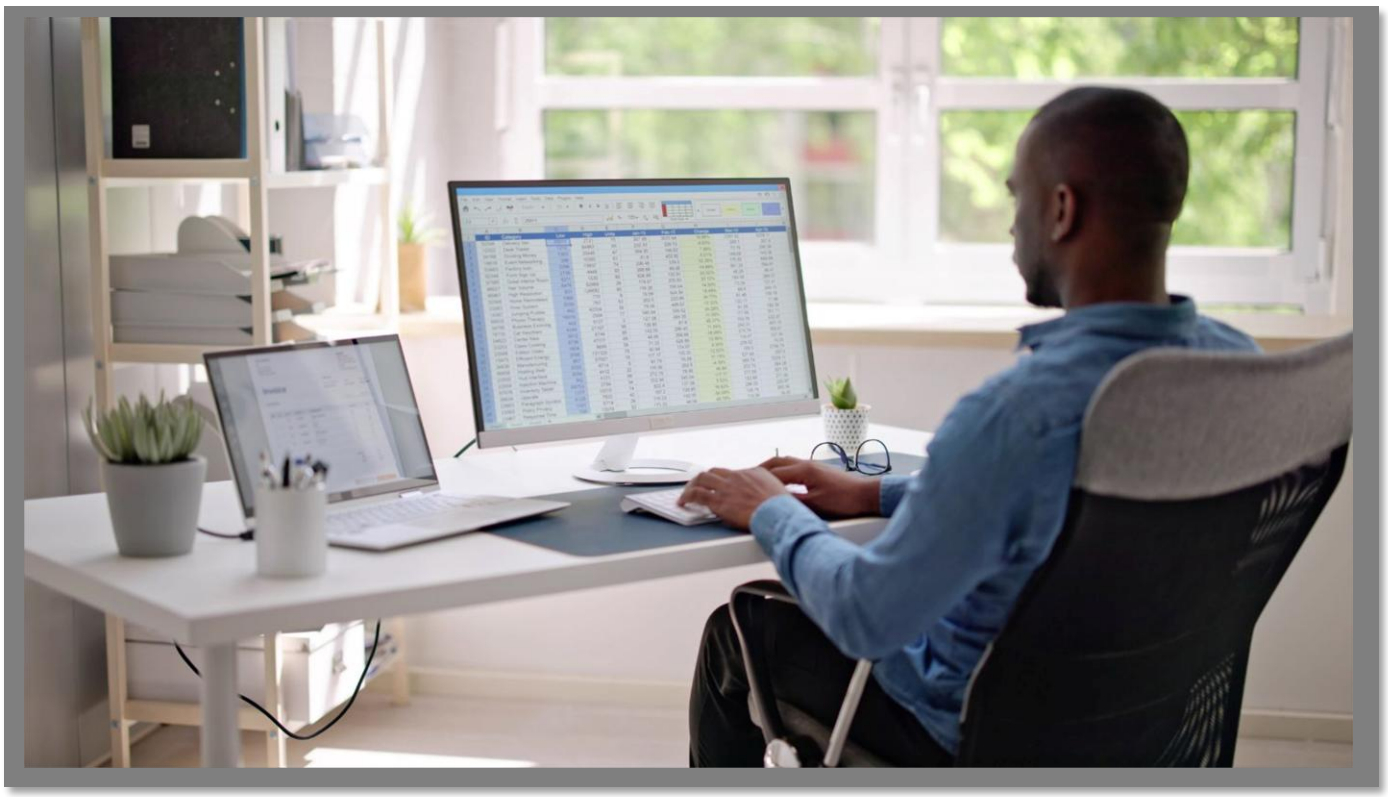
Arbutus Connectors

SAP Netweaver Gateway
CONFIGURATION GUIDE



ARBUTUS
Powerful Analytics Simplified

Contents

Arbutus Connector – SAP Netweaver Gateway

A. Introduction

The purpose of this Guide is to provide assistance with configuring the Arbutus SAP Netweaver Gateway 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 SAP Netweaver Gateway

SAP Netweaver and SAP Gateway

SAP NetWeaver is the main (web-based) platform that supports many SAP programs. It acts like a central hub that helps different SAP systems and outside applications share information and work together. SAP Gateway, often associated with the ODBC driver used for data access, is a part of this platform. It works like a bridge that lets other tools and applications connect safely to SAP data using standard methods such as ODBC. This makes it possible for outside applications to securely reach and use SAP information.

