whatsoever. It ignores the *fixed recs per page* setting, since FirstBase effectively forgets that the report is broken into pages.

- *Word wrap* does two things: each printed field is reformatted to eliminate excess blank spaces, and if a word can't fit on the print line, it will wrap around to the next line, much like a word processor.

- Headers and footers are normally printed flush-left; to center them, type a period as the very first character.

- If you want to print page numbers, you must do it in the header or footer. If you have neither, page numbers will not print.

- When entering the page number position, you must type a two-letter code. A key to the codes appears at the bottom of the screen and on page 86.

- The *record format* describes how the database records will be printed, which is nothing more than a list of print items.

- A *print item* is a piece of information to be printed; it can either be a field or a piece of text (like a label).

- If you need to print a field, you must provide the field name, row, column, and size.

- If you need to print text, you must provide the text, row and column.

- The row number in a print item is relative to the current record. Every time a new record is printed, the row position is set back to 1. To print on the first row for the current record, enter a row of 1. To print on the 5th row, enter a row of 5.

- The column number in a print item is straightforward; it's an absolute measurement. It indicates the column in which the item is to be printed. To print in column 1, enter a 1. To print in column 76, enter a 76.

- While entering print items, they don't have to be in sequential order. You can have the first print item be at row 5, column 1 and the second print item be at row 2, column 30. The items will be sorted in the REPORT program when the actual printing takes place.

- The entire report format is saved as two separate files: the page format under *xxxxxxxx_P* and the record format under *xxxxxxxx_C* (the *xxxxxxxx* is the report name you provide).

- Because of the dual-file system, you can load in either format individually; in this way you can mix and match various formats.

- Remember: a report format must contain both a record and page format!

CHAPTER 10 - Report Generator

In Chapter 10, we explained in detail how to create a report format. It takes a bit of planning on your part to get everything placed properly on the page. Once you've gone through the DEFINE program, you can dive into the REPORT program to get a hardcopy of your hard work.

The REPORT program is remarkably simple to use; you provide only three pieces of information: the database you're going to use, whether you want to print ALL records or just those selected with a query, and if you use a query, the *list filename*.

The most common way to print a report is to first run the QUERY program and select the records you want sent to the printer. By selecting a destination of *Report*, FirstBase creates a file containing a list of records to be printed. The REPORT program can load that in and pull out just those listed records and send them to the printer.

As with the SORT and DEFINE programs, the only way to access the REPORT program is to run EXTRAS. This program can be found on any main menu.

Introduction.....94

Selecting records.....94
Using list file....94
Using ALL records..95

The report name.....95

Printing the report.....95

Summary.....96

Section 1 - Introduction to the REPORT program

Purpose: The REPORT program generates formatted custom reports of a database. You can print all of the records in a database, or only those selected with a QUERY command. Please refer to the QUERY chapter for more information on this command.

Here is the REPORT program's main menu:

```
=====FirstBase:Report Generator=====

          1 = Print Report
          2 = Change Drives
          3 = Define Report
          4 = Main Program
          5 = Extras
          6 = Query
          7 = Update
          8 = Quit FirstBase

    Current Database:
    DSK2.
    ================================
    Please make your selection:_
```

Only one menu option needs to be discussed: **Print Report.**

Upon selecting this menu option, you are asked for a database name. If a database has already been used during this session at the computer, FirstBase automatically opens it.

Section 2 - Selecting the records to print

The first prompt you see is the following:

Use list <f>ile or select <a>ll?

Press *F* to use a list file (i.e. the results of a query command), or press *A* to print all of the records in a database.

Using the list file

In the QUERY program, one of the destinations you may choose is a "Report record list file." This is merely a file which contains a

list of records. By selecting the list file option show above, FirstBase will use this list of records to print your report.

If you choose this option, you are asked for a filename; supply the same one you did for the QUERY program.

Printing ALL records

If you choose to print ALL records, it does just that: it reads in and prints every record.

As with every other program, deleted records are always ignored.

Section 3 - The report name

The second prompt you see asks you for a report name:

Enter report name:

This prompt asks you for the name of the report definition; this tells FirstBase how you want your report to appear on the printed page. The report name is the same name you saved the definition under in the DEFINE program.

Entering a blank report name aborts the report process and brings you back to the main menu.

Section 4 - Printing the report

Make sure you printer is positioned at the top of a new page before typing in a report name. Once the program starts printing the report, press and hold <fctn-4> to abort the printing.

Section 5 - Summary

- The REPORT program can only be accessed through the EXTRAS program. EXTRAS is available through any main menu.

- To print all the records in the database (excluding deleted ones, of course), choose the **A**ll option when asked to use list file or select all records.

- In order to print only a group of records, you need to select them first with the QUERY program, and specify the REPORT program for the destination.

- If you send the results of a query to REPORT, FirstBase saves a temporary file on the output drive that contains a list of the records to be printed. REPORT uses this list to print a report.

- If you choose to use the record list file, you need to enter the file's name in REPORT so FirstBase knows where to look. You can perform several queries, using different filenames for the record list. In this way you can print several reports using different sets of records without exiting REPORT and running QUERY time after time.

CHAPTER 11 - TUTORIAL

In the past ten chapters, you have been shown every nook and cranny of FirstBase. You've hopefully learned how to harness all the power that this program has to offer. Unfortunately, not every likes to read manuals, and people get impatient with boring reference material. This chapter is devoted to those people, as well as anyone who has gotten confused along the way. In this chapter you will find several short and sweet lessons on using the basic functions of FirstBase. Nothing complex or advanced is covered in these pages -- just enough to whet your appetite. Let the lessons begin!

In the lessons that follow, anything printed in **Bold** is to be typed in by the user. Anytime you see something in brackets, it indicates a special keypress (like **<enter>** or **<fctn-4>**).

Introduction.....98

Lesson 1: Creating a db.....98

Lesson 2: Using BROWSE 102

Lesson 3: Using QUERY 103

Lesson 4: Defining a report.....105

Lesson 5: Printing a report.....108

Introduction

The following lessons assume that you've at least read the **Introduction** chapter; it covers the basic loading process to get FirstBase up and running. It also covers some elementary conventions like naming databases and using the full-screen editor. Please glance through this chapter now before plunging ahead.

Lesson 1 - Creating a database

This lesson will introduce you to the CREATE and BROWSE programs by showing you how to create a new database and add some records.

1. Put disk 1 (which contains CREATE and BROWSE) in drive one.

2. Put a blank, initialized disk in drive two. This will be your data disk, which will hold the new database you're about to create. (For one-drive systems, insert the data disk after the program loads. Whenever you need to load a different program, reinsert the program disk.)

3. Follow the instructions in the **Introduction** chapter on how to load FirstBase.

4. Once FirstBase is loaded, you will be in the BROWSE program. Selection option **3 - Change System Configuration**. This will allow you to change the settings to match your computer setup. You specify which drive will hold programs, data, and temporary files, the printer name, and the wildcard symbols. For a detailed explanation on what all the settings mean, please refer to the Installation section of chapter 2.

5. Make any changes so the settings match your system by typing over the information and pressing <enter> after each one.

