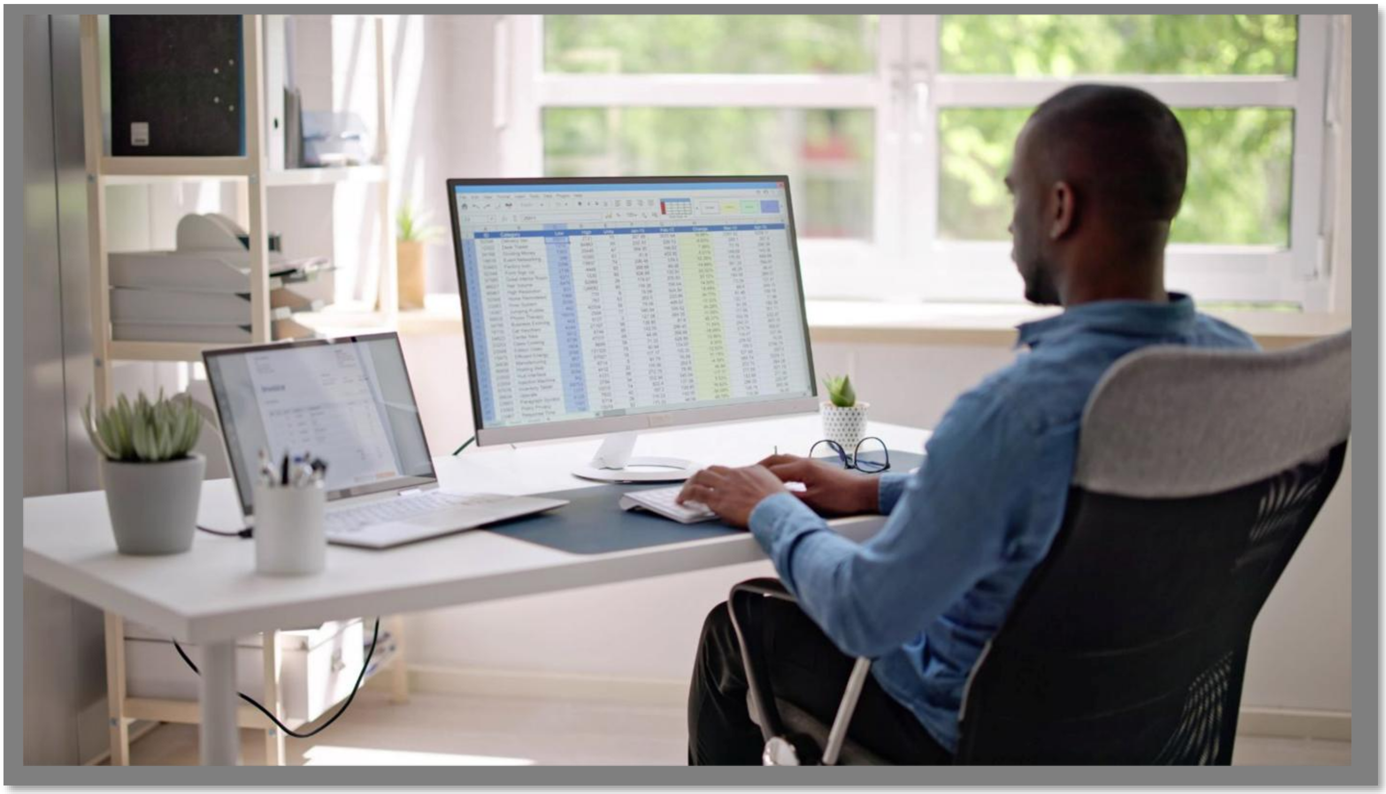
Arbutus Connectors

CouchBase

CONFIGURATION GUIDE



ARBUTUS

Powerful Analytics Simplified

Arbutus Connectors

Contents

Arbutus Connector – CouchBase

A. Introduction

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

Couchbase is a high-performance distributed, NoSQL database designed for scalability, flexibility, and low-latency access. It combines key-value and document storage with a powerful SQL-like query language (N1QL) for working with JSON data. Couchbase is ideal for real-time applications, caching, and big-data use cases. It is used in cloud, on-premises, and hybrid environments, including its managed service, Couchbase Capella.

