GEOLOGICAL SURVEY OF CANADA
OPEN FILE 9231

The Canadian Database of Geochemical Surveys — an archival copy in SQL Server bacpac format

S.W. Adcock

2025

Canada

GEOLOGICAL SURVEY OF CANADA
OPEN FILE 9231

The Canadian Database of Geochemical Surveys — an archival copy in SQL Server bacpac format

S.W. Adcock

2025

This publication is available for free download through the NRCan Open Science and Technology Repository (https://ostrnrcan-dostrncan.canada.ca/).

Publications in this series have not been edited; they are released as submitted by the author.

Contents

List of Figures

Introduction

The Canadian Database of Geochemical Surveys (CDoGS) is a large relational database holding information about geochemical surveys carried out across Canada from the 1950s to the present. The database contains basic metadata for over 1600 surveys, including links to publications and online resources. It also includes analytical data for 300 surveys, stored in a standardized format. Detailed information about CDoGS can be found in Adcock and Spirito (2024), and references cited therein. This Open File is the first in a planned series of Open Files that will document every aspect of the database and the associated website (https://geochem.nrcan.gc.ca).

Open File Contents

This Open File contains three main files:
1. cdogs.bacpac (596 Mb)
    a. cdogs.bacpac contains all of the data within the database, in a highly compressed format. The full database is implemented in Microsoft SQL Server 2022. It occupies approximately 15 Gb of disk space, but much of this space is taken up by indexes. cdogs.bacpac was created using a utility program called SQLPackage.exe, which is freely available from Microsoft. The contents can be restored back to a SQL Server database using the same SQLPackage.exe program.
2. cdogs_schema.pdf (280 Kb)
    a. cdogs_schema.pdf is a large pdf diagram (96" x 48"), which shows the structure of the database. It was created using SQL Server Management Studio v20.
3. cdogs_db_create.sql (233 Kb)
    a. cdogs_db_create.sql is a SQL Server script file, which will create an empty copy of the database.

A cookbook for working with the bacpac data file

SQL Server is not free, open source software. Nevertheless, Microsoft offers a free "Developer Edition", which allows you to work with a personal SQL Server database. Hence, you can use the cdogs.bacpac file to create your own copy of the SQL Server database. You can then work directly with that database, using software such as SQL Server Management Studio, or you can transform the data into some other format that may be better suited to your needs and expertise.

The following "cookbook" is intended to give you some guidance in how to use the bacpac file. The instructions apply to an ordinary computer, running Microsoft Windows 10 or 11.

Step 1 – Software Installation

There are four separate downloads and installs:
1. SQL Server 2022
    a. https://www.microsoft.com/en-ca/sql-server/sql-server-downloads
    b. Choose the free Developer Edition. Do not choose the Basic installation; opt for custom. Install all the software, assuming that disk space is not a problem. For authentication, choose "Mixed Mode", not the default "Windows Authentication". See the screen capture in Figure 1. Assign a strong password (12 characters minimum) to the sa (system administrator) account. Windows authentication is more secure than using the "sa" login, but also more complicated, with more things to potentially go wrong. The database will be used only as a "local" installation, and all the data within it are freely available, so there is no real security issue.

