Creating an ODBC Data Source for Actian PSQL

A White Paper From

GOLDSTAR
SOFTWARE
www.GoldstarSoftware.com

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

Creating an ODBC Data Source for Actian PSQL

Last Updated: 06/24/2022

Many users of Actian PSQL (formerly Pervasive PSQL) see their data through Btrieve-level applications which access the data natively using low-level (and very high-performance) function calls. However, getting access to this data from other applications, such as Microsoft Access, Crystal Reports, UPS WorldShip, and others may require that you configure an ODBC data source for the database engine.

The key to setting everything up properly is to first understand a few key terms that are often bandied about by administrators. We're going to start with the concept of an ODBC DSN. A **Data Source Name** (DSN) is configured by the ODBC Administrator and links the generic ODBC access environment with a specific database engine driver and data source. Three types of DSN's are commonly used:

- **User DSN**: A User DSN is created in the local user's profile of the registry, and therefore is accessibly only to the user that created it. Anyone can create their own User DSN.
- **System DSN**: A System DSN is stored in the machine profile of the registry, and is accessible to all users of the local computer. Setting up System DSN's is an administrative function, so admin rights are needed. In a business environment, it is common to use System DSN's, so that you only have to configure each workstation once, no matter who is using it.
- **File DSN**: A File DSN stores information needed for a data source in a text file. These are not commonly used any more.

Specific to the PSQL environment, there are quite a few more terms to understand, so let's get to these next:

- **Data Folder**: Data files, often known as Btrieve files, Microkernel files, or simply application data files, are used to physically store data on the server. You will need to know the directory in which these files are stored. If you are unsure, check the Monitor or ask your developer for help finding them.
- **Data Dictionary**: The Data Dictionary Files, or DDF's, define the data structures that allow the ODBC/SQL engine to access the data files. These DDF's come in two flavors – v1 and v2. The v1 metadata has files like FILE.DDF, FIELD.DDF, etc. The v2 metadata uses PVFILE.DDF, PVFIELD.DDF, etc. If you don't have DDF's, then contact the developer to get them, as you'll otherwise be stuck.
- **Named Database, Database Name, or DBName**: A Named Database is a construct inside the engine which links a given Data Dictionary with one or more Data Folders. It is a required piece, and you must have administrative rights to configure a DBName on the server.
- **Engine DSN**: An Engine DSN is an ODBC DSN that is created directly on the computer that contains the database engine itself (i.e. the database server). It allows an ODBC application to directly access the engine on the local server.

Engine DSN's were required for older versions of the PSQL environment, though they are completely optional in newer versions and in the 64-bit ODBC driver.

- **Client DSN**: A Client DSN is an ODBC DSN that is created on a Windows workstation that is accessing data on a different computer (commonly known as the database server). This is the required component that will allow your own ODBC applications to access the database directly.

Finally, there is one more consideration in the ODBC world, namely the "bitness" or "bit level" of the ODBC Driver. In short, the bit level of your application MUST match the bit level of the ODBC driver. If you are using Microsoft Access 2.0, which is a 16-bit application, then you must have a 16-bit ODBC interface, and thus will require a 16-bit ODBC driver. (These were included from Btrieve 6.15 through PSQL V8, but removed in PSQL v9 and above.) If you are using a 32-bit application (most common yet today), then you will use a 32-bit ODBC driver. If you have a 64-bit application (like 64-bit Excel, PowerBI, etc.), then you will need to use a 64-bit ODBC driver. Note that the OLDEST version of PSQL that includes a 64-bit driver is PSQL v11. If you have an older version, then you will not be able to use 64-bit applications and ODBC.

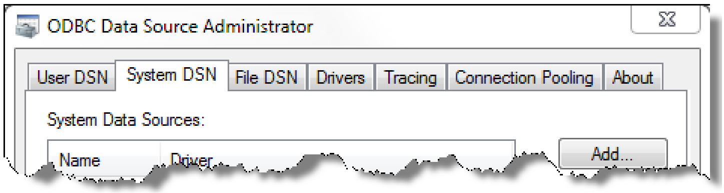
There are actually several ways to set up DSN's, depending on the software version you are using. Using the Control Center may be the quickest way to do it, but the screens look different for each engine version, so building a single document can be more difficult. To use this, check with the online manuals regarding the Control Center. Since database versions tend to share a similar interface at the native ODBC level, we will go through the ODBC Administrator to create the database.