6. For single-drive systems, change the *output drive* and *input drive* to **DSK1.** Remember to enter the period! Once this is changed, FirstBase will use whatever disk is in drive one for all operations, including reading/writing database files and writing temporary files.

7. After the last setting, press **Y** to save the configuration to disk. If no changes were made, press **N.**

8, Now you're brought back to the main menu. Select **5 - Create** to load in the CREATE program. If you have a one-drive system, insert the program disk in the drive before selecting this menu option.

9. After the CREATE program loads, select menu option **1 - Create new database**. This will start you off in creating a new database.

10. You will now see the definition entry screen. Here is where you define the fields you want in your database. Type in the following information:

    Fieldname: **Article Title**
    Field length: **40**
    Index this field? **Y**
    Index length: **40**
    Fixed field? **N**
    Text or numeric? **T**

11. The cursor will jump back up to the fieldname prompt. Enter the following data:

    Fieldname: **<fctn-4>Magazine**
    Field length: **20**
    Index this field? **N**
    Index length:
    Fixed field? **Y**
    Text or numeric? **T**

    <fctn-4> will clear the line so you can type in a new fieldname. The program will not let you define two fields with the same fieldname!

    Back at the fieldname prompt, enter one more field definition:

    Fieldname: **<fctn-4>Issue**
    Field length: **20**
    Index this field? **Y**
    Index length: **15**
    Fixed field? **Y**
    Text or numeric? **T**

12. To end the definition process, press <fctn-9> at the Fieldname: prompt. Pressing <fctn-9> is most places will end the task at hand.

13. You will now see the CREATE program's command prompt. It looks like this:

    List,Change,Add,Save,Delete,Quit?

    Press **L** for List definition. This command displays the database definition on the screen in a tabular fashion. You

database definition on the screen in a tabular fashion. You can now review your entries and note any mistakes.

14. You made a mistake! you wanted the MAGAZINE field to be indexed. At the bottom of the screen you see the command prompt again, so press **C** for Change. This option lets you change any field's definition.

15. You are presented with the a display very similar to the one when you first defined your fields. The cursor is sitting at the Fieldname: prompt. Type **MAG***, which means you want to change a field whose name begins with the three letters MAG. In other words, the MAGAZINE field. The asterisk is the *multiple wildcard symbol*.

16. The only things you want to change are the index status and index length, so press <enter> until you get down to the *Index this field?* prompt. Type **Y** to index the field. The at the next prompt, type in an index length of **20**. This instructs the computer to save the first 20 characters of the MAGAZINE field whenever a record is saved or modified. Press <enter> for the remaining prompts.

17. At the fieldname prompt, press <fctn-9> to get out of the *Change* command. If you'd like to change any other fields, you can instead type in another fieldname and follow the same basic steps.

18. Once you're out of the *Change* function, youg get the command prompt again. Press **L** to list the information again. Notice that the MAGAZINE field has been changed.

19. At the command prompt, press **S** for Save definition. This will save your new definition in the input data drive. Before it can save it, though, you need to give your database a name.

20. When asked for a database name, type in **TEST**. FirstBase will save the definition out to the input data drive. The database definition file will be saved as **TEST_D**. A new, empty index and data file will also be written; the index file will be called **INDEX_I** and the data file will be called **INDEX_F**.

21. After the drive stops spinning, press **Q** (for Quit) at the command prompt to get back to the main menu.

22. Load the BROWSE program by selection **5 - Browse** on the main menu. You're about to add a new record to the database!

23. After the BROWSE program loads, select **1 - Browse database**. FirstBase automatically opens the **TEST** database. If you have a single drive, remove the program disk and insert the data

disk. You may have to reselect option **1 - Browse** database to open it.

24. You are automatically put in *Add* mode, where you can enter new records. Look at the top of the screen: you should see the word ADD.

25. Type in the following information for your first record:

    Why Fast Copiers Don't Work <enter>
    MICROpendium <enter>
    July 1988 <enter>

26. Once you're finished with the last field, you get the browse prompt; this works just like any other command prompt. Type **S** for Save record, then **Y** when asked *Are you sure?* This will save your new record to disk.

27. At the command prompt, press **A** for Add new record. This will put you back in ADD mode, just like in step 24. Notice that the *Article Title* field is cleared, but the other two retain their information; that's because you defined the *Magazine* and *Issue* fields to be fixed. Fixed fields retain their contents when you add records. This saves typing. You can, of course, change their information if you like; just type over the current information.

28. We only want to change the first field, so type the following:

    Animation With The Comic Show Editor
    <enter>
    <enter>

29. Like before, press **S** at the command prompt to save the record.

30. Now that you've added two records, let's look at them. To move backwards a record, press **B** at the command prompt.

31. To move forward one record, press **F**. Now you are viewing the first record you added.

32. At the command prompt, press **Q** to stop browsing records. This brings you back to the main menu. Select option **8 - Quit FirstBase** to end this session.

33. End of lesson 1.

Lesson 2 - Using BROWSE

This lesson will show you a few more tricks in the BROWSE program. This and all following lessons require the sample database called BCS.

1. Insert the program disk and the data disk containing the sample file (called BCS) where appropriate (see lesson 1.)

2. Load FirstBase as usual.

3. Select **1 - Browse database**. This will open a database file. Since this is a new session, you need to type a database name.

4. For a database name, enter **BCS**.

5. FirstBase opens the file and loads in the first record. Instead of being in *Add* mode, you are now in *Display* mode. This is because the database is not empty.

6. Since the records in this database are larger than one page, you see the following prompt at the bottom of the screen:

   Next page,Previous page,Quit?

   Press **N** to display the next page of information.

7. You see the same page prompt again; press **P** to display the previous (first) page of data. You can flip the pages by pressing P and N as many times as you like. If you press N on the last page, FirstBase tells you there are no more pages to view. The same holds true if you press P on the first page.

8. Press the space bar or **C** to get the browse prompt back.

9. Press **F** a few times to view the contents of the database. Once a record loads in, just press the space bar or **C** if you don't want to view pages other than the first page. The **F** command will load in and display successive records in the database, moving forwards.

10. At the browse prompt, press **I**. This loads in an Individual record; you need to supply a record number. Enter **20** to load in record 20. When it's displayed, look at the upper left corner of the screen. You'll see the record number (20) in brackets.

11. Type **E** to edit the record. This allows you to make changes.

12. Move the cursor down to the *Languages* field by using the arrow keys or pressing <enter>.

13. Type in **XBASIC <enter>** in the languages field.

14. Once you get the browse prompt back, type **S** for save, and then **Y** to really save it to disk. The new information overwrites the old stuff.

15. Use **B** and **F** at the browse prompt to move around and view the record you just edited.

16. Press **T** and **L** at the browse prompt to load the Top (or first) record and the Last record, respectively.

17. Press **K** to perform a search. You are first asked for a fieldname: type **disk name**, and for a search string type "**c libraries**" **<fctn-9>**. The quotes are important! FirstBase will now look at the *Disk Name* field of every record for the string *c libraries*. Since you didn't use wildcards, the field must match perfectly. Upper/lower case is always ignored, however.