How the ODBC driver fits in

Even though the ODBC driver might be called "SAP Gateway," it actually works within the larger SAP NetWeaver system. The driver's job is to help data move between SAP and other applications. In simple terms, SAP NetWeaver provides the environment, and SAP Gateway (through the ODBC driver) provides the door that allows data to flow in and out.

How SAP Netweaver stores and supports data

Inside SAP NetWeaver, information is stored in organized tables—similar to a collection of spreadsheets. Each table holds different kinds of business data, such as details about customers, products, or transactions. These tables are connected so that every part of the system can use the same accurate information.

How SAP Netweaver is different from other SAP products

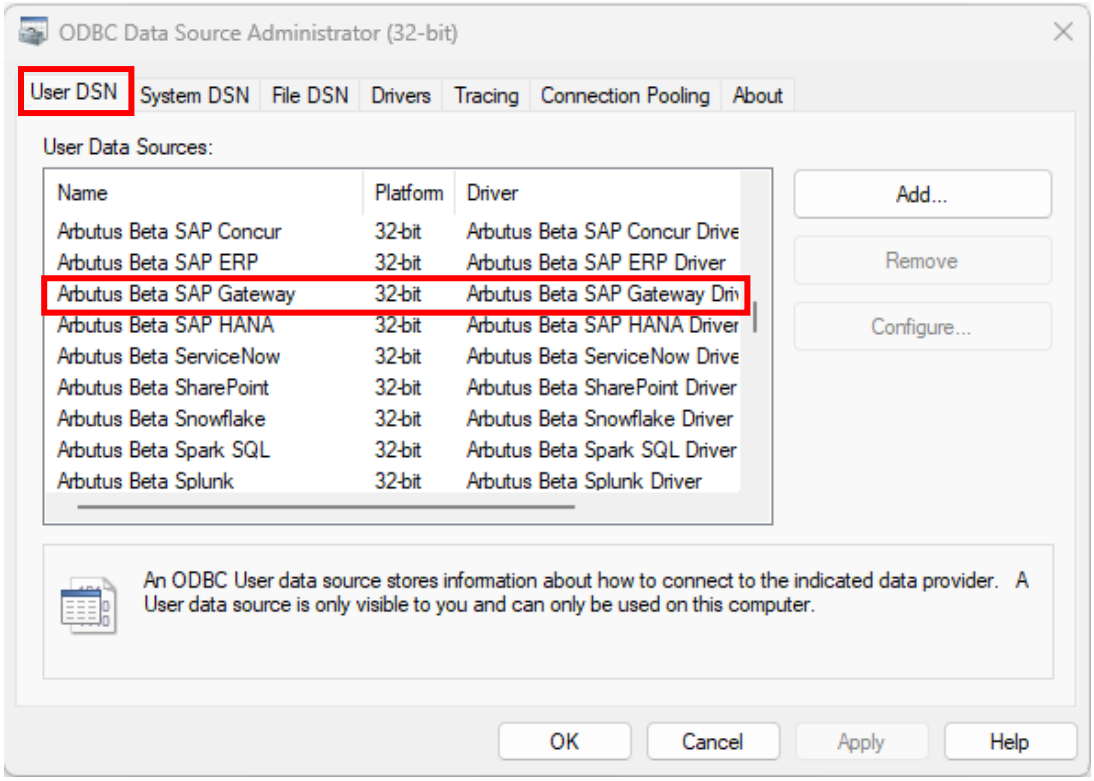
SAP NetWeaver is different from other SAP products like SAP Concur (used for travel and expenses), SAP HANA (a type of fast database), and SAP ERP (used for running daily business operations). It is not a product that you use directly like SAP Concur, SAP HANA, or SAP ERP. Instead, NetWeaver is the base that helps these products work together. Its main role is to provide the foundation that connects these systems and supports tasks like application development, data access, and communication between different parts of an organization.

