GROC Database Management

For Tom W.

Jane Jellison, 1990

INTRODUCTION

This collection of programs and instructions for managing the GROC database has evolved over the years, starting in dBaseII and then being shifted into dBaseIII+. Since dBaseII was a little light in structured help such as the dBaseIII+ Assistant, relatively little use is made of this feature. It is needed to modify report forms, which are not written in code intelligible to humans. All the .prg (which were .cmd in dBaseII) files are in ASCII and can be edited with most word processors in a non-document mode. You can also alter them with the dBaseIII+ *modify command filename* command.

Most of the techniques were worked out by issuing commands at the dot prompt. Some people find this intimidating, but I sort of like that friendly little spot waiting patiently in the lower left hand corner of the screen to do whatever I want, without my having to wade through a bunch of menus or being restricted in the commands I can issue. The trick, of course, is to know what you want, and that is what I am trying to write down in this guide. Some knowledge of dBase will be very helpful, and may even prove necessary to use this manual successfully. There is neither error trapping nor help available if you get off track--you are on your own with that dot!

Good luck.

FUN RUN REGISTRATION

Before beginning the actual data entry process, you will have to do some housecleaning to make ready for the new race data and its processing. Old times, team numbers and places need to be cleaned out, and several program files need to have the current race dates put in them (for use in calculating the ages of the runners).

Start with the main database file, GROCLIST.DBF. If, when you boot up dBaseIII+, the Assistant is on, i.e., you see some menu selections, hit *escape* to send him back to sleep. From the dot prompt,

use groclist
replace all place with 0 (zero, not the letter o)

Wait for the message "nnn replacements made" or whatever the message is. Repeat this process for times and team numbers:

replace all t2mi with " " (five spaces in quotes)
replace all t10k with " " (six spaces in quotes)
replace all tno with 0 (zero. No quotes; this is a numeric field)
replace all runno with 0 (zero again)

The programs NASA2 and NASA10, and the report forms TWOMI and TENKM all contain the race date. Type

modify command nasa2

and this is what you should see:

```
*!!* dBASE CONVERT - dBASE III File Conversion Aid  v2.01  11/19/85
*
SET HEADING OFF
SET SAFETY OFF
*NASA2.CMD Creates a text file in upper case ASCII listing name, time, age,
*code.  Race date and filename must be changed for each race. 10/25/83 JEJ
use groclist index t2mi
goto top
set talk off
set alternate to FALL892
do while .not. EOF()
 if t2mi > "00:00"
   store int((891011-birthdate)*.0001) to age
   store trim(UPPER(firstname))+" "+trim(UPPER(lastname)) to name
   set alternate on
    if sex = "F"
     ? name+"(F)", t2mi, age, code
    else
     ? name, t2mi, age, code
    endif
  endif
 skip
```

```
enddo
release all
set alternate off
```

See where it says "store int((891011-birthdate)*.0001) to age"--this line subtracts the runners birth date from the race date (here October 11, 1989) and truncates it to a two digit number which corresponds to the runner's age on race day. (That's why the birthdate is entered into the database in the form yymmdd as a numeric field rather than being entered as a regular date. It's quite possible dBaseIII+ could do this some easier way, but this is one of the holdover programs from dBaseII and it works just fine). Your task is to change the 891011 into whatever the current rate date is. Cursor to it, and with judicious use of the delete key and your fine typing skills make it say whatever it should.

This program is used to put the race results in the right form to be submitted for the NASA Intercenter competition. The other thing you need to do is update the name of the output file which this program creates. This is done by editing the line that says "set alternate to FALL892." You can pick any name you want, but it should reflect the date and type of race, that is, fall or spring, two-mile or 10K. For example, "FALL9010," "FALL902" and "SPR912" would mean Fall 1990 10K, Fall 1990 two-mile and Spring 1991 two-mile respectively. Remember you are limited by the DOS convention of an eight character file name. When it looks right,

control end

to save it. Do the same thing with NASA10.

For the report forms, we will use the assistant to modify them.

assist

Use the right arrow key to move to "Modify," cursor down to "Report" and hit *enter*.The forms you want to change are TWOMI and TENKM. The field that needs to be changed is called "age." It is the same construction as is used in the NASA2 and NASA10 programs, and needs to be fixed up with the correct race date. For each form, right arrow to "columns" and then *page down* to age. When you alter it, dBase wants to make it a 13-space-wide field: arrow down and make it be 3 spaces. Go down some more and make it say "no" on the line where it asks whether you want to total the field, or else the form will not print correctly. When you are done with each form, arrow over to "exit" and then "save" your changes.