18. After a few seconds, the program finds a record that matches you criteria. Press the space bar once the first page of the record is displayed, then press <enter> to continue the search.

19. Once the search is finished, get the browse prompt by hitting any key, then type **Q** to get back to the main menu.

20. End of lesson 2.

Lesson 3 - Using QUERY

This lesson shows you the ropes of the QUERY program. It uses the enclosed BCS database.

1. Insert the program disk in drive one and the data disk in the appropriate drive. (see lesson 1)

2. When FirstBase first loads, you are immediately brought to the BROWSE program. You want the QUERY program, so select QUERY from the main menu.

3. Once QUERY loads, select option **1 - Query database**. The first thing that the program will do is ask for a database to open. Type in **BCS** for a database. Make sure the BCS disk is in the input data drive!

4. When asked for a query command, type:

   "disk number" > 10 and "disk number" <= 15 <fctn-9>

   This command will search the *Disk Number* field for a number greater than 10 and less than or equal to 15. In other words, it will look for a disk number of 11, 12, 13, 14, or 15.

5. The next thing you see is the *destination prompt*:

   Screen, New, Append, Report?

   Press **S** to send the records found to the screen.

6. FirstBase will load in all the records that met your criteria and display each one. Press <enter> until all of the records are displayed.

7. After record 15 is displayed, you can press <fctn-4> to stop the rest of the search.

8. Back at the main menu, select option **1 - Query database** again. You are about to generate a new database using the same query command. FirstBase automatically opens the last database in use; in this case, the BCS database. If you needed to change database names, do so at the main menu (the *Move to new database* option).

9. The query command you used before will be displayed. Press <fctn-9> to accept the same query command.

10. For a destination, press **N** for New database. This option will send the records found to a new database instead of the screen. When asked for a new database name, type **TEST2**. WARNING: don't type in the name of an existing database! If you do, the information will be overwritten. The database will be created in the output drive. Make sure a disk is there!

11. Wait for the process to complete. It will take a few minutes to scan the BCS database and copy the index, records, and dictionary over to the new file.

12. When finished, the program will bring you back to the main menu. Once there, select option **2 - Move to new database**. This will allow you to change drives and database name.

13. If the output drive is different from the input drive, change the input drive to be the same as the output drive. This is so you can view your new database. Also change the database

name to **TEST2**. For example, if the output drive was DSK2. , make the input drive DSK2. as well. This will force FirstBase to look in DSK2. for a database to read.

14. Select option **1 - Query database** from the main menu.

15. Again, you see your original query command. Press <fctn-7> to clear the input field.

16. Press <fctn-9> to default to the query command of ALL. This will display all records in the database.

17. Enter a destination of **S** for screen to display the records.

18. View the records of the newly create database, pressing <enter> after each one.

19. Back at the main menu, select **Quit FirstBase**.

20. End of lesson 3.

Lesson 4 - Defining a Report

This lesson will teach you how to define a report; a report must be defined before you can actually print a it. The DEFINE program comes shipped on disk 3, but it must be accessed through the BROWSE program on disk 1.

1. Insert the program and data disks in their appropriate drives (see lesson 1).

2. Load FirstBase as usual (see lesson 1).

3. Insert the disk that contains the REPORT program (disk 3 as shipped) in the program drive.

4. Select option **4 - Extras** from the main menu. This will load in a short program which gives you a secondary menu of choices.

5. Once the Extras program loads, select option **1 - Define Report** from the menu. This will load in the DEFINE program. Once in DEFINE, select menu option **1 - Define Report**.

6. You need to specify a database name before you can define a report. Type in **BCS** for a database name.

7. At the bottom of the screen you see the define prompt. Press **B** to define both the page and record formats.

8. You are now presented with the page format screen. A page format describes the general characteristics of the printed page. Enter the following information for the page format. Press <enter> after each entry.

```
Skip perf: Y                          Page length: 66
Top margin: 5                         Bottom margin: 60
Left margin: 1                        Right margin: 80
Fixed # of recs/page: Y               Max no. of recs/page: 40
Word Wrap: Y

Headers? Y
Header 1: .TEST REPORT
Header 2: <fctn-9>
Footers? Y
Footer: <fctn-9>
Position of page number: BC
Number page 1? (y/n) Y
```

9. After answering the last question, you are brought to the record format screen. The record format describes how each record is to be printed. For the number of print lines, enter **2**. This means that for every record, two lines will be used on the paper.

10. Enter the following information:

```
Fieldname or Text (F/T/Q)? F
Fieldname: Disk Number
Row: 1     Column: 1
Size: 3
```

The F means that this is a *field print item*. It indicates that you want to print the contents of a field. The field you want to print is the *Disk Number* field. You want it printed on row 1, column 1. The row number indicates the row for the current record. Each time a new record prints, the row is reset to 1. Row 1 means print on the first row; row 2 means print on the second row. The *Size* tells FirstBase how much of the field you want printed. In this case, you are printing 3 characters.

11. The cursor jumps back to the *Fieldname or Text* prompt. Enter the following:

```
Fieldname or Text (F/T/Q)? F
Fieldname: Disk Name
Row: 1     Column: 5
Size: 42
```

```
Fieldname or Text (F/T/Q)? F
Fieldname: Author
Row: 1     Column: 50
Size: 30
```

12. All of the print items so far have gone on row 1. Now, for row two, we want just a dashed line, to serve as a record separator. Type in the following print item:

```
Fieldname or Text (F/T/Q)? T
Text: (type in 80 dashes here)
Row: 2     Column: 1
```

For the *Text* prompt, don't type in the words shown; type in 80 dashes.

The program automatically enters a size of 80 for this print item.

13. Back at the *Fieldname or Text* prompt, type **Q** for quit.

14. Press **Q** again to exit the define prompt.

15. You are now presented with the main prompt (for lack of a better term):

Display, Change, Save, Load, Quit?

16. Type **D** to display the definition.

17. Type **B** to display both the record and page formats. If you instead press P (for page) or R (for record) you can look at each format individually.

18. Hit any key to move from screen to screen.

19. Back at the display prompt, type **Q** to get back to the main prompt.

20. Press **S** at the main prompt to save the definition to the input drive.

21. At the save prompt, type **B** to save both the page and record formats. You also have the option of saving either the page or record formats individually, but don't do that for now.

22. For the *Report filename:* prompt, enter **TESTREP**.

23. Press **Q** to get back to the main prompt.

24. Press **Q** again to go back to the main menu.

25. Select **5 - Report Generator** (from the menu) to print the report.

Lesson 5 - Printing a report

(continued from lesson 4)

1. If you are continuing from Lesson 4, go to step 3.

2. Insert the program and data disks in their proper drives, and load FirstBase. Select the EXTRAS option, then the **Report Generator** from the menus that appear.

3. Select option **1 - Print Report** from the main menu.

4. Enter a database name of **BCS** if asked. If the database was in use during this session, you are not asked for the name.

5. Press **A** to print All the records in the database. You can optionally press L if you've used the QUERY program to select the records to print. Since you didn't do that for this lesson, stick with the ALL option.

