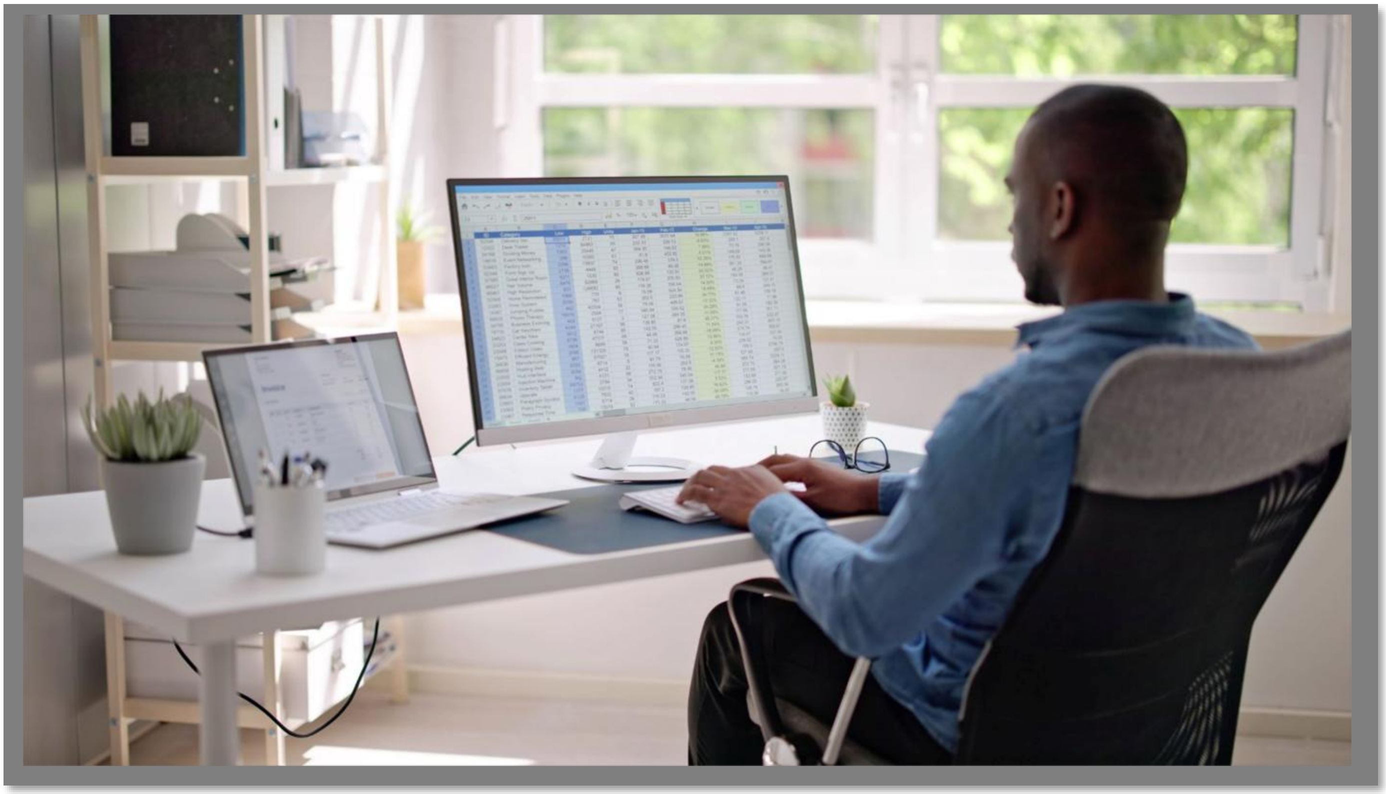
Arbutus Connectors

Snowflake

CONFIGURATION GUIDE



ARBUTUS

Powerful Analytics Simplified

Contents

Arbutus Connector – Snowflake

A. Introduction

The purpose of this Guide is to provide assistance with configuring the Arbutus Snowflake Connector using the ODBC Data Source Administrator. The configuration process can involve several technical steps that require a good understanding of IT systems and database management.

To make the most of this guide, it's advisable to have a good understanding of database connectivity, driver installation, and system settings. The ODBC Data Source Administrator, which is used as part of the configuration process, allows for the setup and management of data sources, enabling applications to access data from various database systems.

Due to the complexity and potential impact of these configurations, it is recommended that only those individuals with IT or database expertise undertake this task. In addition, it should also be understood that each client's network environment is different. A one-size-fits-all approach is rarely effective, as what works well in one environment may not be suitable in another.