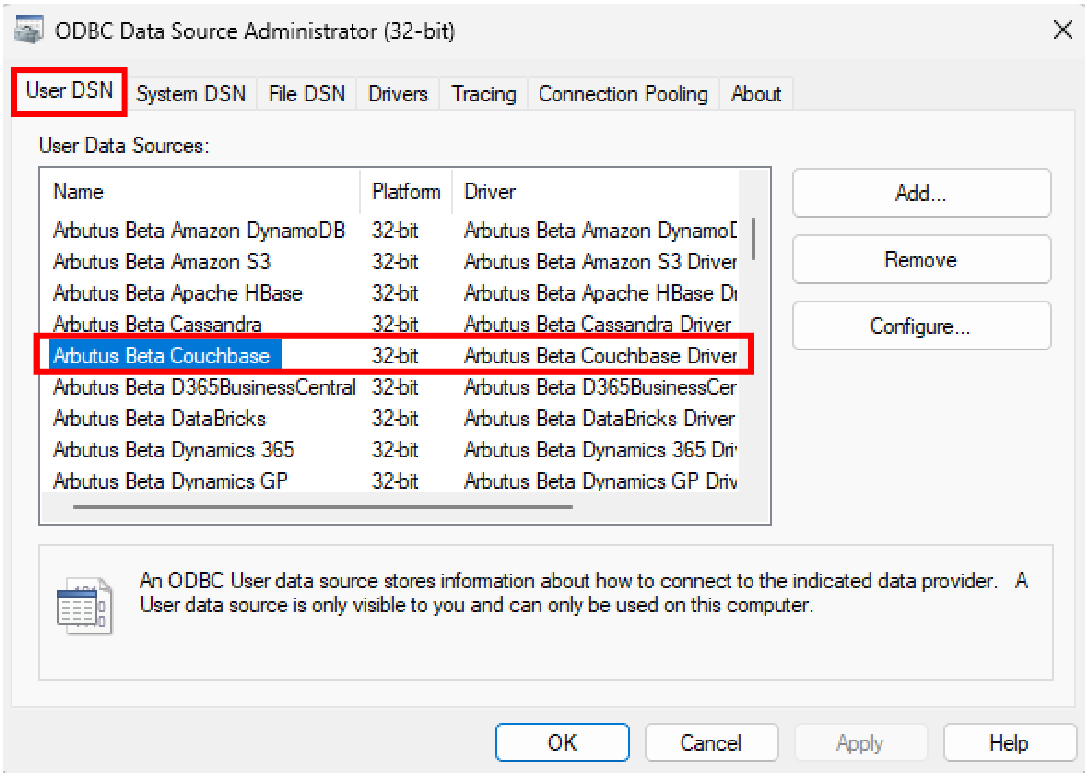
In Couchbase, data is stored as JSON documents within buckets and is distributed across nodes in a clustered architecture. There are different types of buckets, e.g., Couchbase Buckets. Each document has a unique key and is stored in JSON format. Documents can contain structured or semi-structured data.

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus CouchBase 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**.



User Data Sources:

Name	Platform	Driver
Arbutus Beta Amazon DynamoDB	32-bit	Arbutus Beta Amazon DynamoD
Arbutus Beta Amazon S3	32-bit	Arbutus Beta Amazon S3 Driver
Arbutus Beta Apache HBase	32-bit	Arbutus Beta Apache HBase Di
Arbutus Beta Cassandra	32-bit	Arbutus Beta Cassandra Driver
Arbutus Beta Couchbase	32-bit	Arbutus Beta Couchbase Driver
Arbutus Beta D365BusinessCentral	32-bit	Arbutus Beta D365BusinessCer
Arbutus Beta DataBricks	32-bit	Arbutus Beta DataBricks Driver
Arbutus Beta Dynamics 365	32-bit	Arbutus Beta Dynamics 365 Dri
Arbutus Beta Dynamics GP	32-bit	Arbutus Beta Dynamics GP Driv

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 CouchBase 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.

