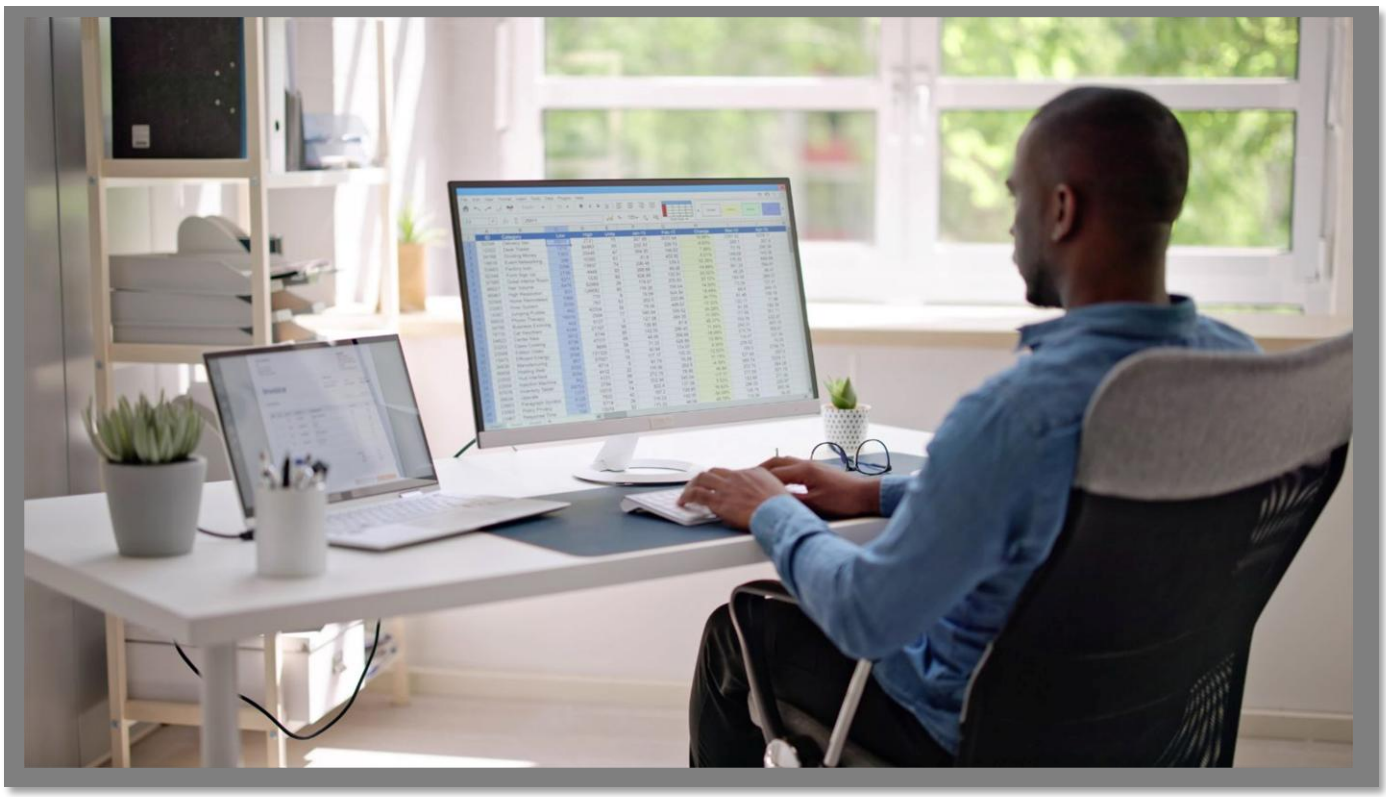
Arbutus Connectors

SQL Server
CONFIGURATION GUIDE



ARBUTUS
Powerful Analytics Simplified

Contents

Arbutus Connector – SQL Server

A. Introduction

The purpose of this Guide is to provide assistance with configuring the Arbutus SQL Server 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 SQL Server

SQL Server is a widely used relational database management system (RDBMS) developed by Microsoft. It is designed to store, retrieve, and manage data for various applications. SQL Server uses Structured Query Language (SQL) for database operations and supports a wide range of data types and functions. It is commonly used for enterprise-level applications due to its scalability, security features, and integration with other Microsoft products.

