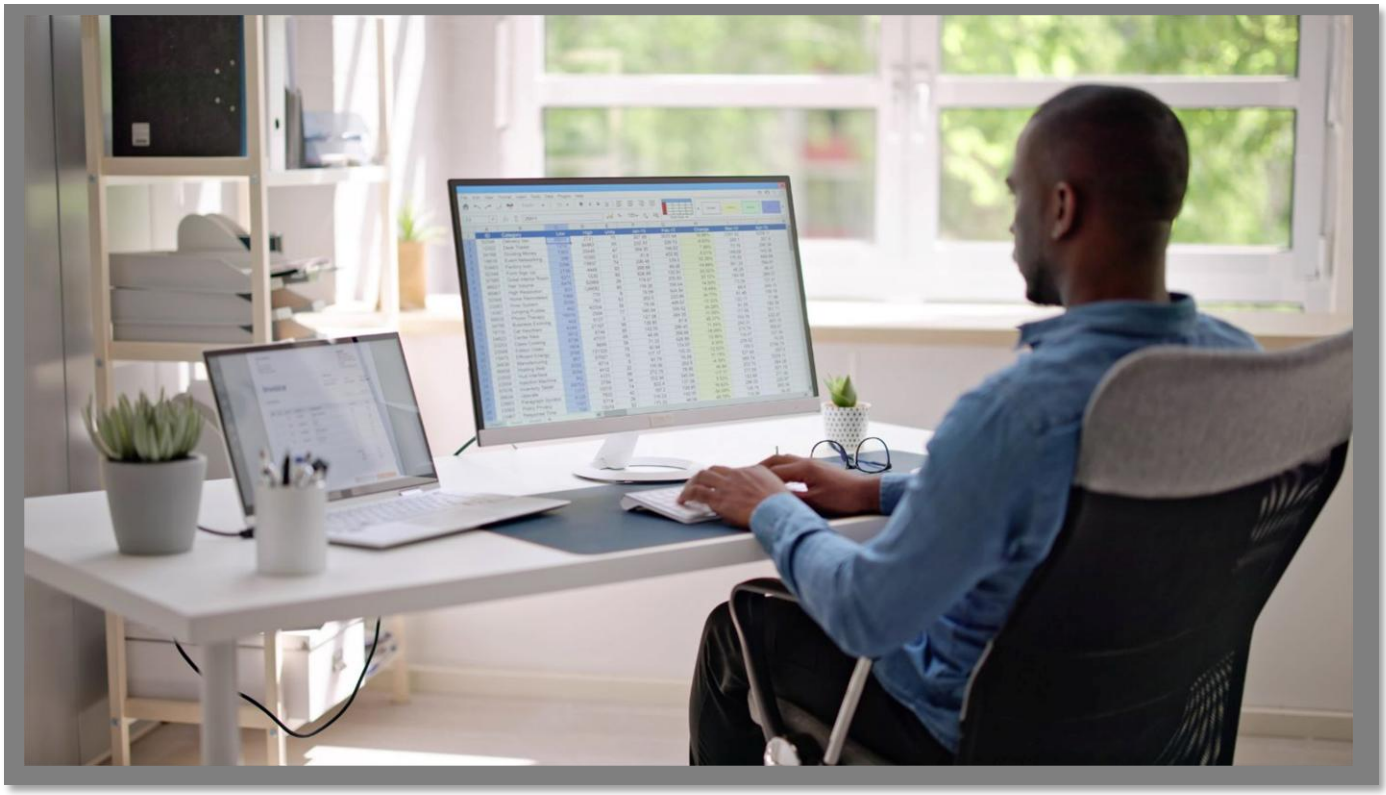
Arbutus Connectors

NetSuite
CONFIGURATION GUIDE



ARBUTUS
Powerful Analytics Simplified

Arbutus Connectors

Contents

Arbutus Connector – NetSuite

A. Introduction

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

NetSuite is a comprehensive cloud-based enterprise resource planning (ERP) software that helps businesses manage core operations like accounting, finance, inventory, customer relationship management (CRM), and e-commerce. It integrates various business processes into a single platform, providing real-time data and automation to improve efficiency. NetSuite is widely used by companies of all sizes for financial management, reporting, and scalability.