B. About Snowflake

Snowflake is a cloud-based data platform that enables organizations to store, manage, and analyze large volumes of data, allowing for scalable and efficient data processing. Snowflake supports diverse workloads, including data warehousing, data lakes, and data sharing, and provides robust security and governance features.

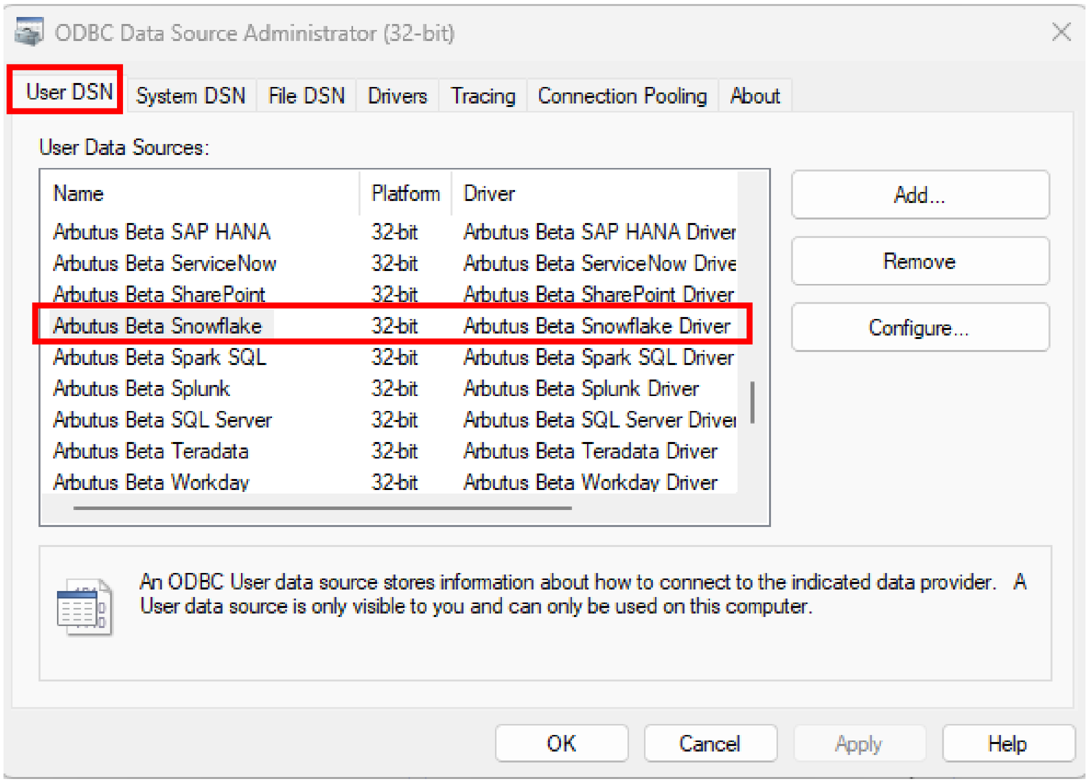
In Snowflake, data is stored in a cloud-based architecture. When data is loaded into Snowflake, it is reorganized into an optimized, compressed, columnar format and stored in cloud storage. Snowflake manages all aspects of data storage, including organization, file size, structure, compression, and metadata. This data is not directly visible or accessible by customers; it can only be accessed through SQL query operations within Snowflake.

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus Snowflake Connector is done at the time of installing the Arbutus software. For more information on this, please see the **Overview Guide Document**.

Once the Connector has been installed, the next step is to configure it.

Prior to configuring it, you can check to see if the Connector has been installed by opening the **32-bit ODBC Data Source Administrator**, pictured below, and clicking the **User DSN** tab. Included below is information on how you can access the **ODBC Data Source Administrator**.



Name	Platform	Driver
Arbutus Beta SAP HANA	32-bit	Arbutus Beta SAP HANA Driver
Arbutus Beta ServiceNow	32-bit	Arbutus Beta ServiceNow Drive
Arbutus Beta SharePoint	32-bit	Arbutus Beta SharePoint Driver
Arbutus Beta Snowflake	32-bit	Arbutus Beta Snowflake Driver
Arbutus Beta Spark SQL	32-bit	Arbutus Beta Spark SQL Driver
Arbutus Beta Splunk	32-bit	Arbutus Beta Splunk Driver
Arbutus Beta SQL Server	32-bit	Arbutus Beta SQL Server Driver
Arbutus Beta Teradata	32-bit	Arbutus Beta Teradata Driver
Arbutus Beta Workday	32-bit	Arbutus Beta Workday Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.

