**Configuring TSTool to interface with a SQL Server Database using ODBC**

**Example using State of Colorado HydroBase Database**

Last updated: 2014-12-17

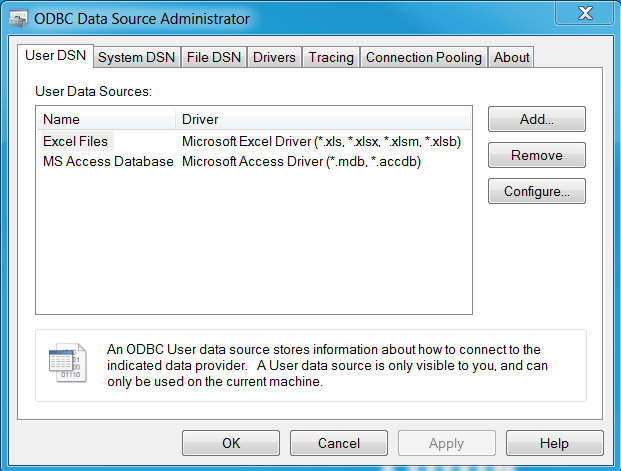
TSTool can be configured to interface with a Microsoft SQL Server database using an Open Database Connectivity Data Source Name (ODBC DSN). When configuring the ODBC DSN success will depend on:

1. The ODBC configuration tool being run
2. The ODBC drivers that are installed and recognized by the configuration tool
3. The SQL Server database that is selected
4. The Java version that is being used to run TSTool

Unfortunately, the above combination can result in mismatches between 32-bit and 64-bit databases, drivers and software, and Microsoft ODBC 64-bit software is not seamlessly compatible with 32-bit versions. This document attempts to explain and help resolve issues. The information is presented by topic area in common order that configuration and troubleshooting would occur and troubleshooting may involve one or more topics.

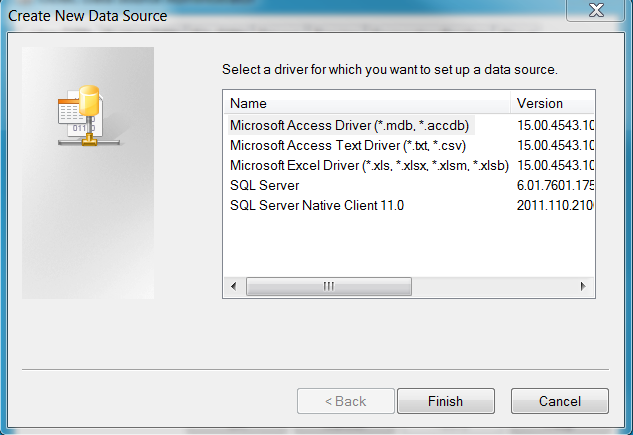
**Defining an ODBC DSN**

An ODBC DSN is generally configured using the “Data Sources (ODBC)” tool. Use the Start menu search tool to find. By default, the version of the configuration tool that is run will match the operating system (e.g., 64-bit tool will be run on a 64-bit Windows 7 installation – even if Office or SQL Server software installed the 32-bit version – is this true?). However, this may result in incompatibilities with the database driver. As a first step, assume that the default tool can be used to define an ODBC DSN. First, start the tool, which will show a dialog as follows. User DSN are accessible only to a specific user and System DSN are accessible to all users on the computer.



