

AFM

Automatic File Manager User's Guide

**for the TRS-80™ I/DD, III
or 4 in III mode.**

by Tim Miller

**PowerSoft Products
Dallas, Texas**

Published by
PowerSoft Products
17060 Dallas Pkwy, Suite 114
Dallas, TX 75248
U.S.A.

TRS-80 & TRSDOS are registered trademarks of Tandy Corp.

Automatic File Manager User's Guide

Copyright 1985 Breeze/QSD, Inc. All Rights Reserved. Printed in the United States of America. No part of this software package, disks and documentation may be reproduced in whole or in part by any means, including, but not limited to, the use of electronic, electromagnetic, xerographic, or optical reproduction devices, or network or BBS information and retrieval systems, without the express written consent of Breeze/QSD, Inc. Unauthorized reproduction of this software is a violation of United States Copyright laws and may subject the violator to civil penalties and/or criminal prosecution. End users are permitted to make copies of this software for backup and/or archival purposes only. Use of this software package is limited to the original purchaser only.

Preface

AFM is an entirely new concept in database programs for the TRS-80 computers. No other database program comes even CLOSE to what this program can do in terms of flexibility. Many database programs force the user to conform to their way of doing things; AFM, on the other hand, operates much in the same way a person managing a file cabinet would operate.

AFM is a free-form system. You don't have to restrict yourself to entering only certain kinds of data items because that's all the program can recognize. People who have worked with typical database managers (called DBMS) will tell you of the agony they go through when, for example, they need to enter two telephone numbers for a particular person in their database, but the program can only accept one! Or the problems that arise when they need to enter a European address into a system designed for US postal addresses. The horror stories are endless. But AFM doesn't have any of those problems! Need to enter a second, or even a third, telephone number? Just pick a blank spot on the screen and type it in! Problem with a European address which won't fit? Move things around until it does fit! Need to add a comment line? Pick a spot and start typing! AFM will take up to 4096 characters. AFM lets you do all of these things without complaint -- but try them on a regular DBMS and watch the system choke.

AFM is so flexible that you can even have an infinite number of different screen formats. You don't have to define a single format and then stick to it for everything. Most regular DBMS require that so they can sort properly. AFM doesn't care, and sorts everything on the fly. Your file is ALWAYS sorted, and on EVERY item, not just on one. You can search on every item, not just the first. And you can have duplicate items. Have you ever seen any database program which allowed something like this:

Name: John Smith
Phone: 555-0212 ext 49 (work)
Phone: 565-0399 (field station)

You can repeat an item as many times as necessary, with different data for each occurrence, and **STILL** be able to search on that item.

You might think this is more than enough, but AFM also includes facilities for transferring data to and from other programs such as word processors, etc. It can write an auxiliary file which can be loaded into a word processor, or read from an ASCII file prepared by a word processor, and add the contents of those files into its own database.

AFM is designed to work as closely as possible to the way you do things. In fact, the closest analogy to AFM's method of operation is that of a human clerk placing folders in a filing cabinet. An AFM record is a "folder"; it can hold various types of information, notes, comments, etc. Just as a person might stick notes into a folder at random, so AFM allows you to add information in a free-form fashion, without restricting you to a particular format. An AFM field is the "tab" on the file folder for easy identification; AFM allows up to 26 electronic tabs on each folder! You can even specify which of nine possible electronic file cabinet "drawers" a folder goes into.

Flexible AFM commands allow you to define your printer's parameters, and allow you to select which data items are printed. You can set a column width for each data item printed, and, if the data is numeric, you can request AFM to maintain a total to be printed after the last item. Full columnar reports, including relational look ups and form letters are available.

Finally, AFM is flexible enough so that it imposes very few restrictions on the type of information you can keep in it. You can maintain something as simple as a list of recipes or something as complex as a parts inventory with multiple items of information.

Contents

Getting Started

Introduction	1
About this manual	7
Installing and starting AFM	8

Tutorial (mailing list)

Defining the database	10
Adding data to the database	12
Managing the database	13

Using AFM

AFM basics	15
Utility	18
Defining a database	20
Adding data to a database	23

Managing a database

Data inquiry	28
Full folder inquiry and modification	28
Search masks	31
Auxiliary relational functions	33
Reports	34

Advanced topics

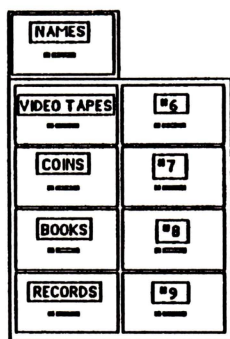
Math functions	39
Report Headers	40
Report form letters	40
Report look ups	41
Data transfers	43
"Help" modification	45

<i>Appendix A - Sample database (inventory)</i>	47
--	-----------

<i>Appendix B - AFM commands</i>	49
---	-----------

<i>Appendix C - Messages</i>	65
-------------------------------------	-----------

<i>Appendix D - Differences from special Beta version</i>	69
--	-----------



(examples)

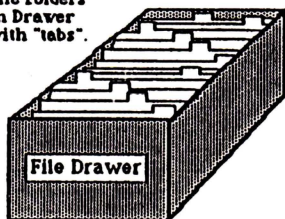
Think of AFM as having up to NINE file drawers, that are there for your filing purposes.

Each DRAWER will contain "data-records". These are the names, tapes, etc. that you will be entering. Think of these "records" as "File Folders" to be filed in the drawers.



File Folder, containing "DATA ITEMS", or better known as... INFORMATION. This is what YOU type in.

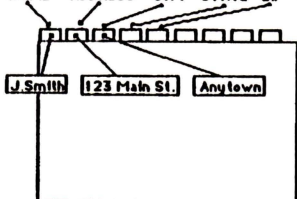
File Folders in Drawer with "tabs".



Each "file-folder" can be identified by "TABS". You may use 1 tab, or up to 26 electronic tabs on each folder!

You name the tabs, so it will be easier to call the data you want at a later time.

NAME ADDRESS CITY STATE ZIP ETC.



You could call up info from this folder based on whatever "tabs" you have set. This is just an example.

Getting started

Introduction

What exactly is a "database"? It is a fancy name for an organized collection of information. The box of recipes in your kitchen is a database. The telephone book is a database. That list of movies in your videotape library is a database. The public library is a database. Your little black book is a database.

The key words in the definition are "organized collection." A random collection of items, facts, etc. cannot really be called a database. Organization allows you to retrieve any piece of information from the collection in a quick and simple manner (well, theoretically, anyway). Your box of recipes might be organized by type of meal. The phone book is organized alphabetically. The books in the library are coded according to an arcane system called Dewey Decimal, whose secrets are known only to the prim young ladies behind the desk. But they are organized, and anyone who knows the method of organization can retrieve any piece of information (recipe, name, book) directly without having to sift through everything.

Several levels of organization can be imposed on a collection of information to make retrieval faster. For example, the Yellow Pages are arranged by category, and within each category, the entries are sorted alphabetically. This is a rough example of a two-level method of organization. Sometimes there are parallel methods of organization. For example, a library's card index might be organized by book title and also by author. All of these methods are there to permit you to undertake the most efficient search strategy for locating what you want.

When a database is small, the time required to extract a certain piece of information is insignificant. I can pull the recipe for fried zucchini out of my recipe box in a few seconds. But when a database grows large,

the time required to find something grows rapidly. This is where computers come in.

Computers are fast. If you can tell them exactly what to do, they can do it faster than you or anybody ever could. It is this speed which makes them ideal tools for searching and maintaining large databases.

But you have to tell them exactly what to do. And that's not a trivial task. To make the task easier, people wrote programs called Database Management Systems, or DBMS. DBMS programs take most of the pain out of organizing and maintaining a database. They allow you to easily add new items to the database, and have built into it the necessary search strategies which allow fast retrieval of any item of information in the database. A discussion of these strategies is the subject of a two-semester graduate level computer course, so we'll skip over them for now.

However, we can say that a DBMS program imposes organization on the database by literally forcing the information into a format that the program can deal with. Their files are organized in a certain way, and so the data must fit into that organization in order to be properly dealt with. For example, a simple DBMS program might require that the "last name" slot in a mailing list be no more than 12 characters in length. So if you have someone with a last name like "Petrosampertini," you've got yourself a problem. Or (and this is a common occurrence), your database program might allow space only for one telephone number, and you find yourself needing to enter two.

What this all means is, you must make the data fit the requirements of the DBMS program.

What the DBMS program gives you in return is speed. It can search a mailing list consisting of literally tens of thousands of names and retrieve a particular one in a few seconds.

A DBMS program typically organizes information in records. A record is a collection of related data items. For example, John Smith's name, address, and telephone number constitute a group of related data items which would be bundled together in a record. Another example would be the title of a book, its author, and publication data. Each data item (title, author, publication data) is placed together with the rest. Using this scheme, the DBMS can retrieve information one record at a time, instead of one data item at a time. So if you ask for "John Smith," you automatically get his address, telephone number, etc. along with his name.

A limitation imposed by many DBMS programs, especially those written to run in the limited memory environment of a microcomputer, is that each record must be the same size as all the rest. The length of each record is often defined by the author of the DBMS to be the amount of data which can be saved in a single disk sector. For TRS-80 computers, this usually means 256 characters of information. Doing it this way simplifies disk file handling. All the program has to do is read one sector from disk, and it has one complete record on hand. Whereas if the size of the record was variable, it would have to worry about where each record began and ended, and maybe read additional sectors in order to assemble the complete record for display. The fewer instructions that must be executed to retrieve a record, the faster the program runs.

But it isn't the most efficient use of disk space. If the data items in a record are less than 256 characters, for example, the program is still going to save 256 characters, resulting in wasted space. This may not be significant when a database is small, but when a database gets large, then the wasted space becomes a very significant factor.

The other problem is, what if the data items for that record total more than 256? Unfortunately for the user, the answer to this question is, you'd better make the data fit in 256 characters or else. This is what we mean by the program imposing its own organization on the data. It

would be wonderful if in the real world data came in packets of 256 bytes, wouldn't it?

