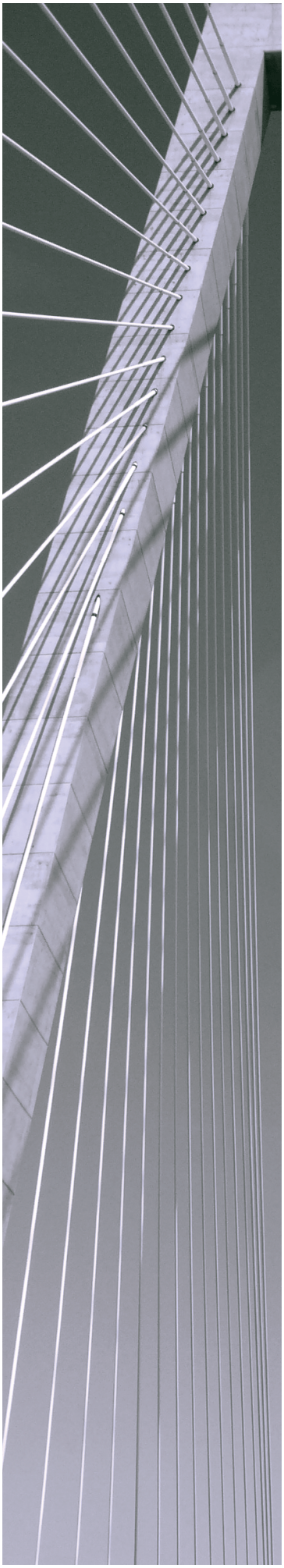

Simba Presto JDBC Driver with SQL Connector

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.1.9

October 2, 2020

Copyright © 2020 Magnitude Software, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Magnitude.

The information in this document is subject to change without notice. Magnitude strives to keep this information accurate but does not warrant that this document is error-free.

Any Magnitude product described herein is licensed exclusively subject to the conditions set forth in your Magnitude license agreement.

Simba, the Simba logo, SimbaEngine, and Simba Technologies are registered trademarks of Simba Technologies Inc. in Canada, the United States and/or other countries. All other trademarks and/or servicemarks are the property of their respective owners.

All other company and product names mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective owners.

Information about the third-party products is contained in a third-party-licenses.txt file that is packaged with the software.

Contact Us

Simba Technologies Inc.
938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel: +1 (604) 633-0008

Fax: +1 (604) 633-0004

www.simba.com

About This Guide

Purpose

The *Simba Presto JDBC Driver with SQL Connector Installation and Configuration Guide* explains how to install and configure the Simba Presto JDBC Driver with SQL Connector on all supported platforms. The guide also provides details related to features of the driver.

Audience

The guide is intended for end users of the Simba Presto JDBC Driver.

Knowledge Prerequisites

To use the Simba Presto JDBC Driver, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba Presto JDBC Driver
- Ability to use the data store to which the Simba Presto JDBC Driver is connecting
- An understanding of the role of JDBC technologies in connecting to a data store
- Experience creating and configuring JDBC connections
- Exposure to SQL

Document Conventions

Italics are used when referring to book and document titles.

Bold is used in procedures for graphical user interface elements that a user clicks and text that a user types.

`Monospace font` indicates commands, source code or contents of text files.

> **Note:**
>
> A text box with a pencil icon indicates a short note appended to a paragraph.

> **Important:**
>
> A text box with an exclamation mark indicates an important comment related to the preceding paragraph.

Table of Contents

About the Simba Presto JDBC Driver 7
About Presto 7
About the Driver 7
System Requirements 8
Simba Presto JDBC Driver Files 9
Installing and Using the Simba Presto JDBC Driver 10
Referencing the JDBC Driver Libraries 10
Registering the Driver Class 11
Building the Connection URL 12
Configuring Authentication 13
Using Kerberos Authentication 13
Using LDAP Authentication 18
Using Password File Authentication 18
Configuring SSL Connections 21
Configuring Proxy Connections 23
Configuring Virtual Private Cloud (VPC) Services 24
Configuring Logging 26
Features 28
Catalog and Schema Support 28
Parameters 28
Supported Connectors 28
Resource Groups 29
Data Types 29
Complex SQL Data Types 31
Additional SQL Data Types 32
Security and Authentication 34
Driver Configuration Options 35
AllowHostNameCNMismatch 35
AllowMetadataFromMultipleCatalogs 35
AllowSelfSignedServerCert 36
APPLICATIONNAME 36

ApplicationNamePrefix 37
AuthenticationType 37
Catalog 38
ClientInfo 38
CLIENTTAGS 38
ConnectionTest 39
CredDelegation 39
DbmsName 40
DisableKerberosReverseDNS 40
DnsResolver 41
DnsResolverArg 41
ExtraCredentials 41
JAASAuthType 42
KerberosPrincipal 42
KrbCacheFilePath 43
KrbConfigFilePath 43
KrbJaasFilePath 44
KrbKeytabFilePath 44
KrbServiceName 44
LogLevel 45
LogPath 46
PWD or Password 46
ProxyAuth 46
ProxyHost 47
ProxyPort 47
ProxyPWD 47
ProxyType 48
ProxyUID 48
Schema 48
SessionProperties 49
ServerVersion 49
SocketFactory 49
SocketFactoryArg 50
SSL 50
SSLKeyStorePath 51
SSLKeyStorePwd 51
SSLKeyStoreType 51
SSLTrustStorePath 51

SSLTrustStorePwd 52
SSLTrustStoreType 52
TimeZoneID 52
UID or User 53
UseProxy 53
enableKerberos (deprecated) 53
Third-Party Trademarks 55

About the Simba Presto JDBC Driver

About Presto

Presto is a low latency distributed query engine capable of querying large datasets from multiple data sources using SQL.

The data sources that Presto supports include MySQL and PostgreSQL. Presto also integrates with the Hive metastore seamlessly to complement existing Hive environments with low latency queries. Presto can query self-describing data as well as complex or multi-structured data that is commonly seen in big data systems.

Note:

For information about connecting Presto to data sources, see the Presto documentation: https://prestodb.io/docs/current/.

About the Driver

The Simba Presto JDBC Driver lets organizations connect their BI tools to Presto. Presto provides an ANSI SQL query layer and also exposes the metadata information through an ANSI SQL standard metadata database called INFORMATION_SCHEMA. The Simba Presto JDBC Driver leverages INFORMATION_SCHEMA to expose Presto's metadata to BI tools as needed.