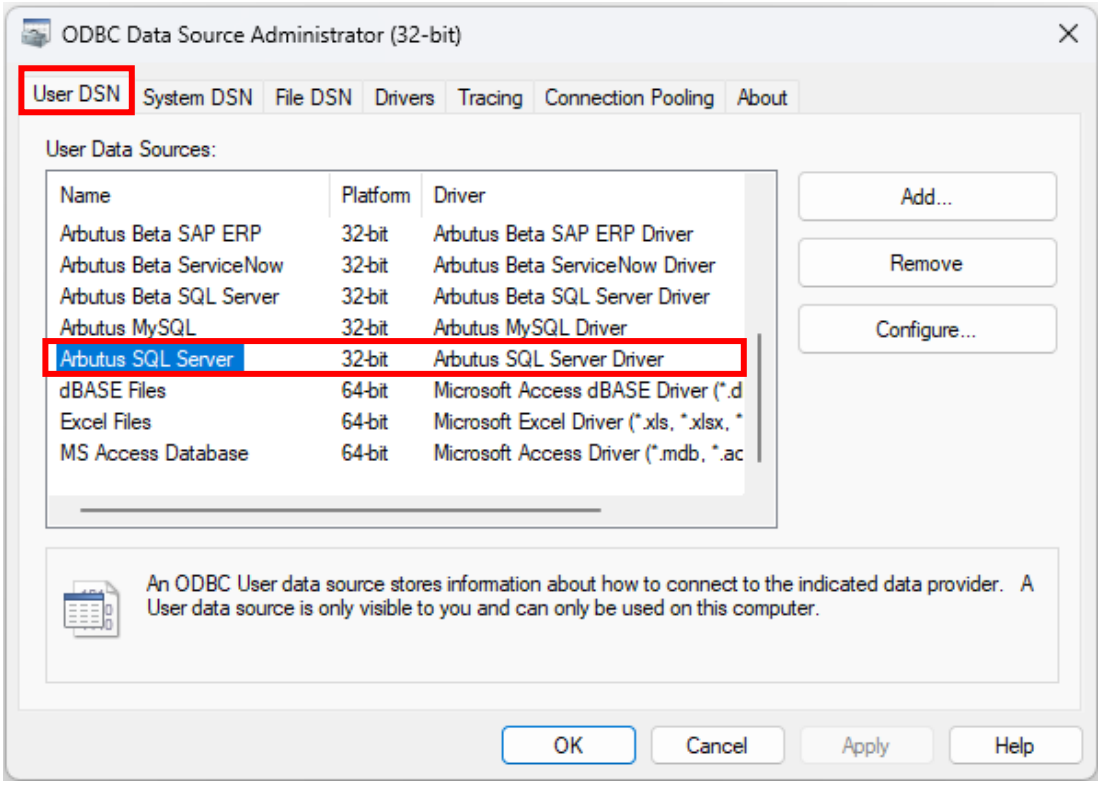
Data is stored in tables using row-based storage format. Tables are organized within databases (relational database model), and each table has columns (fields) and rows (records).

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus SQL Server 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**.



- If the Arbutus SQL Server 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.

Arbutus Connectors

- ***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.

