



Connecting Denodo to Azure Cosmos DB

Revision 20200803

NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2020
Denodo Technologies Proprietary and Confidential

Goal

Azure Cosmos DB is Microsoft's globally distributed multi-model database service. You can use Azure Cosmos DB to quickly create and query key/value databases, document databases, and graph databases, all of which benefit from the global distribution and horizontal scale capabilities at the core of Azure Cosmos DB.

This document describes how to connect Denodo with an Azure Cosmos DB SQL API account.

Prerequisites

The recommended way to connect Denodo with Azure Cosmos DB SQL API account is using Denodo's ODBC connectivity capabilities

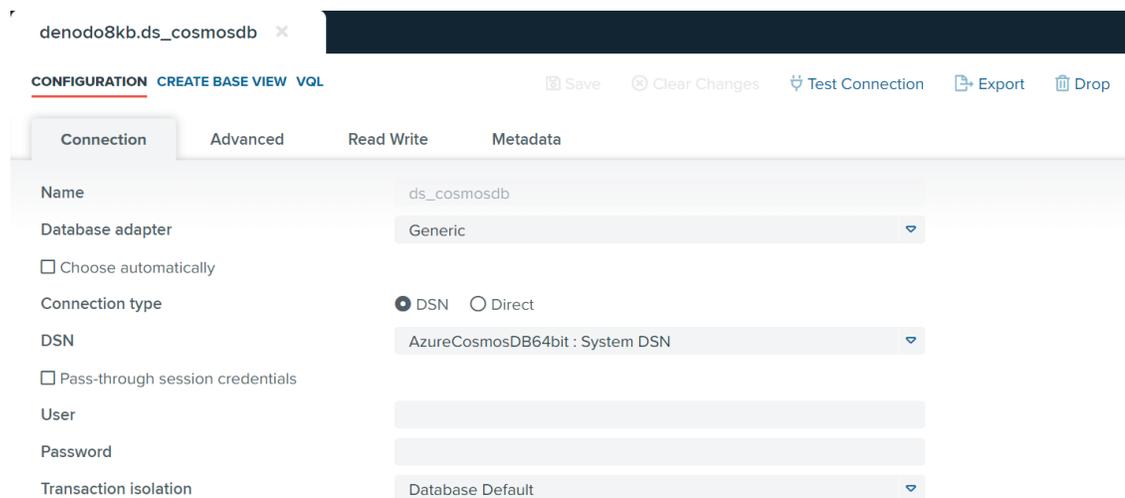
1. Download and install the Microsoft Azure Cosmos DB ODBC Driver. Starting from Denodo 8, the Denodo platform can be installed only on a 64-bit platform, so download the "Microsoft Azure Cosmos DB ODBC 64-bit.msi" driver. The ODBC driver can be downloaded from <https://docs.microsoft.com/en-gb/azure/cosmos-db/odbc-driver>.
2. Create a new 64 bit DSN using the installed Microsoft Azure Cosmos DB ODBC driver. We need to use the data in the "Settings -> Keys" section of the Azure Portal.
 - a. Specify host with this format: `https://XXXXX.documents.azure.com:443/` The CosmosDB account appears in the URI.
 - b. For the Access Key we must use the CosmosDB account PRIMARY KEY. We should use the Read-write keys.

Note: For versions prior to Denodo 8 If there are any issues while using a 64 bit ODBC driver, we recommend to follow the below steps:.

1. For 64-bit Windows, download the "Microsoft Azure Cosmos DB ODBC 32x64-bit.msi for 32-bit on 64-bit Windows" driver.
2. For 32-bit Windows, download the "Microsoft Azure Cosmos DB ODBC 32-bit.msi for 32-bit Windows" driver.
3. Create a new 32 bit DSN using the installed Microsoft Azure Cosmos DB ODBC driver. Use the data in the "Settings -> Keys" section of the Azure Portal.
 - a. Specify host with this format: `https://XXXXX.documents.azure.com:443/` The CosmosDB account appears as URI.
 - b. For the Access Key we must use the CosmosDB account PRIMARY KEY. We should use the Read-write keys.
4. Install a 32 bit Denodo platform. If the current Denodo installation is for 64 bits, the recommended setup is using an additional Denodo 32 bit installation as a proxy. Note that this might require additional Denodo licenses. In the latter case it would be necessary to:
 - a. Import the Cosmos DB desired tables into a 32 bit Denodo platform.
 - b. Import the 32 bit Denodo platform as a new data source into the 64 bit Denodo platform. The Virtual DataPort adapter will be used.
 - c. Import the desired tables from the 32 bit Denodo platform into the 64 bit Denodo platform.
5. Make sure there is not a file called `<DENODO_HOME>/lib/contrib/rt17.jar` in the Denodo installation. If it exists, it must be deleted.

Connectivity

1. From the Denodo Platform Control Center, append the following property to the JVM options of the Virtual DataPort Server:
-XX:CompileCommand=exclude,sun/jdbc/odbc/*.*
2. From the design studio, create a new ODBC data source by navigating to "File > New > Data source > ODBC". Select the previously created CosmosDB DSN from the DSN drop down.
3. Click Test connection to test the datasource connection and click save button to save the datasource.



4. There is an additional step after the ODBC datasource is created. Edit the VQL of the the data source to set "cosmosdb" as DATABASENAME. To make this change, copy the VQL from the VQL tab and modify the DATABASENAME to 'cosmosdb'. The modified VQL should be something like this:

```
CREATE OR REPLACE DATASOURCE ODBC ds_cosmosdb
  DSN = 'cosmos'
  USERNAME = ''
  USERPASSWORD = ''
  CLASSPATH = 'jdbc-odbc'
  DATABASENAME = 'cosmosdb'
  DATABASEVERSION = '00.00.0000'
  FETCHSIZE = 1000
  VALIDATIONQUERY = ''
  INITIALSIZE = 4
  MAXIDLE = -1
  MINIDLE = 0
  MAXACTIVE = 0
  EXHAUSTEDACTION = 1
  TESTONBORROW = true
  TESTONRETURN = false
  TESTWHILEIDLE = false
  TIMEBETWEENEVICTION = -1
  NUMTESTPEREVICTION = 3
  MINEVICTABLETIME = 1800000
```

```
POOLPREPAREDSTATEMENTS = false  
MAXOPENPREPAREDSTATEMENTS = -1;
```

Execute the modified VQL in the VQL shell and refresh the design studio by navigating to File > Refresh.

5. Disable the Datasource Connection Pool. For this we set the “Maximum number of active connections” to 0. This is recommended because the Azure Cosmos DB ODBC driver caches metadata about the existing tables at the connection level. This means that the tables added or modified after a connection has been opened, will not be visible to the clients using the connection. Disabling the connection pool ensures Denodo does not reuse connections so these problems do not appear.
6. Now we can start creating base views over this CosmosDB data source.

References

[ODBC Sources](#)