


Source Connector for Databricks

This guide describes how to configure Digna to connect to Databricks using either the native Python connector or the ODBC driver.

It refers to the screen "**Create a Database Connection**".

 [DASHBOARD](#) [ANALYTICS](#) [PROJEKTE](#) [BENUTZER](#) [INSPEKTIONSZENTRUM](#) [INTEGRATIONEN](#)

[🏠](#) / [Projekte](#) / [Demo Project](#) / [Verbindung](#) / [Neu](#)

[←](#) **Create a Database Connection**

Technology *

Host Address

Host Port

Database Name

Schema Name

User Name

User Password

Profiling Mode *

Work Schema Name

Use ODBC

☐

ABBRECHEN

TEST

ERSTELLEN

Native Python Driver

Library: databricks-sql-connector

Supported Authentication: Personal Access Token (PAT) only

 For other authentication methods, please use the ODBC driver.

Personal Access Token (PAT)

To authenticate using a personal access token, refer to the official Databricks documentation:

👉 [How to obtain a PAT](#)

Digna Configuration (Native Driver)

Provide the following information in the **"Create a Database Connection"** screen:

Technology:	Databricks
Host Address:	Databricks hostname, e.g. "xxxxxxxxxxxxxxxxxxxxx.databricks.com"
Host Port:	443
Database Name:	Schema that contains the source data (same as Schema Name)
Schema Name:	Schema that contains the source data
User Name:	HTTP Path provided by Databricks, e.g. "/sql/1.0/warehouses/xxxxxxxxxxxxxxxxx"
User Password:	Personal Access Token, e.g. "dapixxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
Use ODBC:	Disabled (default)

ODBC Driver

The ODBC driver supports a broader range of authentication and connectivity options. This section focuses on token-based authentication using the **Simba Spark ODBC Driver**.

1. Install the ODBC Driver

Install the **Simba Spark ODBC Driver** by following the vendor's official installation guide.

2. Configure the ODBC Data Source

Follow these steps to configure a new ODBC data source using a Personal Access Token:

Step 1

Simba Spark ODBC Driver DSN Setup

Data Source Name: dignadata_databricks

Description:

Spark Server Type: SparkThriftServer (Spark 1.1 and later) ▾

DFI Options...

Service Discovery Mode: No Service Discovery ▾

Service Discovery Options...

Host(s): dbc-3-1-122.cloud.databricks.com

Port: 443

Database: default

Authentication

Mechanism: OAuth 2.0 ▾

Realm:

Host FQDN: _HOST

Service Name: spark

☒ Canonicalize Principal FQDN

☐ Delegate Kerberos Credentials

User Name: token

Password:

Password Options...

Delegation UID:

OAuth Options...

Thrift Transport: HTTP ▾

Proxy Options... HTTP Options... SSL Options...

Advanced Options... Logging Options...

v2.9.1.1001 (64 bit)

Test OK Cancel

Step 2

OAuth Options

Authentication Flow: Token Passthrough

Access Token:

☐ Use JWT Assertion

Client ID:

Client Secret:

OAuth Scope:

JWT Key Identifier:

JWT Private Key Path:
 Browse...

JWT Private Key Password (optional):

☐ Use OIDC Discovery Endpoint:

Azure Workspace Resource ID:

JWT Private Key Encryption Options... Client Secret Encryption Options...

☒ Ignore SQL_DRIVER_NOPROMPT

OK Cancel

Step 3

HTTP Properties

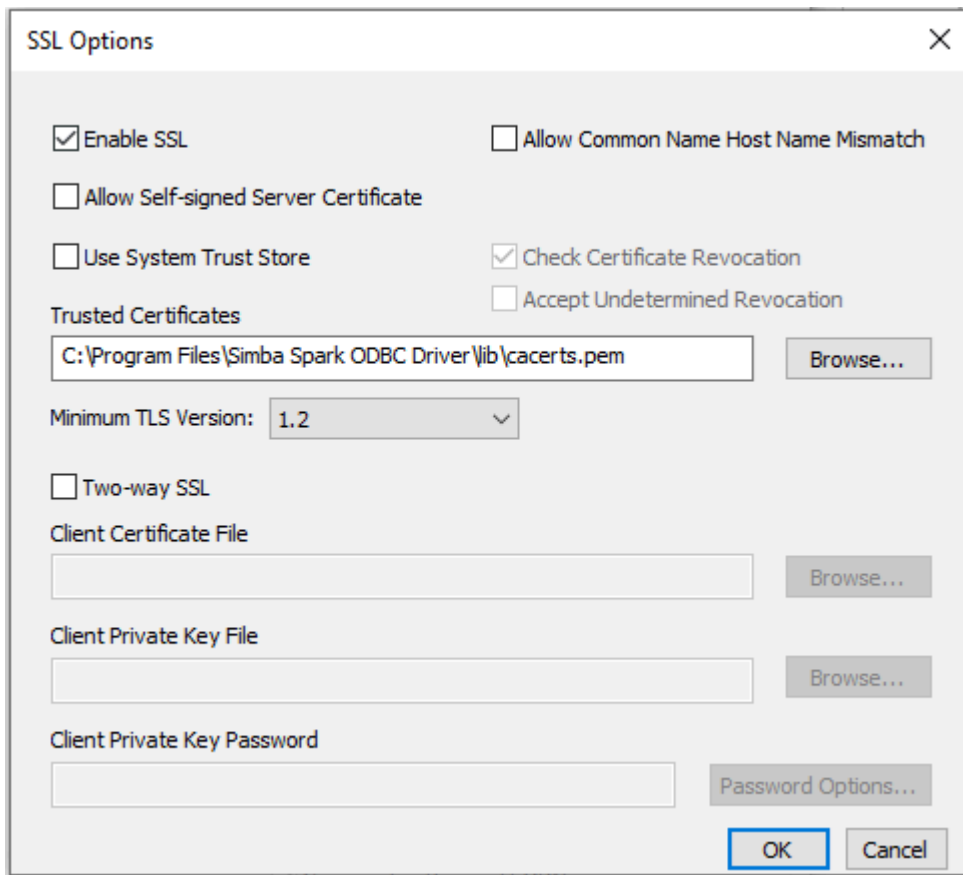
HTTP Path: /sql/1.0/warehouses/fc14...9526