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus SAP Netweaver Gateway 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 SAP Gateway 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.

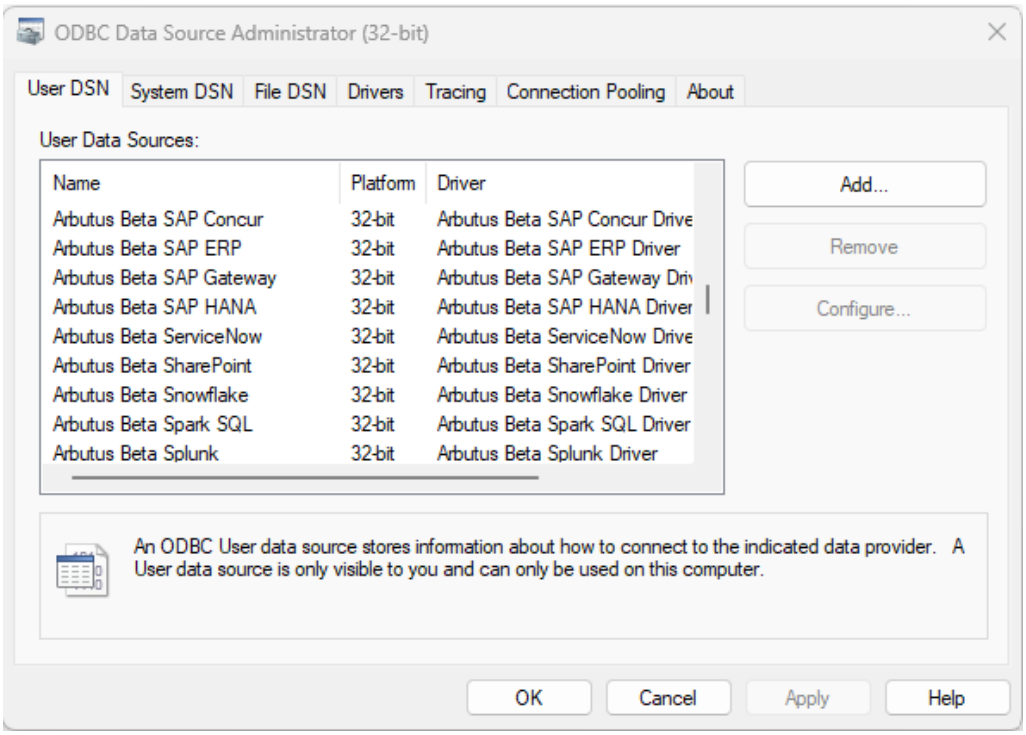
User and system data sources are collectively known as *machine* data sources because they are local to a computer.