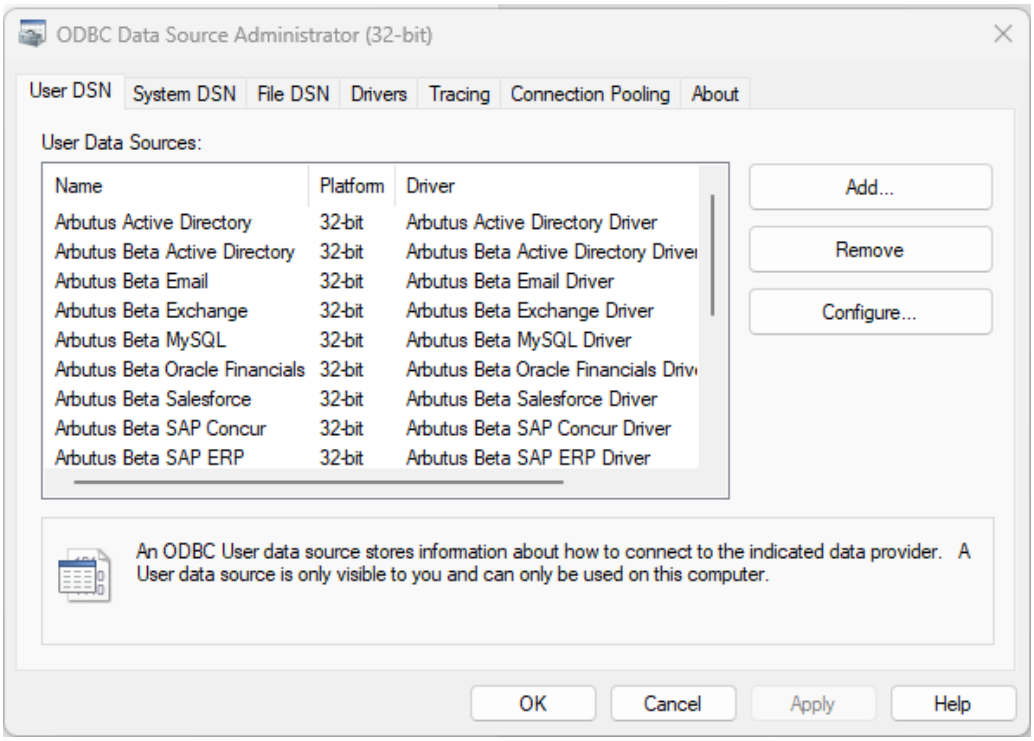
Each of these DSNs has a tab in the **ODBC Data Source Administrator** dialog.

The Arbutus ODBC Driver for SQL Server enables real-time access to SQL Server 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**.



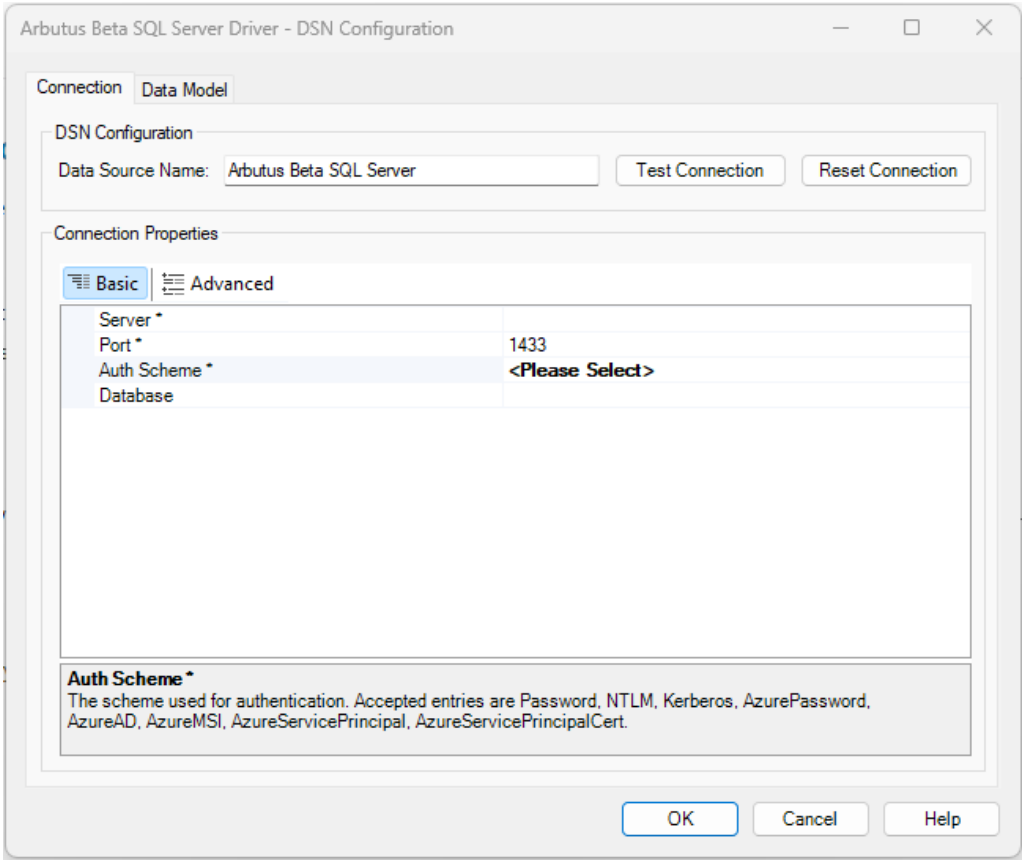
2. Click the **User DSN** tab.