6. Make sure your printer is set to top of form and is ready to print.

7. For a report name, enter **TESTREP**. This is the name of the report definition that you created and saved in the DEFINE program. (See lesson 4 if you haven't already done this.)

8. The report will now print. To stop the printing at any time, press and hold <fctn-4> until the printer stops and you are returned to the main menu.

9. When finished printing, select **Quit** from the main menu.

That concludes the brief introductory tour of FirstBase. Hopefully you have gained some insight and inspiration to explore the depths of this program on your own. Good luck, and have fun!

APPENDIX A - Error Messages

This appendix lists every error message you will see while using FirstBase. The error messages are categorized by the program in which they occur. Some error messages appear in every program, so they will be covered first. Along with each error message is a short explanation of what it means, along with some possible solutions.

Common error messages.....110

The BROWSE program.....110

The QUERY program.....111

The UPDATE program.....112

The CREATE program.....113

The SORT program.....114

The DEFINE program.....115

The REPORT program.....116

Section 1 - Common Error Messages

Program DSK.FIRSTBASE.XXXX not found.

This error message appears at a main menu, when loading a new program (like BROWSE or QUERY), and also at the FirstBase title screen, when it tries to load BROWSE. It means that FirstBase could not find appropriate program. Hit any key, insert the correct program disk, and make your menu selection again.

NOTE: The program name for BROWSE is MAIN1.

Cannot find/open DSK2.XXXX_I ...hit any key:

This message pops up whenever a program is unable to open a database file. A suffix of _I indicates an index file, _D indicates a dictionary file, and _F indicates a data file.

Possible reasons for the error include it could not open the file, perhaps there was no disk in the drive, or the wrong disk was inserted.

Hit any key, and you will be returned to the main menu. The current task will be aborted. Insert the proper disk and try again.

Section 2 - The BROWSE program

Error writing DSK.FIRSTBASE.PTR_ ...hit any key:

This message may appear when saving the configuration file to disk. This is done through the *Change System Configuration* option on BROWSE's main menu.

The message means that FirstBase could not save the file to disk. Perhaps the drive door was open, there was no disk in the drive, the disk was write-protected, or it could not find the proper disk. It always writes the config file to the disk named FIRSTBASE.

Insert the disk named FIRSTBASE in any drive, hit any key, and re-select the *Change System Configuration* option. The changes you made to the configuration will still be in memory, so just accept each setting and elect to save your settings again.

Record xx not on file...

This error appears when using the Individual command in BROWSE to load in a record using the record number.

The message indicates that you typed in a record number that does not exist. For example, if a database contains records 1 through 30, then record 48 will not be on file.

Hit any key, and re-enter the record number. Typing nothing for a record number brings back the browse prompt.

End of data: hit any key...

This error message appears whenever you try to read past the beginning or end of the database with the **F** and **B** commands from the browse prompt.

Out of range: re-enter

This error message appears only when using the Individual command. It means that the record number you entered was out of the valid range of 1-32767. For example, typing a record number of -5 or 500000 will bring up this error message.

Section 3 - The QUERY program

Bad fieldname: xxx

In a query command, you typed in a fieldname that does not exist. Reselect *Query* from the main menu and fix the query command. You do not need to re-type the command!

Bad equals sign: xxx

You typed in an invalid equals symbol. It must be one of the following: =, <>, <, >, <=, >= . Reselect *Query* from the main menu and fix the query command.

No data found: xxx

FirstBase could not get or understand that data you supplied for the query command. The data comes after the equals symbol, and must be a string constant, a numeric constant, or a valid indexed

fieldname. Another possibility is that the query command was incomplete. Reselect *Query* from the main menu and fix the query command.

Invalid logic operator: xxx

You typed in an invalid logic operator. The two valid ones are AND and OR. They connect two or more expressions in a query command. The logic operators must be preceded and followed by at least one space. Reselect *Query* from the main menu and fix the query command.

No records found...hit any key:

FirstBase could not find any records using the query command you gave. Reselect *Query* from the main menu and fix the query command.

Section 4 - The UPDATE program

The same error messages for the query command shown in section 3 apply here as well.

Invalid fieldname: xxx

The fieldname you typed in for the update command was not a valid, indexed fieldname. This is the fieldname that comes before the equals sign, and is sometimes called the *destination field* because the results of the update are stored in this field. Reselect *Update* from the main menu and fix the query command.

Missing equals sign: xxx

An equals sign MUST FOLLOW the destination field in the update command. Reselect *Update* from the main menu and fix the update command.

Must be a number: xxx

This message only appears if you have a string constant as part of a mathematical calculation in the update command. You cannot perform math on words! Reselect *Update* from the main menu and fix the update command.

Can't store string in a numeric field: xxx

You cannot store a string constant in a numeric field, only numbers. Reselect *Update* from the main menu and fix the update command.

Bad math operator: xxx

Valid math operators are: +, -, *, / . This error means that it did not find one of those symbols in the calculation. Perhaps you forgot one. Reselect *Update* from the main menu and fix the update command.

Section 5 - The CREATE program

Blank fieldnames are not allowed!

When defining, adding, or changing fields, you cannot enter a blank fieldname.

This fieldname already exists!

When defining, adding, or changing fields, you cannot have duplicate fieldnames.

Error! That field does not exist.

When changing a field, you must enter an existing fieldname.

File already exists; overwrite?

When saving a newly created database definition, give it a unique name. If you save a definition on top of an existing database, all of the data stored there will be forever gone!

Too large! Must be no greater than xxx.

This occurs when adding, defining, or changing a field length. The error indicates that the field length you just entered would cause the total record size to exceed 3000 bytes.

This also occurs when adding, defining, or changing an index length. This time, it indicates that the index size you just

length. This time, it indicates that the index size you just entered would cause the total index size to exceed 246.

You've run out of room for fields.

This appears when entering a field length. It indicates that you have reached the maximum record size (3000 bytes) and you cannot fit the current field in your database. Press <enter> to abort the current field definition.

Ooops! Must be between 1 and 720.

This occurs when adding, defining, or changing a field length. It indicates that you typed in a field length that is out of range; it must be no less than 1 and no greater than 720 bytes.

No more fields can be indexed.

You have run out of index space. The largest an index can be is 246 bytes, and your current indexed fields fill up that maximum perfectly.

Ooops! Must be between 1 and 246.

This means you have entered an index length that is out of range; it must be no less than 1 and no greater than 246 bytes.

This field does not exist.

This appears when you try to delete a field that does not exist. You can only delete fields that you have previously defined!

Section 6 - The SORT program

Invalid fieldnames, hit any key:

One or more fieldnames you typed in to use as keys was not a valid, indexed fieldname. Hit any key and you are taken back to the main menu where you can re-select *Sort Database* and make changes to the fieldnames you entered.

Error: DB does not exist

The database you want to sort does not exist; perhaps you have the wrong disk in the drive, or you made a mistake in typing the database name.

File already exists; overwrite?

This message appears if, when you choose to generate a new database, the db name you provide already exists. You have the chance to go back to the main menu by press **N** and change the database name. If you press **Y** to overwrite the database, all of the information in that database will be lost.