There is another database which needs attention--TENKAY.DBF. This one contains information relative to the (you guessed it) ten kilometer event. You need to erase all the old records and get ready for the new race and the attendant trophies.

```
use tenkay
delete all
pack
```

This empties out all of the old stuff but retains the structure to accept the new stuff.

Now the real fun begins--entering all that data. Hit *escape* to get back to the dot prompt. Since you may not have been the last person to use the file, or may not have used it in a while, it is a good idea to make sure all the index files are up to date. The name and code

indexes should be OK, but we have just changed data in the runner number field so that index is no longer valid. Don't worry about 2-mile and 10k time indexes at this point.

use groclist
index on lastname to lname (say yes when asked if you want to overwrite the existing index)
index on code to code
index on runno to runno

You don't have to re-index every time you quit and start up again provided that you use them every time you enter some data. The next command takes care of this:

use groclist index lname,code,runno
set format to entry (this will give you a screen that mimics the entry blank, sort of)

Then, look and see if the runner is already in the database:

find Nnnn

where Nnnn is the runner's name with the first letter capitalized and the rest in lower case. If you are in doubt of the spelling, try just the first 3 or 4 letters of the name.

If the name is found, you will see the dot prompt, and the record number will appear in the highlighted bar at the bottom of the screen.

edit

The record now appears on the screen. If it is the person you want, update the information, checking code, phone, etc., and entering the runner number. Make sure there is a birth date--it fouls up the age calculation if this field is blank. If you are unable to find out the person's age, enter a date that will make the runner be 25 years old, for example, 650000 if you are doing the 1990 races. This puts him/her in one of the most competitive classes, which serves her/him right for being coy with GROC. For gender, use upper case M or F as appropriate. The structure of the dues field is as follows: the first character is the last digit of the current year, the second character is either a lower case y or n (y = club member; n = non-member), the next is r for runner, and after that a lower case o if the person has indicated interest in orienteering. For example, "0yro" means a person paid his dues in 1990 (the newsletter label printing program looks for that "y"), runs and likes to orienteer; "0nr" means that the person ran in a race in 1990 (and therefore should not be deleted) but did not join the club, and thinks orienteering is for the birds. For the "CAT" field, the possibilities are "cs" for civil servant, "osc" for on (or near) site contractor, "coo" for Coop student, "ret" for re-tired, "gst" for guest. I hate to tell you this, but there is another database file, CONTRACT.DBF, which contains information that GEWA (Goddard Employees Welfare Association) requires. They want to know the contract number under which each of our contract employees works. This is pretty tough to do, but we try.

use contract
list to print

will give you a listing of what is in there. Make sure your printer is on line before issuing this command. The major problem is people not knowing or not listing what their conract number is. Do your best, but don't accelerate gray hair development over this issue. Put an

upper case Y or N in the release field or leave blank. (Blank equates with "N" since the program that generates a listing of those who are willing to be put on mailing lists asks for the "Y" in this field.)

When you are done with this record,

control end

will save the new info and return you to the dot prompt. If you edited the last field, the next record may have appeared. *Control end* works here too.

If the first record that appears is not the one you want, and you suspect there may be duplicate entries with the same last name, you can *page down* through the Smiths, for example, until you either find the desired record or give up. If you come across the entry you are looking for, *edit* the record, *control end* and proceed on your merry way.

If you have been unsuccessful, type *control end* anyway to get out of there. Try typing the first few letters of the last name and then

browse

Rows of records will now appear. Use the up and down arrow keys or page up and page down to search the neighborhood for the name you are seeking. When you have highlighted the one you want, the record number will appear at the lower right hand corner of the screen. Hit *escape* to get out of the browse mode, type *edit* and off you go.

If the name (or partial name) is not found, the message "no find" will appear. Type

append

and a blank screen will appear. Have at it, and type *control end* when it looks beautiful. If a new blank record appears, hit *enter* to return to the dot prompt. Then *find* the next name and repeat the whole process until you are finished or ready to drop, whichever comes first.