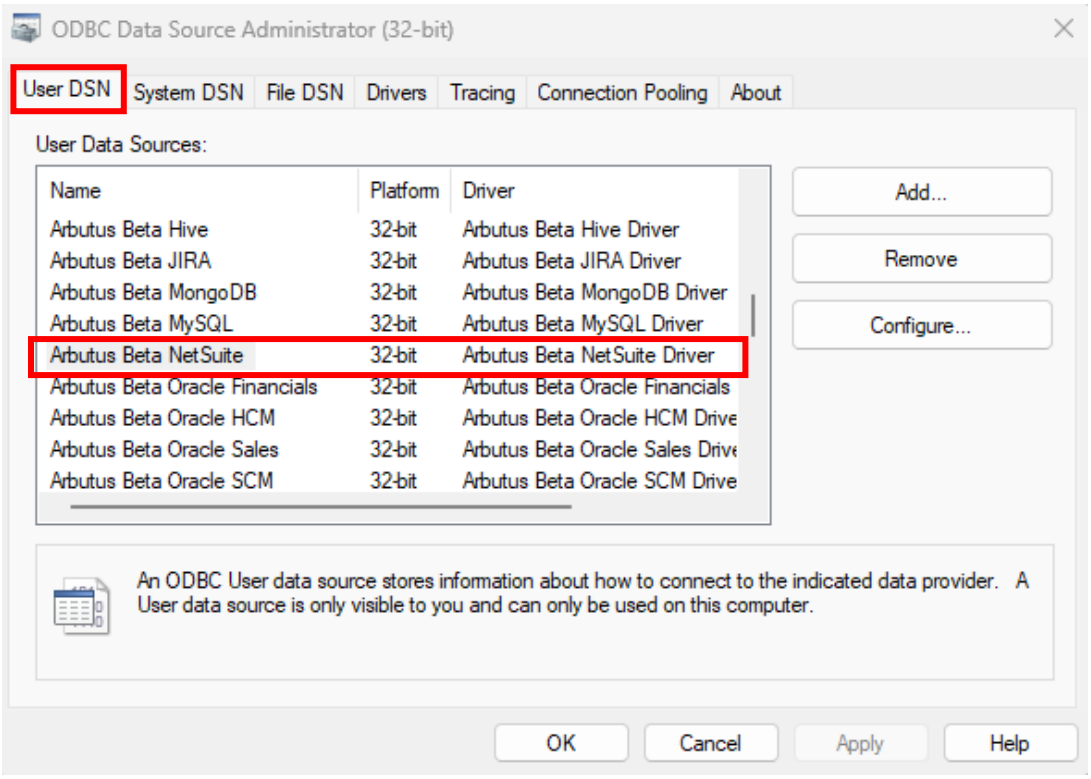
In NetSuite, data is stored in Oracle-managed cloud databases, ensuring accessibility from anywhere with an internet connection.

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus NetSuite 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 NetSuite 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.