- If the Arbutus Snowflake Connector appears in the list, it can be considered as installed.
- If it is not listed, it is likely that you did not select it during the installation or modification of the Arbutus software. In this case, it is recommended to reinstall the Arbutus software and choose the **Modify** option when prompted. For more details, please refer to the **Overview Guide Document**.

Below is the file path to access and run the **ODBC Data Source Administrator** application:

```
C:\Windows\SysWOW64\odbcad32.exe
```

Alternative, you can also try locating and opening the **ODBC Data Source Administrator** application by doing a search on your desktop application.

D. Configuring the Connector after it has been installed

Once you have verified that the Arbutus Connector has been installed, it is time to configure it.

This process is done using the **ODBC Data Source Administrator**. It can be described as “**editing the DSN configuration**”.

DSN, Drivers, and Data Sources

What is a DSN? DSN stands for Data Source Name, and is a unique name used to create a data connection to a database using open database connectivity (ODBC).

A DSN is a data structure that contains the information required to connect to a database. It is essentially a string that identifies the source database, including the driver details, the database name, and often authentication credentials and other necessary connection parameters. DSNs facilitate a standardized method for applications to access databases without needing hard-coded connection details, enhancing flexibility and scalability in database management.

- ***Drivers*** are the components that process ODBC requests and return data to the application. If necessary, drivers modify an application's request into a form that is understood by the data source. The **Drivers** tab in the **ODBC Data Source Administrator** dialog box lists all drivers installed on your computer, including the name, version, company, file name, and file creation date of each driver.

- ***Data sources*** are the databases of files accessed by a driver and are identified by a data source name (DSN). You use the ODBC Data Source Administrator to add, configure, and delete data sources from your system.

All ODBC connections require that a DSN be configured to support the connection. When a client application wants to access an ODBC-compliant database, it references the database using the DSN.

The types of DSNs are:

- **User DSN** – User DSNs are local to a computer and can be used only by the current user. They are registered in the HKEY_Current_USER registry subtree.

- **System DSN** – System DSNs are local to a computer rather than dedicated to a user. The system or any user with privileges can use a data source set up with a system DSN. System DSNs are registered in the HKEY_LOCAL_MACHINE registry subtree.

- **File DSN** – File DSNs are file-based sources that can be shared among all users who have the same drivers installed and therefore have access to the database. These data sources need not be dedicated to a user nor be local to a computer. File data source names are identified by a file name with a .dsn extension.

User and system data sources are collectively known as *machine* data sources because they are local to a computer.

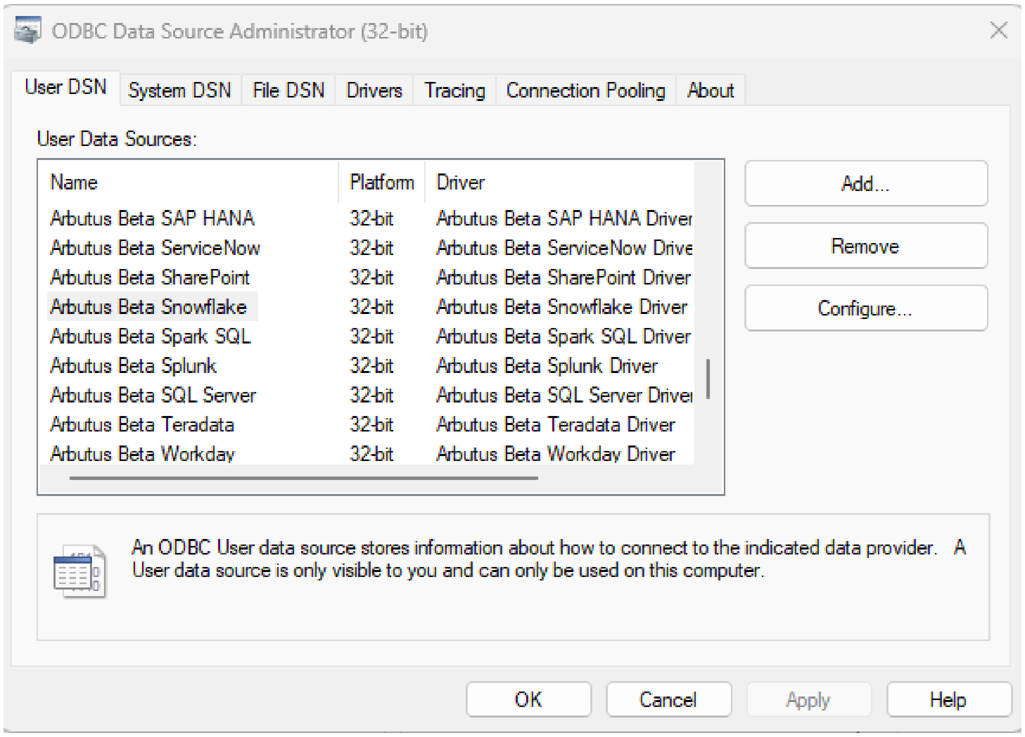
Each of these DSNs has a tab in the **ODBC Data Source Administrator** dialog.