The authors of many DBMS programs for microcomputers insist that 256 characters per record is more than sufficient for most uses. They aren't thinking too clearly on this one. If you wanted to assemble a bibliographic database, for example, 256 characters is woefully inadequate. This is especially so if you want to assemble a database consisting of articles from professional and scientific journals. Take a look at the articles found in the journal Physical Review sometime. Many articles (especially those in the field of subatomic physics) have up to 28 authors listed per article! Try fitting 28 names into 256 bytes and you'll see the problem.

Enter AFM

AFM approaches these problems in a unique way. First off, it does not impose any structure or organization on the data items. Data items can be of any length up to 4096 characters, and can be arranged in any manner, within each record. Data items may be repeated in each record if necessary. Secondly, AFM saves only as many characters of information as are required by the record. If Record Number 1 is 200 characters, it will save 200. If Record number 2 happens to be 2000 characters long, it will save 2000. If Record number 3 is 30 characters, then it will save only 30 for that record. There is no wasted space in an AFM data-file.

AFM takes away almost all of the pain of maintaining a computerized database (it can't take away the pain suffered by two-finger typists, however). Adding new records is quick and simple. You don't have to worry about organizing your information beforehand to fit the program's idea of what constitutes a record. Retrieving records is just as fast and just as simple. Editing a record and changing its contents involve typing in new information and erasing old information without worrying about where the new information goes. AFM is a DBMS

system designed to be flexible and easy to learn and use by just about anyone.

AFM's method of operation is similar to that of a person using a filing cabinet to store bits and pieces of information. The first level of organization consists of the "drawers" of the filing cabinet. One drawer may hold names and addresses of various people, another drawer may hold information about a coin collection, a third drawer may hold information about a business inventory. As you will see shortly, AFM allows you to define up to 9 drawers of information.

Think of AFM's records as being analogous to "folders." A folder can hold many bits and pieces of paper. You can add or remove notes from a file folder. In the same way you can add or remove data items from an AFM record.

If we were to stuff our file folders at random into a file cabinet drawer, the contents of that drawer would not be very organized. In fact what most people do is attach little "tabs" to each folder, which stick out with their little pieces of information so a particular folder can be found quickly. With tabs in place, then the folders can be organized more logically. For example, suppose our folders held personnel records for the employees in a company. We could attach little tabs to the folders which held the person's last name, and then we could arrange the folders alphabetically. Doing this would make it much easier to find a particular person.

We aren't restricted to putting just one tab on each folder. We could put in another once, say with that person's social security number. But there immediately arises a problem. Either we sort our folders by last name, or we sort them by social security number, but we CANNOT do both at the same time.

AFM has no such limitation.

You can attach up to 26 electronic tabs on each AFM folder, each with its own piece of information, and keep all 26 sorted at the same time. So we could search through our AFM database by last name and pull out each person's record in alphabetical order, or search by social security number, and pull out each record in numeric order. AFM maintains its sort internally, and we never have to worry about misplacing a record (how many times have you misplaced a folder from a filing cabinet?).

If you keep this filing cabinet analogy in mind, it will help you understand the operations of AFM as we describe how to input and retrieve data from the AFM's database.

About this manual

This manual is designed to be read from front to back, although you may skip sections that do not interest you. **Appendix A** contains information about the sample database. **Appendix B** is a list of AFM commands, which can be used as a reference guide. **Appendix C** lists messages produced by AFM. **Appendix D** lists changed features from the special Beta version.

The notation **<KEY>** refers to the key labeled **KEY**. **<CTRL>** means the combination of **<SHIFT>** and **<DOWN ARROW>**.

Some examples may show the command line prompt as **+Afr1;**, while your prompt is **+Afm1;**, or vice-versa. This is no cause for concern, and will be explained later.

Installing and starting AFM

The AFM system requires either a TRS-80 Model III (48K) or Model 4/4P (III) (64K) and two or more disk drives, or a Model I with 48K, lower case, double density LDOS, and two or more disk drives.

To use AFM on a floppy disk system, prepare a system disk using your favorite disk operating system. Kill all unnecessary files from this disk. Also prepare a formatted data disk. Use the COPY (or CONVERT if TRSDOS 1.3) command to move the files from the AFM single density distribution disk to your disk. Then remove the distribution disk and insert your data disk. COPY AFMDAT1/DAT to the data disk and kill the copy on the system disk. The program can then be started by entering AFM (typing AFM and pressing <ENTER>) from DOS Ready.

To use AFM on a hard disk system, use COPY to move the files from the AFM single density distribution disk to your disk. The program can then be started by entering AFM from DOS Ready.

Advanced DOS users: If you can allocate space to a file with a command like CREATE, you should set aside as much space as possible for the AFMDATx/DAT file before copying the distribution file to it. For example, to setup an LDOS data disk, use CREATE AFMDAT1/DAT:1 (SIZE=150), then COPY AFMDAT1/DAT:x :1 - this will result in best disk i/o speed when starting to fill up a new file.

Since AFM is a system which allows storage of a large amount of information, it is suggested that careful backup methods be followed. In addition to having a backup of the program for your protection, you should backup your data often. A common method is to have three sets of backups, called A, B, and C. The A and B backups are recent backups, on a rotating basis. This means that the first time, you backup to A, then to B, then to A again. Backup C is made less frequently, on a weekly or monthly schedule, and the disks might be kept in another

location. The reason for Backup C is that you might make a bad backup two times in a row onto A and B, but the slightly older C Backup is still intact.

AFM Specifications

MAXIMUM 26 VOCABULARY WORDS - 60 TOTAL TAB NAMES

MAXIMUM 4096 CHARACTERS PER FOLDER

MAXIMUM NUMBER OF FOLDERS LIMITED TO DOS FILE SIZE

Tutorial (mailing list)

The following is a short tutorial designed to show some of the features of AFM. We will set up a simple mailing list, add a few folders, list them, and change one.

It is suggested that this tutorial be done with a backup copy of the system and data disks. When ready, start the program by entering AFM from DOS Ready. You will need to press <ENTER> after a short time.

After a short time, you should see a mostly blank screen, with the cursor flashing after the following command line.

+Afr1: OK!

Defining the database

Enter T at the command line. The following screen will appear.

1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-

+Afr1: T

The cursor will be to the right of the first line. Use the left arrow key to move the cursor over the "K" of Key and type Name:- and then press the space bar three times so that the line looks like 1Name:- followed by spaces. Use the arrow keys and change the screen to look like the following

1Name:-
1Address:-
1City:-
1State:-
1Zip:-
1Phone:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-
1Key tab:-

+Afr1: T

The colon and dash are mandatory. Now press the <BREAK> key. The screen should clear and the cursor will be flashing on the command line. The mailing list has now been defined using the T - Tab command.

Adding data to the database

Now it is time to add a few names into the database. Enter I (don't forget to press <ENTER>), and make the screen look like this

Name: PowerSoft
Address: 17060 Dallas Pkwy, Suite 114
City: Dallas
State: TX
Zip: 75248-1921
Phone: (214) 733-4475

+Afr1: I

Each colon must be followed by a space. When the screen matches, press the <BREAK> key. The vocabulary words (tabs) should blink - if not, enter E to Edit the screen and press <BREAK> when correct. <BREAK> can be repeated as often as desired.

Now enter S to save the folder to the disk. The data on the screen will clear and the vocabulary words should remain. The cursor should be flashing. Enter two more folders, pressing <ENTER> after each item. Don't forget to press <BREAK> and S each folders. After the final S, press <BREAK> again to return to the command line. A total of three folders should be added into the database.

Managing the database

Now we will look at the information entered. To view the tabs, enter L NAME. A columnar listing should be displayed.

To bring up an entire folder, enter NAME.

The options at the bottom of the screen will be explained in detail later. For now, use <N>o to skip this folder. The next item displayed is one we wish to edit, so type <E>.

Make a small change in some of the information (not a tab name), and press the <BREAK> key. The new information is now stored on the disk.

There are many more features available in AFM, but for now we will end the tutorial and require you to read to learn more. Type QUIT to exit AFM. You will not normally want to use the tutorial disk anymore (it should be a backup), because it has a limited definition and test data.

Using AFM

AFM basics

After starting AFM as already described, an automatic command might be executed. This is not the case with the distribution disk. The command is entered with ***AUTO *command***, where *command* is a valid AFM command. To disable such a command, use ***AUTO AFM 1**.

Help is available when using AFM. Simply enter **HELP** or **H** from the command line and you will be presented with a list of topics. Entering **HELP *topic*** or **H *topic*** will display one screen of help for that topic. After you are finished with the help, press **<BREAK>** to restore your original clipboard.

To exit AFM, enter **QUIT** from the command line. You should not exit by simply removing disks or pressing the **RESET** button - this may cause loss of data. If you want to immediately exit because no changes were made, enter **ABORT**.

Since some information will remain in memory until the **QUIT** command is executed, it is suggested that whenever you have entered more information than you would want to re-enter, you should **QUIT** and start AFM again. A power glitch could require a reconstruct with some loss of data, usually that data entered since the last **QUIT** command.

Almost every operation in AFM uses the keyboard editor, except the command line. Many keys have important special functions within the editor. A few of the keys are not active in the tab name editor.

Pressing **<BREAK>** exits the editor (it may be necessary to type a command to actually save the changes). To exit from **<E>** on the option menu without changes or deleting, press **<CTRL> B**.

Pressing <SHIFT> <CLEAR> erases to the end of the clipboard.

The arrow keys move the cursor non-destructively in the direction of the arrow (up, down, left, or right). The combination of <SHIFT> and the arrow keys causes different action. <SHIFT> <LEFT-ARROW> performs a backspace with delete. <SHIFT> <RIGHT-ARROW> moves the cursor to the start of the next line. <SHIFT> <UP-ARROW> moves the cursor up 14 lines, and <SHIFT> <DOWN-ARROW> Z moves the cursor down 14 lines. Note that the up and down arrows can be used from the command line to scroll the clipboard.

<CTRL> B breaks out of the editor, clearing the screen and jumping to the command line. It is normally used only to exit from the <E> option without changes.

<CTRL> D is the delete function. It deletes the character at the current cursor position. Other results vary slightly, depending on the current mode.

