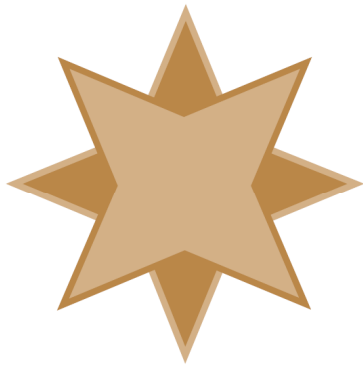


Creating an ODBC Data Source for Zen v15 on Windows

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 Zen v15 on Windows

Last Updated: 09/10/2021

Many users of Actian Zen (formerly 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 & Excel, Crystal Reports, UPS WorldShip, and others may require that you configure an ODBC data source for the database engine.

Previous versions of this paper used the ODBC Administrator to accomplish these tasks, but with Zen v15, this task is easier to do from the Zen Control Center, so we're going that route instead.

Pieces and Parts of an ODBC Connection

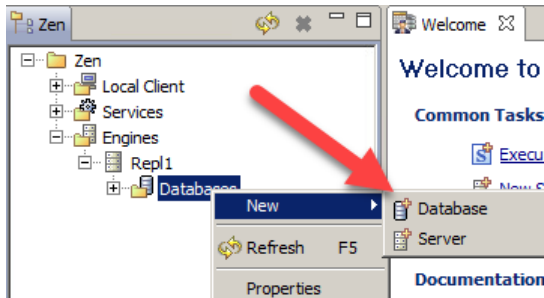
Before we get into the specifics, let's review some of the parts involved:

- **Data Files:** Your application will have *database files* of some kind. They may be named with a specific extension, such as BTR, MKD, DAT, V90, etc., or they may have no extension. You will need to know where these are located, and if the files are spread over multiple directories, you'll need to know that, too.
- **Data Dictionary Files:** The *data dictionary* is used to tell the SQL engine how to interpret the data files, and is stored in a series of DDF files, including FILE.DDF, FIELD.DDF, INDEX.DDF and so on. If you have v2 metadata, you may instead have PVFILE.DDF, PVFIELD.DDF, etc. (If you don't have any DDF's, then you may need to create them first, which is the subject of a different paper.) DDF's are typically stored in the same folder as the data files, but this is not required.
- **Named Database:** A *named database*, also referred to as a *database name* or a *DBName*, is a construct created within the database engine which links the DDF location and the data file location(s) into a single logical component. Named databases are stored with the engine in a file called DBNAMES.CFG, and it is possible to migrate this file forward from one system to another if needed.
- **ODBC Engine DSN:** When constructing an ODBC data source name (DSN) on the computer that is also running the engine, you create an *ODBC Engine Data Source Name (DSN)* which links the ODBC components with the named database. With newer versions the Engine DSN is optional, and the Engine DSN is NOT used with the 64-bit ODBC drivers at all.
- **ODBC Client DSN:** When setting up a workstation with ODBC access to a database, a Client DSN is used. There are both 32-bit and 64-bit Client DSN's, and you'll set up the one that matches the bit level of the application you are running. For example, a 64-bit version of Excel would use a 64-bit DSN, and a 32-bit version of Excel would use a 32-bit DSN. Older versions required that the Client DSN point to the Engine DSN on the server, but newer versions allow the Client DSN to point directly to the Named Database on the server, instead.

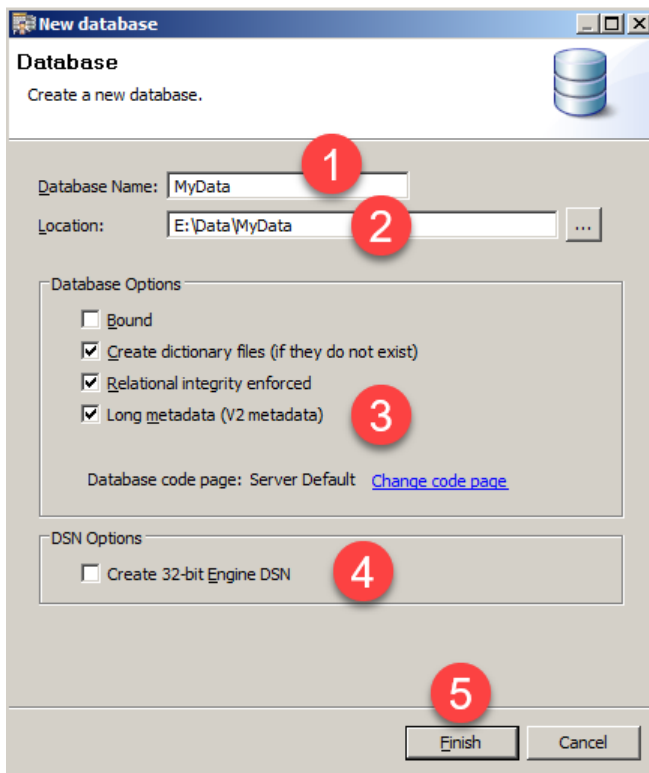
Create the DBName (and Optional Engine DSN) on the Server

We always start on the database server, since this process needs to be done only one time. It is best done directly on the server console (logged in as Administrator) whenever possible, as this avoids needing additional authorizations. If you read the text above, you'll know that you need to know the location of the DDF's and data files. (If you skipped it, please go back and read page 1 now, as it will come in handy later on!)

