

[SQL Server] - Failed with The driver could not establish a secure connection to SQL Server by using Secure Sockets Layer (SSL) encryption

Last updated by | Xiaojin Wang | Aug 29, 2022 at 7:33 PM PDT

Contents


- [Issue](#)
- [Root Cause](#)
- [Resolution](#)
- [Additional Information:](#)

Issue

When customer uses SQL Server connector as source and Sink in dataflow (preview data, debug/trigger run...). He may find job will fail with following error message:

Cannot connect to SQL database: '', Error Message: The driver could not establish a secure connection to SQ

Root Cause

The SQL Server JDBC driver has introduced a breaking change in v 10.2.0. Please suggest customer to check below resolutions. see link: <https://techcommunity.microsoft.com/t5/sql-server-blog/jdbc-driver-10-2-for-sql-server-released/ba-p/3100754> 

Resolution

There are a couple breaking changes in 10.2 over previous releases that may affect a lot of users. Similar to the HTTP to HTTPS default changes made in web browsers a few years back (and the security reasons for them), we are changing the default value of the `Encrypt` connection option from `false` to `true`. With the increased emphasis on secure-by-default, the growing use of cloud databases, and the need to ensure connections are secure, it's time for this backwards-compatibility-breaking change. We realize this will cause some disruption, but letting clients try to connect without encryption by default leaves them open to attack from malicious actors.

We also changed the behavior of `TrustServerCertificate` to not be tied to the `Encrypt` setting. Previously, if `Encrypt` was set to `false`, certificates wouldn't be validated regardless of what `TrustServerCertificate` was set to. This allowed servers using self-signed certificates and Force Protocol Encryption to encrypt their client connections without requiring clients to change their default settings.

The action item if you are affected by the `Encrypt` change is to either (in order of recommendation):

- Install a trusted certificate on your server.
- Change your client's Encrypt connection string setting (or data source property) to false.

If you are using a self-signed certificate and the Force Encryption setting on the server to ensure clients connect with encryption, you will need to do one of the following (in order of recommendation):

- Change to a certificate that is trusted as part of the client's trust chain.
- Add the self-signed certificate as a trusted certificate on the client.
- Change your client's TrustServerCertificate connection string setting (or data source property) to true.

Additional Information:

- Icm Reference: N/A
- Author: Xiaojin Wang
- Reviewer: Shawn Xiao
- Keywords: SQL Server

How good have you found this content?