   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 SQL Server.**

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

This opens the **Arbutus SQL Server Driver – DSN Configuration** dialog.



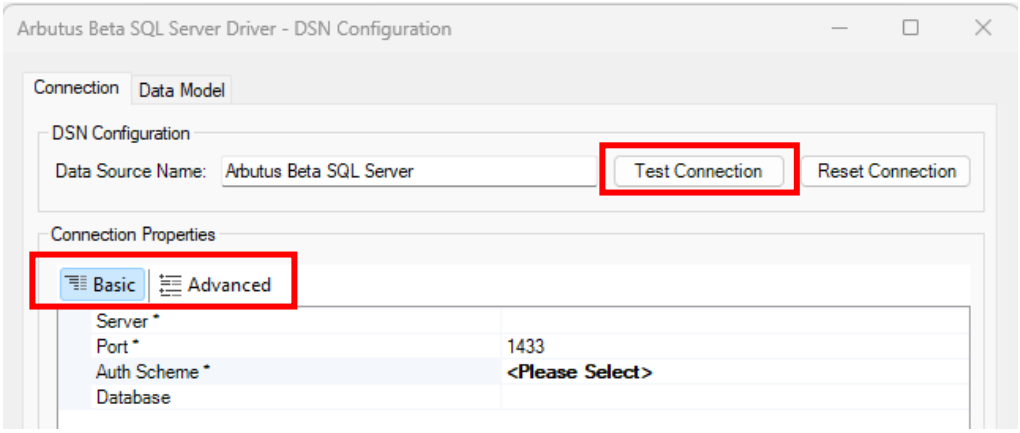
E. Editing the DSN properties – the Basic and Advanced tabs

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 **1433** entry for the **Port** (per screenshot below) to match the Port of the Client's SQL Server.

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 SQL Server by clicking the **Test Connection** button, as highlighted in the screenshot below.



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

1. **Server** – enter the name of the server running SQL Server.

2. **Port** – enter the port of the SQL Server, i.e., the port of the Server hosting the MS SQL Server Database.

   The default value is **1433**.

3. **Auth Scheme** – click the dropdown to select from the list the appropriate scheme used for authentication. The options available are:
   - Password – select this if you are using **standard SQL Server authentication**. This method requires a username and password to authenticate to the SQL Server database.

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

- User – this is he SQL Server user account used to authenticate. with the SQL Server. Together with **Password** (see below), this field is used to authenticate against the SQL Server.

- Password – this is the password associated with authenticating the user. The **User** (see above) and **Password** are together used to authenticate with the server.

The default value is **Password**.

- NTLM (NT Lan Manager) – select this if you are using **NT LAN Manager (NTLM)** for authentication. NTLM is a suite of Microsoft security protocols that provide authentication, integrity, and confidentiality to users.

