Obtaining an MKDE Trace of a Failing Application
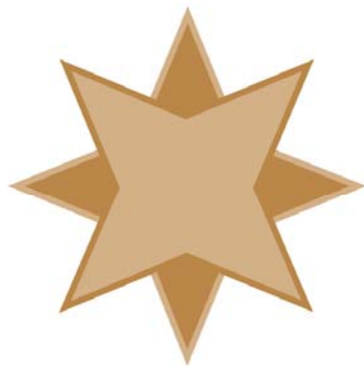
A White Paper From

GOLDSTAR SOFTWARE

www.GoldstarSoftware.com

For more information, see our web site at
http://www.goldstarsoftware.com

Obtaining an MKDE Trace of a Failing Application
Last Updated: 02/04/2021

Sometimes, in the course of working a problem with an application or service like Pervasive PSQL (now Actian Zen), it becomes imperative to capture the exact point at which an application fails, and see exactly what is going to and coming back from the database engine. This is especially important when a database application does not report the Btrieve status codes properly (or at all). The process of obtaining an MKDE trace is an important step that end-users can take to help a support technician (from Actian, Goldstar Software, or the application vendor) quickly determine the problem and devise a resolution to it.

The process is fairly simple. First, we want to make sure that nobody is using the database engine, which helps to minimize the amount of data collected. Then, run the program to the point *immediately before* where the error is expected to occur. Next, enable the tracing feature. After running the program to obtain the error, disable the trace and view the resulting file.

Let's go through it step by step here.

1. Make sure that all users are out of the database.

On a single-computer system running the PSQL/Zen Workgroup engine, this is easy -- simply exit all applications.

On a client/server environment, you should ask all users to exit their database applications and to stay out for the duration of these tests. You can also verify that all users are out by running the PSQL/Zen Monitor application on the database server and checking the Microkernel Sessions screen (Microkernel/Active Users on PSQL v11 or older) to see if anyone shows up there. If there are any latent connections, you can force them out by highlighting the connection and then clicking on the "Delete Selected Session" button.

2. Find the MKDE Tracing Settings on the Database Engine.

This can be done several ways, depending on if you are on the server or a workstation, and if you are running the Server or Workgroup engines. The instructions also differ slightly depending on the version of the database engine that you are running.