<CTRL> E enters (adds) a new line to the clipboard.

<CTRL> F enters the Field mode. In this mode, insert and delete affect the current field only. The current mode type is indicated in the lower right corner of the screen.

<CTRL> L enters the Line mode. In this mode, insert and delete affect the current line only.

<CTRL> R removes a line from the clipboard.

<CTRL> S is an on/off toggle for the insert mode. Press <CTRL> S once to start inserting, and press <CTRL> S again when done.

<CTRL> T enters the Text mode. In this mode, insert and delete affect the entire clipboard.

<SHIFT> @ positions the cursor directly beneath the start of the next word on the current line and duplicates that word at the new position. This is normally used to enter the same tab name repeatedly in a folder.

Pressing **@** positions the cursor directly beneath the start of the next word on the current line. This is normally used to enter columnar data.

Utility

The Automatic File Utility is started by entering AFU from either the AFM command line (TRSDOS & LDOS only), or the DOS command line. Before performing a major AFU function, it is advisable to make a backup. After a prompt appears on the screen, ensure that all data disks are in the drives and press <ENTER>. After pressing <ENTER> again, the AFU screen shows the following

Data-File Status

Last Sector = 00000
Afmdat1/Dat = 00000
Afmdat2/Dat = 65535
Afmdat3/Dat = 65535
Afmdat4/Dat = 65535

Display

Reading = 00000
Writing = 00000

Auto-File Utilities

Menu

<0> = Quit to DOS
<1> = Reconstruct
<2> = Expand
<3> = Remove
<4> = Recover
<5> = Return to AFM

Press Your Selection

To quit AFU, press 0 or 5. The later is normally selected only if you entered AFU from the AFM command line. It may not function on systems other than TRSDOS/LDOS, in such a case you should quit AFU and enter AFM from DOS Ready.

Press 1 to reconstruct the data file. The reconstruct feature of AFU must be run if you get a "Data error" message from AFM. It is also used to reclaim the space made available by deleted folders. In AFM, deletions only mark a folder as deleted for speed purposes. The

reconstruct feature should be run periodically after deletions or changes are made, in order to best utilize the space available.

Press **2** to expand. The expand feature of AFU is used when additional space is available for storage. You would probably not want to expand your data file to the next disk until the first disk had been filled close to capacity. You will be prompted for the drive number on which the new file should be created. Any new items added to the database beyond the last record threshold value are placed in the new data file.

Press **3** to remove. The remove feature of AFU is used when additional space is no longer desired. The data file is not killed, but any data in it is lost and cannot be normally brought back. After running this option, it is necessary to run the reconstruct option.

Press **4** to recover folders marked for deletion. In AFM, deletions only mark a folder as deleted for speed purposes. This feature will "undo" all deletions since the last reconstruct. Note that it is possible for duplicate folders to be created with this process.

Defining a database

The first thing you want to do when creating a new database is to tell AFM what you are going to be putting into the database file. Is it going to be names and addresses? A parts inventory? A videotape library index? In a typical DBMS this procedure would be called "Defining field names." In AFM we call it "setting up the tab names or vocabulary." The term vocabulary is used because the key tab names you enter via the T - Tab command actually become part of AFM's command vocabulary -- that is, the name becomes a valid command to AFM to do something.

Enter the command T at the command prompt. The screen will display the current tab names assigned, and at the same time the cursor will be placed at the top of the screen to the right of the first name. You are now in AFM's editor. The display will look something like:

```
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-  
1Key tab:-
```

```
+Afr1: T
```


There are a total of sixty possible tab names (the first 26 of them key tabs), but only the first 14 are shown. You can see the rest of them by using the <DOWN ARROW>.

What exactly does a tab name consist of? Let's take a closer look at one.

1Key tab:-

The first character is the digit "1." As you might guess, this is the drawer number of our filing cabinet. Every tab name that has a 1 on the left will be placed in drawer number one. You select the drawer number with the AFM command. Selecting a drawer causes only folders in that drawer to be accessed. You might use AFM 1 to access a mailing list and AFM 2 to access an inventory. Remember the limit of sixty tab names.

Next is the phrase "Key tab:" This is the default name assigned in a fresh, unused AFM database file. Think of it as the name of the tab sticking out of the side of that folder. We can change it to anything we want, for example, "Last Name:" or "Widget:" or "Contract Number:" etc. It can be up to 62 characters long, but cannot begin with a space.

Note that the tab name ends with a colon. This colon is mandatory. It is there to differentiate the tab name from similar names.

The last character is a dash. It is there to simply mark the end of the tab name so that AFM would have some way of knowing when to stop reading a particular name on the list. The dash is mandatory, since the following additional information can follow a tab name.

Auxiliary relational functions can follow the tab name (after the colon). The functions available are described later. They may also be entered using the SET command, with the syntax SET 1tab name *relational function*.

Text may follow the tab name (after the colon and dash), and is used as an automatic entry for any folder saved which includes that tab name. For example,

1City:-Dallas

would automatically fill in a tab name defined as City: with Dallas during the save. Any existing data will be overwritten. This information can be entered using the SET command, though it will be converted to lower case.

Although the T command shows all sixty tab names, only the first 26 are Key tabs. They can be defined by using the K, T, or V command. Key tabs are the vocabulary names which can be accessed from the command line. For this reason, it is desirable to start defining tab names which will not be accessed from the command line after the first 26.

Adding data to a database

The blank screen you are looking at right now is the AFM clipboard. It's where we assemble data before sticking it into our folder. The clipboard (and folder) will accommodate 64 lines of 64 characters each. Only 14 lines at a time are shown. Using the <DOWN ARROW> key we can scroll the screen down to see the rest of the clipboard. If the clipboard is empty, this doesn't tell us much.

To start entering information into the clipboard, enter I. The cursor will be transferred to the top of the screen, and you will be in the editor. The special keys for this mode were described earlier.

For now, just use the arrow keys to position the cursor somewhere on the screen, type in a tab name (don't forget the colon), a space, and the data for that tab name. A sample screen was created in the tutorial.

You can use any format you want, and you do not have to use the same format for every folder. The only rules governing entry of data into AFM are the following:

1. When entering a tab name into the clipboard, don't enter the drawer number. Enter a space instead.
2. Do not use a reserved word for a tab name. Reserved words are the AFM commands.
3. Since AFM will retrieve data only until it encounters two or more spaces, don't type two spaces in a row in the middle of a data item. Otherwise that data item may not be retrievable in its entirety.

Since every folder in AFM may have its own screen format, it does not hurt anything to leave off an unused tab name in a particular folder, since you can plug it in whenever you need it.

Last Name: Smith First Name: John
Address: 123 Main Street
City: Newcastle State: PA Zip: 17821

Phone: 555-1212

+Afm1: I

When you have finished entering the folder and are satisfied with the format, press <BREAK>. You will exit the editor and the cursor will return to the command line at the bottom of the screen. In addition, the tab names will blink once. This blinking is used as a check. If a tab name does NOT blink, then it does not match what you set up in the tab name list, and AFM is not recognizing it as a valid tab name. The most likely culprit is misspelling, so you may want to go back in and change it (use the E - Edit) command).

At this point, the folder is still on the clipboard. It has NOT YET been saved as a folder on disk.

To save the folder to disk, enter S. AFM will write the folder out to disk and automatically invoke the I command (unless *NOINPUT is set or <ENTER> was held down). The I command will erase the data items from the screen display. However, it will leave the tab names in place and re-enter the text editor automatically, placing the cursor to the right of the first tab in preparation for a new folder. The command *INPUT is used to set this mode, *NOINPUT is used to disable it. The <ENTER> override is always available.

Although AFM doesn't hold you to a standard screen format for data entry, it makes it easy to enter data items into a formatted screen anyway. Isn't that nice?

Here's where <ENTER> can come in really handy. Remember that while in the text editor, <ENTER> takes the cursor to the first occurrence of two or more spaces following a tab name or other piece of data. So as you enter a new folder, pressing <ENTER> after each data item will position the cursor directly after the next tab name in sequence, making it unnecessary to use the arrow keys for positioning.

Let's examine an example.

Last Name: Jones First Name: Indiana
Address: 4500 Doomed Temple Street
City: Lost State: Ark. Zip: 72999

Phone: 555-9913 Phone: 01-124377 (Paris)

This gentleman is never home. Always flying off somewhere...

+Afm1: I

We have some new elements in this folder. First, notice the second occurrence of the tab name "Phone:". AFM allows you to repeat a tab name as many times as necessary within a folder. Following our file folder analogy, what we have done is simply stuck another "phone" tab on the edge of Mr. Jones' folder with the second telephone number on it. This feature is UNIQUE to AFM. Other DBMS don't allow it.

Secondly, we have entered a notation that says, "This gentleman is never home ...". There is no tab name associated with this note (that is, there is no tab stuck on the side of the folder saying "this gentleman is never home"). Any other DBMS program would absolutely choke if you tried something like this, but it is perfectly valid with AFM. It's as though we penciled a little note in the margin of the sheet that held Indiana Jones' name and address and stuck it back in the folder.

So the format has been changed. But what if we want to go BACK to a previous format? Do we have to laboriously type it back into the screen? No, of course not. AFM wouldn't treat you that way! Just press <SHIFT> <CLEAR> to clear the screen. If this is too hard, try entering the command CLS which will do the same thing. The reason you want to clear the screen is because you want a blank clipboard to work with. AFM always adds things to the clipboard, it doesn't write over what is already there.

Now recall a folder using the previous format that you want and type the command I for input. This will remove the data items from the screen and place you back in the text editor with the tab names intact and arranged the way we want.

So how do we retrieve a previous folder? There's a whole section of this manual devoted to that, but for now, simply select a key tab name and type it in at the main command line. For example,

+Afm1: Last Name <ENTER>

(the underlined words are what you would type in.)

This immediately brings up a folder to the screen -- in the format that we used when we entered the folder! Any added tab names are gone, and so are any additional notes or comments. Notice there are some choices on the bottom of the screen. Ignore them for now and just press <BREAK>. AFM will respond with a message. Now if we enter the I

(for Input) command, the data items disappear from the screen, leaving only the tab names, and we are returned to the text editor. Now we're ready to enter a new folder using an old format.