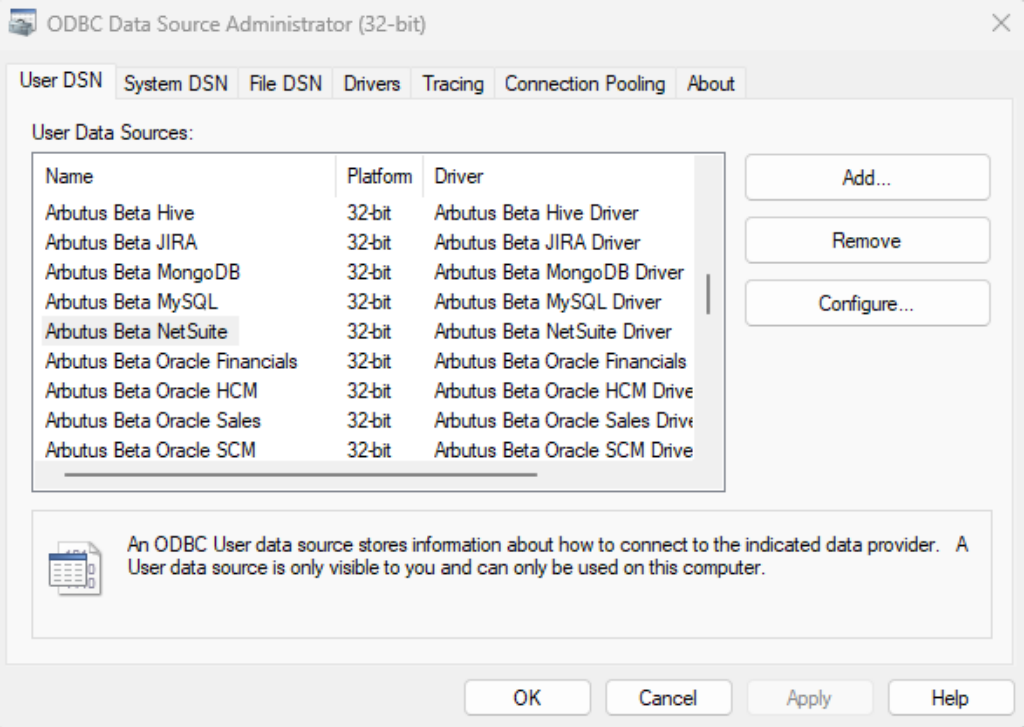
Each of these DSNs has a tab in the **ODBC Data Source Administrator** dialog.

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



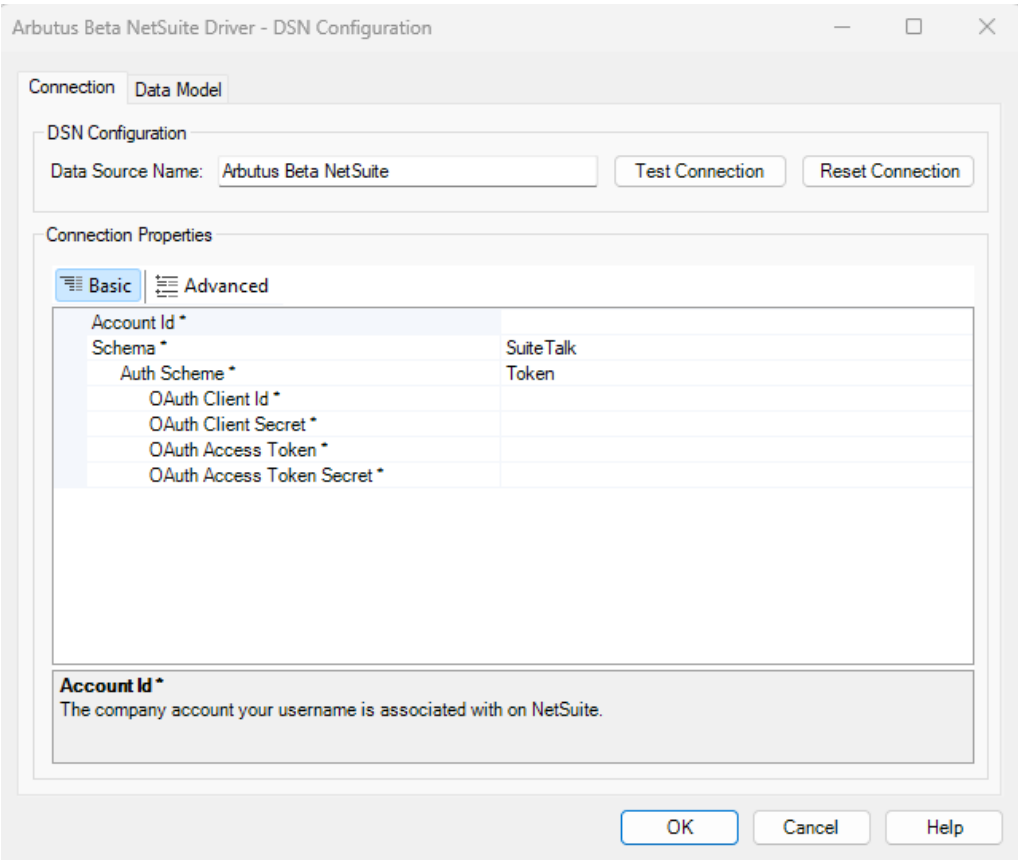
2. Click the **User DSN** tab.