The Arbutus ODBC Driver for Snowflake enables real-time access to Snowflake data, directly from any applications that support ODBC connectivity, the most widely supported interface for connecting applications with data.

Arbutus Connectors

Follow these steps to edit the DSN configuration and configure the Connector.

1. First open the **ODBC Data Source Administrator**.



User Data Sources:

Name	Platform	Driver
Arbutus Beta SAP HANA	32-bit	Arbutus Beta SAP HANA Driver
Arbutus Beta ServiceNow	32-bit	Arbutus Beta ServiceNow Drive
Arbutus Beta SharePoint	32-bit	Arbutus Beta SharePoint Driver
Arbutus Beta Snowflake	32-bit	Arbutus Beta Snowflake Driver
Arbutus Beta Spark SQL	32-bit	Arbutus Beta Spark SQL Driver
Arbutus Beta Splunk	32-bit	Arbutus Beta Splunk Driver
Arbutus Beta SQL Server	32-bit	Arbutus Beta SQL Server Driver
Arbutus Beta Teradata	32-bit	Arbutus Beta Teradata Driver
Arbutus Beta Workday	32-bit	Arbutus Beta Workday Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.

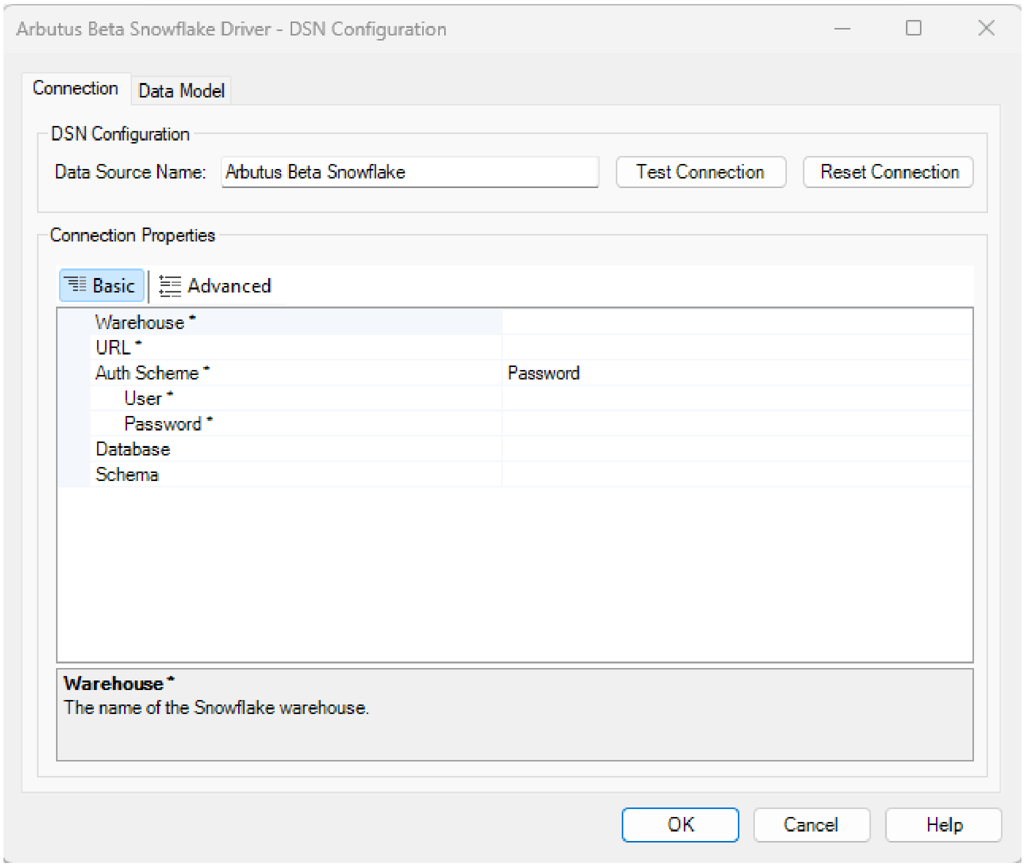
2. Click the **User DSN** tab.

   Selected data connectors are installed as **User DSN's** in Window's 32 Bit **ODBC Data Source Administrator**.