### Pervasive.SQL 2000i or Pervasive.SQL V8

On the Database Server or for Workgroup Engines: Start the Pervasive Control Center. Click the plus sign next to the *Pervasive.SQL Engines* line. Click the plus sign next to the *ServerName* line. Click the plus sign next to the *Configuration* line. Click the plus sign next to the *Server* line. Click on the *Debugging* section to open it up.
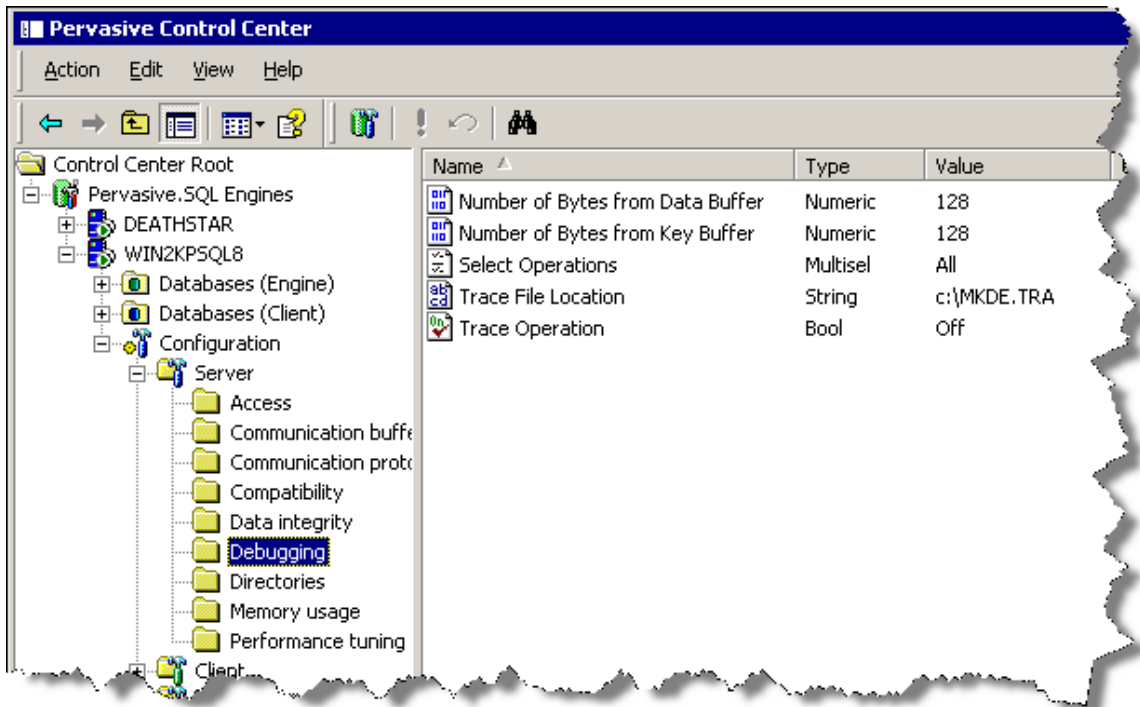
From a Workstation when using a Server Engine: Start the Pervasive Control Center. Click the plus sign next to the *Pervasive.SQL Engines* line. If you do not see the database server listed here, right-click on *Pervasive.SQL Engines* and select the line *Register New Engine*. Enter the server's computer name and press Enter to save it. Click the plus sign next to the *ServerName* line. Click the plus sign next to the *Configuration* line. You will be prompted to provide a username and password -- these should be your network administrator credentials. Click the plus sign next to the *Server* line. Click on the *Debugging* section to open it up.

## Pervasive PSQL v9 or Higher (Through Zen v14)

On the Database Server or for Workgroup Engines: Start the *Pervasive Control Center*, *PSQL Control Center*, or *Zen Control Center*. Click the *Configure Local Engine* item on the right side in the Welcome screen.
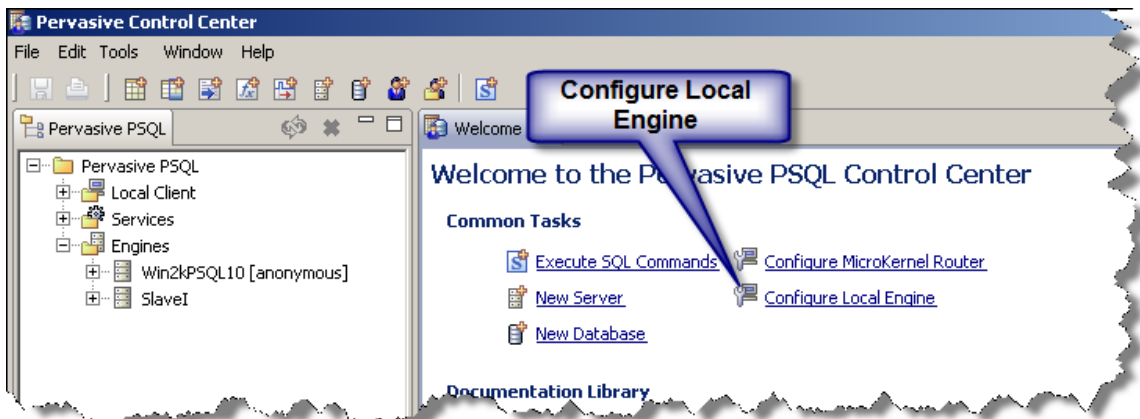

Information Provided By **Goldstar Software Inc.**
http://www.goldstarsoftware.com