   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 NetSuite.**

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

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



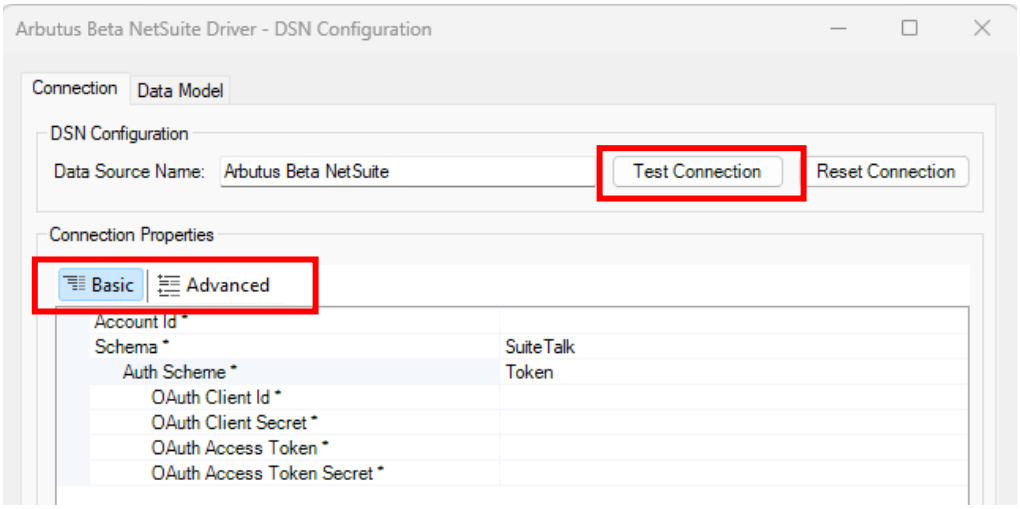
E. Editing the DSN properties – the Basic and Advanced tabs

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 **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 NetSuite system by clicking the **Test Connection** button, as highlighted in the screenshot below.



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

1. **Account Id** – this is the company account your username is associated with on NetSuite.

   Together with **User** and **Password**, this field is used to authenticate to NetSuite.

2. **Schema** – click the dropdown to select from the list the type of schema to use. The options available for selection are as follows:
   - SuiteTalk – select this if you need to both **retrieve and modify data**. SuiteTalk is the older SOAP-based API that supports a wide range of entities and full CRUD (Create, Read, Update, Delete) operations – uses SOAP to enable communication between NetSuite and other systems, allowing for operations such as creating, retrieving, updating, and deleting records.

While **SuiteTalk is NetSuite's implementation of the SOAP API, SuiteScript** is NetSuite's JavaScript-based API that allows for customization and automation within the NetSuite environment. While SuiteTalk is used for external integrations, SuiteScript is used for internal customizations.

Note: SOAP = Simple Object Access Protocol

However, if your primary goal is to **retrieve data** with better performance, you might want to consider using **SuiteQL** instead. SuiteQL is a newer, SQL-like API that offers more efficient data retrieval, including support for JOINs, GROUP BY, and aggregations.

The default value is **SuiteTalk**.

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

- Auth Scheme
  - OAuth Client Id
  - OAuth Client Secret
  - OAuth Access Token
  - OAuth Access Token Secret

For more information on these underlying properties, please see the section below **on Configuring the Authentication Scheme when SuiteTalk is selected as the Schema**.

- SuiteQL – select this if your primary goal is to **retrieve data efficiently**.

  SuiteQL is the foundation of NetSuite's REST Query Service, meaning that if you use REST API to fetch data from NetSuite, you will be writing SuiteQL queries – it is the query language used in NetSuite's REST Query API.

Without SuiteQL enabled, the ODBC connection will use traditional NetSuite schema, which is not as flexible and does not align with REST API query structures.

