Tech Tip #12: Creating a New Database in the PCC

Overview: Creating a DBName/DSN in PCC for an existing data file set

Thus far, all of our Tech Tips have concentrated on the database environment itself, including management and performance optimization. This time, we are going to switch gears and see what it takes to access your database from SQL. To do this, we first need to locate your database files (as documented in an earlier Tech Tip) and your data dictionary files, and then we need to create a new database that the engine can work see.

First, locate your database file directories. To review, you can launch the **PSQL Monitor**, then select *Microkernel/Active Files* to see what files are open, and thus where they are located. This is easy.

You ALSO need to locate the data dictionary files for your application. Data Dictionary Files, also known as DDF's, are special files that describe your underlying data file structure to the SQL engine. They are always provided in sets, including (at least) FILE.DDF, FIELD.DDF, and INDEX.DDF. Additional DDF's will be provided for newer systems. In most cases, DDF's are located in the same folder as the data files, though this is not always true. You may have to hunt for the DDF's in the application folders. Note that DDF's are NOT required to run a Btrieve-based application, so if you don't find any DDF's, contact your application developer to see if DDF's are available. (If not, complain loudly – it is your data, and you should be able to see it!)

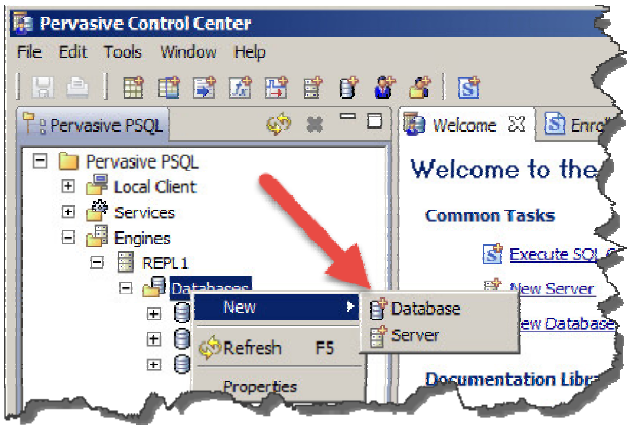

Figure 1: PCC New Database

Once you have found your DDF's and files, it is time to create the new database so that the SQL engine can access the files. First, start the **Pervasive Control Center** on your server. On the left-hand side in the tree view, right-click on the *Databases* entry, then select *New*, then *Database*, as indicated in Figure 1.

This will open up the *New Database* dialog box, as shown in Figure 2. Inside this dialog box, you first need to provide the database name. This name is somewhat limited in length and convention, so keep it short. When saved, it will be saved as all capital letters, so don't worry about capitalization.

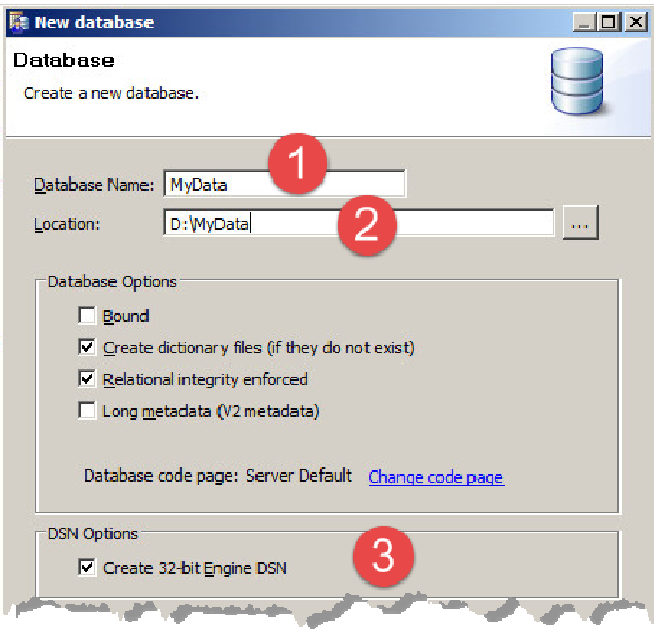

Figure 2: New Database Dialog Box

The second thing you need to provide is the location of your DDF files. Again, if this is the same location as your data files, then just browse to that folder and you're done. If the path is different, then you'll have to manually edit your database to enter the two different paths after you are done here.

Leave the *Database Options* alone, and make sure that the *Create 32-bit Engine DSN* is checked, then

click Finish, and your database will not be created. Go back to the PCC's tree view and drill down until you can see into the Tables list, and you're done! You can now double-click on a table and start accessing the information from SQL queries to your heart's content.

As mentioned above, sometimes the data files are located in a different folder than the DDF's. If this is the case, you may need to edit the database locations manually. After you create the database, right-click on the database name and select *Properties*. Then, go to the *Directories* tab and make any necessary changes. Note that if the application data files are spread out in several data folders, you may need to include several folders in the list. These folders should be specified in priority order – thus the first directory will be checked first, and so on.

Sometimes, however, you'll create a new database, but you won't see any tables inside of it. This tells us that you either provided an incorrect folder name for the data dictionaries (and the system created an empty set of DDF's for you), or that the DDF's you thought were valid were actually empty in the first place. In this case, either you will need to fix up the directory path from the *Properties* window, or you will need to obtain correct DDF's from the application developer.

If the developer is unable to provide DDF's, then things get a LOT more complicated. You can create an empty set of DDF's and then use the **DDF Builder** to try to create the definitions, but this is a complex process. If you need a more detailed explanation of why this is the case, here is a good white paper:

http://www.goldstarsoftware.com/whitepapers-accessing-data.asp

As you can see in that paper, however, doing this for a large set of files may take a lot of time. You will need to arm yourself with either solid documentation or source code for the application, and spend quite a bit of time refining the data definitions. And, even then, your definitions may not be complete or accurate. This presents another whole list of possible problems. While you can use **DDF Builder** to check the table definitions, even this is not a guarantee that all is well.

Luckily, you don't have to go through this alone. If your developer cannot provide clean DDF's for your application, then Goldstar Software may be able to help you create (or clean up) your DDF's. Contact us for more information.

http://www.goldstarsoftware.com/contact.asp

The sanity you save just may be your own.