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

  - User – this is the SQL username provided for authentication with SQL Server. Together with **Password** (see below), this field is used to authenticate against the SQL Server.

  - Password – this is the password associated with authenticating the user. The **User** (see above) and **Password** are together used to authenticate with the server.

  - Domain – this is the name of the domain for a Windows (NTLM) security login.

  - NTLM Version – this property specifies the NTLM version to use. The possible values are 1 or 2.

    The default value is **1**.

- Kerberos – select this if you are using **Kerberos authentication**. Kerberos is a network authentication protocol designed to provide strong authentication for client-server applications.

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

  - Kerberos KDC - enter the Kerberos Key Distribution Center (KDC) service used to authenticate the user.

    If Kerberos KDC is not specified, the driver will attempt to detect these properties automatically from the following locations:
    - KRB5 Config File (krb5.ini/krb5.conf)
    - Domain Name and Host

  - Kerberos Realm – enter the Kerberos Realm used to authenticate the user.

    If the Kerberos Realm is not specified, the driver will attempt to detect these properties automatically from the following locations:
    - KRB5 Config File (krb5.ini/krb5.conf)
    - Domain Name and Host

  - Kerberos SPN - enter the service principal name (SPN) for the Kerberos Domain Controller.

    If the SPN on the Kerberos Domain Controller is not the same as the URL that you are authenticating to, use this property to set the SPN.

  - User – enter the SQL username provided for authentication with SQL Server.

    If the user you are using for the database doesn't match the user that is in the Kerberos database, this should be set to the Kerberos principal name.

- Password – enter the password associated with authenticating the user.

- Kerberos Keytab File – enter the Keytab file containing your pairs of Kerberos principals and encrypted keys.

- Kerberos Ticket Cache – enter the full path to an MIT Kerberos credential cache file.

  This property can be set if you wish to use a credential cache file that was created using the MIT Kerberos Ticket Manager or kinit command.

  If required, more information on the Kerberos properties can be provided.

- Azure Password – select this if you are using **Azure Active Directory (Azure AD) password authentication**. This method allows you to authenticate to your SQL Server database using an Azure AD user account and password.

  Selecting **Azure Password** requires you to specify the following:
  - User – enter the SQL username provided for authentication with SQL Server. Together with **Password** (see below), this field is used to authenticate against the SQL Server.

  - Password – enter the password associated with authenticating the user. The **User** (see above) and **Password** are together used to authenticate with the server.

- Azure AD – select this if you are using **Azure Active Directory (Azure AD)** for authentication. This method allows you to authenticate to your SQL Server database using Azure AD credentials.

- Azure MSI – select this if you are using **Azure Managed Service Identity (MSI)** for authentication. This method allows your application to authenticate to the SQL Server database using the managed identity assigned to an Azure resource, such as a virtual machine or an Azure App Service.

- Azure Service Principal – select this if you are using an **Azure Active Directory (Azure AD) service principal** for authentication. This method allows your application to authenticate to the SQL Server database using a service principal, which is a type of security identity often used for automated processes and applications.

  Selecting **Azure Service Principal** requires you to specify the following:

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

    For instance, contoso.onmicrosoft.com. Alternatively, specify the tenant Id. This value is the directory Id in the Azure Portal > Azure Active Directory > Properties.

    It is a good practice to specify the Tenant, although in general things should normally work without having to specify it.

The Azure Tenant is required when setting **OAuth Grant Type** to **CLIENT**. When using client credentials, there is no user context. The credentials are taken from the context of the app itself. While Microsoft still allows client credentials to be obtained without specifying which Tenant, it has a much lower probability of picking the specific tenant you want to work with. For this reason, we require **Azure Tenant** to be explicitly stated for all client credentials connections to ensure you get credentials that are applicable for the domain you intend to connect to.

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

  As part of registering an OAuth application, you will receive the **OAuth Client Id** value, sometimes also called a consumer key, and a client secret, the **OAuth Client Secret** (see below).