1

     c. Before proceeding with the second step, you should restart your computer.
2. SQL Server Management Studio (SSMS)
     a. https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16
     b. After installing SSMS, check that you can login to the database that was created in the first step. The post at https://sqlblog.org/2020/07/28/troubleshooting-error-18456 has useful info if you have problems connecting to SQL Server. It may be helpful to download and run the SQLCHECK program in order to get diagnostic details on your configuration: https://learn.microsoft.com/en-us/troubleshoot/sql/database-engine/connect/resolve-connectivity-errors-checklist
3. SqlPackage
     a. https://learn.microsoft.com/en-us/sql/tools/sqlpackage/sqlpackage?view=sql-server-ver16
     b. Installation should be straightforward. The executable is installed by default in C:\Program Files\Microsoft SQL Server\160\DAC\bin. SQLPackage.exe is the utility software that is used to export and import a SQL Server database.
4. ODBC Driver
     a. https://learn.microsoft.com/en-us/sql/connect/odbc/microsoft-odbc-driver-for-sql-server?view=sql-server-ver16
     b. The ODBC driver provides a convenient way to access the SQL Server database from other software packages. After installing the driver, you should run the ODBC Data Source Administrator (64 bit), which is a standard component of the Windows operating system, and create a System DSN (Data Source Name) to connect to the SQL Server database. Figure 2 shows the initial screen after launching the ODBC Data Source Administrator and clicking on the "System DSN" tab. The procedure for creating a System DSN involves a sequence of dialog boxes:
         i. Create New Data Source
            1. Select ODBC Driver 18 for SQL Server
        ii. Create a New Data Source to SQL Server (see the screen capture in Figure 3)
            1. Provide a name and description for the DSN
            2. Identify the SQL Server as "(local)"
        iii. How should SQL Server verify the authenticity of the login ID?
            1. Select "With SQL Server authentication using a login ID and password entered by the user"
            2. You will be prompted to provide a login ID and password. The login ID should be "sa" and the password should correspond to the one you created earlier(see the screen capture in Figure 4).
        iv. The next dialog box offers options for configuring the DSN. Accept all the defaults (see the screen capture in Figure 5).
         v. The next dialog box offers more options for configuring the DSN. Accept the defaults except for "Connection Encryption". The default is "Mandatory". Change it to "Optional" (see the screen capture in Figure 6). Click "Finish".
        vi. The next dialog box lists all the options that you selected. There is a button to "Test Data Source…". See the screen capture in Figure 7. You should choose to run the test, to verify that the DSN is configured correctly.
        vii. The DSN has now been created, and it should appear in the list of System DSNs. See the screen capture in Figure 8.
     c. See instructions below on how to work with the SQL Server database via ODBC using MS Access.

Step 2 – Populate the database

This step involves running SqlPackage.exe, to load the contents of the bacpac file into the SQL Server database. In order to run SqlPackage.exe, open a command prompt window and change directory to C:\Program Files\Microsoft SQL Server\160\DAC\bin. A directory listing should include SqlPackage.exe. Execute SqlPackage.exe from the command prompt, supplying three command-line parameters:

1. /Action:Import
2. /SourceFile:"path to bacpac file"
3. /TargetConnectionString:"SQL Server connection string "

An example command would be:

SqlPackage /Action:Import /SourceFile:"C:\tmp\bacpac\cdogs.bacpac"
/TargetConnectionString:"Server=(local);Initial Catalog=cdogs; Encrypt=no; User ID=sa;
password=GSCgeochem2024"

The Action and SourceFile parameters are hopefully self-explanatory. The TargetConnectionString is more complicated. The example above contains five parameters:

1. Server=(local)
   a. This specifies that the database will reside on the local computer, and not on a networked computer.
2. Initial Catalog=cdogs
   a. "cdogs" is the name that will be used to identify the database. You are free to use other names of your choosing.
3. Encrypt=no
   a. This avoids potential complications due to encryption.
4. User ID=sa
   a. "sa" is short for "system administrator". The simplest possible scenario for running SQL Server is to have just one user account, which will be the "sa" account.
5. Password=GSCgeochem2024
   a. A password must be supplied for the "sa" account. It must correspond to the password that was created during the installation of SQL Server 2022.

Step 3 – Modify the ODBC DSN

The ODBC DSN that was created in Step 1 should be modified to point to the "Initial Catalog" that was specified in the SqlPackage import command ("cdogs" in the example above):

1. Open the ODBC Data Source Administrator program.
2. Highlight the System DSN that you created in Step 1 above.
3. Click on "Configure". You will be presented with a sequence of dialog boxes identical to the ones you saw when you initially created the DSN.
4. Be sure to provide the password for the "sa" account (Figure 4).
5. Keep all of the settings from the initial configuration EXCEPT for the prompt to "Change the default database to:". See Figure 5. Tick the check box, and change the default to the "Initial Catalog" specified during the SqlPackage import procedure.
6. Be sure to test the modified data source (Figure 7).