You can understand this better by thinking of the screen formats as a particular type of (whisper it!) government form. We used one form for John Smith, another form for Indiana Jones. Now we need to use the same form as we used before for someone else. So we grab the pad that we used for it, ripped off the top sheet which was already filled out, and got ourselves a fresh blank form. We said before that AFM operates very closely to the way you would normally, didn't we? If you try to relate what AFM is doing to actions you would perform without a computer, you won't have any trouble understanding what is going on.

Another method of entering data is by using a data entry form. This is defined by using the editor to create a clipboard with only tab names, in the location you want them. The clipboard is then saved using the SL command, followed by a name. All current names can be displayed by typing LIB, and a form can be retrieved by LIB followed by a name. The KL command, followed by a name, is used to remove any undesired forms, or to edit one.

Managing a database

Data inquiry

To show information from many folders without displaying each full folder, use the L - LIST command. For example,

L NAME

will display all of the unique items associated with the tab name "name" in columnar format. Items will only be listed up to the first occurrence of two or more spaces.

You are allowed to specify a starting and ending point with the L command. The range is specified using ">" to mean greater than or equal to, and "<" to mean less than or equal to., For example,

L CITY>B<G

will list cities whose first letter is greater than or equal to B and less than or equal to G (notice that there are no spaces anywhere after the data).

Duplicates are not listed with the L - LIST command. If you desire to see duplicates, use the command LD instead.

The width of the columns used can be specified with *C #, where # is a number normally set to 32. The maximum is 255.

Full folder inquiry and modification

AFM allows you to retrieve any folder based on any tab name - either by scanning through the entire file one folder at a time or directly. To retrieve a folder, first press <SHIFT> <CLEAR> to erase whatever is in

the clipboard. Remember that AFM adds data to whatever is in the clipboard area, it does not replace what is already there.

Next, on the command line enter in the key tab name (vocabulary word) you wish to search on. For example,

+Afr1: Name <ENTER>

This will search every folder which has a tab name of Name. You can also abbreviate the tab name to the first three letters, as long as those letters are unique - NAM would have worked as well. When you use this form of retrieval, folders will be retrieved in partially sorted order (by the first letter only). This limitation is inherent in the design of AFM.

If AFM finds a name tab, it will copy the folder to the clipboard and display it. At this point, several choices will appear on the bottom of the screen:

<Y>es <N>o <O>ut <E>dit <D>elete <C>lear <T>ext

These choices describe what you can do with the displayed folder, and except for the first two, are pretty self-explanatory. To understand the first two choices, look at the folder on the screen and ask yourself the question, "Is this what I want?"

If the folder is NOT what you want, you should press N. The folder will be erased from the screen and the search will resume.

If the folder IS what you want, you can either press Y or <BREAK>. Pressing <BREAK> terminates the search, pressing Y leaves the folder in the clipboard and resumes the search. If another folder is found and Y is pressed, the folder will be added below the information already in the clipboard. This is how you can assemble a group of folders for display and examination.

If the 64-line clipboard area fills during retrieval, you will get the message,

Clipboard full <O>ut <C>lear

AFM will not load partial folders into the clipboard - a folder must fit in its entirety.

Pressing <BREAK> terminates the search. If you wish to continue, you must either <C>lear the clipboard, or clear it by using <O>ut to the printer or a file.

When no remaining folders match, a message Searched = xxxxx Found = xxxxx appears. The first number displayed includes folders marked for deletion.

Now for options other than <Y>es and <N>o.

<O>ut allows you to output the entire contents of the clipboard to the printer or a file. The DF table must be reset to get full output, otherwise the output is controlled by the DF table. The clipboard is then erased and the search resumes.

<E>dit places you in the text editor, with the cursor positioned at the top of the folder. It is suggested that you do not edit a folder unless the clipboard was clear before bringing it up. Use the editor to make any changes, and press <BREAK> to save them or <CTRL> B to ignore them. The search resumes after changes are recorded.

<D>elete will delete the displayed folder from the database, clear the clipboard, and resume the search. AFM only marks folders for deletion - the AFU - Utility must be used to actually reclaim the space.

<C>lear will erase the clipboard of any contents and resume the search.

<T>ext will output text through the FORMS filter as defined with the DT command. The syntax of the DT command is DT [tab name...]. Specifying DT with no parameters clears the table, otherwise the tab names specified are added to the current table. A common use would be to output selected fields in their entirety without pre-defining the length of the field. All selected fields will be output through the forms filter with word wrap enabled and a single carriage return.

If you wish to repeat an option directly from the command line, use the R (RUN) command. The syntax is R *O key tab*, where *O*, is the first letter of the desired option. Because of the danger in specifying R D key tab, a warning message is displayed.

Search masks

So far, we have been pulling up every folder with a particular tab name. A search mask is available. The search mask is a string of characters which will be used to compare against the contents of a tab. If we want to find "Jones," we give the command

+Afr1: Name Jones <ENTER>

Note that NAM Jones would have worked as well. If AFM finds a name tab with "Jones" on it, it will copy the folder to the clipboard and display it, and present the options menu described above. Further specification is possible after the entire first word, and additional specification need not be complete. For example, NAM Jones M might retrieve all Dr. Jones' (MD or M.D.).

The space after the tab name is equivalent to specifying an equal sign ("="). In other words, our earlier example is the same as NAM=Jones. The equal sign is mandatory when searching for a number. There are several other ways to retrieve folders.

You can also specify starting and ending points for a search, as with the L - LIST command. For example,

NAME>Jones<Oberding

Another method of retrieval is the instring or "I don't know how to spell it" type search specified using the apostrophe (') in place of the space between the tab name and the search mask. For example,

NAME'ngh

would retrieve a folder with a name like neighbor, in case you can't remember the "I before E" rule (it would also retrieve a folder with a name like night). The specifier causes a search for matches with the specified letters, in that order, and allows other letters to exist between. To speed up this type of search, specify the first letter of the data with a ">" - for example, **NAME >N'ngh**.

If you desire folders which match on one condition OR another, use the asterisk (*) specifier. For example,

NAME=JONES*=SMITH

would find all folders with the name of Jones or Smith. Note that the asterisk must be followed by another specifier, and the asterisk should be avoided in data.

In an example above, we used an example of the implied AND specifier. Multiple conditions are allowed, though it should be noted that ">" and "<" refer to starting and ending positions, and any other conditions which are outside the range are ignored.

Yet another way to retrieve data is the NOT specifier, the exclamation (!) character. This retrieves all folders NOT matching the condition. For example,

NAME!=Jones

would retrieve all folders with a name other than Jones. The NOT (!) specifier can only appear once in a search mask, at the end. For example,

NAME>S<S!=SMITH

This would retrieve all folders with a name starting with the letter S, but not equal to Smith.

The last resort method of retrieving is *SCAN. It is normally needed if you enter a folder without any tab name. The *SCAN command will bring up the options menu for all folders starting with the first entered and continuing to the most recent (other than HELP, LIB, or REP screens). The normal option is either <N>o to skip or <E>dit to add a tab name. The <Y>es option will be inoperative.

Auxiliary relational functions

It was noted before that auxiliary relational functions can be entered following the dash of a tab name using the K, T, or V commands. They can also be entered using the SET command. These functions are only operational when the plus sign shows on the far left of the command line (*R=Y) - they can be disabled with *R=N, which causes the minus sign to show on the command line.

When entered in this manner, the ">" and "<" do not specify starting and ending points. Items inside the range will always be displayed, while data outside the range must meet any other conditions.

If the functions are operational, any retrieval by entering a key tab name will also have to meet any conditions specified.

Reports

Simple reports can be created with the commands above, and setting

F (FORMS) O=P

for output to the printer. Other reports are usually created using a report generator file, which is actually a list of AFM commands executed using the DO command.

To make report generation easier, AFM includes a "boiler plate" report generator file which can be copied and modified for a specific report. To access this file, type REP FORMS. The file will be copied to the clipboard for editing. Pressing <BREAK> and using the SR command (followed by a name) causes it to be saved. The file should look like the following:

'f c=64'

'f l=60'

'f p=66'

'f m=8'

'f o=p'

'f j=0'

'f #=0'

'f w=n'

'f h=n'

'f s=s'

'f i=n'

'f f=n'

'dm'

'dmt'

'dms'

'df'

'dd'

'dl'

'dt'

'df 16,1key tab'

'? r o key tab'

We'll explain each line from this boiler plate. The most advanced topics will be covered in the next section. Many of the commands are F (FORMS) commands, and each command is described in Appendix A of this manual.

The first line sets the number of characters per line. The example of 64 is a common setting for printouts, allowing an 80 column printer with margins on each side. It is preferable for this setting to equal the total width of the DF table, if utilized.

The next line sets the desired number of printed lines on a page. The example of 60 is a common setting for printouts, leaving a margin of 1/2 inch at the top and bottom.

The next line sets the number of lines per page. The example of 66 is a common setting for printouts, for using 11 inch paper with six lines per inch.

The next line sets the size of the left margin, in characters. The example of 8 when used with 10-pitch results in a margin of .8 inches.

The next line selects the type of output desired. The normal setting is P, for printer, but can also be set to S for screen or F for file.

The next line sets the number of lines to skip between printing each folder. The setting of 0 produces a normal report, a setting of T produces a report with one folder per page. Settings of 1-255 skip the specified number of lines.

The next line specifies whether pages are to be numbered. Specifying 0 disables page numbering, anything else specifies the starting page number (maximum is 255).

The next line specifies whether or not word wrap is desired. Specifying Y causes AFM to only print complete words on a line rather than parts on each line.

The next line specifies whether or not a header is desired. This is set to N for no header. Headers are discussed in the next section.

The next line specifies the sort type. A setting of S specifies the standard AFM sort - by first character. This is the fastest, but not necessarily the best. A setting of A specifies alphabetic - a common sort, and N specifies numeric sort. N may also be used for alphabetic or combinations. As long as you have a little extra time and some disk

space, stick with the N type of sort. It will allow you the most versatility. S and A type sorts require that folders to be output are searched by the same tab as first appears in the DF table. A common question about sorts is how to sort dates. They must be formatted, meaning that months and days must always be two digits. For complete date sorts, the format must be yy/mm/dd instead of m/d/yy.