- OAuth Client Secret – enter the client secret assigned when you register your application with an OAuth authorization server.

  As part of registering an OAuth application, you will receive the **OAuth Client Id** (see above), also called a consumer key. You will also receive a client secret, also called a consumer secret. Set the client secret in the **OAuth Client Secret** property.

- Azure Service Principal Cert – select this if you are using an **Azure Active Directory (Azure AD) service principal with a certificate** for authentication. This method allows your application to authenticate to the SQL Server database using a service principal and a certificate, which provides a secure and automated way to manage authentication.

Selecting **Azure Service Principal Cert** requires you to specify the following:

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

  For instance, contoso.onmicrosoft.com. Alternatively, specify the tenant Id. This value is the directory Id in the Azure Portal > Azure Active Directory > Properties.

  It is a good practice to specify the Tenant, although in general things should normally work without having to specify it.

  The Azure Tenant is required when setting **OAuth Grant Type** to **CLIENT**. When using client credentials, there is no user context. The credentials are taken from the context of the app itself. While Microsoft still allows client credentials to be obtained without specifying which Tenant, it has a much lower probability of picking the specific tenant you want to work with. For this reason, we require **Azure Tenant** to be explicitly stated for all client credentials connections to ensure you get credentials that are applicable for the domain you intend to connect to.

- OAuth JWT Cert – enter the JWT Certificate store.

  This is the name of the certificate store for the client certificate.

  The **OAuth JWT Cert Type** (see below) field specifies the type of the certificate store specified by **OAuth JWT Cert**. If the store is password protected, specify the password in **OAuth JWT Cert Password** (see below)

OAuth JWT Cert is used in conjunction with the **OAuth JWT Cert Subject** (see below) field in order to specify client certificates.

If **OAuth JWT Cert** has a value, and **OAuth JWT Cert Subject** is set, a search for a certificate is initiated. Please refer to the **OAuth JWT Cert Subject** field for details.

- OAuth JWT Cert Type - click the dropdown to select from the list the type of key store containing the TWT Certificate. The options available for selection are as follows:

  - USER
  - MACHINE
  - PFXFILE
  - PFXBLOB
  - JKSFILE
  - JKSBLOB
  - PEMKEY_FILE
  - PEKMEY_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 this setting and its properties can be provided.

- OAuth JWT Cert Password – enter the password for the OAuth JWT certificate.

  If the certificate store is of a type that requires a password, this property is used to specify that password in order to open the certificate store.

- OAuth JWT Cert Subject – enter the subject of the OAuth JWT certificate.

  When loading a certificate the subject is used to locate the certificate in the store.

  If an exact match is not found, the store is searched for subjects containing the value of the property.

  If a match is still not found, the property is set to an empty string, and no certificate is selected.

  The special value "*" picks the first certificate in the certificate store.

  The certificate subject is a comma separated list of distinguished name fields and values. For instance "CN=www.server.com, OU=test, C=US"

- OAuth Client Id – enter the client ID assigned when you registered your application with an OAuth authorization server.

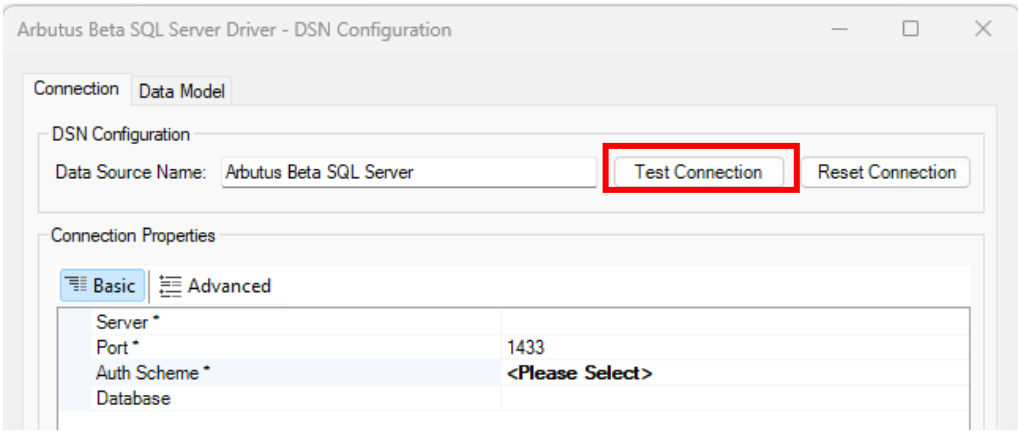
  As part of registering an OAuth application, you will receive the **OAuth Client Id** value, sometimes also called a consumer key, and a client secret, the **OAuth Client Secret** (see above).