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

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

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

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



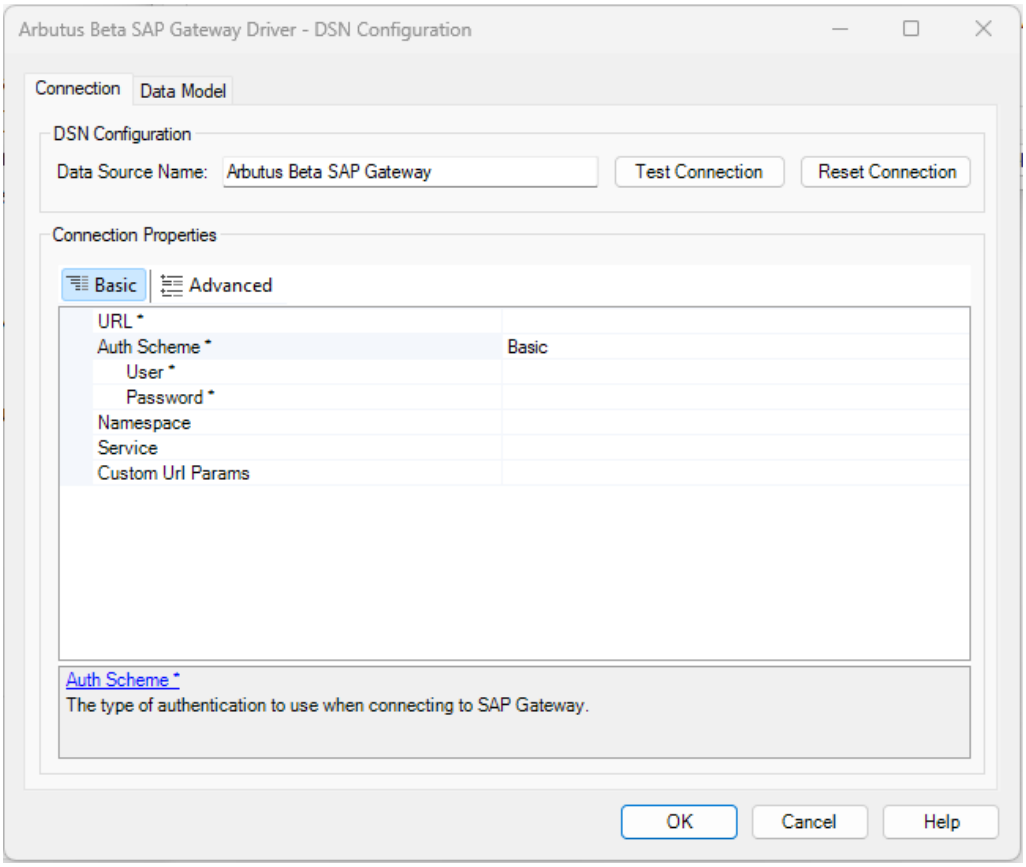
2. Click the **User DSN** tab.

   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 SAP Gateway.**

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

   This opens the **Arbutus SAP Gateway Driver – DSN Configuration** dialog.



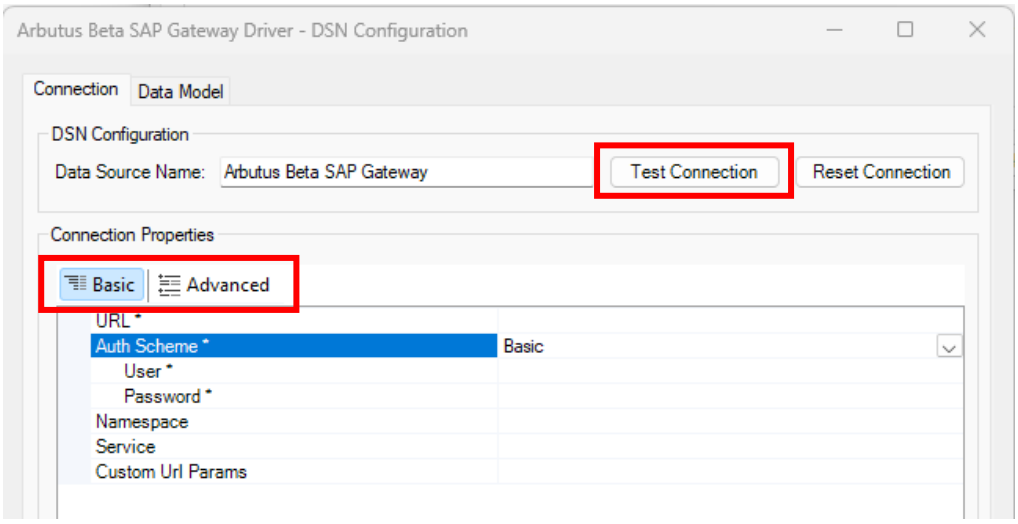
E. Editing the DSN properties – the Basic and Advanced tabs

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 **Basic** setting for the **Auth Scheme** (per screenshot below) to specify the type of authentication to use when connecting to SAP Gateway.