Step 4 – Use the data

After successful completion of Step 2, you will have a full working copy of the CDoGS database. You can now use SQL Server Management Studio to access the database. You can also use other software to access the database. One convenient alternative is to use Microsoft Access, in conjunction with the ODBC driver for SQL Server. The following instructions assume that you have a basic

familiarity with MS Access, and are running MS Office 365. Instructions for other versions of MS Office would be slightly different.

MS Access example

1. Run MS Access and create a new blank database.
2. Select the "External Data" tab, and then select the "New Data Source" button.
3. Select "From Other Sources" and then select "ODBC Database".
4. You will be given a choice of either importing data from the ODBC database or simply creating a link to it. MS Access has an upper file size limit of 2 Gb. This prevents the entire CDoGS database from being loaded into it. You can opt to either:
   a. Import only some of the tables, to stay below the 2 Gb limit
   b. Link to all the tables
5. For this example, choose the "link" option.
6. You will then be presented with a "Select Data Source" window, and two tabs for "File Data Source" and "Machine Data Source". Select the "Machine Data Source" tab. One of the items in the list should correspond to the DSN that you created when you installed the ODBC driver. Select it.
7. Next, you will be presented with a screen asking you to provide a login ID and password. Enter "sa" for the login ID and enter the password that you created earlier. Then click on the "Options…" button. An "Options" pane will appear. For the "Database", select the name that corresponds to the "Initial Catalog" that you created when running SQLPackage. Verify that "Connection Encryption" is set to "Optional". Then click the "OK" button.
8. The next screen will list all the tables that you can link to. Do not select all of them, because a large number are database administration tables, rather than CDoGS data tables. The CDoGS tables all have the prefix "dbo." Ignore any tables that have a prefix of "INFORMATION_SCHEMA" or "sys".
9. As an initial test, you may wish to link just one small table. "dbo.Countries_SHARED" is a good test case. It will show up in the MS Access list of tables with a special icon, and be named "dbo_Countries_SHARED". Right-clicking on the name will give you the option to "Convert to Local Table".
10. Note that any changes that you make on a "Linked Table" will change the table within the SQL Server database. Whereas changes that are made to a "Local Table" will not change the table in the SQL Server database.

Acknowledgement

References

Adcock, S.W. and Spirito, W.A., 2024. The Canadian Database of Geochemical Surveys: historical overview and current challenges; Geological Survey of Canada, Current Research 2024-1, 20 p. https://doi.org/10.4095/332490

Figure 1 Specifying "Mixed Mode" during the SQL Server installation
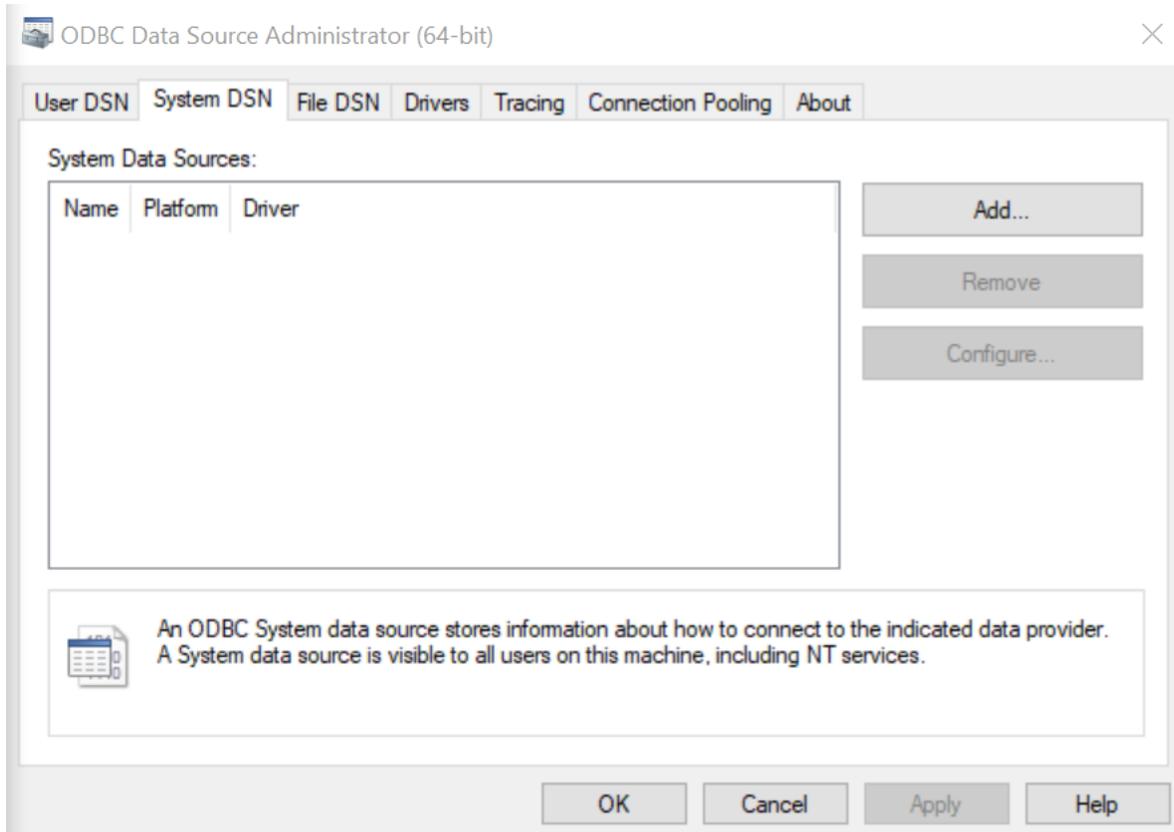
Figure 2 Initial ODBC window, with no DSNs configured