Before you hang it up for the day, it is a good idea to print out a list of runners and their runner numbers. Typos do exist, and the forms are not always right either. Type

use groclist index runno
list runno,lastname,firstname,code,phone for runno > 0 to print

This gives you a listing starting with the record number (which is handy to have for any necessary editing), and then listing in runner number order all the folks you have signed up. Check for duplicate runner numbers and for omissions. The phone numbers will allow you to call the person and ask which number they actually have. Then, at last, with good conscience, you can type

quit

This is very important--dBase gets very upset if you do not *quit* properly (closing files, and stuff like that) before turning off the machine, and you wouldn't like to lose all that work, now, would you?

TEAM ENTRIES

Thought you were done, didn't you, with all those runner numbers, names, birthdates, mail codes neatly entered. Think again. I have not been able to come up with a graceful way to do all the job at one time. Team entries really swell our number of participants by virtue of the team participation trophy, and it's worth it to double the work. I think. The reason we do not accept team affiliations from the individual entry forms is that some people remember teams from years past and put that down, while others forget that somebody recruited them. The only way that works (that I have found so far) is to enter the individual data from all the entry blanks, ignoring any team information on them, and then drag out the official team lists and go through them after all the individual entries have been processed. This is a large bore, and delays race result processing because the team lists are not due until 6:00 p.m. the day before the race. This means that on race day or in the awful aftermath, you, the data entry person, have to make sure that team affiliation is properly recorded. However, team competition is very popular and is necessary if we want to keep the participation level high.

First, assemble all the team lists, and assign a team number to each. To make things a little faster, there is an entry form you can use:

use groclist index lname,tno
set format to teamentr
find Nnnn

This brings up a screen which shows the runner's last name, firstname, code and team number. If the record is the one you want, all is well--just enter the team number and press *enter*. This will flip the screen to the next record, which is no doubt not the one you want. *Control end"* will return you to the dot prompt and you can proceed to the next name. If the team captain has listed this person but the person has not submitted an entry blank, too bad--there is nothing else you can do. If the spelling is bad, you can try the browse mode described above to find the person--some team captains do not know how to spell their team members's names.

The reason that the teams are numbered is to avoid cluttering up the main database, groclist, with a large team name field. There is another database called teamlist.dbf, which contains fields with team name, score, rank, and the very same team number that links this set of information with that contained in groclist. You need to update teamlist with the current information. For right now, just fix the team name, captain, and team number. You will clear out old scores and ranks later.

use teamlist
browse

Look around a bit, using the up and down arrow keys to move vertically, *home* to move one field to the left, *end* to move one field to the right, *control* --> (control right arrow) to move from the rightmost field visible on the screen to the next one to the right, and *control* <-- to move back to the left. You can edit in this mode if you wish, remembering to *control end* when you have finished, or you can hit *escape* to go to the dot prompt and then *edit* each record in turn.

ENTERING RACE RESULTS

When the finish crew and the tag team and the select times folks and finish order recorders and the video crew and the tape recorder maven have all done their thing, you will be presented with a bunch of information. If you are lucky, you will not be involved in this process, which can get pretty hairy, because there is always considerable disagreement among all these reliable sources. The ultimate result of all of this is a Chronomix tape listing finish order (place) and finish time (with some extra decimal places) with penciled in runner numbers. The next task is to put this information into the trusty GROC database. RRCA rules state that the times should be rounded up to the nearest second whatever the number in the tenths place might be--for example, 10:25.2 becomes 10:26. Personally, I don't have the heart to do this with other than the first several finishers who might be in medal competition. I use the normal arithmetic style of rounding up for .5 or more, and down for .4 or less. I also bend a little when five people whom I saw with my own eyes cross the finish line arm in arm have official times which would indicate some finished ahead of the others. After all, the limit of the clicker finger of the Chronomix operator is about 0.2 second, so that multiple finishers often receive different times. However, this is between you and your conscience--this is, after all, primarily a fun event.

Since what you have to work with is a tape with times and runner numbers, the program to use gives you a screen containing the runner number, name, and a place to enter the two-mile time. Type

timesin

and then enter the runner number corresponding to the first time listed on the Chronomix tape. Up comes the runner's name, and a place for you to enter the time, as minutes and seconds (mm:ss) without entering the colon--your friendly programmer, who also cannot reliably type a colon, has programmed it in. Do this until the bitter end. If you make a mistake, enter the runner number again, and the corrected time. This is another instance where there is no opportunity to confirm whether or not you entered the correct information, but you can always overtype it.