**Figure 1 – Listing of User ODBC DSN**

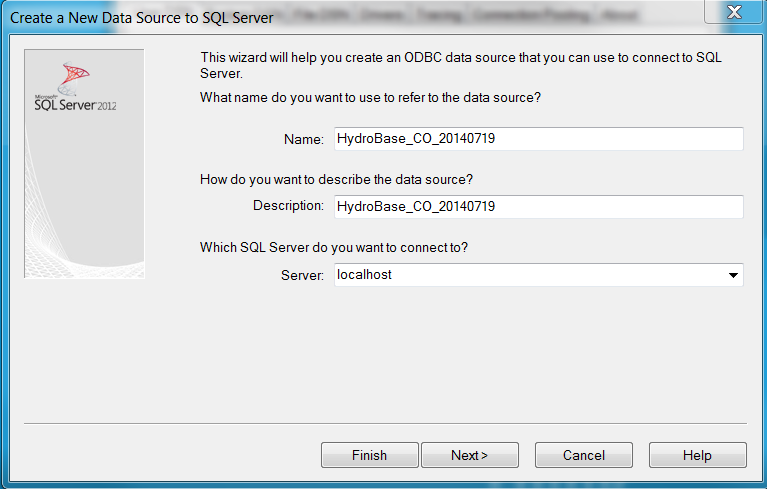
The information shown in Figure 1 is a bit confusing in that it lists defined ODBC DSN (in this case none are defined) as well as generic drivers, but it does not list all drivers. To define a new connection for a SQL Server database, click on **Add…**, which shows a dialog as shown in Figure 2 (or double-click on the data source type in Figure 1, for example MS Access Database if that were appropriate).



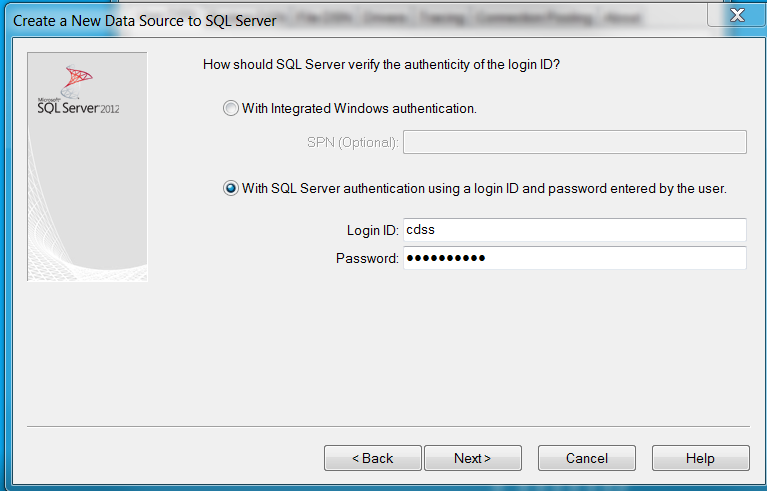
**Figure 2 – Create New Data Source (ODBC DSN) Dialog**

Figure 2 illustrates that in this case, the computer (which happens to be running 64-bit Windows 7 appears to have 32-bit driver (“SQL Server” which has a date of 11/20/2010) and 64-bit ODBC driver (“SQL Server Native Client 11.0”, which has a date of 2/11/2012) installed for SQL Server. Don’t know for sure that the older one is 32-bit but suspect it is.

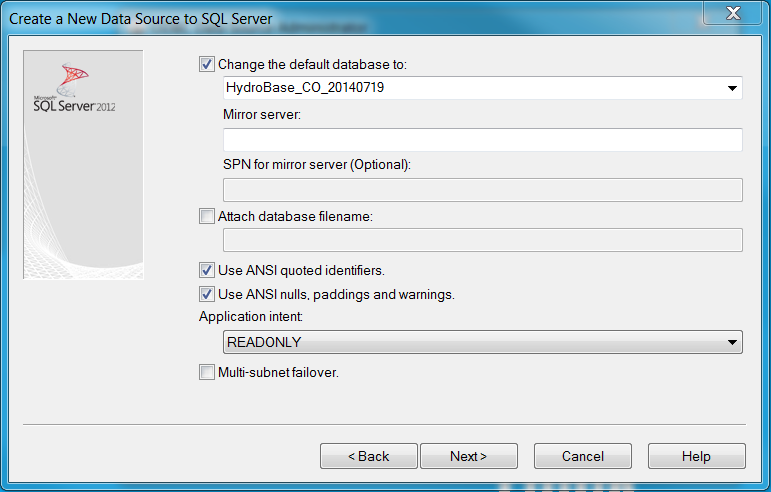
Select the “SQL Server Native Client 11.0” driver for use with the latest HydroBase on Windows 7 64-bit, which uses SQL Server 2012. The following dialog will be displayed:



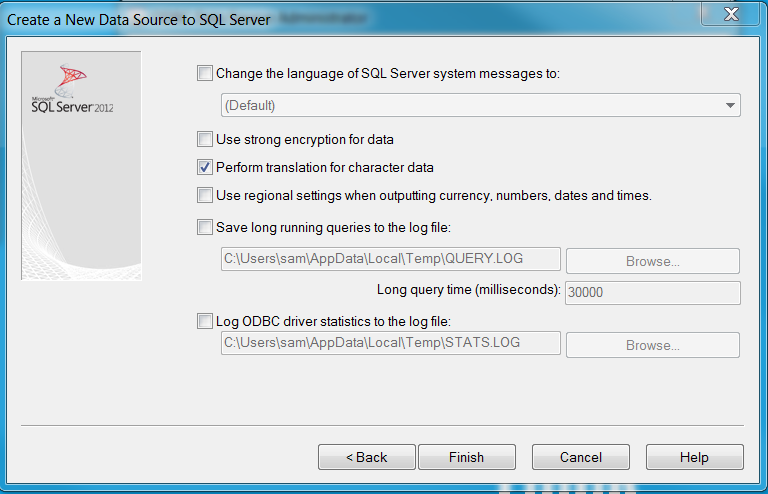
Fill out similar to shown above. A server can be selected from the sever list. However, it may spin for awhile and not return anything (does this require that the SQL Server Browser service is running to list? Looking at local services the “SQL Server Browser” is running but also see “SQL Server (CDSS)”, “SQL Server Agent (CDSS)”, and “SQL Server VSS Writer” so not sure why there is no “SQL Server Browser (CDSS)” ). Enter the special name “localhost\CDSS” to connect to the local CDSS instance of SQL Server. Press ***Next >*** to continue.



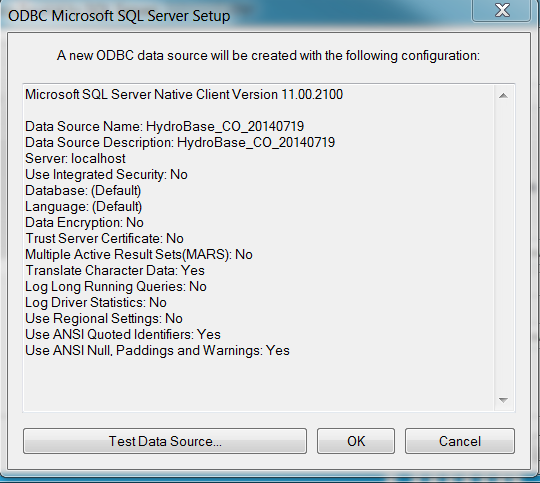
The above dialog will indicate how to authenticate the login to the database. This depends on how the database was set up. If the HydroBase installation only completed the first step, then Windows Authentication may be used. Select the ***With Integrated Windows authentication*** choice in the dialog shown below. **Note that this may give access to all tables and views, which is not the intent of the HydroBase distribution, which focuses on views and specific stored procedures visible to the service account**. If the HydroBase installation completed successfully, then accounts may have been set up and SQL Server authentication is necessary. In this case select With SQL Server authentication using a login ID and password entered by the user. Use a login credentials of cdss and cdss%tools, which are the service account that TSTool uses to access HydroBase. Note that this will give access only to tables, views, and procedures configured for the “cdss” user. Press ***Next >*** to continue.



If the above does not list choices for the default database then there is probably something wrong with the server name or authentication so back up and troubleshoot the problem. Note that the ***Application intent***: is set to READONLY to avoid editing HydroBase. Press ***Next >*** to continue.