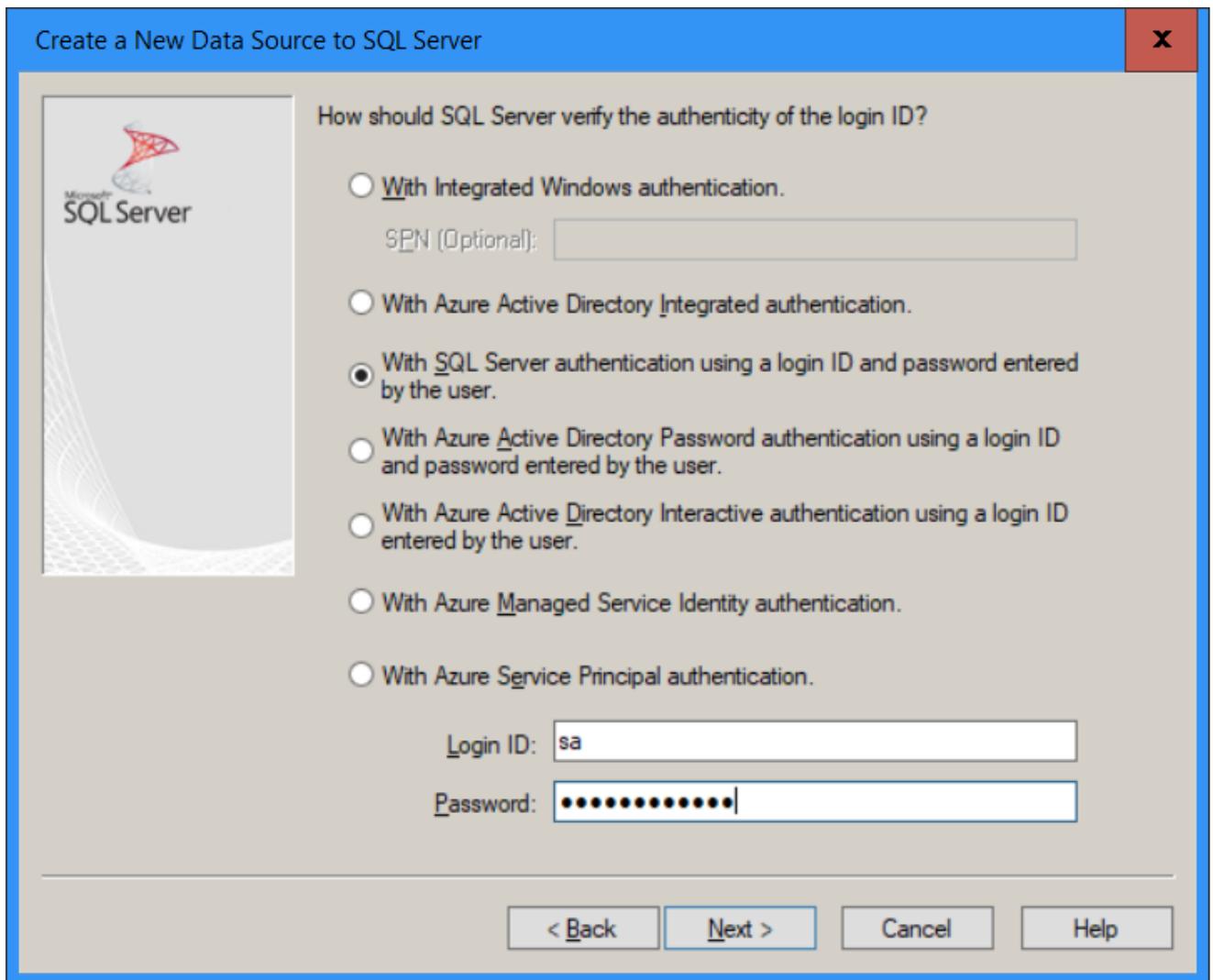*Figure 3 Specifying the ODBC data source*