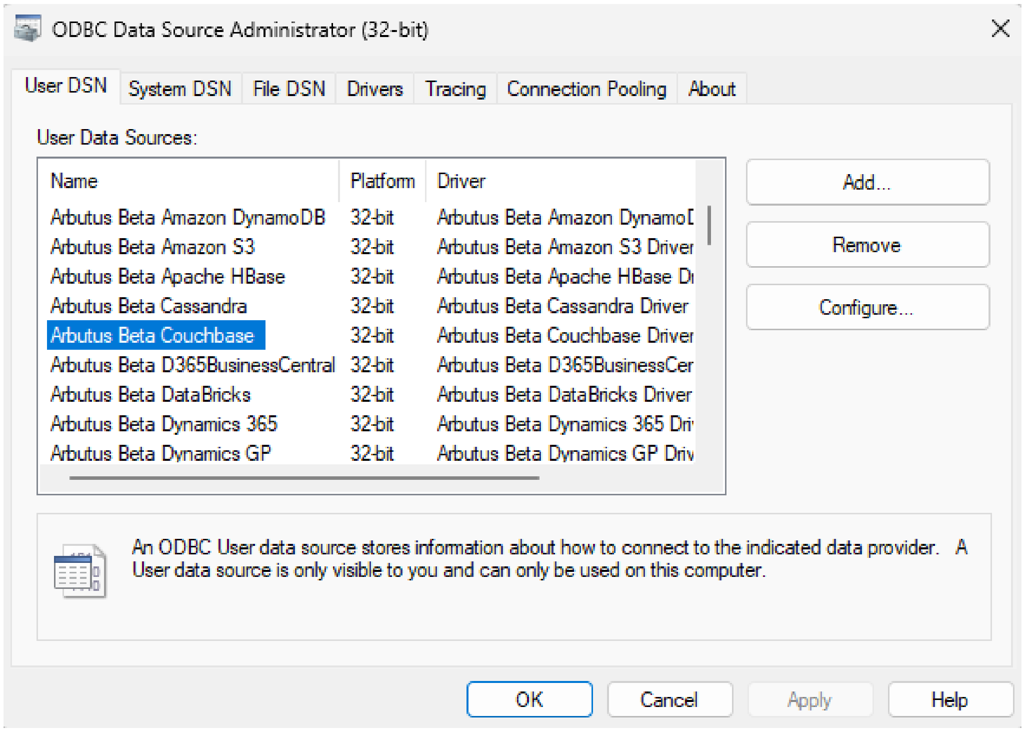
Each of these DSNs has a tab in the **ODBC Data Source Administrator** dialog.

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



Name	Platform	Driver
Arbutus Beta Amazon DynamoDB	32-bit	Arbutus Beta Amazon Dynamo[
Arbutus Beta Amazon S3	32-bit	Arbutus Beta Amazon S3 Driver
Arbutus Beta Apache HBase	32-bit	Arbutus Beta Apache HBase Dı
Arbutus Beta Cassandra	32-bit	Arbutus Beta Cassandra Driver
Arbutus Beta Couchbase	32-bit	Arbutus Beta Couchbase Driver
Arbutus Beta D365BusinessCentral	32-bit	Arbutus Beta D365BusinessCer
Arbutus Beta DataBricks	32-bit	Arbutus Beta DataBricks Driver
Arbutus Beta Dynamics 365	32-bit	Arbutus Beta Dynamics 365 Dri
Arbutus Beta Dynamics GP	32-bit	Arbutus Beta Dynamics GP Driv

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.