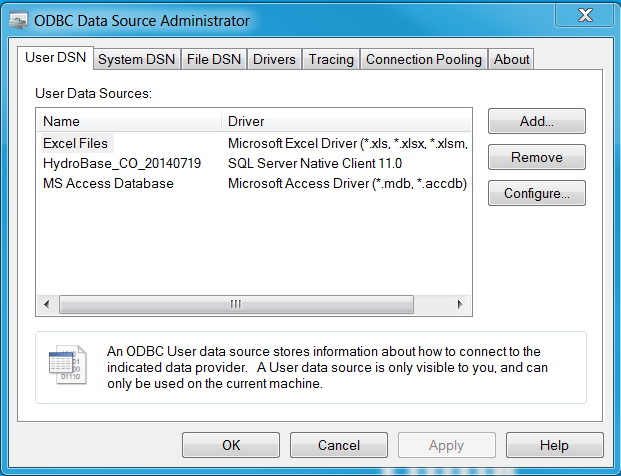
Press ***Finish***.



Press ***Test Data Source…*** to test. If it passes press ***OK***.

|  |  |
| --- | --- |
|  |  |

Press OK a couple of times to get back to the main ODBC Data Source Administrator tool, which should look similar to the following:



Press ***OK***.

**Datastore Configuration**

A TSTool datastore can be configured to interface with a SQL Server database using the GenericDatabaseDataStore or a HydroBaseDataStore datastore type. In the former case, TSTool will only be able to perform generic database datastore operations while in the latter TSTool will assume that the connection will behave as a HydroBase database and intelligent features will be provided. To enable as datastore, include the following in the TSTool.cfg:

[DataStore:HydroBase]

ConfigFile = "HydroBase.cfg"

The GenericDatabaseDataStore datastore configuration file would be similar to:

# Configuration information for CWCB NCNA Access database

# using generic database datastore.

#

# The user will see the following when interacting with the datastore:

#

# Name - datastore identifier used in applications, for example as the

# input type information for time series identifiers (usually a short string)

# Description - datastore description for reports and user interfaces (short phrase)

#

# The following are needed to make database connections in the software

#

# Type - must be GenericDatabaseDataStore

# DatabaseEngine - the database software (SqlServer is current standard)

# DatabaseServer - IP or string address for database server, with instance name

# (e.g., "localhost\CDSS" can be used for local computer)

# DatabaseName - database name used by the server (e.g., HydroBase\_CO\_20120722)

# SystemLogin - service account login (specify for HBGuest account)

# SystemPassword - service account password (specify for HBGuest account)

#

Enabled = True

Type = "GenericDatabaseDataStore"

Name = "HydroBase"

Description = "State of Colorado HydroBase"

DatabaseEngine = "SqlServer"

# Local SQL Server Express installation...

OdbcName = "HydroBase\_CO\_20140719"

# Login and password not needed since provided in the ODBC DSN

The HydroBaseDataStore datastore configuration file would be similar to the following (only the Type property is different):

# Configuration information for CWCB NCNA Access database

# using generic database datastore.

#

# The user will see the following when interacting with the datastore:

#

# Name - datastore identifier used in applications, for example as the

# input type information for time series identifiers (usually a short string)

# Description - datastore description for reports and user interfaces (short phrase)

#

# The following are needed to make database connections in the software

#

# Type - must be GenericDatabaseDataStore

# DatabaseEngine - the database software (SqlServer is current standard)

# DatabaseServer - IP or string address for database server, with instance name

# (e.g., "localhost\CDSS" can be used for local computer)

# DatabaseName - database name used by the server (e.g., HydroBase\_CO\_20120722)

# SystemLogin - service account login (specify for HBGuest account)

# SystemPassword - service account password (specify for HBGuest account)

#

Enabled = True

Type = "HydroBaseDataStore"

Name = "HydroBase"

Description = "State of Colorado HydroBase"

DatabaseEngine = "SqlServer"

# Local SQL Server Express installation...

OdbcName = "HydroBase\_CO\_20140719"

