

Pro/file Updates

The Newsletter For ZX Pro/file Users

Vol. 2, No. 3

July, 1985

ALPHABETIZE YOUR FILES!

Faster more versatile sorting, more than any other capability, has been the major desire of many many Pro/File owners. Figuring out how to do this was a major undertaking, but here at last is the ZX PRO/FILE machine code sort enhancement.

Before we hunker down to the nitty gritty of entering and using this modification, it is a good idea to get at least a small idea of what this enhancement does and how it does it. If you think of each file you store as a 3 x 5 index card with each line you see on the screen representing a line of text you would write on the card, this upgrade lets you alphabetically sort each card on the basis of any specific line a file may use. In other words, if you have a name on line 1, a date on line 2, a numeric quantity on line 3, a zip code on line 4, etc., you can, with this routine, alphabetize your files by name, order them by date, numeric quantity, or zip code. What you do is tell the computer to order on the basis of one specific file line. The machine then looks at each file, finds the line you specify, orders it, and prints it out either on the TV or on your printer.

There is simply an enormous amount of work the computer must do to accomplish this feat. First, it must find a file, then it gets the line it is to order. The program repeats this process with the next file. Once it has file lines of two separate files, the machine compares them character by character. If the second is less than the first it exchanges the two files. Then the computer moves on to the next file and repeats the comparison. The first file is thus compared with every other file in memory. A full data base consisting of several hundred files requires several hundred comparisons—and this is just to get the lowest file in order. The same procedure must be carried out for every file!

Thousands of bytes must be compared hundreds of times. In addition, special exceptions abound. What do you do with files which don't have data on the line it is supposed to order by? How can the computer tell when it has finished ordering all its data? All this byte work results in some interesting outcomes when you check the time required to sort a data base. Naturally, the more data you have the longer it takes to sort. Another major factor in the time requirement is how far out of order your data is before you start. If everything is in order with the exception of just a few files, ordering is much quicker. Finally, sorting is always slower at the beginning. As more files are alphabetized, the time needed to organize the rest shrinks.

All this is readily apparent when you watch this modification work. Sorting a small data base is quick—almost instant. But as your files grow, the computer takes longer. I noticed that several seconds were needed to find the first file in my 64K data base. As ordering progressed, however the program speeded up dramatically. By the time the computer reached the "last leg", files were ordered in the blink of an eye—faster than the printer could print them out.

Compared to a simple search, the sort routine is slow because much more computing must be performed. Compared to Pro/File's original numeric ordering function, well, there's just no comparison. This enhancement is vastly superior both in speed and versatility.

NOW THE NITTY GRITTY

Memory Requirements

If you use 16K of ram on your computer, you'll need to reduce capacity by 1000 bytes to allow enough room for this modification. Procedures are shown in volume 1 of Updates for doing this. Alternatively you can add this mod to a "cleared" copy of the program (see page 40 of the Pro/File manual) and then re-initialize your variables with a D\$ consisting of 10,000 spaces.

In my own highly modified 64K program (D\$=36000 characters) I managed to squeeze everything in without reducing capacity.

Part 1: Extending Space for the Machine Code

As you can see, there is a good sized chunk of machine code which must be added. You may recall from Updates Vol.1 No.2 how space for machine code could be created in D\$ itself. This time I'll show how you can add more space in a REM line for this code. If you LIST the program, you'll observe that lines 2, 3, and 16 are REM lines which hold Pro/File's existing machine code. Immediately after this is line 17 CLS. Because of the nature of machine language, you can't just insert a line between lines 3 and 16 to store new code. Whatever new space you create must

appear AFTER line 16. Since there already exist lines 17, 18, 19, etc. it would appear that doing this is next to impossible. With some judicious poking, it is fairly easy to correct this problem. The directions that follow replace the current line 17 CLS with a new line 17 REM with enough spaces in it for the sort modification plus some extra bytes for future use. In addition we'll poke a pointer or two to make the computer combine lines 2-17 into one big REM line, thus letting us put the old line 17 back in the program.

1. First load Pro/File into your computer. When the Main Menu comes up, break into the listing by typing SHIFT 1, STOP, and ENTER.