The next line specifies whether or not the indent function is desired. A setting of N causes all folders to be printed as specified. A setting of Y causes an indented format by removing duplicate columnar data.

The next line specifies whether or not the form letter function is desired. Settings other than N for no are discussed in the next section.

The next group of lines resets the math functions. Using math functions are described in the next section.

The next group of lines resets the define tables. This is desired before defining a table.

There is one major table necessary to define. It is done using the DF function. The first DF is required to clear out the table. To use the DF function, follow it with the width you desire to output, a comma, and the tab name you desire (with drawer number). Include all tab names desired (abbreviations are allowed). When the command approaches 64 characters, start a new line, and the information will be added to the table. For example,

DF 20,1Name,30,1Address

This causes the name tab to output up to 20 characters, and the address tab up to 30. Since duplicates are not reported, be sure that enough tabs are specified in the DF table to make each folder unique. This is most important when running math totals.

An optional second table is the DD table. This specifies the order desired, and spaces between each tab. A value of 255 causes a carriage return to appear, and is used to position data on another line. The spaces to skip comes first, then the tab name, for example,

DD 5,1Name,5,1Address

The reason that the order can be different from the DF command is to allow one order for sorting, and another for printing.

Note: Tab names need only be specified to the point of being unique; in the above example, 1Nam and 1Addr would have produced the same result.

The DT command is normally null in reports, but could be used with a R T tab name type of report if desired. The desired fields are defined with the DT command, and field lengths are not specified (the word wrap feature is enabled for output).

The last command normally found in a report file is ? R O key tab. This command actually causes the folders to be searched and output to occur after a pause. Pressing <ENTER> at the pause causes all folders to be output, while entering search data causes only matching folders to be output.

Advanced topics

Math functions

AFM math functions allow totaling numeric fields, including multiplication and division, and are accurate to two decimal places using BCD math.

The first DM functions are necessary to clear the tables. If math is desired, lines can be added before the R O tab name command line. DM is used to define the math, and DMT is used to define the output desired. DMS is similar. Example:

DM \$1=\$1+1Amount
DMT 50,\$1

This example causes one of the thirty system variables (\$1) to be assigned as a running total of the Amount tab. At the end of the report, the total will be printed at the 50th column of the printer. Additional variables or tab names can be specified, each preceded by the number of spaces to skip. Since numbers can range from 99,999,999.99 to -99,999,999.99, fourteen spaces are used for printing each number.

Using DMS causes the subtotal to be printed whenever the first column of data changes. It also uses only system variables.

Other math operations are specified by *, /, +, and -. Tab names are allowed in addition to system variables, only one operator is allowed in any DM command. For example, 1Amount=1Qty*1Price is valid, while 1Amount=1Qty*1Cost+10 is not.

The DDM command can be used to display the current DM table to the clipboard, and can be quite useful if you get confused.

Report headers

Headers for reports are specified by creating a file and specifying its name after the **F H=** command. The file is usually created by setting up your desired tables, using the **DH** command to display a sample header to the clipboard, and then editing the clipboard, opening a file, and using **WC** to write to it. The heading can have carriage returns (specified with "<") or whatever is desired. It will be printed at the top of each page unless the printed lines per page and page size are 0 (continuous output).

Report form letters

Form letters can be created using **AFM** by using the editor to create a file with text and tab names embedded. The tab names must include the drawer number and be preceded by a space or comma, but may be abbreviated. Example:

```
<Dear 1Nam,  
<  
<  How are things in 1City? We hope they are fine.
```

Note the use of "<" to force a new line and its use before a desired indentation. It is converted to a carriage return in the actual file.

When written to a file and that file is specified after the **F F=** command, each folder output will be a form letter. The **F J=T** command is usually desirable when using form letters.

Report look ups

The most sophisticated and powerful report feature involves LU - Look Up. Simply put, this feature allows a report to be run with output to a file, and then run another report while using the first report to look up information, and combine the reports into a final output. An example is included in the sample database. Another example would be a database with streets and names of children on that street in some folders, and bus number and pickup streets in other folders. The look up feature would allow producing a report with appropriate children listed with each bus number.

Before using the LU command, a file must be created (usually by using the R O command, but sometimes with a text editor), and the DL command must be used to specify the format of that file. The table is usually whatever the DF table was set to when the R O command was done.

To create the look up file using R O, it should be OPEN and set F J=0,I=N,O=F,N - then *END when done.

The syntax of the actual LU command is LU 1tab name *filespec* r o key tab=, where *filespec* is the look up file. The first tab name must exist in the DL table, and indicates where the search data exists in the look up file. R O is for a report (R T, R E, or R D could be used), and key tab= is the search. Note that the last character must be a relater (=, <, >, ', etc.).

The actual action taken is to pull one line from the *filespec*, and move any data which is specified in both the DL and DF table into the DF buffer. A search is then made for key tab=*data*, where *data* is the data associated with 1tab name. The folders found are output along with the information currently in memory from the *filespec*.

Using the example of our bus route, assume that folders exist with a bus number, and many street entries. Other folders exist with a street and

many student entries. We first DF 5,1Bus,20,1Street. Then OPEN BUSLU/TXT; F J=0,I=N,O=F,N. R O Bus produces the output, and *END finishes the file. We must now DL 5,1Bus,20,1Street to match the file. We must first reset the table with DF, then use DF 5,1Bus,20,1Street,15,1Student. Finally, LU 1Street BUSLU/TXT R O Street= produces the report. OLU can be used in place of LU for a slightly different appearance of the report.

Following is technical "pseudo code" describing the process of LU and OLU, provided for reference.

- 1) Get a record from the specified filespec. The filespec to be used is specified with the LU and OLU command.
- 2) Put the tab specified on the command line. The LU and OLU command specifies the tab to use as data to search with. The position of the tab is known from the DL table. The width is automatically adjusted to fit on the command line. AFM will search multiple words as long as the first word is full, even though the last one is partial.
- 3) Transfer data from the DL table to the DF table if the DF table is not reset. If the DF table is specified during the look up then the data specified by the DF table that is also specified in the DL table is transferred from the DL to the DF table. Adjustments are made for different widths between tables. During the search, any data that is found in the record recovered according to the DF table will overlay any info that may have been there by the look up transfer. This step is repeated until no further information is available for this search.
- 4) Check the next record in the look up filespec to see if the next tab to search on is the same as the present one. This check is made for relationability.
- 5) If OLU then output DF table.
- 6) If not the same then search database and output.

7) If the same then output DF table.

8) Go to step 1.

Uses for the look up command include searching a list made with a word processor and sorted <T>ext output, in addition to the relational, multi-pass reports.

Data transfers

AFM has the ability to read and write an auxiliary file which is separate and distinct from its database files. The auxiliary file might be a file created by another program, and may be used to input data into AFM itself; conversely, AFM could write data into the auxiliary file for later processing by another applications program like a word processor.

Only one auxiliary file may be open at a time. The auxiliary file remains open until explicitly closed, or until an operation is performed which automatically closes the file.

The first major command associated with the auxiliary file is OPEN. OPEN is used to specify the file for access. The syntax is OPEN *filespec*, where *filespec* is a valid file. If the file did not previously exist, it is created. Issuing the OPEN command will issue an automatic CLOSE for any current auxiliary file.

Note: The commands below operate on the current auxiliary file, which must be OPEN. This is also true for the related <O>ut and <T>ext options if F (FORMS) O=F.

Another major command is CLOSE. This command is automatically issued when using the following commands: QUIT, *END, and OPEN.

To file positioning commands available are BF and EF. Issuing BF sets the file pointer to the beginning of the file (the initial point after OPEN),

while issuing EF sets the file pointer to the end of the file (for appending operations).

Issuing *END writes the string "*END" to the current auxiliary file and does an automatic CLOSE command. This string is required at the end of header, form letter, and look up files, and any file loaded by LC.

The LCA command loads the clipboard in ASCII format, i.e. it begins a new line after each carriage return in the file. For most AFM operations, the regular LC command should be used.

The LC command loads the clipboard from the current auxiliary file, changing carriage returns to "<" symbols. These symbols are converted again with the WC command.

The WC command writes the clipboard to the current auxiliary file, changing "<" symbols to carriage returns. It is normally used for creating header and form letter files.

The BU command is used to create a file which contains AFM tab names and data. It is used as a prefix to an AFM inquiry command, i.e. BU NAME. A file created with BU can be added back to AFM with the AD command. Before using BU, remember to OPEN the file, and CLOSE it afterwards.

The G command gets one line from the current auxiliary file. A line is defined as a string followed by a carriage return. This is normally used only to test a file prepared for the AD command.

The GCR command must be followed by a parameter specifying the number of lines (maximum 255). Lines are added using the format in the clipboard if one exists.

The ADCR command must be followed by a parameter specifying the number of lines (maximum 255). It can be considered as the opposite of

the <T>ext option. Lines are added using the format in the clipboard if one exists (same as the GCR command), and the the folder is saved.

The AD command adds one line from the current auxiliary file, saves the folder, and continues until the End of File (*END) is reached. It is the opposite of the BU - Back Up command.

"Help" modification

If you don't like what the help screens say, or you want to put in new help screens, you can do so very easily.

If you are going to edit a help screen, make sure you clear the clipboard first. If the clipboard is not empty, very strange things could happen.

To change an existing screen, enter EH *topic*. The screen will be copied to the clipboard and deleted from the database. Use the editor to make whatever changes you wish. Just remember that a help screen can only be 14 screen lines long. Everything should be visible - only the current video screen is saved.

Once you finish your screen, press <BREAK>. Now we must re-save the clipboard contents as a help screen. This is a bit different from a regular folder, because help screens aren't normally accessible as regular folders, and vice-versa. Pick a topic name you want this help screen saved under (using the old name is probably best) and enter the command SH *topic* to save the screen.