If you need to perform both data retrieval and modifications (CRUD operations), you should use SuiteTalk instead, as **SuiteQL** does not support data modification. It is a read-only query language designed for retrieving data from NetSuite.

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

- Auth Scheme
  - OAuth Client Id
  - OAuth Client Secret
  - OAuth Access Token
  - OAuth Access Token Secret

For more information on these underlying properties, please see the section below on **Configuring the Authentication Scheme when SuiteQL is selected as the Schema**.

Configuring the Authentication Scheme when SuiteTalk is selected as the Schema

Selecting **SuiteTalk** as the **Schema** requires you to specify the following:

- Auth Scheme – this is a dropdown selection to specify the type of authentication to use when connecting to NetSuite. The two options available for selection are:
  - Token – set this to perform Token-Based Authentication (TBA). TBA is a secure and flexible method for authenticating API requests to NetSuite.

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

- 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 when you register your application with an OAuth authorization server.

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

- OAuth Access Token – this is the access token for connecting using OAuth, granting the user access. The access token is used in place of the user's login ID and password, which stay on the server.

  The OAuth access token has a server-dependent timeout, limiting user access. This is set using the **OAuth Expires In** property (see the **Advanced** tab in the ODBC Data Source Administrator DSN Configuration dialog). However, it can be reissued between requests to keep access alive as long as the user keeps working.

- OAuth Access Token Secret – this is the OAuth access token for connecting using OAuth.

  The **OAuth Access Token Secret** property is used to connect and authenticate using OAuth. The **OAuth Access Token Secret** is retrieved from the OAuth server as part of the authentication process. It is used with the **OAuth Access Token** and can be used for multiple requests until it times out.

The default value is **Token**.

- Basic – set this to use BASIC user/password authentication. This method involves using a NetSuite username and password to authenticate your connection.

  Selecting **Basic** requires you to specify the following:
  - User – this is the user of the NetSuite account used to authenticate. Together with **Account Id** and **Password** (see below), this field is used to authenticate to NetSuite.

  - Password – this is the password of the NetSuite user used to authenticate. Together with **User** (see above) and **Account ID**, this field is used to authenticate to NetSuite.

  It is important to note that Basic Authentication is less secure compared to TBA (Token-Based Authentication), as it involves sending credentials over the network. For production environments or integrations requiring higher security, TBA is generally recommended.

Configuring the Authentication Scheme when SuiteQL is selected as the Schema

Selecting **SuiteQL** as the **Schema** requires you to specify the following:

- Auth Scheme – this is a dropdown selection to specify the type of authentication to use when connecting to NetSuite. The two options available for selection are:
  - Token – set this to perform Token-Based Authentication (TBA). TBA is a secure and flexible method for authenticating API requests to NetSuite.

    Selecting **Token** requires you to complete the same set of properties as described in the section above where **SuiteTalk as selected as the Schema**. You may refer to those properties for more information.

    The default value is **Token**.

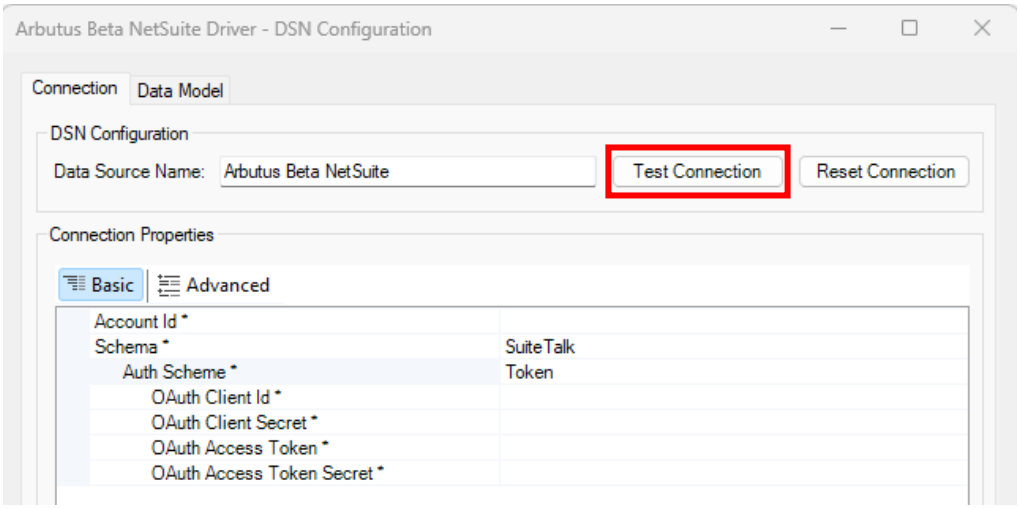
  - OAuth – set this to perform OAuth authentication. This is essentially token based authentication where the token is dynamically created instead of directly in the NetSuite UI.