So, let's create the new database now. Start by launching the Zen Control Center application. In the left-hand tree view, drill down into the computer name and right-click on Databases, then select New, Database:



A new dialog box will pop up to create the new database:



Enter the required data:

1. The database name is a user-entered name which can be anything. For best results, keep it short and avoid embedded spaces. The engine will change it to all caps for you, so case is not important.

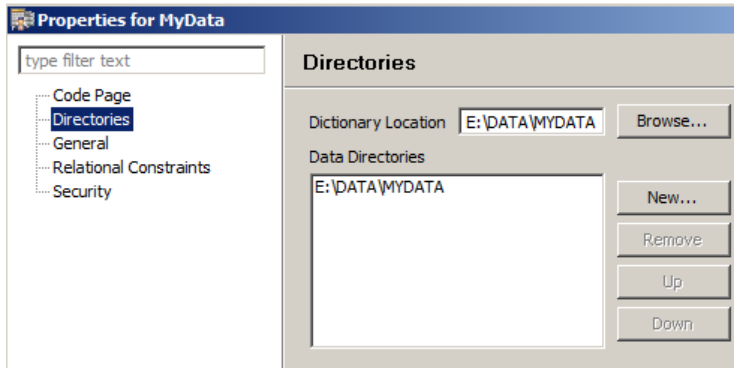
Information Provided By **Goldstar Software Inc.**

<http://www.goldstarsoftware.com>

2. The database location should be the location of the DDF files. If your database files are in the same location, then this will be easy. If your data files are scattered in other directories, then see the section below about adding additional paths to the named database.
3. Here's a quick list of options and when to change them:
 - a. **Bound:** This forces the database engine to ALWAYS require accessing data files through the SQL interface. Most users do NOT want this, so leave the default of UNCHECKED here to avoid issues.
 - b. **Create dictionary files:** This switch is used to create new DDF files if there are none in the directory already. If you already have DDF's, you should uncheck this. If you leave this checked and your DDF's have security enabled, the process may fail.
 - c. **Relational integrity enforced:** This setting tells the database engine to enforce RI constraints. Most PSQL/Zen databases do not use RI, so you can leave this CHECKED, which is the default.
 - d. **Long metadata:** This checkbox determines if you are going to use the older DDF format (e.g. FILE.DDF) or the newer format (e.g. PVFILE.DDF). If you already have DDF's, verify the format and set this box accordingly. Note that the default for v15 is CHECKED (meaning v2 metadata), but very few databases use V2 metadata right now, so you may need to UNCHECK this instead if your application is older. There is no way to change this later other than deleting and recreating the DBName, so make sure this is correct now!
4. Check this box to create a 32-bit Engine DSN at the same time (and with the same name) as the DBName. This is needed only if you are planning on running 32-bit ODBC applications on this computer or if you have older ODBC clients which require the Engine DSN to properly function.
5. Click Finish to complete the dialog box and create the DBName.

Modifying a Named Database on the Server

If you need to modify the properties of a named database, then you can right-click on the database in the Zen Control Center and select Properties. For the purposes of this discussion, we will only cover the need to add more directories to the DBName, so click on the **Directories** tab to get this dialog box:

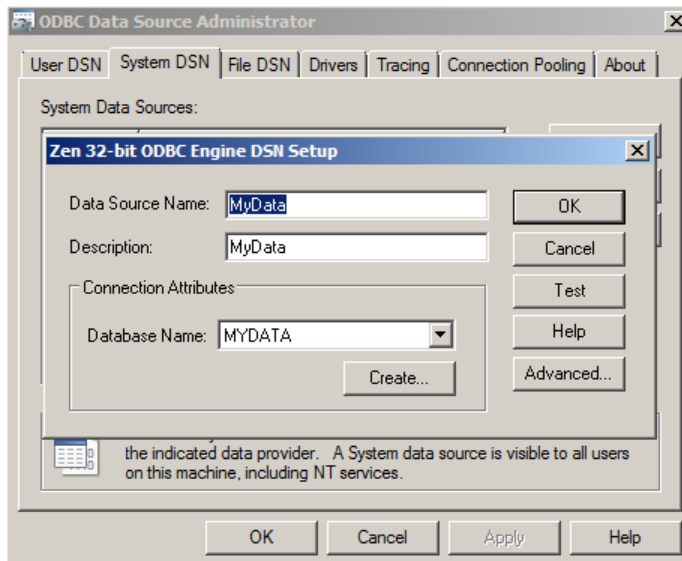


Here you can see the first data directory that was automatically entered when we created the database from the Control Center. If you need to add additional directories to allow the engine to find the data files, click **New...** and add them as needed. Remember that the engine will search for the database files in the order in which the paths are entered here, so use the **Up** and **Down** buttons to arrange the folders as needed. For best results, leave the DDF path first, and make sure that you **NEVER** have the same file name located in two folders at the same time, which will avoid confusion.

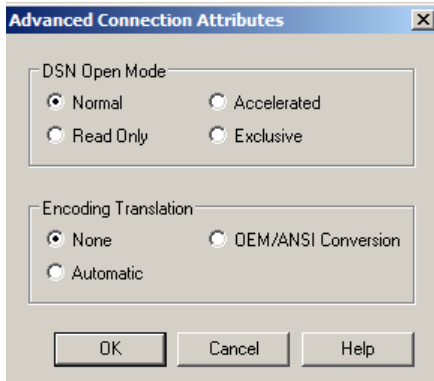
From the **General** page, you can change the settings for *Bound* and *Integrity enforced* and you can view (but not change) the setting for *Long metadata*. From the **Security** page, you can set security options, but this is again outside the scope of this paper.

Modifying a 32-bit Engine DSN on the Server

If you need to modify the properties of a 32-bit Engine DSN, then use the Zen Control Center *Tools* menu to launch the *32-bit ODBC Administrator*, click on the *System DSN* tab, and then double-click on the DSN you want to edit to get this dialog box:



Here, you can rename the DSN or change its text description (which doesn't show up in very many places). You can also change the Database Name that is attached to this Engine DSN, or you can click on the **Advanced...** button to get to the next dialog box:

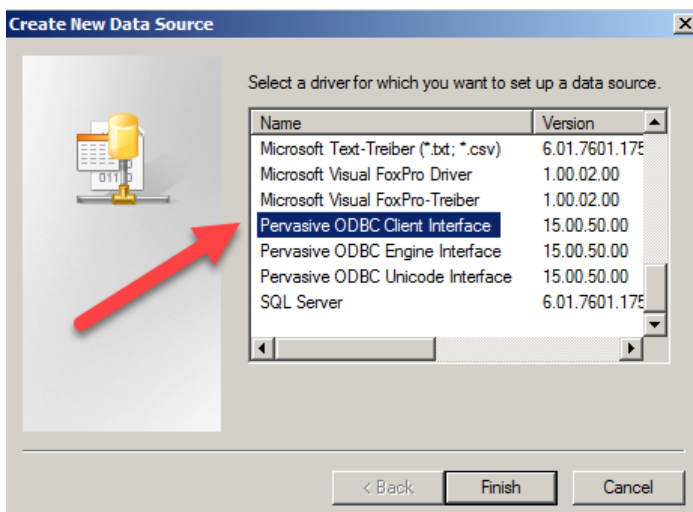


Here, you can force the Engine DSN to be ReadOnly – a very useful feature to prevent people from messing with your data!

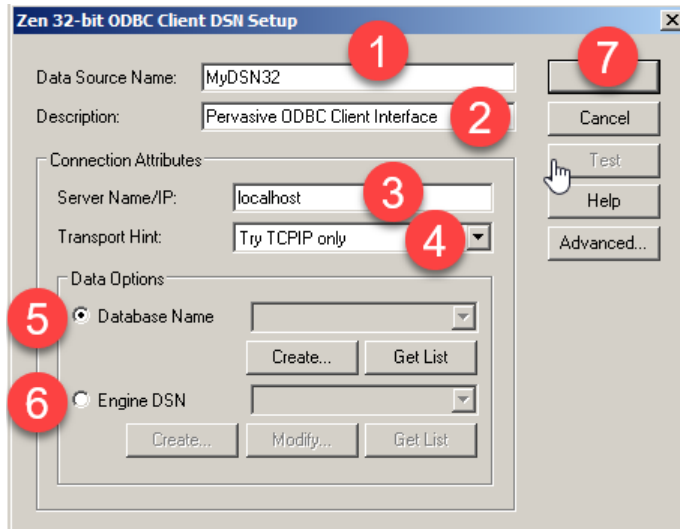
Once the server is configured correctly, it is now time to create the Client DSN on your workstation (or the server, if you are running applications from the server itself). Again, there are two bit levels for ODBC – 32-bit and 64-bit – and you need to confirm which bit level your application is written in. Most applications are still 32-bit, but the number of 64-bit applications is certainly growing rapidly, so check with your app vendor if you're not sure, then continue with the appropriate section below.