Section 7 - The DEFINE program

75 items reached...

You can define up to 75 print items for a single report. FirstBase Warns you with this message if you try to define more than 75 items.

Record format not defined!

You receive this message any time you try to list, change, or save the record format if it hasn't been defined yet.

Page format not defined!

You receive this message any time you try to list, change, or save the page format if it has not been defined yet.

Cannot open DSK3.TEMP_R...
c99 exit - rerun? (y/n)

When the DEFINE program first loads, it must open a temporary file called TEMP_R in the output drive. If the program is unable to open the file, it immediately aborts with this message. Reasons for the error may be there is no disk in the output drive, or the output drive (as defined in the system config file) does not exist in your system. If you simply need to insert a disk, do so and then press **Y** to run the program again. If you need to change the system configuration file, press **N** at this error message to quit FirstBase. Run FirstBase again and go through the BROWSE program to modify the output drive.

APPENDIX B - FirstBase File Formats

This section of the manual is intended only for programmers interested in writing their own routines to access information stored in FirstBase databases.

Each FirstBase database (db) file consists of three separate files. These are the dictionary, the database, and the index. The db has a name of 1 to 8 characters and a two character extension consisting of an underscore followed by a D (dictionary), F (data), or I (index). The dictionary contains all field definition information together with information concerning the current size of the db. The database contains all the data for each record entered. The index file is the key to accessing the db. It contains the indexed fields for each record, a pointer to the corresponding record in the database file, and a flag indicating whether the record is deleted or active.

The Dictionary

The Dictionary is stored in a DIS/FIX 80 file. The first record contains an ASCII 0 with a trailing null. The next record contains the number of the last record in the db as an ASCII string terminated with a null. The following record is the number of dictionary entires for this database, again represented as an ASCII string with a trailing null.

Following these three records are the actual dictionary records. The first 20 bytes are the field name padded with spaces if necessary. The next 2 bytes are a word indicating the size of the field (note this is not an ASCII number). The next byte is the flag byte with only 3 bits currently in use. The other bits should be set to zero for future compatibility. If set the 1 bit indicates an indexed field, the 2 bit indicates a fixed field, and the 4 bit indicates a numeric field. Following the flag byte are two bytes indicating the size of the index for this field. If there are no indexed bytes for this field, these bytes should be zero. Assembly programmers be warned: this integer field begins on an odd address.

After the dictionary entries is a final record, current containing an ASCII 0 followed by a null.

The Data

The data itself is stored as a Dis/Fix 128 file. Since a FirstBase record may be up to 3000 characters long, clearly a single record may be split across multiple physical records. FirstBase calculates the minimum number of physical records that can hold a single logical records by consulting the dictionary - your application must do the same. The data is stored as packed

fields, with no separators. This means that a field can be split across a physical record. Thus, if you plan to manipulate records, the entire record should be loaded into RAM, modified, and then written back out.

The Index

The index is stored as a Dis/Fix format file. The record length is determined at the time the database is created. It will be a minimum of 8 bytes is required, up to 254 is possible. In general the record length will be selected such that an integral number of records can be stored per sector.

The index is the key to accessing the db. When the db is sorted, only the index is changed, the db is left unmodified. To scan the db in order you must scan the index in order. The first 6 bytes (bytes 0 to 5) of each index record contain a number in the form of ASCII terminated string which indicates the number of the corresponding record in the db. This number starts counting at 1, and records are numbered starting at 0 so the number should be decremented before being used to access the disk. Note that the user is always shown record numbers starting with 1 not 0, so you must be careful when presenting record numbers to the user. Also note, this is not a physical record number, but a logical record number. If it 5 physical records are required to hold a single logical record for a particular database, it is your applications responsibility to multiply the logical record number by 5 to determine the correct physical record number. Following the record number is a single byte indicating if the record is deleted or not. This byte is either a 'Y' or 'N' - 'Y' means that the record is deleted. Following the deleted flag is the packed index data beginning at byte 6.

Addendum for FirstBase Manual

10/3/88

FirstBase now handles the configuration file a bit differently than described in the manual. Please read the following explanation about the configuration file and remember this information when reading the manual.

FirstBase now only loads the configuration file *once per session*, when displaying the title screen. Once loaded and the Browse program appears, the configuration information (data drives, program drives, wildcard symbols, and printer device) is saved in VDP RAM. This means that when you move to a different program using the menus, all of the configuration data is preserved, and the config file is never loaded again. This makes operation of the program much smoother than loading the config file every time you move to a new program.

Another result of this new system is that any changes you make to the configuration information remain in effect for the entire session. For example, if you change the input drive in the SORT program, it will remain changed when you move to the BROWSE program, or any other program. The manual states that any changes made to the data/program drive(s) are temporary, and are in effect only for the program in which they were changed. This is no longer true! Changes you make anywhere will remain in effect until you quit FirstBase. Moving from program to program does not reset the configuration settings.

If you want to make permanent changes to the configuration file, you still need to use the **Change System Configuration** option in the Browse program; this is the only way you can save your changes to disk. Any other changes you make, in any program, only remain in effect until you completely leave FirstBase.

FirstBase Utility Programs

Table of Contents

Introduction....1
PR-Base Importer....1
IBM Importer....2
IBM Exporter....4
ReIndexer....4
File Names....6
PC-Transfer Conversions....6

Introduction

FirstBase is shipped with several utility programs designed to allow data transfer between various database programs. In addition, the ReIndexer program can also be used to change some characteristics of a database, even after it has been created and records entered. The utility programs are run separately from the FirstBase system. They are contained on the third disk of the FirstBase set which also contains the sample database files. The programs may be run from Editor/Assembler Option 5, TI-Writer Option 3, or Funnelweb.

Converting data from another database program into FirstBase is a two step procedure. First you must run the converter program to import the data, and then you run the ReIndexer program to build an index for the imported data. To convert data from a PR-Base disk, you would first run the PR-Base converter and then ReIndexer. Similarly, to import data from an IBM (or Macintosh) database file, you would run the IBM Converter and then ReIndexer.

PR-Base Importer

The PR-Base Importer requires two disk drives to operate. This is a consequence of the PR-Base disk format.

To use the PR-Base Importer, first load it as described above. A title screen will be displayed. At this time put the PR-Base disk in one drive and a disk for the new FirstBase file in another drive. A RAM disk or Hard Disk may be used for output drives. Press any key to pass the title screen.

The first prompt is

```
Press A for V2.0 or B for V2.1
```

If you are using the original version of PR-Base Version 2 by William Warren, press A. If you are using the version modified by Mike Dodd to run on the 9640, press B.

Next you are prompted

```
PR-Base Drive #
```

Enter the number of the disk drive that contains the PR-Base data disk, such as 1. The program now gives you the chance to change this data by prompting

```
Correct? Y/N
```

at the bottom of the screen. Press N to reenter the data, or Y to continue. If you press Y the program will check the disk for a valid PR-Base header. If one is not found, you will be returned to the first prompt to reenter the data.