4. **Database** – enter the name of the SQL Server database

E2. Editing the DSN properties in the Advanced tab

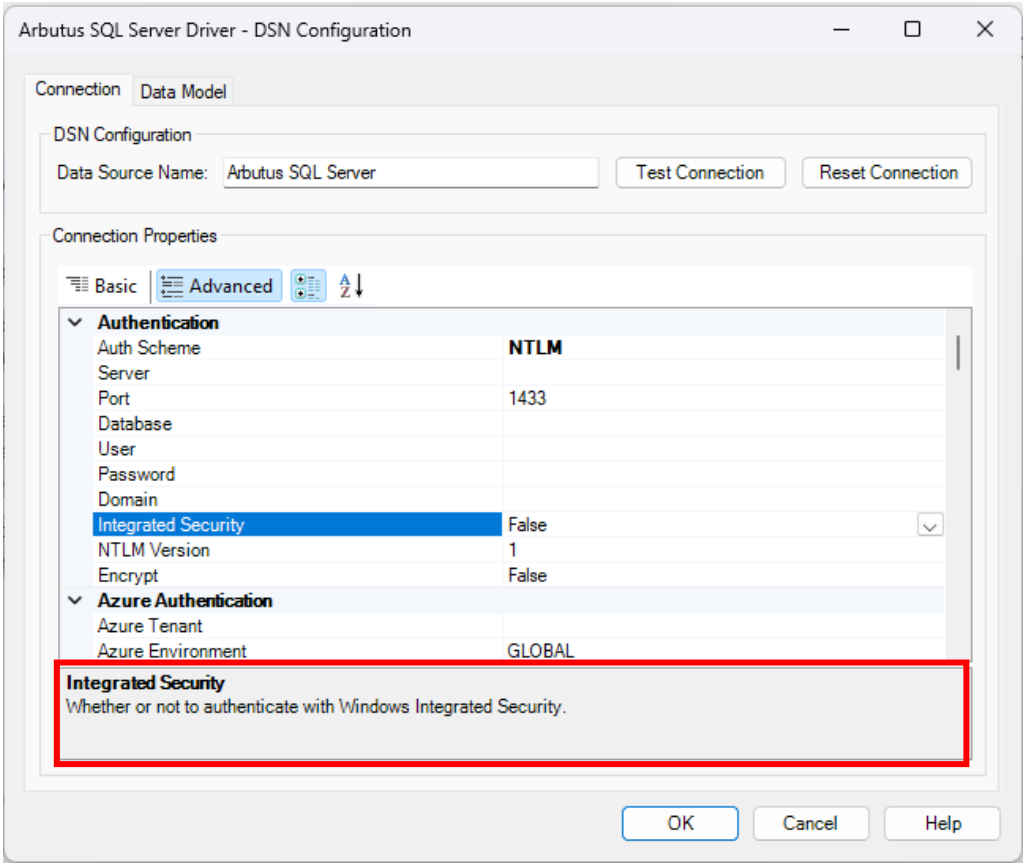
This tab includes more detailed and technical properties. It is intended for those users who need more control over the configuration and are comfortable with more complex options. The **Advanced** tab often includes properties that can fine-tune the behaviour of the system feature.

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 SQL Server by clicking the **Test Connection** button.



There are a lot of 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.



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.