Create an Engine DSN on the Server

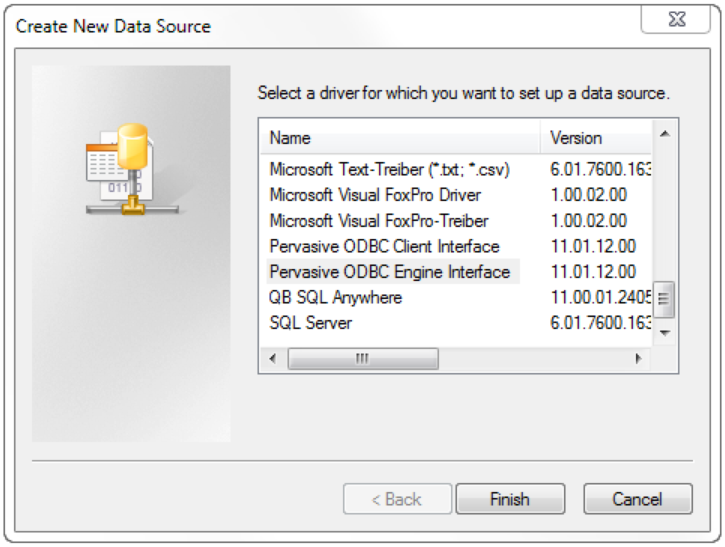
Start by logging into the database server at the console itself as a user with administrative rights. If you don't have admin-level rights, contact your server's administrator first.

Note that if your server is running Pervasive PSQL v9 (or has EVER run PSQL v9), and you are going to use Remote Desktop or Terminal Services to access your server, then you may ***also*** need to set the registry key **HKLM/Software/Pervasive Software/Utilities Interface/Settings/Restricted Access on WTS Client** to 0 (unrestricted access), or this may not work correctly.

1. Start the **32-bit ODBC Administrator** from the Windows **Control Panel**. In newer versions of Windows, this may be located in different locations, such as under **Administrative Tools**. If you are running a 64-bit operating system and a 32-bit database engine, then you will need to launch the 32-bit version of ODBC Administrator. This can be found by going to Start/Run and entering **C:\Windows\SysWOW64\ODBCAD32.EXE**. While newer operating systems will actually show 32-bit or 64-bit in the title bar, older ones will not, so if things are not working, verify the bit level of your ODBC Administrator first.



2. Under the **System DSN** tab, click **Add** to create a new DSN.

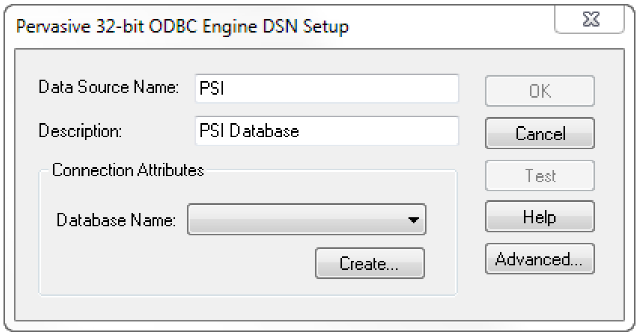


Select a driver for which you want to set up a data source.

Name	Version
Microsoft Text-Treiber (*.txt; *.csv)	6.01.7600.163
Microsoft Visual FoxPro Driver	1.00.02.00
Microsoft Visual FoxPro-Treiber	1.00.02.00
Pervasive ODBC Client Interface	11.01.12.00
Pervasive ODBC Engine Interface	11.01.12.00
QB SQL Anywhere	11.00.01.2405
SQL Server	6.01.7600.163

3. From the resulting *Create New Data Source* dialog box, select the **Pervasive ODBC Engine Interface** option and click **Finish**. (Again, if you don't see this name, you may be in the 64-bit ODBC Administrator.)

4. In the *Pervasive ODBC Engine DSN Setup* dialog box, enter the name of your new DSN as you want it known on the server, and optionally give it a more detailed name, as shown below.