To add a help screen, use either the E - Edit or I - INPUT command to enter the editor and create a screen, observing the 14 line limit. When you finish, press <BREAK> and use the SH command (don't forget the topic). If the screen is similar to an existing one, CH *topic* can be used to copy help to the clipboard without affecting the original.

To completely remove a help screen, use **KH *topic*** to remove it from the database.

Appendix A - Sample database (inventory)

We have included a sample database on the AFM distribution disk. It includes sample folders, a library of data entry screens, and two reports. It is provided as a sample of a fully defined database.

The tab names are defined at the end of each list, using drawer 8 & 9. Part, Description, and Cost are defined in drawer 8, while Customer, Part, and Price are defined in drawer 9. Data entry forms are Inv and Cust; reports are Inv and Cust. Folders exist for Parts 1-5, and Customers Jones and Smith.

Look at the tab names, and you will find that Cost and Price were defined as non-key tabs. The key tabs are saved for items we want to be able to retrieve from the command line. Be sure to set AFM 8, and look at an inventory item by entering Part 1. Set AFM 9, and an invoice can be called up by entering Cust Jones. A list of parts and the total cost can be produced by entering DO INV, and invoices can be produced using the Look Up feature by entering DO CUST. Inventory and Customers can be added by entering LIB INV or LIB CUST under the appropriate drawer number, and using the I and S commands.

When done with the sample database, it should be erased before you use AFM. It can be erased by entering the following commands (note that this will delete any other data you added with the tutorial or on your own):

Run D Part, <ENTER> to delete folders

CV - Clear Vocabulary

T - Tab names - manually if desired.

KL INV; KL CUST - Kill Library

KR INV; KR CUST - Kill Report

Then run the AFU Reconstruct feature to actually reclaim the space.

Appendix B - AFM commands

*AUTO

This command establishes an automatic command executed upon entry to AFM. It is normally set to the null command, *AUTO AFM 1.

*C - Column width

This command establishes the column width for the L - List command. It is normally set to 32 columns, *C 32. The maximum is 255.

*END

This command writes *END to the current (open) auxiliary file and executes a CLOSE.

*INPUT

This is the normal setting (vs. *NOINPUT). It causes re-entry into the Input mode after an S command. The re-entry can be overridden by holding down <ENTER> instead of just pressing it.

*NOINPUT

This is the setting to override the re-entry into Input mode after an S command.

*R - Relational functions

This command allows the AUXILIARY relational functions (described elsewhere) to be turned on or off. The current state is on if the Afr or Afr on the command line is preceded by "+", and off if preceded by "-". *R=Y turns the functions on, *R=N turns them off.

***SCAN**

This command displays ALL valid folders (other than HELP, LIB, and REP folders) from the oldest to the most recent. It is normally a last ditch effort to access folders entered without a key tab name.

***SEND**

This is a command to output a file from disk through the FORMS filter. It is used by entering the command ***SEND *filespec***, where *filespec* is the desired file.

***SORTSPEC**

This command is used to define the *filespec* used by AFM for storage during sorting. It is used by entering the command ***SORTSPEC *filespec***, where *filespec* is the desired name. Our default is SORTAFM:0.

***USE**

This command can add speed when the same data is used for several reports. A file must first be prepared by using DF width, tab name...; FORMS O=F,N; OPEN *filespec*; RUN O search; *END. Then use the syntax ***USE *filespec* O search**, where *filespec* is the file created above and the other parameters are like RUN.

?

This command is used for pausing/prompting. If followed by a space and text, that text is displayed and can be added to or changed before pressing <ENTER>, at which time the command will be executed. This is usually used in a report generator file.

A (AFM)

This command is an abbreviation. See the AFM command for details.

ABORT

This command causes an immediate exit to DOS, without normally closing files. It is NOT a substitute for the normal QUIT command to properly exit AFM.

AD

This command is used to add from a file created with the BU command, or any similar ASCII file (with or without tab names). It adds one line (ASCII characters followed by carriage return) from the current auxiliary file, saves the folder, and continues until the end of file. Ignore any EOF message which appears - there is no error.

ADCR

This command is used to add from a file created with the <T>ext option, or any similar ASCII file (with or without tab names). It uses the data entry format on the clipboard and adds the specified number of lines, preceding each line with an editor <ENTER>, and a save after each set. The operation continues until the end of the file. The syntax is ADCR #, where # is the number of lines desired, maximum 255.

AFM

This command selects one of nine drawers within the AFM file cabinet. It is used by entering AFM #, where # is a digit from 1-9.

AFU

This command executes the Auto File Utility, explained elsewhere in the manual. It may not function on systems other than TRSDOS/LDOS - in such case you should Quit AFM and enter AFU from DOS Ready.

AS - Add Sector

This command uses the information in the DS table to add sectors to the database until the end of file.

BF - Beginning of File

This command causes the auxiliary file pointer to be set to the beginning.

BU - BackUp

This command creates a file which can be added later with the AD command (ASCII characters followed by a single carriage return). It is used as a prefix to a search command. The normal usage would be OPEN filespec; FORMS O=F,N; BU tab name; CLOSE.

CH - Copy Help

This command copies a help screen to the clipboard - usually used to output a hardcopy. It is followed by the help name.

CLS - CLear Screen

This command performs the same function as <SHIFT> <CLEAR> - it clears the clipboard.

CLOSE

This command closes the current open auxiliary file. The current file is automatically closed when another OPEN command is issued, when the AFU command is issued, or when the QUIT command is issued.

CV - Clear Vocabulary

This command will erase any information (automatic entry or relational functions) after the tab name definitions.

DD - Define Data

This command is used to define output data. It is usually used in a report generator file. The syntax is DD [*width*, tab name...], where *width* is the number of characters to skip between columns. A width of 255 is used to specify that a carriage return is desired between folders. The tab name must include the drawer number and must be in the DF table. Specifying DD with no parameters clears the table, otherwise the information is added to the current table. The maximum number of fields which can be defined is thirty, and the maximum total width allowed is 255.

DDM - Display Defined Math

This command displays the current DM table to the clipboard.

DF - Define Field

This command is used to define output fields. It is usually used in a report generator file. The syntax is DF [*width*, tab name...], where *width* is the number of characters desired, and tab name must include the drawer number. Specifying DF with no parameters clears the table, otherwise the information is added to the current table. The maximum number of fields which can be defined is thirty, and the maximum total width allowed is 255. DF should be reset if you want the <O>utput option to deal with all the tab names and data instead of limited.

DH - Display Header

If you have used the DF command, using the DH command will display a header to the clipboard, which you can then use to create an actual header file.

DL - Define Look up

This command is used to define the look up tabs. The syntax is DL [*width*, tab name...], where *width* is the number of characters desired, and tab name must include the drawer number. Specifying DL with no parameters clears the table, otherwise the information is added to the current table.

DM - Define Math

This command is used to define mathematical operations. It is usually used in a report generator file. The syntax is DM [*variable=expression...*], where *variable* is either a tab name (with drawer number) or a system variable (\$0 - \$30), and *expression* is a variable or constant (number) followed by an operator (*, /, +, -), followed by another variable or constant. Specifying DM with no parameters clears the table, otherwise the information is added to the current table. The maximum number of characters in the table is 240. Numbers can range from 99,999,999.99 to -99,999,999.99 (more precise numbers are truncated).

DMS - Define Math Subtotals

This command is used to define math subtotals. The syntax is the same as DMT.

DMT - Define Math Totals

This command is used to define math totals. The syntax is DMT *skip*, *variable* [*skip,variable...*], where *skip* is the number of characters to skip, and *variable* is a system variable.

DO

This command is used to execute a report generator file. The file is a series of commands issued to AFM (any commands, though usually for reports). Nested DOs are not allowed; however, a DO command can be

the last command in a DO file. The commands must be surrounded by apostrophes ('), and cannot be longer than the command line. Other text in the file is ignored.

DS - Define Sector

This command is used to set up a table for the AS and GS commands. The syntax is DS skip,get... All parameters must fit on one line, and they can end in either a skip or a get. The parameters are numbers, signifying how many characters either ignore or use. The sector operations are usually a last-ditch method to move data to AFM.

DT - Define Text

This command is used to define tabs for output with the <T>ext option. The syntax is DT [tab name...]. Tab name must include the drawer number. Specifying DT with no parameters clears the table, otherwise the information is added to the current table.

E - Edit

This command enters the text editor and may be used to create new data or modify existing data. To leave the editor, press <BREAK>

EF - End of File

This command causes the auxiliary file pointer to be set to the end. It is used to append information to a file.

EH - Edit Help

This command deletes a help screen and copies it to the clipboard. Use the SH command to save it. It is followed by the help name.

F (FORMS)

This command is an abbreviation. See the FORMS command for details.

FORMS

This command is used to set parameters for the FORMS filter. Each time the FORMS command is entered, all FORMS settings are reset to a top of form condition and the filter is enabled if the last parameter is not N (the printer should be checked to ensure it matches). To use the command, enter **FORMS *parameter* [*parameter*]...**, where *parameter* is one of the following

= 0 or starting page

If set to 0, no page numbers will be printed. Otherwise, pages will be numbered starting at the specified value. The maximum is 255.

C = Characters per line

Our default setting is 64. This parameter specifies the number of characters printed on each line before forcing a carriage return. This number includes any left margin.

F = Form letter output

The default setting is N for No form letters. To enable form letters, specify the parameter **F=*filespec***, where *filespec* is the name of the form.

H = Header

The default setting is N for No header. To enable the header, specify the parameter **H=filespec**, where *filespec* is the name of the header. **H=Y** can be used to specify the last used header filespec.

I = Indent

Setting **I=Y** causes special indents to occur in a report. Setting **I=N** forces full output.

J = Jump

This command specifies the number of lines to skip between folders. **J=0** causes no skip, **J=T** skips to top of form, any other value skips that many lines.

L = Lines per page

This parameter specifies the number of printed lines on a page. For most printouts, this should be 60 if a top and bottom margin is desired, 66 otherwise. It must be less than or equal to the P parameter. Setting the parameters equal causes a pause between each page, useful for screen output. Specifying 0 causes this parameter to be ignored.

