ODBC Connection

Use it to connect to an ODBC data source.

Connection parameters

To configure the connection, set the following parameters:

- **Caption**: A user-defined name for the connection.
- **Connection string**: A string in the format `DSN | ODBC connection string`. You can select one of the configured data sources in the system or enter a connection string in the format of the installed ODBC driver.

> **Note:** In Megaladata Desktop, a File DSN (a file with connection settings) can be specified.

To display the list of available ODBC sources, click the ... **More** button.

- **Test**: Use this button to test the specified connection settings.

> **Note:** If the **Login prompt** option is active (see below), a login/password will be requested when testing the connection.

- **Username**: The database user's login.
- **Password**: The database user's password.
- **Login prompt**: If this box is checked, user password will not be saved. Instead, the user will be prompted to enter it the first time they run a Database import or export node. The password is then remembered for the rest of their session and they will only need to enter it again after the connection node is deactivated. Please be aware that the password prompt will time out after 60 seconds.

> **Note**: When working in *batch mode*, running a package from the **Scheduler**, or launching it via the **Megaladata Integrator**, even if login prompt is enabled, the system will try to connect **without a password**. (The logs will indicate that the password wasn't specified.)

- **Show system tables**: When this checkbox is selected, the system tables of the database accessible to the user become visible in the wizard of an import node that uses this connection. (Enable this option when connecting to an Excel file).
- **Do not use DBMS client**: This option is not available for ODBC connections.

- **Quote names**: If database object names (e.g., table or field names) contain spaces or reserved characters, enable this checkbox to use quotes marking the beginning and end of the name.
- **Clear pool when deactivated**: When enabled, clears the pool of connections associated with the connection node immediately upon the node's deactivation. When disabled, the connections are deactivated and removed from the pool individually.
- **Connection timeout (s)**: The default timeout is 20 seconds. When performing database import or export in error-ignore mode, if the connection times out, the error will be logged into the **Execution status** output port of the **Database** import/export node. In regular mode, the node will return an error. This setting can be enabled only if supported by the driver.
- **Comment**: Use this form to leave any reference information about the connection.

Configuring ODBC data sources for Windows

Data sources are configured in the Windows ODBC Data Source Administrator. This is also where you can find information about the drivers installed on the system and the format of the connection string.

It is important to note that data sources defined in the **User DSN** section of the ODBC Data Source Administrator are only available to the users who created them. Therefore, these sources may not be accessible to the user under which the Megaladata application/service is running. We recommend to create ODBC data sources in the **System DSN** section, as they are available to all users.

▼ List of available drivers for Windows

> The connection requires the installation of an ODBC driver with the same bitness as the Megaladata application/server. In the operating system, one of the ODBC data sources is the 64-bit version located at `C:\Windows\System32\odbcad32.exe`.
>
> 64-bit:
>
> - SQL Server
>
> List of ODBC drivers for which a connection string can be used:
>
> - Various drivers for SQL Server
> - Microsoft Text Driver
> - Firebird/InterBase driver
> - Microsoft Excel Driver
> - Microsoft Access Driver
> - Microsoft dBASE Driver
> - Drivers for Oracle
> - SQLite3 ODBC Driver
> - Drivers for MySQL
> - Microsoft Paradox Driver
> - Microsoft Visual FoxPro Driver
> - Drivers for PostgreSQL

> - Drivers for Sybase/Adaptive Server Enterprise
> - Drivers for Apache Hive
> - CData Drivers for Excel
> - CData Drivers for Access
> - Drivers for SAP HANA
>
> If a driver is not on this list, you can connect it only by selecting an already configured data source.

Configuring ODBC data sources for Linux

In unixODBC for Linux, data sources are configured using configuration files. The `/.odbc.ini` file is used for user DSNs, and `/etc/odbc.ini` is used for system DSNs. Each database management system (DBMS) driver has its own set of connection parameters. Detailed information about these parameters can be found in the documentation for the respective DBMS.

In unixODBC, ODBC drivers are specified in the `odbcinst.ini` file, which is usually located in the `/etc/` directory. The location of the file may vary in different Linux distributions. For example, in Ubuntu, the file is located at `/etc/odbcinst.ini`.

To find the file, you can use the command `sudo find / -name odbcinst.ini`.

The list of drivers for unixODBC may vary depending on the operating system and available packages.

Installing a driver with Docker Compose

When creating an image, you need to place the driver for the necessary DBMS in the `server/libs` folder. To do this, add a `RUN` directive that will install this driver. For example, for the Ubuntu operating system: `RUN apt install -y /app/<package_name>.deb`.

The DB configuration should be written in the `odbcinst.ini` and `odbc.ini` files.

There are two ways to pass the configuration into the container:

- During the image build. The driver is placed in the image using the `COPY` directive. It is already present in the Dockerfile and copies all necessary Megaladata libraries.
- When launching the container in Docker Compose on the client's host system, a `volumes` directive is added to ensure data persistence and the ability to update them without rebuilding the image. The `volumes` directive mounts files or directories from the host machine into the container, allowing them to be modified directly on the host, with changes immediately reflected inside the container.

See also:

- Wikipedia page
- Connection string principles and examples
- Examples of connection strings for various data sources

Read on: Oracle Connection