Figure 4 Specifying ODBC SQL Server authentication

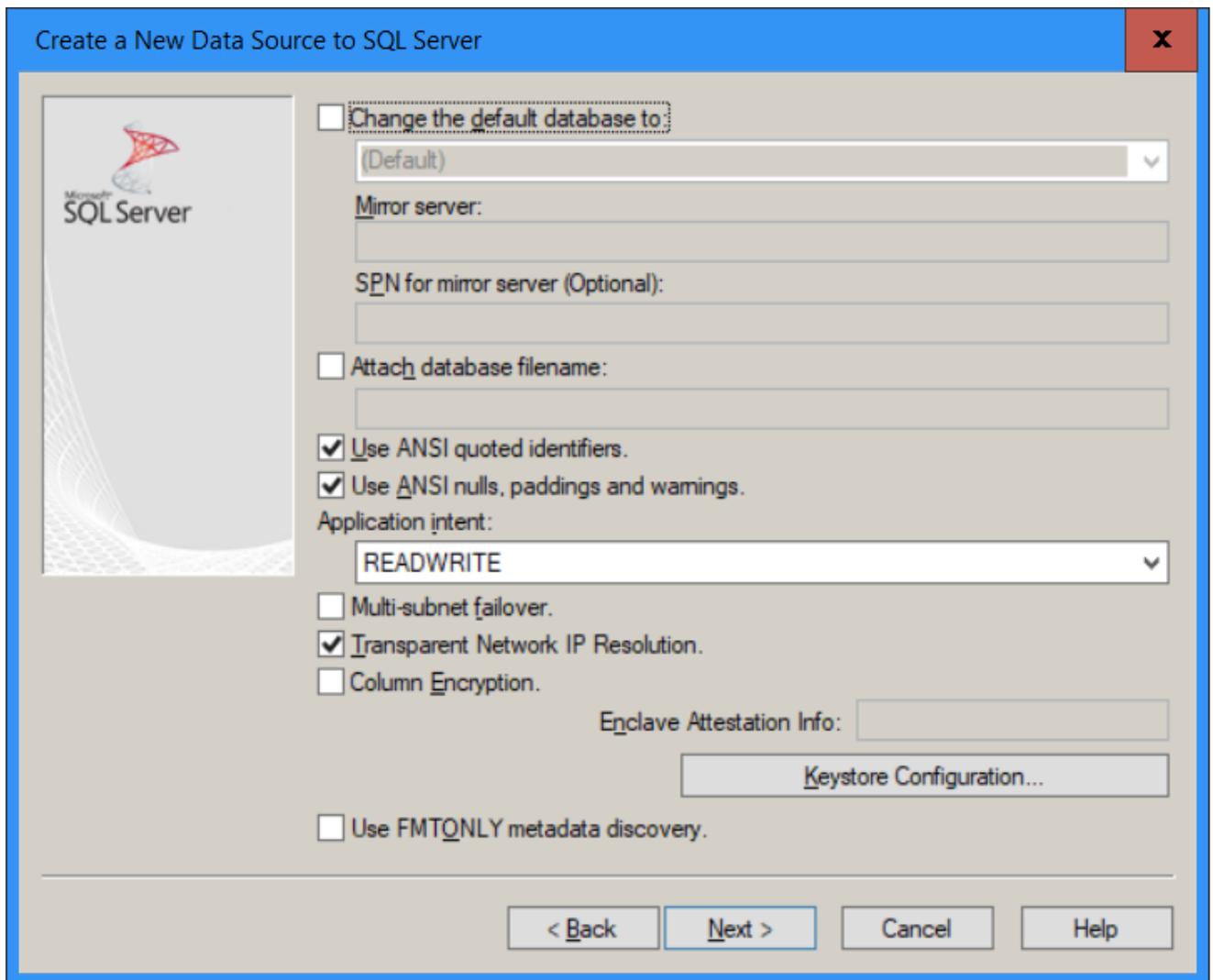