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 on the **Basic** tab, you can go ahead and try testing the connection to the SAP Netweaver Gateway system by clicking the **Test Connection** button, as highlighted in the screenshot below.



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

1. **URL** – set this to the URL your SAP Gateway environment or to the full URL of the OData service. Any additional properties must be added using the **Custom URL Params** property (see below).

2. **Auth Scheme** – click the dropdown to select from the list the appropriate scheme used for authentication – the type of authentication to use when connecting to SAP Gateway. The options available for selection are as follows:
   - Basic – select this if you are using Basic user / password authentication. This is the most common setup for test, development, or controlled internal environments where secure network connections (such as HTTPS) are already in place. This method involves using a username and password to authenticate your connection. While this method is straightforward, it may not be **the most secure** especially if the connection is not encrypted. For production environments or integrations requiring higher security, more advanced authentication methods are generally recommended

     Selecting **Basic** requires you to specify the User and Password.
     - User – this specifies the User ID of the authenticating SAP Gateway user account. The authenticating server requires both **User** and **Password** (see below) to validate the user's identity.

     - Password – this specifies the password of the authenticating user account. The authenticating server requires both **User** (see above) and **Password** to validate the user's identity.

     The default value is **Basic**.

- OAuth – select this as the authentication method when you need a secure and modern way to connect external applications to SAP, especially in environments where single sign-on (SSO) or token-based access is preferred over traditional username/password methods, e.g. **Basic** authentication. In short, **OAuth** helps ensure that only authorized users or systems can access SAP data.

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

  - OAuth Client Id – this specifies the client ID (also known as the consumer key) assigned to your custom OAuth application. This ID is required to identify the application to the OAuth authorization server during authentication.

    This property is required in two cases:

    - When using a custom OAuth application, such as in web-based authentication flows, service-based authentication, or certificate-based flows that require application registration.
    - If the driver does not provide embedded OAuth credentials.

    (When the driver provides embedded OAuth credentials, this value may already be provided by the driver and thus not require manual entry.)

    OAuth Client Id is generally used alongside other OAuth-related properties such as **OAuth Client Secret** (see below) and **OAuth Settings Location** (this can be found under the **Advanced** tab of the ODBC Data Source Administrator dialog – see the **OAuth** section) when configuring an authenticated connection.

OAuth Client Id is one of the key connection parameters that need to be set before users can authenticate via **OAuth**. You can usually find this value in your identity provider's application registration settings. Look for a field labeled **Client ID**, **Application ID**, or **Consumer Key**.

While the client ID is not considered a confidential value like a client secret, it is still part of your application's identity and should be handled carefully. Avoid exposing it in public repositories or shared configuration files.

- OAuth Client Secret – this specifies the client secret assigned to your custom OAuth application. This confidential value is used to authenticate the application to the OAuth authorization server. (Custom OAuth applications only.).

  This property (sometimes called the application secret or consumer secret) is required when using a custom OAuth application in any flow that requires secure client authentication, such as web-based OAuth, service-based connections, or certificate-based authorization flows. It is not required when using an embedded OAuth application.

  The client secret is used during the token exchange step of the OAuth flow, when the driver requests an access token from the authorization server. If this value is missing or incorrect, authentication fails with either an *invalid_client* or an *unauthorized_client* error.

  OAuth Client Secret is one of the key connection parameters that need to be set before users can authenticate via OAuth. You can obtain this value from your identity provider when registering the OAuth application.

- Scope – the permission scopes in space separated values. This is required when authenticating via OAuth2.0 flow.

  Scopes are set to define what kind of access the authenticating user will have; for example, read, read and write, restricted access to sensitive information. System administrators can use scopes to selectively enable access by functionality or security clearance.

  - When **Initiate OAuth** (this can be found under the Advanced tab of the ODBC Data Source Administrator dialog – see the **OAuth** section) is set to **GETANDREFRESH**, you must use this property if you want to change which scopes are requested.

  - When **Initiate OAuth** is set to either **REFRESH** or **OFF**, you can change which scopes are requested using either this property or the Scope input.