M = Margin

This parameter specifies the number of characters for the left margin. **M=8** is the default.

N

Specify the parameter **N** to disable the FORMS filter. It must be the last parameter in order to take effect.

O = Output

Specify **O=F** for File output, **O=P** for Printer output, **O=S** for Screen output.

P = Page length

This parameter specifies the total number of lines on a page. For most printouts, this should be 66. It must be equal to or greater than the **L** parameter. Setting the parameters equal causes a pause between each page, useful for screen output.

S = Sort type

Specify **S=A** specifies full Alpha sort, **S=N** specifies formatted Numeric sort, and **S=S** specifies the Standard sort. The full Alpha or Numeric sort is only useful when used with the **DF** command. Hint: If FORMS **O=S**, you should use **S=N**. Actually, **S=N** is desirable for its versatility unless you don't have a little extra time and disk space. Numeric works for alphabetic and combinations, not just numbers.

T - Top

Specify the parameter **T** to send an immediate top of form to the printer (before resetting the filter).

V - Vertical justification

Setting **V=Y** turns on the feature that causes a folder to be kept together. Setting **V=N** allows a folder to be split between two pages. Vertical justification is only in effect when the DF table is reset.

W = Word wrap

Setting **W=Y** turns on word wrap, which does not split words at the end of a line. It also keeps only one space between words, and two at the end of a sentence. Setting **W=N** disables word wrap. Because of the word wrap function, the maximum word length (string of characters without spaces) in a folder should be 128.

Y

Specify the parameter **Y** to enable the FORMS filter. It is automatically enabled when any other parameter is used unless the last parameter specified is **N**.

G - Get

This command loads a line from the auxiliary file (with or without tab name) at the beginning of the line where the cursor is located. It is normally used for testing purposes.

GCR - Get with Carriage Return

This command loads a specified number of lines from the auxiliary file (with or without tab names). The syntax is **GCR #**, where **#** is the number of lines, 255 maximum. It is normally used only for testing purposes.

GS - Get Sector

This command uses the DS table to add one sector to the clipboard.

H (HELP)

This command is an abbreviation. See the HELP command for details.

HELP

This command lists available HELP topics. For specific HELP, enter *HELP topic*, where *topic* is one listed on the display. The clipboard contents are not affected.

I - Input

This command enters the text editor erasing everything except tab names, and may be used to create new folders. To leave the editor, press <BREAK>. The difference between E and I is that the former clears the entire clipboard, while the later leaves any tab names.

K - Key

This command displays the key tab names (first 26) and enters the text editor. It is the same as the V - Vocabulary command.

KH - Kill Help

This command deletes a help screen and places its contents onto the clipboard (this help feature does NOT protect the clipboard). It is followed by the help name.

KL - Kill Library

This command deletes an data entry format and places its contents onto the clipboard. It is followed by the library name. Hint: this is also how to edit, along with the SL command.

KR - Kill Report

This command deletes a report generator file and places its contents onto the clipboard. It is followed by the report name. Hint: this is also how to edit, along with the SR command.

L - List

This command displays information from many folders in columnar format. A starting and ending point can be specified. The syntax is *L tab name* [>start point[<end point]].

LC - Load Clipboard

This command is used to load the clipboard from a file (usually for header files). It stops loading if it encounters *END, and converts carriage returns to "<" symbols.

LCA - Load Clipboard ASCII

This command is used to load the clipboard from an ASCII file. It stops loading if it encounters *END, and does NOT convert carriage returns to "<" symbols.

LD - List with Duplications

This command is the same as L - LIST, but does display duplicates. See the L command for details.

LIB

Entering this command by itself will produce a list of data entry formats. Entering the command followed by a name will load that data entry format onto the clipboard.

LU - Look Up

The syntax is LU 1tab name *filespec* tab name=, where *filespec* is the name of the Look Up file, tab name is the search, and 1tab name is the reference defined in the DL table for the data.

OC - Output Clipboard

This command outputs the contents of the clipboard through the forms filter.

OLU - Outlined Look Up

This command has the same parameters as the LU - Look Up command. Its output is different.

OMT - Output Math Totals

This command outputs a line and defined totals through the forms filter. It is normally used in a report generator file.

OPEN

This command OPENS an auxiliary file for further access. If the file did not previously exist, it is created. The syntax of the command is OPEN *filespec*.

OV - Output Vocabulary

This command outputs all 60 tab names through the forms filter.

PC - Printer Command

This command is used to send printer commands directly to the printer. The commands are specified in decimal, separated by commas. For example, PC 15 places some printers in condensed mode.

QUIT

This is the normal command to exit AFM and return to DOS Ready. For data integrity, DO NOT exit by any other method (removing disks or re-booting).

R (RUN)

This command is an abbreviation. See the RUN command for details.

REP

Entering this command by itself will produce a list of available report generator files. Entering the command followed by a name will load that report generator file onto the clipboard.

RUN

This command is used to repeatedly execute one of the data inquiry options from the command line. It is followed by the option letter - Y, N, O, E, D, C, or T, and then the search information.

S - SAVE

This command causes a folder to be Saved. In almost all cases, using the text editor and pressing <BREAK> affects only the clipboard, and the S command must be used to record the changes to disk.

SET

This command can be used from the command line (or in most cases, a report generator file) to set the information which follows a tab name definition. It must be followed by a tab name and text or a search specification. All text will be converted to lower case.

SH - Save Help

This command is used to save a help screen (the current displayed 14 lines only). It must be followed by a help name.

SL - Save Library

This command is used to save a defined data entry format. It must be followed by a library name. Any comment can be added after an apostrophe (').

SR - Save Report

This command is used to save a defined report. It must be followed by a report name. Any comment can be added after an apostrophe (').

T - Tab

This command displays the current tab names (all 60) and enters the text editor. To leave the editor, press <BREAK>. Any changes are automatically saved.

V - Vocabulary

This command is the same as K - Key tabs. It displays the current Vocabulary list (first 26 tab names) and enters the text editor. To leave the editor, press <BREAK>. Any changes are automatically saved.

WC - Write Clipboard

This command performs similar to OC - Output Clipboard, except that it converts any "<" symbols on the screen to carriage returns, and adds *END to the end of the file. It is normally used to save header files.

Appendix C - Messages

+Afm1:

This is the normal AFM prompt. Three portions of the prompt are variable. If the plus sign (+) is replaced by a minus (-), then the *R option is off. The Afm may be replaced by Afr which usually indicates that you are in data inquiry mode (report mode). Finally, the 1 may be replaced by 2-9, indicating the current file cabinet drawer.

<ENTER> to continue

Press the <ENTER> key to continue using AFM.

<Y>es <N>o <O>ut <E>dit <D>elete <C>lear <T>ext

This message appears when retrieving folders. It lists the options available. See the section on full folder inquiry and modification for details.

After drives stop, please mount data disks

This message appears when starting the Auto File Utility. Ensure that all data disks are in the drives and press <ENTER>.

AFM done

This is the message to indicate that AFM terminated normally.

Clipboard full <O>ut <C>lear

This message indicates that the clipboard is full and must be output or cleared to continue.

Closing files

This is a normal message when Quitting AFM.

Column width too great

The total of the widths specified with the DF command cannot exceed 255.

Command error

The entered command is invalid. This error causes display of the valid vocabulary (first 26 tab names) for the current AFM setting.

DISK I/O ERROR

The DOS error message should be displayed on the screen. Correct the condition and try again.

DASH "-" missing

This message indicates that a name on the K, T, or V screen is not followed by the "-" required.

Data-error

The is an error message. Use AFU and select the reconstruct feature for correction.

Memory is below F800H Please re-boot

Too many relocatable routines are in memory for AFM to execute. Reset your computer using a fresh system disk.

Loading AFM directory

This is a normal startup message.

Missing drawer number

This message indicates that a name on the K, T, or V screen is not preceded by the 1-9 required.

Not found

This is usually a normal message indicating that AFM did not open a file.

OK!

This is a normal message on the command line.

One moment please

This is a normal message indicating a slight delay. It appears when changing from the Afr mode to the Afm mode, and vice-versa.

Opened

This is a normal message indicating that a file has been opened.

Please mount AFMDIR/DAT

This message appears if the AFMDIR/DAT file cannot be found when closing files. Insert the disk with the file in its original drive and press <ENTER> to continue.

Searched = xxxxx Found = xxxxx

This message appears after data inquiry, indicating that the search has completed. The first number represents the total folders searched, including those marked as deleted, and the second number indicates how many matched.

Sort overflow

If this condition occurs, you should shorten the DF table to sort on less tabs, or sort separately.

TAB less than 3 chars

For practical purposes, AFM requires that any name in the K, T, or V screen be at least three characters long.

Too many entries

A DF or DD command allows a maximum of 30 entries.

Variable error

This message may appear if using a database from the special Beta version which has empty data. The <E>dit function will correct the condition.

W A R N I N G <ENTER>

This message appears when you use the R (RUN) command for the <D>elete option. Press <ENTER> to proceed or <BREAK> to abort the operation.

Writing AFM directory

This is a normal message that appears when Quitting AFM.

Writing sort file

This is a normal message that appears when AFM is sorting.

Appendix D - Differences from special Beta version

If you used the special Beta version of AFM and want to use your old data with the new system, and your old HELP screens, you should perform the following three steps.

- 1 - Use the T - Tab names command to enter your old tab names into the new system. Only 26 key tab (vocabulary) names are allowed.
- 2 - Use the COPY command of your DOS to overwrite the new AFMDAT1/DAT file with your old AFMDAT1/DAT file.
- 3 - Enter AFU from DOS Ready and use the reconstruct feature to complete the conversion.

If you want your old data and the new help screens, you might use the old AFM, F=N, OPEN a file, enter your vocabulary word, select <F>ile, press <ENTER> to repeat the operation, and when done enter CLOSE. Then start the new AFM with the distribution data file, define your vocabulary, and OPEN the file. Finally, use the AD command to bring in the folders.