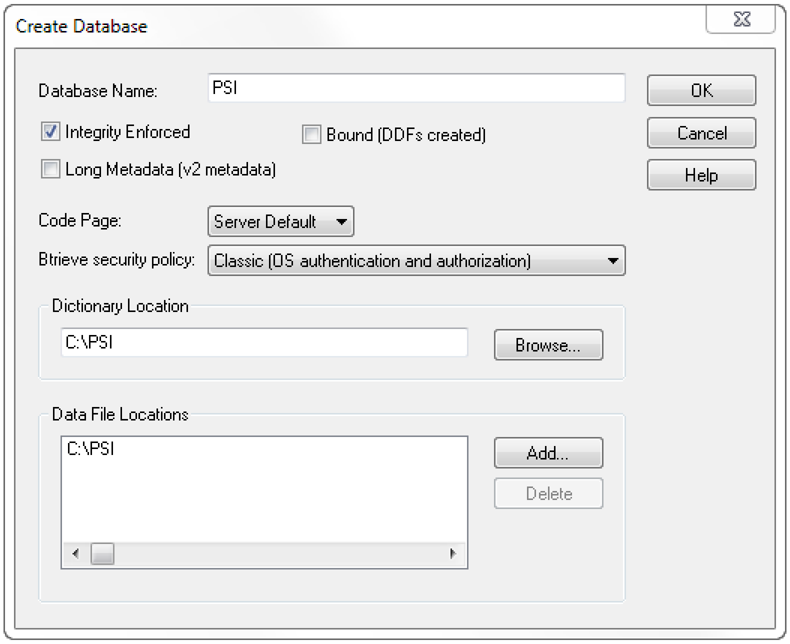
Data Source Name: PSI

Description: PSI Database

Connection Attributes

Database Name:

5. Next, we need to create the Named Database, so click on the **Create…** button. This will take you to the *Create Database* dialog box.



Create Database

Database Name: PSI

☑ Integrity Enforced

□ Bound (DDFs created)

□ Long Metadata (v2 metadata)

Code Page: Server Default

Btrieve security policy: Classic (OS authentication and authorization)

Dictionary Location: C:\PSI

Data File Locations: C:\PSI

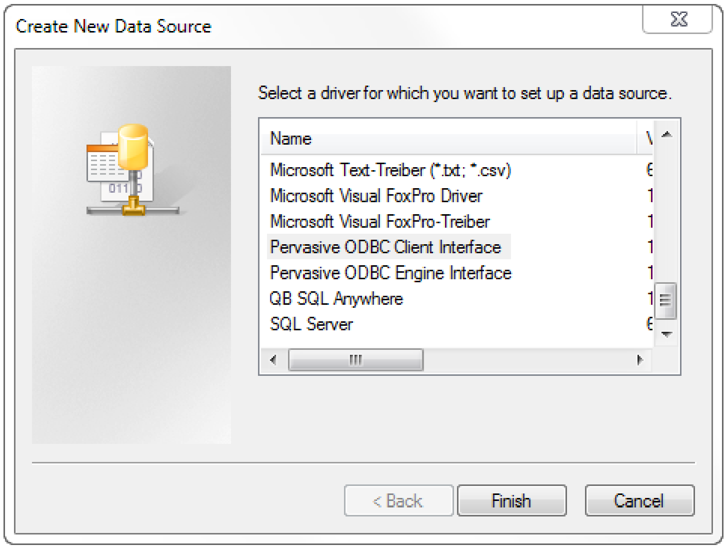
6. On this screen, you must first enter the database name. This name must be unique on the server, but it can (and probably should) be the same as the Engine DSN name, entered in the previous step. For most databases, the default settings of **Integrity Enforced** (On), **Bound** (Off) and **Btrieve Security** (Classic) are just fine. If you think you may need to change these, see your product manuals for more information.
7. In the **Dictionary Location** field, enter the pathname to your DDF files. If you do not have DDF's, this process will create EMPTY DDF's, which may not be very helpful, unless you know how to create the definitions yourself. Be sure to double-check the state of the Long Metadata checkbox and set this based on the metadata version (i.e. DDF's) you are using.
8. If your data files are NOT in the same directory as the DDF's, add the database directory paths to the **Data File Locations** box.
9. Click **OK** to save out of this screen and return to the previous one. Click **OK** one more time to complete the Engine DSN. You are now done with the Engine DSN, and you can log off of the server.

Create the Client DSN on the Workstations

Next, we need to create the Client DSN's on the workstations, so let's go to the client PC.

1. As mentioned above, you need to know the bit level of the application you are running. If you are running a 32-bit application, then you will need to create a 32-bit ODBC DSN. If you are running a 64-bit application, then you will need a 64-bit DSN. Start the correct version (32-bit or 64-bit) of the **ODBC Administrator** as described above, from the Windows **Control Panel**, or the Control Center.

2. Under the **System DSN** tab, click **Add** to create a new DSN. Again, this will require admin rights to the workstation. If you lack these rights, create a User DSN instead.