   Also, each of the data connector's names is prefaced with Arbutus, for example, **Arbutus Snowflake.**

3. Select the Arbutus Connector, in this case it is **Arbutus Snowflake**.
4. Click **Configure**.

This opens the **Arbutus Snowflake Driver – DSN Configuration** dialog.



Data Source Name: Arbutus Beta Snowflake

Warehouse *	
URL *	
Auth Scheme *	Password
User *	
Password *	
Database	
Schema	

Warehouse *
The name of the Snowflake warehouse.

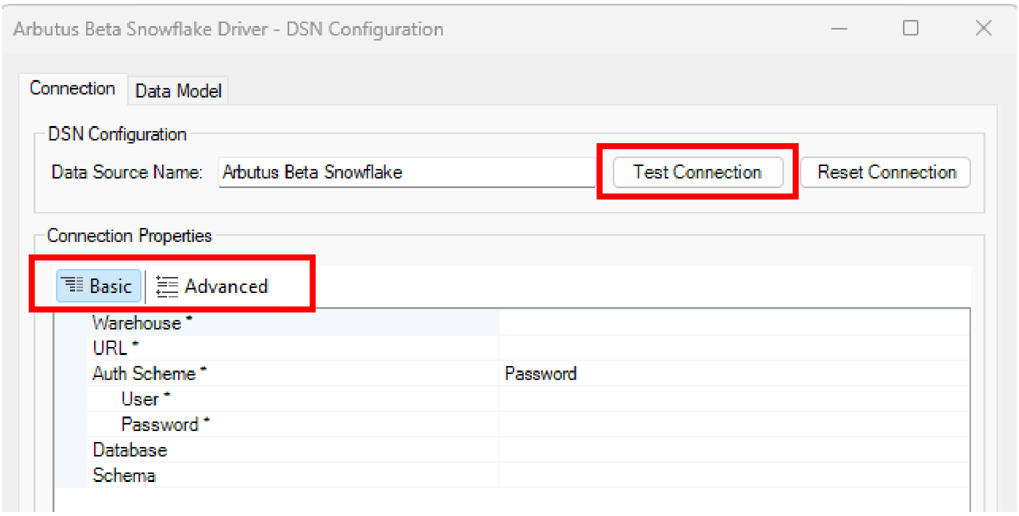
E. Editing the DSN properties – the Basic and Advanced tabs

With the DSN Configuration dialog open, the next step is to edit the DSN properties, where necessary, in the **Basic** and **Advanced** tabs. For example, editing the **Auth Scheme properties** (per screenshot below) to ensure correct authentication to the server is applied.

E1. Editing the DSN properties in the Basic tab

The properties listed in the **Basic** tab are typically the ones that are most commonly used, and as such are designed to be more user-friendly and straightforward, allowing you to quickly make changes without needing in-depth technical knowledge.

Once you have completed editing the properties in the **Basic** tab, you can go ahead and try testing the connection to the Snowflake system by clicking the **Test Connection** button, as highlighted in the screenshot below.



Data Source Name: Arbutus Beta Snowflake

Warehouse *	
URL *	
Auth Scheme *	Password
User *	
Password *	
Database	
Schema	

In the **Basic** tab, there are **five** main properties to review:

1. **Warehouse** – this is the name of the Snowflake warehouse.

2. **URL** – this is the name of the Snowflake database. For example, https://orgname-myaccount.snowflakecomputing.com.

   To find your URL:

   1. Click on your name in the lower left-hand corner of your Snowflake UI
   2. Hover over your **Account ID**
   3. Click the **Copy Account URL** icon to copy your account URL

3. **Auth Scheme** – click the dropdown to select from the list the appropriate scheme used for authentication. The options available for selection are as follows:
   - Password – select this when you want to authenticate using a **username and password** combination. This is a straightforward and common method for connecting to Snowflake, especially when you have user credentials set up with the necessary permissions to access the Snowflake data warehouse.

     This method is suitable for many scenarios, including local development and production environments where you manage user credentials directly.

     Selecting **Password** requires you to specify the User and Password.
     - User – this is the username provided for authentication with the Snowflake database.

     - Password – this is the (user's) password provided for authentication with Snowflake.

   - OKTA – select this when you want to use Okta for authentication. This is particularly useful if your organization uses Okta for single sign-on (SSO) and identity management. By using Okta, you can leverage its security features, such as multi-factor authentication (MFA), to enhance the security of your Snowflake connections.

     This method allows users to authenticate with their Okta credentials, providing a seamless and secure login experience.

Selecting **OKTA** requires you to specify the following:

- User – this is the username provided for authentication with the Snowflake database.

- Password – this is the (user's) password provided for authentication with Snowflake.

- SSO Properties – this is additional properties required to connect to the identity provider in a semicolon-separated list.

