SQLite Connection

Use it to connect to an SQLite database.

This connection also implements the following special features when working with database import:

- Along with the standard character comparison rules (collation) BINARY, RTRIM, and NOCASE, the following non-standard options can be used:
  - **UNICODE**: Language-independent, case-sensitive comparison of Unicode characters.
  - **LOCALIZED**: Language-dependent, case-sensitive comparison of Unicode characters. The result of string comparison depends on the locale of the Megaladata server's operating system.
- In SQL constructs, it is possible to use the `iLIKE` function, which implements the functionality of `LIKE` in a case-insensitive mode for Unicode characters.
- Foreign key check is enabled by default.

Connection parameters

To configure the connection, set the following parameters:

- **Caption**: A user-defined name of the connection.
- **In-memory database**: When this checkbox is selected, a connection to an in-memory DB with a shared cache will be created. Disabled by default.
- **File name**: Specify the path to the DB file containing the necessary data. You can enter the parameters manually or click the **...** **More** button to bring up the file selection window.

If the **In-memory database** checkbox is selected, a DB identifier must be specified. It can include:

```
* Latin alphabet characters
* Digits
* Slashes
* Underscores
* Periods
```

- **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**: Disabled; not used when connecting to SQLite.
- **Password**: The password set by the DB user to protect the data.
- **Login prompt**: When this checkbox is marked, the password entered in the setup wizard will not be saved. It will be requested once upon the first activation of Database import or export nodes and will not be requested again until the connection node is deactivated. The login/password entry dialog box has a timeout of 60 seconds. This option is available if the **Encrypted DB** checkbox (below) is selected.

> **Note:** When working in batch mode or launching the package via the Scheduler or the Megaladata Integrator, even if the **Login prompt** is enabled, the platform will attempt to connect **without password**. (The logs will indicate that the password was not used.)

- **Encrypted DB**: Select this checkbox to enable DB encryption mode. This mode uses the AES256 encryption algorithm and requires a password.
- **Cache size**: A field to set the cache size. The effect of this parameter is similar to executing the DB command `"PRAGMA schema.cache_size = pages | -kibibytes;"`, where a positive number specifies the amount of memory in pages, and a negative number allocates memory in KiB.
- **Sync mode**: This setting controls how the database writes data to the disk, creating a trade-off between performance and data safety. "Off" is the fastest option that offers higher performance but comes with a greater risk of data corruption if the system crashes. In contrast, safer modes like "Full" and "Extra" are slower because they ensure data is securely written, providing the most protection against data loss. This setting is equivalent to the PRAGMA synchronous database command `"PRAGMA schema.synchronous = 0 | OFF | 1 | NORMAL | 2 | FULL | 3 | EXTRA;"`. The following modes are available:
  - **Default** (uses the **full** mode)
  - **Off**
  - **Normal**
  - **Full**
  - **Extra**
- **Transaction journal mode**: Sets the operating mode of the transaction journal. The effect of this parameter is similar to executing the DB command `"PRAGMA schema.journal_mode = DELETE | TRUNCATE | PERSIST | MEMORY | WAL;"`. The following options are available:
  - **Default** (uses the **Delete** mode)
  - **Delete** (delete file)
  - **Truncate** (truncate content)
  - **Persist** (zero out header)
  - **Memory** (store in memory)
  - **Write-ahead log** (WAL)
- **TEMP store**: Determines the storage location for temporary files. The effect of this parameter is similar to executing the DB command `"PRAGMA temp_store = 0 | DEFAULT | 1 | FILE | 2 | MEMORY;"`. The following options are available:
  - **Default**: For an unencrypted DB, it is **File system**; for an encrypted DB, it is **Memory**.
  - **File system**.
  - **Memory**.

- **NOCASE in UNICODE mode**: When checked, this option makes NOCASE text comparisons follow the Unicode standards. This provides a language-independent, case-insensitive method of matching characters.
- **LIKE mode**: You can choose one of the character comparison options for the LIKE function:
  - **Case-insensitive for ASCII characters**
  - **Case-sensitive**
  - **Case-insensitive**
- **Create DB if it does not exist**: When this checkbox is selected, the database specified in the **File name** field will be created if it does not exist. The new DB is created in UTF-16 encoding.
- **Lower, upper for UNICODE**: This checkbox extends the action of the `lower` and `upper` functions to all UNICODE characters. Without this option, the `lower` and `upper` functions only work for Latin alphabet letters.
- **Show system tables**: When this checkbox is selected, the DB system tables available to the user become visible in the wizard of the import node that uses this connection.
- **Do not use DBMS client**: This parameter is not available for SQLite.
- **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 this checkbox is selected, the connection pool associated with the **SQLite** connection node is cleared immediately upon node deactivation. When the checkbox is unchecked, connections begin to gradually deactivate and are removed from the connection pool after the node itself is deactivated.
- **Comment**: Use this form for any reference information about the connection.

> **Note**: Connecting to an SQLite database employs `sqlite3.dll` and `sqlcipher.dll` libraries of the same bitness as the Megaladata application/server. These libraries are supplied as part of the Megaladata distribution. When using third-party libraries, the default values for the **Sync mode** and **Transaction journal mode** parameters may be overridden.

Compatibility

SQLite 3.0 and higher

See also:

- Official SQLite website
- SQL syntax documentation for SQLite
- PRAGMA information

Read on: Connecting to Kafka