2. Now add a new line 17. It should consist of the line number (17), the REM token, and enough periods (.) to fill the line out to 16 full lines. After the 16th line is typed and the cursor is in the left hand column on the 17th line, press ENTER. After you do, the listing on the TV screen should look like this:

```

17~REM .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..
18 PRINT "zx pro\file",,,, "ent
er a search command", "or type la
l to add ", " isavel to sa
ve ", "SPACE OPEN: ";LEN D*-P;"
SLOTS";AT 10,0;"SEPARATE MULTI-
```

3. If you entered the correct number of periods, you can type: PRINT PEEK 17446 and the answer you get should be 118. The address 17446 is the address of the ENTER character at the end of the REM line you just created. If your answer is 27 (the code for ".") the

line is too long. If it is some other number the line is too short. Edit this line and adjust it to the right length by adding or deleting periods as required until PEEK 17446 equals 118. Do not proceed until you get it right.

4. As mentioned above 17446 is the last byte of your new rem line. The first byte of the first rem line (address of the REM token in line 2) is located at address 16513. We can determine the TOTAL number of bytes used by ALL the rem lines by calculating:

$$17446-16513+1=934$$

5. If you read the chapter in the Timex owner's manual entitled "Organization of Storage" you will find that program lines are stored in memory in an organized way. First, two bytes of memory store the line number, Second, two more bytes store the length of the line plus the ENTER character. Finally, the actual text of the line is stored and at the end is the code 118 which signifies the ENTER key. The two bytes which point to the length of a program line concern us here. If you know where these two bytes are located, you can poke them with your own properly calculated length to combine 2 or more lines. Doing this has the effect of merging several smaller lines into one big one. Once done, the smaller lines no longer exist as separate lines.

Now, we know that the REM token in line 2 is located at address 16513. It follows then, that the two preceding bytes (16511 and 16512) point to the line's length. In step 4 we calculated the length of all the REM lines put together as being 934 bytes. If we poke 16511 and 16512 with this new length, we will merge all the rem lines together. The simplest way to start doing this is to type the command:

```
RAND 934
```

This sets the system variable SEED (address 16434, 16435) with the value of 934. Since SEED is a two byte integer, we can simply take the values held there and poke them into 16511 and 16512.

6. Therefore, after you enter the RAND 934 command above, Type in this command:

```
PRINT PEEK 16434,PEEK 16435
```

Two values will be printed on the screen. They should be 166 and 3. These numbers are what must be poked into the length pointers at address 16511 and 16512.

7. Very carefully, with fingers crossed and a few magic words thrown in for luck type:

```
POKE 16511,166 and ENTER  
POKE 16512,3 and ENTER
```

Once you poke these pointers, you are VERY vulnerable to a crash should you accidentally produce a program listing. So be careful. Even after 16512 is poked, you can still crash until you enter this new line which straightens everything out. So before you uncross your fingers and breathe a sigh of relief enter this:

```
17 CLS
```

Now you can breathe again. If you list your program you won't see much different except your line 17 with all the periods is no longer visible. Even though you can't see it, the line is there none the less. What's more, it as well as lines 2, 3, and 16 are now all incorporated into one big REM line, line 2. You now have space for new lines 3 to 16 if you ever need them.

Part 2: Changing the BASIC Listing

This phase is simple. Add, alter, or delete the Pro/File listing so that the lines shown here are incorporated.

```
33 ....DELETE THIS LINE  
123 ....DELETE THIS LINE  
124 ....DELETE THIS LINE  
237 IF Y AND Y$<>"C" AND Y$<>"N  
" THEN GOTO 751  
700 ....DELETE THIS LINE  
701 ....DELETE THIS LINE  
703 PRINT AT 15,0;Q$;AT 16,0;"L  
INE NO.?" ;TAB 0;" OR ENTER"  
706 IF Z$=" " THEN GOTO 17  
708 IF VAL Z$=0 THEN GOTO 1000  
709 IF VAL Z$>12 THEN GOTO 704  
710 CLS  
715 LET X$="*"  
717 PRINT X$
```

```

720 LET X$=X$
725 RAND USR 16820
730 RAND P
735 POKE 16600,PEEK 16434
740 POKE 16601,PEEK 16435
745 POKE 17087,VAL Z$
750 IF USR 17216 THEN
751 CLS
755 LET B=USR 17008
760 IF B=1 THEN LET Y=0
765 IF B<P THEN GOTO 125
770 GOTO 755
1008 LET Y=0
1010 PRINT AT 13,0;Q$;Q$;"SEARCH
COMMAND?";Q$;Q$
1030 GOTO 23
1032 ....DELETE THIS LINE
1033 ....DELETE THIS LINE
1040 ....DELETE THIS LINE
1060 ....DELETE THIS LINE
1070 ....DELETE THIS LINE
1100 ....DELETE THIS LINE
3020 IF USR 16674 THEN