- MFS Passcode – this specifies the passcode to use for multi-factor authentication.

○ Private Key – select this when you want to use **key pair authentication** for connecting to Snowflake. This method is particularly useful for enhancing security, as it uses a private key and a corresponding public key to authenticate the connection.

Key pair authentication is beneficial in scenarios where you want to avoid using passwords and leverage stronger security measures. It is often used in automated processes and environments where security is a high priority.

Selecting **Private Key** requires you to specify the following:

- User – this is the username provided for authentication with the Snowflake database.

- Private Key – this is the private key provided for key pair authentication with Snowflake.

  The path to the file containing the private key or the name of the certificate store for the client certificate. The **Private Key Type** field specifies the type of the certificate store specified by **Private Key**. If the store is password protected, specify the password in **Private Key Password**.

When the certificate store type is PEMKEY_FILE, PFXFILE, etc., this property must be set to the path to the file. When the type is PEMKEY_BLOB, PFXBLOB, etc., the property must be set to the binary contents of the file.

Designations of certificate stores are platform-dependent.

he following are designations of the most common User and Machine certificate stores in Windows:

MY	A certificate store holding personal certificates with their associated private keys
CA	Certifying authority certificates
ROOT	Root certificates
SPC	Software publisher certificates

In Java, the certificate store normally is a file containing certificates and optional private keys.

- Private Key Password – this is the password for the private key specified in the **Private Key** property, if required.

- Private Key Type – this is the type of key store containing the private key to use with key pair authentication. This is a dropdown selection consisting of the following possible key type stores for you to choose from:

  USER, MACHINE, PFXFILE, PFXBLOB, JKSFILE, JKSBLOB, PEMKEY_FILE, PEMKEY_BLOB, PUBLIC_KEY_FILE, PUBLIC_KEY_BLOB, SSHPUBLIC_KEY_FILE, SSHPUBLIC_KEY_BLOB, P7BFILE, PPKFILE, XMLFILE, XMLBLOB

  The default value is **USER**.

  If required, more information on the key type stores listed above can be provided.

- AzureAD – select this when you want to use **Azure Active Directory (AAD)** for authentication. This is particularly useful if your organization uses Azure AD for single sign-on (SSO) and identity management. By using Azure AD, you can leverage its security features, such as multi-factor authentication (MFA), to enhance the security of your Snowflake connections.

  This method allows users to authenticate with their Azure AD credentials, providing a seamless and secure login experience.

  Selecting **AzureAD** requires you to specify the following:

  - User – this is the username provided for authentication with the Snowflake database.

  - Azure Tenant – this is the Microsoft Online (Snowflake) tenant being used to access data. If not specified, your default tenant is used.

    A tenant is a digital representation of your organization, primarily associated with a domain. For example, microsoft.com. The tenant is managed through a Tenant ID (also known as the directory ID), which is specified whenever you assign users permissions to access or manage Azure resources.

    To locate the directory ID in the Azure Portal, navigate to **Azure Active Directory > Properties**.

    Specifying **Azure Tenant** is required when **Auth Scheme** = either **Azure Service Principal** or **Azure Service Principal Cert**, or if **Auth Scheme** = **Azure AD** and the user belongs to more than one tenant.

- OAuth Client ID – this is the client Id assigned when you register your application with an OAuth authorization server.

  OAuth Client Id is one of a handful of connection parameters that need to be set before users can authenticate via OAuth.

- OAuth Client Secret – this is the client secret assigned when you register your application with an OAuth authorization server. Also known as the consumer secret.

  OAuth Client Secret is one of a handful of connection parameters that need to be set before users can authenticate via OAuth.

- Proof Key – this is the **Proof Key** for authentication with Snowflake database. This is usually derived from GetSSOAuthorizationURL call.

- External Token – this is the **External Token** for authentication with the Snowflake database. This is usually derived from the external handler. For example, handle the callback URL from procedure GetSSOAuthorizationURL will get this token.

○ AzureMSI – select this when you want to use **Azure Managed Service Identity (MSI)** for authentication. This is particularly useful if your Snowflake instance is integrated with **Azure** and you want to leverage the security and management features provided by Azure MSI.

Using Azure MSI allows your application to authenticate to Snowflake without needing to manage credentials directly, enhancing security and simplifying management

Selecting **AzureMSI** requires you to specify the following:

- Azure Resource – this is the Azure Active resource to authenticate to - used during Azure Managed Service Identity exchange. It should be set to the App Id URL.

  The resource must be specified if using Azure Managed Service Identity.

