Sidebar:

Staggering performance and a wide array of extensible features make MATXTAB a must-have alternative to FoxPro's native cross-tab generator.

Cool Tool: MATXTAB

By Whil Hentzen

FoxPro's GENXTAB is a wonderful tool for doing numerical analysis of tables. With just a few commands, you can slice and dice data that's hidden in your system and present it in a format that your user is most probably comfortable with - the spreadsheet paradigm of rows and columns. After you've used it a few times, you wonder how you ever got along without it.

However (you knew a "however" was coming, didn't you?), one of the traits that most FoxPro programmers exhibit is a lack of content with the status quo. We're never happy with simple perfection - we always want more.

The GENXTAB wish list

So that little voice inside begins saying, "I wish that..."

- I wasn't limited to a single column as the description for each row.

- I didn't have to have the input table ordered on the description column.

- It worked faster.

- I didn't have to first run the TOTAL command on my table to generate unique rows for the input table.

- I wasn't restricted to 3 input columns in my input table.

- I could do statistical calculations on the columns.

- I could automatically order the output.

MATXTAB: A GENXTAB replacement

If your wish list looks anything like this, it's time you took a look at MATXTAB. This replacement for GENXTAB was released in early November by one of Canada's wunderkind, Val Matison of Toronto, and it is hot!

MATXTAB is completely compatible with GENXTAB's calling requirements - in fact, you can set _GENXTAB = MATXTAB in your config.fp/fpw and use it in place of GENXTAB transparently.

Performance

The first time you run MATXTAB on a table of any significant size, you'll immediately notice the performance difference. Val claims improvements of "up to" 15 times, with normal multiples of 2 to 8, depending on machine configuration and, obviously, the amount of RAM available.

I compared GENXTAB and MATXTAB on a table (STATS.DBF) with 6000 rows and 20 columns. The hardware was a 486/50 with 20 MB of RAM and all the data loaded onto a 4 MB RAM disk.

First, generating the result with GENXTAB first required me to index STATS on the first column; I didn't have to do so with MATXTAB. This tilts the scales toward MATXTAB even before the starting gun went off. GENXTAB created the 246 row, 25 column result in 3 minutes, 10 seconds. MATXTAB took between 9.5 and 10 seconds (sorry, but I blinked.) My math shows that to be a TWENTY-fold improvement - and to answer the question on the tips of your tongues - yes, MATXTAB was written in FoxPro <grin>.

Other features

Let's look at our wish list above. The first request - multiple columns as the description in the result - is easily handled by MATXTAB. You just pass the extra columns you want to show up as descriptions (up to five columns total) as parameters.

GENXTAB requires that your input table be ordered on the first column, and that each row in the input table be unique. If your input table doesn't meet these requirements, you'll have to INDEX and then TOTAL ON before you're ready for GENXTAB. MATXTAB, however, will handle your input table in any order - and the table doesn't require unique rows - rows with duplicate keys will be summed automatically.

You might have noticed that the input table contained 19 columns. And you know that GENXTAB is particularly fussy about requiring 3 columns in the input table, like the Monty Python skit: "Thou shalt have three columns - not four columns, or two. Three is the number of columns, which is to be not less than three, nor greater than three..." MATXTAB handled the 19 column table without blinking.

You can pass parameters to MATXTAB to include MIN, MAX, AVG and other statistical calculations as part of the result table. You can also tell MATXTAB to order the result, and to output the result to an array - assuming you have enough memory to handle it.

There are other features as well - including a set of error codes to return to the calling program and the ability to pass arrays for your own column headings and to filter the result.

Final comments

As an aside, MATXTAB will not run on 2.0, except in trivial cases, because it depends on building a lengthy SQL command. 2.0's limit for macro expansion is 256 characters (vs a 2048 character limit in 2.5) and I've found this to be a significant limit for 2.0 applications when MATXTAB starts building the SQL SELECT.

Take a look at the code. It's very readable - and as you go through it, you'll see another example of the adage "In simplicity lies greatness." In fact, it's likely that you'll slap yourself on the forehead saying "Why didn't _I_ think of that?"

Where to Find It

MATXTAB.ZIP is public domain and is provided on this month's Companion Disk. You can also find it on CompuServe's FoxForum. Val Matison can be reached at 416.256.4495 or 70632,3172.