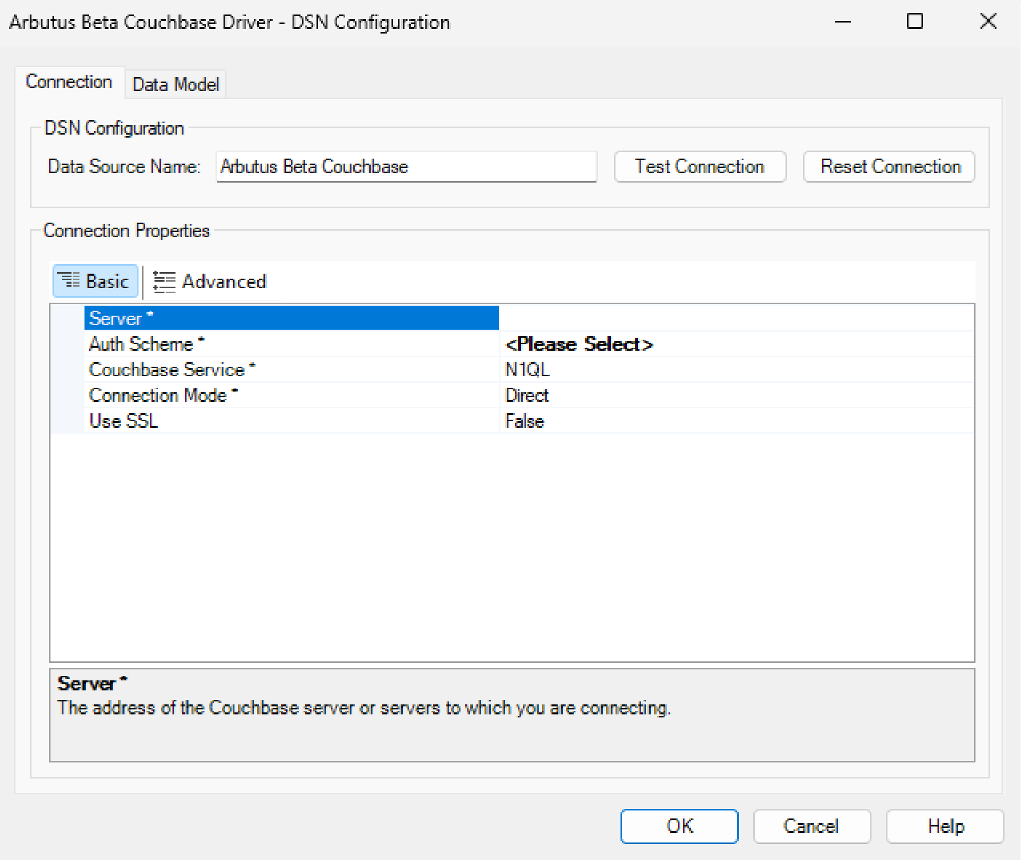
2. Click the **User DSN** tab.

   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 CouchBase.**

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

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



Data Source Name: Arbutus Beta Couchbase

Server *	
Auth Scheme *	**<Please Select>**
Couchbase Service *	N1QL
Connection Mode *	Direct
Use SSL	False

Server *
The address of the Couchbase server or servers to which you are connecting.

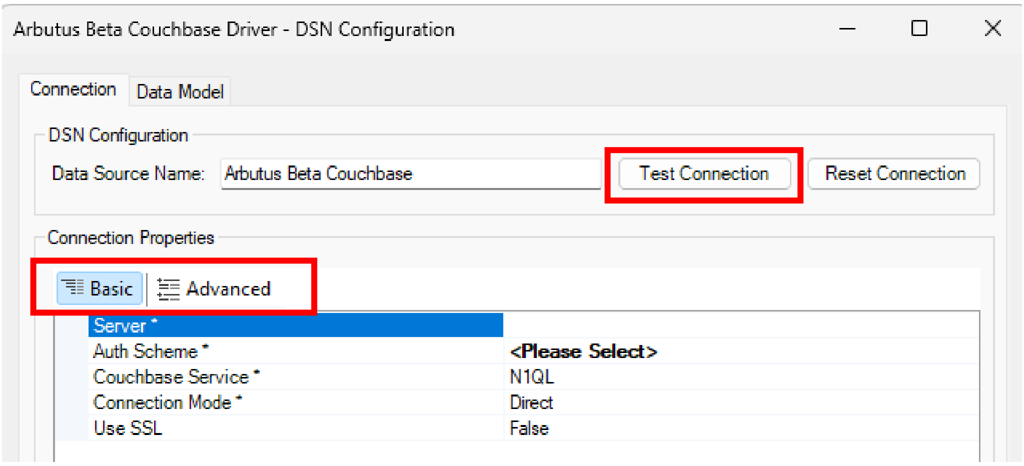
E. Editing the DSN properties – the Basic and Advanced tabs

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



Data Source Name: Arbutus Beta Couchbase

Server *	
Auth Scheme *	<Please Select>
Couchbase Service *	N1QL
Connection Mode *	Direct
Use SSL	False

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

1. **Server** – this is the address of the Couchbase server or servers to which you are connecting.

   This value can be set to a hostname or an IP address, like "couchbase-server.com" or "1.2.3.4". It can also be set to an HTTP or HTTPS URL, such as *"https://couchbase-server.com" or "http://1.2.3.4*". If **ConnectionMode** (see below) is set to Cloud then this should be the hostname of the Couchbase Cloud instance as reported in the control panel.

   If the URL form is used, then setting this option will also set the **UseSSL** (see below) option: if the URL scheme is "*https://*", then **UseSSL** will be set to true, and a URL with "*http://*" will set **UseSSL** to false.