The Simba Presto JDBC Driver complies with the JDBC 4.1 and 4.2 data standards. JDBC is one of the most established and widely supported APIs for connecting to and working with databases. At the heart of the technology is the JDBC driver, which connects an application to the database. For more information about JDBC, see *Data Access Standards* on the Simba Technologies website: https://www.simba.com/resources/data-access-standards-glossary.

This guide is suitable for users who want to access data residing within Presto from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via JDBC.

System Requirements

Each machine where you use the Simba Presto JDBC Driver must have Java Runtime Environment (JRE) 7.0 or 8.0 or 11.0 installed. If you are using the driver with JDBC API version 4.2, then you must use JRE 8.0.

The Simba Presto JDBC Driver supports the following versions of Presto Server:

- Open-source Presto Server versions 0.240 and earlier
- Teradata Presto Server versions 0.208-t and earlier
- Presto Foundation Server 341 and earlier

Simba Presto JDBC Driver Files

The Simba Presto JDBC Driver is delivered in the ZIP archive `SimbaPrestoJDBC_[Version].zip`, where *[Version]* is the version number of the driver.

This archive contains the fat JARs for all of the JDBC API versions that are supported by the driver: JDBC 4.1 and 4.2. Each JAR contains all of the required third-party libraries and dependencies for the driver.

Installing and Using the Simba Presto JDBC Driver

To install the Simba Presto JDBC Driver on your machine, extract the appropriate JAR file from the ZIP archive to the directory of your choice.

! Important:

If you received a license file through email, then you must copy the file into the same directory as the driver JAR file before you can use the Simba Presto JDBC Driver.

To access a Presto data store using the Simba Presto JDBC Driver, you need to configure the following:

- The list of driver library files (see Referencing the JDBC Driver Libraries on page 10)
- The `Driver` or `DataSource` class (see Registering the Driver Class on page 11)
- The connection URL for the driver (see Building the Connection URL on page 12)

Referencing the JDBC Driver Libraries

Before you use the Simba Presto JDBC Driver, the JDBC application or Java code that you are using to connect to your data must be able to access the driver JAR file. In the application or code, specify the appropriate fat JAR file for the JDBC version that you are using.

Using the Driver in a JDBC Application

Most JDBC applications provide a set of configuration options for adding a list of driver library files. Use the provided options to include the appropriate fat JAR file from the ZIP archive as part of the driver configuration in the application. For more information, see the documentation for your JDBC application.

Using the Driver in Java Code

You must include all the driver library files in the class path. This is the path that the Java Runtime Environment searches for classes and other resource files. For more information, see "Setting the Class Path" in the appropriate Java SE Documentation.

For Java SE 7:

- For Windows: http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris: http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html

For Java SE 8:

- For Windows: http://docs.oracle.com/javase/8/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris: http://docs.oracle.com/javase/8/docs/technotes/tools/solaris/classpath.html

Registering the Driver Class

Before connecting to your data, you must register the appropriate class for your application.

The following classes are used to connect the Simba Presto JDBC Driver to Presto data stores:

- The `Driver` classes extend `java.sql.Driver`.
- The `DataSource` classes extend `javax.sql.DataSource` and `javax.sql.ConnectionPoolDataSource`.

The driver supports the following fully-qualified class names (FQCNs) that are independent of the JDBC version:

- `com.simba.presto.jdbc.Driver`
- `com.simba.presto.jdbc.DataSource`

The following sample code shows how to use the `DriverManager` class to establish a connection for JDBC 4.2:

```
private static Connection connectViaDM() throws Exception
{
   Connection connection = null;
   connection = DriverManager.getConnection(CONNECTION_URL);
   return connection;
}
```

The following sample code shows how to use the `DataSource` class to establish a connection:

```
private static Connection connectViaDS() throws Exception
{
   Connection connection = null;
   DataSource ds = new com.simba.presto.jdbc.DataSource();
   ds.setURL(CONNECTION_URL);
   connection = ds.getConnection();
   return connection;
}
```

Building the Connection URL

Use the connection URL to supply connection information to the data store that you are accessing. The following is the format of the connection URL for the Simba Presto JDBC Driver, where *[Host]* is the DNS or IP address of the server and *[Port]* is the number of the TCP port to connect to:

```
jdbc:presto://[Host]:[Port];
```

You can specify additional settings such as any of the connection properties supported by the driver. For a list of configuration options, see Driver Configuration Options on page 35.

The example file paths use primarily Windows examples in which backslashes are prefaced with the appropriate Java escape character, for example: `c:\\temp`. Linux and macOS users should replace these with Unix-style paths, for example: `/tmp`.

The following is the format of a connection URL that specifies some optional settings:

```
jdbc:presto://[Host]:[Port];[Property1]=[Value];
```

For example, to connect to a Presto server using the user name "simba", you would use the following connection URL:

```
jdbc:presto://192.168.203.141:8080;User=simba;
```

! Important:

- Properties are case-sensitive.
- Do not duplicate properties in the connection URL.

Configuring Authentication

Some Presto data stores require authentication. You can configure the Simba Presto JDBC Driver to provide your credentials and authenticate the connection to the database using one of the following methods:

- Using Kerberos Authentication on page 13
- Using LDAP Authentication on page 18
- Using Password File Authentication on page 18

Using Kerberos Authentication

You can configure the driver to use the Kerberos protocol to authenticate the connection. Kerberos is provided as part of the Java Runtime Environment (JRE).

By default, when you use Kerberos authentication, the driver loads the credentials from the Kerberos credential cache. Alternatively, you can provide your Kerberos credentials to the driver, either in a JAAS configuration file or in the connection URL. For detailed instructions, see the topics below:

- Prerequisites on page 13
- Using a Kerberos Credentials Cache on page 14
- Using a JAAS Login Configuration File on page 15
- Using Kerberos Credentials in a Connection URL on page 16

Note:

If Kerberos authentication is enabled, then SSL is automatically enabled. For more information, see Configuring SSL Connections on page 21.