- OAuth – select this when you want to use **OAuth 2.0** for authentication. This method is particularly useful if you want to leverage token-based authentication, which provides enhanced security and allows for seamless integration with various identity providers.

  OAuth 2.0 is commonly used for single sign-on (SSO) and can help manage access tokens securely.

  This approach is beneficial in scenarios where you need to authenticate users through a third-party identity provider or when you want to avoid managing passwords directly.

  Selecting **AzureAD** requires you to specify the following:

  - User – this is the username provided for authentication with the Snowflake database.

  - OAuth Client ID – this is the client Id assigned when you register your application with an OAuth authorization server.

    OAuth Client Id is one of a handful of connection parameters that need to be set before users can authenticate via OAuth.

  - OAuth Client Secret – this is the client secret assigned when you register your application with an OAuth authorization server. Also known as the consumer secret.

    OAuth Client Secret is one of a handful of connection parameters that need to be set before users can authenticate via OAuth.

  - OAuth Authenticator – this determines the authenticator that the OAuth application requests from Snowflake.

    This is a dropdown selection consisting of three possible options for you to choose from:

    a. None

    b. Azure

    c. OKTA

- Ping Federate – select this when you want to use Ping Identity PingFederate for authentication. This is particularly useful if your organization uses PingFederate for single sign-on (SSO) and identity management. By using PingFederate, you can leverage its security features to enhance the security of your Snowflake connections.

  This method allows users to authenticate with their PingFederate credentials, providing a seamless and secure login experience.

  Note:

  PingFederate is a highly regarded enterprise federation server that specializes in user authentication and providing standardized single sign-on (SSO) solutions.

Selecting **Ping Federate** requires you to specify the following:

- User – this is the username provided for authentication with the Snowflake database.

- Password – this is the (user's) password provided for Authentication with Snowflake.

- Proof Key – this is the **Proof Key** for authentication with Snowflake database. This is usually derived from GetSSOAuthorizationURL call.

- External Token – this is the **External Token** for authentication with the Snowflake database. This is usually derived from the external handler. For example, handle the callback URL from procedure GetSSOAuthorizationURL will get this token.

○ External Browser – select this when you want to use **browser-based single sign-on (SSO)** for authentication. This method is particularly useful if your organization uses an identity provider (IdP) that supports SSO, such as Okta, Azure AD, or Ping Identity.

Using the External Browser option allows users to authenticate through a web browser, providing a seamless and secure login experience. This approach leverages the security features of your IdP, such as multi-factor authentication (MFA), to enhance the security of your Snowflake connections.

Selecting **External Browser** requires you to specify the following:

- User – this is the username provided for authentication with the Snowflake database.

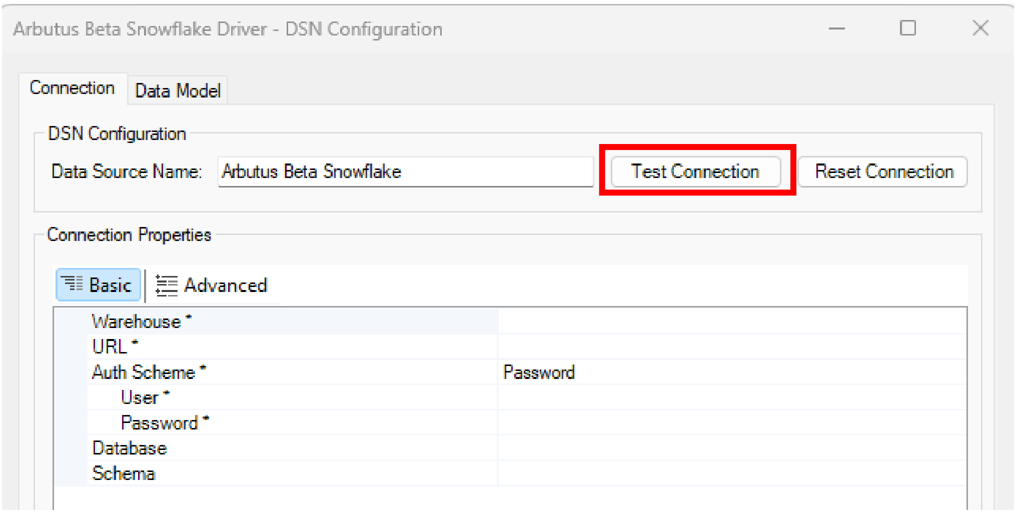
4. **Database** – this is the name of the Snowflake database.

5. **Schema** – this is the schema of the Snowflake database.

E2. Editing the DSN properties in the Advanced tab

This tab includes more detailed and technical properties. It is intended for those users who need more control over the configuration and are comfortable with more complex options. The **Advanced** tab often includes properties that can fine-tune the behaviour of the system feature.