A port value cannot be used as part of this option, so values like "*http://couchbase-server.com:8093*" are not allowed. Please use **WebConsolePort**, **N1QLPort** and **AnalyticsPort**. (Note: You can find these three properties under the **Authentication** section of the **Advanced** tab in the **ODBC Data Source Administrator DSN Configuration** dialog)

This value can also accept multiple servers in the above format separated by commas, such as "1.2.3.4, couchbase-server.com". This will allow the driver to recover the connection in case some of the servers listed are inaccessible.

Note that while the driver will try to recover the connection as a whole, it may lose individual operations. For example, while a long-running query will fail if the server becomes inaccessible while that query is running, that query can be retried on the same connection and the driver will execute it on the next active server.

2. **Auth Scheme –** click the dropdown to select from the list the appropriate type of authentication to use when connecting to Couchbase. The options available for selection are as follows:
   - Basic – select this if your Couchbase server is set up to authenticate users with a simple username and password. This is the most common authentication method for connecting to a Couchbase cluster.

     If your Couchbase deployment does not use more advanced authentication mechanisms like TLS client certificates, LDAP, or Kerberos, then Basic is the appropriate choice.

     Also, if you are working in a development or test environment where security constraints are minimal, Basic authentication is often the easiest to configure.

However, for production environments, it's important to consider more secure authentication methods if available, such as using a credentials file or other more secure schemes.

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

- User – this is the Couchbase user account used to authenticate. The authenticating server requires both **User** and **Password** (see below) to validate the user's identity.

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

The default value is **Basic**.

- CredentialsFile – select this when using a Secure Credentials File instead of entering a username and password. If your organization prefers not to store credentials directly in the DSN (Data Source Name) configuration for security reasons, you can use a credentials file to securely store authentication details.

  If the ODBC connection is used in automated scripts or applications where interactive login is not possible, a credentials file can provide a seamless authentication method.

  Also, in environments where multiple users or applications need access to Couchbase, using a credentials file allows easier updates without modifying individual DSN settings.

  In addition, instead of embedding a username and password in connection strings (which can be a security risk), using a credentials file improves security by keeping authentication details separate.

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

    - Credentials File – use this property if you need to provide credentials for multiple users or buckets. This file takes priority over other forms of authentication.

- SSLCertificate – select this if your Couchbase server requires encrypted connections for data security, in which case you will need to specify an SSL certificate to establish a secure connection.

  The certificate helps ensure that the connection is being made to the legitimate Couchbase server and not an unauthorized or compromised one.

  If the Couchbase cluster is configured to enforce TLS (Transport Layer Security), you must provide the SSL certificate to authenticate the server and encrypt data in transit.

  If two-way SSL authentication is required (where both the client and he sever verify each other's certificates), you may need to provide both a client certificate and key.

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

    - SSL Client Cert – this is the TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).

      This property specifies the client certificate store for SSL Client Authentication. Use this property alongside **SSL Client Cert Type** (see below), which defines the type of the certificate store, and **SSL Client Cert Password** (see below), which specifies the password for password-protected stores. When SSL Client Cert is set and **SSL Client Cert Subject** (see below) is configured, the driver searches for a certificate matching the specified subject.

Certificate store designations vary by platform. On Windows, certificate stores are identified by names such as MY (personal certificates), while in Java, the certificate store is typically a file containing certificates and optional private keys.

The 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

- SSL Client Cert Type – this is a dropdown selection requiring you to specify the type of key store containing the TLS/SSL client certificate.

  The options available for selection are:

  - USER
  - MACHINE
  - PFXFILE
  - PFXBLOB
  - JKSFILE
  - JKSBLOB
  - PEMKEY_FILE
  - PEMKEY_BLOB
  - PUBLIC_KEY_FILE
  - PUBLIC_KEY_BLOB
  - SSH PUBLIC_KEY_FILE
  - SSH PUBLIC_KEY_BLOB
  - P7BFILE
  - PPKFILE
  - XMLFILE
  - XMLBLOB

  This property determines the format and location of the key store used to provide the client certificate. Supported values include platform-specific and universal key store formats.

If required, more information on the available values and their usages on each of the above key stores can be provided.

The default value is **USER**.

- SSL Client Cert Password – this is the password for the TLS/SSL client certificate.

  This property provides the password needed to open a password-protected certificate store. This property is necessary when using certificate stores that require a password for decryption, as is often recommended for PFX or JKS type stores.

  If the certificate store type does not require a password, for example USER or MACHINE on Windows, this property can be left blank. Ensure that the password matches the one associated with the specified certificate store to avoid authentication errors.

