Randolph

Configuring TAOS 1.3 Configurations

By William Baker

I hadn't been using TAOS 1.2.2 very long when TAOS 1.3 came out, so I persuaded Randolph to do his duty and upgrade, and I acquired a copy for my office.

BASIS had told us that the new TAOS compiler was most notable for its speed. Like I expected, Randolph thought that it was too fast. "I was just getting used to having a little break -- sometimes a little snack -- during each compile, but with the faster compiler I'm stuck at the terminal all day," he complained.

"But you can get more programming done if you don't have such long interruptions," I said.

"Maybe you're right, but what's more important, having a pleasant day or doing more programming?"

"I'm not getting into an argument about that. But I found another feature in 1.3 that might interest you. Have you learned anything about configuring TAOS?"

"Well," Randolph said, "I took all the defaults when I installed it. I hope that's all the configuring I need to worry about."

"That will probably work just fine until you start working on different projects for different people -- I mean rabbits -- and you want to divide your work into different directories. And if you start using more than one data dictionary or more than one set of data files, then you'll have to start worrying about configuration."

"I'll be sure to let you know if I should get in such dire straits," Randolph said.

It wasn't long before Randolph was writing TAOS reports for four different clients. At first he kept the reports separate by putting a client code in the file name of each procedure. Then he added a data set of large files copied from a client site. He figured out where TAOS was looking for data, and he began the painful procedure of renaming data files when he wanted to change data sets. This got to be too inefficient even for Randolph, so he came by and reluctantly admitted he was ready -- or at least resigned -- to learning how to handle multiple projects.

"You might recall from working with the EUS Toolkit how the utility _acu.utl reas a configuration file and sets global strings," I said. "I'm sure you've noticed a similar configuration file in your TAOS directory, with a default name of config.TPM. As you would expect, the TAOS front end and programs generated by TAOS call _acu.utl also, and they use some of the same globals, and a lot of new ones."

"I thought this was going to get more complicated," Randolph groused.

"The key to handling multiple projects in TAOS is -- no surprise here -- the PROJECT global string. The PROJECT global is typically a full path, and the other directories are normally below it. So a config.TPM file for TAOS might start out like this:"

```
   PROJECT=/usr/project1/
   DATA=(PROJECT)data/
   DICTIONARY=(PROJECT)BBDICT/
   BBX=(PROJECT)bbx/
   RWI=(PROJECT)rwi/
   FORMS=(PROJECT)forms/
   HELP=(PROJECT)help/
```

"When you want to change projects, you change the PROJECT global, and then you're pointing to a different set of directories under a different path. The cumbersome way to do this would be to quit TAOS, edit your config.TPM file, and restart TAOS. But both versions of TAOS we've used give you an easier way. They have a menu option to change the PROJECT global on the fly. When you change the PROJECT global, all the directory references that use that global are going to be changed."

"I have noticed a menu option called 'Project'," Randolph said, "but it was too scary-looking for me."

"Well, now you know that you can use that option to point to a whole new set of directories for data, the data dictionary, and so on. That's very nice, but there are still some problems, like the one you have where you have one set of procedures and two sets of data files. One solution that is available in both versions is to use the Configuration Editor. That screen allows you to change any individual global string on the fly. So you could put your data sets in two separate directories and use the Configuration Editor to change the DATA global to point to the directory you want."

"I suppose I could handle that," Randolph said, "with some practice."

"But that's not all," I said. "The new feature I was talking about is even more interesting. In your situation where you just have two data directories, I will grant you that changing just one global string is pretty simple, but as you add data sets and data dictionaries and projects, the possible combinations can get out of hand, and you can start losing track of the different paths. I know I have, anyway."

"Well, that's just like you, always running off in ten directions at once. If you would settle down and finish one project--as in one job--at a time, you wouldn't have these problems."

"I don't see how you can project the complexities of my work into a personality problem, but multiple projects aren't a problem any more--I fixed it. The solution is to set up a configuration file for each combination of globals that you will want to use. Then you tell the TAOS front end which configuration file you want to use, and it clears every configuration global and loads new globals from the specified file. That's where one of those scary-looking options under the Project menu comes in. There is a choice to Open SQL.INI File'. The SQL.INI file is a list of what BASIS calls *data sources*, and it might look like this:"

```
[BASIS Data Sources]
Normal configuration
Test configuration

[Normal configuration]
CONFIG=/usr/bin/config1.TPM

[Test configuration]
CONFIG=/usr/bin/config2.TPM
```

"When you take the option to open an SQL.INI file, you are prompted for the name of the file you want to open -- SQL.INI is the default. After opening this file, the data sources 'Normal configuration' and 'Test configuration' will show up in the 'Project' menu. If you select either of those data sources, the TAOS front end will read the associated configuration file and load the global strings defined there."

Randolph blinked a few times, either in confusion or amazement. Finally he said, "This has got to beat everything -- a configuration file that is a list of configuration files."

"Let me make sure I have this straight," Randolph said, scribbling on his yellow notepad. "You pick an SQL.INI file that is a list of configuration files. You pick one of the configuration files that is a list of global strings. TAOS clears all the old global strings and loads the global strings in the new configuration file, which is identified in the 'Project' menu as a data source, which you got by opening an SQL.INI file through the 'Project' menu."

"That's exactly right -- I think."

"Why does it have to be so complicated? Why not just pick a new configuration file and be done with it?"

"Well, I suppose BASIS was writing this for people like me who might have so many configuration files that we can't remember all of them. And the data source name lets us assign an alias to a configuration file, which is more descriptive than just a file name."

Randolph sighed. "I suppose now you're going to tell me there is another configuration file that lists all the configuration files that list configuration files."

"No, but we can always look forward to the next release."