Creating a 32-bit Client DSN

To create a 32-bit ODBC DSN, you must use the 32-bit ODBC Administrator. The easiest way to find this tool is from the Zen Control Center's *Tools* menu. Launch this tool, select the *System DSN* tab, and click **Add** to get to the following dialog box:



Select the **Pervasive ODBC Client Interface** driver and click **Finish** to get to the next dialog box. Note that if you do not see this exact name, you may be in the 64-bit ODBC Administrator! In that case, close that window and re-open the 32-bit module.

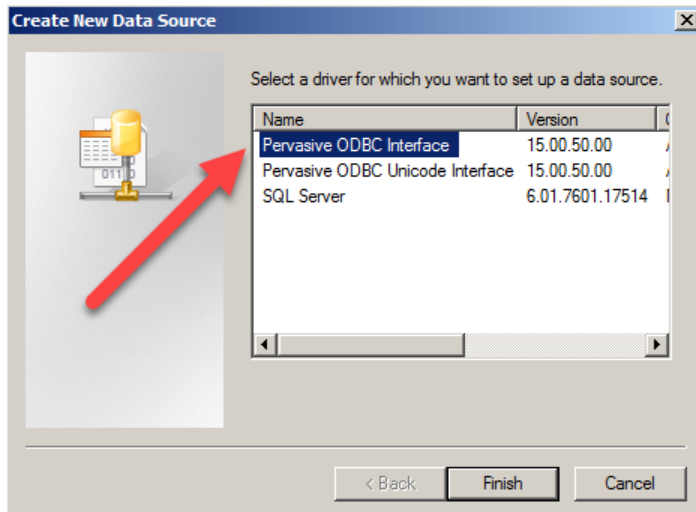


Complete this dialog box as needed:

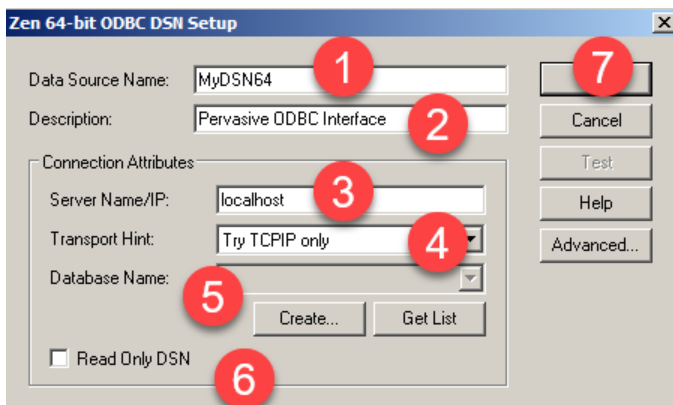
1. The **DSN Name** can be set here. If you need both 32-bit and 64-bit DSNs, then you may wish to distinguish them accordingly.
2. The **Description** field can be anything you like.
3. Enter the **Server Name** (or IP address) here. Keep the default of localhost if you are working directly on the server.
4. The **Transport Hint** is an old setting that has only one option now, so leave it.
5. If you are connecting to a **Database Name** on the server, select this option, click **Get List** to retrieve the list of DBNames from the server, and select one from the drop-down list.
6. If you are connecting to an **Engine DSN** on the server, select this option, click **Get List** to retrieve the list of DSN's from the server, and select one from the drop-down list.
7. Click OK to close this box and create the DSN.

Creating a 64-bit Client DSN

To create a 64-bit ODBC DSN, you must use the 64-bit ODBC Administrator. The easiest way to find this tool is from the Zen Control Center's *Tools* menu. Launch this tool, select the *System DSN* tab, and click **Add** to get to the following dialog box:



Select the **Pervasive ODBC Interface** driver and click **Finish** to get to the next dialog box. Note that if you do not see this exact name, you may be in the 32-bit ODBC Administrator! In that case, close that window and re-open the 64-bit module.



Complete this dialog box as needed:

1. The **DSN Name** can be set here. If you need both 32-bit and 64-bit DSNs, then you may wish to distinguish them accordingly.
2. The **Description** field can be anything you like.
3. Enter the **Server Name** (or IP address) here. Keep the default of localhost if you are working directly on the server.
4. The **Transport Hint** is an old setting that has only one option now, so leave it.
5. Click **Get List** to retrieve the list of **Database Names** from the server, and select one from the drop-down list.
6. If you want to limit the user to a read-only connection check this box.
7. Click **OK** to close this box and create the 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 need 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!