  Scopes are set to define what kind of access the authenticating user will have; for example, read, read and write, restricted access to sensitive information. System administrators can use scopes to selectively enable access by functionality or security clearance.

- Token – set this to authenticate using an **API Key** or application-token. An API key is like a digital password or access card that lets an app or program talk to another system securely. Selecting the **Token** option means you're choosing to authenticate using an **API Key** rather than a username/password (Basic authentication) or OAuth flow. You may use the Token option when your SAP Gateway environment is set up to accept API Key-based authentication. The API Key acts as a secure identifier for the application making the request, and it's passed in the connection setup to validate access.

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

- API Key – set this to the key of the application you have created. If you set this, User and Password are not used for authentication.

- OAuth PKCE – use this when connecting via OAuth 2.0 using the PKCE (Proof Key for Code Exchange) flow, typically when no client secret is used. It is a secure version of the OAuth 2.0 Authorization Code flow. It's designed for scenarios where the client (like your desktop app using the ODBC driver) **cannot safely store secrets**, such as a **client ID and secret**. Instead, it uses a temporary one-time code (called a "code verifier") to securely complete the login process.

  Selecting **OAuth PKCE** requires you to specify the following:

  - OAuth Client ID – this specifies the client ID (also known as the consumer key) assigned to your custom OAuth application. This ID is required to identify the application to the OAuth authorization server during authentication.

    For more information, please see this same setting under the **OAuth** section covered above.

- SAP BTP – set this to retrieve authentication details from your configured SAP BTP (Business Technology Platform) Destination. This means the driver is able to offload the authentication management to the BTP Destination service, so you don't manually manage the user credentials, tokens, or client secrets in the ODBC setup. It's especially useful when you want to simplify credential management: you just reference the BTP Destination and let BTP handle the token retrieval and refresh automatically.

The **Destination URL** and **Destination Name** settings for the **SAP BTP Authentication** can be found under the **Advanced** tab of the ODBC Data Source Administrator dialog – see the **SAP BTP Authentication** section.

Selecting **SAP BTP** does **NOT** require you to specify any further property requirements.

3. **Namespace** – this is to specify the service namespace you want to retrieve data from. For example, in the external OData service, *http://services.odata.org/Northwind/Northwind.svc/* , 'Northwind' is the Service Namespace. This is required if the service is not specified in the URL.

4. **Service** – this is to specify one or a comma-delimited list of services you want to retrieve data from. If the service is set in the URL, this property is ignored. In cases where the service is neither set in the URL nor via this property, the SAP Gateway's catalog service is used to discover all the available services.

5. **Custom URL Params** – this is a string of custom URL parameters to be included with the HTTP request, in the form field1=value1&field2=value2&field3=value3.

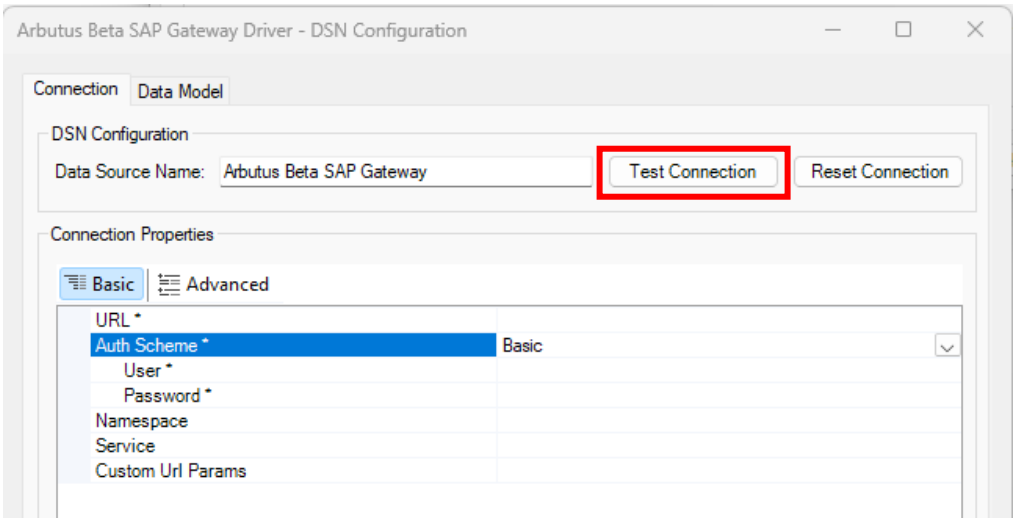
   This property enables you to specify custom query string parameters that are included with the HTTP request. The parameters must be encoded as a query string in the form field1=value1&field2=value2&field3=value3, where each value is URL encoded. URL encoding converts the characters in the string that can be transmitted over the internet as follows:

   - Non-ASCII characters are replaced with their equivalent in the form of a "%" followed by two hexadecimal digits.
   - Spaces are replaced with either a plus sign (+) or %20.

E2. Editing the DSN properties in the Advanced tab

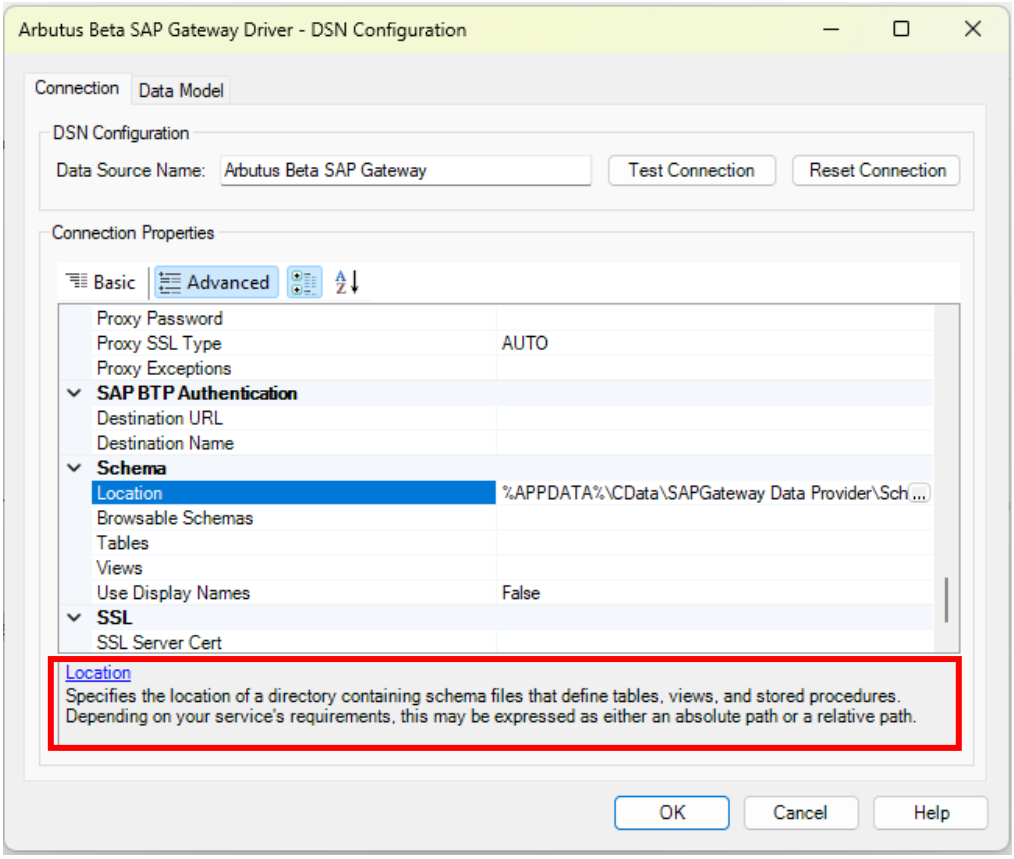
This tab includes more detailed and technical properties. It is intended for those users who need more control over the configuration and are comfortable with more complex options. The **Advanced** tab often includes properties that can fine-tune the behaviour of the system feature.

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 SAP Netweaver Gateway 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.