Prerequisites

Before you can use Kerberos authentication with the Simba Presto JDBC Driver, you must do the following:

1. On your Presto server, in the `/etc/presto/config.properties` file, set the following properties:

```
http.server.authentication.krb5.service-name=HTTP
http.server.authentication.krb5.keytab=HTTP.keytab
```

2. On your client machine, in the `java.policy` file for your Java environment, include the following line:

```
permission java.util.PropertyPermission
"javax.security.auth.useSubjectCredsOnly", "write";
```

3. On your client machine, in your Java environment, install the appropriate Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

Using a Kerberos Credentials Cache

By default, when you use Kerberos authentication, the driver loads the credentials from the Kerberos credential cache.

- On Windows, if the Windows machine has been joined to the appropriate Active Directory domain and the domain user has been granted access to the Presto service, then when you log in to the Windows machine it automatically caches your Kerberos credentials.
- Otherwise, a Kerberos ticket must be generated before you run the driver. To generate a Kerberos ticket, run the `kinit` Kerberos command with the appropriate principal.

For instructions on configuring Kerberos authentication by specifying your Kerberos credentials, see Using a JAAS Login Configuration File on page 15 or Using Kerberos Credentials in a Connection URL on page 16.

To configure Kerberos authentication using a Kerberos Credentials Cache:

1. Run the `kinit` command using the following syntax, where *`[Keytab]`* is the Kerberos credential and *`[Principal]`* is the Kerberos user principal to use for authentication:

   ```
   kinit -kt [Keytab][Principal]
   ```

2. Connect to the Presto server using a connection URL written in the following format:

   ```
   jdbc:presto://[Host]:[Port];AuthenticationType=Kerberos
   Authentication;SSLTrustStorePath=
   [TrustStoreFilePath];SSLTrustStorePwd=
   [TrustStorePassword]
   ```

   The variables are defined as follows:

   - *[Host]* is the DNS or IP address of the server.
   - *[Port]* is the number of the TCP port to connect to.
   - *[TrustStoreFilePath]* is the full path and file name of the Java TrustStore containing the SSL certificates to use during authentication.
   - *[TrustStorePassword]* is the password for accessing the defined Java Truststore.

For example:

```
jdbc:presto://presto-demo-
cdh:7778;AuthenticationType=Kerberos Authentication;
SSLTrustStorePath=C:\\Program Files\\Java\\jre1.8.0_
92\\lib\\security\\cacerts;SSLTrustStorePwd=changeit
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using a JAAS Login Configuration File

You can provide your Kerberos credentials in a JAAS login configuration file.

The file must specify `doNotPrompt=true`. It also must include either a keytab file and principal name, or a credential path.

To configure Kerberos authentication using a JAAS Login configuration file:

1. Create a JAAS login configuration file that includes either a keytab file and principal name, or a credential path.

   For example, the configuration file below includes a keytab file and principal name:

   ```
   Client {
   com.sun.security.auth.module.Krb5LoginModule required
   useKeyTab=true
   keyTab="C:\kerberos\keytab.krb"
   principal="host@REALM"
   doNotPrompt=true;
   };
   ```

   As another example, the configuration file below includes a credential path:

   ```
   Client {
   com.sun.security.auth.module.Krb5LoginModule required
   useTicketCache=true
   ticketCache="C:\Kerberos\ticketcache"
   doNotPrompt=true;
   };
   ```

2. Connect to the Presto server using a connection URL written in the following format:

```
jdbc:presto://[Host]:[Port];AuthenticationType=Kerberos
Authentication;SSLTrustStorePath=
[TrustStoreFilePath];SSLTrustStorePwd=
[TrustStorePassword];JAASAuthType=true;KrbJaasFilePath=
[JaasFilePath];KrbConfigFilePath=[KrbFilePath]
```

The variables are defined as follows:

- *[Host]* is the DNS or IP address of the server.
- *[Port]* is the number of the TCP port to connect to.
- *[TrustStoreFilePath]* is the full path and file name of the Java TrustStore containing the SSL certificates to use during authentication.
- *[TrustStorePassword]* is the password for accessing the defined Java Truststore.
- *[JaasFilePath]* is the full path and file name of the JAAS login configuration file.
- *[KrbFilePath]* is the full path and file name of the Kerberos `krb5.ini` configuration file.

For example:

```
jdbc:presto://presto-demo-
cdh:7778;AuthenticationType=Kerberos Authentication;
SSLTrustStorePath=C:\\Program Files\\Java\\jre1.8.0_
92\\lib\\security\\cacerts;SSLTrustStorePwd=changeit;JAA
SAuthType=true;KrbJaasFilePath=C:\\Users\\employee.DRIVE
RS\\Desktop\\jaas_
keytab.config;KrbConfigFilePath=KrbConfigFilePath=C:\\Us
ers\\employee.DRIVERS\\Desktop\\krb5.ini
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using Kerberos Credentials in a Connection URL

You can provide your Kerberos credentials to the driver in the connection string.

You must provide the Kerberos principal and the path to the Kerberos configuration file. In addition, you must provide either the path to the keytab file, or the path to the Kerberos cache file.

To configure Kerberos authentication using Kerberos credentials:

1. In the connection URL, set the `AuthenticationType` property to `Kerberos Authentication`.
2. Set the `SSLTrustStorePath` property to the full path of the TrustStore that you want to use.
3. Set the `SSLTrustStorePwd` property to your password for accessing the TrustStore.
4. Set the `KerberosPrincipal` property to the Kerberos principal.
5. Choose one:
   - Set the `KrbKeytabFilePath` property to the full path and name of the Kerberos keytab file.
   - Or, set the `KrbCacheFilePath` property to the full path and name of the Kerberos cache file.
6. Set the `KrbConfigFilePath` property to the full path and name of the Kerberos `krb5.ini` configuration file.