- SSL Client Cert Subject – this is the subject of the TLS/SSL client certificate.

  This property determines which client certificate to load based on its subject. The driver searches for a certificate that exactly matches the specified subject. If no exact match is found, the driver looks for certificates containing the value of the subject. If no match is found, no certificate is selected.

  The subject should follow the standard format of a comma-separated list of distinguished name fields and values. For example, CN=www.server.com, OU=Test, C=US.

Other common fields include: O = Organization; L = Locality ; S = State; E = Email address.

The default value is *

3. **Couchbase Service** – this is a dropdown selection that determines the Couchbase service to connect to. The options available for selection are as follows:
   - N1QL – this allows you to use SQL-like queries to interact with Couchbase, making it easier for users familiar with SQL to work with Couchbase. N1QL provides powerful querying capabilities, including joins, subqueries, and aggregation functions, which are not available with other Couchbase services.

     In addition, if you are connecting Couchbase to BI tools like Tableau, Power BI, or Excel, which expect SQL-like queries, selecting N1QL enables better compatibility.

   - Analytics – the Analytics service in Couchbase is optimized for complex, long-running queries on large volumes of data. It is better suited for batch processing, aggregations, and trend analysis than the N1QL Query service.

     The Analytics service runs separately from the N1QL service, so selecting it ensured that analytical workloads don’t slow down real-time application queries. Also, if you are performing heavy joins, scans, or aggregations, Analytics is a better choice than N1QL. Because the Analytics service uses a columnar storage engine designed for analytical processing, this allows faster scans over large datasets without requiring predefined indexes.

     Analytics is also useful for running ah-hoc reports, BI queries, and Machine Learning workloads.

If you need real-time, indexed queries for application performance, use the N1QL query service instead.

The default is **N1Ql.**

4. **Connection Mode** – this is a dropdown selection to determine how to connect to the Couchbase server. The options available for selection are as follows:
   - Direct – this is to establish a direct connection to a single Couchbase node instead of routing through a cluster-wide load balancer or gateway. This is useful if you want to target a specific node for queries or reduce network hops in controlled environments.

     Direct mode connects to Couchbase using the server address provided, without needing additional routing or proxy configurations.

     Direct connections bypass the cluster management overhead, potentially reducing latency for queries and as such is ideal for low-latency applications that need direct access to the Couchbase Query or Analytics service, e.g., where the Couchbase server is directly accessible.

     Note that Direct mode does not automatically failover to another node if the selected node goes down.

   - Cloud – select this if your database is hosted on **Couchbase Capella** (the managed cloud version of Couchbase). In this case, you must select Cloud to properly authenticate and connect.

     Cloud mode automatically handles TLS encryption, authentication, and networking settings specific to Capella. This ensures that your connection meets Capella's security and compliance requirements.

Unlike the Direct mode (above), which requires specifying node Ips, Cloud mode simplifies the connection process by using Capella's built-in connection settings.

For on-premises or private cloud deployments, use Direct mode instead.

The default value is **Direct**.

5. **Use SSL** – this is a True/False dropdown selection to confirm whether to negotiate TLS/SSL when connecting to the Couchbase server.

   When this is set to true, the defaults for the following options change:

Property	Plaintext Default	SSL Default
Analytics Port	8095	18095
N1QL Port	8093	18093
Web Console Port	8091	18091

   Note: You can find the three properties listed above under the **Authentication** section of the **Advanced** tab in the **ODBC Data Source Administrator DSN Configuration** dialog)

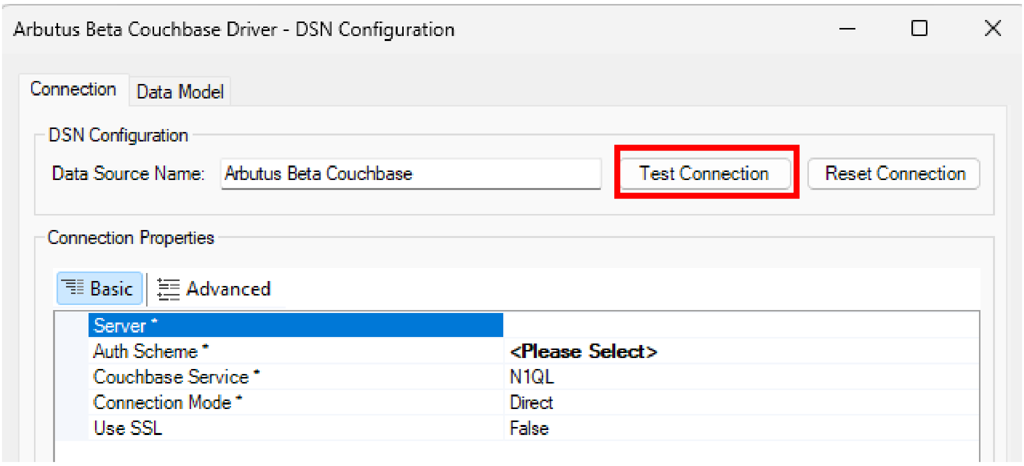
   This option should be enabled when connecting to Couchbase Capella because all Capella deployments use SSL by default.