Select the Debugging tab on the left side of the resulting dialog box.
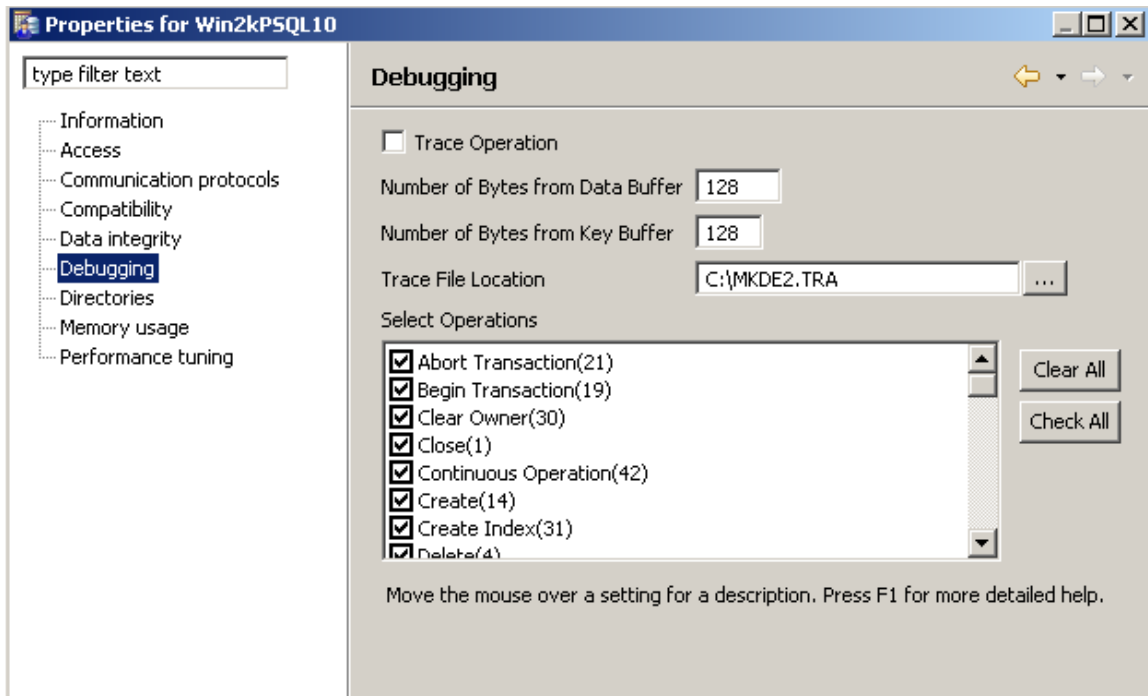
From a Workstation when using a Server Engine: Start the Control Center. If the server is not listed under the Engines item, right click and select *New...Server* to add the server to the list. Then, right-click the server and select *Properties*.
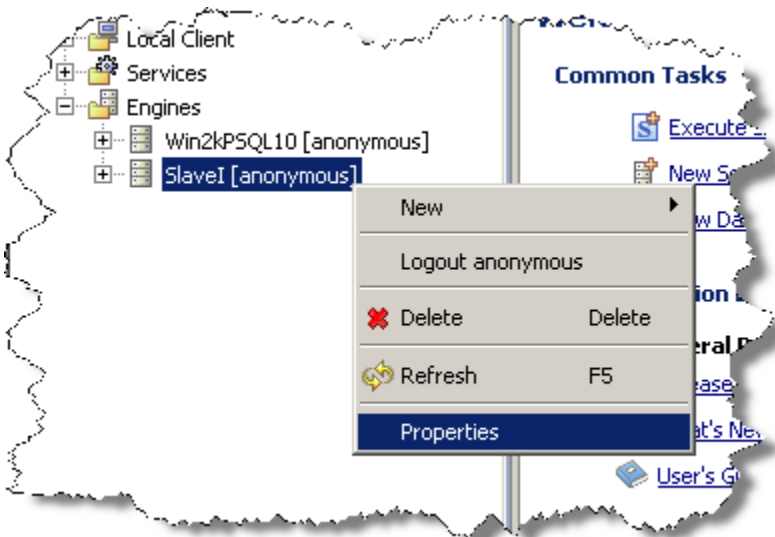
You will then see the login dialog box:
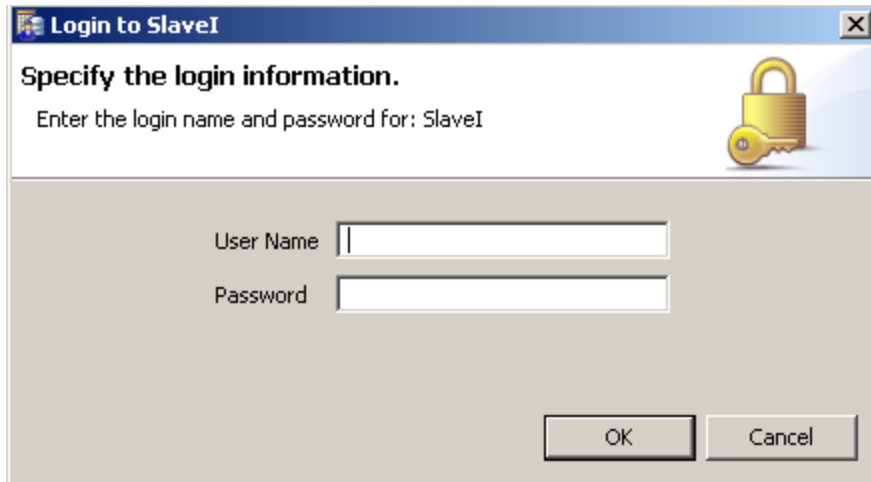
Provide your login credentials for the server, then go to the debugging tab as indicated above.