When this is done, it is a good idea to print out a list of the runner numbers and times that you have put in and compare them with the Chronomix tape. To accomplish this, you need to do the following:

use groclist
index on t2mi to t2mi (say Yes if asked whether to overwrite the index)
use groclist index t2mi
list off t2mi, runno for t2mi > "00:00" to print

Compare this with the Chronomix tape and edit if necessary. You can do this in a couple of ways, either by running the *timesin* program again, or

use groclist index lname,runno,t2mi
find Nnnn
edit

and fixing things up that way.

FUN RUN RESULTS AND TEAM SCORING

Before you begin the scoring, type *dir *.dbf* to get a listing of the database files that are cluttering up the place. Get rid of the last race's stuff (you might want to copy these to a floppy first to save for posterity):

erase funrace.dbf
erase funmen.dbf
erase funwomen.dbf
erase funteam.dbf
erase funplace.dbf

Don't worry about their associated index files; you will write over them later.

Another housekeeping chore that needs to be done is to modify the report form so that people's ages are correct. (The ages are obtained by subtracting the birth date from the race date.) To change the race date, we will need the Assistant.

assist

Arrow over to modify, down to report, select drive c, select twomi, *enter*. Arrow to select fields, or whatever it says, and *page down* to the age field. Type over the previous date with the correct date, and follow the instructions to save it (I think it's *control end* followed by *enter*). Arrow over to Exit and Save. Hit *escape* to get back to the dot prompt.

The following instructions assume that you have entered all the fun run times and team numbers into groclist. The next tasks are 1) put the finishers in time order, 2) make a new database file containing just the people who ran in the race (in finish order), 3) assign place positions and 4) print out the results.

use groclist
index on t2mi to t2mi
use groclist index t2mi
copy to funrace for t2mi > "00:00"
use funrace
do place
report form twomi to print

This report gives you the official fun run results. Now for the team scoring.

Peter Hui's formula for normalizing the scores of the women to those of the men requires that the finishers be separated by gender and that these two hypothetical races (men and women) be scored separately. The place positions for the women (assuming that there are more men than women) are then multiplied by a factor and diddled around so that the first man and the first woman both have a place of 1, and that the last man and the last women also have the same place number, whatever that turns out to be. The formula is

$$P = n(p-1) + 1$$

where P is the normalized place that you want to end up with, p is the place the woman earned in the women only race, that is, her place relative to the other women, and n is a

constant that varies from race to race depending on how many men and women there are

n = (men - 1)/(women - 1)

Don't ask me why, just accept it. It all makes sense to Peter. For example, suppose there are 496 finishers with 161 women and 335 men. n then becomes 334/160 = 2.0813. Use all ~~four~~ decimal places.

When the formula has been applied to the women's race, these results need to be recombined with the men's, and the whole thing sorted on place.

OK, got that? Here we go.

---

use funrace

(You are already using it if you started at the beginning of this chapter, but maybe you are just back from lunch)

copy to funmen for sex = "M"
copy to funwomen for sex = "F"
use funwomen
modify structure

What you want to do is change the place field to a width of 8 with 4 decimal places. This is easy. Arrow down to place, over to where it says 3, make it say 8, put a 4 in the decimal slot. Do *control end* and whatever else dBase wants, like maybe *enter* or something (I forget). When you get back to the dot prompt,

use funmen
modify structure

and do the same thing there. Then

do place-m
use funwomen
do place-f
report form twomi to print
use funmen
report form twomi to print

You should now have the results for the men and women scored separately. Hang these on the gym wall, people like to see them.

Now we start operating. Figure out what n is, and write it down someplace (to four decimal places).

use funwomen
replace all place with place-1
*replace all place with place*2.0813*

(Use your own number that you just calculated--this is just an example, OK?)

replace all place with place+1

Now combine the two races:

copy structure to funteam
use funteam
append from funmen
append from funwomen
sort on place to funplace
use funplace

Are you still with me? At this point you may, if you wish,

report form twomi to print

which yields a rather odd listing of the outcome of the race. Having pondered this for a while, continue with

use funplace (I know, you already are)
index on tno to funtno
use funplace index funtno
list off tno,place,t2mi,lastname,firstname,sex,code for tno > 0 .and. t2mi > "00:00" to print

Sum the places for the first five finishers on each team. This is the team score according to speed. If a team has fewer than five finishers, too bad for them. Ignore them.