```

Part 3: Poking in the Machine Code

1. To the end of the Basic listing, add lines 9900 to 9950 below. They serve as a temporary input utility to let you type in the machine code.

```

9900 FOR X=16953 TO 17254
9910 PRINT X;"=";
9920 INPUT Y
9930 POKE X,Y
9940 PRINT PEEK X
9950 NEXT X

```

2. Use the command, GOTO 9900 to start this loader program going. An address will be printed on the screen and the computer will wait for you to input a number. Refer to the Poker Table below and type the value associated with the address given. For example, when you see the number 16953 on the screen, you type 209 and ENTER. After you do this, the next address in the table will be displayed on the TV. You will type 27.

As values are input, the screen slowly fills until finally the computer stops with a report 5 error: Screen is Full. When this happens type CONTINUE to clear the screen and proceed to the next address. The loader will continue to operate in this fashion until the last address, 17254, is poked with the proper value. Then the computer will stop with a

report 0/9950. At this time you can delete the program lines 9900 to 9950. They have no effect on the operation of ZX Pro/File other than to take up memory space. After they have served their purpose, they're no longer needed.

As you input values be careful to watch for the report 5 error code when the screen fills. Don't forget to press CONT when you see it. Otherwise, you might accidentally delete a few program lines without ever realizing it. If you don't notice the error code at the bottom of the screen and you type a number, when you press ENTER the program line matching the number you type in will be deleted. Results could range from inconsequential to disastrous. USE CARE!

Part 4. Initializing the Variables

You do not need to do this if you did not work this enhancement into a CLEAR copy of the program. However, if you did start with a Pro/File cleared of variables you need to set everything to their initial values.

First type CLEAR and ENTER to clear out the X and Y variables created by the Poker Routine.

Now in the immediate mode, enter the following:

```

DIM D$(10000)    or more if you use 64K
LET P=19
LET Q$=D$( TO 32)
LET D$( TO 20)="* SEARCH COMPLETE *"
LET Y$=""
LET S=1
LET C1=3
LET C2=11

```

Pro/file Updates is published 4 times a year, in January, April, July, and October.

Subscription rate is \$9.95 annually.

Edited and Published by:

Thomas B. Woods
P.O. Box 64, Jefferson, NH 03583
(603) 586-7734

Copyright 1985 Thomas B. Woods

Machine Code Sort Poker Table

Address=Value	Address=Value	Address=Value	Address=Value	Address=Value	Address=Value	Address=Value
16953=209	16997=1	17041=66	17085=66	17129=82	17173=43	17217=18
16954=27	16998=1	17042=26	17086=6	17130=209	17174=66	17218=0
16955=42	16999=0	17043=190	17087=1	17131=225	17175=42	17219=34
16956=47	17000=237	17044=56	17088=205	17132=35	17176=55	17220=55
16957=66	17001=66	17045=163	17089=53	17133=40	17177=66	17221=66
16958=213	17002=34	17046=32	17090=67	17134=214	17178=9	17222=229
16959=237	17003=53	17047=240	17091=48	17135=43	17179=34	17223=205
16960=83	17004=66	17048=235	17092=27	17136=24	17180=55	17224=83
16961=49	17005=225	17049=35	17093=62	17137=224	17181=66	17225=67
16962=66	17006=32	17050=19	17094=23	17138=84	17182=205	17226=209
16963=235	17007=224	17051=62	17095=190	17139=93	17183=162	17227=25
16964=237	17008=205	17052=140	17096=40	17140=27	17184=64	17228=34
16965=82	17009=83	17053=190	17097=24	17141=24	17185=167	17229=43
16966=35	17010=67	17054=40	17098=62	17142=206	17186=42	17230=66
16967=34	17011=237	17055=18	17099=140	17143=42	17187=238	17231=34
16968=51	17012=75	17056=62	17100=190	17144=47	17188=64	17232=47
16969=66	17013=55	17057=23	17101=35	17145=66	17189=237	17233=66
16970=235	17014=66	17058=190	17102=32	17146=1	17190=91	17234=201
16971=209	17015=9	17059=40	17103=245	17147=0	17191=53	17235=42
16972=237	17016=205	17060=13	17104=16	17148=0	17192=66	17236=16
16973=67	17017=185	17061=235	17105=32	17149=62	17193=237	17237=64
16974=53	17018=66	17062=62	17106=42	17150=23	17194=82	17238=1
16975=66	17019=32	17063=140	17107=43	17151=205	17195=200	17239=6
16976=26	17020=122	17064=190	17108=66	17152=53	17196=167	17240=1
16977=229	17021=237	17065=40	17109=1	17153=67	17197=33	17241=9
16978=98	17022=83	17066=221	17110=0	17154=48	17198=1	17242=229
16979=107	17023=41	17067=62	17111=0	17155=60	17199=0	17243=237
16980=43	17024=66	17068=23	17112=62	17156=35	17200=237	17244=75
16981=237	17025=229	17069=190	17113=23	17157=3	17201=66	17245=50
16982=75	17026=42	17070=40	17114=3	17158=190	17202=32	17246=64
16983=51	17027=43	17071=216	17115=35	17159=32	17203=234	17247=9
16984=66	17028=66	17072=24	17116=190	17160=251	17204=201	17248=43
16985=237	17029=34	17073=224	17117=32	17161=237	17205=229	17249=34
16986=184	17030=47	17074=26	17118=251	17162=91	17206=197	17250=39
16987=225	17031=66	17075=190	17119=201	17163=47	17207=237	17251=66
16988=119	17032=225	17076=40	17120=167	17164=66	17208=75	17252=225
16989=237	17033=205	17077=210	17121=201	17165=237	17209=39	17253=35
16990=91	17034=185	17078=195	17122=229	17166=83	17210=66	17254=201
16991=49	17035=66	17079=57	17123=213	17167=53	17211=237	
16992=66	17036=32	17080=66	17124=237	17168=66	17212=66	
16993=229	17037=105	17081=84	17125=91	17169=34	17213=193	
16994=42	17038=229	17082=93	17126=43	17170=47	17214=225	
16995=53	17039=42	17083=34	17127=66	17171=66	17215=201	
16996=66	17040=41	17084=43	17128=237	17172=34	17216=33	

Part 5. Final Strokes (and POKES)

A few miscellaneous items remain for various configurations of ZX Pro/File.

AERCO Disk Users must POKE 17187,238
Everyone else must POKE 17187,123

If you made any past enhancements which

reserved space in D\$ for machine code (line insert, break-proof editing, and number cruncher mods) you must POKE 17240,1 and type LET D\$(266 TO 275)="COMPLETE ". There are two spaces after word COMPLETE. Use inverse characters.

If you DO NOT make use of machine code in D\$ you must POKE 17240,0 and type LET D\$(10 TO

19)="COMPLETE ". Again, use inverse characters and spaces.

TRYING IT OUT

Before you actually start the program rolling, make a back-up on a spare tape. Just use the SAVE "ZX" command. If your machine crashes during the testing phase, you can reload this copy and not need to re-enter everything from scratch.

Now you're ready to test out your work. Type GOTO 17 to start the Pro/File up. If you made this mod on a copy of the program which does not have any files added yet, create a few test files like these short 3 liners. For starters keep things simple: a name on line 1, a date on line 2, and a file number on line 3 is sufficient:

*TOM	*DICK	*HARRY
1976	1953	1960
FILE NO. 1	FILE NO. 2	FILE NO. 3

This new sort function can alphabetize these files based on any line the file holds. In the case of the example files given, this would mean by name, date, or file number. To see how this works type AUTO when the main menu is on the screen. This will put Pro/File in the AUTOSEARCH mode. The next prompt to come on the TV asks you if you want a printout. Like the old Pro/File, this means do you want printer output or a TV display of your data? You respond with either "Y" or "N" followed by ENTER. This time around, type "N" to tell the machine you don't want printer output.