Two other options are to use the new AFM, OPEN a file, and use KH, <BREAK>, WC, CLOSE for each help screen. Then use the new AFM with your old, reconstructed data and help screens (first option), use KH for all the old help, and OPEN, LC, and SH for the screen files you made earlier. The last option is to use your new, reconstructed AFM and help screens (first option), use the OPEN and BU command, then start with the distribution data file again, and use the AD command to bring your folders back in. The first two options should be easier for most.

It is now required that <SHIFT> <CLEAR> always be pressed rather than <CLEAR> alone.

The <P>rint and <F>ile options have been changed to <O>ut. Use the F (FORMS) O parameter with this command.

The EDIT, INPUT, LIST, and SAVE command are only allowed in abbreviated form (E, I, L, and S).

The BOF - Beginning of File command has been changed to BF.

CT is no longer used.

The EOF - End of File command has been changed to EF.

The F (FORMS) command is changed - see FORMS. The S parameter of FORMS has been changed to the J - Jump parameter.

The FB - File Buffer command has been change to OC - Output Clipboard. You will notice that we now refer to the buffer as the clipboard.

FO has been changed to DF - Define Fields.

The PB - Print Buffer command has been changed to OC - Output Clipboard.

The PT - Print Totals command has been changed to OMT - Output Math Totals.

The R - Relational functions command has been changed to *R.

The SCAN command has been changed to *SCAN.

Other changes are either cosmetic or enhancements. Check elsewhere in the manual for all the details.

AFM Demonstration Program

by Tim Miller ©1985 by Breeze/QSD, Inc. - All Rights Reserved

The AFM program supplied on the special DEMO/TUTORIAL diskette is set up specifically to help you get acquainted with the powers and scope of the AFM (*Auto File Manager*) data-base system. The demo takes about an hour and runs through most functions, including reports.

This demo disk is for Model III or 4 systems (in the III mode).

To run the demo, reset your computer with a copy of TRSDOS™ 1.3 in Drive :0 and our AFM DEMO Disk in Drive :1. The DEMO disk is formatted for this system.

Even if you might normally use an operating system other than TRSDOS™ 1.3, please go ahead and boot TRSDOS™ 1.3 for the purpose of running this demo.

To start demo, type: AFM:1 <enter>

When you execute AFM/CMD, control of the program will be handled automatically. Please follow these rules while running the DEMO:

- 1) ONLY do what the program asks you to do. No extra key-strokes.
- 2) Please do ALL that the program asks you to do!
- 3) When asked to press the [BREAK] key, please just tap it as briefly as possible. If pressed for too long, the demonstration could be aborted.
- 4) Please take notes as you go along. There won't be too many.
- 5) If you have trouble with the instructions and want to start over, press the back-space key to erase the current command on the Command Line prompt at the bottom and type, DO DEMO1 [enter] to restart.

The DEMO/TUTORIAL gives you an opportunity at several places to take a break, if you wish to. The program picks up from where you left off automatically. If you exit early or the sequence gets messed up, you can always backspace the cursor on the Command Line and type DO DEMO1 to start again. If you follow the program's prompts, you should have no problems, however.

AFM Registered Owners

The demo disk enclosed has another purpose besides showing itself off. You can update YOUR current AFM/CMD to the latest version with it - *IF you use the following procedure(s) ONLY.*

To update your working copy, first move the AFM/CMD file from the DEMO DISK to YOUR AFM system disk. "How?", you ask. Below are the detailed instructions for various DOSes. You've got a few other steps too, but let's get it on your disk first.

*TRSDOS™ 1.3 USERS

The DEMO DISK is formatted with TRSDOS™ 1.3, so this is easy. Put your current AFM System disk in Drive :0 and the DEMO DISK in Drive :1. Type:

COPY AFM/CMD:1 :0 [ENTER]

***LDOS™ Mod I/III USERS**

Confirm that CONV/CMD is present on your AFM LDOS system disk. Do this by typing DIR CONV(I) on your AFM LDOS System disk in Drive :0. If present, put the DEMO disk in Drive :1 and type:

CONV AFM/CMD:1 :0 [ENTER]

(If using a hard drive, use the proper floppy drive number. If CONV/CMD is not there, COPY it off your MASTER LDOS disk.)

***DOSPLUS 3.5 USERS**

Make sure that CONVERT/CMD is present on your AFM DOSPLUS 3.5 system disk by typing DIR CONV*/CMD,I. If so, put the DEMO disk in Drive :1 and type:

CONVERT AFM/*:1 :0,V13 [ENTER]

(if using a hard drive, use the proper floppy drive number. If CONVERT/CMD is not there, COPY it off your MASTER DOSPLUS 3.5 disk.)

*NEWDOS/80 USERS

Set your PDRIVE to read TRSDOS™ 1.3 and COPY the file over.

Don't forget to reset your PDRIVE.

STEP 2. - IMPORTANT!! <<<<<<<<<<<<<<<<<<

After getting the new **AFM/CMD** over to your System disk, type:

AFM [ENTER]

Then type these commands *EXACTLY as listed* and in this order.

Make sure your DATA disk is on-line!

*AUTO=AFM 1[ENTER] *Don't forget the SPACE AND the "1"!

*SORTSPEC=Sortfile/AFM [ENTER]

***INPUT [ENTER]**

*C=32 [ENTER]

***That's it! Do NOT copy the sample DATA files over!
Just AFM/CMD.***

AFM Registered Owners

Enclosed is a NEW disk. It replaces the disk you may have recently received in the mail from us. To our dismay, a short time after we sent that new disk out to you a problem was found involving some versions of TRSDOS™ 1.3 (there were more than one version), and a small report generator bug. Problems did not show up on all DOS systems and we didn't hear back from that many of you having a problem, but we did correct the programs, re-checked and tightened up the sample data file; and we're now sending you a new disk that contains the latest corrected versions of the program. We are very sorry if you had problems with the previous disk. The enclosed Demo disk has another purpose besides showing itself off and helping you learn AFM visually. You can update YOUR current AFM/CMD to the latest version 2.5 with it - ***IF you use the following procedure(s) ONLY.***

You may remember that we encouraged running the demo under TRSDOS™ 1.3 only - regardless of the DOS you normally run under. There were two reasons for this. One was that we could control the environment you would be running in. The other reason was that we didn't want to take a chance on your accidentally getting your current data messed up by 'being on-line at the same time possibly as the sample data enclosed with this disk (example: if using a hard disk). We have decided to allow you to use your favorite DOS, but you must take several precautions or you might lose your current data.

If you want to run the Demo/Tutorial from a hard drive, back your data files off onto floppies before doing anything else. *This is most important.* If you are running from a floppy-based system, don't put your data disk in any drive while running the demo. OK? If that's understood, you may use the DOS you like to use (Model III or I-Double Density, except for 2.7 or 2.8DD)

RUNNING THE DEMO - We still suggest that for the purposes of running this demo, it's easiest to just put the enclosed data disk in Drive :1 and boot a "fresh" copy of TRSDOS™ 1.3 in Drive :0 and type: AFM <ENTER>. When you execute AFM/CMD, control of the program will be handled automatically.. Please do ONLY what the program tells you to do with NO extra keystrokes. Also, please do ALL that the program tells you to do. See the enclosed data sheet for further details.

When done with the demo, you may use the DOS of your choice to run AFM from, if you prefer something other than TRSDOS 1.3. But it's up to you. Copy *ALL* the files from this disk over to whatever DOS you want or to your hard disk if you prefer. Some DOS's might need the <BREAK> key activated (See your DOS manual for instructions if needed.) If the demo is not working properly, try running it under TRSDOS™ 1.3. As stated before, this is the format of this disk. Model I owners that do not have any way to move the files off this disk (meaning that you don't own Super Utility™, PowerTool™, LDOS, DOSPLUS, NEWDOS/80, or MULTIDOS), return the disk to us and we will put it on TRSDOS 2.8 DD format for you.

You may also use this disk for updating your working copy of AFM. First move the AFM/CMD and AFU/CMD files from the DEMO DISK to *YOUR* AFM system disk. Below are the detailed instructions for various DOSes. You've got a few other steps too, but let's get it on your disk first. Then remember to do STEP 2!

Instructions for UPDATING your current AFM work disk...

*TRSDOS™ 1.3 USERS

The DEMO DISK is formatted with TRSDOS™ 1.3, so this is easy. Put your current AFM System disk in Drive :0 and the DEMO DISK in Drive :1. Type the following two commands:

```
COPY AFM/CMD:1 :0 [ENTER]
COPY AFU/CMD:1 :0 [ENTER]
```

***LDOS™ Mod I/III USERS**

Confirm that CONV/CMD is present on your AFM LDOS system disk. Do this by typing DIR CONV(I) on your AFM LDOS System disk in Drive :0. If present, put the DEMO disk in Drive :1 and type:

CONV AF/CMD:1 :0 [ENTER]
(if using a hard drive, use the proper floppy drive number. If CONV/CMD is not there, COPY it off your MASTER LDOS disk.)

***DOSPLUS 3.5 USERS**

Make sure that CONVERT/CMD is present on your AFM DOSPLUS 3.5 system disk by typing DIR CONV*/CMD,I. If so, put the DEMO disk in Drive :1 and type:

CONVERT AF*/CMD:1 :0,V13 [ENTER]
(if using a hard drive, use the proper floppy drive number. If CONVERT/CMD is not there, COPY it off your MASTER DOSPLUS 3.5 disk.)

***NEWDOS/80 USERS**

Set your PDRIVE to read TRSDOS™ 1.3 and COPY the files over.
Don't forget to reset your PDRIVE!

STEP 2. - IMPORTANT!! <<<<<<<<<<<<<<<<<<

After getting the new AFM/CMD and AFU/CMD onto your System disk, make sure that your AFM data file disk(s) are on-line and type: AFM [ENTER]

Then type these commands **EXACTLY as listed** and in this order. Note that all lines **DO** start with an asterick [*]. Again... **make sure your DATA file(s) are on-line!**

```
*AUTO=AFM 1 [ENTER] <<< Don't forget the SPACE and the "1"!
*SORTSPEC=SORTFILE/AFM [ENTER]
*INPUT [ENTER]
*C=32 [ENTER]
```

That's it! Thank you.