The program displays the name of the PR-Base database that you selected. You are then prompted for the name of the FirstBase database to create. This name must meet the same requirements for a FirstBase database name as described in the FirstBase User Manual. A valid name would be DSK2.RECIPES, for example. You are again prompted

```
Correct? Y/N
```

If the database name is not what you wanted, press N and you can reenter it. Otherwise press Y and the program will continue to the field naming screen.

The screen format you designed for the PR-Base database is displayed (any graphics characters will either appear as garbage, or not at all). Each field is highlighted by filling it with boxes, and at the top of the screen you are prompted:

```
Name this field:
```

Because of the way PR-Base stores its data, there is no way for the converter program to know what the names of the fields are, just their position on the screen. At this prompt, enter the name you want to use for this field, up to 20 characters long.

After entering the field name you are prompted

```
String or Number:
```

If the field is a Text field, press S for String. If it is a numeric field, press N for number. After entering the field name and type you are again prompted

```
Correct? Y/N
```

If the information displayed is correct, press Y and you will continue on to the next field until all fields are complete. If the data is incorrect, press N to reenter it.

When you have finished naming all fields the converted automatically goes to work building FirstBase Database and Dictionary files. If an error occurs during the process (usually due to a disk error or lack of disk space), you will be alerted to the error and the conversion will stop. If the conversion is successful, you will be prompted

```
Conversion Complete. Hit any key.
```

Press any key and you will be returned to the main TI title screen. At this time you should run the ReIndexer program to complete the conversion process so that the database can be used by FirstBase.

IBM Importer

The IBM Importer program allows you to import data from any database program on the IBM PC or Macintosh which can save files in what is called "delimited format." This is a standard database exchange format, and almost every database program that takes itself seriously provides a way to handle this format. Unfortunately, like many so called "standards" this format has many variations. The IBM Importer program has been successfully tested with files from dBase III on the IBM, FileMaker II on the Macintosh, and several others.

To use the program, you must first get the delimited database file onto a TI disk. This can be done using an XModem file transfer, or using the PC-Transfer conversion routine described below. Once the file is on a TI disk, run the IBM Importer program to convert the file to FirstBase format.

The IBM Importer program cannot assign real names to the database fields it converts. This is because the delimited database format contains only data, no field names. Because of this, the database fields are assigned default names by number such as

```
FIELD 0001
```

and so on. When you run the ReIndexer program, these field names may be changed to something more meaningful. If you have previously imported a database file of the same format to FirstBase, you are given the option of using the dictionary file from that database. This saves you the hassle of having to rename the fields each time. This feature is explained further below.

When the program starts up, a title screen is presented. Press any key to continue on to the first prompt which is

```
Filename of PC database:
```

At this prompt enter the name of the file that contains the IBM database file. If this file cannot be opened (it should be a Dis/Fix 128 file on disk) or does not exist, you will be returned to this prompt.

You are now asked some questions about the format of the IBM database file. First you are asked

```
Enter field separator:
```

Usually imported databases use commas as the field separator, so usually you will want to enter a comma here. Other characters are sometimes used, so check the manual of the IBM program you are using to be sure. Sometimes the fields are separated with tabs. Since there is no easy way to enter a tab character from the TI keyboard, a substitute character is used. To specify that the fields are separated with tabs, use the tilde (~) character.

Next you are prompted

```
Enter string delimiter:
```

Many times the IBM database indicates that a field is a string by surrounding it with a special character. This character is usually the double quotes ("). Some programs (notably dBase) allow this character to be changed. The IBM Importer program converts any fields surrounded by the string delimiter character to strings, and any others to numbers. These can be changed later using the ReIndexer program.

Once you have completed these two prompts, the program scans the entire IBM database file to determine maximum field sizes. It also checks the database file for consistencies. If it finds that the file does not contain a valid database, you are warned with the message

```
Inconsistent database contents
```

and the conversion process is aborted. This warning should only occur when the database does not contain the same number of fields in each record.

When the program has successfully complete the scan of the IBM database file, you are prompted for the name of the FirstBase file to create. This filename must meet the rules for FirstBase database files described in the Users Manual. A valid filename might be, DSK2.IBMDATA, for example. The prompt for the FirstBase filename looks like

```
Name of new FirstBase file:
```

After entering the name of the new FirstBase database, you are prompted

```
Use existing FirstBase dictionary?
```

If you have previously converted this database to FirstBase format or have converted a database of the same format, enter Y, otherwise press N. If you press Y, you must supply the filename of an existing FirstBase database in response to the prompt

```
Name of existing FirstBase file:
```

The IBM Importer program will then open the dictionary and read it into memory. If the dictionary of the existing database is radically different from the information the program obtained from its scan of the file, you will be warned

```
Invalid Dictionary
```

and the use of that dictionary will not be allowed. If the dictionary is similar to but not identical to the scanned information you will be warned

```
xxx differences in dictionaries.
Dictionary not identical. Still use?
```

In general it is OK to use non-identical dictionaries. At worst, some fields may be clipped as they are imported. If there are many differences, you may not want to use the existing dictionary.

After finishing with the dictionary issues, the actual conversion process is begun. First a dictionary is written out, and then the data is converted from IBM to FirstBase format. You should make sure that there is at least as much free space on the output drive as there are sectors in the IBM file, or you could run out of disk space. In many cases the size of the FirstBase file will be dramatically larger than the IBM file, so provide as much storage as possible.

After the conversion is complete, you are prompted

```
Conversion Complete.
```

Press any key to return to the TI title screen. At this point you should run the ReIndexer program to make the database usable by FirstBase.

IBM Exporter

The IBM Exporter program allows you to take a FirstBase database file and convert it into a format that can be handled by most PC and Macintosh database programs. It essentially undoes what the IBM Importer program does.

When the program starts up you are prompted for the

```
FirstBase Database Name:
```

After entering the name of the database you want to convert, its dictionary file is read in. You are then prompted to

```
Enter field separator:
```

The default at this prompt is the comma, which is the most common choice. If you want to use another separator enter that character. Another popular field separator is the tab character which cannot be entered conveniently on the TI keyboard. To get around this problem, the program recognizes the tilde ("~") character as meaning tab.

Next you are prompted to

```
Enter string delimiter:
```

The default at this prompt is the double quotes (") , as this is the most common choice. Programs like dBase allow almost any character to be used a a string delimiter. Probably the only time that you should use something else is when there are double quotes in your database. If you don't want any string delimiter (a very special case) enter the null string at this prompt by pressing Function 3.

Finally you are prompted for the

```
IBM Database Filename:
```

Simply enter the name of the file you want to be created in the IBM delimited database format. The program then goes to work and builds the database. In general this file will be somewhat smaller than the _F file of the associated database.

When the conversion is complete you can either use PC-Transfer or some telecommunications technique to get the file to the IBM or Macintosh for use.

ReIndexer

The ReIndexer program allows you build a new index for any FirstBase database file. This is useful if you have created a database with index characteristics that you later want to change to make better use of the Query capabilities of FirstBase.

The ReIndexer is required to import files, as neither the IBM or PR-Base Converter programs create an index.. The ReIndexer allows you to change field names, and index length. It then rebuilds the index from the database and dictionary files.