Now a new prompt comes up which says "LINE NO.? or ENTER". It is here where you tell the computer which line you want to order by. Using the test files above, if you type 1, the computer will order files by name. A 2 entered here will order them by date. And if you type in a 3, you'll get files ordered by file number.

Besides entering a bonafide line number, you also have the choice of pressing just ENTER which quits the Autosearch mode and returns you to the Main Menu, or you can enter a zero. If you do this you're telling the computer not to order the data, but rather,

to simply spit files out as they're found. Whenever you type in a zero at this prompt, the machine will follow with a request for you to type in a Search Command. Any search command you would type in from the main menu is valid here. Once it is entered, the computer will automatically scan your data looking for matches. Each file that has a match will go to the printer or the TV as per your Y/N response to the printout prompt.

Try ordering based on line 1. Type the number 1 followed by ENTER. What you'll get is a display of EVERY file in memory ordered on the basis of line one. After a file is displayed, the Display Options are printed as usual. You can select any option except R-RETURN TO PREVIOUS FILES. This option is ignored and is the same as pressing ENTER which will continue to the next file in order.

After the last file is printed, the computer prints SEARCH COMPLETE on the screen. At this time you CAN press "R" to repeat the ordering sequence.

POINTS TO PONDER

When you run an alphabetized sort on your files, the computer puts EVERY file in memory in proper order. EVERY file will either be printed out on the screen or the printer. Suppose you have a data base of names and addresses. You want to produce a list of names ordered alphabetically, but you don't want every name. Instead, you want just the people from California. Or suppose you want several completely separate lists: people from California, Texas, and New York. In each, you want the names alphabetized. Can Pro/File do this? No sweat! Here's how.

Let's assume that names always appear on line 1 just like in the example files. Run an Autosearch. Select the "Y" option for Yes, print files out on the printer, but turn OFF the printer. Select line 1 as the file line to order by and let Pro/File do its thing. If you send data out to a printer, but the printer is connected and turned OFF, the computer will order everything without stopping until the SEARCH COMPLETE message is shown.

Go back to the Main Menu now. Run another Autosearch, but this time type zero when the

machine asks you for a line number to order by. When this is done, you can input a search command for California, Texas, New York, or whatever. Turn your printer back on if you want a print out. The computer will search and print only the files you select. However, since EVERY file is alphabetized by name, any group of files found by a search will also be alphabetized by name. This will remain so until you add or edit a file. Then the new file will appear last in the list. Re-ordering with another Autosearch will be necessary.

ORDERING PROTOCOL: Which Comes First?

When this routine gets down to the business of deciding what comes before what, it does so in a manner very similar to the way ZX Basic compares strings. The computer looks at each line as a series of characters. When two lines are compared, the first character of one line is compared with the first character of the second line. If the code of one is less than that of the other, the lower code goes first. If the codes are the same, the second characters of each of the two lines are compared. This sequence repeats until either two characters are no longer the same or until an end of a line is encountered. The shorter line will then go first.

It is important to remember that ALL characters, letters, numbers, punctuation marks, even blank spaces are treated equally when the comparison is made. Sometimes this can produce some surprising results. For example:

1 always comes before 2 but
100 will also come before 2.

In both cases, the code for the first digit of the first number is less than the code for the first digit of the second number. To overcome this apparent mistake, always make sure your numbers have the same number of digits in them. If the number 2 was written as "002", ordering would be correct. Refer to the Appendix in the Timex owner's manual to determine codes for each character. The blank space will always come first. This is followed by punctuation marks, which in turn are followed by the digits 0-9, then the

letters of the alphabet. Inverse characters come last.

As stated previously, if the computer finds equal characters in the two lines it is comparing, it checks the next character on each line. This process continues until a non-equal pair is found or if the end of one of the two lines is encountered. When this happens, the shorter line is placed first. What this means to you as you order your data is that the shorter word of two that are similar will always appear first in the arrangement of data. For example:

ROBERT comes before ROBERTA
MR comes before MRS
ZX comes before ZX81

LONG AND SHORT FILES

In your data, it is quite likely that some files will be 3 lines long, others may be 7 lines long, and still others may have just 1 single line. If you want to order your 7 line files based on line 6, the shorter files—those which don't have 6 lines—will be placed first in your data. This is an extension of the previous rule concerning longer and short words. A line which has NOTHING on it is shorter than the line which has SOMETHING on it. The shorter line goes first.