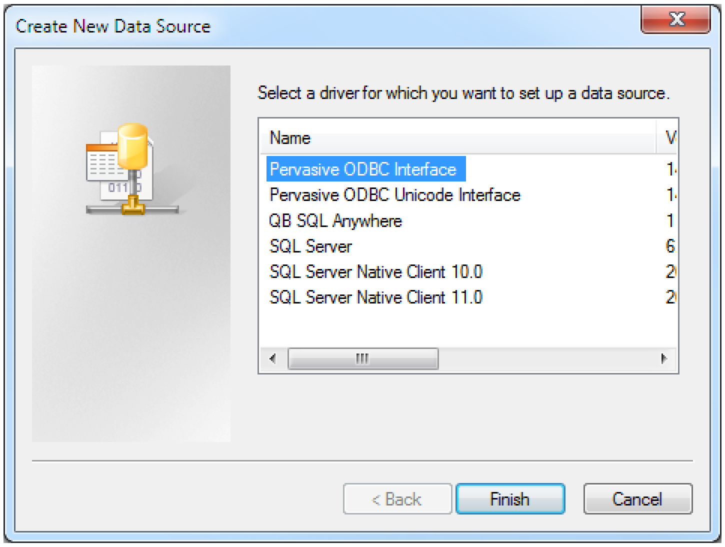


Create New Data Source

Select a driver for which you want to set up a data source.

Name	V
Microsoft Text-Treiber (*.txt; *.csv)	6
Microsoft Visual FoxPro Driver	1
Microsoft Visual FoxPro-Treiber	1
Pervasive ODBC Client Interface	1
Pervasive ODBC Engine Interface	1
QB SQL Anywhere	1
SQL Server	6

Note: If you are running a 64-bit ODBC application and setting up a 64-bit ODBC data source, then you will see s slightly different name, as shown here:



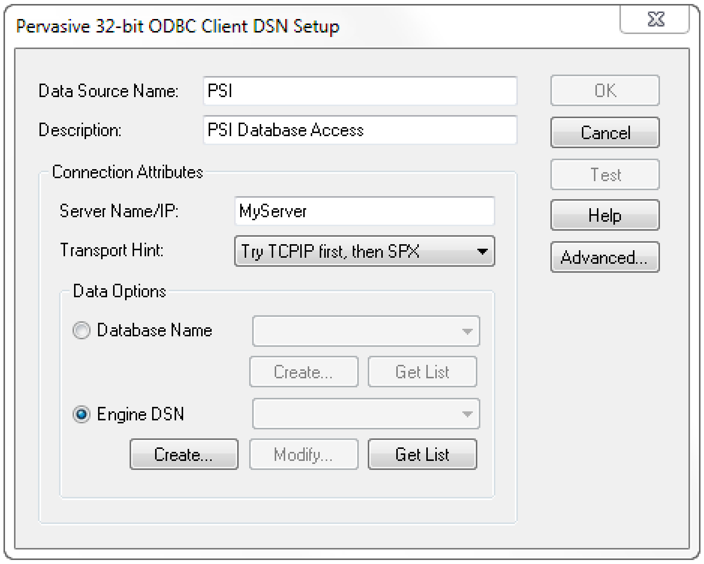
Create New Data Source

Select a driver for which you want to set up a data source.

Name	V
Pervasive ODBC Interface	1
Pervasive ODBC Unicode Interface	1
QB SQL Anywhere	1
SQL Server	6
SQL Server Native Client 10.0	2
SQL Server Native Client 11.0	2

3. From the resulting *Create New Data Source* dialog box, select the **Pervasive ODBC [Client] Interface** option and click **Finish**.

4. In the *Pervasive ODBC Client DSN Setup* dialog box, enter the name of your new DSN as you want it known on the workstation, and optionally give it a more detailed description, as shown below.



Pervasive 32-bit ODBC Client DSN Setup

Data Source Name: PSI

Description: PSI Database Access

Connection Attributes

Server Name/IP: MyServer

Transport Hint: Try TCPIP first, then SPX

Data Options

○ Database Name

● Engine DSN

5. Next, enter the name of the server into the **Server Name** box in the middle of the screen. If you know that you are using a single network protocol (usually TCP), then change the **Transport Hint** to **Try TCP Only**.
6. Decide if you want to use an Engine DSN or a Database Name to connect. In our case, we just created an Engine DSN, so we can use that. (With a 64-bit DSN, you will ONLY have the option of using a Database Name.) Click on the appropriate **Get List** button, and a list of server-side DSN's or named databases will be loaded into the drop box. Open the drop box and select one that you just created on the server.
7. After selecting the target and reviewing the rest of the data, click **OK** to close this dialog box.

You're now done with the Client DSN!

Test the DSN

Finally, you may wish to test your DSN. Start your ODBC application, point it to the newly created data source, and verify that it works correctly.

Note that if your server name changes, you may need to alter the server name field on each workstation with a Client DSN installed on it. However, if you simply move data from one directory to another on the server, you must only change the Engine DSN or the Named Database (on the server) to the new path, and you'll be all set!

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