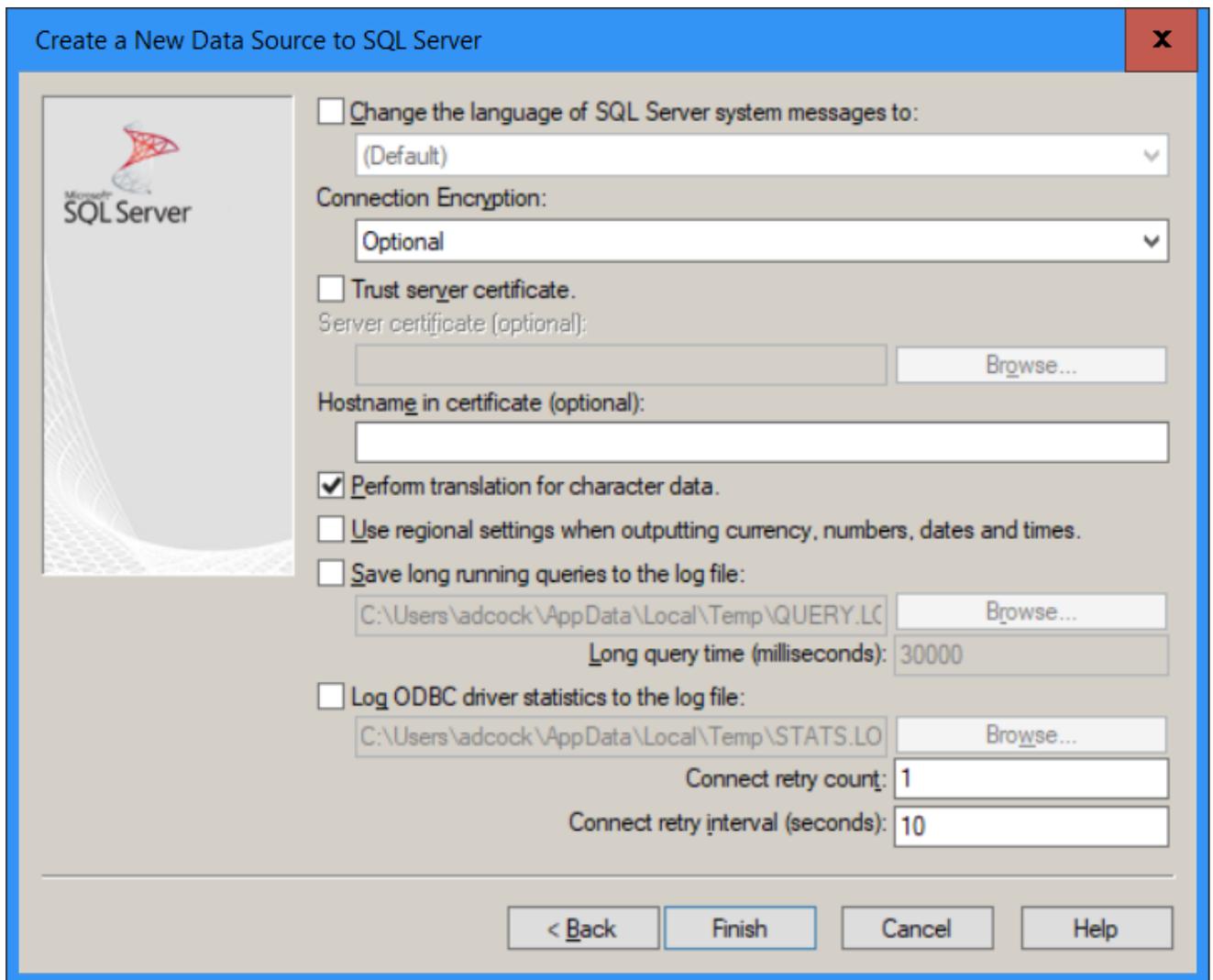*Figure 5 ODBC driver configuration (continued)*