DOUBLE AND TRIPLE SUB-SORTING

A feature of this alphabetizing routine which occurs more or less by accident is the ability to order sub-groups of ordered data. As an example, consider a data base which has a dollar amount on line 1, an account name on line 2, and a date written as a number (85-07-04 signifies July 4, 1985) on line 3. You could if you want, produce a list of accounts ordered alphabetically, and for each account sub-group, transactions could be arranged chronologically or by dollar amount.

This is accomplished by running two Autosearches. First, order your data by line 3, the date. When the ordering is completed, Re-order everything by account name. You will find that each account name will appear in

alphabetical order. Each group of accounts will be further ordered by date. The rule that this example illustrates is that whenever an ordering process encounters a group of files with the ordering lines being the same, the group will remain in the order it was placed in by the last Autosearch. Sub-sorts can be extended to any depth. It is quite possible to produce sub-sorts of sub-sorts of sub-sorts. Using the accounting example, you could triple sort the data so that for each account, transactions of the same date would be ordered by dollar amount. This would be done by first running a sort based on line 1, then sorting by line 2, and finally by line 3.

In summary, the ordering protocol is quite simple and straightforward. Numbers which are to be ordered must have the same number of digits in them. Words which are similar but shorter will come first in the ordering process. Empty lines are thought to be the shortest lines of all and therefore, will be ordered first. Finally, the results of a sort on any given line will be further sub-sorted by the file line selected in the previous sort.

As you merrily sort away using this enhancement, you will discover that computer sorting is not the same thing as people sorting. To the computer, there is a distinct difference between J.P. SMITH and JP SMITH which the human mind has no trouble seeing as being the same. Punctuation marks, blank spaces, and other seemingly obscure differences in two words can produce surprising results. With practice, however, you'll find a system for adding data which the sort routine will order properly and consistently. Actually, consistency is probably the key to success. Once you find a method you like, stick with it. Your computer will love you for it and will give you good data when you ask for it.

PAUSE ∅

A recent ad in National Geographic Magazine for Nissan automobiles featured the very latest in today's high technology. A completely digital computerized dashboard that "tells you everything except the age of the Passenger." What's it called? Why the 300 ZX, naturally!

UPGRADED ZX PRO/FILE CASSETTES AVAILABLE

Many readers who would love to spruce up their old Pro/Files with the enhancements covered in this newsletter, but who lack the time or ability to enter them in by hand have expressed a wish to be able to purchase cassettes of "ZX" incorporating some of the more extensive modifications.

In an attempt to satiate the extensive demand of the dozen or so Updates readers who wish to partake of this deal, I am offering cassettes of an Enhanced ZX Pro/File in the following format:

ZX Pro/File Plus....

Includes NOT Search, File Count, DO file capability, Break-proof line editing, Number Crunching, Machine Code Sorting, Extended Display Options, Line Insert, Idiot's Idiot Line. Please specify memory size and whether your printer is the Timex 2040 or the Memotech Centronics. Sorry, other interfaces cannot be catered for. No documentation is included with the tape since you already have it in the form of this newsletter. Every function will work exactly as described, however slight changes (improvements) in programming may be incorporated. This offer is good for Updates subscribers only. Price: \$16.95 plus \$1.50 handling and postage.

UNCLASSIFIED

Sell that piece of gagey that failed the smoke test, or that extra printer, or memory pack. Non-commercial ads: \$5.00 for 5 lines

WANTED: Memotech RS232 interface and cable for TS1000. Will pay \$50. Anthony J. Oresteen 452 Orion, Batavia, IL 60510. (312) 879-5608

FOR SALE: Aerco Disk System: ZX Interface board, Power Supply, Panasonic DSDD drive, 2 ribbon cables. \$350. pp R. Combes, 471 Carriage Dr. N.E., Atlanta, GA 30328 (404)255-7076

This issue was so tight for space I didn't have room for the disassembly of the sort algorithm. Hopefully, next issue I can squeeze it in.