Custom HTTP Headers

Add... Edit Remove

OK Cancel

Step 4

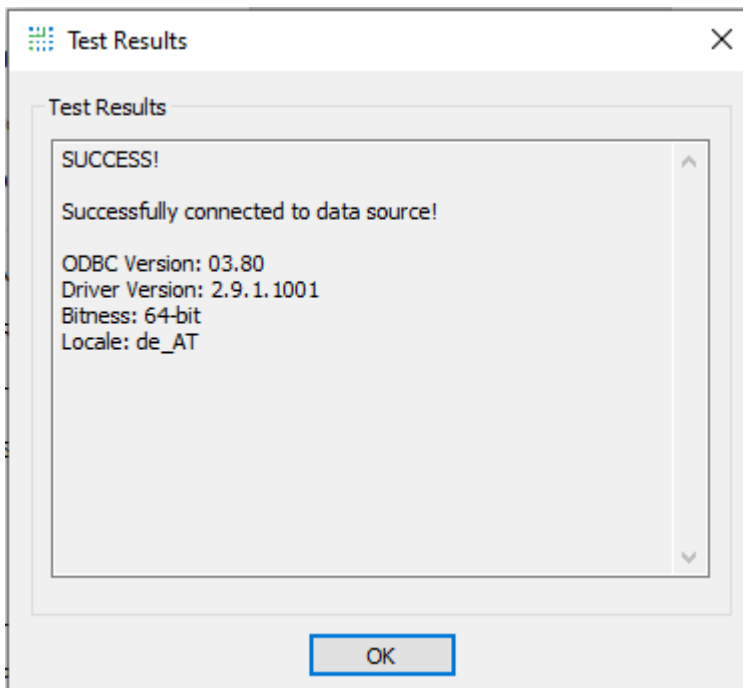


The 'SSL Options' dialog box contains the following elements:

- Checkboxes: ☒ Enable SSL, ☐ Allow Common Name Host Name Mismatch, ☐ Allow Self-signed Server Certificate, ☐ Use System Trust Store, ☒ Check Certificate Revocation, ☐ Accept Undetermined Revocation.
- Trusted Certificates: A text field containing 'C:\Program Files\Simba Spark ODBC Driver\lib\cacerts.pem' and a 'Browse...' button.
- Minimum TLS Version: A dropdown menu showing '1.2'.
- Two-way SSL: ☐ Two-way SSL.
- Client Certificate File: A text field and a 'Browse...' button.
- Client Private Key File: A text field and a 'Browse...' button.
- Client Private Key Password: A text field and a 'Password Options...' button.
- Buttons: 'OK' and 'Cancel' at the bottom right.

Step 5 – Test the connection

Click the **TEST** button. A successful connection should look like this:



The 'Test Results' dialog box displays the following information:

- Header: 'Test Results' with a close button.
- Message: 'SUCCESS!' followed by 'Successfully connected to data source!'.
- Details: 'ODBC Version: 03.80', 'Driver Version: 2.9.1.1001', 'Bitness: 64-bit', and 'Locale: de_AT'.
- Button: 'OK' at the bottom.

Now you can configure Digna to use the ODBC connection, either with a **DSN (Data Source Name)** or a **DSN-less** setup.

A. DSN-Based Configuration

Digna Configuration

In the **"Create a Database Connection"** screen, provide the following:

```
Technology:      Databricks
Database Name:   Schema that contains the source data (same as Schema Name)
Schema Name:     Schema that contains the source data
Use ODBC:        Enabled
```

ODBC Properties

```
name: "DSN",    value: "dignadata_databricks"
```

- ◆ The `DSN` must match the name defined in your ODBC driver configuration.

B. DSN-less Configuration

Digna Configuration

In the **"Create a Database Connection"** screen, provide the following:

```
Technology:      Databricks
Database Name:   Schema that contains the source data (same as Schema Name)
Schema Name:     Schema that contains the source data
Use ODBC:        Enabled
```

ODBC Properties

```
name = "Driver",      value = "{Simba Spark ODBC Driver}"
name = "Host",        value = "xxxxxxxxxxxxxxxxxxxxx.databricks.com"
name = "Port",        value = "443"
name = "HTTPPath",    value = "/sql/1.0/warehouses/xxxxxxxxxxxxxxxxxx"
name = "SSL",          value = "1"
name = "ThriftTransport", value = "2"
name = "AuthMech",    value = "3"
name = "UID",          value = "token"
name = "PWD",          value = "dapixxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```