For example, the following configuration URL uses a Kerberos keytab file:

```
jdbc:presto://presto-demo-
cdh:7778;AuthenticationType=Kerberos
Authentication;SSLTrustStorePath=C:\\Program
Files\\Java\\jre1.8.0_
92\\lib\\security\\cacerts;SSLTrustStorePwd=changeit;Kerbero
sPrincipal=host@REALM;KrbKeytabFilePath=C:\\Kerberos\keytab.
krb;KrbConfigFilePath=C:\\Users\\employee.DRIVERS\\Desktop\\
krb5.ini;
```

As another example, the following configuration URL uses a Kerberos cache file:

```
jdbc:presto://presto-demo-
cdh:7778;AuthenticationType=Kerberos
Authentication;SSLTrustStorePath=C:\\Program
Files\\Java\\jre1.8.0_
92\\lib\\security\\cacerts;SSLTrustStorePwd=changeit;Kerbero
sPrincipal=host@REALM;KrbCacheFilePath=C:\\Kerberos\ticketca
che;KrbConfigFilePath=C:\\Users\\employee.DRIVERS\\Desktop\\
krb5.ini;
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using LDAP Authentication

You can configure the driver to use the LDAP protocol to authenticate the connection.

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

Note:

If LDAP authentication is enabled, then SSL is automatically enabled. For more information, see Configuring SSL Connections on page 21.

To configure LDAP authentication:

1. In the connection URL, set the `AuthenticationType` property to `LDAP Authentication`.
2. Set the `SSLTrustStorePath` property to the full path of the TrustStore that you want to use.
3. Set the `SSLTrustStorePwd` property to your password for accessing the TrustStore.
4. Set the `UID` property to an appropriate user name for accessing the Presto server.
5. Set the `PWD` property to the password corresponding to the user name you provided.

For example:

```
jdbc:presto://presto-demo-cdh:7778;AuthenticationType=LDAP
Authentication;SSLTrustStorePath=C:\\Program
Files\\Java\\jre1.8.0_
92\\lib\\security\\cacerts;SSLTrustStorePwd=changeit;UID=sim
ba;PWD=simba123
```

Using Password File Authentication

You can configure the driver to use the password file protocol to authenticate the connection.

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

Note:

If Password File authentication is enabled, then SSL is automatically enabled. For more information, see Configuring SSL Connections on page 21.

- Prerequisites on page 19
- Using Password File Credentials in a Connection URL on page 19

Prerequisites

Before you can use Password File authentication with the Simba Presto JDBC Driver, you must do the following:

1. On your Presto server, in the `/etc/presto/config.properties` file, set the following properties:

```
http-server.authentication.type=PASSWORD
http-server.https.enabled=true
http-server.https.port=8443
http-
server.https.keystore.path=/var/lib/presto/security/cace
rt.jks
http-server.https.keystore.key=changeit
```

2. Replace the `password-authenticator.properties` file with a file containing the following:

```
password-authenticator.name=file
file.password-file=/etc/presto/password.db
```

3. Create a `password.db` file in `/etc/presto/`. In the file, specify a user name and password separated by a colon (`:`). To include multiple pairs of user names and passwords, put each pair on a separate line. Passwords must be encoded with the bcrypt format. Password files utilizing the bcrypt format can be created using the htpasswd utility from the Apache HTTP Server.

Using Password File Credentials in a Connection URL

1. In the connection URL, set the `AuthenticationType` property to `Password File Authentication`.
2. Set the `SSLTrustStorePath` property to the full path of the TrustStore that you want to use.
3. Set the `SSLTrustStorePwd` property to your password for accessing the TrustStore.

4. Set the `UID` property to the user name from the `password.db` stored in the Presto server.
5. Set the `PWD` property to the corresponding password from the `password.db` stored in the Presto server.

For example:

```
jdbc:presto://presto-
ldap.sentest.com:8443;AuthenticationType=Password File
Authentication;Catalog=hive;Schema=sen;SSLTrustStorePath=D:\
\PrestoCert\\cacert.jks;SSLTrustStorePwd=changeit;UID=test;P
WD=presto
```

Configuring SSL Connections

Note:

In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

If you are connecting to a Presto server that has SSL enabled, you can configure the driver to connect to an SSL-enabled socket. When connecting to a server over SSL, the driver uses one-way authentication to verify the identity of the server. Before configuring SSL in the driver, make sure that you have a TrustStore containing a signed, trusted SSL certificate for verifying the identity of the server.

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

Note:

If Kerberos, LDAP, or Password File authentication is enabled, then SSL is automatically enabled. Make sure to configure the driver to use an appropriate TrustStore.

To configure an SSL connection:

1. If you are not using Kerberos, LDAP, or Password File authentication, then set the `SSL` property to `1`.
2. Set the `SSLTrustStorePath` property to the full path of the TrustStore that you want to use.
3. Set the `SSLTrustStorePwd` property to your password for accessing the TrustStore.

For example:

```
jdbc:presto://192.168.203.141:8080;SSL=1;SSLTrustStorePath=C
:\\Documents\\Presto_
TrustCerts.jks;SSLTrustStorePwd=simba123
```

To configure a two-way SSL connection:

1. If you are not using Kerberos, LDAP, or Password File authentication, then set the `SSL` property to `2`.

2. Set the `SSLKeyStorePath` property to the full path of the KeyStore that you want to use.
3. Set the `SSLKeyStorePwd` property to your password for accessing the KeyStore.

For example:

```
jdbc:presto://192.168.203.141:8080;SSL=2;SSLKeyStorePath=C:\
\Documents\\Presto_KeyCerts.jks;SSLKeyStorePwd=simba123
```

Configuring Proxy Connections

You can configure the driver to connect through a proxy server instead of connecting directly to the Presto service. When connecting through a proxy server, the driver supports basic authentication.

You provide the configuration information to the driver in the connection URL. For more information about the syntax of the connection URL, see Building the Connection URL on page 12.

To configure a proxy connection:

1. Set the `UseProxy` property to `1`.
2. Set the `ProxyHost` property to the IP address or host name of your proxy server.
3. Set the `ProxyPort` property to the number of the TCP port that the proxy server uses to listen for client connections.
4. If the proxy server requires authentication, do the following:
   a. Set the `ProxyAuth` property to `1`.
   b. Set the `ProxyUID` property to your user name for accessing the server.
   c. Set the `ProxyPWD` property to your password for accessing the server.

For example, the following connection URL uses a proxy server with authentication:

```
jdbc:presto://192.168.203.141:8080;UseProxy=1;ProxyHost=192.168.55.3;ProxyPort=189;ProyAuth=1;ProxyUID=skroob;ProxyPWD=12345
```

Configuring Virtual Private Cloud (VPC) Services

You can configure the driver to connect to Presto using a Virtual Private Cloud (VPC) service. To do this, use the following connection properties:

- `SocketFactory`
- `SocketFactoryArg`
- `DnsResolver`
- `DnsResolverArg`

The `SocketFactory` property extends `javax.net.SocketFactory`, and the `DnsResolver` property implements `com.interfaces.networking.CustomDnsResolver`. The `SocketFactoryArg` and `DnsResolverArg` properties pass any required arguments to these services.

The properties in the following instructions are all optional, and only need to be used if you are connecting through the relevant service.

To configure the driver to use a VPC:

1. If necessary, set the `SocketFactory` property to the fully-qualified class path for a class that extends `javax.net.SocketFactory` and provides the socket factory implementation.
2. If necessary, set the `SocketFactoryArg` to a string argument to pass to the constructor of the class indicated by the `SocketFactory` property.
3. If necessary, set the `DnsResolver` property to the fully-qualified class path for a class that extends the `DnsResolver` interface for the driver, `com.interfaces.networking.CustomDnsResolver`.
4. If necessary, set the `DnsResolverArg` to a string argument to pass to the constructor of the class indicated by the `DnsResolver` property.

For example, the following connection URLs show how to connect to data sources using the supported VPCs.

Using SocketFactory:

```
jdbc:presto://192.168.203.141:8080;User=simba;SocketFactory=
com.simba.junit.jdbc.utils.CustomSocketFactory;SocketFactory
Arg=Args;
```

Using DnsResolver:

```
jdbc:presto://192.168.203.141:8080;User=simba;DnsResolver=com.simba.junit.jdbc.utils.TestDnsResolver;DnsResolverArg=agrs;
```

Using both SocketFactory and DnsResolver:

```
jdbc:presto://192.168.203.141:8080;User=simba;DnsResolver=com.simba.junit.jdbc.utils.TestDnsResolver;DnsResolverArg=Agrs;SocketFactory=com.simba.junit.jdbc.utils.CustomSocketFactory;SocketFactoryArg=Args;
```

Configuring Logging

To help troubleshoot issues, you can enable logging in the driver.

! Important:

Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

The settings for logging apply to every connection that uses the Simba Presto JDBC Driver, so make sure to disable the feature after you are done using it.

In the connection URL, set the `LogLevel` key to enable logging at the desired level of detail. The following table lists the logging levels provided by the Simba Presto JDBC Driver, in order from least verbose to most verbose.

LogLevel Value	Description
`0`	Disable all logging.
`1`	Log severe error events that lead the driver to abort.
`2`	Log error events that might allow the driver to continue running.
`3`	Log events that might result in an error if action is not taken.
`4`	Log general information that describes the progress of the driver.
`5`	Log detailed information that is useful for debugging the driver.
`6`	Log all driver activity.

To enable logging:

1. Set the `LogLevel` property to the desired level of information to include in log files.
2. Set the `LogPath` property to the full path to the folder where you want to save log files. To make sure that the connection URL is compatible with all JDBC applications, escape the backslashes (`\`) in your file path by typing another backslash.

For example, the following connection URL enables logging level 3 and saves the log files in the `C:\temp` folder:

```
jdbc:presto://localhost;LogLevel=3;LogPath=C:\\temp
```

3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

The Simba Presto JDBC Driver produces a log file named `DriverLog.log` in the location specified in the `LogPath` property.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

To disable logging:

1. Set the `LogLevel` property to `0`.
2. To make sure that the new setting takes effect, restart your JDBC application and reconnect to the server.

Features

More information is provided on the following features of the Simba Presto JDBC Driver:

- Catalog and Schema Support on page 28
- Parameters on page 28
- Supported Connectors on page 28
- Resource Groups on page 29
- Data Types on page 29
- Complex SQL Data Types on page 31
- Additional SQL Data Types on page 32
- Security and Authentication on page 34

Catalog and Schema Support

The Simba Presto JDBC Driver supports both catalogs and schemas to make it easy for the driver to work with various JDBC applications. The driver provides access to all of the schemas/databases that are listed under this catalog, ensuring compatibility with standard BI tools.

Parameters

A parameterized query contains placeholders that are used for parameters. The values of those parameters are supplied at execution time.

The Simba Presto JDBC Driver fully supports parameterized queries.

Supported Connectors

The Simba Presto JDBC Driver supports the following connectors:

- MySQL
- PostgreSQL
- Hive
- Cassandra

Resource Groups

Resource groups are a Simba Presto JDBC Driver feature that allows administrators to control resource usage and query scheduling.

To use resource groups, in the connection URL, define either the `APPLICATIONNAME` or `CLIENTTAGS` properties, and the `ApplicationNamePrefix` property if required. If the Presto server has a resource group that selects for those values, then the queries are executed according to the policies defined for that resource group.

Alternatively, you can set the `APPLICATIONNAME` and `CLIENTTAGS` properties on the server side. To do this, call `setClientInfo()` using the `APPLICATIONNAME` and `CLIENTTAGS` keywords.

! Important:

When you call `setClientInfo()`, make sure that the `APPLICATIONNAME` and `CLIENTTAGS` keywords are in capital letters.

For example:

```
java.util.Properties pr = new java.util.Properties();
pr.put("APPLICATIONNAME", "Simba");
pr.put("CLIENTTAGS", "Presto, ClientApp" );
conn.setClientInfo(pr);
```

Or, as another example:

```
conn.setClientInfo("APPLICATIONNAME", "simba");
conn.setClientInfo("CLIENTTAGS", "Presto, Simba");
```

For more information, see the following:

- APPLICATIONNAME on page 36
- ApplicationNamePrefix on page 37
- CLIENTTAGS on page 38

Data Types

The Simba Presto JDBC Driver supports many common SQL and Java data types, and converts between them.

The following table lists the supported data type mappings.

SQL Type	Java Type
ARRAY	ARRAY
BIGINT	BIGINT
BOOLEAN	BOOLEAN
CHAR	CHAR
DATE	DATE
DECIMAL	DECIMAL
DOUBLE	DOUBLE
FLOAT **Note:** Deprecated in Teradata Presto Server 0.152-t and later.	REAL
HYPERLOGLOG	JAVA_OBJECT
INTEGER	INTEGER
INTERVAL DAY TO SECOND	VARCHAR
INTERVAL YEAR TO MONTH	VARCHAR
IPADDRESS	JAVA_OBJECT
IPPREFIX	JAVA_OBJECT
JSON	VARCHAR
KHYPERLOGLOG	JAVA_OBJECT
MAP	JAVA_OBJECT
P4HYPERLOGLOG	JAVA_OBJECT

SQL Type	Java Type
QDIGEST	JAVA_OBJECT
REAL **Note:** On Teradata Presto Server, only supported in versions 0.152-t and later.	REAL
ROW	JAVA_OBJECT
SMALLINT	SMALLINT
TIME	TIME
TIME(P)	TIME
TIME WITH TIME ZONE	VARCHAR
TIME(P) WITH TIME ZONE	VARCHAR
TIMESTAMP	TIMESTAMP
TIMESTAMP(P)	TIMESTAMP
TIMESTAMP WITH TIME ZONE	VARCHAR
TIMESTAMP(P) WITH TIME ZONE	VARCHAR
TINYINT	TINYINT
UUID	JAVA_OBJECT
VARBINARY	VARBINARY
VARCHAR	VARCHAR

Complex SQL Data Types

The driver supports the complex SQL data types ARRAY, MAP, and ROW. Nested complex data types are also supported, such as ARRAY[MAP] or ROW(ARRAY

[INTEGER]).

Data stored in these types can also be bound in prepared statements.

For example, to pass ARRAY data in a prepared statement, use the following syntax:

```
Array intArray = connection.createArrayOf("SQL_INTEGER", new
Integer[]{5,6});
preparedStatement.setArray(i, intArray);
```

As another example, to pass MAP data in a prepared statement, use the following syntax:

```
HashMap<Integer, Long> hmap = new HashMap<>();
hmap2.put(3, (long) 5);
hmap2.put(4, (long) 6);
preparedStatement.setObject(1, hmap, Types.JAVA_OBJECT);
```

And as another example, to pass ROW data in a prepared statement for the following query:

INSERT INTO hive.default.rowtable VALUES ROW(CAST(ROW(REAL '2.3', DOUBLE '4.9') AS ROW(x REAL, y DOUBLE)))"

Use the following syntax:

```
LinkedHashMap<Object, Object> linkedHashMap = new
LinkedHashMap<>();
linkedHashMap.put("x", (float)2.3);
linkedHashMap.put("y", 4.9);
preparedStatement = connection.prepareStatement("insert into
hive.default.rowtable values row(?)");
preparedStatement.setObject(1, lhmp, Types.JAVA_OBJECT);
```

Additional SQL Data Types

The driver supports additional SQL data types, such as IPADDRESS, IPPREFIX, UUID, HYPERLOGLOG, KHYPERLOGLOG, P4HYPERLOGLOG, and QDIGEST.

IPADDRESS, IPPREFIX, and UUID data can also be bound in prepared statements. HYPERLOGLOG, KHYPERLOGLOG, P4HYPERLOGLOG, and QDIGEST are not supported with parameterized queries in prepared statements.

For information about additional data types, see the following:

- IPADDRESS, IPPREFIX, and UUID on page 33
- Other Data Types on page 33

IPADDRESS, IPPREFIX, and UUID

For example, to pass IPADDRESS data in a prepared statement, use the following syntax:

```
preparedStatement.setObject(1, "10.54.3.10", Types.JAVA_
OBJECT);
```

For example, to pass IPPREFIX data in a prepared statement, use the following syntax:

```
preparedStatement.setObject(1, "10.54.3.10/24", Types.JAVA_
OBJECT);
```

For example, to pass UUID data in a prepared statement, use the following syntax:

```
preparedStatement.setObject(1, "12151fd2-7586-11e9-8f9e-
2a86e4085a59", Types.JAVA_OBJECT);
```

Other Data Types

Other SQL data types that are mapped to the Java SQL type VARCHAR are returned as a STRING representation of the returned value.

For example:

INTERVAL DAY TO SECOND
Query: SELECT INTERVAL '2' DAY.
Driver Result: A STRING with the value 2 00:00:00.000.

INTERVAL YEAR TO MONTH
Query: SELECT INTERVAL '3' MONTH.
Driver Result: A STRING with the value 0-3.

TIMESTAMP WITH TIMEZONE
Query: SELECT TIMESTAMP '2001-08-22 03:04:05.321 America/Los_Angeles'.
Driver Result: A STRING with the value 2001-08-22 03:04:05.321 America/Los_Angeles.

TIME WITH TIMEZONE
Query: SELECT TIME '01:02:03.456 America/Los_Angeles'.
Driver Result: A STRING with the value 01:02:03.456 America/Los_Angeles.

JSON
Query: SELECT CAST('{\"id\": 1, \"name\": \"test\"}' as json).
Driver Result: A STRING with the value "{\"id\": 1, \"name\": \"test\"}".

Security and Authentication

To protect data from unauthorized access, some Presto data stores require connections to be authenticated with user credentials and the SSL protocol. The Simba Presto JDBC Driver provides full support for these authentication protocols.

Note:

In this documentation, "SSL" indicates both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports industry-standard versions of TLS/SSL.

The driver provides a mechanism that enables you to authenticate your connection using the Kerberos protocol or the LDAP protocol. For detailed configuration instructions, see Configuring Authentication on page 13.

Additionally, the driver supports the following types of SSL connections:

- One-way authentication
- Two-way authentication

It is recommended that you enable SSL whenever you connect to a server that is configured to support it. SSL encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For detailed configuration instructions, see Configuring SSL Connections on page 21.

The SSL version that the driver supports depends on the JVM version that you are using. For information about the SSL versions that are supported by each version of Java, see "Diagnosing TLS, SSL, and HTTPS" on the Java Platform Group Product Management Blog: https://blogs.oracle.com/java-platform-group/entry/diagnosing_tls_ssl_and_https.

Note:

The SSL version used for the connection is the highest version that is supported by both the driver and the server, which is determined at connection time.

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Simba Presto JDBC Driver.

You can set configuration properties using the connection URL. For more information, see Building the Connection URL on page 12.

Note:

Property names and values are case-sensitive.

AllowHostNameCNMismatch

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether the driver requires the name of the CA-issued SSL certificate to match the host name of the Presto server.

- `0`: The driver requires the names to match.
- `1`: The driver allows the names to mismatch.

Note:

This property is applicable only when SSL connections are enabled.

AllowMetadataFromMultipleCatalogs

Default Value	Data Type	Required
true	Boolean	No

Description

This option specifies whether metadata is retrieved from all catalogs when the driver makes a call to SQLTables or SQLColumns.

- `true`: The driver retrieves metadata from all catalogs when making calls to SQLTables or SQLColumns.
- `false`: The driver only retrieves metadata from the specified catalog when making calls to SQLTables or SQLColumns.

Note:

- If this option is disabled, you must specify a catalog to make calls to SQLTables or SQLColumns. You can specify a catalog in the call to SQLTables or SQLColumns, or in the `Catalog` connection property.
- Disabling this option may improve driver performance.

AllowSelfSignedServerCert

Default Value	Data Type	Required
`0`	Integer	No

Description

This property specifies whether the driver allows the server to use self-signed SSL certificates.

- `1`: The driver allows self-signed certificates.

  Important:

  When this property is set to `1`, SSL verification is disabled. The driver does not verify the server certificate against the trust store, and does not verify if the server's host name matches the common name in the server certificate.

- `0`: The driver does not allow self-signed certificates.

Note:

This property is applicable only when SSL connections are enabled.

APPLICATIONNAME

Default Value	Data Type	Required
None	String	No

Description

Set this property to an application flag that you want to apply to the queries sent by this driver. If the application flag has been specified in a Presto resource group, then the queries are run according to the policies defined in that resource group.

To set this property on the server side, call `setClientInfo()` using the `APPLICATIONNAME` keyword.

ApplicationNamePrefix

Default Value	Data Type	Required
None	String	No

Description

Use this property to apply any required prefixes to the `APPLICATIONNAME` property. For more information, see APPLICATIONNAME on page 36.

AuthenticationType

Default Value	Data Type	Required
`No Authentication`	String	No

Description

This option specifies the authentication mechanism to use, if any:

- `No Authentication`: The server does not use any authentication.
- `Kerberos Authentication`: The server uses Kerberos authentication.
- `LDAP Authentication`: The server uses LDAP authentication.
- `Password File Authentication`: The server uses password file authentication.

Note:

- If either Kerberos Authentication or LDAP Authentication are specified, SSL is automatically enabled.
- This option replaces and supersedes the deprecated `enableKerberos` option.

Catalog

Default Value	Data Type	Required
None	String	No

Description

The current catalog context for all requests against the server.

ClientInfo

Default Value	Data Type	Required
None	String	No

Description

Set this property to add extra information about the client.

For example:

```
ClientInfo=PrestoJDBCDriver
```

CLIENTTAGS

Default Value	Data Type	Required
None	String	No

Description

Set this property to a comma-separated list of resource group tags that you want to apply to the queries sent by this driver. If the tags have been specified in a Presto resource group, then the queries are run according to the policies defined in that resource group.

To set this property on the server side, call `setClientInfo()` using the `CLIENTTAGS` keyword.

ConnectionTest

Default Value	Data Type	Required
`1`	Integer	No

Description

This option specifies whether the driver should automatically attempt to test the connection by contacting the server while establishing the connection.

- `1`: The driver automatically tests the connection while establishing the connection.
- `0`: The driver does not automatically test the connection.

Note:

- Disabling this option may improve driver performance.
- If this option is disabled, you should specify the version of the Presto server in the `ServerVersion` configuration option (see ServerVersion on page 49).

CredDelegation

Default Value	Data Type	Required
`false`	String	No

Description

This option specifies whether the driver delegates credentials when using Kerberos authentication.

- `true`: The driver delegates credentials for Kerberos authentication to other applications.
- `false`: The driver does not delegate credentials.

DbmsName

Default Value	Data Type	Required
PRESTO	String	No

Description

This option specifies the name that the driver reports as the DBMS product accessed by the driver.

For example, when using SQLWorkBench, setting this property to `TEXT` causes the application to generate catalog metadata calls in which the wildcard characters are escaped. This can result in faster metadata retrieval and improved performance.

Important:

Be aware that setting this property incorrectly may cause unexpected driver behavior.

DisableKerberosReverseDNS

Default Value	Data Type	Required
0	Integer	No

Description

This property specifies whether to disable Kerberos reverse DNS lookup.

- `1`: The driver does not use Kerberos reverse DNS lookup.
- `0`: The driver uses Kerberos reverse DNS lookup.

DnsResolver

Default Value	Data Type	Required
None	String	No

Description

The fully-qualified class path for a class that extends the `DnsResolver` interface provided by the driver, `com.interfaces.networking.CustomDnsResolver`. Using a custom `DnsResolver` enables you to provide your own resolver logic.

DnsResolverArg

Default Value	Data Type	Required
None	String	No

Description

A string argument to pass to the constructor of the class indicated by the `DnsResolver` property.

ExtraCredentials

Default Value	Data Type	Required
None	String	No

Description

Set this property to a comma-separated list of key-value pairs that you want to pass to an external service.

For example, to set the key-value pairs `Hadoop=Presto` and `Driver=Presto`, you would set this property as follows:

```
ExtraCredentials=Hadoop:Presto,Driver:Presto
```

JAASAuthType

Default Value	Data Type	Required
`false`	String	No

Description

This option specifies whether the driver uses a JAAS login configuration file for Kerberos authentication.

- `true`: The driver uses the JAAS login configuration file specified in `KrbJaasFilePath` for Kerberos authentication.
- `false`: The driver does not use a JAAS login configuration file for Kerberos authentication.

KerberosPrincipal

Default Value	Data Type	Required
None	String	Yes, if Kerberos authentication is used and the Kerberos credentials are passed in the connection URL.

Description

The Kerberos principal to use with Kerberos authentication.

KrbCacheFilePath

Default Value	Data Type	Required
None	String	Yes, if all of the following apply: • Kerberos authentication is used. • The Kerberos credentials are passed in the connection URL. • The `KrbKeytabFilePath` property is not set.

Description

The full path and file name of the cache file for Kerberos authentication.

KrbConfigFilePath

Default Value	Data Type	Required
None	String	Yes, if Kerberos authentication is used and the driver does not load the credentials from the Kerberos credential cache.

Description

The full path and file name of the `krb5.ini` configuration file for Kerberos authentication.

KrbJaasFilePath

Default Value	Data Type	Required
None	String	Yes, if `JAASAuthType` is set to `true`.

Description

The full path and file name of the JAAS login configuration file for Kerberos authentication.

KrbKeytabFilePath

Default Value	Data Type	Required
None	String	Yes, if all of the following apply: • Kerberos authentication is used. • The Kerberos credentials are passed in the connection URL. • The `KrbCacheFilePath` property is not set.

Description

The full path and file name of the keytab file for Kerberos authentication.

KrbServiceName

Default Value	Data Type	Required
`HTTP`	String	No

Description

The primary name of the service principal to use for Kerberos authentication.

LogLevel

Default Value	Data Type	Required
0	Integer	No

Description

Use this property to enable or disable logging in the driver and to specify the amount of detail included in log files.

! Important:

Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

The settings for logging apply to every connection that uses the Simba Presto JDBC Driver, so make sure to disable the feature after you are done using it.

Set the property to one of the following numbers:

- `0`: Disable all logging.
- `1`: Enable logging on the FATAL level, which logs very severe error events that will lead the driver to abort.
- `2`: Enable logging on the ERROR level, which logs error events that might still allow the driver to continue running.
- `3`: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.
- `4`: Enable logging on the INFO level, which logs general information that describes the progress of the driver.
- `5`: Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the driver.
- `6`: Enable logging on the TRACE level, which logs all driver activity.

When logging is enabled, the driver produces a log file named `presto.log` in the location specified in the `LogPath` property.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

LogPath

Default Value	Data Type	Required
The current working directory.	String	No

Description

The full path to the folder where the driver saves log files when logging is enabled.

Note:

To make sure that the connection URL is compatible with all JDBC applications, it is recommended that you escape the backslashes (\) in your file path by typing another backslash.

PWD or Password

Default Value	Data Type	Required
None	String	No

Description

The password corresponding to the user name that you provided using the property UID or User on page 53.

The password for LDAP authentication.

ProxyAuth

Default Value	Data Type	Required
0	String	No

Description

This option specifies whether the proxy server that you are connecting to requires authentication.

- `1`: The proxy server that you are connecting to requires authentication.
- `0`: The proxy server that you are connecting to does not require authentication.

ProxyHost

Default Value	Data Type	Required
None	String	No

Description

The IP address or host name of your proxy server.

ProxyPort

Default Value	Data Type	Required
None	Integer	No

Description

The listening port of your proxy server.

ProxyPWD

Default Value	Data Type	Required
None	String	Yes, if connecting through a proxy server that requires authentication.

Description

The password that you use to access the proxy server.

ProxyType

Default Value	Data Type	Required

`0`
or
`HTTP Proxy`
	String	Yes, if connecting to a SOCKS proxy.

Description

This option specifies which type of proxy is to be used. The Default ProxyType is `0` or `HTTP Proxy`. To use a SOCKS proxy, set ProxyType to `1` or `SOCKS Proxy`.

ProxyUID

Default Value	Data Type	Required
None	String	Yes, if connecting through a proxy server that requires authentication.

Description

The user name that you use to access the proxy server.

Schema

Default Value	Data Type	Required
None	String	No

Description

The current schema context for all requests against the server.

SessionProperties

Default Value	Data Type	Required
None	String	No

Description

The session properties to set for the system or catalog, specified as a list of key-value pairs separated by a comma (`,`) and key-values separated by a colon (`:`).

For example, to set the query_max_execution_time and the query_max_run_time properties to 0.2h, you would set this property as follows:

```
SessionProperties=query_max_execution_time:0.2h,query_max_
run_time:0.2h;
```

ServerVersion

Default Value	Data Type	Required
`0.155`	String	No

Description

This option specifies the version of the Presto server that the driver connects to, in the event that the driver cannot automatically detect the server version.

Note:

If `ConnectionTest` is set to `0`, this option should be set to the version of the Presto server that is being used.

SocketFactory

Default Value	Data Type	Required
None	String	No

Description

The fully-qualified class path for a class that extends `javax.net.SocketFactory` and provides the socket factory implementation. Using a custom SocketFactory enables you to customize the socket that the driver uses.

SocketFactoryArg

Default Value	Data Type	Required
None	String	No

Description

A string argument to pass to the constructor of the class indicated by the `SocketFactory` property.

SSL

Default Value	Data Type	Required
`0`	Integer	No

Description

This property specifies whether the driver communicates with the Presto server through an SSL-enabled socket.

- `1`: The driver connects to SSL-enabled sockets using one-way authentication.
- `2`: The driver connects to SSL-enabled sockets using two-way authentication.
- `0`: The driver does not connect to SSL-enabled sockets.

Note:

- SSL is configured independently of authentication. When authentication and SSL are both enabled, the driver performs the specified authentication method over an SSL connection.
- If `AuthType` is set to either `Kerberos Authentication`, `LDAP Authentication`, or `Password File Authentication` SSL is automatically enabled.

SSLKeyStorePath

Default Value	Data Type	Required
None	String	Yes, if two-way authentication is enabled.

Description

The full path of the Java KeyStore containing the certificate for verifying the client.

SSLKeyStorePwd

Default Value	Data Type	Required
None	String	Yes, if two-way authentication is enabled.

Description

The password that you use when checking the integrity of the KeyStore.

SSLKeyStoreType

Default Value	Data Type	Required
JKS	String	No

Description

The type of SSL KeyStore key. For example, JKS, PKCS12, etc.

SSLTrustStorePath

Default Value	Data Type	Required
The system Java trust store path.	String	No

Description

The full path of the Java TrustStore containing the server certificate for one-way SSL authentication.

See also the property SSLTrustStorePwd on page 52.

SSLTrustStorePwd

Default Value	Data Type	Required
The system Java trust store password	String	No

Description

The password for accessing the Java TrustStore that you specified using the property SSLTrustStorePath on page 51.

SSLTrustStoreType

Default Value	Data Type	Required
JKS	String	No

Description

The type of SSL TrustStore key. For example, JKS, PKCS12, etc.

TimeZoneID

Default Value	Data Type	Required
None	String	No

Description

This option specifies the local time zone that the driver uses. If this value is not specified, the driver uses the system's current time zone ID.

Valid values for this option are specified in the IANA Time Zone Database. For a complete list of time zones, see https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.

UID or User

Default Value	Data Type	Required
`PrestoJDBC_Driver`	String	No

Description

The user name that you use to access the Presto server.

UseProxy

Default Value	Data Type	Required
`0`	String	No

Description

This option specifies whether the driver connects through a proxy server.

- `1`: The driver connects to the Presto server through a proxy server.
- `0`: The driver connects to the Presto server directly.

enableKerberos (deprecated)

Default Value	Data Type	Required
`False`	Boolean	No

Description

This option is deprecated. Use `AuthenticationType` instead (see AuthenticationType on page 37).

This option specifies whether the driver uses Kerberos authentication.

Note:

If Kerberos is enabled, SSL is enabled automatically.

Third-Party Trademarks

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

All other trademarks are trademarks of their respective owners.