    Selecting **OAuth** does not require you to complete editing any underlying properties.

E2. Editing the DSN properties in the Advanced tab

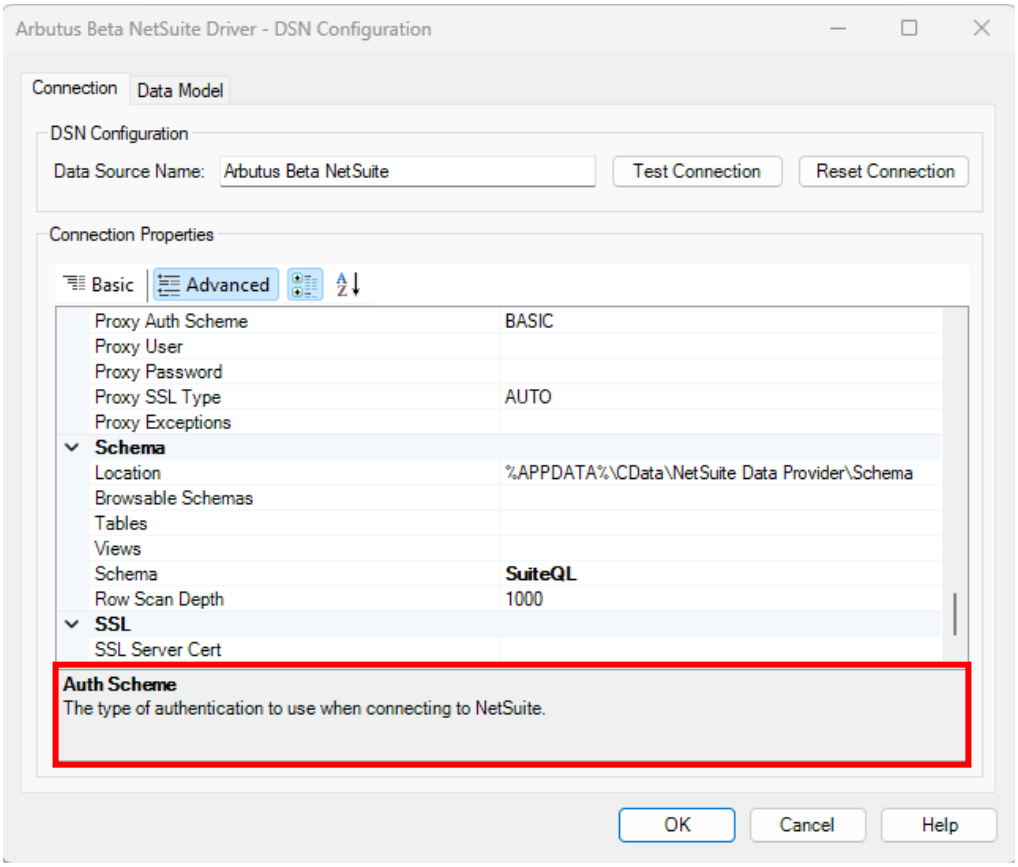
This tab includes more detailed and technical properties. It is intended for those users who need more control over the configuration and are comfortable with more complex options. The **Advanced** tab often includes properties that can fine-tune the behaviour of the system feature.

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 NetSuite system by clicking the **Test Connection** button.



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.



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.