   The default value is **False**.

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

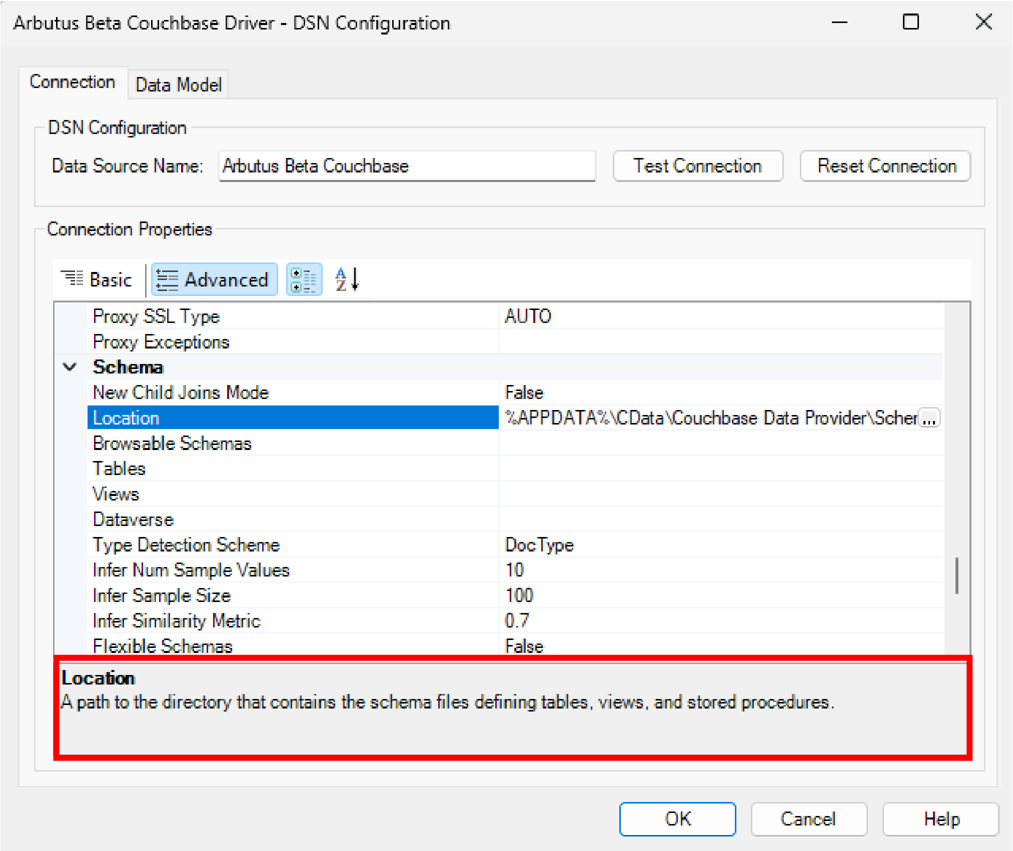


Data Source Name: Arbutus Beta Couchbase

Server *	
Auth Scheme *	<Please Select>
Couchbase Service *	N1QL
Connection Mode *	Direct
Use SSL	False

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.



Data Source Name: Arbutus Beta Couchbase

Proxy SSL Type	AUTO
Proxy Exceptions	
Schema	
New Child Joins Mode	False
Location	%APPDATA%\CData\Couchbase Data Provider\Scher ...
Browsable Schemas	
Tables	
Views	
Dataverse	
Type Detection Scheme	DocType
Infer Num Sample Values	10
Infer Sample Size	100
Infer Similarity Metric	0.7
Flexible Schemas	False

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.