If you have already completed editing the properties in the **Basic** tab, as required, you do not necessarily need to also complete editing the properties in the **Advanced** tab. Instead, once you have completed editing the properties in the **Basic** tab, you may opt to proceed to testing the connection to the Snowflake system by clicking the **Test Connection** button.

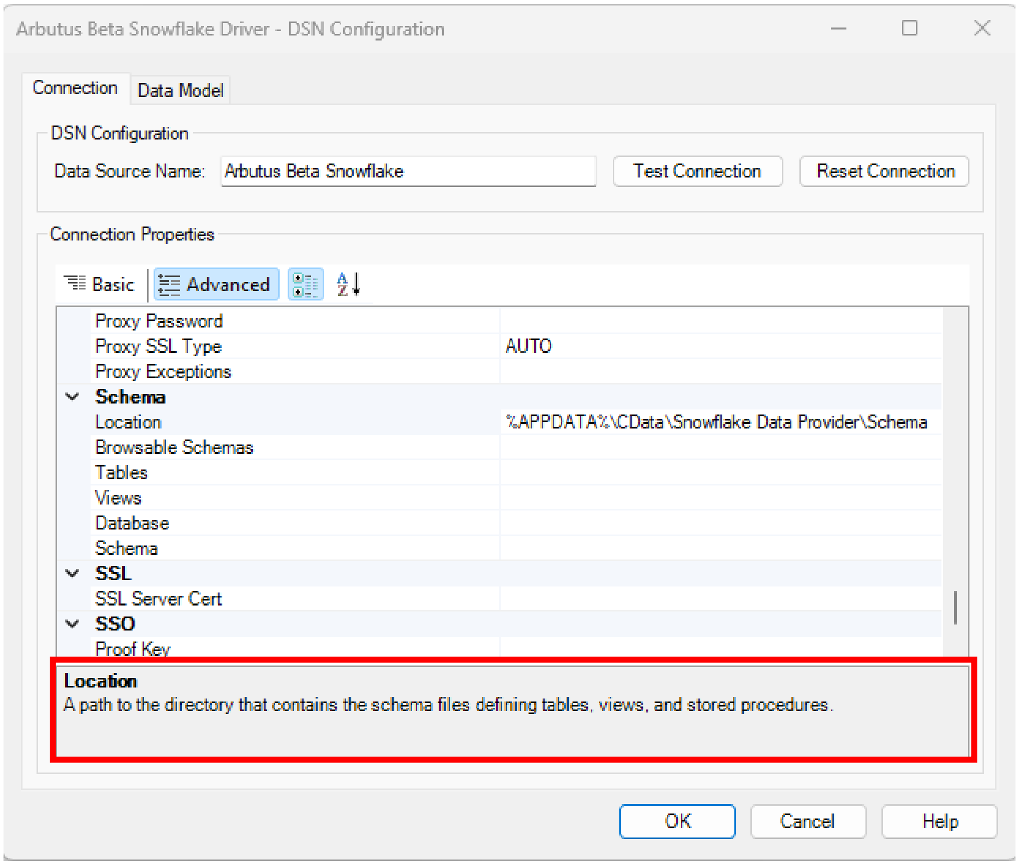


Data Source Name: Arbutus Beta Snowflake

Warehouse *	
URL *	
Auth Scheme *	Password
User *	
Password *	
Database	
Schema	

There are a lot more properties included for editing in the **Advanced** tab.

However, it is useful to know that each property does provide a short description of it and as such serves as a guide in terms of what to edit and/or enter. This short description can be seen at the bottom of the **DSN Configuration** dialog box, as seen in the screenshot below.



Arbutus Beta Snowflake Driver - DSN Configuration

DSN Configuration

Data Source Name: Arbutus Beta Snowflake

Connection Properties

Proxy Password	
Proxy SSL Type	AUTO
Proxy Exceptions	
Schema	
Location	%APPDATA%\CData\Snowflake Data Provider\Schema
Browsable Schemas	
Tables	
Views	
Database	
Schema	
SSL	
SSL Server Cert	
SSO	
Proof Key	

Location
A path to the directory that contains the schema files defining tables, views, and stored procedures.

If it is deemed necessary to complete some/all the properties in the **Advanced** tab, it is recommended that you refer to the description shown for any of the properties being edited and/or entered.

If required, more information on the properties listed in the **Advanced** tab can also be provided.

F. Other questions and/or request for assistance

There may be times when you need to consult with the technical support team at Arbutus Software. If so, please send an email request to support@ArbutusSoftware.com.

For more information, please refer to the CONTACT US page on our website.