ReIndexer can also be useful in attempting to recover a damaged database. The index file is the most likely file to be damaged if something should go wrong. Using ReIndexer, you can quickly regenerate the index file and continue

When the ReIndexer starts up it displays a title screen. Press any key to pass this screen and continue on the the main part of the program. You are first prompted for the name of the FirstBase database you want to reindex. The prompt reads

```
Enter Database name:
```

After entering the filename, the program tries to open the database. If it is unsuccessful you are returned to the same prompt. Otherwise you are prompted

```
Modify index?
```

This option allows you to modify the dictionary characteristics - field names and index lengths. If you choose no, the ReIndexer will run and build the new index based on the existing dictionary. If you choose no and the index size is zero you will be forced into the edit mode anyway. This case will occur whenever you are running ReIndexer after converting a PR-Base or IBM database as these converter programs both set all index lengths to zero.

If you choose yes, you enter into a mode where you can edit the field names and index size for each field. The total index size is displayed on the screen. Along the bottom of the screen, a list of your options is displayed, as follows

```
E)dit N)ext B)ack S)ave Q)uit
```

The information displayed on screen for each record looks like the following

```
Field Name: Disk Number

Field size: 00003

Index size: 00003
```

Press N to display the next field, press B to display the previous field. Press Q to abort the operation and restart the ReIndexer program. When you have the data exactly as you want it, press S and the new index will be built. When you press S the Total Index size must be greater than zero (0) and less than 249 or you will remain in edit mode to fix the error. If you press E to edit, you first are positioned on the "Field name" and then on the "Index size." To keep the defaults in either field just press enter. The Total Index size is always updated after each edit. Note that you can never enter an index size greater than the field size. Once all fields are set, press S and the dictionary will be saved to disk and then the new index will be built.

If there is an error during conversion you will be given an error message on the bottom of the screen. Press any key and the program will be restarted.

When the conversion is complete you are prompted

```
Finished. Press any key to restart.
```

Simply press a key and the ReIndexer's title screen will be displayed.

File Names

The following table gives the filenames for each of the conversion utilities. The files are located on FirstBase Disk 3.

PR-Base Importer	`FROMPRB`
IBM Importer	`FROMIBM`
IBM Exporter	`TOIBM`
Reindexer	`REINDEX`

PC-Transfer Conversions

For owners of Genial Computerware's PC-Transfer program, a utility file named PCT/IBMFB is provided. This file allows you to transfer delimited database files between TI and IBM disks by placing the IBM disk in a TI Double Sided/Double Density disk drive. To use the file, run PC--Transfer as you normally would. When prompted for "Conversion Filename" on the first entry screen of PC-Transfer, enter the filename for the IBM converter routine. For example, DSK2.PCT/IBMFB. From this point on, PC-Transfer will convert delimited database files between TI and IBM disks rather than text files. Simply use PC-Transfer in the manner described in its manual.

Once you have the IBM database on a TI disk, you must then run the IBM Converter program, and then the ReIndexer program. Thus using PC-Transfer to import files from an IBM requires a three step procedure. Once you have run through this procedure once or twice you will find it very simple and straight forward.

Addendum for FirstBase v1.1 Manual

10/26/89

This addendum documents changes in the first maintenance upgrade of FirstBase. The version number has been changed to FirstBase v1.1, and consists of bug fixes, a bit of re-working of the configuration system, and some significant enhancements.

Bug Fixes

1. Occasionally, moving from program to program within the FirstBase system caused lockups. This no longer happens; you will be able to access all programs from the menus.

2. The report generator contained several bugs. All of the following bugs have been fixed: losing the last character of a print line; word wrap handled multiple lines incorrectly.

Enhancements

Several major enhancements have been made to FirstBase:

1. QuickSearch Program

   QuickSearch is a method by which you can search a sorted database much faster than an unsorted file. Just as in Browse, you can only search on one field at a time, but can either display the record or send it to the report generator to create a formatted report. QuickSearch uses a binary search algorithm, which means that the database must be in sorted order...adding even one record that is out of order will throw off the entire search.

2. Similarity searching

   Similarity searching is an innovative new way to search for information heretofore unseen in a /4A database manager. Instead of telling the computer exactly what you want to look for, or using wildcards, you instead specify a percentage to determine the accuracy of the search. During a comparison, the program keeps track of how many characters matched, and computes a percent-accuracy based on its findings. FirstBase only deems a record a match if the computed accuracy is equal or greater than the percentage you specify. For example, you can set an accuracy percentage of 80%, which means that only those records that match at least 80% of your search string will be found. Similarity

searching can be done in the BROWSE and QuickSearch programs.

3. Display Dictionary in Browse

A new command has been added to the Browse prompt, as you are viewing a record. Typing a question mark ('?') at the Browse prompt will display the database's dictionary in the same fashion as the CREATE program.

4. Configuration Program

The Browse program no longer contains the "Save Configuration" portion of the program; it now resides in its own little program, and it can be accessed from the main BROWSE menu. This was done in order to make room for the QuickSearch and "Display Dictionary" routines.

5. Universal program loading

Although not visible on any of the main menus, we have added a new menu option to all main menus: pressing 0 allows you to type in the program name to load. The program will ask you to enter a program name; type in the filename of the program you want to load, without the DSKn. prefix. FirstBase automatically uses the program drive (defined in the system configuration) when loading any program. In this way, you are not limited to the choices on the current menu. If you want to move from Browse directly to Report, you need not load the Extras program first; simply hit 0 at the Browse main menu, and type in the program name for Report, ie. REPORT1.

For your convenience, filenames for all the programs are listed below.

Function of Program	Type in this filename
Browse	MAIN1
Create	CREATE1
Config	CONFIG1
Query	QUERY1
Update	UPDATE1
Report	REPORT1
Report define	DEFINE1
QuickSearch	QSEARCH1
Sort	SORT1
Extras	XTRA1

Addendum 2a - Similarity Search

A new and innovative way to search and match information has been introduced in version 1.1 of FirstBase - similarity matching.

Quite simply, instead of juggling wildcard characters to explicitly depict how you want information to appear in a search string, you just provide a regular string representing approximately what you're looking for. You also provide a percent-accuracy number -- a number that tells FirstBase how close you want to come. For example, a 95% accuracy means that you only want those records that match 95% of your search string. A 100% value means you are only interested in perfect matches. On the other hand, a 30% accuracy value will find a lot of records, since only 30% of the strings have to match. The percent-accuracy value is entered only once, and is used for all matches. At any time you can change the accuracy, which will effect all future searches.

Similarity searches are available in two areas of FirstBase: the QuickSearch program, and the Browse program. Detailed instructions on using it in the QuickSearch program are provided in the new QuickSearch chapter of the manual, included with the version 1.1 update.

Using similarity searches in BROWSE

Switching from standard, wildcard searching to similarity searching is easy to do in Browse, and can be done one of two ways.

Main Menu

The main menu of Browse has been redone; option 3 is now "Set Search Method." Once you choose this option, you can select (W)ildcard or (S)imilarity searching. Press W or S to make your choice. If (S)imilarity was chose, you must also enter a percent-accuracy value. Type a number from 1 to 100. A default of 75% will be accepted if you just press <enter>.