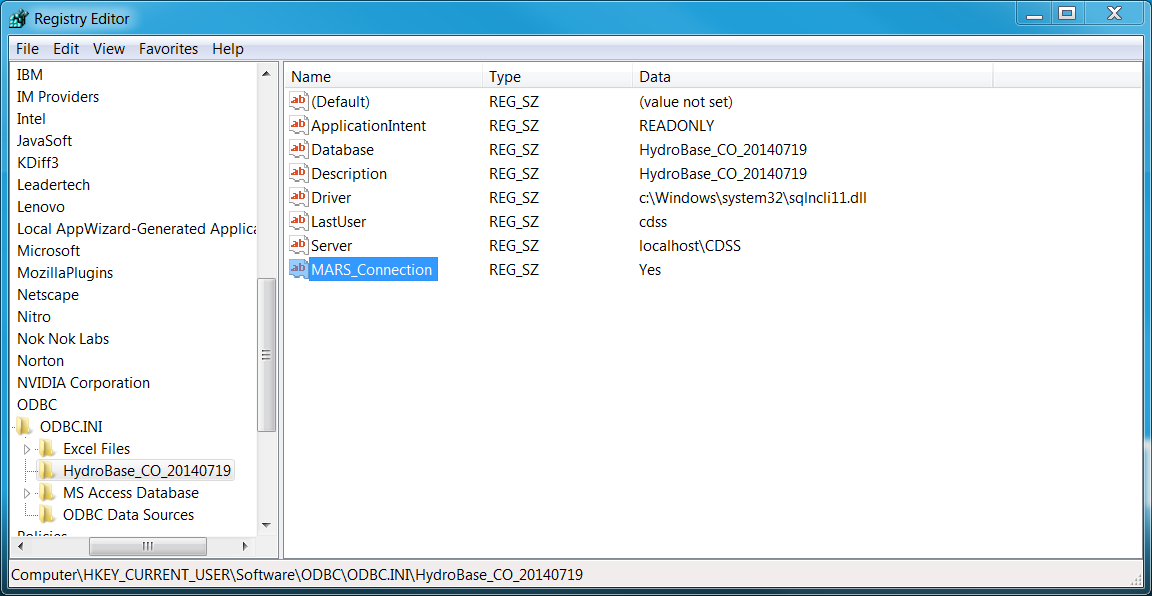
# Login and password not needed since provided in the ODBC DSN

**Troubleshooting DSN Registry Settings**

The above approach configuring a HydroBaseDataStore resulted in errors in TSTool about not being able to do queries because the database is busy with previous commands. The following article was found:

http://serverfault.com/questions/302169/odbc-sql-server-how-do-i-turn-on-multiple-active-result-setsmars-for-a-syst

Editing the registry with “regedit” as follows solved the problem. Search for the ODBC name that was assigned to find the registry settings to edit. An attempt was made to determine how to set the property in the connection URL but no documentation could be found.



The following notes may not be relevant. Need to test out some SQL Server ODBC connections first.

**Java Version**

The Java version distributed with TSTool has traditionally been the 32-bit operating system version and will run on 32-bit and 64-bit systems. However, 64-bit TSTool installations are distributed with 64-bit Java and will be indicated in the installer name. Need to indicate here how to use TSTool Help About to figure out whether 32-bit or 64-bit Java runtime is being used.

**Troubleshooting TSTool**

The following message may be output to the TSTool logfile during datastore setup if the ODBC DSN configuration is 32-bit/64-bit incompatible:

Caused by: java.sql.SQLException: [Microsoft][ODBC Driver Manager] The specified DSN contains an architecture mismatch between the Driver and Application

This error is also displayed if the 64-bit ODBC DSN administration tool is run when Access 32-bit is installed and “MS Access Database” is double-clicked on in Figure 1.

To resolve, verify that the following components are the same (32-bit or 64-bit):

* Java runtime (check version being used in development and TSTool installer)
* ODBC DSN connection (use the correct configuration tool)
* Microsoft Access executable (check the Process Manager for \*32 next to the process name to indicate 32-bit).

So… is it possible to run 32-bit TSTool (the current distributable) with 64-bit Access? Maybe not. To work-around, replace the TSTool jre\_16 folder with a 64-bit Java Runtime Environment (JRE). In the future, 64-bit TSTool installers for Windows will be made available.