3. Verify the existing database tracing settings.

Change the settings if needed to the following values:

* ✷ Number of Bytes from Data Buffer: 128
* ✷ Number of Bytes from Key Buffer: 128
* ✷ Select Operations: All
* ✷ Trace File Location: C:\MKDE.TRA (or SYS:MKDE.TRA for NetWare)
* ✷ Trace Operation: Off

The crucial setting is the Trace File Location. Make sure that this goes to a drive with plenty of free space, and one that you will be able to access from your user login when you are done. You may wish to put this into the application data directory for easiest access. When you are done making changes, click on the red exclamation point button on the tool bar (for PSQL2000i or PSQLV8) or the Apply button on the button of the dialog box (for PSQLv9 and higher) to apply the settings to the database. Minimize the Control Center, but do NOT close it at this time.

4. Run the program to just BEFORE the error occurs.

If the error occurs when starting the application, then you need do nothing at this time. However, if the error occurs somewhere inside the application, get the application started and navigate as far as you can before you expect the error to occur. This will allow you to quickly get to the "interesting" spot prior to starting to collect the trace data. Doing so will reduce the amount of data you will need to sift through.

5. Enable MKDE Tracing.

Switch back to the Control Center. Change the Trace Operation setting to "On" (or checked) and click the exclamation point or the Apply button to apply the setting. This

starts logging ALL database calls to the trace file. Minimize the Control Center but do NOT close it.

6. Run the program to get the error.

Run your application to duplicate the error condition. When you get the error, leave the program running, but stop doing anything within it. Do not acknowledge any error dialogs at this time if you can avoid it. The goal is simply to hit the error spot without doing too much else within the database environment.

7. Disable MKDE Tracing.

Switch back to the Control Center, change the Trace Operation setting to "Off", and apply the setting one last time to stop the logging. You can now safely close the Control Center.

WARNING: Please note that if you forget this step and you leave logging turned ON in a production environment, the database will be very slow, the MKDE log will fill up ALL of your available disk space, and your server will likely crash. You MUST remember to turn this setting OFF when you are done!

8. Submit your trace file.

You can now ZIP up the MKDE.TRA file (which should be located exactly where you specified) and submit it to your support technician for follow-up.

One last thought – if your trace file is empty (or contains only the header line), then there were no database calls made in that portion of the application to this server. This could indicate that your application is talking to a different server entirely, or that it never made the connection to the server in the first place. Check the application configuration and make sure that it is talking to this server. Or, if you are still stuck, try enabling the Debug/Trace requesters instead, as this will give you visibility into exactly what the application is trying to do (and on which server).

If you still can't get it to work, contact Goldstar Software and let us work with you to help!