Browse Prompt

Once you have a database loaded and are sitting at the browse prompt, you can press M (for Method) to obtain the above-described selections. You can change the method and percent-accuracy.

The letter 'M' is not displayed in the Browse prompt because there simply was not enough room on the line to include it! The command, however, is indeed available.

Use of Wildcards

With similarity searching activated, you <u>cannot</u> use wildcards in your strings; they will be treated as standard characters, and no special action will take place. This is because similarity searching is provided as an alternative to wildcard searching; allowing you to use both would destroy the purpose of similarity searches.

Addendum 2b - Config Program

Due to additions made to the Browse program, the configuration portion has been separated into its own program. It's called CONFIG1 and CONFIG2 on Disk 1 (as shipped), and it can be loaded from the main Browse menu.

The program looks and behaves exactly like any other FirstBase program. The first menu option is "Change System Configuration", and works exactly as before. You can make changes to all of the configuration information, including input/output drives, program drive, printer name, and wildcard characters. After the last item, you can elect to save the changes to disk. Doing so will make the changes permanent; otherwise, they will only be in effect for the current FirstBase session.

Chapter 12 - The QuickSearch program

FirstBase offers many different ways to search for information: a simple field search in the BROWSE program, as well as a comprehensive multi-field search in the QUERY program. You now have a third alternative: QuickSearch, which employs a very fast binary search method.

A binary search is much more intelligent than other methods used in FirstBase, but with a price: the database records must be sorted. A binary search operates on the fact that it is much quicker to find information if you know how the information is stored. What if your phone book were not organized alphabetically by last name? You would have to read through every name, starting from the beginning. That is exactly what BROWSE and QUERY do when looking for information. If your database has been sorted by at least one field, you can use the QuickSearch program to find records in a fraction of the time.

Records found by the program can be send to the screen or the report generator for printing.

The QuickSearch program can be accessed only from the EXTRAS program menu. Both programs are located on disk 3 as shipped. The files it loads are QSEARCH1, QSEARCH2, and QSEARCH3.

Introduction.....127

Quick Search.....129
Normal.....129
Similarity.....129

Set Search Method.....130

Summary.....132

Section 1 - Introduction to the QuickSearch program

Purpose: QuickSearch facilitates the rapid searching for information in a database - much faster than by the conventional methods used in the Browse and Query programs. Instead of a sequential searching method, QuickSearch relies on the fact that a sorted file is much easier to look through than an unsorted file. Therefore, a database must be sorted in order to use this program.

The main menu for QuickSearch appears as follows:

```
=============== QuickSearch ============

              1 = Quick Search
              2 = Set search method
              3 = Change Drives
              4 = Extras
              5 = Browse
              6 = Query
              7 = Update
              8 = Quit FirstBase

     DSK2.
     ======================================
```

Breifly, the menu choices are:

Quick Search:

This enters the heart of the program, where you can search for information by typing in a search string, and sending the records found to the screen or the report generator.

Set search method:

There are two methods of finding matching strings in QuickSearch: Normal, where you merely typing in an exact string, and Similarity, where you specify a percentage reflecting the desired accuracy of the matching.

Important Note:

Wildcard matching cannot be done in QuickSearch, due to the use of a binary search algorithm. The two techniques are simply not compatible on a theoretical and practical level. Using a single and/or multiple wildcard character will do nothing; the program will consider those symbols to be part

of the search string, and will attempt to match those symbols.

Change Drives:

This option lets you change the default input, output, and program drives, as well as the database name. Please note that changes made here will remain in effect throughout the current FirstBase session, but the settings will not be saved to disk. Use the CONFIG program (accessed from the main Browse menu) to make any changes permanent.

Extras, Browse, Query, Update:

These options load in the appropriate programs from disk.

Quit FirstBase:

This menu selection is used to end the current FirstBase session, and returns you to the TI title screen.

Section 2 - Quick Search
(menu selection 1)

Upon selecting this option, you are asked for a database name if one is not already in use. Type in a valid database name (8 characters, no spaces or quotes), and you are presented with the following messages.

```
Search Method: Normal

Enter a fieldname to search. Do NOT
     enclose fieldnames in quotes!
     Spaces are allowed.

Fieldname:_
```

You can only search on one field in this program; that is due to the way a binary search works. In addition, if a fieldname contains spaces, don't use quotation marks to enclose the fieldname; type it in just as it appears.

The top of the screen indicates which search-string method is in effect, normal or similarity. The search method can be changed by selecting main menu option 2, "Set search method." An explanation of each method follows.

Normal:

This is a straight, "perfect match" string search. The program attempts to find all records which match exactly with the search string. No wildcards are allowed.

Similarity:

This is a completely new way to search for information in FirstBase. Again, all you type is a string, wildcards not allowed. The program tries to find a string that approximately matches your search string; exactly how much of the strings match depends on the percent-accuracy value. The higher the percentage, the more accurate the match must be. The percentage must be set through main menu option 2, "Set search method."

NOTE: Both the Normal and Similarity matching methods operate on individual words in a given field, not the entire field. When you type in a search string, the program will attempt to find a match anywhere in the field in question.

Next on the screen, you see:

Enter search string for FIELDNAME:

The filename you specified for the search will be displayed in the message; type in a search string (remember, no wildcards allowed!). The next prompt asks for a destination.

Destination: Screen, Report?

The destination is the place where the matching records will be sent; Screen means the records will merely be dipslayed on the screen. Report means that FirstBase will generate a report record list, which is used by the Report Generator. This procedure is exactly the same as if you were performing a Query, and decided to send the resultant records to the report generator to print a formatted report. Please refer to the Query and Report Generator chapters for more information.

If Screen was selected, the disk drive will run, and the program will attempt to find your records. If any are found, they will be displayed.

If Report was chosen, the program will ask for a record list filename; type in a valid filename. Then the program will look for the records you want, and store their record numbers in this file. After the program is done, you can move to the Report Generator and print a report of the records found; just provide the same filename you typed here when asked for a record list filename.

Section 3 - Set Search Method
(menu selection 2)

This option is used to change the method of searching for strings, and if using the Similarity method, to change the percent-accuracy.

The prompts appear as follows:

Search Method: Normal

(N)ormal or (S)imilarity? N

In this case, the current search method is "Normal", and the cursor sits on top of the N, the default value. If you want to leave it alone, just press <enter>. Type an S if you would like to change to the similarity search.

If you typed 'S', you are asked the following question:

Enter accuracy percentage: 75

The cursor sits atop the number 75, which is the default accuracy percentage. This means that if at least 75% of the search string matches a string in the database, it will be considered a match and the record will be processed. You can change the value to anything between 1 and 100.

Section 4 - Summary

- The QuickSearch program searches a sorted database, yielding especially fast search times compared to the conventional, sequential searches used elsewhere in FirstBase.

- The database MUST BE SORTED in order for this program to function!

- The "Normal" search-string matching method requires that the search string match perfectly with the data in a record.

- The "Similarity" search method compares strings on a percentage basis; the higher the percentage, the more accurate the match must be. Lower percentages will find more records, while higher percentages will find less data.