Figure 6 ODBC configuration options (final)

ODBC Microsoft SQL Server Setup   [X]

A new ODBC data source will be created with the following configuration:

Microsoft ODBC Driver for SQL Server Version 18.03.0003

Data Source Name: geochem
Data Source Description: CDoGS
Server: (local)
Use Integrated Security: No
Database: (Default)
Language: (Default)
Data Encryption: Optional
Trust Server Certificate: No
Multiple Active Result Sets(MARS): No
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Regional Settings: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes

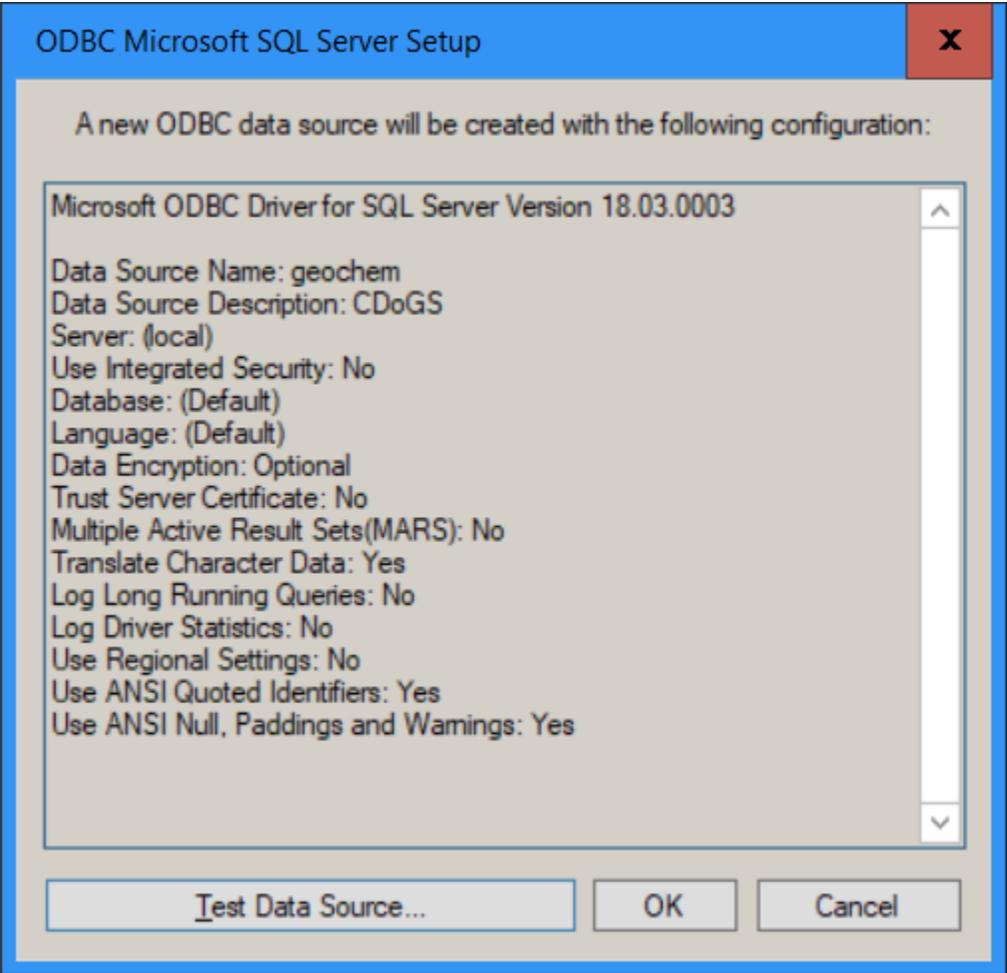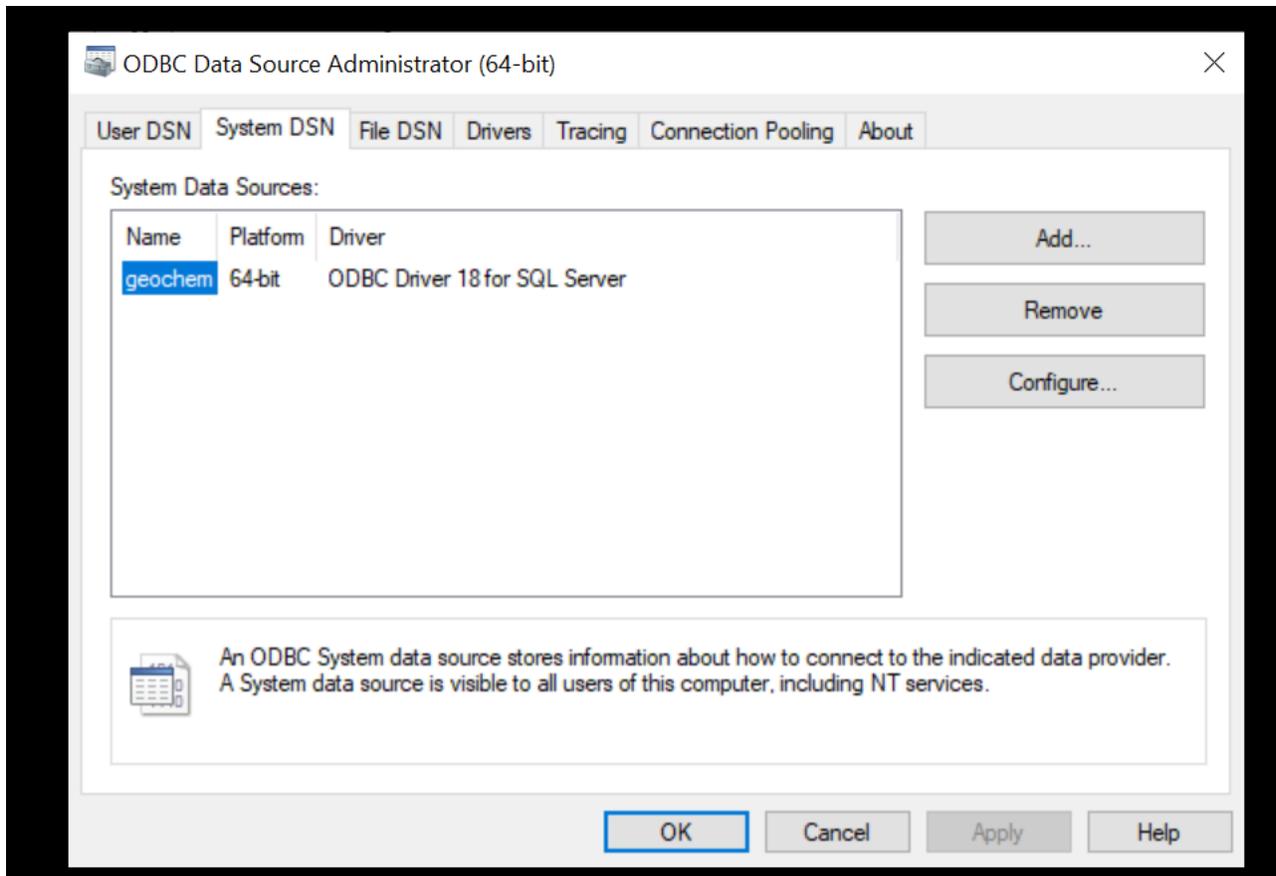[Test Data Source...]   [OK]   [Cancel]

Figure 7 ODBC configuration summary

Figure 8 ODBC window showing the newly created DSN