Then count the number of finishers for each team for use in the team participation scoring. Current rules (October 1990) state that volunteers helping on Fun Run day instead of running are eligible to be counted toward their total team complement. Get a list of volunteers who actually showed up and helped from the Race Director, and add them to the number of finishers for the appropriate teams. Write all this stuff down on a piece of paper because we have some more housekeeping to do.

use teamlist
replace all score with 0
replace all rank with 0
replace all numfin with 0

Then enter the scores in the score field and the number of finishers in the numfin field. You know how to do this, right? After all you have presumably entered five million names, codes, times, etc., into groclist.

index on score to rank
index on 100-numfin to numfin

If you are so fortunate as to have more than 100 finishers on a team, then index on XXX-numfin, where XXX is bigger than the number of finishers. (This orders them in descending order, with the biggest number first.)

use teamlist index rank

browse

Fill in the rank field with 1, 2, 3... as appropriate starting with the lowest non-zero score. Finish with *control end.*

report form teamlist for score > 0 to print
use teamlist index numfin
report form teamlist for numfin > 0 to print

and you're done. Go pour yourself a cool one. You don't have to worry about the 10 km race for a day or so.

TEN KILOMETER RACE SCORING

This event is typically much more low key than the two-mile fun run. The number of participants is smaller, and the whole thing is much easier to deal with. I have not devised any elaborate ways to enter results--just edit each record:

use groclist index lname,t10k
find Nnnn
edit

and fill in the 10K time. (You erased all previous 10K times a while back.) The 10K team competition is for participation (number of finishers) only; there is no trophy for the fastest team. After you put all the 10K times in, you need to transfer this information to a database that contains only those who ran (or walked) in this event, and which will have provision for 10K team listings and for individual and team placememt.

use tenkay [you cleaned it out earlier]
append from groclist for t10k > " " [six spaces inside the quotes]

This will copy selected data, i.e., name, age, code, 10K time, and whatever fields the two databases have in common to TENKAY.DBF. Fields that you don't care about at this point, such as two mile time, release, two mile team number, will not be copied. You will then have to *use* this database and fill in the team affiliation, if any, for each runner. When you have done so,

use tenkay
index on 10kteam to 10kteam
use tenkay index 10kteam
list off 10kteam,lastname firstname,t10k,to print

and this should give you a list from which you can count up the number of participants on each team, and this will be used to award the team participation trophy.

For the 10K individual time trophy, you need to do something else. After the main event (usually one week after the Two-Mile Fun Run, when you have entered the 10K times:

use groclist
index on t10k to t10k

report form tenkm to print

Consult with the GROC President as to exactly what the rules are for trophies. They may include the make-ups and then again they may not.

NASA INTERCENTER RUNNING COMPETITION (IRC) RESULTS

The rules for IRC eligibility differ from those of GROC. At our local events, dependents (immediate family members) can participate fully, according to GEWA regulations governing club activities. For the IRC, dependents may be listed but cannot contribute to the Center's score. Dependents are flagged with "*DEP" in the code field. The IRC race competition period lasts for a month, and individuals who run recognized events, either two mile or 10K, may submit times achieved (honor system). We also usually hold make-up events for those who cannot compete on the official race days.

At the end of the eligibility period (for us, the months of April and October), gather up all stray results and enter them into the groclist database. The Fun Run results are already in there, although some people who ran in make up events may have faster times that they wish to enter for the IRC.

When all times have been entered, the results need to be put into a format required by the programs that integrate the results for all competing NASA centers. This is done by the two programs called NASA2 and NASA10. During the preparation for the current competition these programs will have been updated with the correct race dates. All you have to do now is to issue the command

do nasa2

which will create an ASCII file containing the runners' names, times, ages and mail codes for the two-mile event, and

do nasa10

which accomplishes the same thing for the 10K finishers and will also print it out.

Copy the two resulting files (called whatever you named them when you modified the programs earlier) onto an IBM 360K disk, label it "GSFC Results" and give it to the Race Director.

Jane Jellison
October 20, 1989
Revised October 5, 1990